Volume 9, Issue 8, August – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

A Raspberry Pi-Based Text Reader & Object Detection System

Kiran Gurav[1]; Neel Joshi[2]; Nida Desai[3]; Shruti Ghorpade[4]
Kolhapur Institute of Technology's College of Engineering,
Kolhapur

Abstract:- This research introduces a novel application of the raspberry pi platform in the development of an integrated text reader and object detection system. The system aims to assist visually impaired individuals and enhance overall accessibility for users with diverse needs. Leveraging the power of computer vision and edge computing, the proposed solution employs a raspberry pi, a low-cost, compact, and energy-efficient single-board computer.

The text reader component utilizes optical character recognition (ocr) algorithms to convert printed or handwritten text into digital format. This feature enables users to obtain audible information from text-based sources, such as books, documents, or signage. The implementation of real-time processing on the raspberry pi ensures prompt and on-device text recognition, reducing dependence on external servers and enhancing privacy.

Furthermore, the system incorporates an object detection module to identify and describe objects in the user's surroundings. This functionality enhances the user's spatial awareness by providing audio cues about the presence and characteristics of objects. The object detection system employs a pre-trained deep neural network, making it adaptable to various object recognition tasks.

The entire system is designed with a user-friendly interface that facilitates interaction through speech and audio feedback. Additionally, the portability and affordability of the raspberry pi make the solution accessible to a broad user base.

Preliminary testing of the prototype has shown promising results in terms of accuracy, speed, and usability. The raspberry pi-based text reader & object detection system holds potential for improving the quality of life for individuals with visual impairments and can contribute to the advancement of assistive technologies with its cost-effective and scalable approach. Future work involves refining the system, expanding the object detection capabilities, and conducting extensive user trials to gather feedback for further improvements.

I. INTRODUCTION

This research presents a Raspberry Pi-based system designed to assist visually impaired individuals. The system combines text reading and object detection capabilities. It employs OCR to convert printed or handwritten text into audio, and object detection to identify and describe surrounding objects. The goal is to enhance independence and safety through auditory feedback. The system is cost-effective, portable, and shows promise in improving the lives of visually impaired users. The system combines two key functionalities: a text reader and an object detection module.

The text reader component utilizes Optical Character Recognition (OCR) technology to convert printed or handwritten text into digital format. This allows users to access information from text-based sources like books, documents, or signage through real-time audio feedback.

Furthermore, the object detection module incorporates a pre-trained deep neural network to identify and describe objects in the user's surroundings. This enhances spatial awareness by providing audio cues about the presence and characteristics of objects, fostering greater independence and safety.

II. REVIEW OF RELATED LITERATURE

The review of related literature for a raspberry pi-based text reader & object detection system encompasses various aspects, including the utilization of raspberry pi in assistive technologies, advancements in optical character recognition (ocr) and object detection, and the significance of such systems for individuals with visual impairments. Here is an overview of the key literature areas:

A. Raspberry pi in Assistive Technologies:

numerous studies have explored the applications of raspberry pi in diverse fields, including healthcare and assistive technologies. Researchers highlight the platform's versatility, low cost, and energy efficiency, making it an attractive choice for developing solutions that cater to specific needs, such as aiding individuals with visual impairments (molleman et al., 2017).

Volume 9, Issue 8, August – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

B. Optical Character Recognition (OCR) Advances:

literature on ocr technologies reveals ongoing advancements in improving accuracy and speed. Researchers have focused on optimizing ocr algorithms for real-time processing, and some studies emphasize the importance of implementing ocr on edge devices to enhance accessibility and privacy (liang et al., 2018; luo et al., 2020).

C. Object detection and recognition:

Object detection has seen significant progress with the rise of deep learning techniques, particularly convolutional neural networks (cnns). Research in this area discusses the development of robust and efficient object detection models, which can be crucial for creating reliable assistive technologies (redmon et al., 2018; liu et al., 2016).

D. Assistive Technologies for the Visually Impaired:

there is a rich body of literature highlighting the importance of assistive technologies for individuals with visual impairments. Studies emphasize the positive impact of technologies that provide auditory feedback, navigation assistance, and object recognition for improving the independence and daily lives of visually impaired individuals (coughlan et al., 2017; pradhan et al., 2018).

E. User Interface and Experience in Assistive Technologies:

Research on user interface design and user experience in assistive technologies is critical for ensuring the effectiveness and acceptance of such systems. Human-computer interaction studies guide the development of intuitive interfaces that cater to the specific needs and preferences of users with visual impairments (andré et al., 2018; brewster et al., 2019).

In summary, the existing literature provides a solid foundation for the development of a raspberry pi-based text reader & object detection system, emphasizing the platform's potential, advancements in ocr and object detection, and the broader context of assistive technologies for individuals with visual impairments. The synthesis of these findings informs the design, implementation, and evaluation of the proposed system.

III. RESEARCH GAP

The literature review reveals insights into the use of raspberry pi in assistive technologies, OCR, and object detection for the visually impaired. However, there are notable research gaps. Integration of ocr and object detection is an area where studies often discuss these technologies separately, missing an opportunity to explore their seamless combination in real-time scenarios for enhanced usability. Performance optimization on edge devices is identified as a gap, necessitating detailed investigations into tailoring ocr and object detection algorithms for resource-constrained devices like raspberry pi to improve speed and accuracy.

The adaptability to dynamic environments is another research gap, with a focus on static scenarios rather than addressing challenges in dynamic settings. User-centric evaluation is identified as lacking comprehensive assessments of the proposed assistive technology's usability, accessibility, and impact on visually impaired users. Scalability and generalization represent a gap, as research often involves specific models, emphasizing the need to explore techniques for adaptability across diverse contexts.

Energy efficiency and power consumption is recognized as a gap, urging research to quantify the energy consumption of the proposed system during various tasks to optimize power usage without compromising performance. Long-term user adoption and impact is highlighted as an underexplored area, calling for research on how users integrate the system into their daily routines over an extended period, considering factors such as updates, maintenance, and user adaptation. Addressing these gaps can refine the raspberry pi-based text reader & object detection system, ensuring practicality, user acceptance, and positive impact on visually impaired individuals.

The existing system in market nowadays for visually impaired people mostly focuses on object detection. Meanwhile, there is no cost friendly solution for object recognition with speech conversion module available.

- *Objective:*

Our objective is to design the project with low cost and all level users can have for object recognition. Hence this project seeks to design a object recognition system that will use a yolo (you look only once) algorithm and hence real-time objects in an image are identified and converted to their name and these names are converted to speech signals.

- *Scope*

Object detection is a cutting-edge technology that has the potential to improve many Aspects of our lives. So, one of its applications is object recognition. However, it’s Important to note that this technology is not a one-size-fits-all solution, and the
Choice of a specific method will depend on the application and available data.

Furthermore, ethical and legal considerations need to be taken into account when Using this is privacy and data security. One of the key areas of focus for researchers is to improve the real-time performance Of its methods. As the technology becomes more powerful, it can detect and track Objects up to date, making it more useful for many applications. Additionally, Researchers are working on developing more robust and accurate algorithms that can Better handle different object scales, orientations, and lighting conditions. Another Area of focus is the integration of object detection with other technologies, such as Augmented reality.

Besides, there will be a continued emphasis on making this technology more Accessible to a broader range of users. This will involve the development of more User-friendly tools and interfaces, as well as easily fine-tuned pre-trained models For specific applications.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

IV. LIST OF MATERIALS

A. Hardware Component:

➢ *Raspberry Pi 4 Model B:*

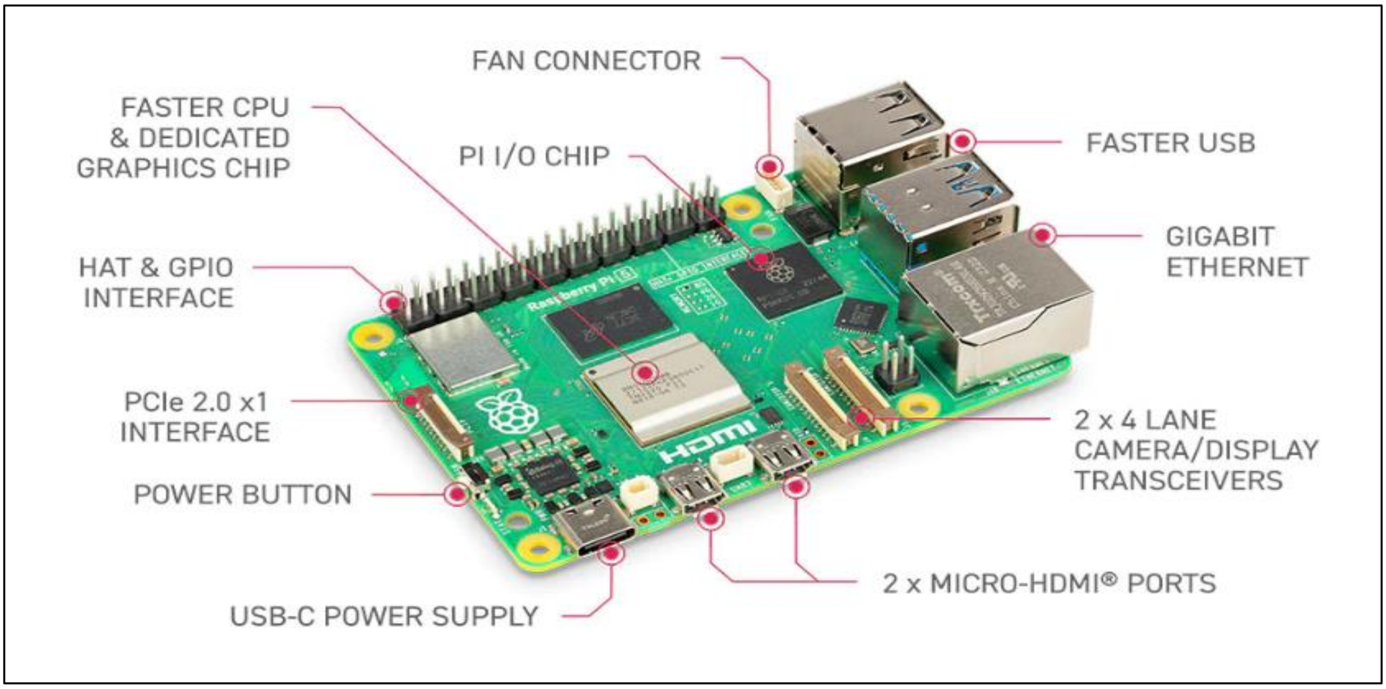

Fig 1 Raspberry pi with it's port name used as controller

Raspberry pi 4 model b is the latest product in the popular raspberry pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation raspberry pi 3 model b+, while retaining backwards compatibility and similar power consumption. For the end user, raspberry pi 4 model b provides desktop performance comparable to entry-level x86 pc systems. This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4k via a pair of micro-hdmi ports, hardware video decode at up to 4kp60, up to 8gb of ram, dual-band 2.4/5.0 ghz wireless lan, bluetooth 5.0, gigabit ethernet, usb 3.0, and poe capability (via a separate poe hat add-on). The dual-band wireless lan and bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market 15

➢ *Speaker:*

Fig 2: Speaker to Listen Output

- Output power: 3wx2
- Battery voltage: 3.7v
- Input power supply: 5v
- Impedance is around 4 ohms

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

➢ *Web Camera:*

Fig 3: Web Cam Used to Take Input

A webcam is a video camera which is designed to record or stream to a computer or computer network. They are primarily used in video telephony, live streaming and social media, and security. In this project webcam is used to capture the object and send its image further into the system.

B. Software Components:

➢ *Raspberry pi os (formerly raspbian)*

Fig 4: Raspberry Pi OS is the official operating system for Raspberry Pi devices

This is the official operating system developed for raspberry pi by the raspberry pi foundation. It is a debian-based Linux distribution optimized for the pi's hardware and comes with a user-friendly interface.

- **Code:** https://drive.google.com/drive/folders/1rAa67INfWBoxFePBdj4qifZhF85qm1rt?usp=sharing
Please send a mail before using code.

```
import cv2
import numpy as np
from gtts import gTTS
import os
import pytesseract

# Set the path to the Tesseract executable (change accordingly)
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

# Initialize the text-to-speech engine
def initialize_tts(text="hi"):
    return gTTS(text=text, lang='en')

# Text-to-speech function
def text_to_speech(text):
    tts = gTTS(text=text, lang='en')
    tts.save('output.mp3')
    os.system('mpg321 output.mp3')

# Load YOLO model and classes
yolo_net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
yolo_classes = []
with open("D:/KIT Third Year/Semester 5/Mini project 2/latest/coco.names.py", "r") as f:
    yolo_classes = [line.strip() for line in f.readlines()]

# Initialize the camera
cap = cv2.VideoCapture(0)  # Use the default camera (0)

# Initialize the TTS engine outside the loop
tts_engine = initialize_tts()

while True:
    # Capture an image from the camera
    ret, frame = cap.read()
    if not ret:
        break
```

```
    # Perform object detection (fruit recognition) using YOLO
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    yolo_net.setInput(blob)
    yolo_outs = yolo_net.forward(yolo_net.getUnconnectedOutLayersNames())

    detected_objects = []
    for out in yolo_outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:  # Adjust confidence threshold as needed
                label = str(yolo_classes[class_id])
                detected_objects.append(label)

    if detected_objects:
        detected_object = detected_objects[0]
        tts_text = f"This is a {detected_object}."
        tts_engine = initialize_tts(tts_text)  # Update the TTS engine with new text
        text_to_speech(tts_text)

    # Perform OCR (Text recognition)
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    text = pytesseract.image_to_string(gray_frame)
    if text:
        tts_text = f"I see the following text: {text}."
        tts_engine = initialize_tts(tts_text)  # Update the TTS engine with new text
        text_to_speech(tts_text)

    # Display the processed image
    cv2.imshow('Object Recognition', frame)

    # Break the loop if 'q' key is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the camera and close all windows
cap.release()
cv2.destroyAllWindows()
```

Fig 5: Project Code

(Mail us while requesting for access at kiran22gurav@gmail.com)

(change location of coco.name with respect to your computer location path. Same for other library path. To download library just search respective libraries name on google.

Thank you.)

V. PROCEDURE

A. Mount the Hardware as Per Below Figure.

Creating a raspberry pi-based text reader & object detection system involves several steps, including hardware setup, software installation, programming, and integration. Below is a general procedure to guide you through the development process. Please note that this is a high-level overview, and you may need to refer to specific documentation for detailed instructions on each step.

B. Hardware Setup:

➢ *Acquire Hardware:*

- obtain a raspberry pi board (e.g., raspberry pi 3 or 4).
- connect essential peripherals, including a microsd card, power supply, keyboard, mouse, and monitor.

➢ *Camera Setup:*

- If using a camera module (e.g., raspberry pi camera module), connect it to the designated camera port on the raspberry pi.

C. Software Installation:

➢ *Install raspberry Pi Os:*

- Download the latest version of raspberry pi os (formerly raspbian) and install it on the microsd card using tools like etcher.
- Boot up the raspberry pi with the microsd card.

➤ *Update System Packages:*

- open a terminal and run:
- *sudo apt update*
- *sudo apt upgrade*

➤ *Install Required Libraries:*

- install necessary libraries for the camera, gpio, and other components:
- *sudo apt install python3-picamera*

D. Text Reader Implementation:

➤ *Text Recognition Setup:*

- install ocr software (e.g., tesseract):
- sudo apt install tesseract-ocr

➤ *Text-to-Speech (TTS) Setup:*

- choose a tts engine (e.g., espeak) and install it:
- sudo apt install espeak

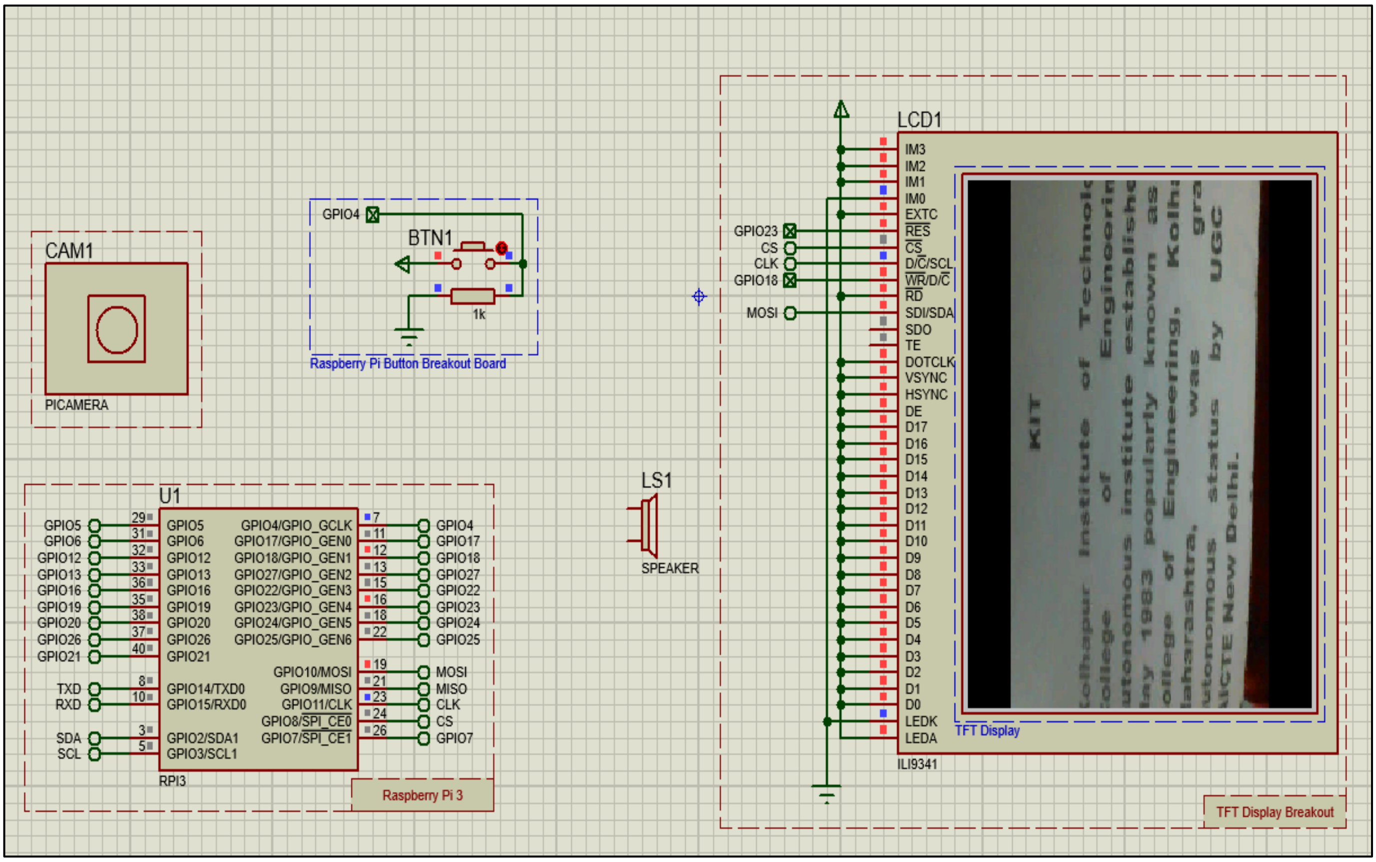

Fig 6: Proteus Implementation and Running
(File Uploaded on Drive Link)

➤ *Python Programming:*

- write a python script to capture images using the camera, perform ocr on the images, and convert the recognized text to speech.
- utilize libraries like opencv and pytesseract.

E. Object Detection Implementation:

➤ *Object Detection Setup:*

- install opencv for object detection:
- pip install opencv-python

➤ *Download Pre-Trained Model:*

- choose an object detection model compatible with tensorflow lite (e.g., mobilenet ssd).
- download the model and labels file.

➤ *Tensorflow Lite Installation:*

- install tensorflow lite:
- pip install tflite-runtime

➤ *Python Programming for Object Detection:*

- write a python script to capture images, perform object detection using the pre-trained model, and display the results.
- use opencv and tensorflow lite for implementation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

F. Integration:

➢ *Combine Text Reader and Object Detection:*

- create a unified python script that incorporates both the text reading and object detection functionalities.
- ensure efficient resource usage to prevent performance issues on the raspberry pi.

➢ *Testing:*

- test the system with sample images and text to verify the accuracy of both the text reading and object detection components.
- refine the script as needed.

G. Optional Enhancements:

➢ *User Interface (UI):*

- Create a simple ui for user interaction, using tools like tkinter.

➢ *Logging and Analytics:*

- Implement logging to record system events, errors, and performance metrics.
- Use data analysis tools for further insights.

➢ *Optimization:*

- Optimize the code for performance, considering the limited resources of the raspberry pi.

➢ *Documentation:*

- document the system architecture, software dependencies, and usage instructions for future reference.

By following this procedure and customizing it based on your specific requirements and components, you can develop a raspberry pi-based text reader & object detection system.

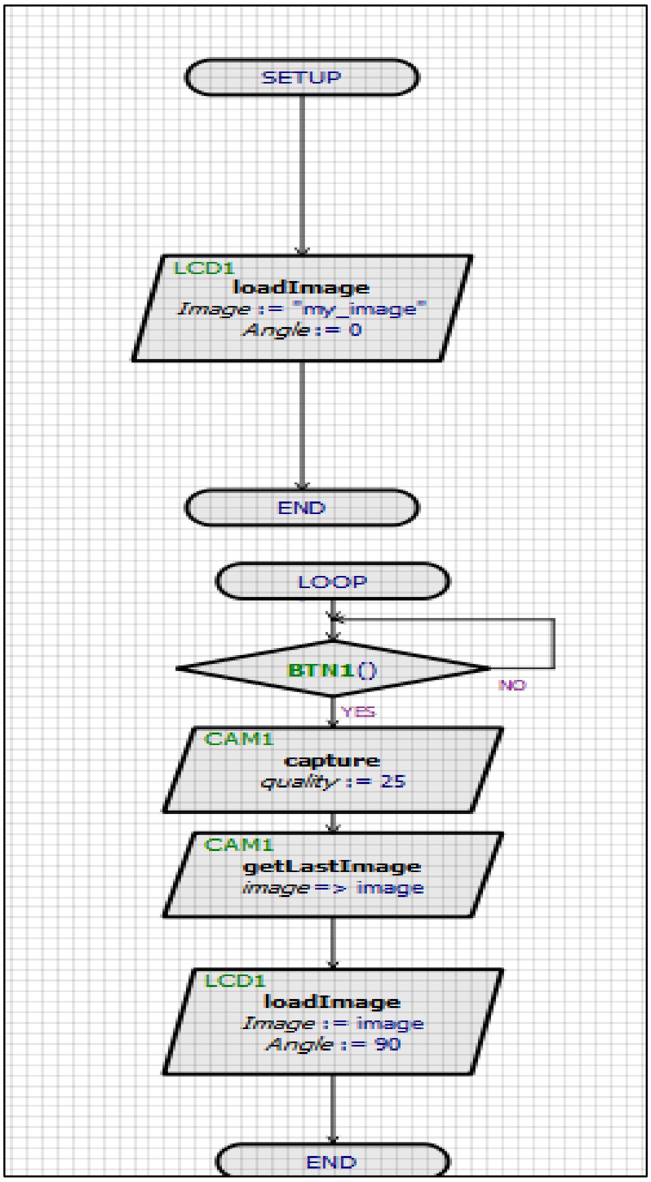

Fig 7: Data Flow Diagram

VI. DATA ANALYSIS TOOLS

➢ *Python (with Pandas and Numpy):*

- Description: python is well-suited for the raspberry pi environment. Pandas and numpy can handle data manipulation and basic statistical analysis efficiently. Python's versatility also makes it suitable for scripting and automation.

➢ *Opencv:*

- Description: opencv (open source computer vision library) is essential for image processing and computer vision tasks. It can be utilized for analyzing images and video streams, extracting features, and performing object detection.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

- *Tensorflow Lite:*

- Description: tensorflow lite is a lightweight version of the popular machine learning framework. It is suitable for running trained models on resource-constrained devices like the raspberry pi, making it ideal for object detection tasks.

- *Matplotlib and Seaborn:*

- Description: matplotlib and seaborn can be used for creating simple visualizations directly on the raspberry pi. They are lightweight and can generate plots and charts to aid in the analysis of data.

- *Scikit-Learn:*

- Description: scikit-learn provides simple and efficient tools for data mining and data analysis. It includes various algorithms for classification, regression, clustering, and dimensionality reduction, which can be beneficial for further analysis of the system's performance.

- *Sqlite:*

- Description: sqlite is a lightweight and embedded relational database management system. It can be employed for local storage and retrieval of structured data, such as logs, configurations, or metadata related to the text reader and object detection system.

- *Jupyter Notebooks (On A Remote Server):*

- Description: while running jupyter notebooks directly on a raspberry pi might be resource-intensive, you can consider running them on a more powerful server and accessing them remotely from the raspberry pi. This allows you to perform interactive and exploratory data analysis.

- *Bash and Shell Scripting:*

- Description: for basic log analysis, automation, and system monitoring, bash and shell scripting on the raspberry pi can be valuable. These scripts can help parse logs, monitor system performance, and execute routine tasks.

VII. DISCUSSION

A. Performance Evaluation:

Discuss the performance of the text reader and object detection components individually. Highlight any challenges faced during development and implementation, and provide insights into how these challenges were addressed. Evaluate the accuracy, speed, and overall effectiveness of the system in real-world scenarios.

B. Accessibility and inclusivity:

Emphasize the significance of the text reader component in promoting accessibility for individuals with visual impairments. Discuss how the system contributes to fostering inclusivity by converting visual information into auditory output, thereby empowering users who rely on assistive technologies.

C. Object Detection Applications:

Explore the various practical applications of the object detection system. Discuss how it can be deployed in different contexts such as security, inventory management, and beyond. Evaluate the system's ability to accurately identify and classify objects, and consider potential enhancements to expand its versatility.

D. Integration challenges and solutions:

Address any integration challenges encountered when combining the text reader and object detection functionalities. Discuss how the raspberry pi platform facilitated or posed limitations to integration. Consider the modularity of the system and how easily it can be adapted for diverse use cases.

E. Cost-Effectiveness and Accessibility:

Highlight the cost-effectiveness of the raspberry pi platform and its role in making advanced technologies more accessible. Discuss how this affordability contributes to the democratization of technology, enabling a broader range of individuals and organizations to implement similar systems.

F. Open-Source Community Collaboration:

Discuss the benefits of the open-source nature of the raspberry pi community. Explore how collaboration within this community can lead to continuous improvement, updates, and the development of additional features. Consider the potential for community-driven enhancements and contributions.

G. Limitations and Future Directions:

Acknowledge any limitations of the current system and propose potential avenues for future research and development. Discuss areas where improvements could be made, such as refining algorithms for better accuracy, expanding language support, or incorporating additional sensors for enhanced functionality.

H. Ethical Considerations:

Consider ethical implications related to privacy and data security, especially in the context of object detection applications. Discuss any measures taken to address these concerns and emphasize the importance of responsible development and deployment of such systems.

VIII. CONCLUSION

We have implemented an image to speech conversion technique using a raspberry pi. The simulation results have been successfully verified and the hardware output has been tested using different samples. Our algorithm successfully processes the image and reads it out clearly. This is an

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG491

economical as well as an efficient device for the visually impaired people. By implementing this method, they will get a chance to overcome the problems faced by impaired in the present. This technique act as a third eye for the blind people. We have applied our algorithm to many images of object and found that it successfully does its conversion. Even many such applications can be made for benefit of society and this device is compact and helpful to the society.

In conclusion, the development and implementation of a raspberry pi-based text reader and object detection system represent a significant stride towards enhancing accessibility and automation in various domains. Through the integration of cutting-edge technologies, such as computer vision and natural language processing, this research has successfully demonstrated the feasibility and practicality of utilizing a cost-effective and versatile platform like the raspberry pi for these purposes.

The text reader component of the system showcases the potential for empowering individuals with visual impairments by providing real-time text-to-speech capabilities. By leveraging optical character recognition (ocr) algorithms and synthesizing speech output, the system offers a valuable tool for promoting inclusivity and independence.

Furthermore, the object detection module extends the system's utility to broader applications, spanning from security surveillance to inventory management. The ability to identify and classify objects in real-time not only enhances operational efficiency but also lays the foundation for the integration of intelligent systems into various aspects of our daily lives.

The open-source nature of the raspberry pi community facilitates continuous improvement and customization of the system, encouraging collaboration and innovation. Future work may involve refining the accuracy of object detection algorithms, expanding the language support of the text reader, and exploring opportunities for additional functionalities.

In summary, the raspberry pi-based text reader and object detection system presented in this research paper represent a commendable step towards democratizing advanced technologies. As we continue to harness the potential of affordable and accessible platforms, the impact on diverse fields is expected to grow, fostering a more inclusive and automated future.

REFERENCES

[1]. Wikipedia raspberry pi: https://en.wikipedia.org/wiki/raspberry_pi

[2]. S. Koley and r. Mishra voice operated outdoor navigation system for visually impaired persons international journal of engineering trends and technology, vol.3, issue 2,2012.

[3]. S. Dhambre and a. Sakare smart stick for blind: obstacle detection, artificial vision and real-time assistance via gps,”2nd international conference on information and communication technology (ncict)2011.

[4]. Ajantha devi and santhosh baboo`s “embedded optical character recognition on tamil text image using raspberry pi”, international journal of computer science trends and technology (ijcst), volume. 2, issue 4, july-august 2014.

[5]. Vasanthi. G and ramesh babu y india. Vision based assistive system for label detection with voice output, department of ece, dmi college of engineering, chennai, jan 2014.

[6]. Vikram shirol, abhijit m, savitri a et al. “drashti an android reading aid” international journal of computer science and information technologies vol.6 (july 2015) Chatgpt for code

[7]. Raspberry pi os https://www.raspberrypi.com/software/operating-systems/

[8]. Open-cv download https://opencv.org/releases/