A Project Report On

Smart Assistive Device: AI-Driven Object Recognition and Wireless Connectivity for the Visually Impaired

Dept. Of CSE,

M S Ramaiah Institute of Technology,

Banglore

ABSTRACT

In a world where over 2.2 billion individuals grapple with partial or complete blindness, relying on sight for 80% of environmental information poses significant challenges. The advancement of Artificial Intelligence (AI) and Internet of Things (IoT) technologies can enhance the lives of visually impaired individuals. The conventional aids like guide dog and canes have significant negative effects that increase the risk of falls. The proposed model addresses the limitations of existing aids which do not provide real-time information about immediate surroundings and are often costly and less functional. The proposed model is to develop a cost-effective Intelligent Navigation System integrating object detection and AI-based real-time object detection, providing instant voice feedback. The Smart Cane is designed and developed using MobileNetV2 deep learning algorithm and YOLO v3. Smart cane detects real-time objects and gives instant voice feedback for clear communication of detected objects. Cane is supported with panic button to alert guardians during emergencies by sharing the user's location. A companion mobile application is also developed to track the user's location and receive alerts, ensuring comprehensive support and safety for visually impaired individuals. MobileNet is much faster and has adequate accuracy, which makes it more appropriate for applications in real time object detection when compared to YOLO v3. The Smart Cane significantly improves the independence and mobility of visually impaired individuals, providing a reliable and affordable navigation aid. This innovative assistive technology promises to enhance the quality of life for visually impaired individuals, offering them greater autonomy and ease of navigation in their daily lives.

TABLE OF CONTENTS

Declaration			***iii***
Acknowledgements			***iv***
Abstract			***v***
Table of Contents			***vi***
List of Figures			***viii***
List of Tables			***x***
List of Abbreviations			***xi***
Chapter No.		**Title**	**Page No.**
1	**INTRODUCTION**		**1**
	1.1	General Introduction	1
	1.2	Problem Statement	2
	1.3	Objectives of the Project	3
	1.4	Project deliverables	3
	1.5	Current Scope	3
	1.6	Future Scope	4
2	**PROJECT ORGANIZATION**		**5**
	2.1	Software Process Models	5
	2.2	Project Planning and Management	6
3	**LITERATURE SURVEY**		**8**
	3.1	Introduction	8
	3.2	Related Works with the Citation of the References	9
	3.3	Conclusion of Survey	18
4	**SOFTWARE REQUIREMENT SPECIFICATIONS**		**19**
	4.1	Introduction	19
	4.2	Product Overview	19
	4.3	External Interface Requirements	21
		4.3.1 User Interfaces	21
		4.3.2 Hardware Interfaces	21
		4.3.3 Software Interfaces	21
		4.3.4 Communication Interfaces	21

	4.4	Functional Requirements	22
	4.5	Non Functional Requirements	22
	4.6	Hardware Requirements	23
	4.7	Software Requirements	23
	4.8	Use case description with Use case diagram	23
5	**DESIGN**		**24**
	5.1	Introduction	24
	5.2	High level Design	25
	5.3	Low Level Design	27
	5.4	Conclusion	30
6	**IMPLEMENTATION**		**31**
	6.1	Tools Introduction	31
	6.2	Technology Introduction	42
	6.3	Overall view of the project in terms of implementation	46
	6.4	Information about the implementation of Modules	46
	6.5	Conclusion	49
7	**TESTING**		**50**
	7.1	Introduction	50
	7.2	Testing Tools and Environment	51
	7.3	Test Cases Design and Execution	52
	7.4	Experiential Result	53
8	**CONCLUSION and SCOPE FOR FUTURE WORK**		**67**
	REFERENCES		**68**
	APPENDIX		

LIST OF FIGURES

Figure Number	Title	Page Number
1.1	Types of Assistive Devices for Visually Impaired	1
2.1	Agile Model	5
2.2	Gantt Chart	7
3.1	Taxonomy of the Literature Survey Papers	8
4.8	Use case diagram	23
5.1	Object Detection Process	26
5.2	System Architecture for the Proposed System	27
5.3	Sequence Diagram of SmartCane	28
5.4	Activity Diagram of SmartCane	29
5.5	Graphical User Interface for Object Detection	30
6.1	RealVNC Server	31
6.2	RealVNC Application – Platform to run Raspberry Pi	32
6.3	Firebase Realtime Database Console	34
6.4	Raspberry Pi 4 Model B Mounted with Pi Camera 5MP	44
7.1	Accuracy Analysis	56
7.2	Analysis of Multiply-Adds (Millions)	56
7.3	Analysis of Model Size (MB)	56
7.4	SmartCane	58
7.5	Object Detection Code	59

7.6	Vehicle Detection in Object Recognition Module	59
7.7	Pedestrian Detection	60
7.8	Objects Detection-1	60
7.9	Objects Detection-2	61
7.10	User Signup Process for Mobile App Registration	61
7.11	User Login Page with Accessibility Options for Visually Impaired and Parent/Guardian Logins	62
7.12	Login Interface for Visually Impaired Users	62
7.13	Waiting Status in Emergency Situations	63
7.14	Parent Login Process	63
7.15	Location Coordinates of Visually Impaired Person in Panic Displayed in Parent's Mobile App	64
7.16	Parent Viewing Location Coordinates of Visually Impaired Person via Google Maps	64
7.17	Safety Status of Users in Firebase (Value 0 Represents Safety)	65
7.18	User Data Safety Status in Firebase (Value 1 Indicates Unsafe/Panic)	65
7.19	Firebase Database Structure for Visually Impaired Person and Parent Login Data	66
7.20	Authentication Details for Both Parents and Visually Impaired Individuals	66

LIST OF TABLES

Table Number	Title	Page Number
1	Comparison of papers related to Object detection in the Taxonomy	12
2	Comparison of different visual aid devices	17
3	Comparison of Object Detection Algorithms	55

LIST OF ABBREVIATIONS

Abbreviation	Full Form
CNN	Convolutional Neural Network
YOLO	You Only Look Once
gTTS	Google Text-to-Speech
Pyttsx3	Python Text-to-Speech Version 3
NLP	Natural Language Processing
GPS	Global Positioning System
VIP	Visually Impaired Person
SAPI	Speech Application Programming Interface

CHAPTER 1

INTRODUCTION

1.1 General Introduction

People who are not able to use their eyes to see the objects around them are called blind people,and those who are not able to identify tiny details with their eyes are called visually impaired people. According to the WHO, there are more than 2.2 billion people around the world who are either partially or completely ob. Humans perceive around 80% of the information aboutthe environment using their eyes. Therefore, it is very difficult for them to survive in this world. They generally walk on the streets with sticks, dogs, or another person (as shown in Figure 1.1), and their natural senses(commonly referred to as the 6th sense), and thus have difficulty knowing about passing placesand obstacles. Researchers always try to find solutions and try to develop products with their innovative ideas, available technologies and resources like micro-controllers, sensors, cameras, etc. so that they can walk down the streets smoothly and safely.

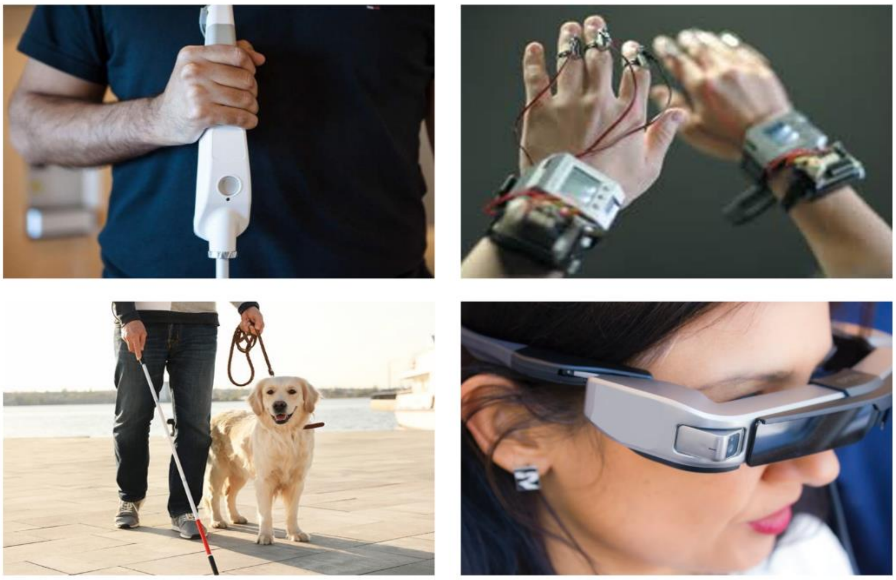


Figure 1.1: Types of Assistive Devices for Visually Impaired

Visually impaired individuals face numerous challenges in navigating their surroundings independently, facing obstacles in their daily life. A SmartCane represents a cutting-edge assistive technology designed to empower individuals with visual impairments, enhancing their ability to navigate their surroundings with increased safety and independence. This innovative device incorporates a sophisticated camera module, enabling it to detect and identify various objects in its vicinity. Through the integration of advanced image recognition technology, the SmartCane provides real-time feedback to users via a voice guidance system.

By leveraging its camera capabilities, the SmartCane not only alerts users to the presence of obstacles but also identifies and communicates the nature of these objects, offering a more comprehensive understanding of the surrounding environment. This real-time object recognition enhances the user's situational awareness and fosters a more confident and secure navigation experience.

At the core of our innovation is a smart cane device seamlessly integrating the Internet of Things (IoT) and Artificial Intelligence (AI). This convergence enables the device not only to detect objects in real-time but also to communicate vital information through a speech output module. By harnessing object detection technology, the smart cane identifies and analyzes objects in the user's path, providing instant auditory feedback to enhance situational awareness.

This technology-driven approach aims to empower visually impaired individuals, offering them greater independence and confidence as they navigate the world around them. In essence, our project seeks to address prevalent issues faced by the visually impaired community, enriching their lives through enhanced mobility and safety.

1.2 Problem Statement

Visually impaired individuals face challenges in navigating their surroundings independently, often encountering obstacles that pose safety risks. Existing solutions lack advanced features Such as real-time, object detection and informative speech output. The main challenge is to address these limitations and develop an intelligent solution that significantly improves the daily mobility experience of the visually impaired.

1.3 Objectives of the project

- To develop a smart assistive device system for object detection and identificationfeatures tailored for visually impaired users.
- To develop a voice feedback module aimed at delivering detected object information ina clear and concise manner.
- To implement a panic button feature that allows users to notify their guardians in theevent of an emergency or danger.
- Develop a companion mobile application that will help parents/guardians to track thevisually impaired location and receive alert messages.

1.4 Deliverables of the Project

The project delivers a Smart cane with a hardware prototype integrating an object detection camera, Raspberry Pi, panic button, and user interface. It includes a robust software system with embedded real-time object detection, speech output, and a user interface for customization. Detailed documentation covering system architecture, user manuals, and technical specifications is also provided, offering a comprehensive solution to enhance the independence and safety of visually impaired individuals.

1.5 Current Scope

The proposed system aims to develop a prototype SmartCane with essential features like objectdetection and speech output, ensuring reliable operation in various environments. By integrating AI, the SmartCane utilizes Raspberry Pi and a camera to offer visually impaired individuals an advanced navigation tool for daily activities, aiding in obstacle avoidance and object identification. The Raspberry Pi, acting as a mini computer, runs diverse programs and operating systems, implementing algorithms like MobileNetV2 for tasks such as object detection. The smart stick enhances user experience through voice feedback mechanisms.

1.6 Future Scope

The future scope of the SmartCane project includes ongoing refinement of object detection algorithms to enhance accuracy, integration with emerging technologies like machine learning to improve object recognition, collaboration with accessibility organizations to collectuser feedback for iterative improvements, and exploration of additional features such as GPS integration for enhanced navigation assistance.

CHAPTER 2

PROJECT ORGANIZATION

Project organization refers to the structure and arrangement of resources, tasks, and responsibilities within a project. It involves establishing a framework that specifies how variouscomponents of a project are coordinated, managed, and executed to achieve project objectivesand deliver desired outcomes. The project organization of the project includes the software process model chosen that is the agile model as referred to in the figure 2.1.

2.1 Software Process Models

Developing a Smart Cane based on object detection requires integrating hardware sensors with software algorithms to identify objects and aid visually impaired individuals in navigating their surroundings safely. Employing an Agile software development model for this project offers flexibility and adaptability, enabling iteration through development cycles and the inclusion of feedback from stakeholders, particularly visually impaired individuals. This model is well-suited to the iterative nature of AI development and emphasizes continuous improvement.

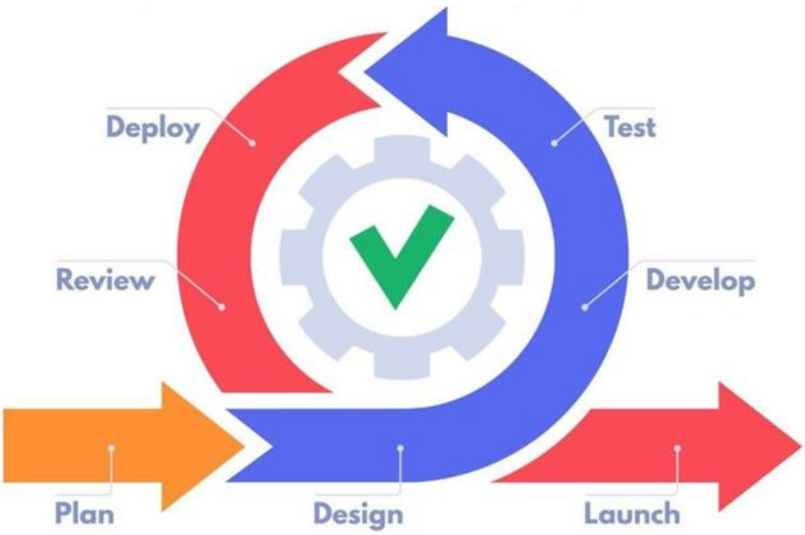


Figure 2.1: Agile Model

2.2 Project Planning and Management

Project planning and management is the process of organizing and overseeing all the tasks and resources required to achieve specific project goals within defined constraints, such as time, budget, and scope. It involves various activities, including defining project objectives, creating a project plan, assigning responsibilities, monitoring progress, and making necessary adjustments to ensure successful project completion.

- Requirements analysis involves consulting visually impaired individuals to understand their needs and defining features such as object detection range, sensitivity, portability, and feedback mechanisms for the smart cane.

- The system design involves architecting the overall structure, integrating hardware components like sensors and processors, selecting suitable object detection algorithms considering accuracy and processing power, and designing an intuitive user interface for feedback delivery such as audio alerts or vibrations.

- During the development phase, software modules for object detection, data processing, navigation algorithms, and user interface are developed, hardware components are integrated with the software using suitable drivers and communication protocols, and comprehensive unit testing for individual modules as well as integration testing for the entire system are implemented.

- Thorough testing in controlled environments assesses object detection accuracy, obstacle avoidance, and path navigation, followed by user testing with visually impaired individuals to gather feedback on usability, comfort, and real-world effectiveness, facilitating system refinement based on test results and user feedback.

The figure 2.2 shows the visual representation for the activities and their status while implementation of the project. It provides a clear timeline view of the project's tasks. The Gantt chart showcases the starting and ending dates each activity, allowing to track progress and identify any potential delays or bottlenecks. The chart also helps in monitoring the status of each activity, whether it is ongoing, completed, or delayed.

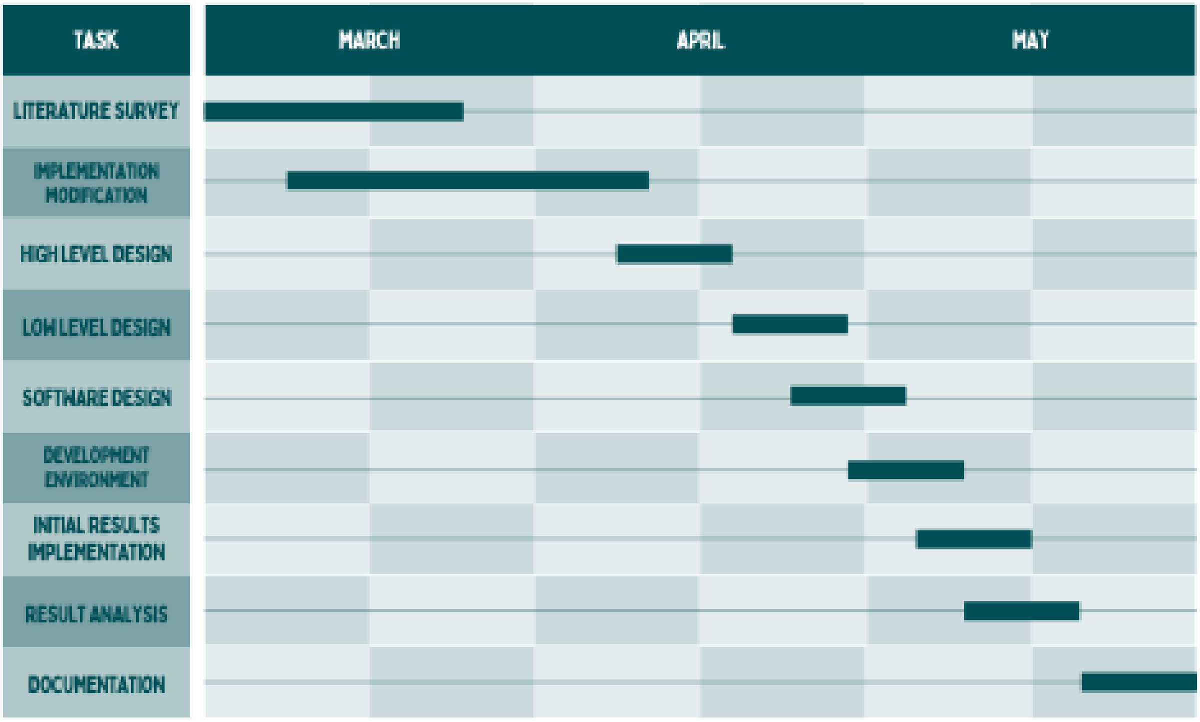


Figure 2.2: Gantt Chart

CHAPTER 3

LITERATURE SURVEY

A literature survey is a thorough examination and synthesis of previous studies and academic publications on a particular subject or research question. It is sometimes referred to as a review of literature or a systematic review. To completely grasp the present state of research and identify any gaps or opportunities for additional inquiry, it entails reviewing and assessing a variety of sources, including books, academic journals, conference papers, and dissertations.

3.1 Introduction

In the last two decades, many technologies have been proposed in order to assist blinds or visually impaired persons to navigate in closed spaces. The literature survey explores a diverse range of assistive technologies aimed at enhancing the independence and safety of visually impaired individuals. From smartphone-based navigation systems employing computer vision and auditory feedback to wearable devices incorporating deep learning for comprehensive assistance, each paper contributes to the evolving landscape of assistive solutions. This chapter illustrates how the current/previous knowledge in the area of the research work, identifies the advantages/disadvantages and the methods adapted in the previous work as shown in the figure 3.1. It presents an up-to-date understanding of the research area and its significance to practice.

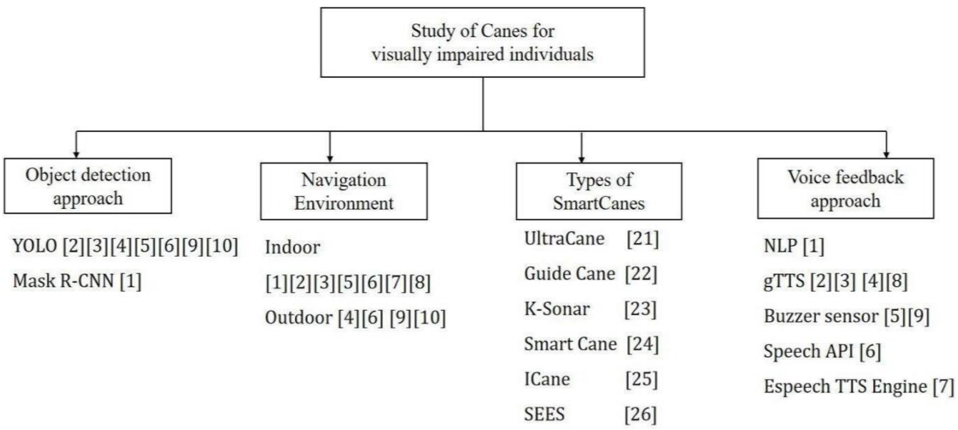


Figure 3.1: Taxonomy of the Literature Survey Papers

3.2 Related works with citation of the references and comparison

In the paper [1], the author addresses the challenges faced by visually impaired individuals in navigating unfamiliar environments by introducing the SMART_EYE system. The literature survey emphasizes the limitations of existing assistive devices and the need for cost-effective alternatives. The study integrates AI and sensor technology in a smart app to enhance obstacle detection and navigation, showcasing the potential for advancements in assistive technology. The proposed model's evaluation and ranking contribute to a basis for future research and development in the field, emphasizing the significance of integrating innovative solutions to improve the quality of life for visually impaired individuals.

In the paper [2], the author presents a smart solution for visually impaired individuals using modern technologies such as deep learning, natural language processing, and IoT. The proposed smart stick integrates a Raspberry Pi, ultrasonic sensor, and camera to enhance navigation. The literature survey emphasizes the limitations of traditional assistance tools and highlights the potential of technology-driven solutions. The study showcases successful implementation, addressing obstacle detection and text recognition, while proposing future enhancements such as GPS integration, voice commands, live text recognition, expanded object recognition, and RFID utilization for further improving the system's capabilities.

In the paper [3], the author introduces a low-cost assistive system for visually impaired individuals, focusing on obstacle detection and ambient environment description using deep learning techniques. The study utilizes TensorFlow object detection API and SSDLite MobileNetV2, optimizing the model with gradient particle swarm optimization. The proposed device, integrated into a head cap, outperforms traditional white canes in efficiency. The research introduces an "ambiance mode" for describing surroundings and evaluates the system's performance on both desktop and Raspberry Pi setups. Future work is suggested to enhance contextual information, expand datasets, implement natural language processing, and explore more powerful hardware options, emphasizing open-source availability for collaborative development.

In the paper [4], the author introduces "The Third Eye," an innovative technology aimed at assisting visually impaired individuals in their daily routines. The proposed device combines AI Assistant, Image Detector using YOLO, sensors, and additional capabilities within eyewear. The study addresses the challenges of travel for the visually impaired by offering a user- friendly, smart guiding gadget that understands vocal commands and provides auditory feedback. This work marks a new era in technology, acting as a companion to ease the hardships faced by visually impaired individuals in navigating complex environments.

In the paper [5], the author addresses the challenges faced by visually impaired individuals in independent mobility. The study employs a deep learning approach, utilizing the convolutional neural network (CNN) architecture AlexNet, along with Raspberry Pi, ultrasonic sensors, a camera, and auditory feedback components. The proposed system demonstrates impressive validation accuracy (99.56%) for real-time object recognition. The work contributes to a high-level framework for interpreting semantic entities, providing a comprehensive solution for visually impaired individuals to navigate both indoor and outdoor environments. Future enhancements include GPS integration for outdoor navigation and tracking, aiming to further improve the system's usability and effectiveness.

In the paper [6], the author introduces a smart and intelligent system designed for visually impaired persons (VIPs) to enhance mobility and safety. The work leverages technology and innovative devices to provide real-time navigation using an automated voice and a web-basedapplication for family tracking. The MobileNet architecture is employed for low-power devicecompatibility, achieving an 83.3% accuracy in object detection and recognition. The proposedsystem, tailored to VIPs' day-to-day requirements, outperforms existing devices, scoring 9.1/10 in a quantitative comparative analysis. Future considerations include additional criteria for feature selection and user feedback integration to further enhance device performance.

In the paper [7], the author addresses the challenges faced by visually impaired individuals, particularly in low- and middle-income regions, where risks to their safety are heightened. Theresearch focuses on developing an inexpensive, mobile, and life-enhancing assistive device using smart technology like sensors, cameras, and voice commands. Deep learning and TensorFlow Lite are employed for obstacle detection, specifically distinguishing between "Threat" and "Neutral" persons. The Raspberry Pi serves as the main microcontroller, facilitating the integration of sensors and hosting the deep learning model. The research concludes with insights into the prototype's effectiveness, highlighting the need for a more diverse dataset, improved response time, and consideration of real-time object detection for enhanced protection.

In the paper [8], the author introduces a novel approach using deep learning for a wearable model to assist blind and visually impaired individuals in recognizing and navigating their environments. The study employs the YOLOv5 algorithm, combining a 3D-designed wearable device and a vision-based smart stick to enhance navigation accuracy. The proposed model achieves an 89.24% overall accuracy, outperforming previous models and addressing deficiencies by incorporating a single neural network for bounding box predictions. The research emphasizes the potential societal impact by enabling visually impaired individuals tonavigate effectively in both indoor and outdoor environments.

In the paper [9], the author presents a smart cane with a face recognition system designed to aid visually impaired individuals in recognizing human faces. The system integrates a camera mounted on glasses for real-time face detection, ultrasonic sensors for obstacle detection, a water sensor for puddle detection, and a GPS module for location monitoring. The conclusionhighlights the significant role of technology in improving the lives of visually impaired individuals. The future scope suggests advancements such as using image processing for obstacle volume assessment and incorporating high-range ultrasonic detectors for further applications, emphasizing the affordability and accessibility of the proposed gadget.

In the paper [10], the author presents a Smart Blind Walking Stick using Arduino designed to assist visually impaired individuals in navigating urban environments. The stick employs ultrasonic sensors for obstacle detection, providing vibration and voice feedback to the user. The system is controlled by Arduino and incorporates a moisture sensor to detect water obstacles. Additionally, GPS and GSM modules enable location sharing with family membersvia SMS. The conclusion highlights system limitations, including the ultrasonic range and usertrust factors, while emphasizing its potential as a low-cost solution for the visually impaired.

Table 1: Comparison of Papers Related to Object Detection in the Taxonomy.

Ref no.	Hardware used for Object detection	Navigation Environment	Distance covered	Feedback System	Approach used for Voice module
[1]	Raspberry Pi module,Pi Camera module forcapturing images	Indoor and Outdoor	Not stated (closest obstracle)	Voice commands	Natural Language Processing (NLP)
[2]	Raspberry Pi 3 B+ module for placing Pi Camera for captureimages of the surroundings	Outdoor	1.5 meters	Audio feedback	Google Text- to- Speech (gTTS)
[3]	Raspberry Pi 4 ModelB+, Raspberry Pi camera v2, 10000 mAh power bank	Indoor and Outdoor	Not stated	Audio feedback	Google Text-to-Speech (gTTS)
[4]	Raspberry Pi 2 board, Pi camera to detect objects and obstracles	Outdoor	Not stated	Audio feedback	Google Text-to-Speech (gTTS)
[5]	Raspberry Pi 3 Model B, Pi camera and ultrasonic sensor to measure the distance of an object	Indoor and Outdoor	40 centimeters	Voice feedback	Buzzer sensor

[6]	Raspberry Pi 3 Model B, GPS module, Camera and headphones	Indoor and Outdoor	Not stated	Voice feedback	Speech API (SAPI) is used to generate audio
[7]	Raspberry Pi 4, Ultrasonic sensor, Pi Camera Module 5M 1080p, Anker Soundcore mini (Speaker)	Indoor and Outdoor	1.2 meters	Voice command	Espeak TTS engine
[8]	Raspberry Pi 3, Ultrasonic Sensor, GPS Module, Camera, headphone, power supply	Indoor	1 - 1.5 meters	Voice command	Text to speech (TTS) generator
[9]	Arduino Uno, Node MCU, Ultrasonic Sensors, Moisture Sensor	Outdoor	Not stated	Audio feedback	Buzzer sensor
[10]	Arduino Uno, Ultrasonic sensor, Moisture sensor - detect water comes in blind person's path.	Indoor and Outdoor	2- 400 centimeters	Haptic and voice feedback	Android app which reads the text out loud from the inbuilt speakers of the phone.

In the paper [11], the author introduces DeepNAVI, a smartphone-based navigation assistant designed for visually impaired individuals. Focusing on portability and convenience, the system leverages deep learning to provide detailed information about detected obstacles, including type, position, distance, motion status, and scene information. The study includes a pilot test validating the system's practicality, offering a promising solution for real-time navigation assistance. Future enhancements may involve multimodal output, an enriched voice assistant, and the incorporation of reinforcement learning for continuous model improvement.

In the paper [12], the author introduces a Blind Assistive System utilizing the You Only Look Once (YOLO) algorithm and OpenCV library for object detection and recognition in images and video streams. The study focuses on aiding visually impaired individuals by identifying various objects, including persons, chairs, oven, pizza, mugs, bags, seats, etc. The system, implemented in Python, demonstrates satisfactory performance in detecting and recognizing objects in the environment. The research emphasizes the potential for future collaboration and development to address diverse challenges and improve applications in various fields.

In the paper [13], the author conducts a comprehensive comparative analysis of assistive devices for visually impaired persons (VIPs). The study categorizes these devices based on functionality and working principles, addressing tasks like object/obstacle detection, navigation, and mobility. The analysis highlights the attributes, challenges, and limitations of existing techniques, offering a score-based quantitative assessment. The conclusion emphasizes the need for an intelligent system covering essential features to better support VIPs, providing valuable insights for researchers and scientists in the field.

In the paper [14], the author introduces a wearable device for visually impaired persons (VIPs) using a Raspberry Pi, camera module, and a pre-trained convolutional neural network integrated into smart glasses. The system offers real-time auditory or haptic feedback for object identification and distance estimation, addressing limitations in existing assistive technologies. The conclusion emphasizes the project's complexity, challenges, and the importance of knowledge in computer vision, machine learning, and embedded systems. The study underscores the significance of time management and highlights the author's learning experiences, providing insights into the development process.

In the paper [15], the author introduces a CNN-based assistive device for visually impaired individuals, utilizing on-device inference for real-time and accurate object recognition. The study addresses the challenges of running state-of-the-art object detectors on devices with limited resources and explores lightweight models and compression techniques for improved speed/accuracy trade-offs. The conclusion emphasizes the benefits of on-device inference, including lower latency, increased security, privacy, reliability, and reduced costs. The future work aims to incorporate temporal information from videos to predict movements and prevent collisions.

In the paper [16], the author addresses the challenges faced by visually impaired individuals and proposes an innovative solution using a low-cost smart cane with obstacle detection and identification features. The research incorporates deep learning, computer vision, embedded systems, and wireless communications, aiming to provide an all-in-one device for enhanced navigation. The cost-effective approach involves the use of low-cost sensors and computing devices. The research highlights the need for continuous improvement, considering potential risks such as cyber security threats and device-related issues.

In the paper [17], the author introduces a live object recognition system as a blind aid, addressing the challenges faced by visually impaired individuals in recognizing people or objects. The proposed computer vision-based navigation system aims to enhance independence for individuals with total or partial blindness. The research highlights the efficiency and accessibility of the application, emphasizing its advantages over existing solutions. Future enhancements are suggested, including the addition of GPS for navigation, proximity sensors, integrated reading mechanisms, and battery consumption optimization.

In the paper [18], the author introduces a real-time system utilizing Raspberry Pi, camera, and ultrasonic sensors to assist blind and visually impaired individuals. The proposed wearable attachment employs deep learning and computer vision for object and face recognition, text reading, and obstacle detection, providing real-time audio feedback to enhance the user's independence. While user experimentation is pending, future improvements are suggested, including enhanced software for text detection, face recognition through a mobile application, and improvements in object detection using techniques like Generative Adversarial Networks (GANs). The paper also proposes advancements in distance measurement using multiple ultrasonic sensors and explores communication with other systems, emphasizing the importance of reducing the glasses' weight and addressing environmental challenges.

In the paper [19], the author conducts a detailed literature review on object detection, covering traditional, two-stage, and one-stage techniques. The study includes analysis of dataset preparation, annotation tools, and performance evaluation metrics. The review addresses various applications, challenges, and future directions in object detection, emphasizing the significance of technology for visually impaired individuals. The paper explores advancements in small object detection, the integration of one-stage and two-stage detectors, and the need for domain transfer and automatic annotation in supervised object detection. Additionally, it discusses challenges related to feature fusion, multi-task learning, multi-source information, and the inadequacy of benchmark datasets.

In the paper [20], the author introduces a smart stick concept designed for the visually impaired, employing Arduino UNO and Voice Module for artificial vision, object detection, water and fire detection, and real-time support. The focus is on providing comprehensive assistance beyond simple mobility. The system utilizes ultrasonic sensors and voice feedback to offer cost- effective navigation, obstacle detection, and environmental information for visually impaired individuals. The conclusion emphasizes the gadget's safety benefits, and future work is proposed to enhance object recognition and charge capacity, incorporating more image databases and advanced algorithms for dynamic image recognition.

Table 2: Comparison of Different Visual Aid Devices

Model	Objective	Parameters	Walking functions	Drawbacks
UltraCane [21]	Obstacle detection at the head and ground levels	Ultrasonic waves	Detection of obstructions in front of the user's head and chest	Costly, large
GuideCane [22]	To overcome the issues of NavBelt	Ultrasonic waves	Servomotor and ultrasonic obstacle detection combined to guide the user in the identified direction	does not have an audio output and cannot detect obstacles nearby
K-sonar [23]	Obstacle recognition	Ultrasonic waves	Front obstacle detecting through echolocation	restricted functioning, large weight
SmartCane [24]	To identify obstacles higher above the knees	Ultrasonic waves	Ultrasonic sensors for knee above obstacle detection	No support for direction
iCane [25]	To identify obstacles & provide environmental information	Ultrasonic waves, radio frequency signals	RFID provides the user with surrounding information and stores navigation data in a database.	Static control might cause user navigation issues.

3.3 Conclusion of Survey

Table 1 summarizes the key features of the SmartCane system for visually impaired individuals. The system employs advanced object detection through integrated sensors, facilitating enhanced navigation in various environments. A real-time feedback system is incorporated to provide users with immediate information, utilizing deep learning algorithms for accurate object recognition. The algorithm employed in the Voice module ensures seamless interaction and communication, contributing to the overall effectiveness of the SmartCane in improving mobility and safety for visually impaired users.

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATIONS

A Software Requirement Specification (SRS) is a comprehensive document that outlines the functional and non-functional requirements of a software application. It serves as a formal agreement between stakeholders, such as clients, developers, and project managers, detailing what the software should do and how it should perform. The SRS includes descriptions of the system's purpose, features, user interactions, performance standards, design constraints, and validation criteria. By providing a clear and detailed blueprint, the SRS ensures that all parties have a mutual understanding of the project scope and objectives, facilitating smooth development and reducing the risk of misunderstandings or misaligned expectations.

4.1 Introduction

The Software Requirements Specification defines the functional and non-functional requirements for the SmartCane with object detection software, aiming to provide a safe and assistive navigation tool for visually impaired users. Serving as a communication tool between stakeholders, it delineates expected functionalities, performance criteria, and interfaces of the software to ensure alignment with user needs and project objectives. Subsequent sections will elaborate on specific requirements including object detection and identification, obstacle avoidance and path navigation, user interface for feedback and interaction, system performance criteria such as accuracy and battery life, as well as safety and security considerations.

4.2 Product overview

The SmartCane system software operates independently but can interact with external systems like hardware sensors for environment data and the underlying operating system for low-level functions. It also allows for potential expansion, enabling connectivity with GPS or smartphone apps for advanced features. Detailed interfaces and protocols will be outlined in the Software Requirements Specification (SRS) to ensure smooth data exchange and system operation.

The SmartCane system software assists visually impaired users by detecting and identifying objects in their path using sensor data. It provides real-time feedback through audio alerts and vibration patterns. Optional features may include GPS integration for location awareness and Bluetooth connectivity for customization. The SRS will specify input/output formats, algorithms, and user interaction methods.

The primary target users of the SmartCane software are individuals who are blind or visually impaired, ranging from those with complete blindness to low vision. Additionally, the software assumes users who are comfortable with using assistive technologies and have a basic understanding of how to navigate using such tools. These users rely on the software to provide them with essential navigation assistance, enabling them to navigate safely and independently in various outdoor environments during their daily activities.

The SmartCane system software is designed for use in various outdoor environments to assist users with navigation during their daily activities. This includes sidewalks, streets, parks, and other outdoor areas where visually impaired users might walk for exercise, commute, or run errands. The software is expected to function effectively under a range of outdoor conditions, including varying weather and lighting conditions. While the primary focus is on outdoor use, basic functionality indoors may also be considered depending on sensor capabilities and lighting conditions.

The Software Requirements Specification (SRS) assumes that the chosen SmartCane system hardware meets specific requirements to function effectively with the software. This includes reliable camera function under various operating conditions, sufficient processing power to run object detection algorithms efficiently, and appropriate communication interfaces to connect with sensors and external systems. Additionally, the SRS acknowledges any limitations of the system in specific environments, such as adverse weather conditions affecting sensor performance, and ensures that appropriate feedback is provided to users in such situations.

4.3 External Interface Requirements

4.3.1 User Interfaces

The primary User Interface will focus on non-visual feedback mechanisms such as audio alerts and descriptive soundscapes to convey information about the environment and navigation instructions.

4.3.2 Hardware Interfaces

The Raspberry Pi Camera module connects to the Raspberry Pi 4 via the CSI (Camera Serial Interface) port, utilizing a high-speed dedicated interface for camera communication. The camera can output raw image data (Bayer format) or compressed formats like JPEG. The SRS will specify the preferred format based on processing requirements and available storage space. The CSI interface utilizes a custom protocol for data transfer between the camera module and the Raspberry Pi. The software will leverage Raspberry Pi Camera libraries for communication and data retrieval.

4.3.3 Software interfaces

The software will utilize a computer vision library for object detection, capable of processing camera image data and identifying objects within the image frame. Detect predefined objects relevant to SmartCane users, providing information like object class, bounding box coordinates, and confidence score. A user interface library may be employed for managing non-visual feedback mechanisms, such as playing audio alerts with specified content and voice characteristics. Interaction with standard OS libraries for functionalities like file system access, networking, and low-level hardware interaction is anticipated.

4.3.4 Communication Interfaces

Potential future interfaces for connecting with external systems like GPS or smartphone apps may involve Bluetooth or Wi-Fi communication protocols. Protocols for communication with external systems, such as GPS location data or user settings from a smartphone app, will be specified. Security considerations for data transmission and storage will also be addressed in the SRS.

4.4 Functional Requirements

- Object Detection Module: The system should integrate computer vision cameras for object detection, capable of identifying and categorizing objects within a specified range.
- Speech Output Module: The system should include a speech synthesis module to convertdetected object information into clear and concise audible output.
- Panic Button Module: The panic button is a small push button integrated with the smart cane. It sends an alert message to the guardian and provides the current location of the visually impaired individual.

4.5 Non-Functional Requirements:

- Performance: The system should have a real-time response to detected objects, ensuringtimely feedback to the user.
- Reliability: The device should be reliable in different environmental conditions, including low-light and noisy environments.
- Security: The communication between the device components should be secure to prevent unauthorized access or tampering.
- Scalability: The system should allow for future upgrades or enhancements to accommodate new technologies or improvements.

Functional and non-functional requirements are most important and helps to know how theproject flow occurs. Functional requirements describe how the system works and non-functional requirements provide details how the flow must perform.

4.6 Hardware Requirements

- Raspberry Pi Model 4B
- Pi Camera – 5 Mega Pixel
- Rj45 Cabel for Data transfer
- Type C cabel for Power Supply
- SD Card -32 GB

4.7 Software Requirements

- Raspbian OS
- Python Version 3
- MobileNet V2 Architecture
- COCO Dataset
- Libraries – cv2, Pyttsx3

4.8 Use case description and Use case diagram

When the user activates the SmartCane, it initiates the object detection and speech output modules. The system detects obstacles and provides auditory feedback, describing the type of the detected object. This process enhances the user's awareness of their surroundings, ensuring safer and more informed navigation as shown in figure 4.8.

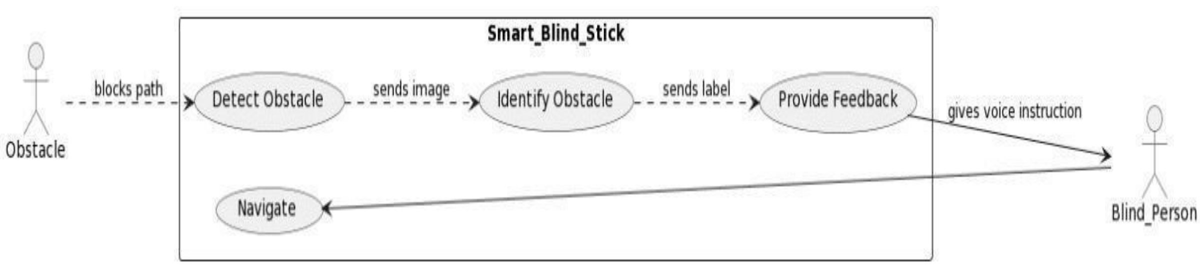


Figure 4.8: Use case diagram

CHAPTER 5

DESIGN

Software design encompasses the process of defining the architecture, components, interfaces, and data for a software system to meet specified requirements effectively and efficiently. It involves translating the requirements specified in the Software Requirement Specification (SRS) into a blueprint that guides the implementation phase. Software design focuses on structuring the system components and their interactions, ensuring modularity, reusability, maintainability, and scalability. Design decisions involve choosing appropriate algorithms, data structures, and patterns to achieve desired functionality and performance. Effective software design not only aims to fulfill functional requirements but also considers non-functional aspects such as usability, security, and reliability, aiming to create a robust and adaptable software solution.

5.1 Introduction

The intelligent cane emerges as a crucial aid for individuals facing visual impairments, offering a heightened level of convenience in navigating their surroundings. Going beyondthe conventional cane, it incorporates sophisticated functionalities, leveraging deep learning technology for object detection via a Raspberry Pi and camera setup. This AI- infused smart cane represents a noteworthy advancement in assistive technology, fosteringuser confidence and independence in daily activities. Key attributes include audio feedbackthrough a built-in speaker and a panic button that instantly communicates the user's location to designated guardians, emphasizing a commitment to safety. The meticulously crafted user-friendly design underscores its potential to significantly enhance the overall quality oflife for those with visual impairments.

5.2 High Level Design

The SmartCane system consists of three layers:

1. Hardware Layer : Raspberry Pi 4, sensors, user interface components, battery
2. Software Layer: Object Detection Module, User Interface Module
3. User Interaction Layer: Audio alerts.

Data Flow

- The Raspberry Pi continuously captures data from the camera.
- The object detection module processes the sensor data using computer vision libraries to identify objects within the captured image.
- The path planning module receives information about the detected objects and user location (potentially from GPS) to generate safe navigation paths.
- The user interface module generates audio alerts and vibration patterns based on the object detection.
- The device control module transmits audio alerts to the speakers.
- The user receives non-visual feedback through audio alerts.

Design Considerations

- Real-time processing with minimal latency.
- Power efficiency for extended battery life.
- Modularity for easy development, testing, and future modifications.
- Scalability to accommodate potential future functionalities or external system integration.

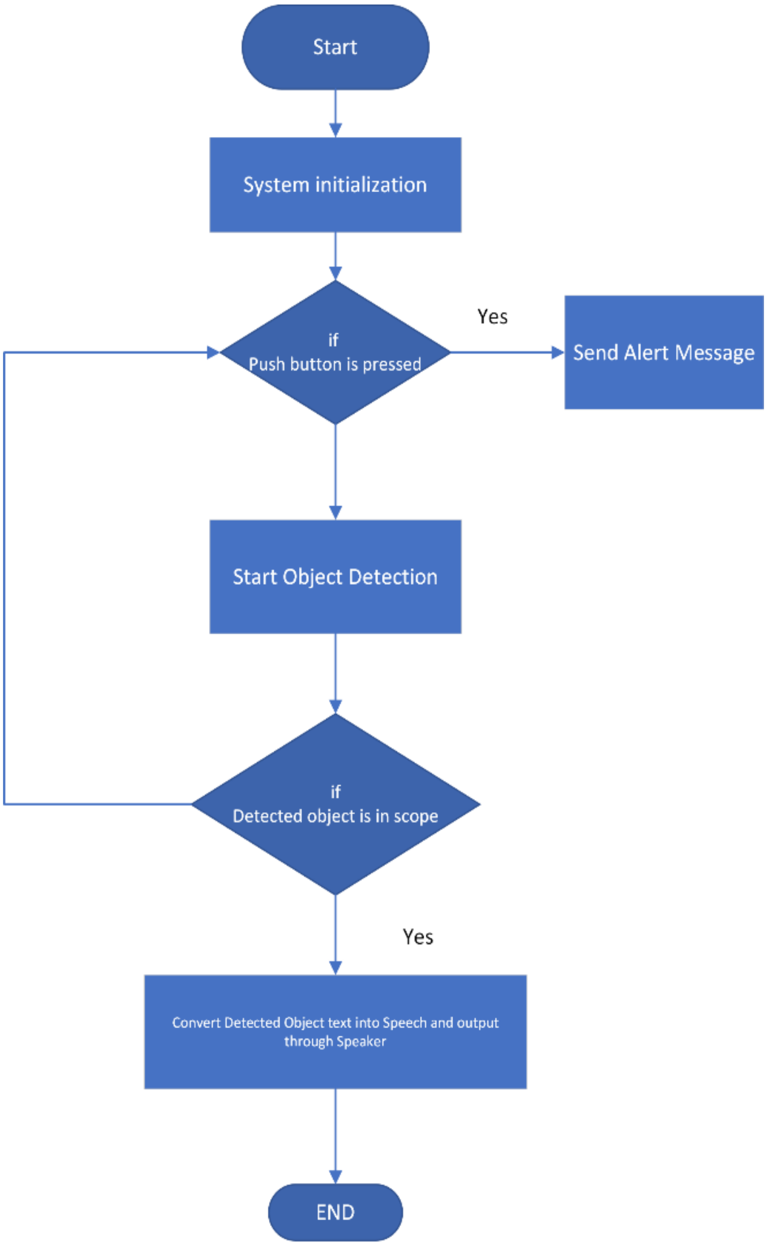


Figure 5.1: Object Detection Process

The object detection approach, which begins with system initialization, is shown in the flowchart. The system detects whether a push button has been pressed during initialization. It starts object detection if it is not pressed, and delivers an alarm message otherwise. The discovered object's scope is then verified by the system. The process ends with the detected object text being translated into speech and output through a speaker if the object is in scope as shown in figure 5.1.

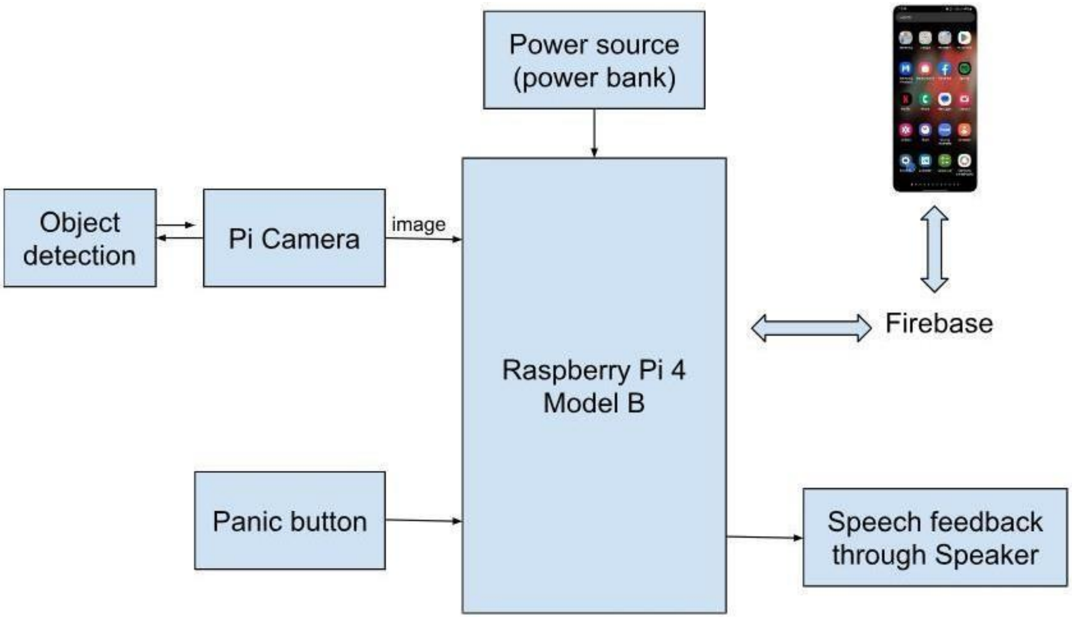


Figure 5.2: System Architecture for the Proposed System

Figure 5.2 shows the proposed SmartCane's system architecture. Panic button, Raspberry Pi, object detection camera, and user interface are all integrated into the hardware prototype. With Firebase, the SmartCane App offers an adaptable user interface, real-time object detection, and speech output via a speaker.

5.3 Low Level Design

The low-level design details the functionalities and components of each module, comprising the Object Detection Module, User Interface Module, Device Control Module, and optional Graphical User Interface. It includes components such as the Image Acquisition Library and Computer Vision Library for object detection, an Audio Alert Library for real-time feedback, and communication interfaces for efficient information exchange. Furthermore, it highlights considerations like error handling mechanisms, security measures, and the selection of libraries and algorithms based on system constraints and requirements.

Sequence Diagram

A sequence diagram visually represents the interaction between different components of a system in a time-ordered sequence. It shows how objects interact with each other and in what order these interactions occur. For instance, in a visually impaired assistance system, the sequence diagram would illustrate the process from the system start-up to the activation of the camera as the person starts walking, the detection of objects, and the subsequent capturing and processing of images for text using OCR. If text is detected, the diagram would show the conversion of this text to audio using Python Text-to-Speech (Pyttsx3) and the output of the audio. It would also depict the handling of a panic button press, where the Raspberry Pi sends a signal to Firebase, which then updates the user interface to alert relevant parties. This detailed visualization helps in understanding the flow of events and interactions within the system as shown in the figure 5.3.

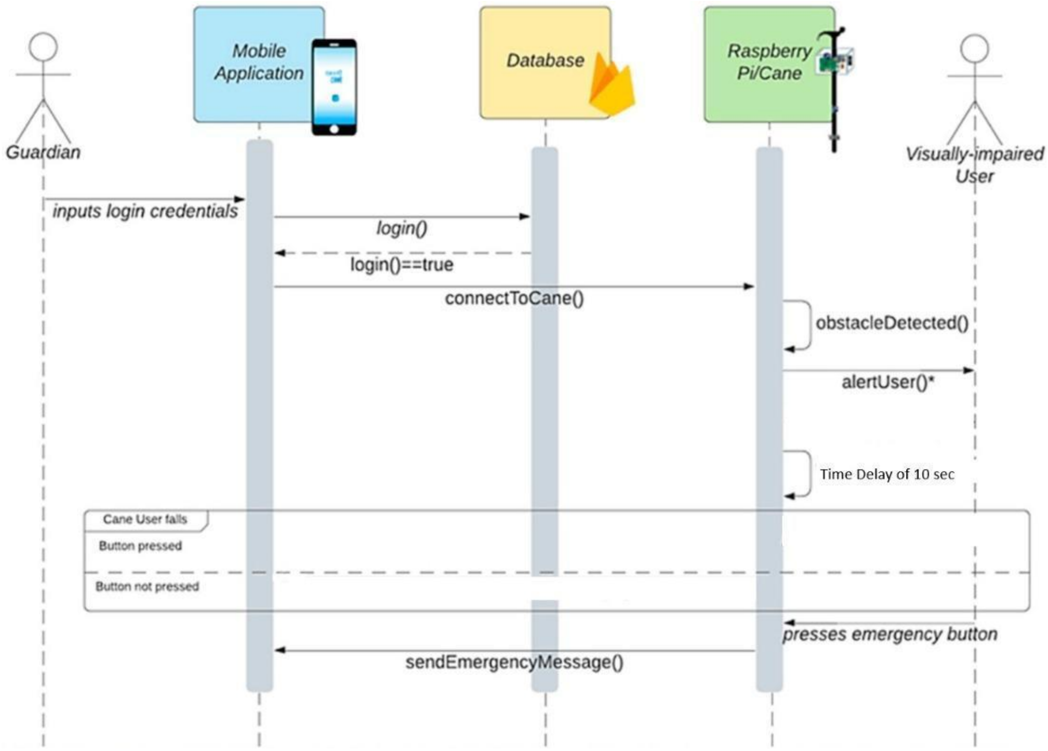


Figure 5.3: Sequence Diagram of SmartCane

Activity Diagram

An activity diagram visually represents the flow of actions within a system, including decision points and sequential phases. It uses nodes and arrows to show activities and their control flow, aiding in understanding both conditional and parallel operations. This diagram acts as a roadmap, illustrating how system components interact step-by-step, from initialization to camera activation upon user movement, object identification, and image capture. It further depicts text extraction using OCR technology, text-to-speech conversion with Pyttsx3, and handling of a panic button push by signaling Firebase and updating the UI. This representation clarifies the system's interactions and event sequence as shown in the figure 5.4.

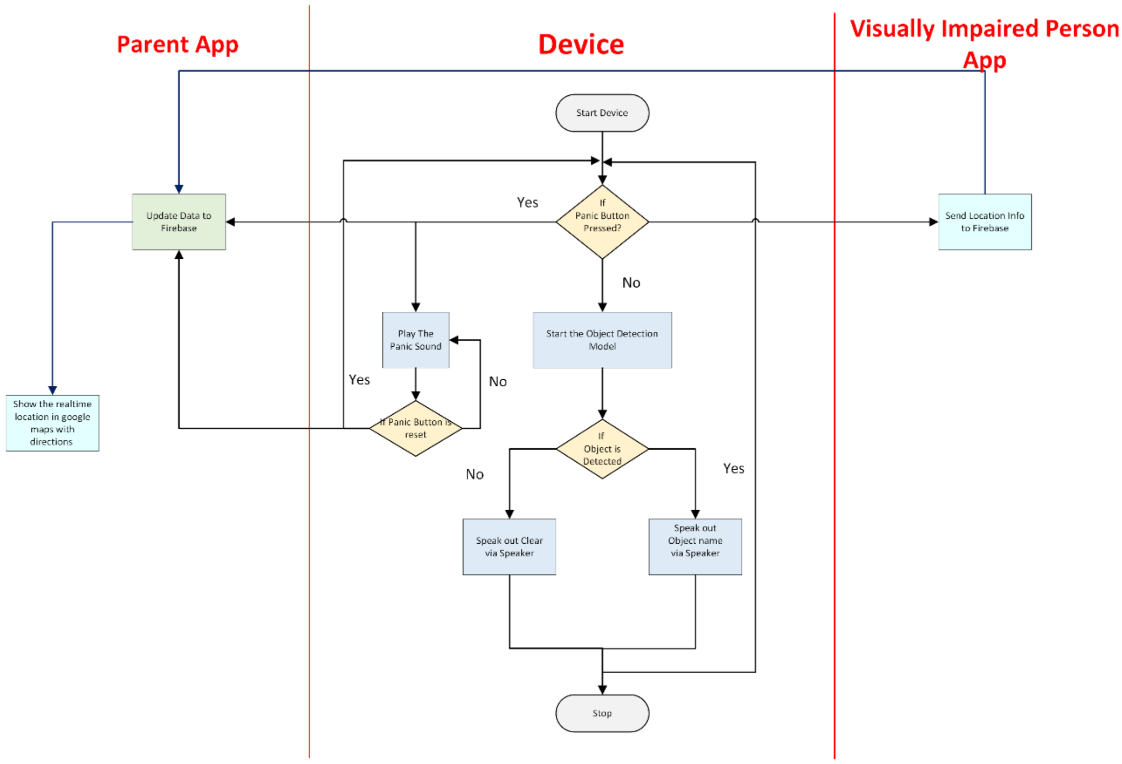


Figure 5.4: Activity Diagram of SmartCane

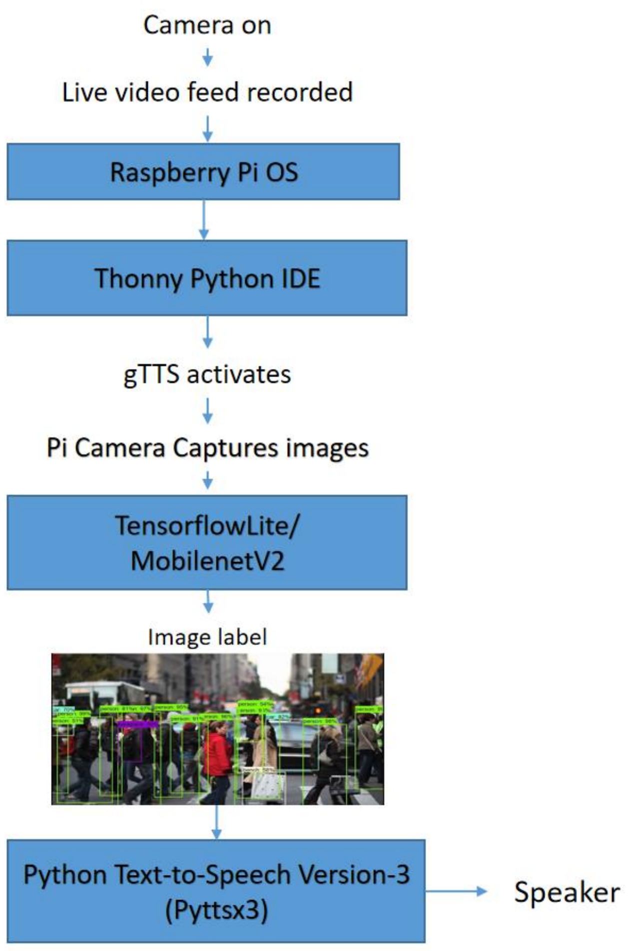


Figure 5.5: Graphical User Interface for Object Detection

The setup for object detection with a Raspberry Pi is shown in the diagram. The camera is turned on to start the process and record a live video feed. gTTS (Google Text-to-Speech) is enabled by the Raspberry Pi OS through the Thonny Python IDE, after which the Pi Camera is used to take pictures. TensorFlow Lite/MobileNetV2 is used to process these photos and produce image labels. These labels are then translated into voice using Python Text-to-voice (pyttsx3) and output through a speaker, as illustrated in figure 5.5.

5.4 Conclusion

The application's low-level features and high-level design support object detection, making development, maintenance, and future updates easier. The entire user experience is improved with a user-friendly interface and effective data handling.

CHAPTER 6

IMPLEMENTATION DETAILS

Implementation details involve the practical steps and technical procedures needed to execute a project or system after planning and design. This phase includes coding, configuration, installation, testing, and deployment. It transforms theoretical designs into functional reality, ensuring all components work seamlessly to achieve project goals.

6.1 Tools Introduction

The RealVNC application serves as a versatile solution for remote access and control, particularly when deployed in conjunction with Raspberry Pi. This application enables users to connect to their Raspberry Pi devices from virtually anywhere, facilitating seamless management and monitoring.

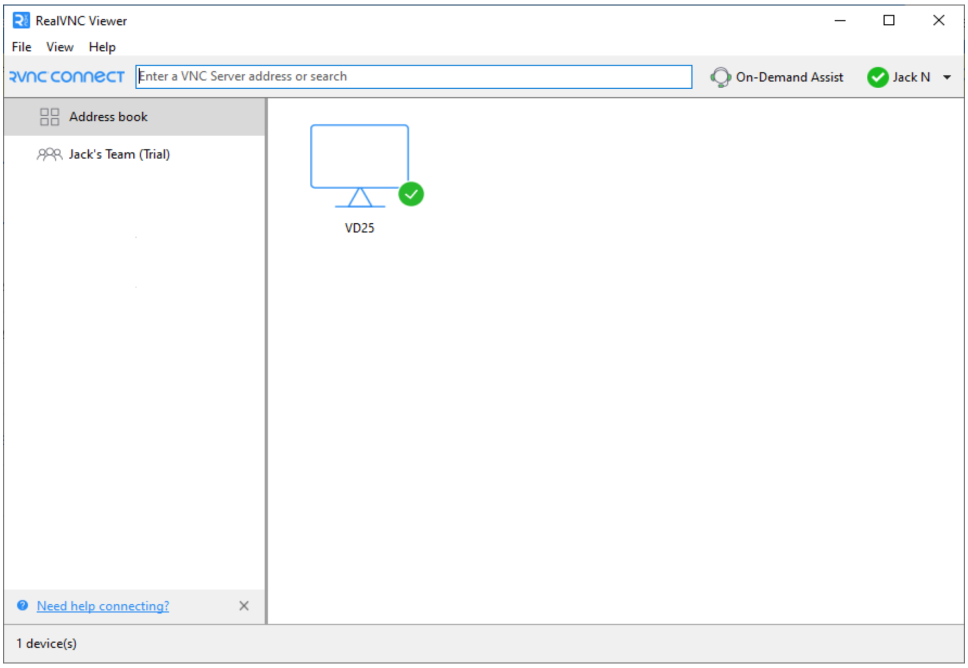


Figure 6.1: RealVNC Server

RealVNC utilizes a client-server model, where the RealVNC server runs on the Raspberry Pi, allowing it to share its desktop environment as shown in the figure 6.1. Users can then access and control their Raspberry Pi remotely through the RealVNC viewer on another device, such as a computer, tablet, or smartphone. The implementation of RealVNC on Raspberry Pi simplifies the process of administering and interacting with the device, providing a user-friendly interface that fosters efficient control and monitoring for various applications and projects as shown in figure 6.2.

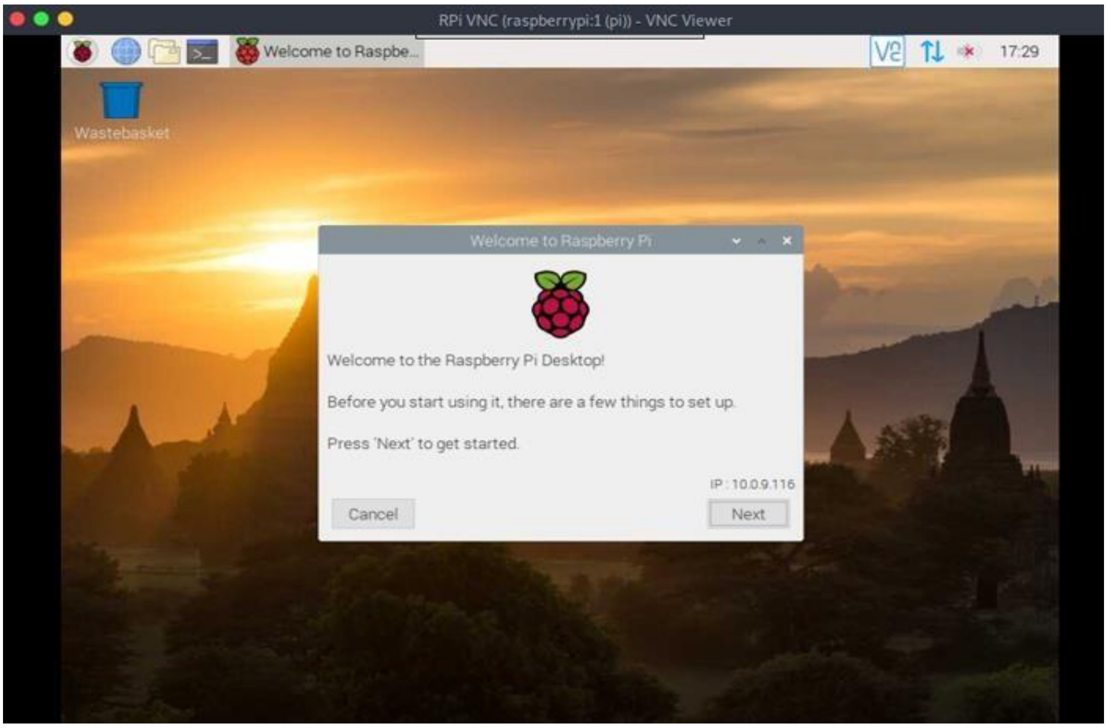


Figure 6.2: RealVNC Application – Platform to run Raspberry Pi

Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python's extensive standard library and ecosystem of third-party packages make it suitable for a wide range of applications, from web development to scientific computing and artificial intelligence.

With a number of improvements targeted at improving code readability, simplicity, and performance, Python 3 expands upon its predecessor. Along with additional syntactic components like type annotations, async/await for asynchronous programming, and enhanced exception handling techniques, some of the key improvements are a more uniform approach to string handling and Unicode support. Combining these advances with optimizations and standard library improvements, Python continues to be a popular and flexible programming language in a wide range of fields, including scientific computing, web development, and artificial intelligence.

OpenCV is an open-source computer vision and machine learning software library. It provides a wide range of functions for tasks such as image and video processing, feature detection, object recognition, and machine learning. OpenCV is written in C++ and has bindings for various programming languages, including Python. It is widely used in fields such as robotics, augmented reality, medical imaging, and surveillance systems.

TensorFlow Lite is a lightweight version of Google's TensorFlow framework designed for mobile and embedded devices. It enables efficient deployment of machine learning models on resource-constrained platforms like smartphones, IoT devices, and microcontrollers. TensorFlow Lite supports various model formats and optimizations for speed and memory footprint, making it suitable for real-time inference tasks such as image classification, object detection, and natural language processing.

Thonny is a beginner-friendly integrated development environment (IDE) for Python programming. It provides features such as syntax highlighting, code completion, and simple debugging tools, making it easy for novice programmers to write and debug Python code. Thonny's user-friendly interface and built-in Python interpreter simplify the learning process and make it a popular choice for educators and learners alike.

Python Text-to-Speech version 3, or Pyttsx3, is a flexible toolkit that lets programmers easily incorporate text-to-speech features into their apps. Developers may provide accessibility and improve user experience by transforming written content into natural-sounding voice in real-time using pyttsx3. From interactive speech interfaces to assistive technology, its rich customization possibilities and simple API make it suited for a wide range of applications. With its offline features and support for many speech synthesis engines, Pyttsx3 is a dependable option for applications where internet connectivity may be restricted or nonexistent. All in all, pyttsx3 gives programmers the tools they need to design captivating and user-friendly apps that successfully converse with users orally.

Firebase Database is a cloud-hosted NoSQL database provided by Google as part of the Firebase platform. It offers real-time data synchronization and offline support, allowing developers to build responsive and collaborative applications across various platforms. With Firebase Database, developers can store and sync data in real-time between users and devices, eliminating the need for manual data fetching and updating. It supports structured data in JSON format, making it flexible and easy to integrate into both web and mobile applications as shown in the figure 6.3.

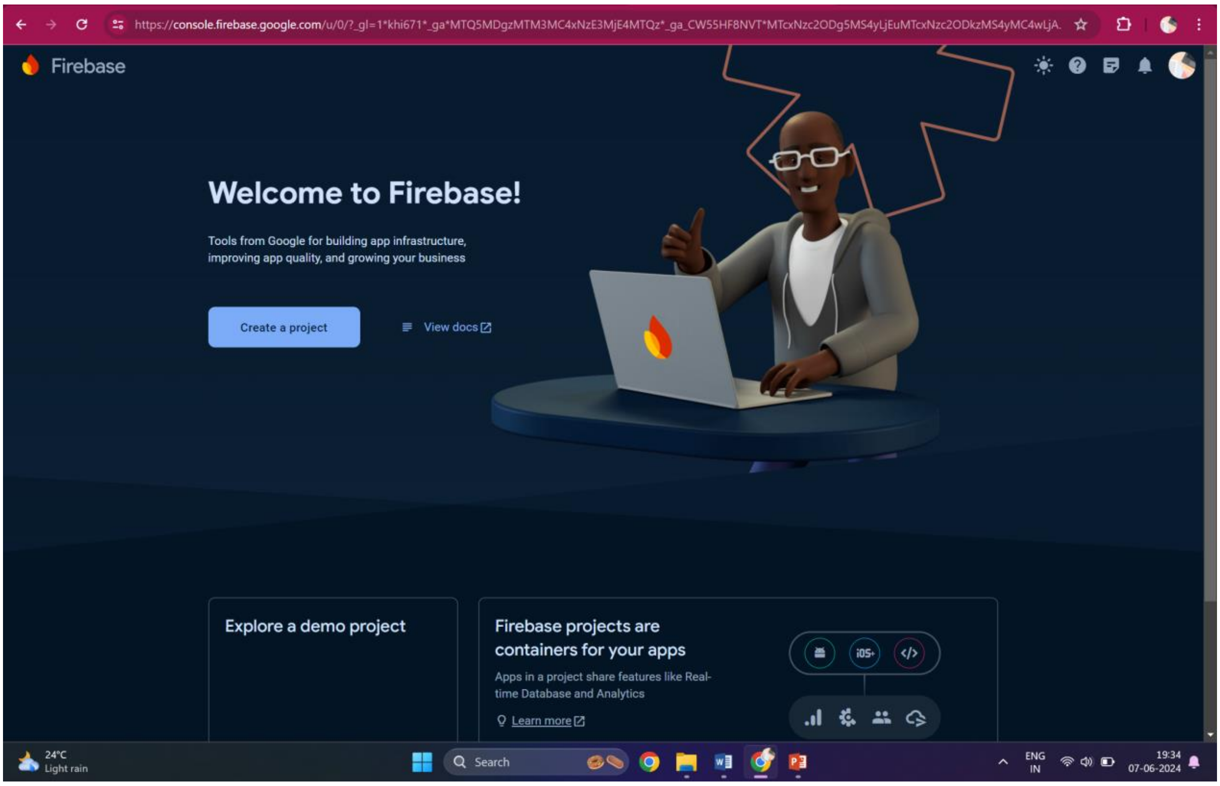


Fig 6.3: Firebase Realtime Database Console

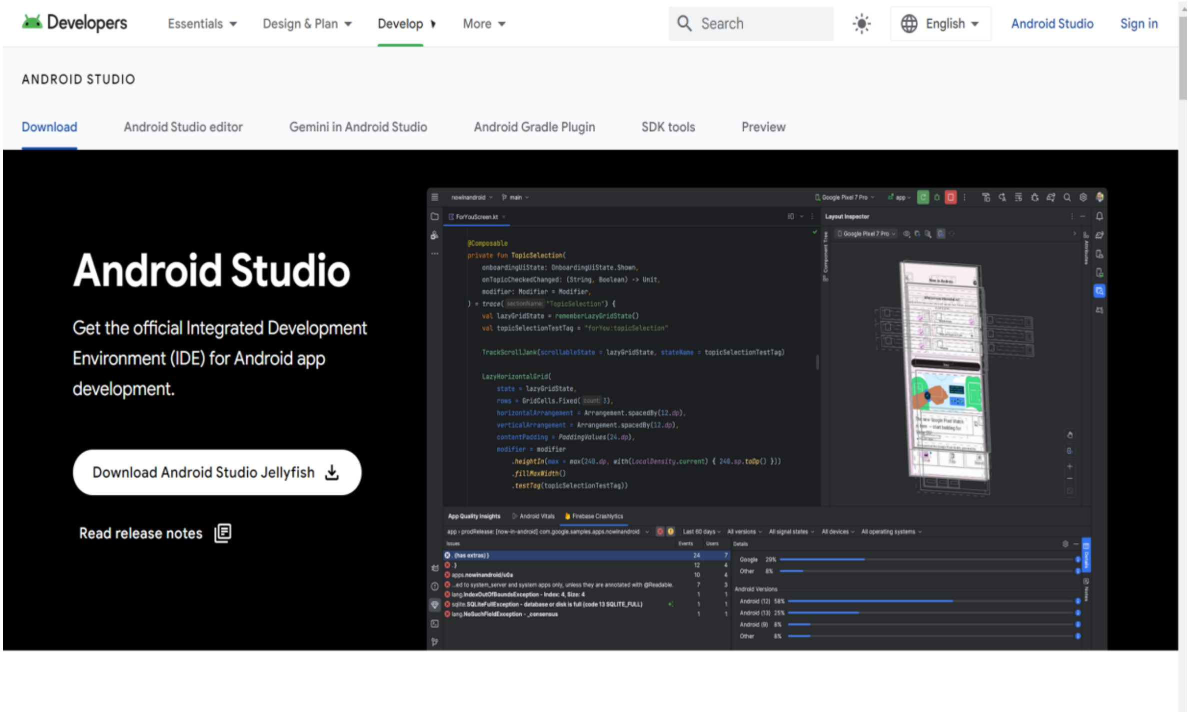
Android Studio is the official IDE (Integrated Development Environment) for Android app development and it is based on JetBrains' IntelliJ IDEA software. Android Studio provides many excellent features that enhance productivity when building Android apps, such as

- A blended environment where one can develop for all Android devices.
- Apply Changes to push code and resource changes to the running app without restarting the app.
- A flexible Gradle-based build system
- A fast and feature-rich emulator
- Extensive testing tools and frameworks
- Easy integration with real-time database ‘firebase‘.

Steps for Steps to Install Android Studio on Windows

Step 1: To download the Android Studio, visit the official Android Studio website in your web browser.

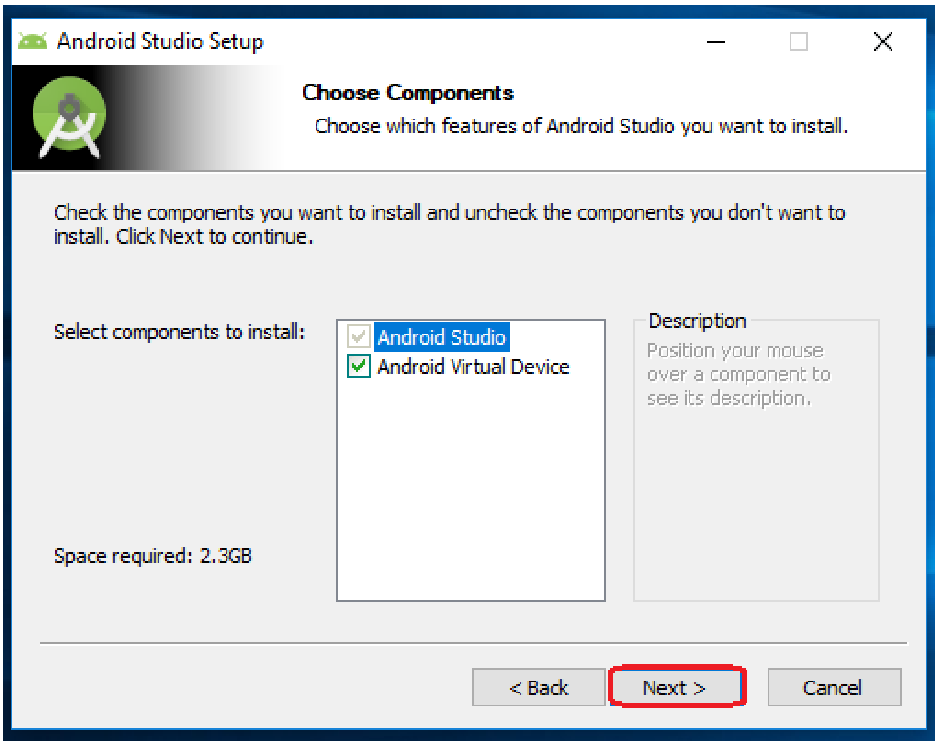
Step 2: Click on the "Download Android Studio" option.



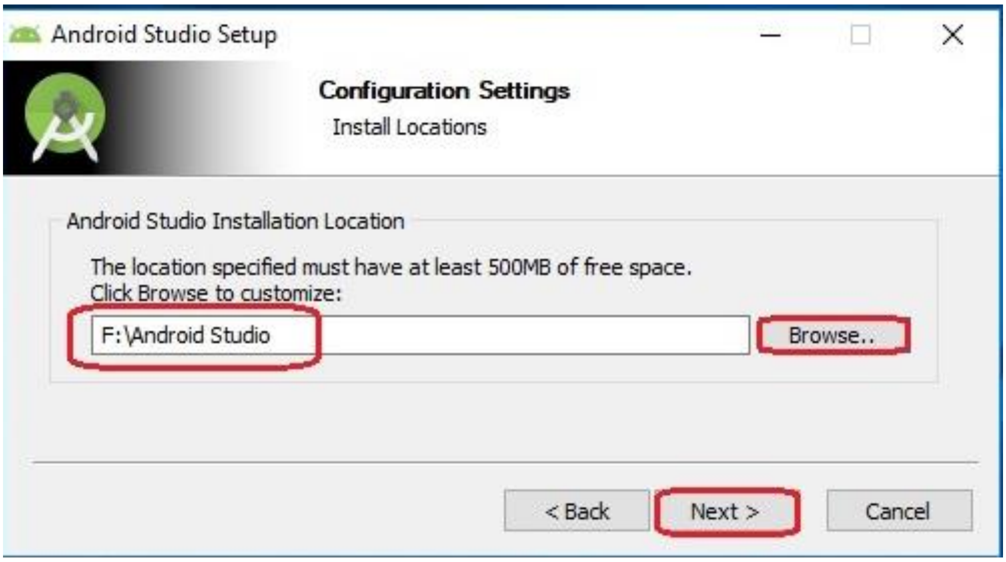
Step 3: Double-click on the downloaded "Android Studio-ide.exe" file.

Step 4."Android Studio Setup" will appear on the screen and click "Next" to proceed.

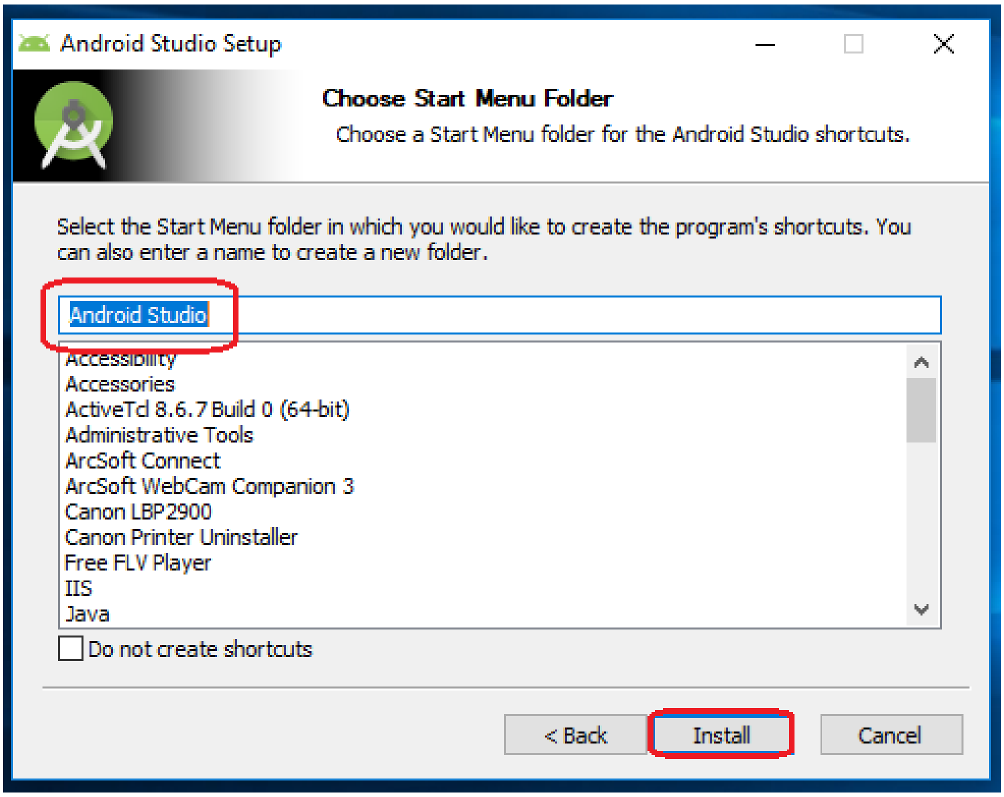
Step 5: Select the components that you want to install and click on the "Next" button.



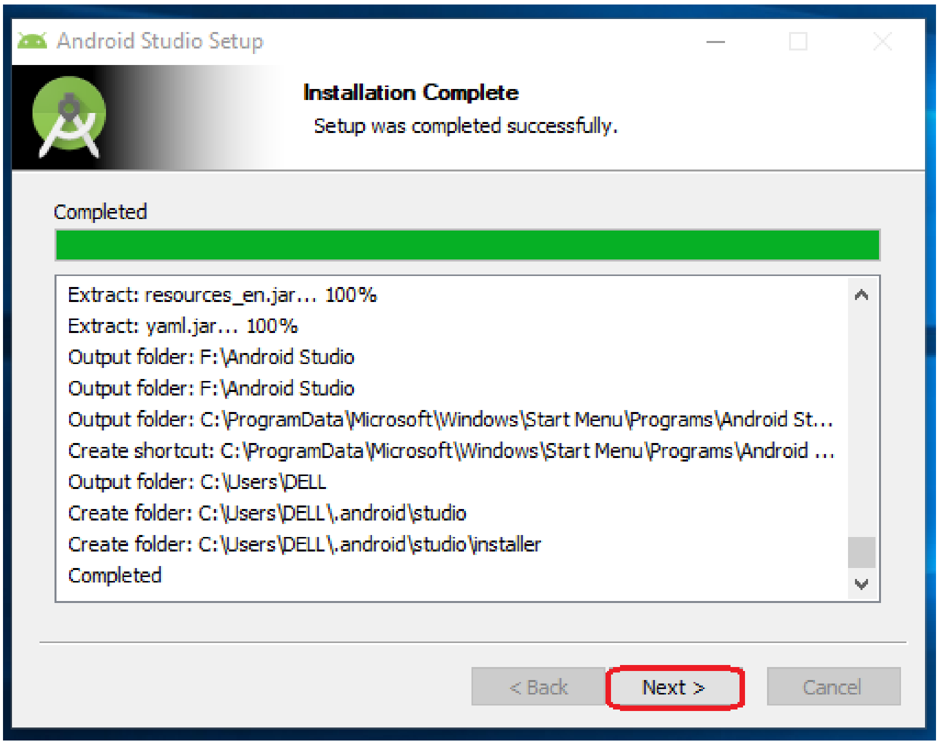
Step 6: Now, browse the location where you want to install the Android Studio and click "Next" to proceed.



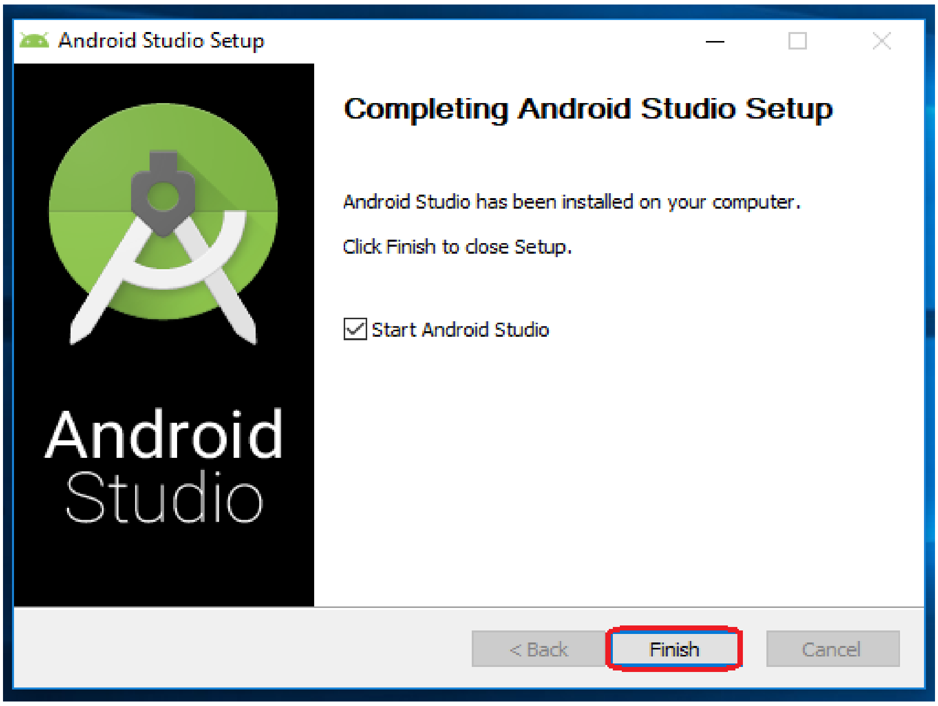
Step 7: Choose a start menu folder for the "Android Studio" shortcut and click the "Install" button to proceed.



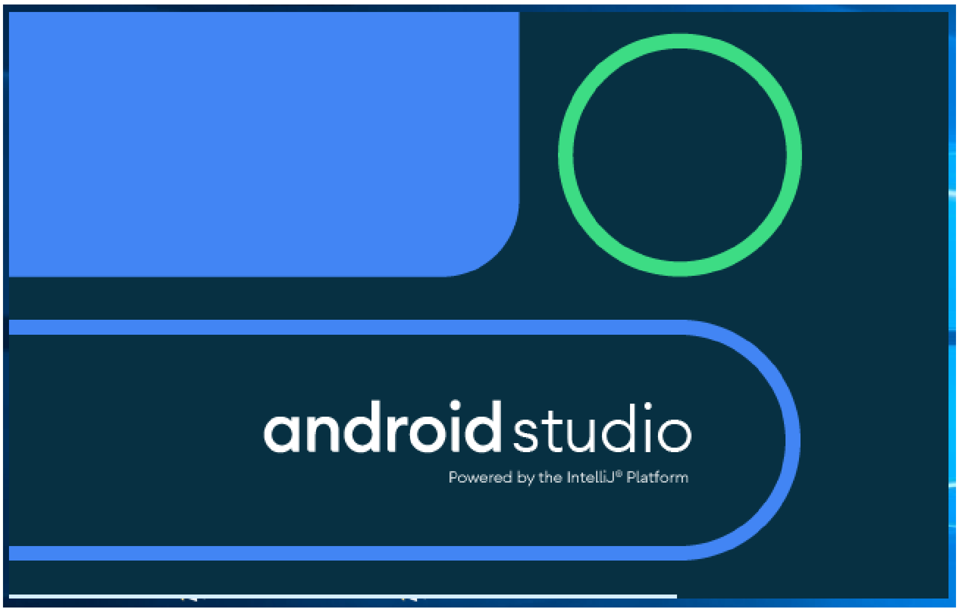
Step 8: After the successful completion of the installation, click on the "Next" button.



Step 9: Click on the "Finish" button to proceed.

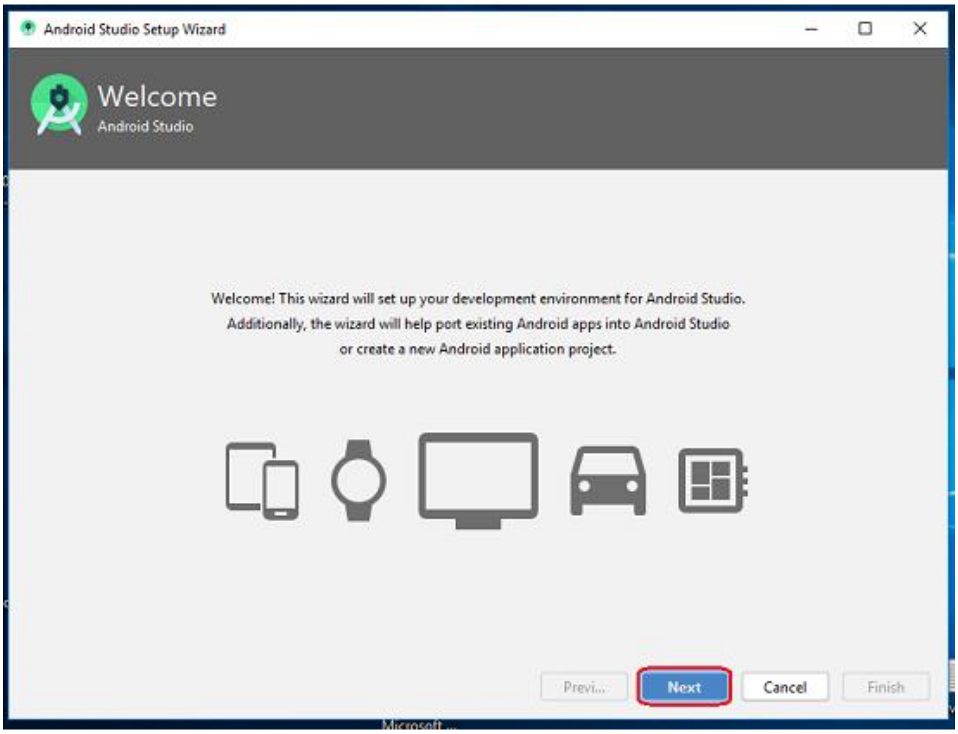


Now, your Android studio welcome screen will appear on the screen.

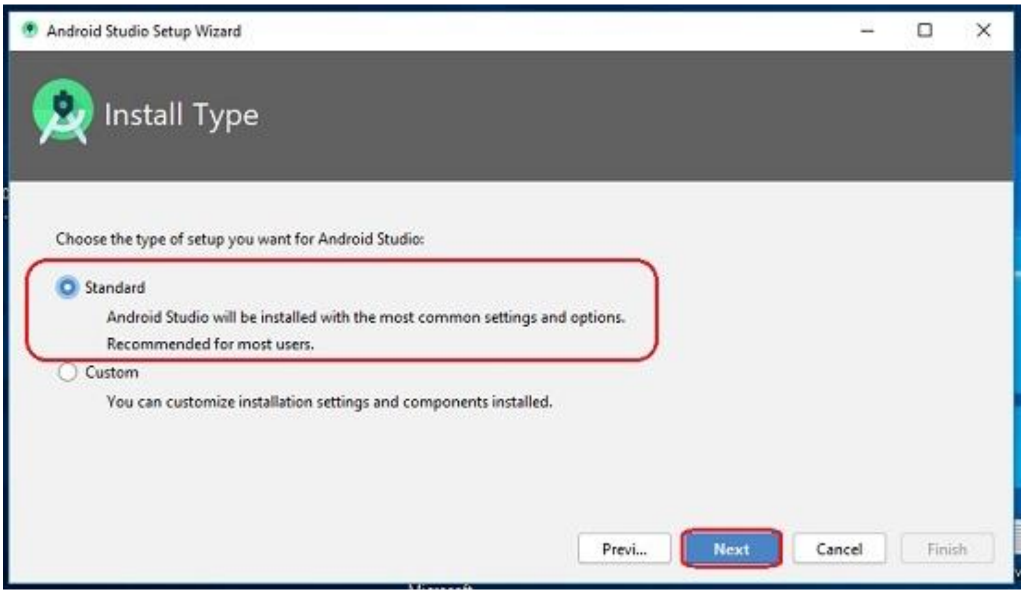


Android Studio Setup Configuration

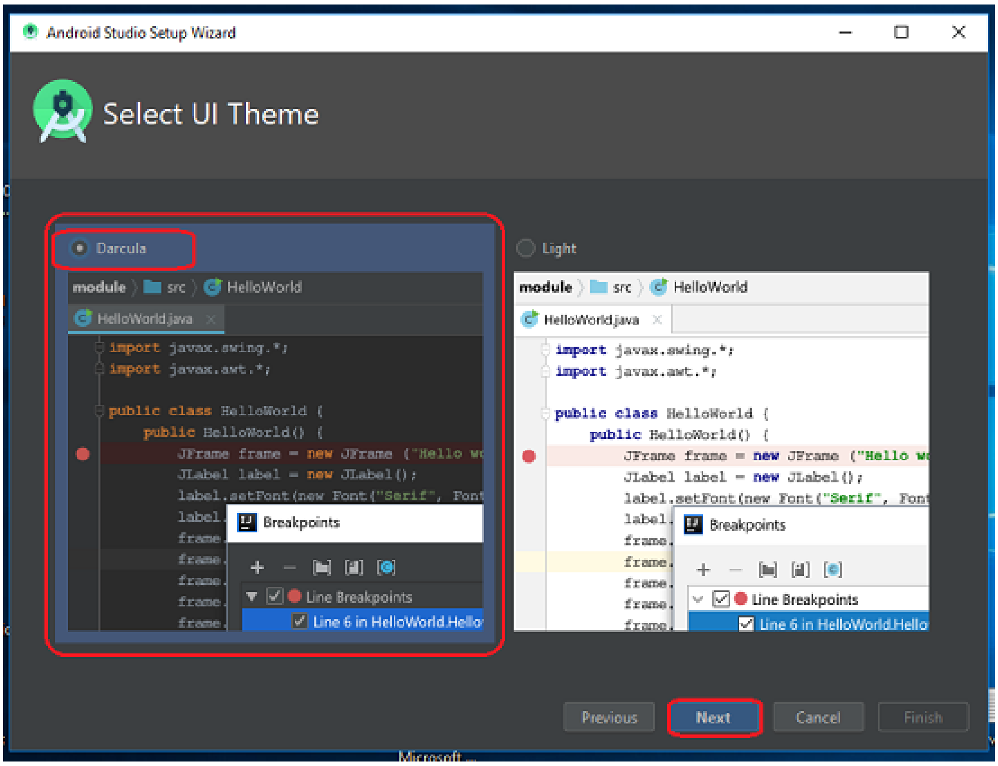
Step 10: "Android Studio Setup Wizard" will appear on the screen with the welcome wizard. Click on the "Next" button.



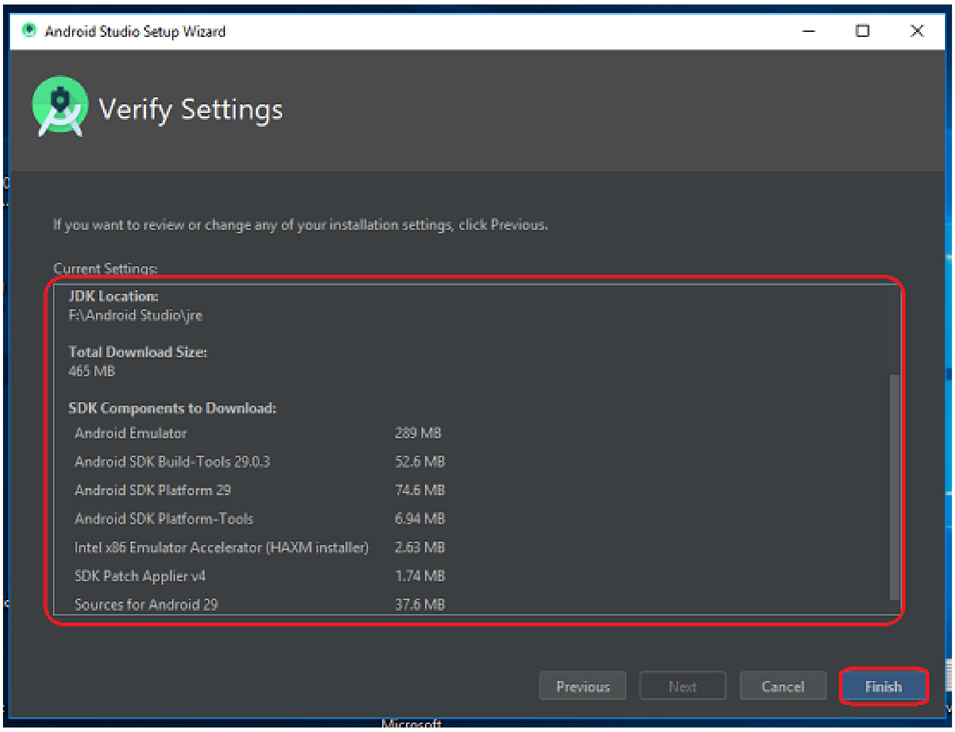
Step 11: Select (check) the "Standard" option if you are a beginner and do not have any idea about Android Studio. It will install the most common settings and options for you. Click "Next" to proceed.



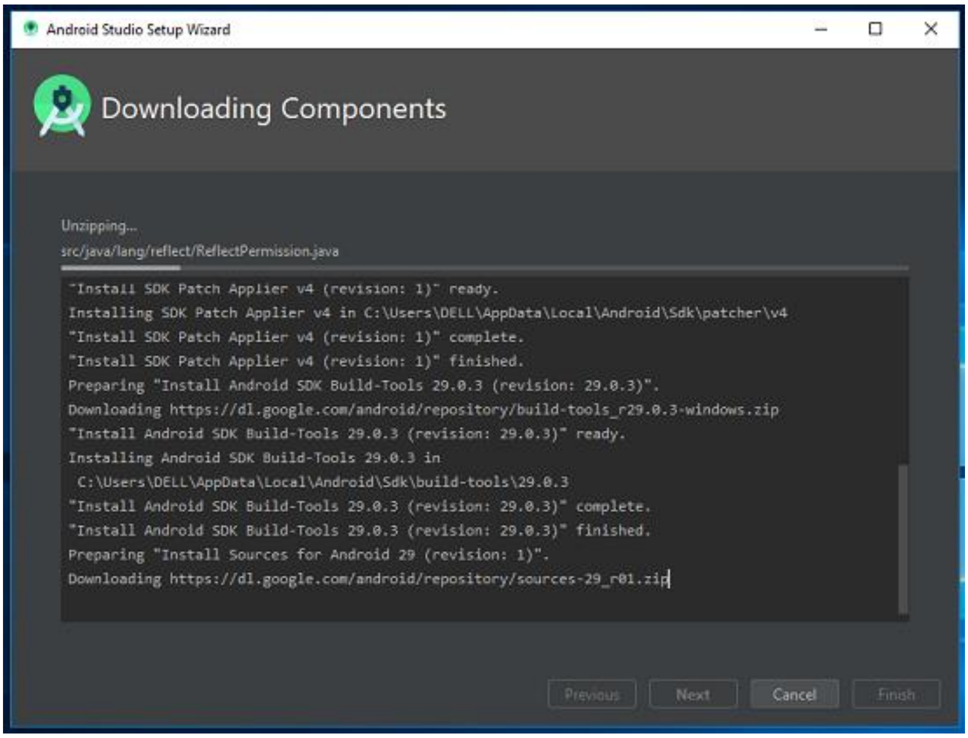
Step 12: Now, select the user interface theme as you want. Then, click on the "Next" button.



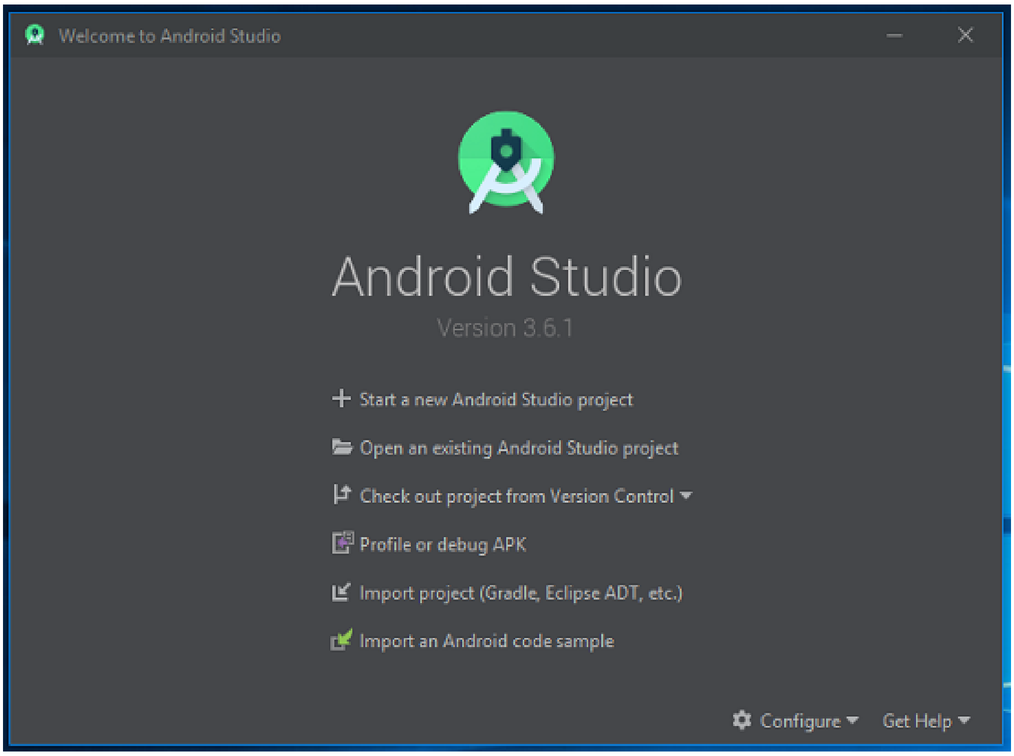
Step 13. Now, click on the "Finish" button to download all the SDK components.



And, the downloading and installation process of components gets started.



Step 14. After downloading all the necessary components, click on the "Finish" button.



6.2 Technology Introduction

Convolutional Neural Networks (CNNs) represent a fundamental breakthrough in the field of deep learning and computer vision. Inspired by the organization of the visual cortex in animals, CNNs are designed to automatically and adaptively learn spatial hierarchies of features from raw input data. At their core, CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

The convolutional layers serve as feature extractors, detecting patterns such as edges, textures, and shapes within the input data by applying filters across the input image. These filters are learned during the training process, enabling the network to identify relevant features for the given task. The pooling layers then down sample the feature maps generated by the convolutional layers, reducing the spatial dimensions of the data while preserving important features.

As the network progresses through its layers, it learns increasingly complex and abstract representations of the input data. Finally, the fully connected layers combine these representations to make predictions or classifications. Through a process called back propagation, CNNs adjust the weights of their neurons during training to minimize the difference between predicted and actual outputs, thereby optimizing their performance.

CNNs have revolutionized various computer vision tasks, including image classification, object detection, image segmentation, and more. Their ability to automatically learn hierarchical representations from raw data has led to significant advancements in fields such as autonomous driving, medical imaging, and facial recognition. With ongoing research and advancements, CNNs continue to push the boundaries of what is possible in computer vision and artificial intelligence.

Convolutional neural networks like MobileNetV2 are engineered for effective deep learning on embedded and mobile platforms. It aims to achieve state-of-the-art performance with much lower computing cost. It is an extension of Google's original MobileNet architecture. In order to maximize model efficiency and retain high accuracy in tasks like object detection, semantic segmentation, and picture classification, MobileNetV2 makes use of depthwise separable convolutions, inverted residuals, and linear bottlenecks. Its architecture is made up of several building blocks that allow it to balance efficiency and accuracy, such as linear bottlenecks and inverted residual blocks.

In object recognition, MobileNetV2 works by processing input images through a series of convolutional layers to extract features at different spatial resolutions. These features are then passed through multiple layers of depthwise separable convolutions and linear bottlenecks, which reduce the computational cost while preserving valuable information. The network learns to recognize objects by analyzing these features and their spatial relationships. During training, MobileNetV2 adjusts its parameters using techniques like backpropagation and stochastic gradient descent to minimize prediction errors. Once trained, the model can efficiently classify objects in new images by comparing their features to those learned during training. MobileNetV2's efficient architecture makes it well-suited for real-time object recognition tasks on resource-constrained devices like smartphones and embedded systems.

Key Technologies Involved in SmartCane Software

Hardware Platform

The Raspberry Pi 4 single-board computer serves as the backbone of the SmartCane software, owing to its compact form factor, cost-effectiveness, and formidable processing capabilities. It provides the necessary horsepower to handle essential tasks such as object detection utilizing libraries like OpenCV or TensorFlow Lite, along with sensor data management. With its versatility and ample resources, the Raspberry Pi 4 ensures smooth and efficient operation of the SmartCane system software, empowering users with enhanced navigation assistance.

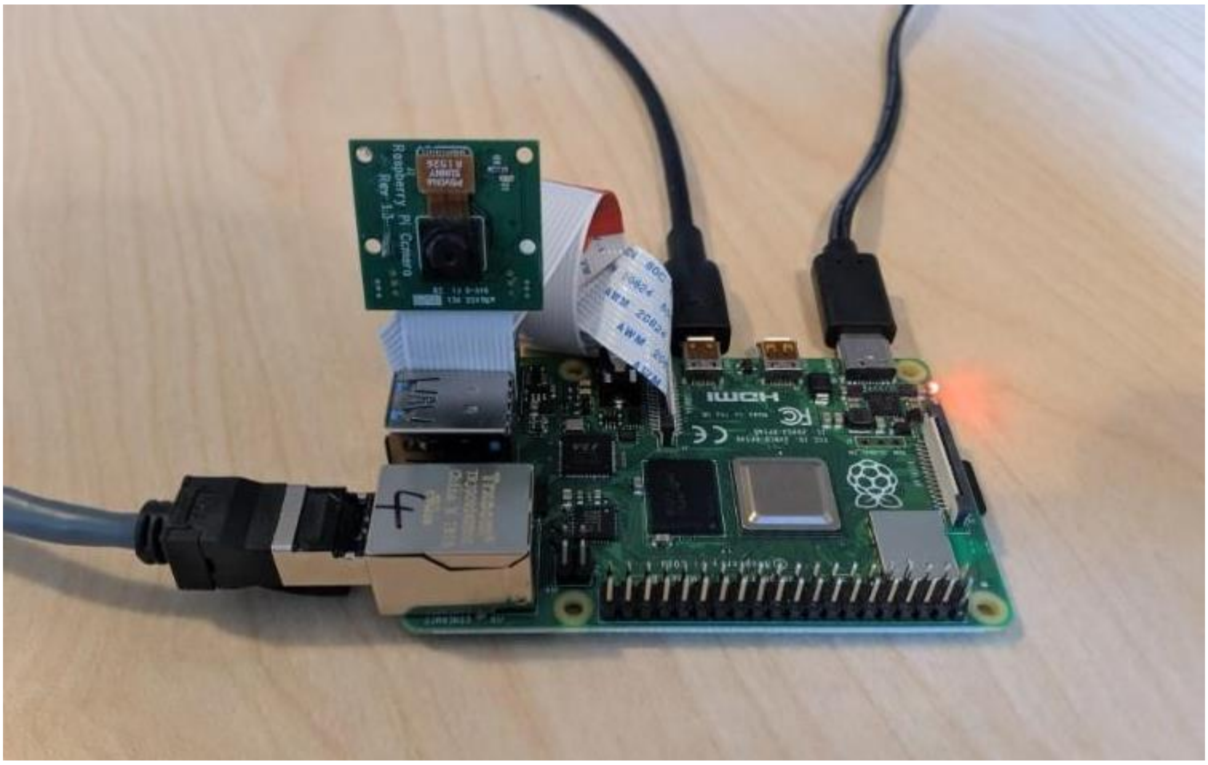


Figure 6.4: Raspberry Pi 4 Model B Mounted with Pi Camera 5MP

Sensors

Raspberry Pi Camera: This cost-effective and widely adopted camera module integrates seamlessly with the Raspberry Pi 4 as shown in Figure 6.4. It serves as the primary sensor for capturing images crucial for object detection tasks leveraging computer vision libraries.

Software Technologies

Real-time Operating System (RTOS) (Optional): In scenarios demanding advanced functionalities and stringent timing requirements, the integration of a real-time OS becomes a consideration. It ensures consistent and predictable response times, particularly for critical tasks such as object detection.

Linux-based OS (Default): For standard operations where real-time constraints are less critical, a Linux-based operating system like Raspbian suffices, providing a stable foundation for the Raspberry Pi.

Computer Vision Libraries

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. Originally developed by Intel, it was later supported by Willow Garage and Itseez. The library includes over 2,500 optimized algorithms, which can be used for various applications, such as facial recognition, object identification, image processing, and more. OpenCV supports multiple programming languages, including C++, Python, Java, and MATLAB, making it highly accessible for developers and researchers. It is widely used in academic and commercial projects due to its comprehensive functionality and ease of integration with other libraries and frameworks.

TensorFlow is an open-source machine learning framework developed by the Google Brain team. It provides a comprehensive ecosystem for building, training, and deploying machine learning models across various platforms, including desktops, mobile devices, and servers. TensorFlow supports deep learning and neural network research, making it a popular choice for developing applications in image and speech recognition, natural language processing, and reinforcement learning. With its flexible architecture and extensive community support, TensorFlow enables developers to experiment with and implement complex machine learning models efficiently. Additionally, TensorFlow offers tools like TensorBoard for visualization and TensorFlow Lite for mobile and embedded device deployment.

Other Libraries

Audio Output Libraries: Facilitating the control of speaker playback for delivering crucial audio alerts, these libraries play a vital role in enhancing user safety and awareness.

6.3 Overall view of the project in terms of implementation

The foundation of the object detection system lies in the utilization of cutting-edge technologies, primarily relying on computer vision algorithms and hardware integration, to improve user safety and navigation assistance. The implementation process involves careful planning, covering various stages such as data acquisition, model training, and real-time inference. At its core, the Raspberry Pi 4 single-board computer plays a pivotal role, overseeing the entire operation due to its compact form factor and robust computational capabilities. By incorporating libraries such as OpenCV and TensorFlow Lite, the software can effectively analyze camera images in real-time, detecting potential obstacles and hazards along the user's path. This implementation demands not only sophisticated algorithms for precise object detection but also seamless coordination with sensors like the Raspberry Pi Camera. The success of the project depends on meticulous attention to detail, optimal resource management, and a design philosophy centered around the needs of visually impaired individuals, ensuring the creation of a dependable and user-friendly navigation aid.

6.4 Information about the implementation of Modules

Pseudo code for working of SmartCane

1. System Start
2. Person starts walking
3. Camera is active
    a) If any objects are found within 10 meters:
        i) Capture the image
        ii) Process the image for text using Optical Character Recognition (OCR)
4. Speech Output
    a) If text is detected in the image:
        i) Convert the text to audio using Python Text-to-Speech (Pyttsx3)
        ii) Output the audio

5. Handling Panic Button

a) User triggers panic button

b) Raspberry Pi detects the button press

c) Send signal to Firebase

i) Transmit a signal to Firebase indicating the panic button activation

6. Firebase triggers user interface update

a) Upon receiving the signal, Firebase triggers an update in the user interface to alert relevant parties.

Pseudocode for MobileNetV2-SSD

Input: Image

Output: Audio Feedback of detected object

Start:

- Import Libraries: MobileNetV2, SSD, OpenCV, Numpy, pyttsx3.
- Initialize: MobileNetV2-SSD model and TTS engine.
- Preprocess Image: Resize to 300x300, normalize.
- Detect Objects: Use the model to predict detections from preprocessed image.

```
Function detect_objects(image):    preprocessed_image = preprocess_image(image)
       detections = model.predict(preprocessed_image)
 return detections
```

- Process Detections: Filter detections by confidence, extract label and bounding box.

```
Function process_detections(detections, threshold=0.5):
      results = []
      For each detection in detections:
        confidence = detection['confidence']
        If confidence > threshold:
            label = detection['label']
bbox = detection['bbox']             results.append((label, confidence, bbox))
    return results
```

- Provide Audio Feedback: Generate and speak messages for each detection.

End

The Object Detection Module utilizes Python libraries such as OpenCV or TensorFlow Lite to process images and identify objects within the camera's field of view. By integrating a pre-trained object detection model like MobileNet, trained on a dataset containing relevant objects, the module ensures accurate detection. It extracts essential data about detected objects, including class labels (e.g., pedestrian), bounding box coordinates, and confidence scores, providing valuable information for navigation.

The User Interface Module focuses on providing intuitive feedback to the user through audio alerts. It incorporates functions for generating informative voice messages about detected obstacles and path guidance. This involves utilizing text-to-speech functionality or pre-recorded audio files, delivering clear and concise instructions to the user, enhancing their situational awareness and safety during navigation.

The Device Control Module is responsible for managing the hardware components and system functionality of the SmartCane. It interacts with the camera using appropriate libraries or drivers to capture sensor data effectively. Additionally, it controls the speakers for audio alert playback, ensuring timely and audible notifications to the user. Moreover, the module manages the vibration motor for generating vibration patterns, providing haptic feedback to the user when necessary. To optimize the device's battery life, it implements power management techniques by intelligently regulating sensor usage and processing power based on the current operating conditions, ensuring prolonged operation and user convenience.

6.5 Conclusion

The SmartCane software system represents a comprehensive and innovative solution for enhancing navigation assistance and safety for visually impaired individuals. By leveraging cutting-edge technologies such as the Raspberry Pi 4 single-board computer, alongside computer vision libraries like OpenCV and TensorFlow Lite, and intuitive user interface design, the project aims to empower users with enhanced situational awareness and control over their surroundings. Through the integration of modules for object detection, user interface feedback, and device control, the software provides a seamless and user-friendly experience, enabling users to navigate their environment with confidence and ease. Meticulous planning, implementation, and attention to detail underscore the project's commitment to addressing real-world challenges and improving the quality of life for individuals with visual impairments.

CHAPTER 7

TESTING

7.1 Introduction

Testing is commonly performed in various domains, such as software development, manufacturing, engineering, and scientific research. Its primary goal is to identify defects, errors, or deviations from expected behavior and ensure that the system or product functionsas intended.

There are different methods that can be used for software testing. They are,

Black-Box Testing: Black-box testing is the process of testing an application without having any knowledge of its internal workings. The tester is unable to view the system's source code and is not familiar with its design. During a black-box test, a tester will often interact with thesystem's user interface by providing inputs and assessing outputs without being aware of howor where the inputs are processed.

White-Box Testing: White-box testing is a detailed examination of the internal structure and logic of the code. Other names used for white-box testing are glass testing and open-box testing. In order to do white-box testing on an application, a tester has to be conversant with the inner workings of the code. In order to locate the precise unit or portion of code that is performing inappropriately, the tester must look at the source code.

7.1.1 Types of Testing

- Unit testing: Unit tests are extremely straightforward and located close to the implementation's source code those involved in testing techniques. Functional domains for the programmers, pieces, or parts that your apps utilize. Most of the time, a continuous development server can perform unit tests rapidly and affordably.

- Integration testing: Integration tests are a type of testing that ensures the various components or services used by your programme are operational. For example, it might focus on ensuring that micro service interactions happen as planned or testing data interactions. These tests cost more to do because the application's numerous components must function.

- Functional testing: Functional tests are primarily concerned with the business requirements of an application. When performing an action, they don't even look at the system's intermediate stages; instead, they only look at the result of the operation. Integration examinations and functional examinations are occasionally mistaken for one another since they both call for a number of elements that communicate with one another. A functionality test will assume that specific standards will be obtained from the directory in accordance with outcome Dept. of CSE, Ramaiah Institute of Technology 60 requirements, whereas a combined trail will merely demonstrate if the author can query the directory.

- End-to-end testing: End-to-end testing eliminates the need to use the structure across the entire application domain. The author demonstrates if a specific person can carry out a task as expected and can involve anything, be it as basic as loading a webpage or as challenging as sending out emails or making online payments. End-to-end tests are utilized, which can be expensive to execute and, if automated, difficult to manage. Less important edge testing or reliance on the testing's final phase are advised in order to detect breaking thinking.

7.2 Testing Tools and Environment

The Raspberry Pi is operated via RealVNC Server, which permits remote access and control. The SmartCane software system runs Python 3 code for object detection. This code is executed using the Thonny Editor IDE platform. The integration allows for efficient remote management of the Raspberry Pi. The combination enhances the functionality and accessibility of the SmartCane system. Together, they provide a robust solution for remote object detection operations.

7.3 Test Cases Design and Execution

Ensuring the reliability and effectiveness of the SmartCane System software is paramount to its success in enhancing navigation assistance and safety for visually impaired individuals. A comprehensive testing strategy is essential to validate the system's functionalities and performance across various scenarios.

Unit Testing forms the foundational phase, where each software module within the SmartCane System undergoes rigorous testing in isolation. Through unit testing frameworks like unittest or pytest in Python, test cases are designed to simulate diverse input scenarios and validate the expected outputs of each module. Object detection accuracy is scrutinized using controlled image datasets with known object locations, while UI component functionalities, such as audio alert generation, are thoroughly assessed. Unit testing also verifies device control functionalities to confirm proper communication with sensors like the camera and speakers.

Integration Testing follows, where successfully tested modules are integrated to evaluate their interaction and data flow. Real-world use cases are simulated to feed data from the object detection module to the path planning module (if applicable) and UI module, verifying the overall system behavior. This phase ensures smooth data exchange between all modules for seamless system operation, and assesses how the UI module translates object detection data into clear audio alerts and appropriate vibration patterns.

System Testing evaluates the entire SmartCane System as a whole in a simulated environment resembling real-world conditions. A simulated environment, possibly a virtual reality setup or a controlled physical environment, mimics real-world scenarios with objects, obstacles, and paths. Object detection accuracy is tested under various lighting conditions, distances, and object orientations, while the path planning functionality, if applicable, is assessed for suggesting safe paths considering detected obstacles. Usability and effectiveness of the non-visual user interface, particularly audio alerts, are evaluated from a user's perspective, focusing on clarity and ease of understanding.

User Testing, the final crucial phase, involves real visually impaired individuals interacting with the SmartCane System prototype in a safe and controlled environment. A diverse group of visually impaired individuals is recruited, provided with training on safely using the SmartCane System. Observations of user interaction with the system, including their ability to navigate using audio cues and vibration patterns, are gathered. User feedback on the clarity and usefulness of audio alerts and overall user experience is collected, informing refinements to the non-visual UI and improvements to the system's effectiveness for real-world use.

7.4 Experiential Result

The results for real-time object detection were highly promising, affirming the effectiveness of the implemented algorithms. The system demonstrated the capability to swiftly identify and track objects within the cane's vicinity, providing instantaneous feedback. This real-time functionality is crucial for users with visual impairments, enhancing their situational awareness and navigation.

MobileNetV2-SSD combines the MobileNetV2 architecture, optimized for mobile and embedded vision applications, with the Single Shot Multibox Detector (SSD), a widely used object detection framework. This integration allows for efficient object detection with good accuracy, making it suitable for deployment on edge devices and real-time applications. Through the utilization of depth-wise separable convolutions and inverted residual blocks, MobileNetV2-SSD achieves computational efficiency, requiring fewer resources while maintaining satisfactory accuracy levels. Its efficient architecture enables fast inference speeds, facilitating real-time object detection scenarios where timely processing is essential. While it may not achieve the highest accuracy compared to some other models, MobileNetV2-SSD strikes a balance between speed and accuracy, making it adaptable to a wide range of real-world applications.

YOLO v3 (You Only Look Once version 3) is a state-of-the-art object detection model celebrated for its exceptional speed and accuracy. Its architecture involves dividing the input image into a grid and directly predicting bounding boxes and class probabilities from the entire image. This approach streamlines the detection process, enabling swift inference speeds. YOLO v3's ability to process images in real-time or near real-time on robust hardware is widely acknowledged. Moreover, it excels in accuracy, particularly in detecting small objects, thanks to its capability to identify objects at multiple scales within a single pass. However, it's important to note that YOLO v3 is more computationally intensive compared to models like MobileNetV2-SSD. This heightened complexity necessitates more powerful hardware for optimal performance, limiting its suitability for deployment on resource-constrained devices or real-time applications with stringent latency requirements.

Mean Average Precision (mAP) is a performance metric that is frequently used in object detection tasks in deep learning to assess the model's accuracy. By evaluating the accuracy and recall of objects identified at different thresholds, it offers a thorough evaluation of a model's capacity to accurately locate and identify objects in an image. The area under the Precision-Recall (PR) curve for each class is called the Average Precision (AP), and mAP is computed by averaging the AP for all classes. Plotting precision versus recall at various confidence levels, the PR curve shows how the two criteria are traded off. Better model performance is shown by a greater mean annual percentage point (mAP), which shows that the model can reliably detect items that have high precision and recall among many categories. For applications requiring accurate and dependable object identification, such as autonomous driving, monitoring, and imaging for medical purposes, this metric is critical.

In deep learning, intersection over union (IoU) plays a vital role in determining mean average precision (mAP), especially when it comes to object detection. The overlap between an object's actual bounding box and its expected bounding box is measured by IoU. It can be expressed as the area where the anticipated and actual boxes intersect divided by the area where they unite. A predicted bounding box is classified as a true positive (if it overlaps with the ground truth sufficiently) or a false positive (if the overlap is not adequate) using the IoU method. Various IoU thresholds (0.5, 0.75, etc.) are employed in the computation of mAP to assess the recall and precision of detections. Only forecasts that exhibit a significant degree of overlap with the actual truth are deemed accurate when the IoU criterion is raised. This contributes to a more thorough assessment of the model's performance by evaluating the model's accuracy both in identifying the presence of items and accurately finding them within an image.

When comparing the object detection algorithms MobileNet-SSD and YOLO v3, several key metrics highlight their differences. MobileNet-SSD achieves a higher accuracy of 80.5%, significantly outperforming YOLO v3, which has an accuracy of 60.6% as observed in figure 7.1. Additionally, MobileNet-SSD is more computationally efficient with 36,000 million Multiply-Adds compared to YOLO v3's 65,000 million, making it a better choice for applications requiring faster processing as depicted in figure 7.2. Furthermore, MobileNet-SSD has a smaller model size of 26 MB, which is less than half the size of YOLO v3's 61 MB, making MobileNet-SSD more suitable for deployment in environments with limited storage and computational resources as seen in the figure 7.3. Overall, MobileNet-SSD offers a balanced trade-off between high accuracy, lower computational demands, and smaller model size, making it ideal for real-time object detection in resource-constrained scenarios, as shown in Table 3.

Table 3: Comparison of Object Detection Algorithms

Metric	MobileNet-SSD	YOLO v3
Accuracy	80.5%	60.6%
Multiply-Adds (Millions)	36000	65000
Model Size (MB)	26	61

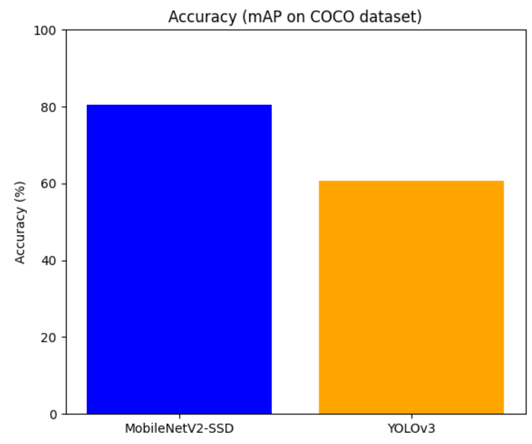


Figure 7.1: Accuracy Analysis

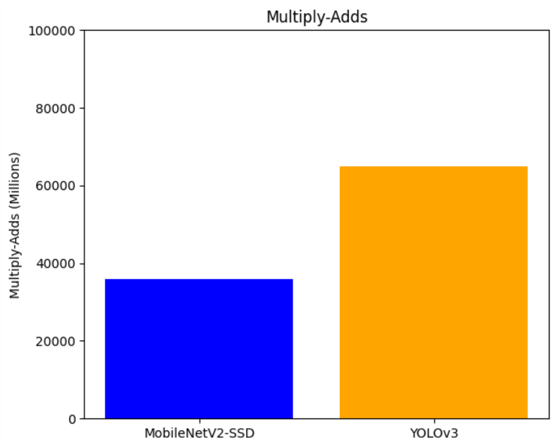


Figure 7.2: Analysis of Multiply-Adds (Millions)

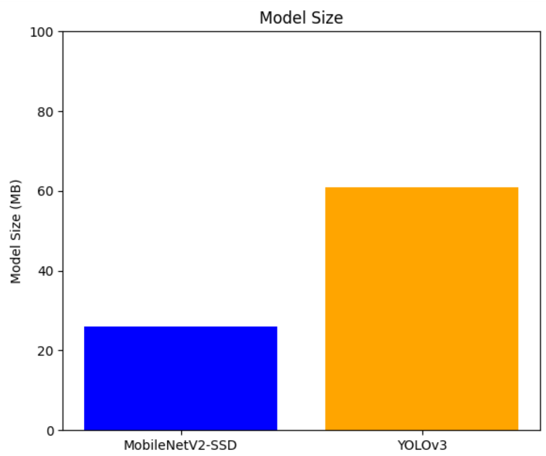


Figure 7.3: Analysis of Model Size (MB)

Here is why MobileNet SSD is best for Object detection in Real-time

- Speed: In comparison to YOLO v3, MobileNet-SSD provides quicker detection speed. Rapidity is essential for prompt answers in real-time applications, particularly those that include surveillance or video analysis.
- Decent Accuracy: While MobileNet-SSD still gets a reasonable mAP, YOLO v3 offers a bit greater mAP at 0.5 IOU, making it more appropriate for a wide range of real-world applications.
- Optimal Threshold: The best possible mAP of MobileNet-SSD was achieved at a threshold of 0.5 IOU, which is in good accordance with the usual specifications of numerous object identification jobs.
- For practicality and accessibility, object detection operating on hardware components like an Arduino SSD or Raspberry Pi is ideal.
- Trade-off between Speed and Accuracy: The balance between detecting speed and accuracy is properly maintained by MobileNet-SSD. Compared to YOLO v3, it is much faster and has adequate accuracy, which makes it more appropriate for applications in real time where accurate identification is crucial.

Hence, MobileNet-SSD appears to be a better choice for real-time object detection due to its combination of speed, accuracy, and practical hardware requirements.

Steps of MobileNet-v2 SSD Algorithm

1. **Input Processing**: Receive an input image, which is preprocessed to a fixed size (e.g., 320x320 pixels) and normalized to values between 0 and 1.
2. **Feature Extraction**: Use MobileNetV2 to extract high-level features from the preprocessed image using depthwise separable convolutions. Capture features from multiple layers of MobileNetV2 to detect edges, textures, and complex patterns.
3. **Feature Scaling**: Scale the extracted features through additional convolutional layers to different resolutions. Generate multi-scale feature maps to detect objects of varying sizes in the image.

4. **Prediction:** Predict class scores (probabilities of object classes) and bounding box coordinates for each location on the feature maps. Utilize predefined anchor boxes of various sizes and aspect ratios to assist in predicting object locations.
5. **Filtering Detections**: Non-Maximum Suppression (NMS) refines predicted detections by first removing boxes with low confidence scores. Then, it suppresses overlapping bounding boxes by retaining only those with the highest confidence scores and IoU (Intersection over Union) greater than a specified threshold (e.g., IoU > 0.5), ensuring only the most confident and non-overlapping bounding boxes are retained as final detections.

Figure 7.4: SmartCane

The SmartCane integrates several hardware components, including a Pi camera for object detection, a speaker for speech output, and a panic button to inform guardians, as shown in Figure 7.4.

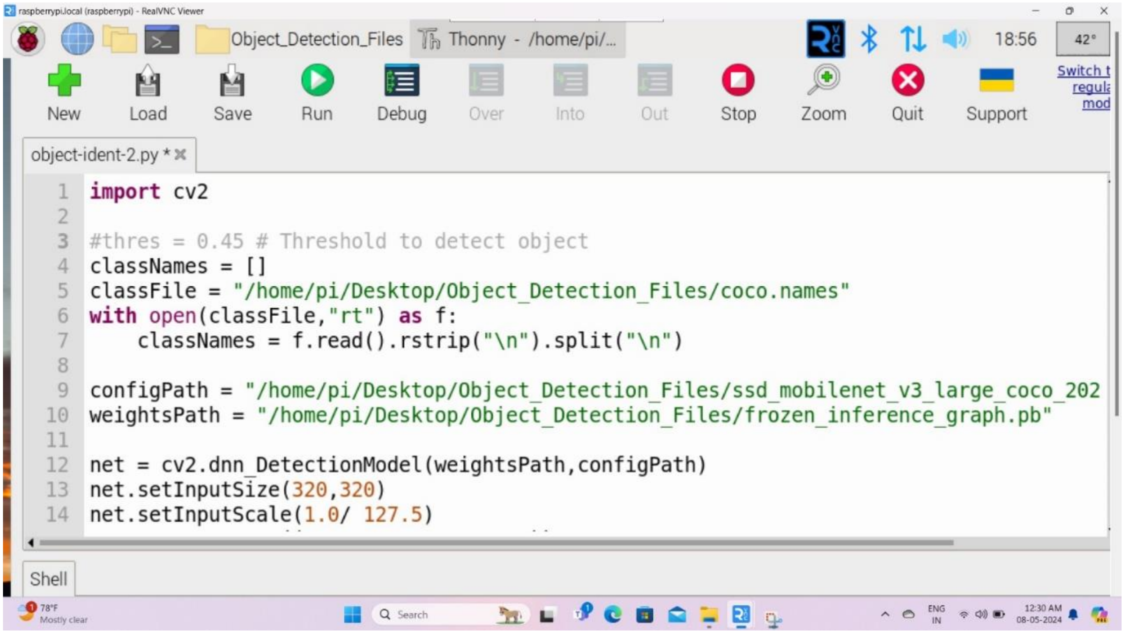


Figure 7.5: Object detection code

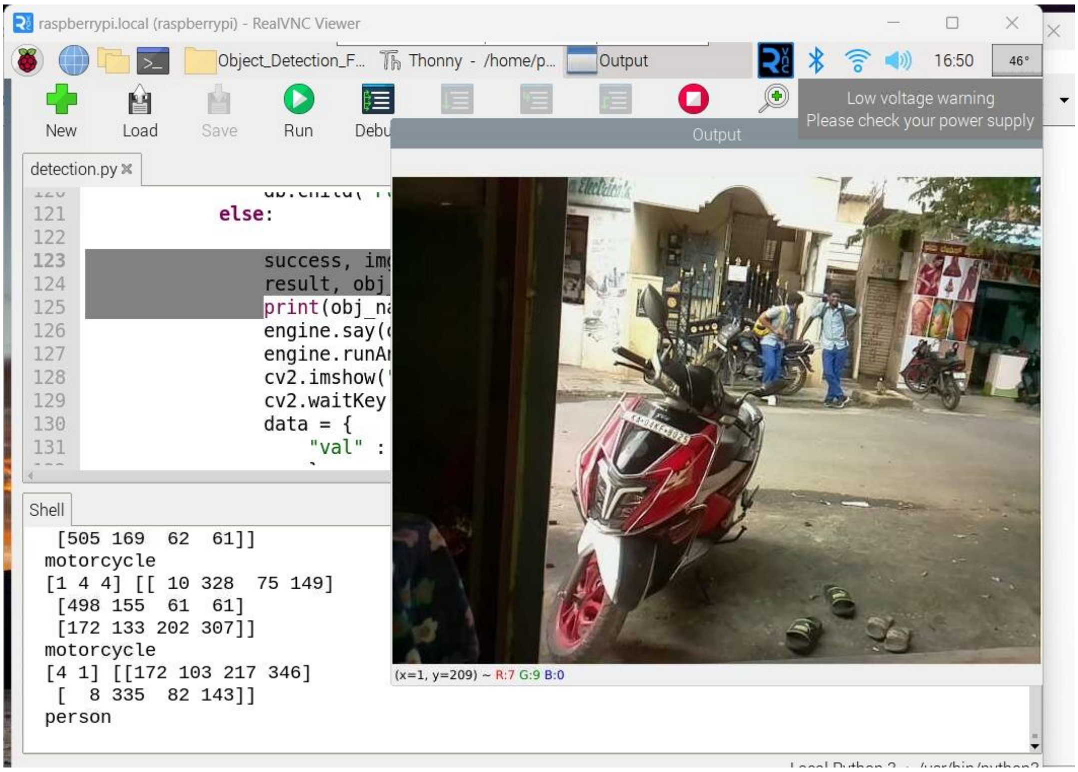


Figure 7.6: Vehicle Detection in Object Recognition Module

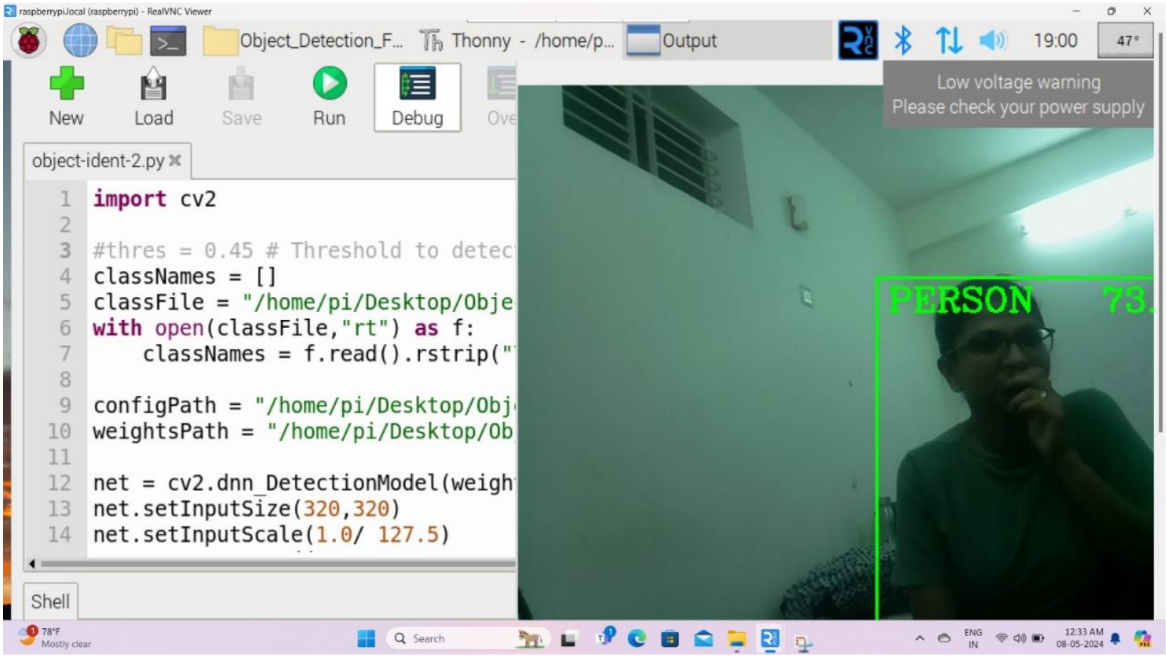


Figure 7.7: Pedestrian Detection

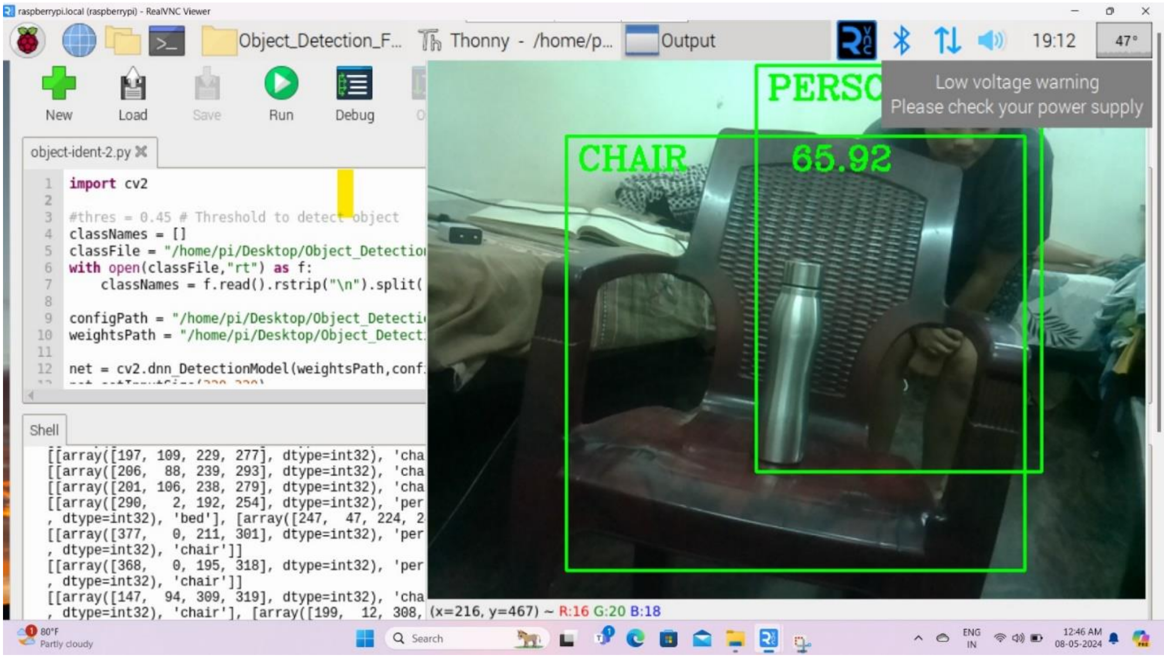


Figure 7.8: Objects Detection-1

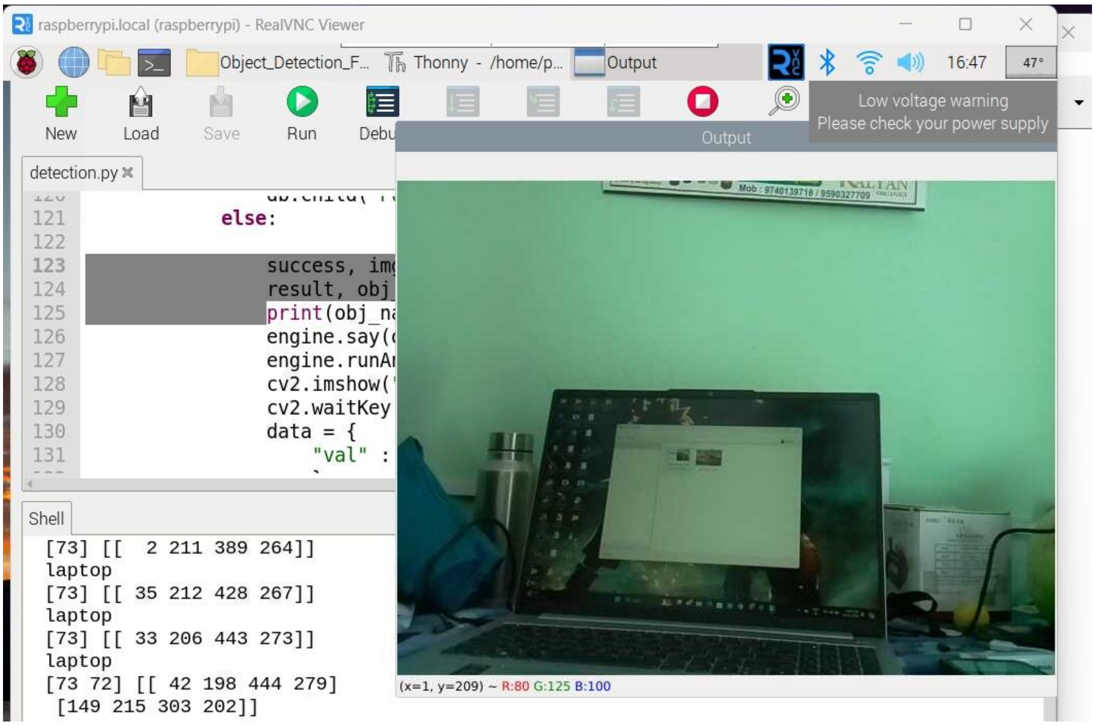


Figure 7.9: Object Detection-2

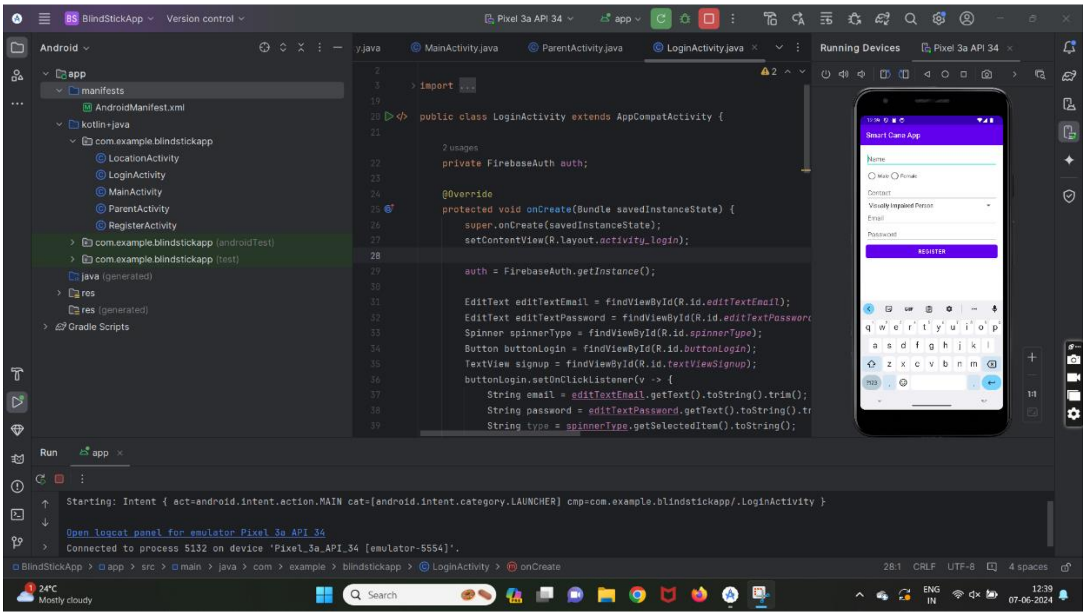


Figure 7.10: User Signup Process for Mobile App Registration

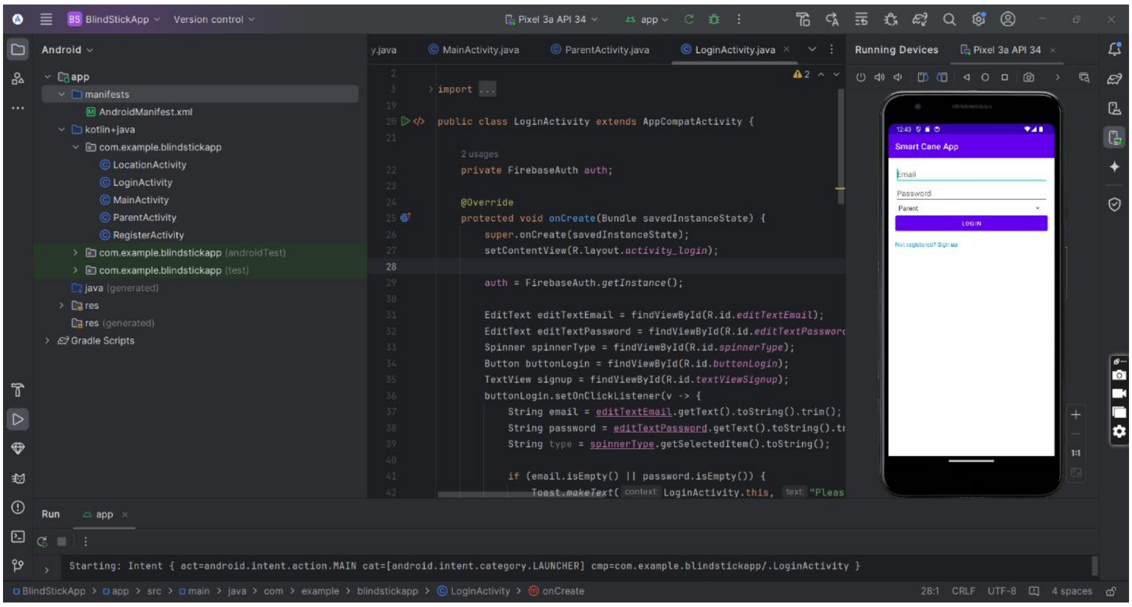


Figure 7.11: User Login Page with Accessibility Options for Visually Impaired and Parent/Guardian Logins

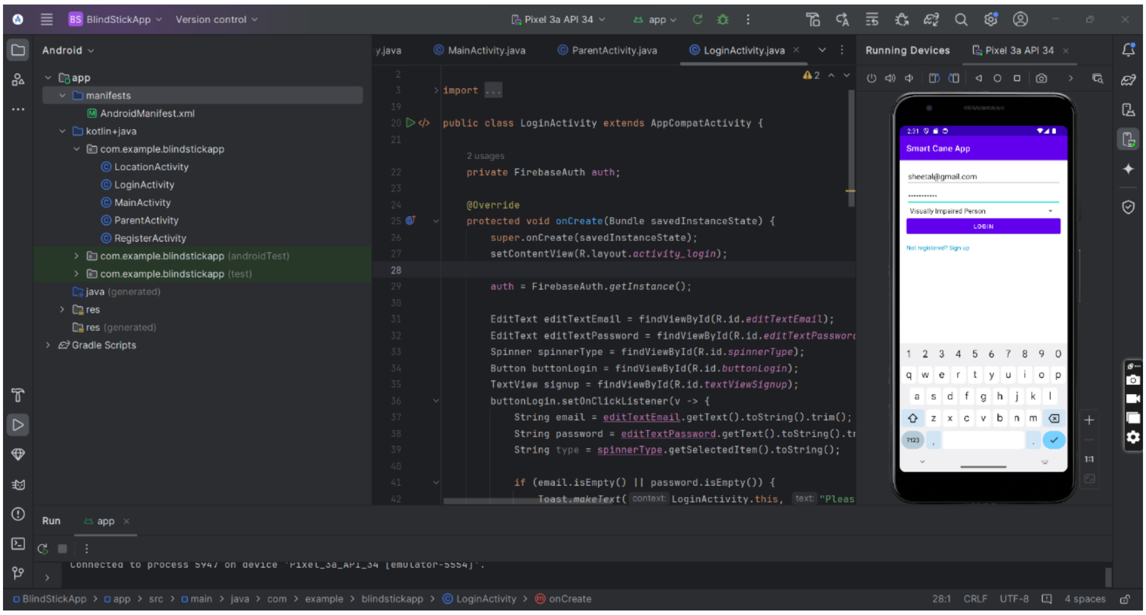


Figure 7.12: Login Interface for Visually Impaired Users

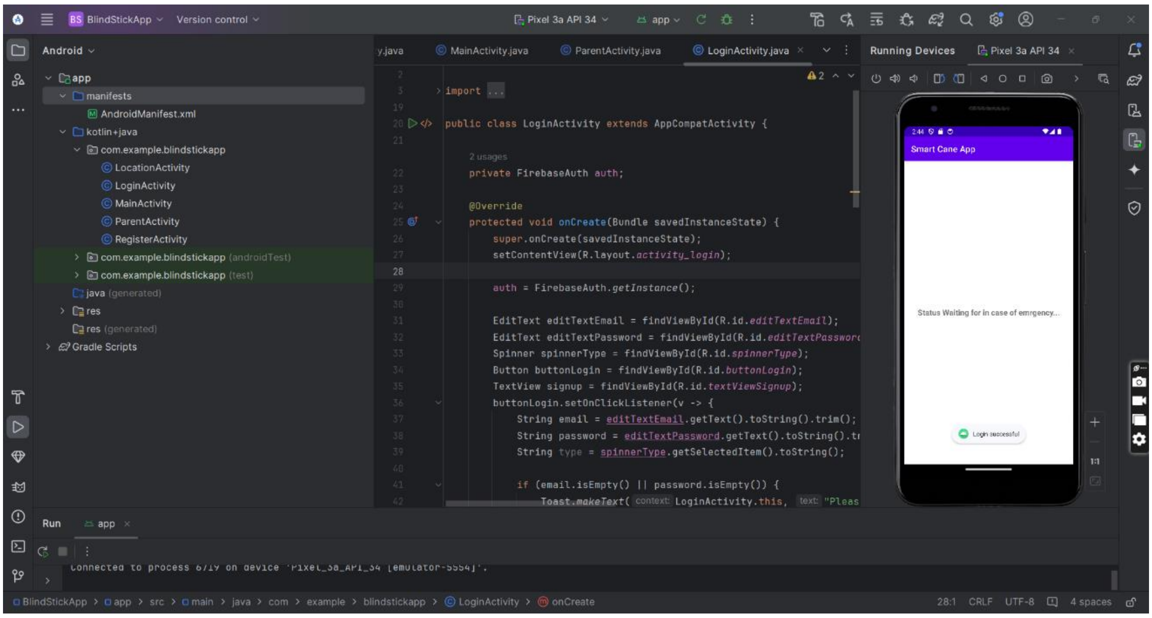


Figure 7.13: Waiting Status in Emergency Situations

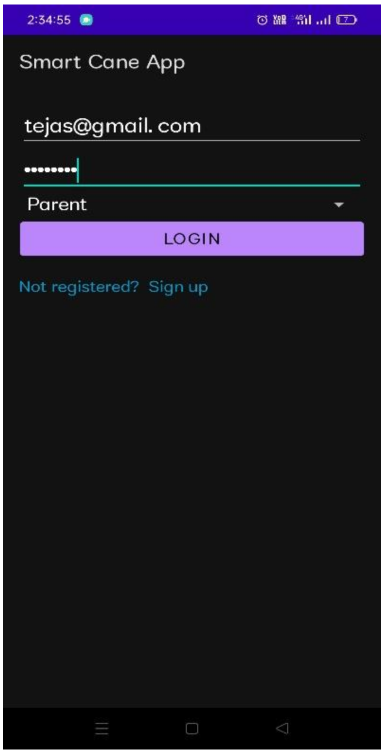


Figure 7.14: Parent Login Process

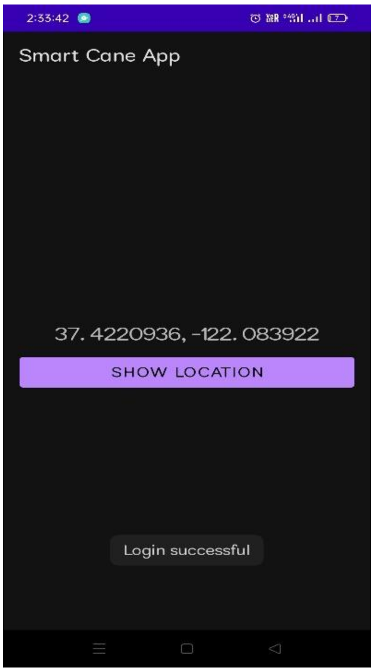


Figure 7.15: Location Coordinates of Visually Impaired Person in Panic Displayed in Parent's Mobile App

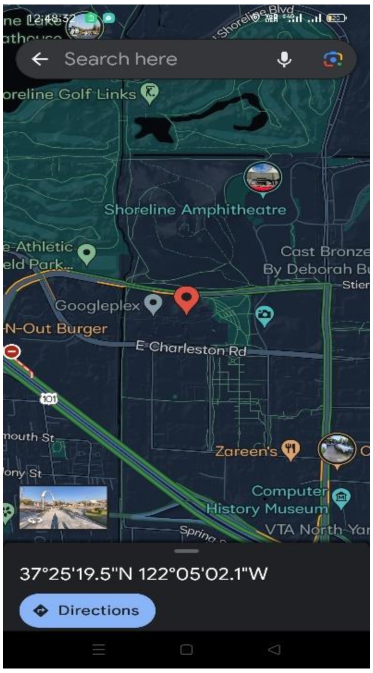


Figure 7.16: Parent Viewing Location Coordinates of Visually Impaired Person via Google Maps

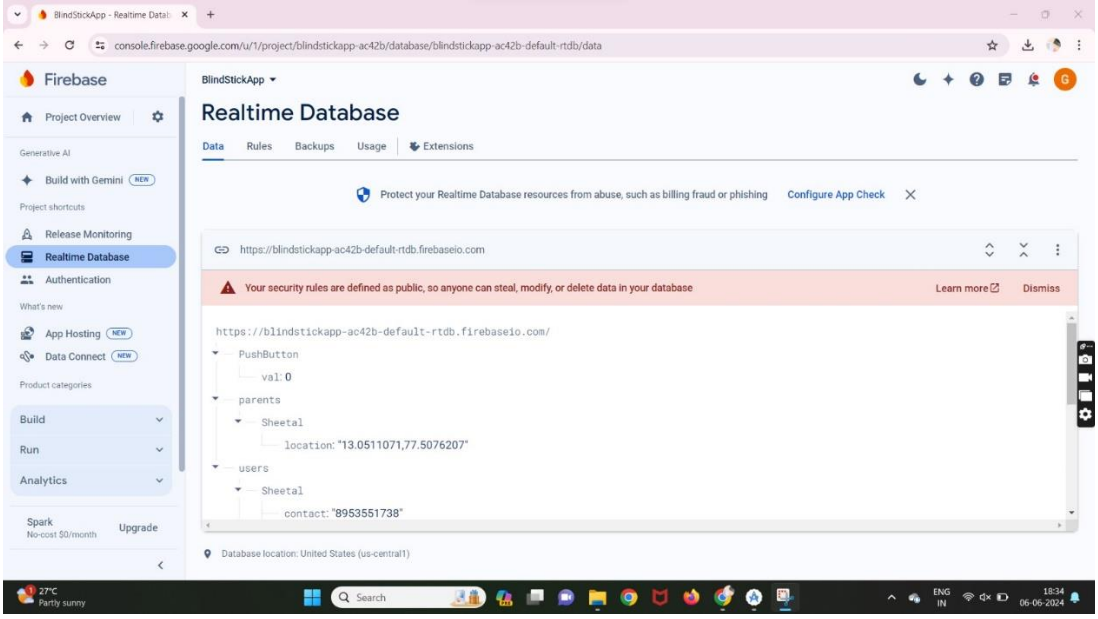


Figure 7.17: Safety Status of Users in Firebase (Value 0 Represents Safety)

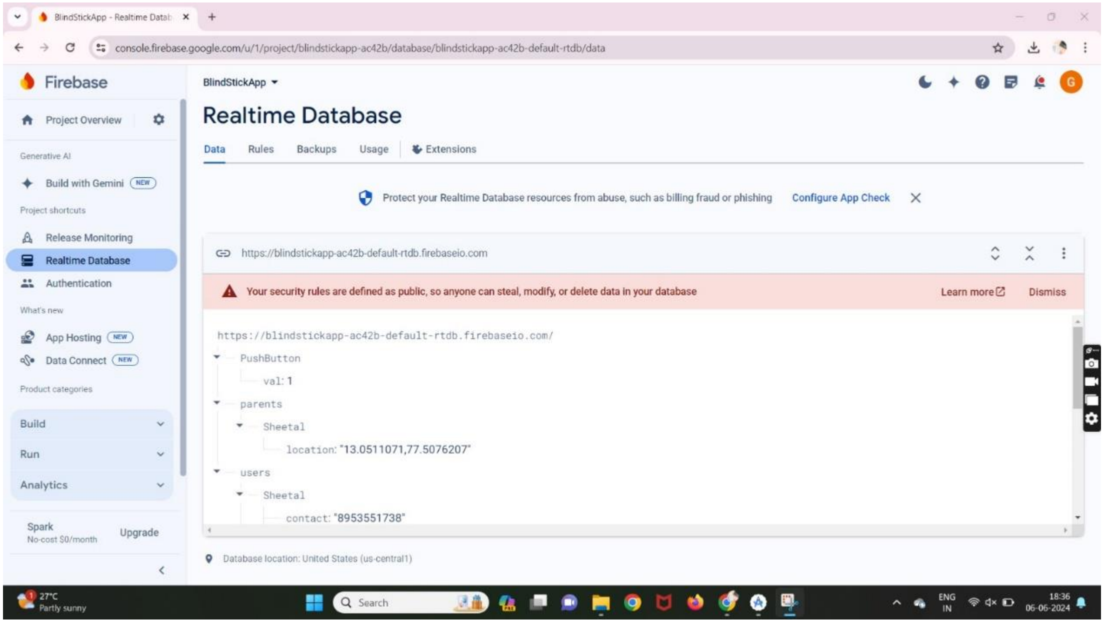


Figure 7.18: User Data Safety Status in Firebase (Value 1 Indicates Unsafe/Panic)

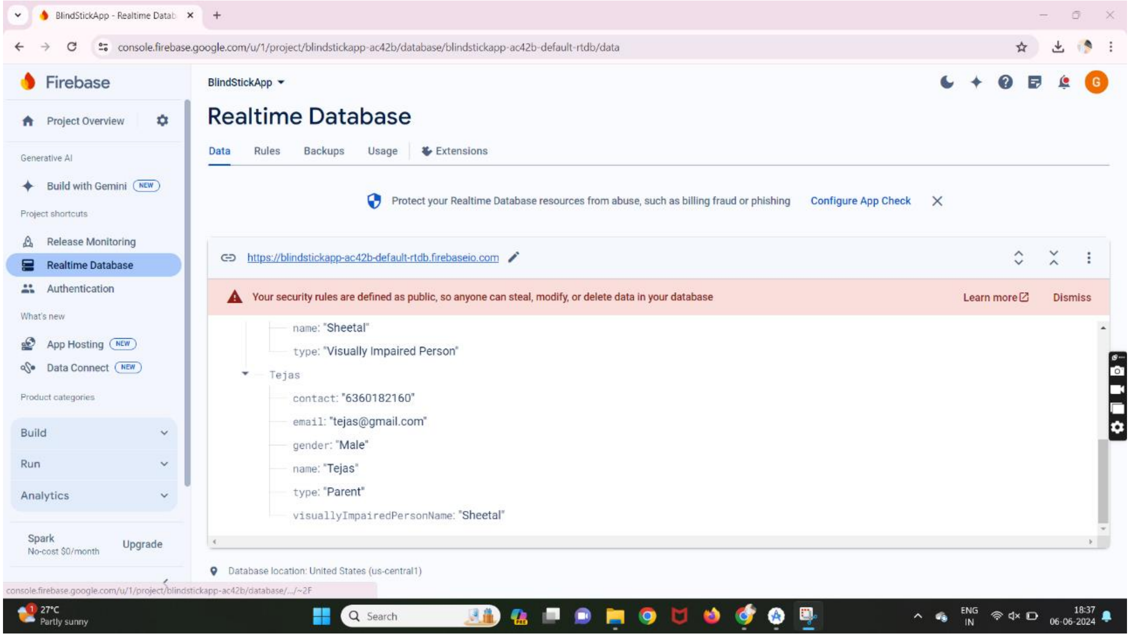


Figure 7.19: Firebase Database Structure for Visually Impaired Person and Parent Login Data

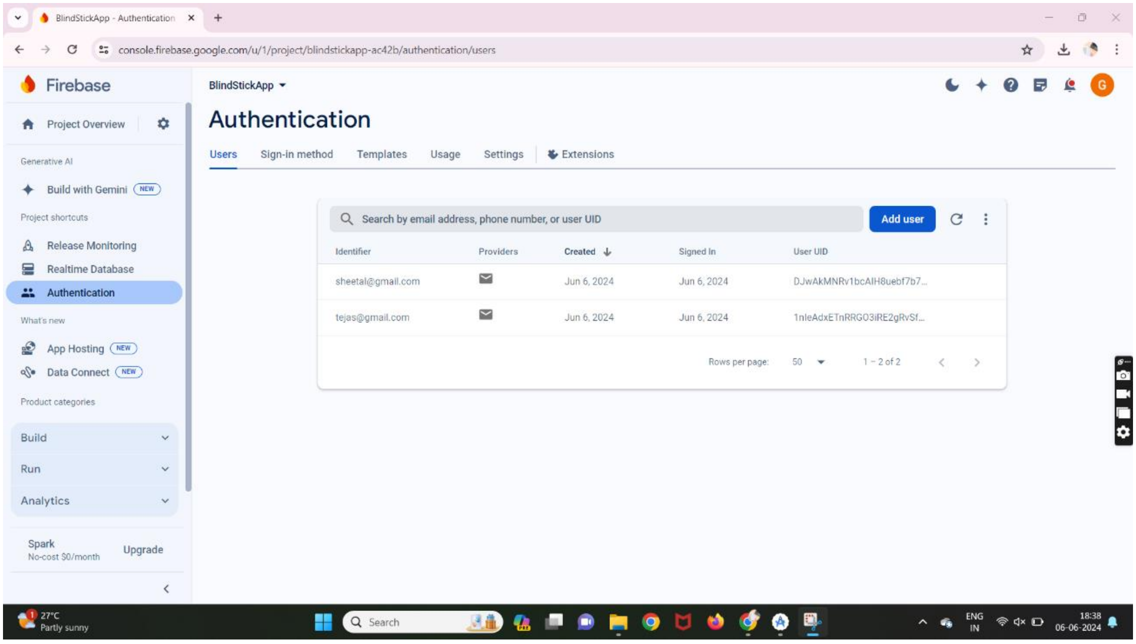


Figure 7.20: Authentication Details for Both Parents and Visually Impaired Individuals

CHAPTER 8

CONCLUSION

AI-based smart cane stands as a noteworthy contribution to assistive technology for the visually impaired. Smart cane is designed using Raspberry Pi and a camera for object detection. Deep learning model such as MobileNetV2 and YOLO v3 for object detection provides quicker object detection. Smart Cane is supported with many notable features, including audio feedback through a speaker and a panic button that notifies guardians with the user's current location, enhance mobility but also prioritize safety. With its user-friendly design, this innovative solution holds the potential to significantly elevate the overall quality of life for individuals with visual impairments. The SmartCane proves to be a valuable tool for individuals with visual impairments, offering enhanced ease in navigating their surroundings. Its capabilities extend beyond traditional canes, providing advanced information and guidance to boost the confidence and independence of users in their daily activities.

Smart cane is merely an experimental model; it may be improved in terms of appearance or functionality. Future work is to add a GPS (Global Positioning System) module to track the smart cane's location in case it is lost or to help the user would be one possible improvement. An additional feature might be the addition of a fall sensor, which would keep an eye on the user's health while they use the system and guard against the chance of them running into an obstruction and falling. Also to develop the can with vibration motors that would minimize the number of parts in the system and enhance the cane's aesthetics.

REFERENCES

[1] Pydala, B., Kumar, T. P., & Baseer, K. K. (2023). Smart_Eye: A Navigation and Obstacle Detection for Visually Impaired People through Smart App. Journal of Applied Engineering andTechnological Science (JAETS), 4(2), 992-1011.

[2] Khan, D. A., Zamir, M. A., Umar, M. S., & Haider, Z. (2022, November). Assistive Stick for Visually Impaired People. In 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT) (pp. 1-6). IEEE.

[3] Islam, R. B., Akhter, S., Iqbal, F., Rahman, M. S. U., & Khan, R. (2023). Deep learning basedobject detection and surrounding environment description for visually impaired people. Heliyon,9(6).

[4] Uma, R., Harish, P. S., Ragav, N. G., & Navin, A. (2023, February). Third Eye–A Smart Wearable Glass With Deep Learning Technology Powered With Artificial Intelligence for Visually Impaired People. In 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-4). IEEE.

[5] Muhammad, Y., Jan, M. A., Mastorakis, S., & Zada, B. (2022, August). A deep learning- based smart assistive framework for visually impaired people. In 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS) (pp. 1-6). IEEE.

[6] Ashiq, F., Asif, M., Ahmad, M. B., Zafar, S., Masood, K., Mahmood, T., ... & Lee, I. H. (2022). CNN-based object recognition and tracking system to assist visually impaired people. IEEE Access, 10, 14819-14834.

[7] Christopherson, P. S., Eleyan, A., Bejaoui, T., & Jazzar, M. (2022, November). Smart stick for visually impaired people using raspberry pi with deep learning. In 2022 InternationalConference on Smart Applications, Communications and Networking

(SmartNets) (pp. 1-6). IEEE.

[8] Kumar, N., & Jain, A. (2022). A Deep Learning Based Model to Assist Blind People in TheirNavigation. Journal of Information Technology Education: Innovations in Practice, 21, 095-114.

[9] Telkar, N., Hiremath, P., Ambika, K. R., Prathap, M., & Harashetti, R. (2023). Vinetra Parijana: Smart Cane and Smart Glasses for Visually Impaired People.

[10] Sravani, N., Yamuna, P., Deepika, M., Padmavathi, M., Sukanya, K., & Akheel, T. S. (2023). SMART WALKING ASSISTIVE DEVICE FOR BLIND PEOPLE USING ARDUINO. Journal of Engineering Sciences, 14(05).

[11] Kuriakose, B., Shrestha, R., & Sandnes, F. E. (2023). DeepNAVI: A deep learning based smartphone navigation assistant for people with visual impairments. Expert Systems with Applications, 212, 118720.

[12] Kadhim, M. R., & Oleiwi, B. K. (2022). Blind assistive system based on real time object recognition using machine learning. Engineering and Technology Journal, 40(1), 159-165.

[13] Zafar, S., Asif, M., Ahmad, M. B., Ghazal, T. M., Faiz, T., Ahmad, M., & Khan, M. A. (2022). Assistive devices analysis for visually impaired persons: a review on taxonomy. IEEE Access, 10, 13354-13366.

[14] Leong, X., & Ramasamy, R. K. (2023). Obstacle Detection and Distance Estimation for Visually Impaired People. IEEE Access, 11, 136609-136629.

[15] Jouini, K., Maaloul, M. H., & Korbaa, O. (2021, October). Real-time, CNN-based assistivedevice for visually impaired people. In 2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) (pp. 1-6). IEEE.

[16] Tarik, H., Hassan, S., Naqvi, R. A., Rubab, S., Tariq, U., Hamdi, M., ... & Cha, J. H. (2023). Empowering and conquering infirmity of visually impaired using AI-technology equipped with object detection and real-time voice feedback system in healthcare application. CAAI Transactions on Intelligence Technology.

[17] Shendge, R., Patil, A., & Kadu, S. (2022). Smart Navigation for Visually Impaired people using Artificial Intelligence. In ITM Web of Conferences (Vol. 44, p. 03053). EDP Sciences.

[18] Mohamed, I., Salah, M., & Farghal, A. (2023). Camera-Based Navigation System for Blindand Visually Impaired People. Sohag Engineering Journal, 3(1).

[19] Kaur, J., & Singh, W. (2022). Tools, techniques, datasets and application areas for object detection in an image: a review. Multimedia Tools and Applications, 81(27), 38297-38351.

[20] Abiramee, M. R., Nithya, M. P., Kumar, M. M. N., Muhammed, M. T. S., & Sowmiya, M. S. (2023). SMART BLIND STICK USING VOICE MODULE.

[21] Technology, S.F.: About the UltraCane. https://www.ultracane.com/about_the_ultracane. Accessed 4 February 2020.

[22] Borenstein, J., Ulrich, I.: The GuideCane-applying mobile Robot technologies to assist thevisually impaired. IEEE Trans. Syst. Man Cybern. Syst. Hum. 31(2), 131–136 (2001). https://doi.org/10.1109/3468.911370

[23] Terlau, T.: 'K' Sonar Curriculum Handbook. American Printing House for the Blind, USA (2008)

[24] Rohan, P.: 'Smart' cane for the visually impaired: technological solutions for detecting knee-above obstacles and accessing public buses. In: 11th International Conference on Mobility and Transport for Elderly and Disabled Persons (2007)

[25] Chang, T.H.: iCane-A partner for the visually impaired. In: IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (2005)

[26] Yusro, M., Hou, K.: SEES: Concept and design of a smart environment explorer stick. In: 6th International Conference on Human System Interactions (HSI), pp. 70–77 (2013). https://doi.org/10.1109/HSI.2013.6577804