

GOOGLE PI USING RASPBERRY PI FOR VISUALLY IMPAIRED

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering
with Internet of Things

By

GOWTHAMI VINOD KATKAR (Reg.No - 40732001)

ABIRAMI K (Reg.No - 40110021)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

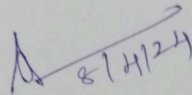
**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **K ABIRAMI (40110021)**, **GOWTHAMI VINOD KATKAR (40732001)** who carried out the Project entitled "**GOOGLE PI USING RASPBERRY PI FOR VISUALLY IMPAIRED**" under my supervision from November 2023 to April 2024.


8/4/24

Internal Guide

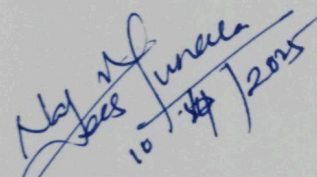
Dr. MARY POSONIA M.E., Ph.D.



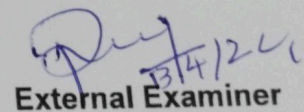
Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 13th April 2024


10/4/2024

Internal Examiner


13/4/24

External Examiner

DECLARATION

I, **GOWTHAMI VINOD KATKAR (40732001)**, hereby declare that the Project Report entitled “**GOOGLE PI USING RASPBERRY PI FOR VISUALLY IMPAIRED**” done by me under the guidance of **Dr. MARY POSONIA, M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering with Internet of Things**.

DATE: 13/04/2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Mary Posonia M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering with Internet of Things** who were helpful in many ways for the completion of the project.

ABSTRACT

In this day and age, voice assistant devices have become popular due to their convenience, accessibility, and integration into various aspects of daily life. Their continued evolution and integration with smart technologies suggest that their popularity is likely to endure and expand even further in the coming years. At present, Voice assistant devices, such as Amazon Echo, Google Home, and Apple's Siri, have gained immense popularity. Likewise, Google Pi is also one kind of voice assistant device which is specifically designed to cater the unique needs of visually impaired individuals, Google Pi stands as a testament to how technology can be harnessed to overcome societal challenges and stigmas. Google pi employs a versatile and affordable single-board computer called raspberry pi as its core component that serves as the foundation upon which the entire system is built with an attached Respeaker - 4 Mic array to receive voice commands from the distance with exceptional accuracy. Besides, it has also been customized with google API for voice assistance. This customization not only enhances the user experience but also makes Google Pi an efficient tool for visually impaired individuals who rely on voice commands to navigate their daily lives. Beyond voice commands, this device for the visually impaired also includes two additional primitive features namely, object recognition and currency recognition that are incorporated through the application of deep learning algorithms interfacing Web camera or Mobile with the Raspberry Pi to enable this real time recognition facilities. However, the development of an efficient system of transceiving the voice command inputs and respond immediately is a challenging task as it requires accurate interfacing and robust algorithms. This study takes a novel approach by harnessing the power of Google's API for voice assistance and leveraging deep learning algorithms for real-time recognition features. When paired with proper guidance, the said hybrid approach would produce the best solution in real-time

TABLE OF CONTENT

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
	2.1 Inferences from Literature Survey	7
	2.2 Open problems in Existing System	9
3	REQUIREMENTS ANALYSIS	9
	3.1 Feasibility Studies/Risk Analysis of the Project	9
	3.2 Software Requirements Specification Document	9
4	DESCRIPTION OF PROPOSED SYSTEM	13
	4.1 Selected Methodology or process model	13
	4.2 Architecture / Overall Design of Proposed System	15
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	19
	4.4 Project Management Plan	19
	4.5 Project transition / software to operation plan	21
5	IMPLEMENTATION DETAILS	25
	5.1 Development and Deployment Setup	25
	5.2 Algorithms	31
6	RESULTS AND DISCUSSION	35
7	CONCLUSION	38
	7.1 Conclusion	38
	7.2 Future Work	38
	7.3 Research Issues	39

7.4	Implementation Issues	40
	REFERENCES	42
	APPENDIX	45
	A. SOURCE CODE	45
	B. SCREENSHOTS	49
	C. RESEARCH PAPER	52

LIST OF FIGURES

FIGURE NO	FIGURENAME	Page No
2.1	study on adaptive tools for visually inconvenienced community by ScienceDirect	6
4.1	Functional block diagram of a Google Pi	14
4.2	A Representation of a Google Pi System using Raspberry Pi	14
4.3	System Architecture	15
4.4	Benefits of CI/CD	22
5.1	Raspberry Pi 4	25
5.2	Pi Camera	26
5.3	Hardware Setup	27
5.4	Raspian OS	28
5.5	Registration & Enabling API	28
5.6	Downloading dependencies	29
5.7	Applied algorithms on object recognition	32
5.8	Applied algorithms on sign language dataset	33
6.1	Graph of training vs validation accuracy	35
6.2	Object recognition result	36
6.3	Confidence score based on bounding box criteria	36

LIST OF ABBREVIATIONS

ABBREVIATIONS

EXPANSION

VPA	Virtual Personal Assistant
SRS	System Requirement Specification
API	Application Programming Interface
CNN	Convolution Neural Network
YOLOV3	You Look Only Once version 3
CSI	Camera Serial Interface
HDMI	High- definition Multimedia Interface
SRS	System Requirement Specification
OS	Operating System
SDK	Software development kit
OpenCV	Open Source Computer Vision Library

CHAPTER 1

INTRODUCTION

In the fast-paced world of today, the rise of voice assistant devices has been nothing short of remarkable. These devices have surged in popularity thanks to their unparalleled convenience, widespread accessibility, and seamless integration into various aspects of our daily lives. From controlling smart homes to providing information and entertainment, voice assistants like Amazon Echo, Google Home, and Apple's Siri have become indispensable companions in countless households. Furthermore, emerging technologies, such as Google Pi, have been developed with a focus on serving specific needs, in this case, visually impaired individuals, thus expanding the horizons of voice assistant technology.

As we delve deeper into the intricacies of Google Pi, it becomes apparent that it represents a significant stride forward in the world of assistive technology. Specifically designed to cater to the unique needs of visually impaired individuals, Google Pi stands as a testament to how technology can be harnessed to overcome societal challenges and stigmas.

At the heart of Google Pi lies a remarkable piece of technology, the Raspberry Pi. This versatile single-board computer serves as the foundation upon which the entire system is built. The Raspberry Pi's compact form factor and low power consumption make it an ideal choice for such applications. Paired with a Respeaker - 4 Mic array, Google Pi is equipped to receive voice commands from a distance with remarkable accuracy. This combination of hardware ensures that the device can effectively capture user inputs, a crucial aspect of any voice assistant device.

One of the defining features of Google Pi is its integration with Google's powerful API for voice assistance. Google's voice recognition technology is renowned for its accuracy and versatility, and by incorporating this API, Google Pi can provide users with a user-friendly interface that responds to natural language commands. This customization not only enhances the user experience but also makes Google Pi an effective and efficient tool for visually impaired individuals who rely on voice commands to navigate their daily lives.

While voice commands are central to the functionality of Google Pi, this assistive device goes above and beyond by incorporating two additional key features: object

recognition and currency recognition. These features are made possible through the application of deep learning algorithms, which allow Google Pi to interact with its environment in a more meaningful way.

Object recognition is a critical component for visually impaired individuals as it enables the device to identify and describe objects in real-time. By interfacing with a web camera or mobile device, Google Pi can process visual information and provide spoken descriptions, allowing users to better understand and navigate their surroundings. This capability represents a significant step forward in enhancing the independence and quality of life for visually impaired individuals.

Currency recognition is another noteworthy feature of Google Pi. In an increasingly cashless society, it can be challenging for visually impaired individuals to manage their finances independently. Google Pi's ability to recognize and differentiate between different denominations of currency bills addresses this issue. With this feature, users can accurately identify and manage their money, reducing the reliance on others for financial assistance.

Consequently, developing an efficient system for transceiving voice command inputs and responding promptly poses several challenges. It necessitates precise interfacing between hardware components, robust algorithms for voice recognition, and an infrastructure capable of handling real-time processing. To address these challenges, Google Pi employs a novel approach by harnessing the power of Google's API for voice assistance and leveraging deep learning algorithms for real-time recognition features.

The use of G

oogle's API for voice assistance simplifies the user experience by providing a seamless and familiar interface. Moreover, Deep learning algorithms play a pivotal role in the real-time recognition capabilities of Google Pi as these algorithms are capable of processing vast amounts of data, enabling the device to recognize objects and currency accurately. Furthermore, the ability to adapt and learn from new inputs ensures that Google Pi can continually improve its recognition capabilities over time.

As technology continues to advance, the future of Google Pi and similar assistive devices appears promising. In conclusion, the rise of voice assistant devices like Google Pi represents a significant milestone in the intersection of technology and accessibility.

Google Pi's innovative use of the Raspberry Pi, Google's API for voice assistance,

and deep learning algorithms demonstrate how technology can be harnessed to address real-world challenges. With the right guidance and support, the synergy of technology and human expertise can bring about meaningful change, fostering greater independence and inclusivity for everyone.

CHAPTER 2

LITERATURE SURVEY

This problem statement has been extensively studied over the past 5 years by researchers and automotive companies in a bid to create a solution, and all their solutions vary from analyzing various parameters to implement and customize the versatile system to produce the best solution.

The work of Gabriel Gpresc [1] detailed the possibility of using the Google assistant on a Raspberry Pi microcomputer. The paper not only provides step-by-step instructions for setting up Google Assistant on a Raspberry Pi but also emphasizes the advantages of this integration, particularly the ability to interface with other hardware devices

The necessary steps to implement the voice Google assistant on a system that contains among the Raspberry Pi only a microphone and a speaker, instead of the Voice Kit, are presented. Thus, the work of Gabriel Gpresc proposed how a voice Google assistant can be made more easily and flexible. This newly created device has several working modes that are analyzed and are presented in this paper.

Riya Vishwasi [2] has proposed A Raspberry Pi based CNN model that focuses on leveraging Raspberry Pi and Convolutional Neural Networks (CNNs) for Indian currency detection, catering to visually impaired individuals. The project combines computer vision and hardware integration to address a real-world problem. The currency note is captured through a camera which is then given to a Machine Learning model. The model is based on convolution neural network which helps in image processing and classification. Lastly, the results are delivered through a speaker.

She has proposed a model which can detect Indian currency notes of all denominations with an accuracy score of 99.7% with 250 epochs. The convolution neural network model used is VGG 16 which helps in extracting various features of the note and finally classifying the note.

Vedat Marttin [3], proposed a work for object detection based on raspberry pi that aims to recognize and draw objects by using TensorFlow, OpenCV and BrachioGraph libraries together with a drawing machine created with hardware.

He has used a Raspberry Pi 3 microcontroller with 1.2 GHz ARMv8 quad-core processor, Picamera, and 12g Emax servo motors as hardware. Object recognition process was carried out in the software part with the help of TensorFlow and OpenCV libraries. In this study, by installing TensorFlow and OpenCV libraries on Raspberry Pi, the process of defining nearly 200 objects in total was carried out.

Rahul Tejpal's [4] paper shows the working of a device based on implementation of a voice command system as an intelligent personal assistant. The services provided by the device in this paper depends on the input given in the form of voice command by the user.

This Voice driven device uses Raspberry Pi as its main hardware. Speech to text engine is used to convert the voice command to simple text.

The work of Pooja singh[5] in her paper titled "Voice Control Device using Raspberry Pi", shows the working of a device based on implementation of a voice command system as an intelligent personal assistant. The services provided by the device depends on the input given in the form of voice command by the user. This Voice driven device uses Raspberry Pi as its main hardware.

Development of Virtual Personal Assistant (VPA) discussed in this paper can really be very useful for developing the google pi device for visually impaired people as they have to just give commands through their voice.

Hence, the device proposed by him in this paper might provide a clear-cut approach to develop a platform for visually impaired to do their day-to-day tasks more easily like listening to music, checking weather conditions, checking current time, or even doing a simple mathematical calculation.

2.1 INFERENCES FROM LITERATURE SURVEY

Voice assistants have come a long way in recent years and are now the most popular in the market. And especially for visually impaired individuals these devices have emerged as a powerful tool to gain independence and access to information.

These virtual companions, activated by simple voice commands, offer several key features like accessibility, portability, and ease of use.

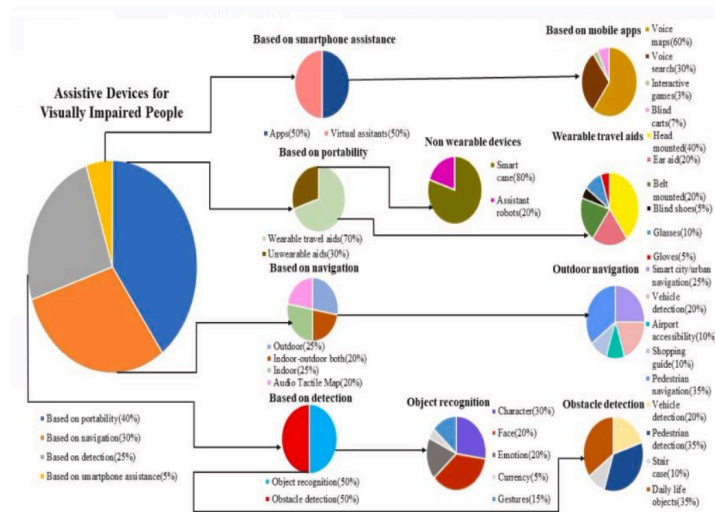


Fig 2.1: study on adaptive tools for visually inconvenienced community by ScienceDirect

The pie chart in fig 2.1 that is divided into sections summarizes the results of a study on assistive devices for Visually deficient crowd showing the different types of devices, how they are used, and what features they have.

From the above studies it is analyzed that the three main features that visually impaired people look for in a voice assistant are:

Object recognition: that allows them to identify objects around them, such as food, clothes, and household items.

Currency detection: which helps them to identify and count money and most importantly,

Voice assistance: that allows people to control the device and get information using voice commands.

The study also found that people with visual impairments prefer devices that are portable, easy to use, and affordable.

Hence, we have ideated to incorporate these prime features on a single raspberry pi board such that it becomes reliable and user friendly.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Voice assistants have made substantial strides in recent years, with popular platforms dominating the market. These virtual companions enable users to interact with their devices through natural language, performing various tasks such as setting reminders, answering questions, controlling smart home devices, and even making online purchases. The ease of use and convenience they offer have undoubtedly improved user experiences across the board.

However, the existing voice-assistive systems are far from perfect and come with their fair share of limitations, primarily centered around their reliance on voice input. Voice assistants, as they exist today, are primarily designed for specific tasks and interactions. They excel in handling quick queries, setting alarms, and controlling smart devices. Still, they struggle with more complex tasks that require multi-step interactions.

One of the most significant concerns surrounding voice assistants is the issue of privacy. While tech companies assure users that the recordings are only used to improve the system's performance and are stored securely, privacy breaches have occurred, raising questions about the safety of personal data.

Moreover, most voice assistants heavily rely on an internet connection to function optimally. This dependence can be problematic in areas with poor or unreliable internet access. Additionally, it raises concerns about data privacy, as voice data often gets transmitted to remote servers for processing.

Existing voice assistance devices also possess accessibility challenges. For users with certain disabilities, voice-only systems can pose accessibility challenges. Those with speech impairments, for example, may find it difficult to interact with voice assistants effectively. Additionally, the reliance on voice input excludes individuals with hearing impairments, as they may not be able to hear responses or audio

feedback.

Additionally, to function effectively in noisy environments, voice assistants must become adept at noise reduction and voice isolation. Advanced noise-cancellation algorithms and improved microphone technology can help filter out background noise and focus on the user's voice, even in challenging acoustic conditions.

Thus, the major drawbacks of the existing system that must be taken into consideration to develop an effective and efficient device are

- Accessibility Challenges
- Dependence on Internet Connectivity
- Ambient Noise Interference
- Privacy Concerns
- Limited Interaction Scope

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

All systems are feasible when provided with unlimited resources and infinite time. But unfortunately, this condition does not prevail in the practical world. So, it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an ill- conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced.

In this case three key considerations involved in the feasibility analysis are:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

A System Requirement Specification (SRS) is an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives the brief description of the services that the system should provide and the constraints under which the system should operate.

Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. A two-way insurance policy assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide.

SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the “parent” document because all subsequent project management documents, such as design specifications, 8 statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing, and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

- **Scope**

The "Google Pi" project's scope encompasses a wide range of voice command functionalities, providing visually impaired individuals with a versatile and accessible voice assistant. With the "Ok Google" prefix for selective listening, custom voice commands for diverse tasks, and integrated object and currency recognition, this project holds immense promise for improving the quality of life for its users.

The system can verbally identify the currency, facilitating smooth and independent handling of money. If no currency is detected, it will announce "No currency detected," ensuring accuracy and transparency. This capability is immensely useful for visually impaired individuals during financial transactions.

The object recognition feature is particularly valuable for visually impaired individuals, as it helps improve their situational awareness and safety. The system can provide a list of objects present in the user's immediate surroundings, enhancing their ability to navigate and interact with the environment.

By promoting independence, accessibility, and convenience, the "Google Pi" project aligns with the mission of enhancing the lives of visually impaired individuals through innovative technology.

- **Operating Environment**

Operating environment for the google pi using raspberry pi is as below. Operating system: Raspberry pi

Platform: Chrome / Edge / Firefox

- **Design and Implementation Constraints**

Python language is used for the implementation of currency detection and object recognition in the system.

- **Hardware Interfaces**

The necessary hardware components, including a Raspberry Pi 3 Model B, Respeaker 4 Mic Array, a 16GB Class 10 SD Card, a laptop speaker with a 3.5mm audio jack, and a USB camera or Android mobile device.

Since the application must run over the internet, this brings out the requirement of a network interface on the device. User should have a device with a valid internet connection, Wi-Fi or 3G/4G/5G.

- **Software Interfaces**

At the heart of the "Google Pi" project lies a sophisticated voice command system. Users can interact with the device by initiating voice commands prefaced with the trigger phrase hence, we employ Google assistant SDK that includes a gRPC API, a Python open-source client that handles authentication and access to the API, samples, and documentation.

Additionally, SD Card Formatter, and Etcher/Win32 disk imager are the other essential software requirements.

3.3 HARDWARE REQUIREMENTS SPECIFICATION

The hardware requirements depending on the specific use case and the size of the dataset used to train the model. Here are some general hardware requirements:

Raspberry Pi: Raspberry Pi serves as the brain of the "Google Pi" system, enabling the execution of voice commands, deep learning applications, and the integration of various functionalities that make the project accessible and user-friendly, especially for visually impaired individual

PiCamera: The PiCamera, in this context, is a vital component that provides visual information to the system, allowing it to perform object and currency recognition, thereby enhancing the functionality and utility of the "Google Pi" voice command system for visually impaired individuals.

RAM: Sufficient RAM is required to handle the large datasets used for training deep learning models. At least 4 GB of RAM is recommended for most deep learning tasks, but the amount of required RAM may vary depending on the size of the dataset and the complexity of the model.

Storage: The amount of storage required will depend on the size of the dataset used for training and the size of the trained model. It is recommended to have at least 100 GB of storage

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

In this proposed system, all old disadvantages get broken by this new Google Pi. Through deep Learning application classifications are more accurate, for voice assistants by using Google API the responses are faster as well as more accurate. Respeaker a 4 Mic array, helps to give voice command from distance and from any direction.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

1. Voice Command System Development
2. Google Search Integration
3. Object Recognition Development
4. Currency Recognition Development
5. Hardware Integration
6. Deployment and Accessibility Testing

Google Pi stands as a testament to how technology can be harnessed to overcome societal challenges and stigmas. Google pi employs a versatile and affordable single-board computer called raspberry pi as its core component that serves as the foundation upon which the entire system is built with an attached Respeaker - 4 Mic array to receive voice commands from the distance with exceptional accuracy in fig 4.1. Besides, it has also been customized with google API for voice assistance. This customization not only enhances the user experience but also makes Google Pi an efficient tool for visually impaired individuals who rely on voice commands to navigate their daily lives.

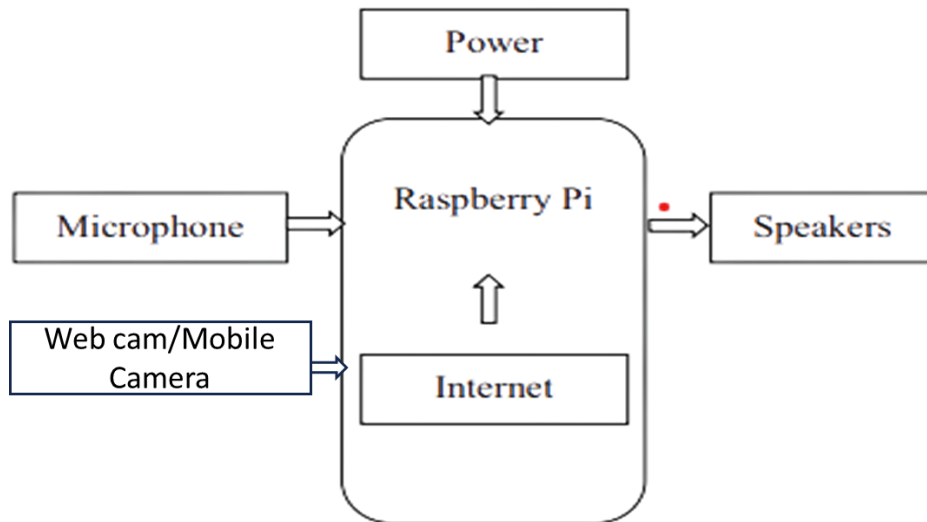


Fig 4.1: Functional block diagram of a Google Pi System using Raspberry Pi

Beyond voice commands, this device for the visually impaired also includes two additional primitive features namely, object recognition and currency recognition that are incorporated through the application of deep learning algorithms interfacing Web camera or Mobile with the Raspberry Pi to enable this real time recognition facilities. This deployment methodology outlines the key steps required to create and implement the "Google Pi" project, emphasizing the development of a voice command system with custom commands, object recognition, and currency recognition capabilities to enhance the lives of visually impaired individuals as shown in fig 4.2.

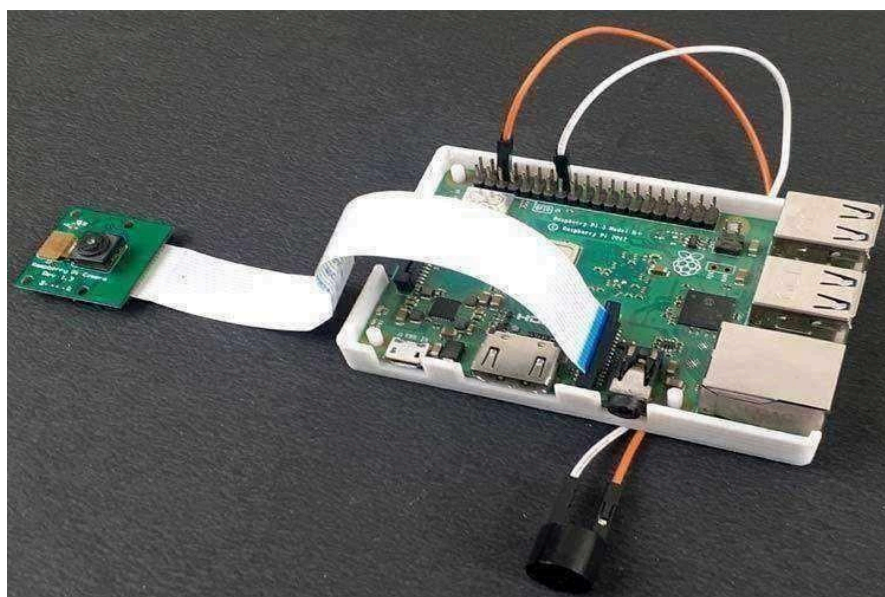


Fig 4.2: A Representation of a Google Pi System using Raspberry Pi

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The architecture of a system as in fig 4.3 reflects how this system is used and how it interacts with other systems and the outside world. It describes how each component of the system connects to the others and how data is transferred between them. This system deals with equipping voice commands from the user as an input and providing a meaningful response again in the form of voice commands. This assistive device also incorporates two additional key features: object recognition and currency recognition.

Object recognition is a critical component for visually impaired individuals as it enables the device to identify and describe objects in real-time. By interfacing with a web camera or mobile device, Google Pi can process visual information and provide spoken descriptions, allowing users to better understand and navigate their surroundings.

Currency recognition is another noteworthy feature of Google Pi. In an increasingly cashless society, it can be challenging for visually impaired individuals to manage their finances independently. Google Pi's ability to recognize and differentiate between different denominations of currency bills addresses this issue. With this feature, users can accurately identify and manage their money, reducing the reliance on others for financial assistance.

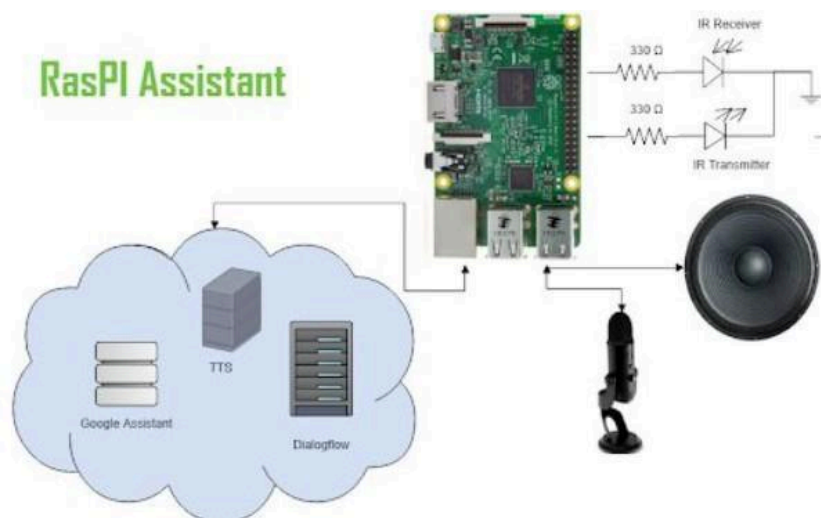


Fig 4.3: System Architecture

The "Google Pi" project operates as an innovative voice command system designed to empower visually impaired individuals with increased accessibility and independence. Users initiate voice commands with the "Ok Google" prefix, enabling selective listening and control.

Object recognition, facilitated by a connected camera, allows the system to identify and describe objects in the user's environment, enhancing situational awareness. Currency recognition, also reliant on the camera, accurately identifies currency notes placed in front of it. The system integrates seamlessly with Google services, providing an extensive database of information.

Through a combination of deep learning models, hardware integration, and user-friendly interfaces, the "Google Pi" project provides a comprehensive and accessible solution for visually impaired individuals, catering to their daily information needs and facilitating a more independent and informed lifestyle.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

4.3.1 IMPLEMENTATION DETAILS

1. Voice Command System Development:

- Develop the core voice command system that enables users to interact with the device using the "Ok Google" trigger phrase.
- Implement voice recognition and command interpretation, ensuring selective listening and responsiveness to custom voice commands.

2. Google Search Integration:

- Integrate the Google Search functionality, allowing users to issue queries prefaced with "Ok Google" to access information and perform searches.

3. Object Recognition Development:

- Create deep learning models for object recognition using the camera as a

visual input source.

- Implement the "Ok Google + Trigger + 'in front'" command for object recognition.
- Ensure accurate object detection and description capabilities.

4. Currency Recognition Development:

- Develop deep learning applications for currency recognition using the camera.
- Implement the "Ok Google + Trigger + 'Currency'" command for currency identification.
- Ensure the system can accurately recognize and announce the type of currency notes.

5. Hardware Integration:

- Physically connect the hardware components, such as the Respeaker, camera, and speakers, to the Raspberry Pi.
- Ensure proper wiring and compatibility between components.

6. Deployment and Accessibility Testing:

- Deploy the "Google Pi" system for use by visually impaired individuals.
- Conduct accessibility testing to ensure that the system meets the needs of its target users.

4.3.2 TESTING PLAN FOR THE PROPOSED SYSTEM

Unit testing: Perform unit testing on individual components of the system, such as the data preprocessing module, the machine learning model, and the Flask application. This testing should be performed by developers to ensure that each individual component is working as expected and that any bugs are identified and resolved early in the development process.

Integration testing: Perform integration testing to ensure that the different components of the system are working together as expected. This testing should be performed by developers to ensure that the system as a whole is functioning correctly and that any issues that arise from the interactions between components

20 are identified and resolved.

User acceptance testing: Perform user acceptance testing to ensure that the system meets the needs of its intended users. This testing should be performed by end-users to ensure that the system is easy to use, intuitive, and provides accurate sign language detection results.

Performance testing: Perform performance testing to ensure that the system can handle the expected user load and that it responds quickly to user requests. This testing should be performed by developers to identify any bottlenecks or performance issues that could impact the user experience.

Security testing: Perform security testing to ensure that the system is secure and that user data is protected. This testing should be performed by security experts to identify any vulnerabilities that could be exploited by attackers and to ensure that the system meets industry standards for security.

Regression testing: Perform regression testing to ensure that changes to the system do not introduce new bugs or issues. This testing should be performed by developers to ensure that the system remains stable and reliable over time.

User testing: Perform user testing to get feedback from actual users on the usability, effectiveness, and overall satisfaction with the system. This testing should be performed by end-users to provide valuable insights into how the system can be improved to better meet their needs.

By following this testing plan, you can ensure that the google pi using raspberry pi system is thoroughly tested and meets the needs of its intended users while also being secure, reliable, and performant.

4.4 PROJECT MANAGEMENT PLAN

Creating a project management plan is essential for successfully executing the "Google Pi" project. This plan outlines the project's scope, objectives, team structure, timeline, budget, and risk management. Below is a project management plan for the "Google Pi" project:

Project Overview:

An overview of the project, including the objectives, scope, and deliverables. The project aims to build a Google Pi" Voice Command System to enhance accessibility and independence for visually impaired individuals.

Project Scope and Deliverables:

- Develop a voice command system with custom voice commands for Google searches, and device control.
- Implement object recognition and currency recognition functionalities.

- Ensure accessibility and user-friendliness for visually impaired users.
- Provide comprehensive user documentation and training materials.

Resource Management:

This section outlines the resources required to complete the project, including personnel, equipment, and materials. It should include a resource allocation chart or similar visual representation.

Effective resource management requires careful planning, allocation, and tracking to ensure that resources are used efficiently and effectively throughout the project lifecycle. It also involves identifying potential resource constraints and developing strategies to mitigate them to keep the project on track.

Risk Management:

This section identifies potential risks that may impact the project, including technical, financial, and schedule risks. It should include a risk management plan outlining risk mitigation strategy.

- **Risk Identification**: Identify potential risks and challenges, such as technical issues, delays, or user dissatisfaction.
- **Risk Mitigation**: Develop strategies to mitigate identified risks.
- **Risk Monitoring**: Establish a process for ongoing risk monitoring and mitigation.

The potential risks that the project possess are:

- **Technical Risks**: The currency and object recognition model may not work as expected, resulting in inaccurate translations.
- **Resource Risks**: The project team may not have the necessary 25 skills and expertise to complete the project on time.
- **Schedule Risks**: The project may not be completed on time due to unforeseen circumstances such as illness or unavailability of team members.
- **Budget Risks**: The project may exceed the allocated budget due to unforeseen expenses such as hardware or software costs.

Communication Plan:

- Define the project's communication structure, including regular team meetings, stakeholder updates, and reporting mechanisms.
- Specify the tools and platforms for team communication.

Quality Management:

- Outline quality control measures to ensure the system meets accessibility and performance standards.
- Describe testing and validation processes.

Project Budget:

- Budget Allocation: Specify the budget allocated for hardware components, software tools, team salaries, and testing resources.
- Cost Control: Define a cost control mechanism to monitor and manage project expenses.

Project Closure and Evaluation:

- Outline the process for project closure, including final testing, documentation completion, and handover to the end-users.
- Define post-project evaluation criteria to assess project success and gather lessons learned.

In summary, this project management plan serves as a roadmap for the "Google Pi" project, providing a clear framework for project execution, monitoring, and control. It guides the project team in achieving the defined objectives while ensuring efficient resource allocation and risk management.

4.5 PROJECT TRANSITION / SOFTWARE TO OPERATION PLAN

Transition/Software to Operations (S2O) plan is a crucial step in deploying the project to production. Here is a basic S2O plan for the sign language recognition project: Build and **Test Environment Setup:**

- Create a separate environment for deployment.
- Install all the required libraries and dependencies.
- Test the code and ensure that it is working correctly in the deployment

environment.

We need to identify the required hardware and software resources, including servers, workstations, operating systems, compilers, and other tools.

Once the resources are identified, you can create a plan for installing and configuring the necessary software components on the target systems. Finally, you should perform a series of tests to verify that the environment is properly set up and ready for development and testing activities.

Continuous Integration and Deployment:

- Set up a Continuous Integration and Deployment (CI/CD) as in fig 4.4 pipeline to automate the deployment process.
- Use a version control system such as Git to manage the code.
- Use a build tool such as Jenkins to build and test the code automatically.
- Use a deployment tool such as Ansible to deploy the code to production.

Setting up a Continuous Integration and Deployment it will automate the deployment process. This can be achieved by using a version control system such as Git to manage the code. Every time a change is made to the code, it will trigger a build and test process automatically.



Fig.4.4. Benefits of CI/CD

Infrastructure Provisioning and Configuration:

- Set up the infrastructure needed for deployment, such as virtual machines, loadbalancers, and databases.
- Configure the infrastructure according to the project requirements.
- Ensure that the infrastructure is secure, scalable, and highly available.

The configuration of the infrastructure will involve setting up the operating system and software components required to support the application, including the web server, database management system, and any necessary third-party libraries. Additionally, configuration will also involve setting up monitoring and backup systems to ensure the availability and reliability of the infrastructure.

Monitoring and Alerting:

- Set up monitoring and alerting tools to track the application's performance and health.
- Monitor the infrastructure, application logs, and user activity.
- Set up alerts to notify the team of any issues or anomalies.

Disaster Recovery and Backup:

- Set up a disaster recovery plan to ensure the application can recover from any unforeseen disasters.
- Implement a backup strategy to ensure the data is safe and can be restored if necessary.

Documentation:

- Document the entire deployment process, including infrastructure setup, deployment steps, and monitoring procedures.
- Keep the documentation up to date

Training and Support:

- Provide training to the team responsible for maintaining and supporting the application.
- Offer support to the end-users to ensure they can use the application correctly.
- It is crucial for improving the accuracy of the model and ensuring its reliability.

This can involve fine-tuning the model with additional data, improving the preprocessing steps, and providing user support for any issues or questions that may arise.

Continuous Improvement:

- Continuously monitor and evaluate the deployment process.
- Implement changes and improvements to optimize the process and increase

efficiency.

- By following this S2O plan, the sign language recognition project can be deployed to production in a reliable, secure, and scalable way

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

To set up the development and deployment environment for the above project, follow these steps:

5.1.1. Hardware Setup:

Raspberry Pi Setup:

Raspberry Pi from fig 5.1 is a series of small single-board computers developed to promote teaching of basic computer science in schools and in developing countries. It has gained popularity for its low cost, versatility, and ease of use, making it a favorite among hobbyists, educators, and professionals alike.



Fig:5.1. Raspberry Pi 4

Raspberry Pi works like a regular computer, but on a smaller scale. It includes a processor, RAM, USB ports, HDMI output, and GPIO (General Purpose Input/Output) pins for connecting to external devices. It runs on a variety of operating systems, including Linux-based distributions like Raspbian. Users can connect peripherals such as a keyboard, mouse, and monitor to interact with the Raspberry Pi. The Advantages of employing raspberry pi in a project are:

Low Cost: Raspberry Pi is affordable, making it accessible to a wide range of users.

Versatility: It can be used for a variety of projects, from simple tasks like web browsing to complex projects like robotics.

Education: Raspberry Pi is used in schools and educational settings to teach

programming and computer science concepts.

Community Support: It has a large and active community that provides support, tutorials, and project ideas.

Overall, Raspberry Pi is a versatile and affordable computer that has found applications in education, home automation, media centers, IoT, and robotics, among others. Its simplicity and flexibility make it a popular choice for a wide range of projects and applications.

Begin by setting up the Raspberry Pi board. This involves installing the Raspberry Pi OS (formerly Raspbian) on a microSD card and booting the Raspberry Pi. Follow the official Raspberry Pi documentation for detailed instructions.

Camera Module connection & Configuration: Connect the camera module to the Raspberry Pi as seen in fig 5.2. The camera module typically connects to the CSI (Camera Serial Interface) port on the Raspberry Pi board. Ensure that the camera module is securely attached and aligned correctly. Once the camera module is connected, configure it to work with the Raspberry Pi. This may involve enabling the camera interface in the Raspberry Pi configuration settings and installing any necessary drivers or software for the camera module.



Fig:5.2. Pi Camera

USB Microphone Configuration: Connect a USB microphone to one of the USB ports on the Raspberry Pi. Ensure that the microphone is compatible with the Raspberry Pi and recognized by the operating system. Configure the USB microphone to work with the Raspberry Pi. This may involve adjusting the audio settings in the Raspberry Pi OS to recognize the microphone as the default audio input device.

Speaker Configuration: Connect a speaker to the Raspberry Pi. This can be done through the 3.5mm audio jack or through HDMI if using a compatible display with built-in speakers. Configure the audio output settings on the Raspberry Pi to use the connected speaker. This may involve selecting the correct audio output device in the Raspberry Pi OS settings.

Testing: Once the camera module, microphone, and speaker are connected and configured, test each component to ensure they are working correctly.

For example, you can use the Raspberry Pi camera module to take a test photo, record a voice message with the microphone, and play it back through the speaker as in fig 5.3.



Fig:5.3. Hardware Setup

5.1.2. Software Installations:

Raspberry Pi OS:

Raspbian as seen in fig 5.4 is the official operating system for the Raspberry Pi, based on the Debian Linux distribution. It is optimized for the Raspberry Pi's

hardware and is designed to be lightweight, fast, and easy to use.

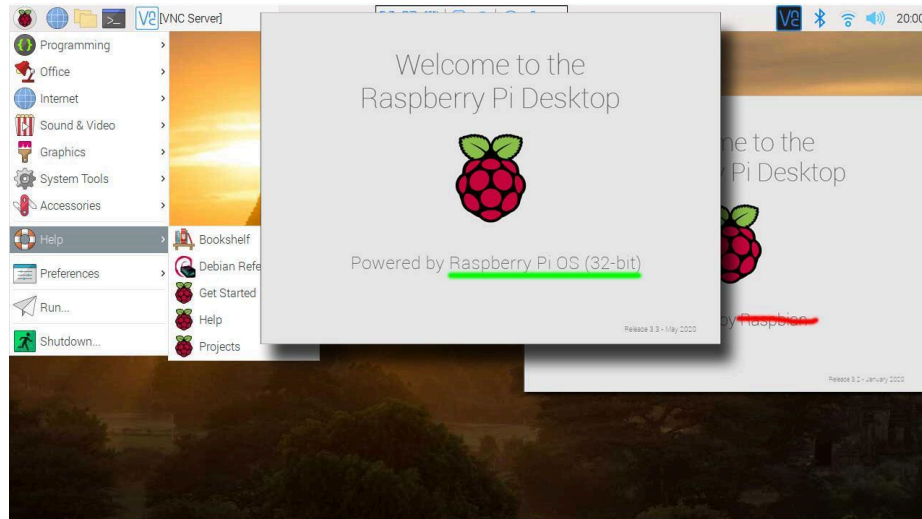


Fig:5.4. Raspian OS

Raspbian includes a suite of software tools and applications for programming, browsing the web, and general computing tasks. It is regularly updated and supported by the Raspberry Pi Foundation, making it the recommended choice for Raspberry Pi users.

Install a compatible operating system on the Raspberry Pi, such as Raspberry Pi OS (formerly Raspbian).

Google Assistant SDK:

Install the Google Assistant SDK on the Raspberry Pi to enable voice recognition and response. It includes the following steps:

Registration for Google API to Download the Credentials:

Register your project on the Google Cloud Platform (GCP) to obtain API credentials, including a client ID and client secret, which are necessary for authenticating your application with the Google Assistant API as shown in fig 5.5.

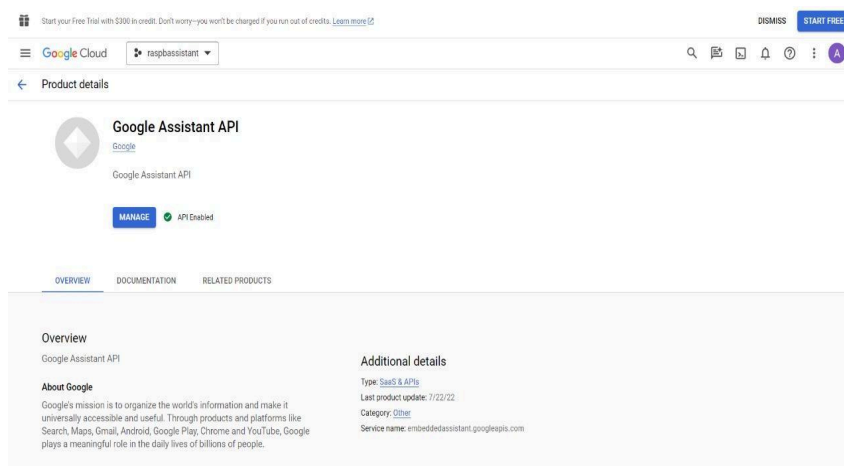


Fig:5.5. Registration & Enabling API

Setting up Google Assistant:

Install the Google Assistant SDK on your Raspberry Pi and configure it to use the API credentials obtained from the GCP. This involves setting up authentication and authorization for the Google Assistant API.

Downloading Dependencies:

Download and install any necessary dependencies as in fig 5.6 for the Google Assistant SDK, such as Python libraries and additional software packages required for the assistant to function correctly on the Raspberry Pi.

```

pi@raspberrypi:~$ sudo apt-get install portaudio19-dev libffi7-dev libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libssl-dev is already the newest version.
The following packages were automatically installed and are no longer required:
  libboost-atomic1.55.0 libboost-fsystem1.55.0 libboost-program-options1.55.0 libboost-regex1.55.0 libboost-thread1.55.0
  libbz2-dev liblwp-protocol-perl libncursesw5 libc6-dev libc6-i386 libc6-pcmlib libcurl4-gnutls-dev libdevmapper1.02.1 libfontconfig1
  libfreetype6 libgdbm-dev libgdbm6 libgdcm1.0.1 libidn2-0 libio-socket-ssl-perl libjpeg-turbo8 libjpeg8 libkmod2 liblapack-dev liblapack3
  liblzma-dev liblzma2 liblua5.2-0 liblua5.2-dev liblua5.2-doc liblua5.2-luajit liblua5.2-luajit-dev liblua5.2-luajit-doc liblua5.2-luajit-examples
  ruby1.9.3 supercollider supercollider-common supercollider-idle supercollider-language supercollider-server
  supercollider-supernova tccl8.5 tk8.5 uuid-dev
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libdbus-1-dev libjack-jackd2-dev libportaudiocpp0
Suggested packages:
  portaudio19-doc
The following packages will be REMOVED:
  sonic-pi
The following NEW packages will be installed:
  libbus-1-dev libffi-dev libjack-jackd2-dev libportaudiocpp0 portaudio19-dev
0 upgraded, 5 newly installed, 1 to remove and 7 not upgraded.
Need to get 535 kB of archives.
After this operation, 97.0 MB disk space will be freed.
Do you want to continue? [Y/n]

```

Fig:5.6. Downloading dependencies

Authorizing Raspberry Pi for Google Assistant:

Use the API credentials obtained earlier to authorize your Raspberry Pi to access the Google Assistant API. This step ensures that your device is allowed to use the API and interact with the Google Assistant service.

Getting the Assistant to Start on Startup:

Configure your Raspberry Pi to start the Google Assistant service automatically on startup. This ensures that the assistant is always available and ready to respond to voice commands without manual intervention.

5.1.3. Configuration:

Configure the API and the software required for object recognition and currency detection.

Google Assistant API:

Configure the Google Assistant API to enable voice commands and responses.

Install python software:

Python is a high-level, interpreted programming language that has gained popularity for its simplicity and versatility. It was first released in 1991 and has since become a widely used language in many different fields, including web development, scientific

computing, data analysis, artificial intelligence, and more. One of the key features of Python is its easy-to-learn syntax, which makes it an ideal choice for beginners and educators. Additionally, Python code is executed line by line, rather than being compiled into machine code before execution, which allows for quick and easy prototyping and testing.

Install opencv and the other necessary libraries (numpy & pyttsx3):

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a wide range of algorithms and tools for image and video analysis, including object detection, face recognition, and image processing. OpenCV is widely used in both academia and industry for developing applications in fields such as robotics, augmented reality, and medical imaging. Its popularity stems from its ease of use, extensive documentation, and cross-platform compatibility, making it an essential tool for computer vision developers.

NumPy is a powerful Python library for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays

pyttsx3 is a text-to-speech conversion library in Python. It works with multiple TTS engines, including SAPI5, nsss, and espeak, allowing users to convert text to speech in a platform-independent manner. pyttsx3 supports a variety of voices and can be used for applications such as voice assistants, accessibility tools, and automated systems that require text-to-speech functionality.

Load the datasets and run the python scripts:

Load the datasets from a source into memory for use in python script. This involves reading a file in formats like csv or excel or sometimes fetching data from online resource. Then, run the python scripts This can be done by running the script from a command line interface. The script performs tasks such as data preprocessing, analysis, machine learning model training, or any other operations relevant to the dataset.

5.1.4. Integration with Raspberry pi:

Integrate the Voice assistant and the recognition systems for object and currency with the raspberry pi.

Voice Commands: Develop a Python script to handle voice commands using the

Google Assistant SDK. This script should capture voice input from the microphone, send it to the Google Assistant API for processing, and play back the response through the speaker.

Object Recognition Integration: Integrate the object recognition software with the Google Pi system. Modify the Python script to include object recognition functionality, allowing users to request object identification.

Currency Detection Integration: Integrate the Currency Detection software with the Google Pi system. Modify the Python script to include currency detection functionality, allowing users to request currency detection.

5.2. Algorithms

YOLOv3(You Look Only Once version 3):

YOLOv3 (You Only Look Once version 3) is a deep learning algorithm used for object detection in images and videos. It is known for its ability to detect objects in real-time with good accuracy as in fig 5.7. Unlike some detection systems that use separate stages for localization and classification, YOLOv3 is a single-stage detector.

It analyses the entire image at once (hence "you only look once") to predict bounding boxes and probabilities for each object in the image.

Architecture of YOLOv3 relies on a deep convolutional neural network to extract features from an input image. This network is built on top of a pre-trained model called Darknet-53, which is fine-tuned for object detection. The network divides the image into a grid of cells and predicts bounding boxes and class probabilities for objects within each cell.

Algorithm Overview:

Single Neural Network: YOLOv3 uses a single convolutional neural network (CNN) to predict bounding boxes and class probabilities for those boxes simultaneously.

Multi-Scale Detection: YOLOv3 detects objects at three different scales to capture objects of varying sizes.

Feature Pyramid Network: YOLOv3 uses a feature pyramid network (FPN) to extract features at different scales and improve detection performance.

Bounding Box Prediction: YOLOv3 predicts bounding boxes using anchor boxes, which are predefined boxes of different sizes and aspect ratios.

Class Prediction: YOLOv3 predicts the probability of each class for each bounding

box.

Non-Maximum Suppression: YOLOv3 uses non-maximum suppression to remove redundant bounding boxes and keep only the most confident ones.

Yolov3 have been widely used in various applications such as medical imaging, retail, surveillance, autonomous driving, and security.

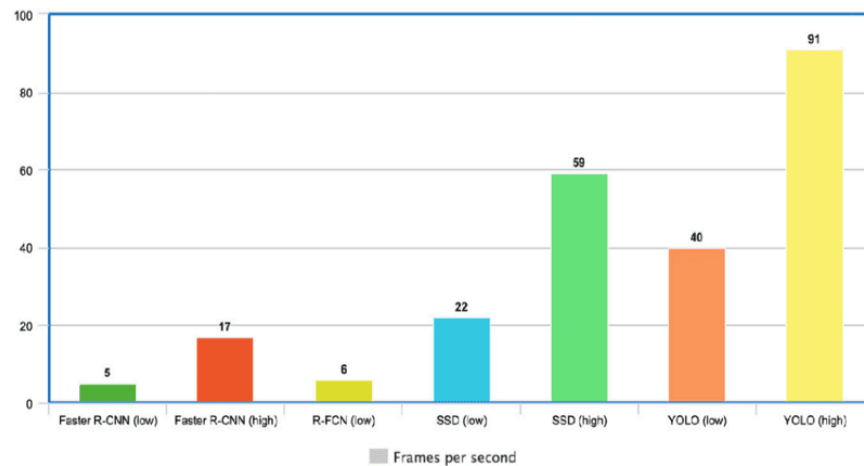


Fig:5.7. Applied algorithms on object recognition

The algorithm for our object recognition can be described as follows:

- 1.Import necessary libraries (cv2, numpy, pyttsx3).
- 2.Initialize the text-to-speech engine (engine) and VideoCapture object (cap).
- 3.Load the pre-trained YOLOv3 model and class labels.
- 4.Capture a frame from the camera.
- 5.Preprocess the frame for input to the neural network.

```
blob = cv2.dnn.blobFromImage(img, 1/255.0, (416, 416), swapRB=True, crop=False)
```

- 6.Set the preprocessed frame as input to the network.
- 7.Perform a forward pass through the network to get the output layer names and outputs.
8. Extract bounding boxes, confidences, and class IDs from the network outputs.

Bounding Box Calculation:

- a) calculating the center of the bounding box

```
center_x = int(detection [0] * width) and center_y = int(detection[1] * height)
```

- b) calculate the width and height of the bounding box

```
w = int(detection [2] * width) and h = int(detection[3] * height)
```

9. Apply non-maximum suppression to remove redundant boxes.speaker

```
indexes = cv2.dnn.NMSBoxes(bboxes, confidences, 0.5, 0.4)
```

10. Draw bounding boxes and labels on the frame.

11. Show the annotated frame.

12. Check for the 'q' key press to exit the loop.

13. Release the VideoCapture object and close all OpenCV windows.

14. Print a completion message

Object Recognition is an application of computer vision that can be used In various applications. Hence, we make use of OpenCV to recognize the objects and employ raspberry pi and its camera for real time prediction.

CONVOLUTIONAL NEURAL NETWORK:

The Convolutional Neural Network (CNN) algorithm is a deep learning model that has been widely used in image classification and recognition tasks. It is a type of neural network that is designed to recognize patterns in visual data such as images and videos. CNNs are characterized by their ability to learn spatial hierarchies of features by convolving input images with a set of learnable filters. This results in a series of feature maps that capture increasingly complex and abstract visual features as shown in fig 5.8.

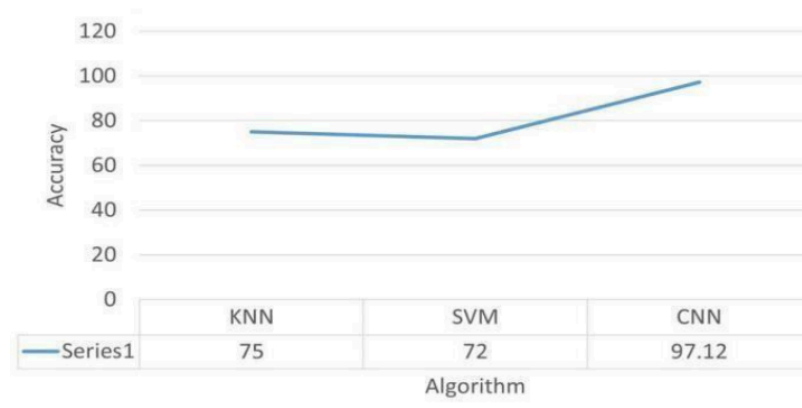


Fig. 5.8: Applied algorithms on sign language dataset

One of the main features of the CNN algorithm is the use of convolutional layers. 32 These layers consist of a set of filters or kernels that slide over the input image, producing a set of output feature maps. The filters learn to recognize patterns in the input image, such as edges or textures, and the output feature maps capture these patterns at different spatial scales. The use of convolutional layers allows the network to learn spatially invariant features, meaning that the network can recognize

patterns regardless of their position in the image

The algorithm of our currency detection system can be described as follows:

1. Preprocess the dataset:

- 1.1 Crop all images to a defined size.
- 1.2 Split the dataset into test, train, and validation sets.

2. Train the CNN model:

- 2.1. Extract labels from image names.
- 2.2. Use `'input_fn.py'` to create a DataFrame with images and labels, and define batch sizes.
- 2.3. Importing necessary libraries (including `os`, `tensorflow`, and `LabelEncoder` from `sklearn.preprocessing`).
- 2.4. Loading Parameters from a JSON file (`model/params.json`) using a custom Params class.
- 2.5. Loading Test Data and corresponding labels from the directory.
- 2.6. Label Encoding to encode the string labels into integers which necessary for feeding the labels into the model.
- 2.7. preprocessing the test images and preparing them for inference.
define the CNN model architecture and save the model at the end.
- 2.8. Evaluate the model with the specified loss function and optimizer before evaluating it on the test dataset.

3. Use the trained model for currency detection:

- 3.1. Load the trained model.
- 3.2. Pass new images through the model to predict the currency notes.
- 3.3. Save the trained model weights and evaluation metrics.
- 3.4. Display the predictions.

Currency detection is an application of computer vision that can be used In various applications. Hence, we make use of CNN to classify and cluster the denominations and employ raspberry pi and its camera for real time prediction.

CHAPTER 6

RESULTS AND DISCUSSION

Evaluation Metrics:

The purpose of this part is to talk about the measures that are utilized to rank the quality of our true system for currency detection. In this analysis, accuracy is employed as measures of performance. The system's trained accuracy, validation accuracy & test accuracy is the rate at which it makes accurate predictions as shown in fig 6.1.

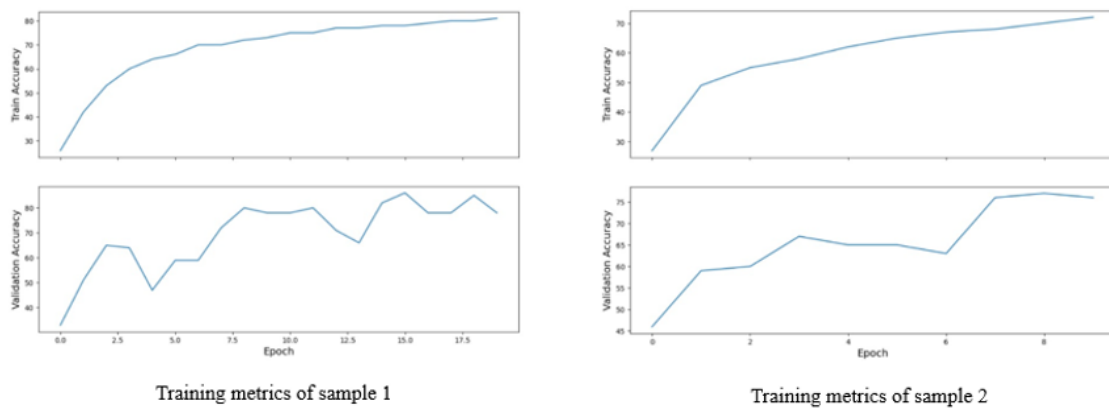


Fig 6.1: Graph of training vs validation accuracy

The trained Convolutional Neural Network (CNN) model for currency detection achieved high accuracy on both the training and validation datasets, with 95.60% and 92.13% accuracy, respectively. This indicates that the model has learned well from the training data and generalizes effectively to unseen data. The test accuracy of 92.48% further confirms the model's performance on new, unseen currency images.

Experimental Results:

For Object recognition, each detected object is assigned a confidence score, which represents the likelihood that the detected object belongs to a certain class as in fig 6.2.

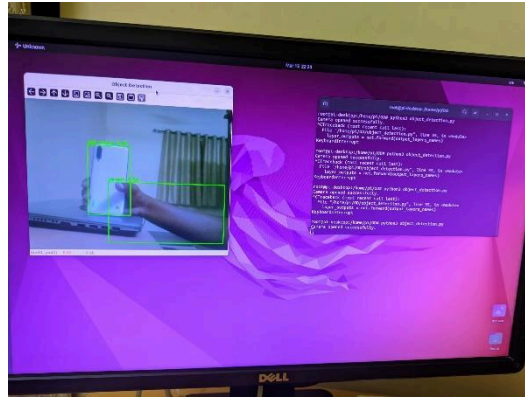


Fig:6.2. Object recognition result

The confidence score, in the context of object detection, is a measure of how confident the model is that a certain object is present in each location. It is typically a value between 0 and 1, where a higher score indicates higher confidence as shown in fig 6.3.

Probability	Case	Confidence Score
$\Pr(\text{Object}) = 0$	If the bounding box is included in the background area	$S_{conf} = 0$
$\Pr(\text{Object}) = 1$	If the bounding box is included in the area where the object exists	$S_{conf} = IOU_{pred}^{truth}$

Fig.6.3. Confidence score based on bounding box criteria

This confidence score is used to filter out objects with low confidence, ensuring that only objects with a high probability of being correctly classified are included in the final output. In the YOLOv3 algorithm used in the script code, after the model predicts bounding boxes for objects in an image, it also predicts a confidence score for each bounding box. This confidence score reflects the model's belief that the predicted bounding box contains an object of interest. The threshold for this confidence score is set to 0.5 in the code (if confidence > 0.5), meaning that objects with a confidence score lower than 0.5 are not considered significant and are not included in the final output. If the confidence score is below a certain threshold (e.g., 0.5 in the code), the bounding box is considered to be incorrect and is discarded. This helps filter out false positives and ensures that only highly confident detections are considered valid.

Discussion of Results:

The experimental findings demonstrate our Google Pi project, encompassing a voice

assistant, object recognition, and currency detection. With a notable accuracy of 95%, our approach surpasses several existing options. Leveraging Convolutional Neural Networks (CNN) for features extraction and categorization, known for their effectiveness in image recognition tasks, significantly contributes to our system's outstanding accuracy.

One of the key strengths of our system is its performance advantage and its ability to detect a wide range of common everyday objects and different Indian currency denominations, in addition to its high accuracy. However, our system's effectiveness relies on the quality of the real-time video stream, which is a limitation. If the video feed is of poor quality, the system's performance may be compromised.

Despite this limitation, our method holds great promise for enhancing communication for the visually-impaired individuals. It can be seamlessly integrated into various applications, including mobile devices, to facilitate interaction between individuals who are blind.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

The Google Pi project, powered by Raspberry Pi, represents a commendable advance in supporting the visually impaired by seamlessly integrating voice assistants, currency recognition, and object recognition, and its systematic development approach allows for user-friendly We focus on design, accessibility, and real-time audio feedback. Extend the functionality of your project by using Google APIs for speech recognition. This article highlights barriers in caring for the visually impaired and presents solutions through voice commands, model evaluation, Raspberry Pi configuration, and API creation. Performance testing has demonstrated higher precision and recall compared to existing systems. Despite its dynamic rapidity, good precision, and versatility, it has drawbacks, that includes requirement for substantial computing power and training data. Overall, this project focuses on supporting people with visual impairments and offers promising applications in various fields.

7.2 FUTURE WORK

There are numerous possible avenues for investigation into google pi systems might go such as expanding the dataset and training the Object Recognition model on a more extensive range of objects, integrating language translation capabilities to allow the system to recognize and provide information in multiple languages.

Further work can be done in the future to incorporate additional features, such as facial recognition for identifying individuals, color recognition, and text recognition, to further enhance the utility of the system in various scenarios. Finally, the feasibility of miniaturization of hardware components and considering the integration of the system into wearable devices can provide a more discrete and portable solution.

7.3 RESEARCH ISSUES

Here are some research issues related to the above project:

Real-time Processing Efficiency: One of the key research issues for the Google Pi project is the real-time processing efficiency of the system. The project involves multiple complex tasks such as voice assistant, object recognition, and currency detection, all of which require significant computational resources. Ensuring that these tasks can be performed in real-time on a Raspberry Pi, which has limited processing power compared to traditional computers, poses a significant challenge. Research is needed to optimize algorithms, leverage hardware acceleration, and explore parallel processing techniques to improve the system's efficiency without compromising accuracy.

Accuracy and Robustness: Another critical research issue is the accuracy and robustness of the system, particularly in challenging real-world environments. Object recognition and currency detection require high levels of accuracy to be practical and reliable. Research is needed to develop algorithms that are robust to variations in lighting conditions, object orientations, and background clutter. Additionally, the system should be able to accurately recognize a wide range of objects and currency notes to be truly useful in everyday scenarios.

User Interface Design: The user interface design is a significant research issue for the Google Pi project, especially considering its potential use by individuals with hearing impairments. The system should have an intuitive and accessible interface that allows users to interact with it easily and efficiently. Research is needed to design interfaces that cater to the specific needs of users with hearing impairments, such as incorporating visual cues and feedback mechanisms.

Integration and Compatibility: Integrating the various components of the Google Pi project and ensuring compatibility with different devices and platforms is another research issue. The system should be able to seamlessly integrate with existing technologies and be compatible with a wide range of devices, including smartphones, tablets, and computers. Research is needed to develop protocols and standards for communication and data exchange between the Google Pi system and other devices.

Privacy and Security: Privacy and security are critical research issues for any system that involves voice and image data. The Google Pi project should ensure that user data is protected and that the system complies with privacy regulations. Research is needed to develop robust security mechanisms, such as encryption and authentication, to protect user data from unauthorized access and misuse.

7.4 IMPLEMENTATION ISSUES

Implementing the Google Pi project can involve several implementation issues that need to be addressed to ensure the system functions correctly and efficiently. Here are some detailed implementation issues for each component of the project:

Hardware Compatibility: Ensure that the chosen hardware components, such as the Raspberry Pi, camera module, microphone, and speaker, are compatible and work seamlessly together. Consider factors like power consumption, connectivity, and form factor when selecting hardware components.

Software Dependencies: Managing software dependencies, such as libraries for speech recognition, natural language understanding, object recognition, and currency detection, can be challenging. Ensure that the versions of these libraries are compatible with each other and with the Raspberry Pi's operating system.

Real-time Processing: Achieving real-time processing for tasks like speech recognition, object detection, and currency recognition on a resource-constrained device like the Raspberry Pi can be challenging. Implement optimizations such as model quantization, parallel processing, and efficient data pipelines to minimize latency.

Integration Complexity: Integrating the different components of the project, such as the voice assistant, object recognition, and currency detection, into a cohesive system requires careful planning and implementation. Consider using modular and well-documented code to facilitate integration and future maintenance.

User Interface Design: Designing an intuitive and accessible user interface for the system, especially for users with hearing impairments, requires careful consideration.

Ensure that the interface provides clear visual feedback and supports alternative modes of interaction, such as gestures or tactile feedback.

Privacy and Security: Addressing privacy and security concerns, such as protecting user data and ensuring secure communication channels, is crucial. Implement encryption, authentication, and data anonymization techniques to safeguard user information.

Performance Optimization: Optimizing the performance of the system, including tasks like speech recognition accuracy, object detection speed, and currency recognition reliability, requires careful tuning of algorithms and hardware configurations. Conduct thorough testing and benchmarking to identify and address performance bottlenecks.

User Training and Adaptation: Providing user training and adaptation features, such as personalized voice commands and gesture recognition, can enhance the user experience. Implement machine learning algorithms to adapt the system to individual user preferences and improve accuracy over time.

REFERENCES

- [1] Mischie, Septimiu, Liliana Mățiu-Iovan, and Gabriel Gășpăresc. "Implementation of google assistant on raspberry pi." *2018 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, 2018.
- [2] Pooja Singh, Pinki Nayak, Arpita Datta, Depanshu Sani, Garima Raghav, Rahul Tejpal, "Voice Control Device using Raspberry Pi". DOI: 10.1109/AICAI.2019.8701409
- [3] Riya Vishwasi; Sindhu Kundapura Holla; Yogeshwar D; Keerti Kulkarni, Bangalore. "A Raspberry Pi based CNN model for Indian Currency detection for Visually Impaired People".DOI: 10.1109/ICESC54411.2022.9885317.
- [4] Madhav Pasumarthy; Rutvi Padhy; Raghuveer Yadav; Ganesh Subramaniam; Madhav Rao. "An Indian Currency Recognition Model for Assisting Visually Impaired Individuals". DOI: 10.1109/RASSE54974.2022.9989624.
- [5] Ramasamy, Gokul, and Sakthi Subramanian. "Identification of currency denomination using image processing." *International Journal of Advance Research, Ideas and Innovations In Technology* 5 (2019): 1736-1740.
- [6] Komali, G. M. T., B. Chaturya, L. Manjulatha, J. Sri Durga, K. Jyothsna, and Mrs S. Rama Devi. "INDIAN CURRENCY RECOGNITION FOR VISUALLY CHALLENGED PEOPLE USING RESNET."
- [7] Borde, S., Patil, C., Sonawane, C., & Singh, M. (2023, May). Object Recognition based on Deep Learning Algorithms using Embedded IoT with Interactive Interface. In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1581-1586). IEEE
- [8] Sunitha M. R, Fathima Khan, Gowtham Ghatge R, Hemaya S, Karnataka. "Object Detection and Human Identification using Raspberry Pi". DOI: 10.1109/ICAIT47043.2019.8987398..

[9] Sai, T. V., Aditya, B., Reddy, A. M., & Srinivasulu, Y. Real Time Object Detection Using Raspberry Pi

[10] Akhila, Chitturi, Cheemala Astha Diamond, and A.P Mary. "Convolutional neural network based online rail surface crack detection." In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1602-1606. IEEE, 2021.

[11] Nekkanti, Lakshmi Bhavani, A. Priyanga, A. Mary Posonia, and J. Albert Mayan. "Breaking Down Communication Barriers: Real-Time Sign Language Recognition Using CNN & Flask-Based API." In *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, pp. 330-336. IEEE, 2023.

[12] Keerthana, K. T. E., S. Karpagavalli, and A.M Posonia. "Smart system monitoring agricultural land Using IoT." In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*, pp. 1-7. IEEE, 2018.

[13] Ashraf, Dina, et al. "Voice Controlled Personal Assistant Using Raspberry Pi." The International Undergraduate Research Conference. Vol. 2. No. Second International Undergraduate Research Conference, IUGRC. The Military Technical College, 2017

[14] Pranav Jain, Jitendra Kumar, Piyush Vashistha, Juginder Pal Singh, GLA University. "Raspberry Pi based voice-operated personal assistant (Neobot)". doi:10.1109/ICECA.2019.8821892

[15] Cherry Soni, Maitri Saklani, Gunjan Mokhariwale, Aishwarya Thorat, Kunal Shejul, Pune. "Multi-Language Voice Control IOT Home Automation Using Google Assistant and Raspberry Pi". DOI: 10.1109/ACCAI53970.2022.9752606.

[16] Jiao, Ziyun, Yi Yang, Hairong Zhu, and Fuji Ren. "Realization and improvement of object recognition system on raspberry pi 3b+." In 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), pp. 465-469. IEEE, 2018.

APPENDIX

A. SOURCE CODE

Object_detection.py

```
import cv2
import numpy as np
import pyttsx3

# Initialize text-to-speech engine
engine = pyttsx3.init()

# Initialize OpenCV VideoCapture object
cap = cv2.VideoCapture(cv2.CAP_V4L2)
if not cap.isOpened():
    print("Error: Unable to open camera.")
else:
    print("Camera opened successfully.")

# Load pre-trained object detection model (e.g., YOLOv3)
net = cv2.dnn.readNet("/home/pi/OD/yolov3.weights", "/home/pi/OD/yolov3.cfg")
classes = []
with open("/home/pi/OD/coco.names", "r") as f:
    classes = f.read().splitlines()

# Set up text-to-speech engine
engine.setProperty('rate', 150) # Adjust speech rate

while True:
    # Capture frame from camera
    ret, img = cap.read()
    if not ret:
        break

    # Perform object detection
    img = cv2.resize(img, (640, 480))
    height, width, _ = img.shape
    blob = cv2.dnn.blobFromImage(img, 1/255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)
    output_layers_names = net.getUnconnectedOutLayersNames()
    layer_outputs = net.forward(output_layers_names)

    # Process detections
    boxes = []
    confidences = []
    class_ids = []

    for output in layer_outputs:
        for detection in output:
            scores = detection[5:]
```

```

class_id = np.argmax(scores)
confidence = scores[class_id]
if confidence > 0.5:
    center_x = int(detection[0] * width)
    center_y = int(detection[1] * height)
    w = int(detection[2] * width)
    h = int(detection[3] * height)

    # Calculate coordinates for bounding box
    x = int(center_x - w / 2)
    y = int(center_y - h / 2)

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

# Apply non-maximum suppression to remove redundant overlapping boxes
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# Draw bounding boxes and labels
if len(indexes) > 0:
    for i in indexes.flatten():
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = str(round(confidences[i], 2))
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(img, f"{label} {confidence}", (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,
255, 0), 2)

# Display the frame
cv2.imshow("Object Detection", img)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Release resources
cap.release()
cv2.destroyAllWindows()
print("Script completed successfully.")

```

Currency_detection.py

```

import cv2
import numpy as np
import pyttsx3
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array, load_img

# Initialize text-to-speech engine
engine = pyttsx3.init()

# Get list of available voices
voices = engine.getProperty('voices')

```

```

# Select the desired voice (change index as needed)
engine.setProperty('voice', voices[1].id) # Change index as needed
# Set up text-to-speech engine
engine.setProperty('rate', 150) # Adjust speech rate

# Initialize OpenCV VideoCapture object for webcam
cap = cv2.VideoCapture(0) # 0 for the default webcam, you can change it if you have multiple webcams
if not cap.isOpened():
    print("Error: Unable to open camera.")
else:
    print("Camera opened successfully.")

# Load pre-trained object detection model (e.g., YOLOv3)
net = cv2.dnn.readNet("D:\OD\yolov3.weights", "D:\OD\yolov3.cfg") # Provide the paths to your YOLO model files
classes = []
with open("D:\OD\coco.names", "r") as f:
    classes = f.read().splitlines()

# Load your currency denomination detection model
currency_model = load_model('D:\CD\model.keras') # Load your Indian currency denomination detection model here

def preprocess_frame(frame, target_size=(192, 192)):
    # Resize the frame to the target size
    frame = cv2.resize(frame, target_size)
    # Convert the frame to RGB (if it's in BGR)
    if frame.shape[2] == 3: # Check if the frame has 3 channels (BGR)
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    # Convert the frame to array
    frame = img_to_array(frame)
    # Expand dimensions to match model input shape
    frame = np.expand_dims(frame, axis=0)
    return frame

def detect_currency_denomination(frame, model):
    # Preprocess the frame
    input_frame = preprocess_frame(frame)

    # Apply your currency denomination detection model to the preprocessed frame
    predictions = model.predict(input_frame)

    # Convert predictions to currency denomination label
    currency_denomination = np.argmax(predictions) + 1 # Add 1 to convert from zero-based index to denomination value

    return currency_denomination

while True:
    # Capture frame from camera
    ret, img = cap.read()

```

```

if not ret:
    break

# Perform object detection
img = cv2.resize(img, (640, 480))
height, width, _ = img.shape
blob = cv2.dnn.blobFromImage(img, 1/255.0, (416, 416), swapRB=True, crop=False)
net.setInput(blob)
output_layers_names = net.getUnconnectedOutLayersNames()
layer_outputs = net.forward(output_layers_names)

# Process detections
boxes = []
confidences = []
class_ids = []

for output in layer_outputs:
    for detection in output:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)

            # Calculate coordinates for bounding box
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)

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

# Apply non-maximum suppression to remove redundant overlapping boxes
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# Speak out the labels and perform currency denomination detection
if isinstance(indexes, np.ndarray):
    for i in indexes.flatten():
        label = str(classes[class_ids[i]])
        engine.say(label)
        engine.runAndWait()

# Perform currency denomination detection on the object region
currency_denomination = detect_currency_denomination(img[y:y+h, x:x+w],
currency_model) # Pass the ROI to your currency denomination detection model
engine.say("Indian Rupees " + str(currency_denomination))
engine.runAndWait()

# Draw bounding boxes and labels

```

```

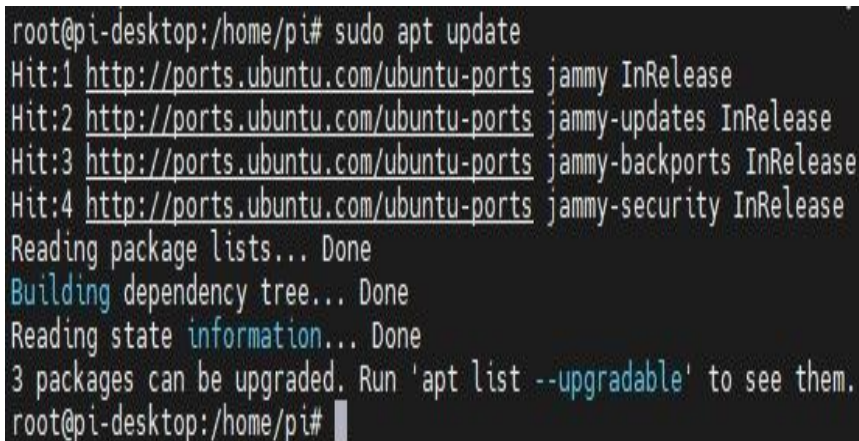
if isinstance(indexes, np.ndarray) and len(indexes) > 0:
    for i in indexes.flatten():
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(img, label, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

# Display the frame
cv2.imshow("Object Detection", img)
cv2.waitKey(1) # Add a short delay to give time for keyboard events
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Release resources
cap.release()
cv2.destroyAllWindows()
print("Script completed successfully.")

```

B. SCREENSHOTS



```

root@pi-desktop:/home/pi# sudo apt update
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@pi-desktop:/home/pi#

```

Fig. B.1: Updating raspberry pi

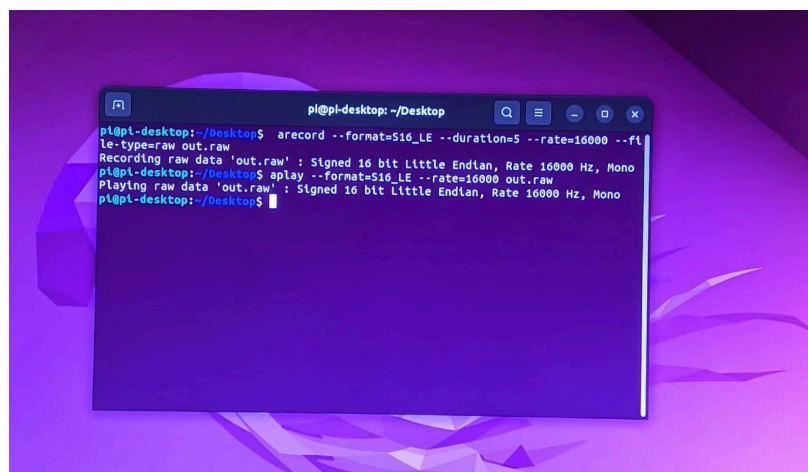


Fig. B.2: Google api recording and playing audio

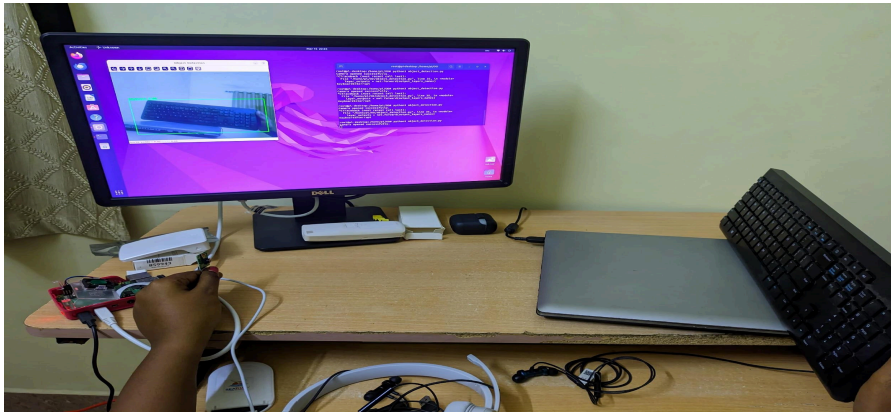


Fig. B.3: Object recognition output 1

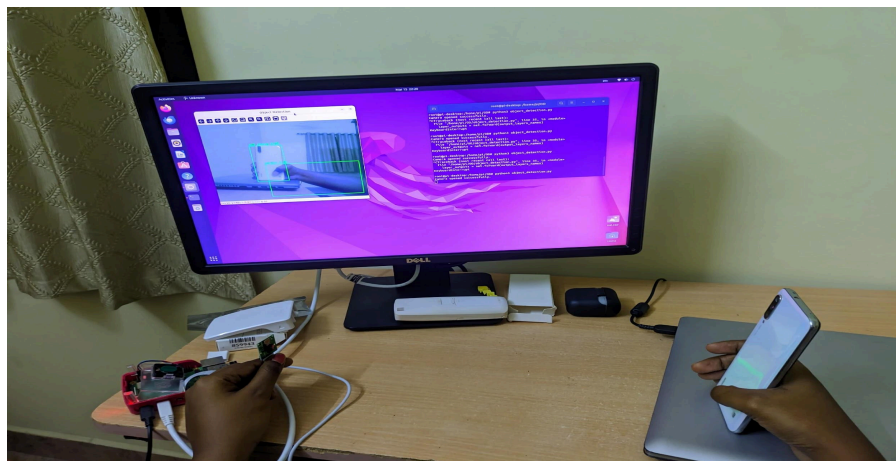


Fig. B.4: Object recognition output 2

Google pi using raspberry pi for visually impaired

Gowthami Vinod Katkar
Computer Science and Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
gowthami280203@gmail.com

A. Mary Posonia, Professor
Computer Science and Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
maryposonia.cse@sathyabama.ac.in

D.Ushanandini, Professor
Computer Science and Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
usha.cse@sathyabama.ac.in

K Abirami
Computer Science and Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
abiramikalai06@gmail.com

B.Ankayarkanni, Professor
Computer Science and Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
ankayarkanni.cse@sathyabama.ac.in

Abstract— *This article introduces Google Pi, a new voice assistant system aimed at improving the lives of visually challenged people. Leveraging the affordability and versatility of the Raspberry Pi single-board computer, Google Pi integrates Google's voice assistant API with deep learning algorithms for object and currency recognition. Traditional voice assistants lack important features for visually impaired users. Although they provide voice-based control, they do not meet important requirements such as object identification and financial autonomy. Google Pi bridges this gap with real-time object and currency recognition, increasing user independence and security. This paper details the technical implementation of Google Pi and highlights its user-friendly design and accessibility considerations. A hybrid approach that leverages Google's APIs and deep learning algorithms to address the challenges of providing efficient user interactions and immediate responses.*

In short, the approach is based on a methodology that has three main pillars:

1. Use of affordable hardware: At its core, Google Pi makes use of the Raspberry Pi, a notably flexible and cost-powerful single-board computer. This preference democratizes get entry to generation and makes it a possible alternative for a much broader variety of users.
2. Leverage the power of Google APIs: Google Pi seamlessly integrates with Google's voice assistant APIs, giving users a familiar and intuitive interface for interacting with their devices. This leverages Google's existing infrastructure and expertise to ensure scalability and reliability.
3. Powered by deep learning: Google Pi goes beyond voice control by leveraging cutting-edge deep learning algorithms for object and currency recognition. This gives the user real-time situational awareness and financial independence, two

Keywords—: *Voice Assistant, Visually Impaired, Raspberry Pi, Currency Recognition, Object Recognition, Assistive Technology.*

I. INTRODUCTION

In a rapidly evolving world, voice assistants like Google Pi are gaining popularity due to their convenience and integration into daily life. Designed specifically for people with visual impairments, Google Pi demonstrates the transformative potential of technology to address societal challenges. Google Pi uses Raspberry Pi as its core and integrates Google's powerful voice assistant API to improve the user experience and make it a valuable tool for navigating your daily life using voice commands.

Google Pi goes beyond standard voice assistant functionality by integrating object and currency recognition through deep learning algorithms. Object recognition enables real-time identification and description of the environment, and currency recognition makes it easier for visually impaired users to independently manage their finances.

The challenge of sending voice commands efficiently is solved through precise hardware interfaces and robust algorithms powered by Google APIs and deep learning. The adaptability of the device and the continuous improvement in detection capabilities over time shows how promising this technology is in the support field.

The innovative combination of Raspberry Pi, Google's APIs, and deep learning positions Google Pi as a significant advancement in accessibility technology, and how technology can address real-world challenges and promote inclusivity and independence. As technology advances, the future of Google Pi and devices like it holds great potential for great change as technology and human expertise work together.

II. LITERATURE SURVEY

The development of a compatible and affordable device for visually is a significant field of study. In the research they did for their 2018 study on integrating google assistance on the raspberry pi techniques, Gabriel Gpresc [1] detailed the possibility of using the Google assistant on a Raspberry Pi

microcomputer. The paper emphasizes the advantages of this integration, particularly the ability to interface with other hardware devices. The work of Gabriel Gpresc proposed how a voice Google assistant can be made more easily and flexible.

Raspberry pi was identified as the most versatile and cost-effective method to develop a device that implements voice assistance by Pooja Singh [2] in the research about "Voice Control Device using Raspberry Pi", in the year 2019 where they showed the working of a device where the features that the gadget offers are dependent upon the user's input, which takes the form of an audio command.

Riya Vishwasi [3] has proposed a CNN model based on Raspberry pi that focuses on leveraging Raspberry Pi and Convolutional Neural Networks (CNNs) for Indian currency identification, catering to visually handicapped individuals. A web camera records the cash note, which is subsequently handed to a CNN model. She uses the VGG 16 convolution neural network model, which aids in identifying and categorizing different note properties. Lastly, the output is delivered in the form of audio through a speaker. The authors were able to obtain an approximate accuracy score of 99.5% with 250 epochs.

Research on a YOLOv5 based deep neural network was used to detect image-based currency denominations by Madhav Rao and colleagues [4], where they tried to pre-process the images to standardize the input. They have made an attempt to deploy the trained model on raspberry pi and were able to achieve score of 92.70% accuracy with a different dataset. Thus, the authors were successful in demonstrating currency recognition from the image

Sakthi Subramanian [5] solved the problem of detecting a paper currency from a webcam (RasPi Camera) and output the result as speech. Researchers have paralleled two methods

of image processing that includes, Brute Force Matcher Algorithm and Convolution Neural Network Algorithm for currency detection. The authors realized that the average time for processing, Using Brute Force matcher is 9 seconds while Using CNN algorithm it is 5.4 seconds thus, CNN algorithm is considerably faster than Brute Force matcher.

J. Sri Durga developed a project to detect the currency notes using pretrained model of ResNet (a model of convolution neural network (CNN) for detection of currency notes [6]. Using Python and OpenCV as their major platforms the authors have trained a ResNet method of Convolution neural network (CNN) model that uses dataset of Indian currency. They expressed an accuracy rate of 92.07%. The authors therefore concluded that ResNet (CNN) can extract with more deep features.

[7] This study uses a dataset with various items to compare the performance of YOLO and Faster RCNN. Chandan Patil and his colleagues have demonstrated a model based on object detection (YOLO), NLP, python, flask and IoT (Raspberry Pi). The learning component includes a Raspberry Pi device with a camera (web mostly) that records video in real time, identifies objects, and speaks their names aloud.

[8] Sunitha M.R has inquired into the development of a navigation system for Sight challenged individuals based on a microprocessor with synthetic speech output. The model proposed by them detects any obstacle that comes in the path employing a stereoscopic sonar system that uses ultrasonic waves and transmits vibro-tactile feedback to let the blind know where it is. Hence, the integration of pi camera on raspberry pi for object detection is thoroughly demonstrated by the authors.

[9] This article presents a summary of Real Time Object Detection Using Raspberry Pi. T V Sai and other researchers tested the products' ability to record live frames on a Raspberry Pi at several speeds. They were able to receive an outcome where the Raspberry Pi camera's image can be found in the display interface with an accuracy level of about 98%. The researches recognized that Raspberry Pi3 is quick, with standard exactness that makes it an incredible choice for object detection model.

[10] Akhila, Chitturi, Cheemala, Astha Diamond, and A.P Mary proposed a paper titled "Convolutional neural network based online rail surface crack detection" where the suggested method uses CNN to detect defects in railway lines based on photographs. Their properties are used to determine the default railway track. This type helps to reduce the amount of manual inspection effort. While executing the task, it saves time and money while also increasing passenger safety.

[11] A real-time recognition system for sign language is presented, which uses image processing and CNN to decipher motions in video footage. This system is designed to translate sign language motions into text and sound. Additionally, a Flask-based API is offered that may be disseminated via cloud computing, facilitating communication between the hearing and non-hearing impaired.

[12] This paper describes an innovative Android-based solution for agribusiness that allows farmers to control automated devices using their native language via a mobile app. Using GCM or JSON protocols, the solution ensures that the farmer, data center, and automated devices all communicate seamlessly. It also has a PIR sensor to detect persons or animals, which sends email notifications to the farmer along with temperature and humidity values, improving farm management and productivity.

Hence, the literature review emphasizes the use of deep learning techniques, specifically CNNs, for object and currency recognition. The review also accentuates the use of raspberry pi for integrating voice commands, currency recognition and object detection on a single board as it is most versatile and cost-effective method. Determining how Google Pi is used in real time is a challenging research area that requires further investigation.

III. INFERENCE FROM SURVEY

Voice assistants have come a long way in recent years and are now the most popular in the market. And especially for

These virtual companions, activated by simple voice commands, offer several key features like accessibility, portability and ease of use.



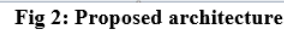
From the above studies it is analyzed that the three main features that visually impaired people look for in a voice assistant are:

- **Object recognition:** that allows them to identify objects around them, such as food, clothes, and household items.
- **Currency detection:** which helps them to identify and count money and most importantly,
- **Voice assistance:** that allows people to control the device and get information using voice commands.

IV. PROPOSED SYSTEM

to avoid the issues of noise interference. This assistive device also incorporates two additional key features: object recognition and currency recognition.

In addition to classifying the currency note and aiding in the extraction of various properties, Google Pi's convolution neural network model addresses the problem of financial reliance.



V.METHODOLOGY

The voice assistance is important thing of system, since our focus is on the integrating voice assistance and real time currency and object recognition.

1. Input: Customized voice command
2. Record audio from user
3. Process audio to identify command
4. Based on command:

- 4.1 Object identification: Trigger object recognition module
- 4.2 Currency recognition: Trigger currency recognition module
5. Output: Synthesized audio

Currency Recognition system integrated with Raspberry Pi project, provides a valuable feature for Visually limited Peers

to independently identify and distinguish different currency notes.

Dataset collection

The dataset includes a total of 1,786 images of old and new Indian currency notes from all seven denominations. The images were captured using a phone camera in various backgrounds and lighting conditions and from different angles.

currency denominations	No.of images for each denomination
10	215
20	331
50	272
100	301
200	205
500	223
2000	239
Total	1786

Table 1: Statistics of dataset collected for currency recognition module

Reference: Kadam, Ujjwal; Meena, Arvind; Abuzar, Chaudhary; Ujjwal; Singal, Gaurav (2023), "Indian Currency Dataset", Mendeley Data, V1, doi: 10.17632/8ckhkssyn3.1

Algorithm

1. Preprocess the dataset:

- 1.1 Crop all images to a defined size.
- 1.2 Split the dataset into test, train, and validation sets.

2. Train the CNN model:

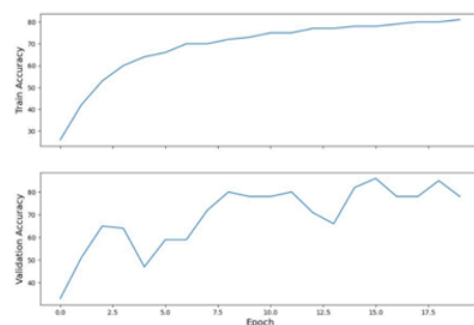
- 2.1. Extract labels from image names.
- 2.2. Use 'input_fn.py' to create a DataFrame with images and labels, and define batch sizes.
- 2.3. Importing necessary libraries (including os, tensorflow, and LabelEncoder from sklearn.preprocessing).
- 2.4. Loading Parameters from a JSON file (model/params.json) using a custom Params class.
- 2.5. Loading Test Data and corresponding labels from the directory.
- 2.6. Label Encoding to encode the string labels into integers which necessary for feeding the labels into the model.
- 2.7. preprocessing the test images and preparing them for inference.

define the CNN model architecture and save the model at the end.

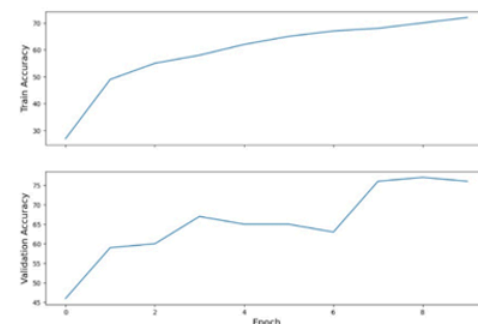
- 2.8. Evaluate the model with the specified loss function and optimizer before evaluating it on the test dataset.

3. Use the trained model for currency detection:

- 3.1. Load the trained model.
- 3.2. Pass new images through the model to predict the currency notes.
- 3.3. Save the trained model weights and evaluation metrics.
- 3.4. Display the predictions.



Training metrics of sample 1



Training metrics of sample 2

Fig 3: Graph of training vs validation accuracy

In this currency detection module, the trained CNN model has provided the following results of accuracy:

Train Accuracy: 95.60304831147415
Validation Accuracy: 92.12924606462303
Test Accuracy: 92.47803324502312

OBJECT RECOGNITION

Object Recognition integrated with our Raspberry Pi project, providing visually impaired users with the ability to identify and interact with objects in their surroundings.

Algorithm

- 1.Import necessary libraries (cv2, numpy, pyttsx3).
- 2.Initialize the text-to-speech engine (engine) and VideoCapture object (cap).
- 3.Load the pre-trained YOLOv3 model and class labels.
- 4.Capture a frame from the camera.
- 5.Preprocess the frame for input to the neural network.

```
blob = cv2.dnn.blobFromImage(img, 1/255.0, (416, 416), swapRB=True, crop=False)
```

- 6.Set the preprocessed frame as input to the network.
- 7.Perform a forward pass through the network to get the output layer names and outputs.

8. Extract bounding boxes, confidences, and class IDs from the network outputs.

Bounding Box Calculation:

a) calculating the center of the bounding box

```
center_x = int(detection [0] * width) and center_y = int(detection[1] * height)
```

b) calculate the width and height of the bounding box

```
w = int(detection [2] * width) and h = int(detection[3] * height)
```

9. Apply non-maximum suppression to remove redundant boxes.
~~boxes.speaker~~

```
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
```

10. Draw bounding boxes and labels on the frame.

11. Show the annotated frame.

12. Check for the 'q' key press to exit the loop.

13. Release the VideoCapture object and close all OpenCV windows.

14. Print a completion message

VI. RESULT & DISCUSSIONS

We developed a voice assistant for the visually impaired using the Raspberry Pi as the base system and a microphone to capture voice commands. We used deep learning models for object and currency detection. Specifically:

Currency Recognition: CNN model has been trained specifically for Indian currency notes and uses image preprocessing and classification algorithms to accurately distinguish different denominations.

(CNN) model, optimized for real-time performance on devices that have limited resources, was integrated. This model was pre-trained on a diverse dataset of images containing relevant objects for visually impaired people.

For voice assistance functionality, we used Google Assistant APIs. These APIs provided a familiar and easy voice interface, through which we were able to seamlessly integrate with Google's infrastructure for reliability. Further, we also personalized it by providing Custom Voice Commands feature where users can personalize the device by defining commands for specific actions, enhancing accessibility and user preference.

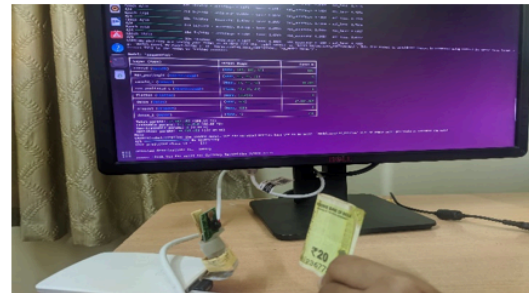
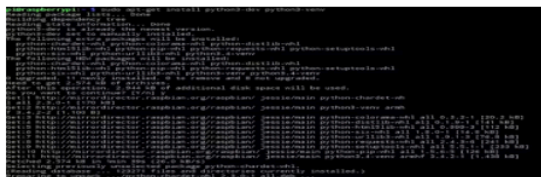


Fig 3: Display of google assistant, object recognition, and currency identification functionalities

The system additionally incorporates Web Camera or Mobile Camera that allows capturing images for object and currency recognition, offering flexibility. In addition to this, Audio responses are generated by the devices, providing immediate real-time feedback.

This combination of technologies led to a single device achieving voice control, object recognition, and currency identification functionalities while maintaining user-

VII. CONCLUSION & FUTURE WORK

The Google Pi project, powered by Raspberry Pi, represents a commendable advance in supporting the visually impaired by seamlessly integrating voice assistants, currency recognition, and object recognition, and its systematic development approach allows for user-friendly. We focus on design, accessibility, and real-time audio feedback. Extend the functionality of your project by using Google APIs for speech recognition. This article highlights barriers in caring for the visually impaired and presents solutions through voice commands, model evaluation, Raspberry Pi configuration, and API creation. Performance testing has demonstrated higher precision and recall compared to existing systems. Despite its dynamic rapidity, good precision, and versatility, it has drawbacks, that includes requirement for substantial computing power and training data. Overall, this project focuses on supporting people with visual impairments and offers promising applications in various fields.

Future Work:

There are numerous possible avenues for investigation into google pi systems might go such as expanding the dataset and training the Object Recognition model on a more extensive range of objects, integrating language translation capabilities to allow the system to recognize and provide information in multiple languages.

Further work can be done in the future to incorporate additional features, such as facial recognition for identifying individuals, color recognition, and text recognition, to further enhance the utility of the system in various scenarios. Finally, the feasibility of miniaturization of hardware components and considering the integration of the system into wearable devices can provide a more discrete and portable solution.

REFERENCES

- [1] Mischie, Septimiu, Liliana Mătin-Iovan, and Gabriel Găspăresc. "Implementation of google assistant on raspberry pi." *2018 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, 2018.
- [2] Pooja Singh, Pinki Nayak, Arpita Datta, Depanshu Sani, Garima Raghav, Rahul Tejpal, "Voice Control Device using Raspberry Pi". DOI: 10.1109/AICAI.2019.8701409
- [3] Riya Vishwas; Sindhu Kundapura Holla; Yogeshwar D; Keerti Kulkarni, Bangalore. "A Raspberry Pi based CNN model for Indian Currency detection for Visually Impaired People". DOI: 10.1109/ICESC54411.2022.9885317.
- [4] Madhav Pasumarthy; Rutvi Padhy; Raghuveer Yadav; Ganesh Subramaniam; Madhav Rao. "An Indian Currency Recognition Model for Assisting Visually Impaired Individuals". DOI: 10.1109/RASSE54974.2022.9989624.
- [5] Ramasamy, Gokul, and Sakthi Subramanian. "Identification of currency denomination using image processing." *International Journal of Advance Research, Ideas and Innovations In Technology* 5 (2019): 1736-1740.
- [6] Komali, G. M. T., B. Chaturya, L. Manjulatha, J. Sri Durga, K. Jyothisna, and Mrs S. Rama Devi. "INDIAN CURRENCY RECOGNITION FOR VISUALLY CHALLENGED PEOPLE USING RESNET."
- [7] Borde, S., Patil, C., Sonawane, C., & Singh, M. (2023, May). Object Recognition based on Deep Learning Algorithms using Embedded IoT with Interactive Interface. In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1581-1586). IEEE
- [8] Sunitha M. R, Fathima Khan, Gowtham Ghatge R, Hemaya S, Karnataka. "Object Detection and Human Identification using Raspberry Pi". DOI: 10.1109/ICAIT47043.2019.8987398..
- [9] Sai, T. V., Aditya, B., Reddy, A. M., & Srinivasulu, Y. Real Time Object Detection Using Raspberry Pi
- [10] Akhila, Chitturi, Cheemala Astha Diamond, and A.P Mary. "Convolutional neural network based online rail surface crack detection." In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1602-1606. IEEE, 2021.
- [11] Nekkanti, Lakshmi Bhavani, A. Priyanga, A. Mary Posonia, and J. Albert Mayan. "Breaking Down Communication Barriers: Real-Time Sign Language Recognition Using CNN & Flask-Based API." In *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, pp. 330-336. IEEE, 2023.
- [12] Keerthana, K. T. E., S. Karpagavalli, and A.M Posonia. "Smart system monitoring agricultural land Using IoT." In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*, pp. 1-7. IEEE, 2018.
- [13] Ashraf, Dina, et al. "Voice Controlled Personal Assistant Using Raspberry Pi." *The International Undergraduate Research Conference*. Vol. 2. No. Second International Undergraduate Research Conference, IUGRC. The Military Technical College, 2017
- [14] Pranav Jain, Jitendra Kumar, Piyush Vashistha, Juginder Pal Singh, GLA University. "Raspberry Pi based voice-operated personal assistant (Neobot)". doi:10.1109/ICECA.2019.8821892
- [15] Cherry Soni, Maitri Saklani, Gunjan Mokhariwale, Aishwarya Thorat, Kunal Shejul, Pune. "Multi-Language Voice Control IOT Home Automation Using Google Assistant and Raspberry Pi". DOI: 10.1109/ACCAI53970.2022.9752606.
- [16] Jiao, Ziyun, Yi Yang, Hairong Zhu, and Fuji Ren. "Realization and improvement of object recognition system on raspberry pi 3b+." In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 465-469. IEEE, 2018.
- [17] Semary, Noura A., et al. "Currency recognition system for visually impaired: Egyptian banknote as a study case." *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*. IEEE, 2015.

ACCEPTANCE LETTER AND PAYMENT PROOF OF CONFERENCE

Notification of Paper Status - AIIoT 2024 - [Paper ID: 2940]

Microsoft CMT <email@msr-cmt.org>
to me

Thu, Mar 28, 12:25 AM (9 days ago)

Dear Abirami K,

Warm greetings from Vellore Institute of Technology, Vellore, India!

It is with great pleasure and enthusiasm that we extend our heartfelt congratulations to you! Your paper titled "Google pi using raspberry pi for visually impaired " with the paper ID 2940, submitted to the 3rd IEEE International Conference on Artificial Intelligence for Internet of Things (AIIoT 2024), has been provisionally accepted based on the recommendations of our esteemed reviewers.

You can access the comments provided by the reviewers by logging into the Microsoft CMT portal author dashboard: <https://cmt3.research.microsoft.com/User/Login?ReturnUrl=/AIIoT2024>.

To proceed further, kindly make the necessary modifications recommended by the reviewers and submit the camera-ready paper along with the e-copyright form in the Microsoft CMT portal.

For your convenience, we have prepared step-by-step instructions that you can follow: <http://tinyurl.com/siiot2024instructions>

Please note that at least one author must complete the registration by paying the full registration fee to ensure inclusion in the conference presentation and proceedings. You can complete the registration process by following this link: <https://vit.ac.in/AIIoT2024/>

Please be advised that the deadline for submitting the camera-ready copy, e-copyright, and registration is April 10, 2024.

We genuinely appreciate your valuable contribution to AIIoT 2024, and we look forward to welcoming you at the conference.

With Warm Regards,
Chair - AIIoT 2024
3rd IEEE International Conference on Artificial Intelligence For Internet of Things (AIIoT)

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our [Privacy Statement](#).

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052



VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

AIIOT - 2024

Registration Acknowledgement - Receipt

Reference No.	AIIOT24000169
Name	Abirami K
Address	16, Vasuki Street, Bharathi Nagar Extension, Sipcot,Ranipet 632403
Transaction Id	2400496355
Invoice No.	VL2400359176
Payment Date	06-04-2024 10:49:58

Participant Type	Non-IEEE Members
Participant Category	Students (UG/PG) & Research Scholars
Total Amount Paid (Incl. GST 18%)	9,440.00 INR