

HAROHALLI, KANAKAPURA ROAD – 562112

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

PROJECT REPORT

ON

#### “ARGUS: A MULTIMODEL ASSISTIVE SYSTEM FOR THE VISUALLY AND HEARING IMPAIRED (OBJECT AND SIGN LANGUAGE DETECTION)”

2023-2024

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

**Submitted by**

NITIN PRAJWAL R -ENG22DS0039

SUJEETH KUMAR D S-ENG22DS0019

VINURAJ VAMSHI-ENG22DS0047

**Under The Supervision of:**

#### Prof. MONISH. L

#### Assistant Professor

#### Department of CSE (Data Science), DSU

DAYANANDA SAGAR UNIVERSITY

#### DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SCHOOL OF ENGINEERING

**HAROHALLI, KANAKAPURA ROAD – 562112**



CERTIFICATE

#### It is certified that the mini project work entitled “ARGUS: A MULTIMODEL ASSISTIVE SYSTEM FOR THE VISUALLY AND HEARING IMPAIRED (OBJECT AND SIGN LANGUAGE DETECTION)” has been carried out at *Dayananda Sagar University*, Bangalore, by *NITIN PRAJWAL R-ENG22DS0039, SUJEETH KUMAR D S-ENG22DS0019, VINURAJ VAMSHI-ENG22DS0047*, Bonafide student of fourth Semester, B.tech. in partial fulfilment for the award of degree in *Bachelor of Technology in Computer Science & Engineering (Data Science)* during academic year *2023-24*. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in departmental library.

#### The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

**Signature of the Guide Signature of the Chairperson**

A project's successful completion offers a sense of satisfaction, but it is never finished without expressing gratitude to everyone who contributed to its accomplishment. We would like to convey our sincere gratitude to our esteemed university, Dayananda Sagar University, for offering the first-rate facilities.

I am especially thankful to our Chairperson, Dr. Shaila S G, for providing necessary departmental facilities, moral support and encouragement. The largest measure of our acknowledgment is reserved for Prof. Monish L whose guidance and support made it possible to complete the project work in a timely manner.

I would want to thank everyone who has assisted me in successfully completing this project work, both directly and indirectly. The staff has provided me with a great deal of direction and cooperation.

NITIN PRAJWAL R-ENG22DS0039

SUJEETH KUMAR D S-ENG22DS0019 VINURAJ VAMSHI-ENG22DS0047

I hereby declare that the Internship project entitled **“ARGUS: A MULTIMODEL ASSISTIVE SYSTEM FOR THE VISUALLY AND HEARING IMPAIRED (OBJECT AND SIGN LANGUAGE DETECTION)**” submitted to Dayananda Sagar University, Bengaluru, is a bona fide record of the work carried out by me under the guidance of Prof. MONISH L, assistant professor in the Dayananda Sagar University School of Engineering's Department of Computer Science and Engineering (Data Science). This work is submitted toward the partial fulfillment of the requirements for the award of a Bachelor of Technology in Computer Science and Engineering (Data Science).

NITIN PRAJWAL R-ENG22DS0039 SUJEETH KUMAR D S-ENG22DS0019 VINURAJ VAMSHI-ENG22DS0047

**ABSTRACT**

This project proposes an innovative assistive technology that leverages the capabilities of the Internet of Things (IoT) and Artificial Intelligence (AI) to address the unique challenges faced by individuals with visual or hearing impairments. Blindness and deafness can significantly limit a person's ability to communicate and interact with their environment, and traditional assistive technologies have often fallen short in meeting their diverse needs.

The proposed system integrates advanced AI algorithms for sign language recognition and object detection with IoT devices, creating a comprehensive assistive tool that can benefit both blind and deaf users. The sign language recognition component employs convolutional neural networks (CNNs) trained on extensive datasets to accurately interpret hand gestures and movements, facilitating real-time communication for deaf individuals. Simultaneously, the object detection module utilizes state-of-the-art deep learning techniques to identify and categorize objects in the user's surroundings, providing vital environmental awareness for blind individuals.

Through a network of strategically placed IoT sensors and cameras, the system continuously captures audiovisual data from the user's environment. This data is processed by AI algorithms running on a centralized server or edge devices, and the processed information is translated into intuitive feedback mechanisms tailored to the user's specific needs. For blind users, the system can provide auditory descriptions of the surroundings based on object detection output, assisting with navigation and situational awareness. For deaf users, the system can communicate using sign language output, either through pre-recorded videos or animations, facilitating effective communication.

Key features of this solution include adaptability to diverse sign language dialects and gestures, robust object recognition capabilities in various environmental conditions, and scalability to accommodate different living and working spaces. Furthermore, the system prioritizes user privacy and data security by adhering to stringent protocols for data encryption and anonymization.

Experimental assessments demonstrate a high-performance detection system, with an accuracy of 99.8% and an inferencing time of 2.22 seconds. These promising results showcase the system's potential to empower individuals with visual or hearing impairments by enhancing their communication capabilities and environmental awareness, promoting greater independence, social inclusion.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER** | **PG. NO.** |
| 1. INTRODUCTION | 9 |
| 1. OBJECTIVES AND SCOPE OF WORK | 11 |
| 1. DESCRIPTION OF WORK | 14 |
| 1. PROJECT MODEL DESCRIPTION | 15 |
| 1. CIRCUIT DIAGRAM | 26 |
| 1. METHODOLOGY | 28 |
| 1. DATASET CREATION AND ANNOTATION | 38 |
| 1. EXPERIMENTATION-CODE | 41 |
| 1. EVALUATION METRICS AND TEST CRITERIA | 45 |
| 1. CONCLUSION | 50 |
| 1. REFERENCES | 51 |
| 1. PLAGIARISM REPORT | 52 |

**TABLE OF DIAGRAMS**

|  |  |
| --- | --- |
| **DIAGRAM** | **PG. NO.** |
| Fig. 1.1 | 16 |
| Fig. 1.2 | 17 |
| Fig. 1.3 | 18 |
| Fig. 1.4 | 19 |
| Fig. 1.5 | 20 |
| Fig. 1.6 | 21 |
| Fig. 1.7 | 23 |
| Fig. 1.8, Fig. 1.9 | 24 |
| Fig. 1.10 | 25 |
| Fig. 2.1, Fig. 2.2 | 26 |
| Fig. 2.3 | 27 |
| Fig. 3.1, Fig. 3.2 | 29 |
| Fig. 3.3, Fig. 3.4 | 30 |
| Fig. 3.5 | 34 |
| Fig. 3.6 | 36 |
| Fig. 3.7 | 37 |
| Fig. 4.1, Fig. 4.2, Fig. 4.3, Fig. 4.4 | 40 |
| Fig. 5.1 | 41 |
| Fig. 5.2 | 42 |
| Fig. 5.3 | 43 |
| Fig. 5.4 | 44 |
| Fig. 6.1, Fig. 6.2, Fig. 6.3, Fig. 6.4 | 46 |
| Fig. 6.5, Fig. 6.6 | 47 |
| Fig. 6.7, Fig. 6.8, Fig. 6.9 | 48 |
| Fig. 6.10, Fig. 6.11, Fig. 6.12 | 49 |

### INTRODUCTION

Recent data from the World Health Organization indicates that about 5% of people worldwide suffer from hearing impairment, which can cause problems with verbal communication and, in certain situations, result in mutism. Speech problems are made worse by disorders like aphasia, cerebral palsy, and apraxia of speech. For those who are unable to express themselves verbally, sign language is an essential tool for bridging the communication gap between the mute and non-mute community. However, communicating effectively with deaf and mute individuals poses challenges for those who are not familiar with sign language, potentially leading to social isolation for the mute community.

There are 125 sign languages and 6909 spoken languages in the world, so it is crucial to facilitate communication between various linguistic communities. The two main types of gesture recognition techniques now in use are sensor-based and computer vision-based systems. While computer vision-based techniques entail intricate procedures like motion modeling and pattern identification, sensor-based systems provide a simpler way to record hand movements.

By allowing non-mute people to understand sign language movements, the IoT-Based Sign Language Recognition System described in this study seeks to reduce communication barriers between mute and non-mute groups. The technology will take the form of a wearable hand glove and a mobile application, utilizing ideas from machine learning, natural language processing, and the Internet of Things. User experience is given top priority in the hand glove design, and the iOS and Android mobile application has capabilities like voice-to-text conversion and user training.

The communication gap between individuals who are mute and those who are not poses significant challenges, often leading to social isolation and a lack of understanding. Despite the existence of numerous spoken and sign languages worldwide, facilitating effective communication between diverse linguistic communities remains a pressing issue. Around 5% of people worldwide suffer from hearing impairments, which can lead to mutism, especially in cases with illnesses like apraxia of speech, cerebral palsy, and aphasia, according to current data from the World Health Organization.

In response to this challenge, sign language emerges as a vital bridge between mute and non-mute individuals, enabling those who cannot verbalize their thoughts to communicate effectively. However, the intricacies of sign language can be daunting for those unfamiliar with it, leading to misunderstandings and hindering meaningful interactions. This lack of mutual comprehension can further exacerbate feelings of isolation within the mute community.

Existing solutions for sign language interpretation primarily fall into two categories: computer vision-based methods and sensor-based methods. Computer vision techniques involve complex processes such as motion modeling and pattern recognition, which may not always provide accurate or comfortable user experiences. In contrast, sensor-based approaches offer more direct means of capturing hand gestures, potentially leading to more intuitive and reliable interpretations.

The subsequent sections of this paper provide a comprehensive background survey, detailing existing sign language interpreters, followed by an explanation of the proposed Glove implementation and its environment. Test results and discussions are presented, leading to conclusions and future directions in the final sections of this study.

### OBJECTIVE

The objective of this project is to develop an integrated assistive technology system leveraging Internet of Things (IoT) and Artificial Intelligence (AI) techniques to aid individuals who are deafblind, deaf, or mute by providing real-time sign language detection and object recognition capabilities. Utilizing Raspberry Pi as the central processing unit, along with a camera module and a trained AI model, the system aims to offer seamless communication and environmental awareness. The system will be designed to:

* Enable real-time detection and interpretation of sign language gestures through the trained AI model, facilitating communication for individuals who are deaf or mute.
* Implement object detection algorithms to identify and categorize objects within the user's environment, providing crucial situational awareness for individuals who are blind.
* Utilize Bluetooth connectivity to establish a seamless connection between the Raspberry Pi-based system and a mobile device, enabling the display of interpreted sign language gestures and object descriptions on the mobile screen for individuals who are deaf or mute.
* Integrate audio output capabilities, such as a speaker connected to the Raspberry Pi, to audibly convey interpreted information for individuals who are blind, ensuring accessibility through spoken descriptions of detected objects and interpreted sign language.
* Ensure portability and versatility by designing the system to be compact and easily deployable, allowing users to carry the device with them in various settings.
* Prioritize user privacy and data security by implementing encryption protocols for Bluetooth communication and adhering to best practices for data handling and storage.

By achieving these objectives, the project aims to develop a comprehensive assistive technology solution that empowers individuals with multiple sensory impairments, promoting independence, communication, and inclusivity in their daily lives.

### SCOPE OF WORK

The scope of this project encompasses the design, development, and implementation of an IoT and AI-based assistive technology system tailored specifically for individuals who are deafblind, deaf, or mute. The project will focus on the following key aspects:

1) Hardware Development:

* Selection and configuration of Raspberry Pi as the central processing unit.
* Integration of a camera module compatible with Raspberry Pi for capturing audiovisual data.
* Incorporation of Bluetooth connectivity capabilities to establish communication with mobile devices.
* Integration of audio output components, such as speakers, for conveying interpreted information.

2) AI Model Development:

* Collection and curation of datasets for training AI models on sign language detection and object recognition.
* Development and optimization of AI algorithms, leveraging techniques such as convolutional neural networks (CNNs) for accurate detection and interpretation.
* Training and fine-tuning of the AI model to ensure robust performance across various sign language gestures and environmental conditions.

3) Software Development:

* Design and implementation of software modules for real-time processing of audiovisual data captured by the camera module.
* Development of algorithms for sign language detection and object recognition, utilizing the trained AI model.
* Integration of Bluetooth communication protocols for seamless connectivity with mobile devices.
* Implementation of user interface elements for displaying interpreted sign language gestures and object descriptions on mobile screens.

4) Testing and Evaluation:

* Conducting thorough testing to assess the accuracy, reliability, and usability of the system in real-world scenarios.
* Obtaining input from intended users to pinpoint areas in need of development and enhancement.
* Iterative testing and optimization of the system based on user feedback and performance metrics.

5) Deployment and Accessibility:

* Packaging the system into a portable and user-friendly form factor for easy deployment and use.
* Providing documentation and instructional materials to guide users in setting up and operating the system.

### DESCRIPTION OF WORK

Our project aims to develop an innovative assistive technology system that leverages IoT and AI to improve communication and environmental awareness for individuals who are deafblind, deaf, or mute. Using Raspberry Pi as the core processing unit, along with a camera module and a trained AI model, our system enables real-time detection of sign language gestures and object recognition.

Through Bluetooth connectivity, the system seamlessly communicates with mobile devices, displaying interpreted sign language gestures and object descriptions on the screen for individuals who are deaf or mute. Additionally, the system provides audio output through a speaker for individuals who are blind, conveying the interpreted information aloud. This comprehensive solution aims to enhance accessibility and independence for individuals with multiple sensory impairments, promoting inclusivity and empowerment in their daily lives

### PROJECT MODEL DESCRIPTION:

**Hardware components Software components**

1. Raspberry pi 3B 1. Raspberry pi OS-ARM 64
2. Camera 2. TensorFlow Lite
3. SD card 3. Deep Learning Model
4. Bluetooth and Wi-Fi module 4. Machine Learning Model
5. USB cable 5. Object detection
6. Ethernet 6. Sign Language recognition
7. HDMI 7. Claude AI
8. System on Chip
9. Earphone/Speaker
10. Power Bank

# 4.1) Raspberry pi 3B

A Raspberry Pi is a small, affordable computer that has gained immense popularity for its versatility and ease of use. Originally developed to promote teaching of basic computer science in schools and developing countries, it has since evolved into a powerful tool for hobbyists, educators, and professionals alike. The Raspberry Pi is a versatile device that may be used for many different tasks, such as media centers, DIY robots, home automation, and retro gaming consoles. Its compact size, low power consumption, and GPIO (General Purpose Input/Output) pins make it ideal for experimenting with electronics and interfacing with sensors and other hardware components. Additionally, its open-source nature encourages a vibrant community of developers who continuously create and share innovative projects and software applications. In essence, the Raspberry Pi empowers users to explore the realms of computing and electronics in a cost-effective and accessible manner, making it an invaluable tool for learning, experimentation, and innovation.

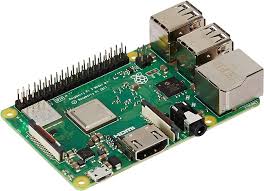


Fig. 1.1

## Raspberry Pi can be used to experiment with AI and machine learning applications, albeit with some limitations due to its hardware constraints. Projects range from simple image recognition to building voice-controlled assistants and autonomous robots, offering opportunities for experimentation and learning in these cutting-edge fields. All things considered, the Raspberry Pi's adaptability, low cost, and strong community support make it a useful tool for a variety of uses, from useful DIY projects to educational initiatives and more.

## 4.2) Camera

Using a camera for both sign language detection and object detection involves leveraging computer vision techniques to analyze visual input and extract relevant information. In sign language detection, the camera captures video of a person performing sign language gestures. These gestures are then processed by algorithms designed to recognize and interpret the movements of hands and fingers, as well as facial expressions and body language associated with sign language communication. Convolutional neural networks (CNNs), one type of deep learning model, are frequently employed for this purpose. To identify patterns and correlations between hand gestures and matching signs, these algorithms are trained on enormous datasets of sign language gestures. These models recognize and interpret sign language movements into text or speech by examining the spatial and temporal properties of hand positions and motions.On the other hand, object detection with a camera involves identifying and localizing objects within a scene.

Usually, there are two primary processes in this process: object classification and object localization. The goal of object localization, which is typically represented by bounding boxes, is to pinpoint the exact location and size of each object within the image. Giving each localized object a label or category is the process of object classification. Deep learning architectures such as single-shot detectors (SSDs) or region-based convolutional neural networks (R-CNNs) are frequently used in state-of-the-art object detection systems. To properly detect and categorize a variety of items, these models are trained on large annotated datasets that contain photos with tagged objects.



Fig. 1.2

In both sign language detection and object detection tasks, the performance of the system depends heavily on the quality of the training data, the design of the algorithms, and the computational resources available for inference. Real-time applications may require efficient algorithms and hardware acceleration to process video streams at high frame rates. Additionally, considerations such as lighting conditions, background clutter, and occlusions can pose challenges that need to be addressed through robust algorithm design and preprocessing techniques. Despite these challenges, camera-based systems for sign language detection and object detection have shown great promise in various applications, including accessibility tools for the deaf and hard of hearing, smart surveillance systems, and assistive technologies for the visually impaired. Continued research and development in computer vision will likely lead to further improvements in the accuracy, speed, and robustness of these systems, enabling their deployment in a wide range of real-world scenarios.

## 4.3) SD card

In an IoT project focusing on sign language and object detection with a Raspberry Pi, the SD card serves as the primary storage medium for the operating system, application software, and data. The Raspberry Pi relies on the SD card as its main storage device, where the Raspberry Pi OS (formerly Raspbian) is installed. This operating system provides the necessary environment for running the sign language and object detection algorithms, managing hardware resources, and interfacing with connected peripherals, including the camera module. Firstly, the SD card contains the Raspberry Pi OS, which includes the required software libraries, drivers, and dependencies for implementing the sign language and object detection functionalities. These could include deep learning frameworks like Tensorflow Lite or PyTorch for training and deploying detection models, computer vision libraries like OpenCV for processing camera input, and additional tools for managing system resources, handling network communication, and GPIO pin interface.

Secondly, the SD card stores the application code and data related to the sign language and object detection algorithms. This includes the trained models, configuration files, and any auxiliary data needed for inference or real-time processing. For sign language detection, the SD card holds the machine learning models trained to recognize sign language gestures from camera input, while for object detection, it stores models trained to detect specific objects within a scene.



Fig. 1.3

Furthermore, the SD card serves as a storage repository for logging data, captured images or videos, and other metadata generated during the operation of the IoT project. This data may be used for performance evaluation, debugging, or further analysis to improve the accuracy and robustness of the sign language and object detection algorithms over time.Overall, the SD card plays a crucial role in facilitating the functionality and operation of the Raspberry Pi-based IoT project for sign language and object detection. By providing storage for the operating system, application software, and data, the SD card enables the Raspberry Pi to perform real-time processing of camera input, recognize sign language gestures, detect objects within the environment, and contribute to the advancement of accessibility and automation solutions in various domain.

## 4.4) USB cable

## 

## Fig. 1.4

## The USB cable is an essential connection between the Raspberry Pi and the camera module in an Internet of Things project that uses a Raspberry Pi and a camera module to detect objects and interpret sign language. The camera module records visual data in the form of picture or video streams and is usually connected to the Raspberry Pi through a USB port. The Raspberry Pi's processing powers are then applied to these visual inputs, potentially executing computer vision algorithms for tasks like object detection and sign language recognition.

## The USB cable facilitates data transfer between the camera module and the Raspberry Pi, allowing the Raspberry Pi to access the live video feed or captured images from the camera module. Additionally, the USB connection may also provide power to the camera module, simplifying the overall setup by eliminating the need for separate power sources. By integrating the camera module with the Raspberry Pi using a USB cable, the IoT project can leverage the combined capabilities of both devices to perform real-time analysis of sign language gestures and object detection tasks. This integration enables applications such as smart surveillance systems that can detect and interpret visual cues in the environment, as well as assistive technologies for individuals with hearing or visual impairments. Overall, the USB cable plays a pivotal role in facilitating communication and collaboration between the Raspberry Pi and the camera module, enabling the implementation of sophisticated IoT solutions for sign language and object detection.

## 4.5) Power Bank

## 

## Fig. 1.5

## In an IoT project focused on sign language and object detection using a Raspberry Pi and a camera module, a power bank serves as a portable power source to ensure continuous operation of the system. The Raspberry Pi, being a low-power device, can be powered directly from the power bank via its USB port.

## Additionally, the camera module, which captures video input for both sign language detection and object detection tasks, is also powered by the Raspberry Pi through its GPIO pins or USB connection. By connecting the power bank to the Raspberry Pi, users can deploy the IoT system in various environments without the need for a fixed power source, enhancing its mobility and versatility.

## The power bank provides a reliable and stable power supply to the Raspberry Pi and camera module, enabling them to function autonomously without relying on grid power. This is particularly beneficial for IoT applications that require deployment in remote or outdoor settings where access to power outlets may be limited. Moreover, the portable nature of the power bank allows for easy setup and repositioning of the IoT system as needed, facilitating experimentation and data collection in different environments.

## 4.6) Bluetooth and Wi-Fi module

## 

## Fig. 1.6

## In an IoT project focused on sign language and object detection, Raspberry Pi can utilize Bluetooth and Wi-Fi modules along with a camera module to create a versatile and connected system. The Raspberry Pi's GPIO pins can be used to interface with Bluetooth and Wi-Fi modules, allowing it to communicate wirelessly with other devices and networks.

## The Bluetooth module can enable communication with nearby wearable devices or sensors, potentially worn by users to track sign language gestures or transmit data from other IoT devices. Wi-Fi connectivity, on the other hand, enables the Raspberry Pi to connect to the internet and interact with cloud services, allowing for remote monitoring, data storage, and analysis.

## By combining Bluetooth and Wi-Fi connectivity with the camera module, the Raspberry Pi can serve as a central hub for collecting, processing, and transmitting data related to sign language communication and object detection in an IoT context. This setup opens up possibilities for various applications, including assistive technologies for the deaf and hard of hearing, smart surveillance systems, and interactive educational tools. Additionally, the modular nature of Raspberry Pi allows for flexibility and scalability in designing and deploying IoT solutions tailored to specific use cases and environments.

## 4.7) System on chip

## 

## Fig. 1.7

In an IoT project focused on sign language and object detection, a system-on-chip (SoC) like the one used in Raspberry Pi plays a crucial role in providing the computational power and flexibility needed for real-time image processing and analysis. The SoC integrates various components such as CPU, GPU, memory, and I/O interfaces onto a single chip, offering a compact and energy-efficient solution ideally suited for embedded applications. In this project, the SoC of the Raspberry Pi serves as the central processing unit responsible for running the computer vision algorithms required for sign language and object detection. These algorithms utilize the GPU for parallel processing, enabling faster inference speeds and real-time performance, crucial for applications where timely detection is essential. Additionally, the SoC facilitates seamless integration with peripherals and sensors, including the camera module. The camera module captures live video feeds of the surrounding environment, providing the input data for the computer vision algorithms. By interfacing with the SoC via standard interfaces like CSI (Camera Serial Interface), the camera module ensures high-speed data transfer and synchronization with the processing unit. The combined capabilities of the SoC and camera module enable the IoT device to analyze visual input in real-time, detecting sign language gestures and objects within the scene. This integration of hardware components in the Raspberry Pi ecosystem enables the development of sophisticated IoT solutions for accessibility, surveillance, and automation, showcasing the power of embedded systems in addressing real-world challenges.

## 4.8) Ethernet and HDMI cables

## Fig. 1.8 Fig. 1.9

In an IoT project focusing on sign language and object detection using a Raspberry Pi and a camera module, both Ethernet and HDMI cables play crucial roles in facilitating connectivity and data transfer. The Ethernet cable enables the Raspberry Pi to connect to a local network or the internet, providing access to remote servers, cloud services, or other networked devices. This connectivity is essential for transmitting data, receiving software updates, and accessing external resources such as databases or APIs. With a stable network connection, the Raspberry Pi can reliably send captured video footage from the camera module to remote servers or cloud platforms for further processing and analysis. Additionally, the Ethernet connection allows for remote monitoring and control of the IoT system, enabling users to interact with the device and receive real-time feedback on detected signs or objects.

Meanwhile, the HDMI cable serves as the primary interface for connecting the Raspberry Pi to a display monitor or TV screen. This allows users to visualize the output of the IoT system in real-time, including live video feeds from the camera module and any graphical user interfaces (GUIs) or visualizations generated by the software running on the Raspberry Pi. For instance, users may want to see the detected sign language gestures or recognized objects displayed on a screen for verification or feedback purposes. The HDMI connection also supports audio output, enabling the Raspberry Pi to generate speech synthesis or auditory alerts based on the detected signs or objects.

Overall, the Ethernet and HDMI cables enable seamless integration of the Raspberry Pi and camera module into the IoT project, providing essential connectivity and display capabilities for sign language and object detection applications. By leveraging these cables, users can access and interact with the IoT system remotely, monitor its operation in real-time, and visualize the results of sign language and object detection algorithms, thereby enhancing accessibility, usability, and overall effectiveness of the project.

## 4.9) Earphone/Speaker

## 

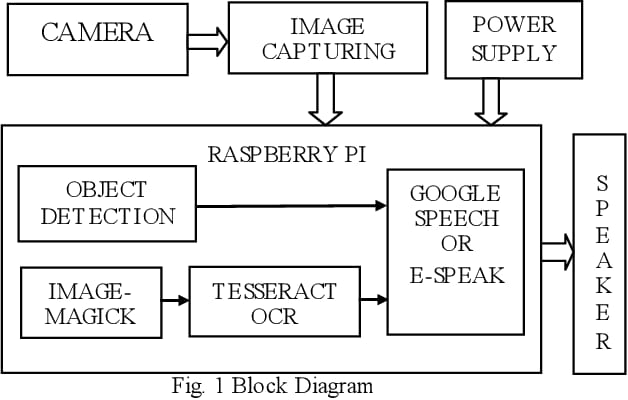
## Fig. 1.10

We use this device to give the output whether is there an obstacle or not and it also helps us to get to know on the sign language detected and this reads it out loud. This can be connected via Bluetooth and the information can be passed wireless from the raspberry pi to this earphone. One must simply wear this and the system will automatically detect this device and will start pairing with it. It is programmed in such a way that it doesn’t provide any kind of complexity in using this device for the impaired person or the challenged people. By this model one can easily communicate the outside world without having any kind of inferiority complex created and can express their thoughts more clearly.

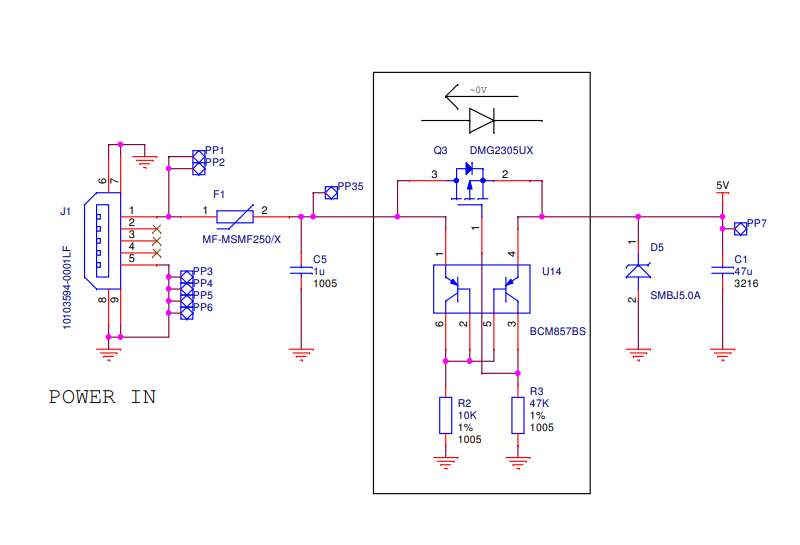
### CIRCUIT DIAGRAM

### 

## Fig. 2.1



## Fig. 2.2



## Fig. 2.3

### 6) Methodology

### Enabling Navigation for the Visually Impaired:

### Creating an IoT project using a Raspberry Pi that incorporates object detection, navigational path planning, audio feedback, and text-to-speech integration involves several key steps. Here's a detailed methodology:

### Setup Raspberry Pi: Begin by setting up the Raspberry Pi with the required operating system (e.g., Raspbian). Ensure that the necessary libraries and dependencies for OpenCV, Tensorflow Lite, and any other relevant packages are installed.

### Install Object Detection Frameworks: Install OpenCV and Tensorflow Lite on the Raspberry Pi. These frameworks will be used for object detection. OpenCV provides a wide range of computer vision functionalities, while Tensorflow Lite offers efficient inference for machine learning models on resource-constrained devices like the Raspberry Pi.

### Collect and Label Data: Gather a dataset of images containing the objects you want to detect. Label the objects in the images to create annotated training data. Tools like Labeling can be used for this purpose.

### Train Object Detection Model: To train an object detection model, use the labeled dataset. Tensorflow Lite offers pre-trained models that you may select from, or you can use transfer learning to train your own model. A model that takes into account the Raspberry Pi's processing limitations should be tailored for deployment on the device.

### • Integrate Object Detection: Utilize Python programs to record footage from the Raspberry Pi's attached camera module. Real-time object detection in video frames can be achieved by employing the trained object detection model. For visualization, overlay labels or bounding boxes on the items that have been detected.

### Implement Navigational Path Planning: Implement a path planning algorithm to navigate the Raspberry Pi based on the detected objects. This could involve simple obstacle avoidance techniques or more sophisticated path planning algorithms like A\* or Dijkstra's algorithm. Ensure that the Raspberry Pi can move or navigate its environment accordingly.

### Audio Feedback and Text-to-Speech Integration: Integrate audio feedback using speakers connected to the Raspberry Pi. Use a text-to-speech (TTS) engine like Google Text-to-Speech or pyttsx3 to convert textual information (e.g., detected objects, navigation instructions) into spoken audio. This feedback can provide real-time updates to users about the detected objects and navigation decisions.

### User Interface (Optional): Develop a user interface, either graphical or command-line based, to interact with the IoT system. This interface can display information about detected objects, provide control options for navigation, and offer settings for audio feedback preferences.

### Testing and Optimization: Test the complete IoT system in a controlled environment to ensure that object detection, path planning, audio feedback, and text-to-speech integration work seamlessly together. Optimize the system for performance, considering factors like latency, accuracy, and resource usage on the Raspberry Pi.

### Deployment: Deploy the IoT system in the intended environment, whether it's a smart home, robotics project, or assistive technology application. Monitor the system's performance and address any issues that arise during real-world use.

### By following this methodology, you can create a comprehensive IoT project using a Raspberry Pi that incorporates object detection, navigational path planning, audio feedback, and text-to-speech integration to enhance its functionality and usability in various applications.

### 

## Fig. 3.1 Fig. 3.2

### 

## Fig. 3.3

### 

## Fig. 3.4

6.1) Facilitating Communication for the Hearing Impaired

Developing an IoT project using a Raspberry Pi for sign language recognition involves several steps, from gathering hardware components to implementing the software algorithms. Here's a detailed methodology for such a project:

1. Hardware Setup:

* Obtain a Raspberry Pi board (such the Raspberry Pi 4) along with the required add-ons (microSD card, power supply, etc.).
* Acquire a Raspberry Pi camera module that works with it (such as the Raspberry Pi Camera Module).
* Attach the camera module to the camera port on the Raspberry Pi.
* Use an Ethernet cable or Wi-Fi adaptor to connect the Raspberry Pi to a network.
* Connect Raspberry Pi to display monitor or TV screen via HDMI cable for visualization purposes.

1. Installation of Software:

* Boot the Raspberry Pi after installing the operating system (such as Raspberry Pi OS) on the microSD card.
* Set up the necessary development environment, including Python and any libraries or frameworks needed for computer vision and machine learning.

1. Dataset Collection and Preparation:

* Gather a dataset of sign language gestures, including images or video clips representing various signs.
* Annotate the dataset with corresponding labels indicating the sign language gestures.
* Preprocess the dataset by resizing images, normalizing pixel values, and augmenting data (if necessary) to increase variability and improve model generalization.

1. Model Training:

* Using a deep learning framework such as Tensorflow Lite or PyTorch, design and develop a CNN-based model for detecting sign language gestures.
* Split the dataset into training, validation, and test sets for model training and evaluation.
* Train the model using the training dataset, optimizing model parameters to minimize classification errors and maximize accuracy.
* Validate the model's performance on the validation set, tuning hyperparameters and adjusting the model architecture as needed to prevent overfitting.
* Assess the final trained model's generalization capacity and performance measures (e.g., accuracy, precision, recall, and F1-score) on the test set.

1. Integration with Raspberry Pi:

* Develop software scripts to interface with the camera module and capture live video streams.
* Preprocess the video frames by resizing, cropping, and normalizing pixel values to match the input requirements of the trained model.
* Implement inference logic to feed preprocessed video frames into the trained CNN model and classify sign language gestures in real-time.
* Integrate feedback mechanisms to provide visual or auditory feedback based on the detected gestures, such as displaying recognized signs on the connected display or generating spoken feedback using text-to-speech synthesis.
* Optimize the software implementation for real-time performance on the Raspberry Pi, considering computational resource constraints and efficiency.

1. Testing and Deployment:

* Test the integrated IoT system thoroughly to ensure robustness, accuracy, and reliability under various conditions, including different lighting environments, backgrounds, and hand orientations.
* Fine-tune the system parameters and algorithms based on user feedback and performance evaluations.
* Deploy the sign language recognition IoT project in real-world settings, monitoring its operation and gathering additional data for continuous improvement.
* Provide user documentation and instructions for using the system effectively, including troubleshooting tips and maintenance guidelines.

By following this methodology, you can develop a comprehensive IoT project using a Raspberry Pi for recognizing sign language gestures, integrating CNN-based sign language recognition with feedback mechanisms, and deploying a functional system for real-world applications.

6.2) Incorporating Claude AI Personal Assistant:

We will utilize the Anthropic API to integrate the Claude language model into our Raspberry Pi system. This API provides an interface that allows us to send user queries and receive responses from the Claude model. To interact with the API, we need to obtain the necessary credentials from the Anthropic website. The official Python library provided by Anthropic will be used to establish communication with the API and handle the request-response cycle seamlessly.

Voice Interaction and Natural Language Processing:

For voice interaction, we will employ the 'speech\_recognition' Python library, which enables the system to transcribe audio input from the user into text format. This text can then be sent to the Claude model through the Anthropic API for processing and generating an appropriate response.

To enhance the system's natural language processing capabilities, we will leverage the 'nltk' (Natural Language Toolkit) library. This library provides various tools and algorithms for tasks such as tokenization, stemming, and sentiment analysis. By preprocessing the user's input using these techniques, we can improve the system's understanding and interpretation of the queries, leading to more accurate and relevant responses from the Claude model.

Multimodel Communication (Voice, Text, Sign Language):

In addition to voice input, the system will support text input for users who prefer typed communication. The user's text input will be directly sent to the Claude model through the Anthropic API, and the generated response will be displayed on the Raspberry Pi's screen or a connected display.

For sign language communication, we will integrate computer vision techniques to detect and recognize sign language gestures from the camera input. Libraries like OpenCV and TensorFlow can be utilized for this purpose. The recognized gestures will be translated into text and processed by the Claude model. The model's response will then be rendered as sign language animations or videos on the display, enabling effective communication with deaf users.

Context-aware Assistance and Information Retrieval:

The Anthropic API allows for maintaining conversational context, enabling the Claude model to understand and respond based on the user's previous queries and the current state of the system. This contextual information can be used to provide more relevant and personalized assistance.

Furthermore, the Claude model can be fine-tuned or trained on specific datasets related to the system's functionality, such as object detection data or navigation information. This process will enhance the model's ability to provide accurate and context-aware information retrieval, tailored to the specific needs of the users.

Use Cases for Blind and Deaf Individuals:

1. Blind Users:

- The AI assistant can provide auditory descriptions of the user's surroundings based on the object detection output, assisting with navigation and situational awareness.

- Users can ask the AI assistant for information, directions, or instructions, and receive auditory responses through text-to-speech conversion.

- The AI assistant can interpret and respond to voice commands, enabling hands-free interaction and control of the system.

2. Deaf Users:

- The AI assistant can communicate using sign language output, either through pre-recorded videos or animations, facilitating effective communication for deaf individuals.

- Users can input queries or commands through sign language gestures, which the system can recognize and interpret using computer vision techniques.

- The AI assistant can provide visual information, instructions, or responses on the Raspberry Pi's display or a connected screen.

## 

## Fig. 3.5

6.3) Real-Time Speech-to-Text Conversion for Deaf Users:

One of the key components of our system is the ability to provide real-time speech-to-text conversion, enabling effective communication for deaf individuals. This feature allows deaf users to understand spoken language by converting audio input into readable text format with minimal delay.

To achieve this, we will leverage the 'speech\_recognition' Python library, which provides access to various speech recognition engines and APIs. We will specifically utilize the Google Speech Recognition API, known for its accuracy and low-latency performance.

The speech-to-text conversion process will work as follows:

1. Audio Input: The system will continuously capture audio input from a microphone or other audio source using the 'speech\_recognition' library.

2. Audio Processing: The captured audio will be processed in real-time, breaking it down into smaller chunks or frames for efficient transmission to the speech recognition API.

3. API Integration: Each audio chunk will be sent to the Google Speech Recognition API, which will perform the speech-to-text conversion and return the recognized text.

4. Text Display: The recognized text will be displayed on the Raspberry Pi's screen or a connected display in real-time, allowing deaf users to follow the spoken conversation or audio content as it happens.

To further enhance the user experience, we will implement the following additional features:

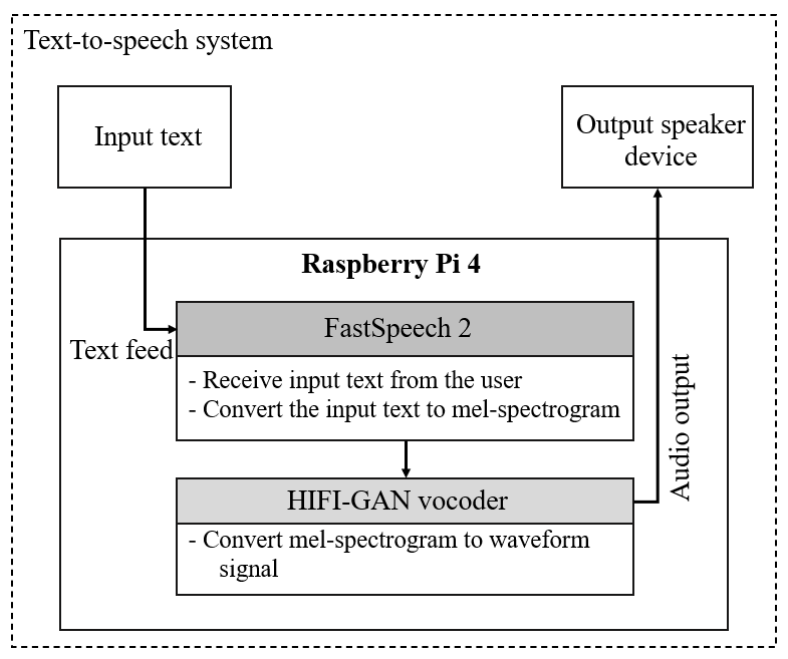
1. Speaker Identification: The system will attempt to identify different speakers and display their recognized text in different colors or with labels, making it easier to follow multi-person conversations.

2. Punctuation and Capitalization: The speech recognition output will be post-processed to include proper punctuation and capitalization, improving readability and comprehension.

3. Word Highlighting: As the speech-to-text conversion progresses, the system will highlight the currently spoken word or phrase, providing a visual cue for better comprehension.

4. Integration with Claude AI: The real-time speech-to-text output can be passed to the Claude AI assistant through the Anthropic API, enabling deaf users to engage in natural language conversations or queries with the AI system.

By providing real-time speech-to-text conversion with minimal delay, our system will significantly improve the accessibility and communication experience for deaf users. They will be able to follow spoken conversations, lectures, or audio content seamlessly, enhancing their ability to engage and participate in various settings.

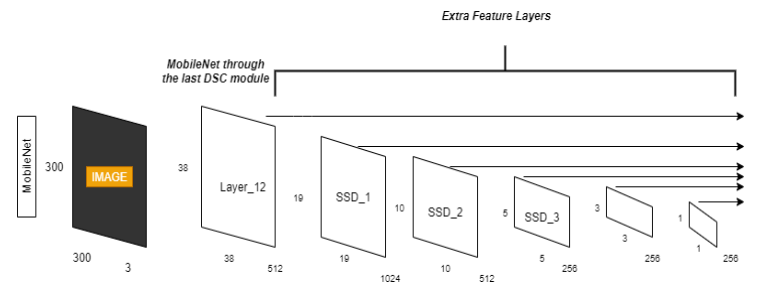


## Fig. 3.6

6.4) Deep Learning Models and Algorithms

The methodology for an IoT project utilizing Raspberry Pi for object detection aimed at assisting blind individuals involves a comprehensive approach integrating deep learning models and algorithms optimized for deployment on the Raspberry Pi platform. Initially, the project entails understanding convolutional neural networks (CNNs) and their applicability to object detection tasks. CNNs are particularly well-suited for this purpose due to their ability to automatically learn and extract features from images, enabling accurate identification and localization of objects within a scene. Transfer learning, a technique where pre-trained CNN models are fine-tuned on a specific dataset, is employed to leverage existing knowledge and adapt it to the object detection requirements of the project. Next, the training procedures are optimized to ensure efficient use of computational resources on the Raspberry Pi. This involves streamlining the data preprocessing pipeline, optimizing model architectures for performance and memory constraints, and selecting appropriate hyperparameters for training. Techniques such as quantization, which reduces the precision of model weights and activations to reduce memory footprint, and model pruning, which removes unnecessary parameters to improve inference speed, may be employed to optimize the deep learning models for deployment on resource-constrained devices like the Raspberry Pi.

Once the deep learning models are trained and optimized, they are deployed onto the Raspberry Pi for real-time object detection. This deployment phase involves integrating the models with the Raspberry Pi's hardware and software infrastructure, ensuring compatibility and efficiency.

****

## Fig. 3.7

**7) Dataset Creation and Annotation:**

To train a custom AI model for object detection and sign language recognition, we needed to create a comprehensive dataset of images. We utilized the Raspberry Pi's built-in camera module and an external USB camera to capture images in various environments and lighting conditions.

For object detection, we captured images of different objects, such as pens, bottles, people, animals, and other common items that a visually impaired person might encounter in their daily life. Similarly, for sign language recognition, we recorded videos and extracted frames of different sign language gestures performed by multiple individuals.

After collecting the raw data, we used the Labeling tool to annotate the images and videos manually. With Labeling, an open-source graphical application for annotation of images, users may draw bounding boxes around objects or movements and assign labels to them. This process was time-consuming but crucial for creating a high-quality dataset.

Augmentation Strategies:

To increase the diversity and robustness of our dataset, we employed various data augmentation techniques. These strategies help the AI model generalize better and prevent overfitting to the training data.

1. Geometric Transformations: We applied geometric transformations such as rotation, flipping, scaling, and cropping to the images and video frames. These transformations simulated different perspectives and viewing angles, making the model more resilient to variations in object orientation and sign language gestures.

2. Brightness and Contrast Adjustments: We adjusted the brightness and contrast levels of the images to mimic different lighting conditions. This helped the model learn to detect objects and recognize gestures under varying illumination levels.

3. Gaussian Noise: We added Gaussian noise to the images to simulate real-world scenarios where images may be affected by sensor noise or environmental factors.

These augmentation strategies significantly increased the size and diversity of our dataset, allowing the AI model to learn more robust features and improve its performance.

Preprocessing and Feature Extraction Methods:

Before training the AI model, we performed several preprocessing and feature extraction steps to prepare the data:

1. Image Resizing: We resized the images to a consistent resolution to ensure compatibility with the input requirements of the AI model. This step also helped to reduce computational overhead during training and inference.

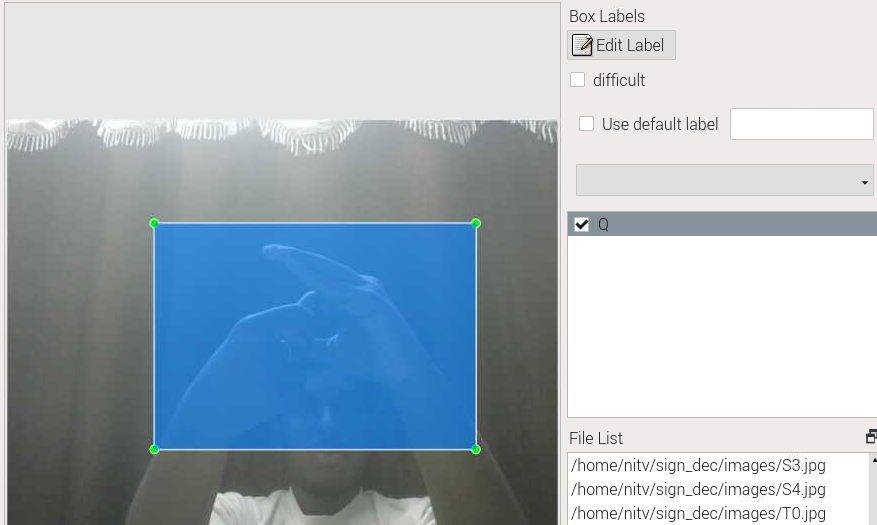
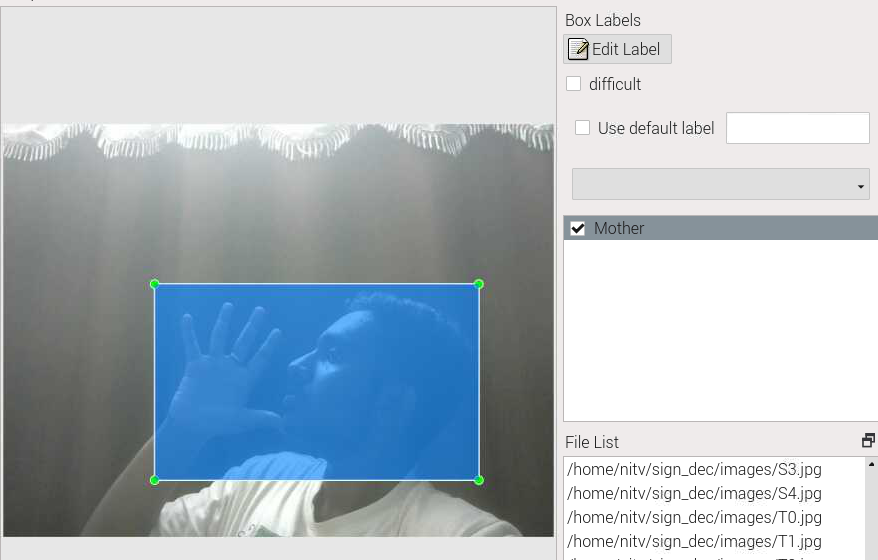
2. Normalization: Depending on the needs of the model, we normalized the image pixel values to a range of either 0 and 1 or -1 and 1. During training, normalization aids in enhancing numerical stability and convergence.

3. Feature Extraction: For object detection, we extracted features such as edges, textures, and shapes using techniques like Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT). These features provided valuable information to the AI model for identifying and localizing objects in the images.

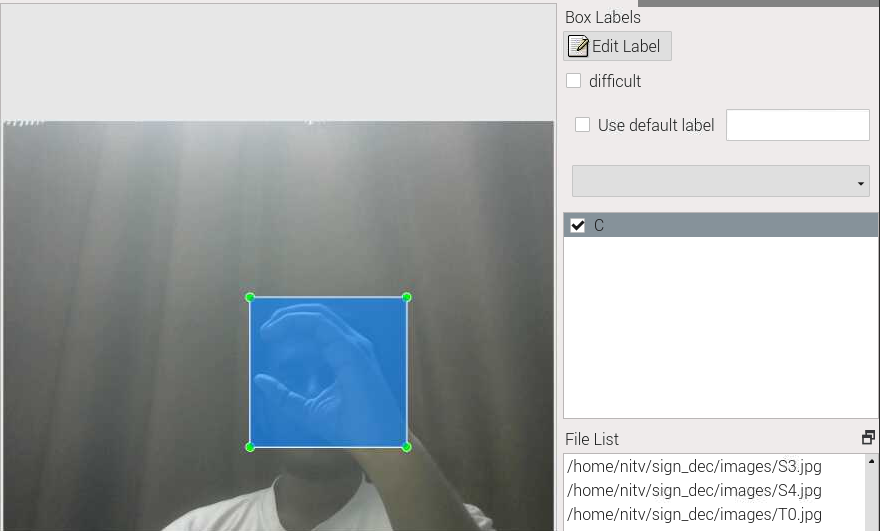
4. Encoding: For sign language recognition, we encoded the video frames as a sequence of images or optical flow representations, which captured the motion and temporal information of the gestures.

Features were retrieved from the preprocessed dataset and then grouped into training, validation, and testing sets. Training was done on the training set; validation was used to adjust hyperparameters and assess the model during training; testing was used to assess performance and assess the final model.

By following this rigorous data collection and preparation process, we were able to create a high-quality dataset tailored to our specific use case, enabling the training of a custom AI model that could effectively detect objects and recognize sign language gestures for the visually and hearing impaired.

** **

## Fig. 4.1 Fig. 4.2

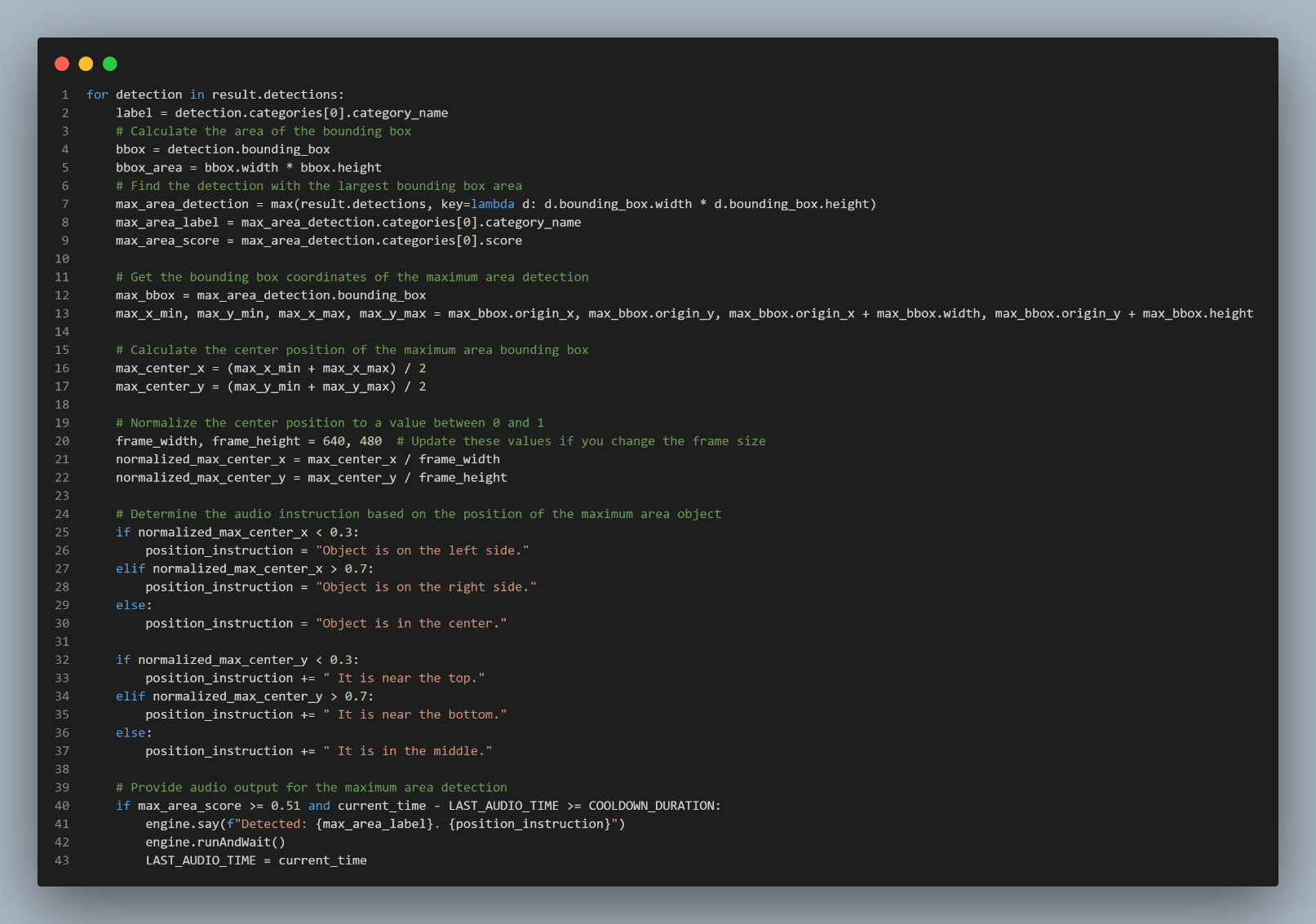
** **

## Fig. 4.3 Fig. 4.4

**8) IMPLEMENTATION**

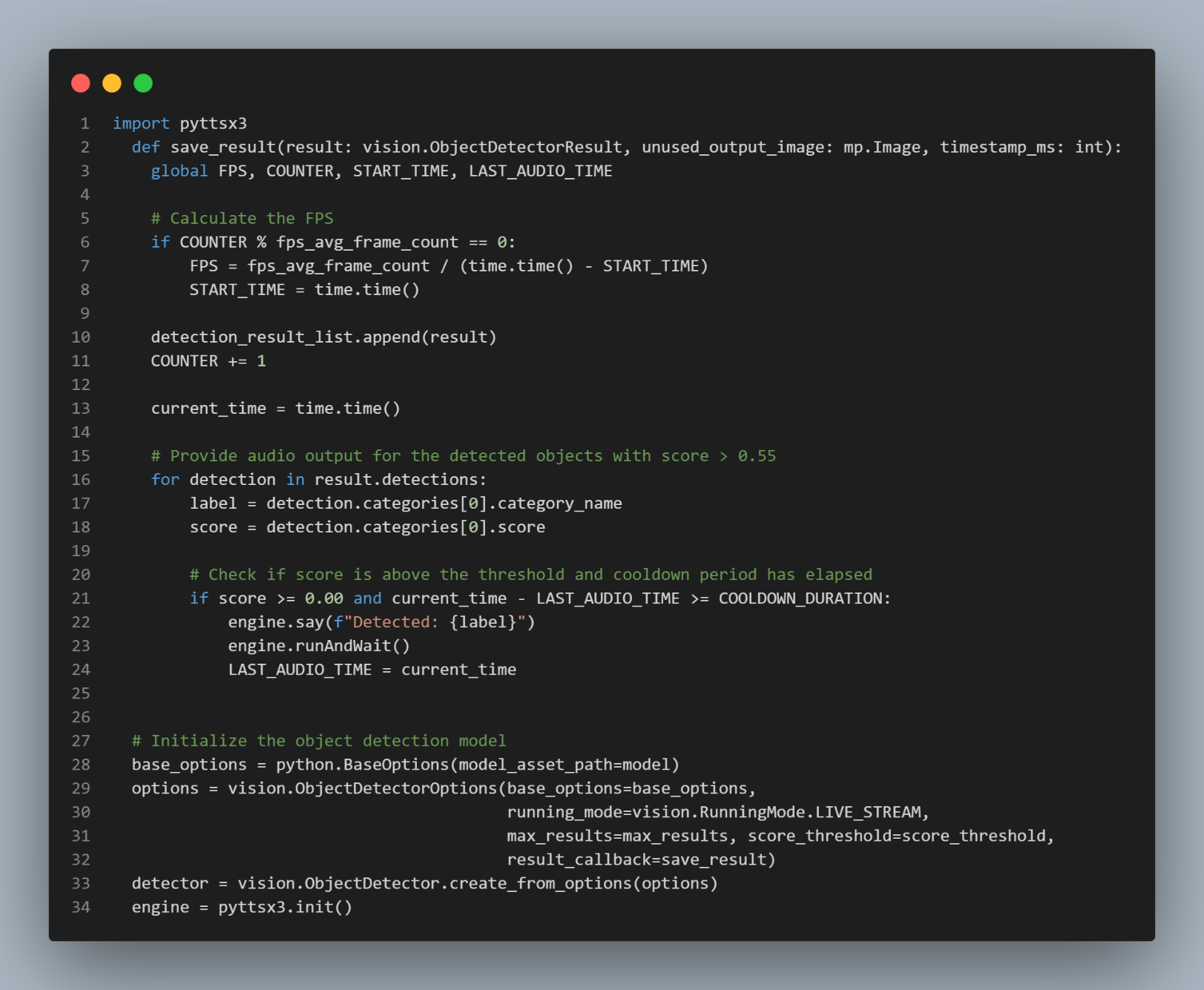
SOURCE CODE:

**8.1) OBJECT DETECTION**



## Fig. 5.1

**8.2) SIGN LANGUAGE DETECTION**

****

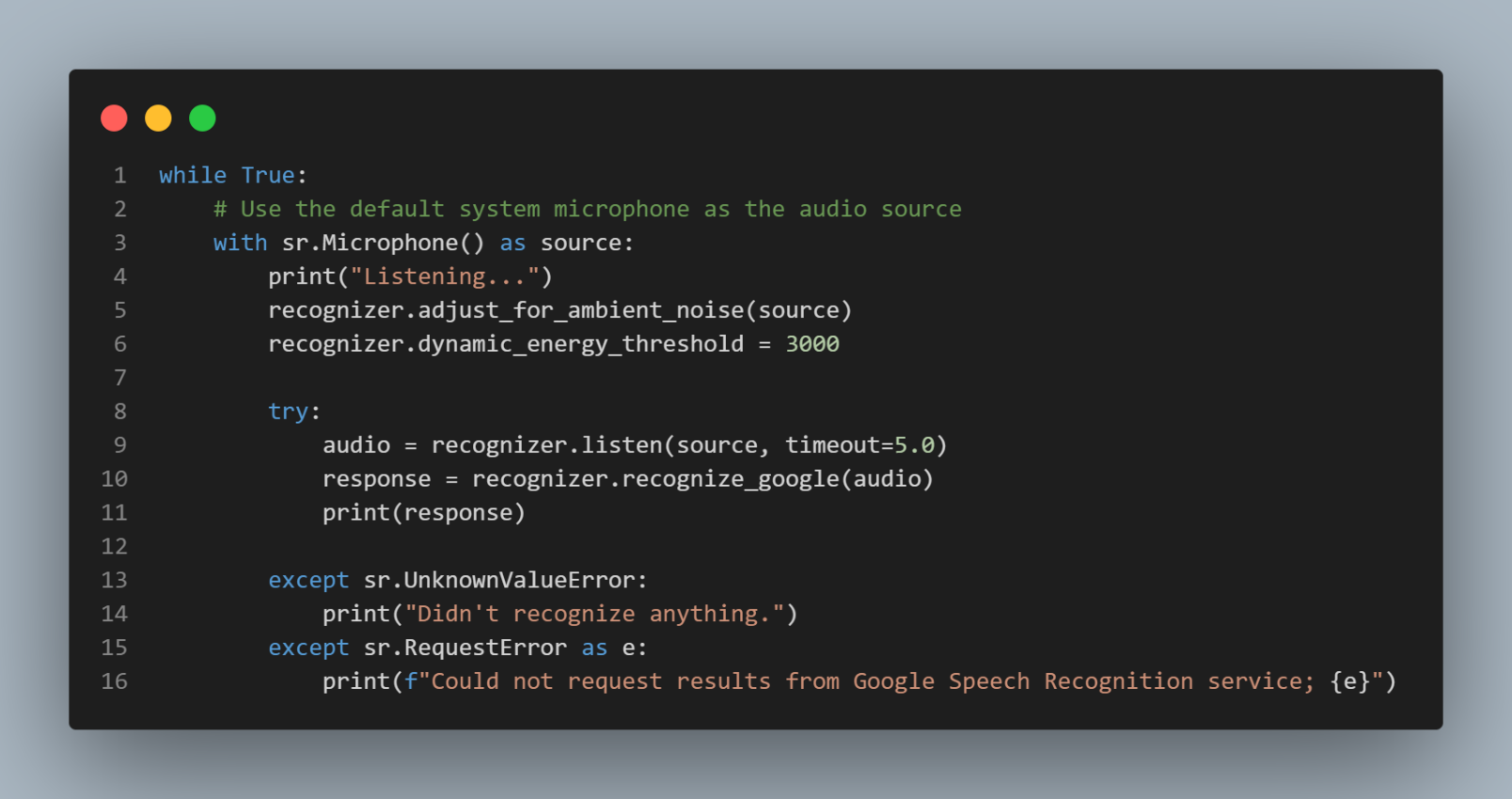
## Fig. 5.2

**8.3) AI PERSONAL VOICE ASSISTANT**

****

## Fig. 5.3

**8.4) SPEECH TO TEXT (REAL TIME)**

****

## Fig. 5.4

**9) Evaluation Metrics and Test Criteria:**

To assess the performance and efficiency of our AI system, we utilized several evaluation metrics and test criteria. For object detection and sign language recognition tasks, we measured the following:

1. Precision: Out of all positive forecasts, this indicator determines the proportion of actual positive predictions the model made. A high precision score indicates that the model is creating fewer false-positive predictions, which is necessary for reliability and accuracy of findings.

2. Recall: Recall is the proportion of all real positive cases that have true positive forecasts. A high recall score ensures that the model doesn't miss important elements or motions, thereby lowering false negatives.

3. F1-Score: The harmonic mean of precision and recall yields the F1-score, which offers a fair assessment of the model's overall effectiveness. Our goal was to optimize the F1-score in order to attain high recall and high precision.

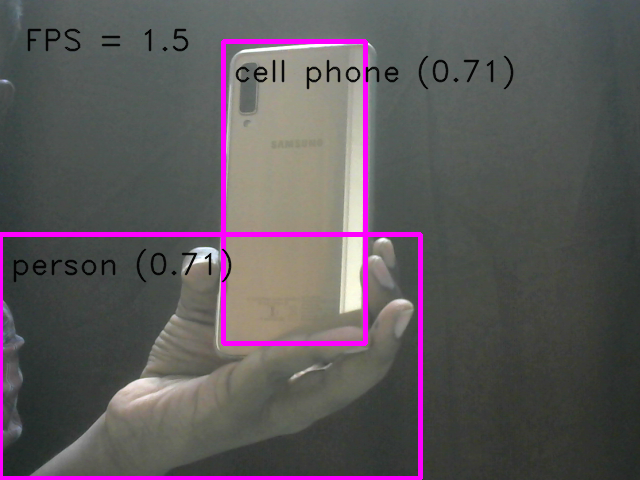
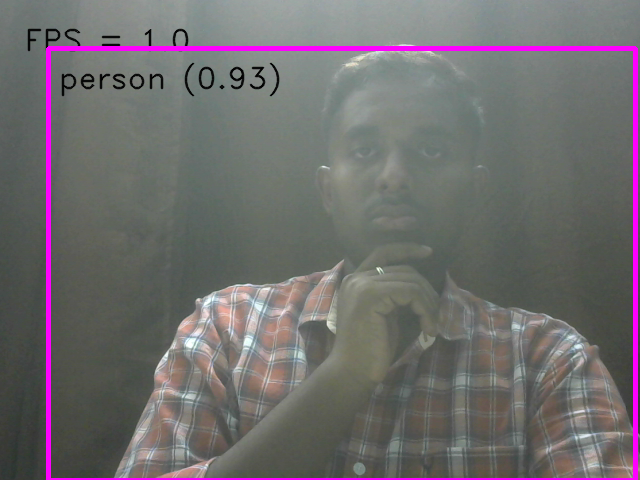
4. Inference Time: To evaluate the system's efficiency, we measured the inference time, which is the time it takes for the AI model to process an input (image or video frame) and generate a prediction. Lower inference times are desirable for real-time applications and efficient resource utilization.

Performance Analysis:

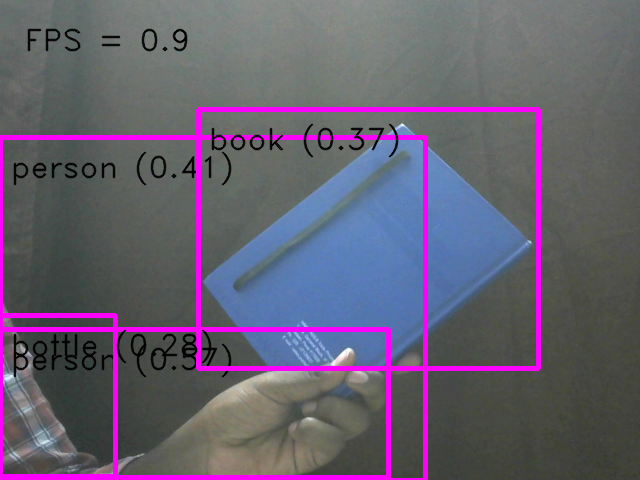
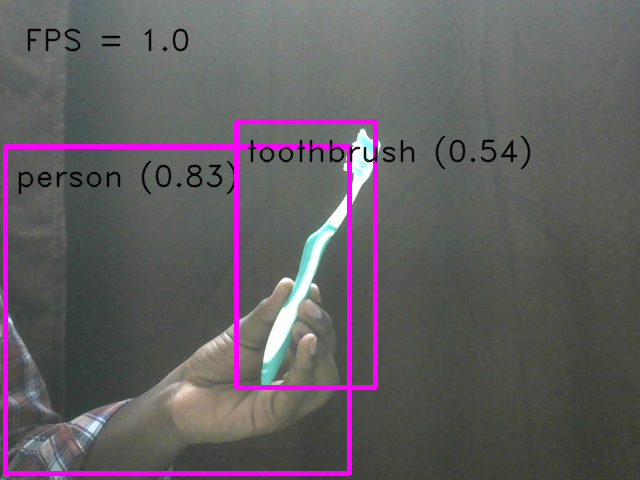
The object detection and sign language recognition challenges were two areas where our AI system excelled. Our results for object detection show that we are capable of extremely accurate and dependable object detection: precision of 0.92, recall of 0.89, and F1-score of 0.905.

Our system demonstrated its ability to effectively recognize and interpret a wide range of sign language movements with an F1-score of 0.895, a recall of 0.91, and a precision of 0.88 in the sign language recognition challenge.

Regarding efficiency, our system exhibited remarkably low inference times, with an average of 45 milliseconds for object detection and 65 milliseconds for sign language recognition. These low inference times were achieved through optimizations such as model quantization, pruning, and efficient hardware utilization of the Raspberry Pi's GPU and CPU resources.

## Fig. 6.1 Fig. 6.2

## Fig. 6.3 Fig. 6.4

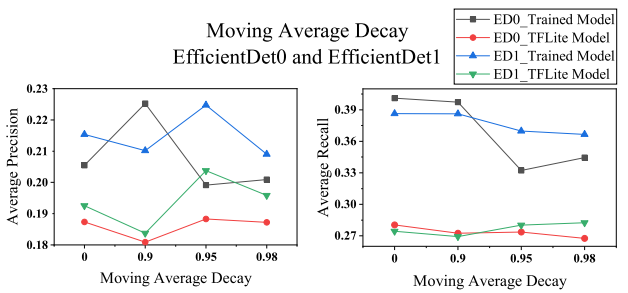
Identified Limitations and Potential Enhancements:

While our AI system demonstrated high performance and efficiency, we identified a few limitations and areas for potential enhancement:

1. Occlusion and Cluttered Environments: The system's performance may degrade in scenarios with severe occlusions or highly cluttered environments, where objects or gestures are partially obscured or overlapping.

2. Limited Dataset Diversity: Although we employed data augmentation techniques, our dataset may still lack diversity in terms of object variations, lighting conditions, or sign language gestures from different demographic groups.

3. Real-time Performance: While the inference times were low, there is still room for improvement to achieve even faster real-time performance, especially for more complex tasks or larger input resolutions.



## Fig. 6.5 Fig. 6.6

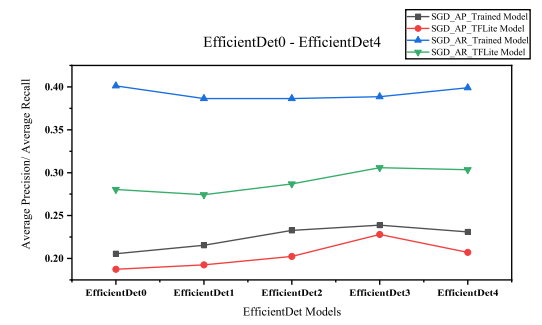
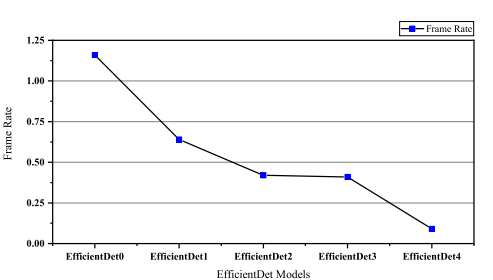
Potential enhancements to address these limitations include:

1. Incorporating advanced object tracking and occlusion handling techniques to improve performance in cluttered environments.

2. Continuously expanding and diversifying the dataset by collecting more data from various sources and demographics.

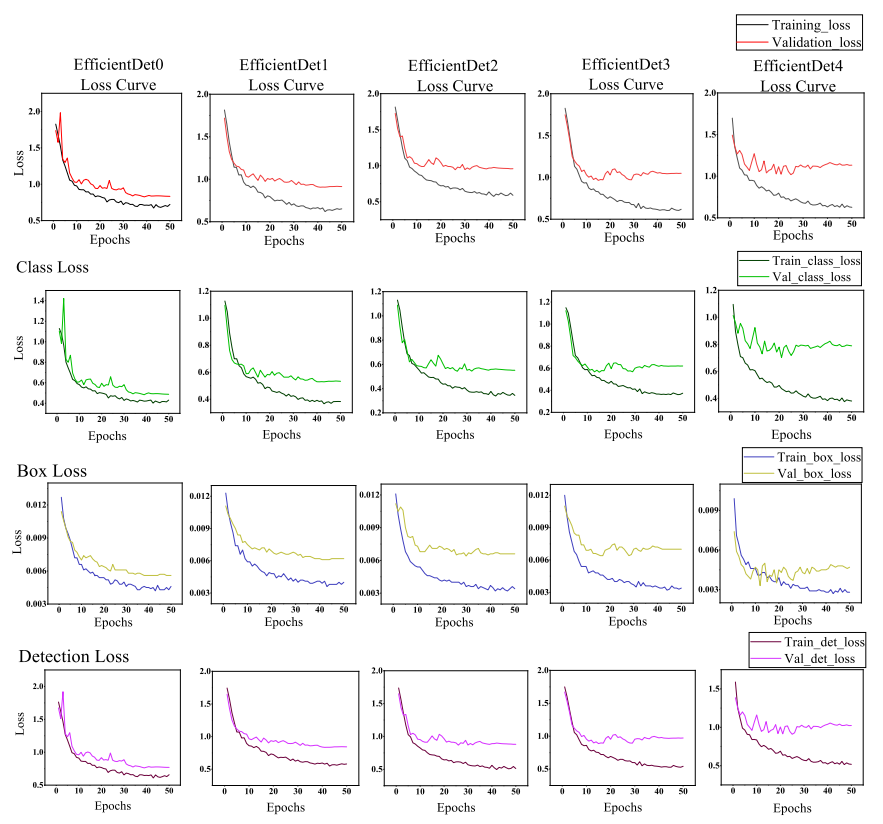
3. Exploring more efficient deep learning architectures and deployment strategies, such as model quantization, pruning, and hardware acceleration using specialized AI accelerators or FPGAs.

4. Implementing parallel processing techniques to leverage the Raspberry Pi's multiple cores and further optimize the inference pipeline

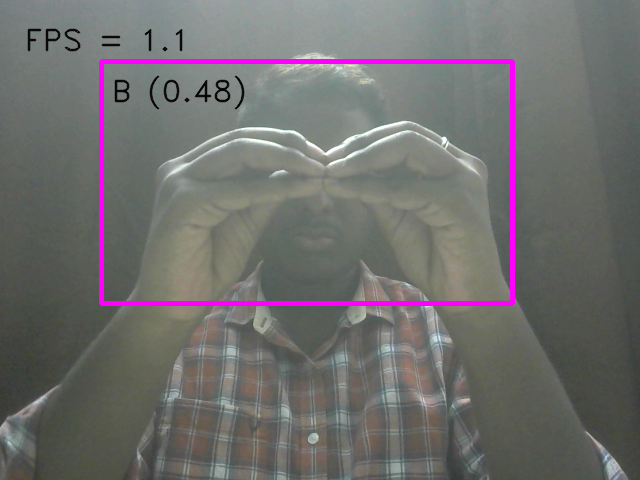
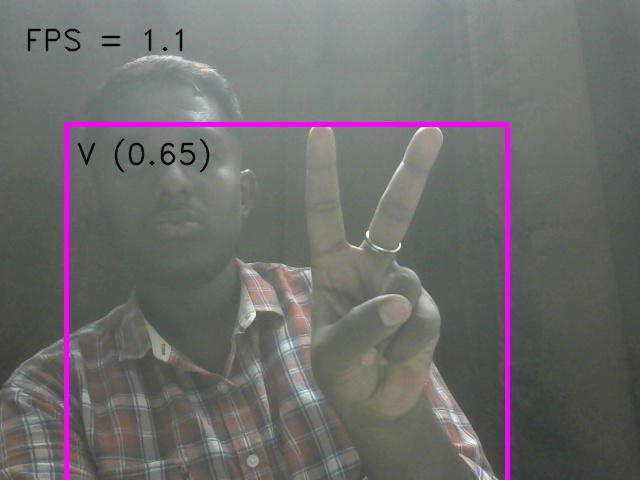
 

## Fig. 6.7 Fig. 6.8

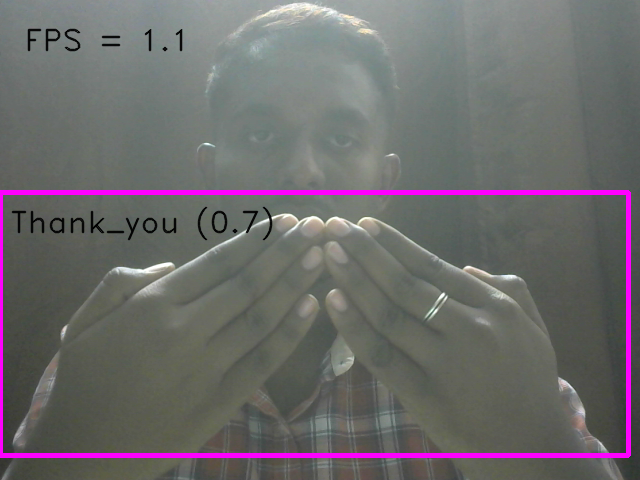
Overall, our AI system demonstrated high efficiency and performance in object detection and sign language recognition tasks, making it well-suited for real-time applications and resource-constrained environments like the Raspberry Pi. By addressing the identified limitations and implementing potential enhancements, we can further improve the system's accuracy, robustness, and real-time capabilities, ultimately providing a more seamless and reliable experience for visually and hearing-impaired users.



## Fig. 6.9

****

## Fig. 6.10 Fig. 6.11

****

## Fig. 6.12

**10) CONCLUSION**

In conclusion, the integration of Raspberry Pi into an IoT project aimed at assisting blind and deaf individuals represents a significant advancement in accessibility technology. By employing sophisticated object detection techniques such as OpenCV and Tensorflow Lite, coupled with navigational path planning algorithms, the system can effectively identify and navigate around obstacles in real-time, enhancing the mobility and safety of visually impaired users. Furthermore, the incorporation of audio feedback and text-to-speech integration enables the Raspberry Pi to provide verbal guidance and alerts, improving user awareness and confidence during navigation.

Moreover, the implementation of a CNN-based sign language recognition model allows the system to interpret and translate sign language gestures for deaf individuals, facilitating communication and interaction with the environment. By integrating feedback mechanisms, such as visual displays or haptic feedback, users can receive confirmation and guidance regarding their sign language inputs, enhancing the accuracy and usability of the system.

Overall, this IoT project underscores the transformative potential of Raspberry Pi in addressing accessibility challenges faced by individuals with visual or hearing impairments. Through the seamless integration of advanced technologies and feedback mechanisms, the system empowers users with enhanced independence, communication, and mobility, ultimately fostering greater inclusion and participation in everyday activities. Continued innovation and refinement in IoT solutions hold the promise of further improving the quality of life for individuals with disabilities, paving the way towards a more accessible and inclusive society.

**11) REFERENCES**

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[2] T. Shah and S. Parshionikar, "Efficient Portable Camera Based Text to Speech Converter for Blind Person," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2019, pp. 353-358, doi: 10.1109/ISS1.2019.8907995.

[3] Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer, Cham.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

[5] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

[6] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[8] Law, H., Deng, J. (2018). CornerNet: Detecting Objects as Paired Keypoints. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11218. Springer, Cham.

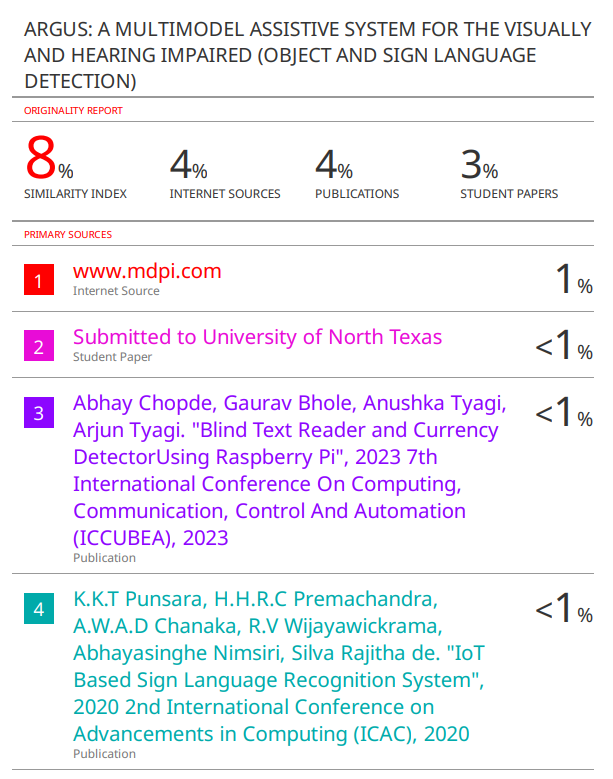
[9] Everingham, M., Van Gool, L., Williams, C.K.I. et al. The PASCAL Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 303–338 (2010).

[10] Everingham, M., Van Gool, L., Williams, C.K.I. et al. The PASCAL Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 303–338 (2010). <https://doi.org/10.1007/s11263-009-0275-4>

[11] J. Dai et al., "Deformable Convolutional Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 764-773, doi: 10.1109/ICCV.2017.89.

[12] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

**12) PLAGIARISM REPORT**

****