**Pi vision - Reader for blind using Raspberry Pi**

**TERM PROJECT REPORT**

*Submitted by*

CB.EN.U4CCE21031 KAVIN VELAVAN

CB.EN.U4CCE21042 PALANIAPPAN AN

CB.EN.U4CCE21048 PUBESH KUMAR K S

CB.EN.U4CCE21077 V KRISHNA MOORTHY



**B.TECH COMPUTER AND COMMUNICATION ENGINEERING**

**2021 BATCH**

**ABSTRACT:**

This report presents the development of an application utilizing the Raspberry Pi Camera (PiCamera) and OpenCV to enable real-time frame capture and display. The primary functionality of the application involves initiating text recognition on captured images. The system employs Tesseract OCR, a powerful optical character recognition engine, to process the images and extract textual information. The extracted text is then converted into audible output using the pyttsx3 library, facilitating a seamless auditory experience for users.

The application initializes the PiCamera with specified parameters, including resolution and framerate, to capture high-quality frames in real-time. These frames are continuously displayed using OpenCV, creating a live video feed. Users can interact with the system by triggering text recognition on demand.

Upon user initiation, the system captures the current frame and submits it to the Tesseract OCR engine for text extraction. The recognized text is subsequently converted into speech using the pyttsx3 library, providing an accessible and user-friendly auditory output. Additionally, the extracted text is displayed on the console for visual confirmation.

The application is designed to be user-friendly, with simple keypress interactions. Users can initiate text recognition by pressing the designated key ('s'). Furthermore, an exit mechanism is implemented, allowing users to terminate the application gracefully by pressing the space bar.

Overall, this system combines the capabilities of the Raspberry Pi Camera, OpenCV, Tesseract OCR, and pyttsx3 to create a versatile and accessible solution for real-time text recognition, demonstrating the potential of integrating hardware components with powerful software libraries for practical applications.

**CODE:**

import cv2

import pytesseract

import pyttsx3

from picamera.array import PiRGBArray

from picamera import PiCamera

# Initialize the text-to-speech engine

engine = pyttsx3.init()

# Set camera parameters

camera = PiCamera()

camera.resolution = (640, 480)

camera.framerate = 30

rawCapture = PiRGBArray(camera, size=(640, 480))

engine.say("Pi Vision started. Press 's' to capture and recognize text.")

engine.runAndWait()

continuous\_recognition = True

while continuous\_recognition:

    for frame in camera.capture\_continuous(rawCapture, format="bgr", use\_video\_port=True):

        image = frame.array

        cv2.imshow("Frame", image)

        key = cv2.waitKey(1) & 0xFF

        rawCapture.truncate(0)

        if key == ord("s"):

            engine.say("Capturing image. Please wait for text recognition.")

            engine.runAndWait()

            text = pytesseract.image\_to\_string(image)

            if text:

                engine.say("Text recognized. The text is:")

                engine.say(text)

                engine.runAndWait()

print(“The recogonized text is:”,\n)

        print(text)

            else:

                engine.say("Text recognition failed. Please try again.")

                engine.runAndWait()

            cv2.imshow("Frame", image)

        elif key ==ord(" "):

            continuous\_recognition = False

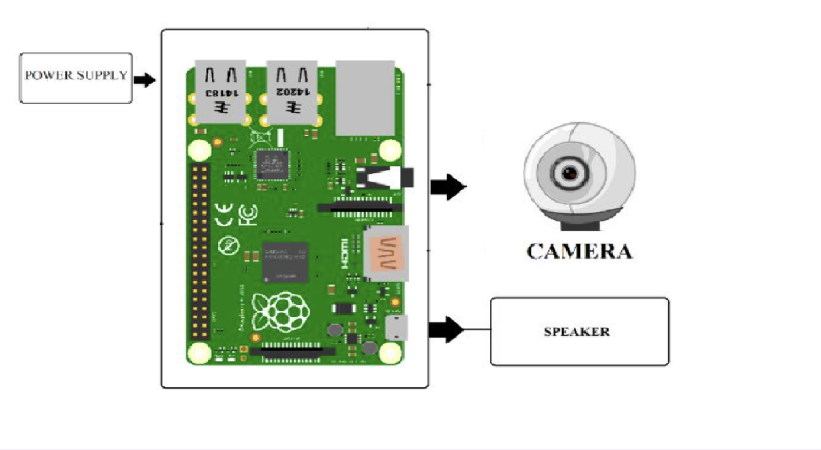
            break

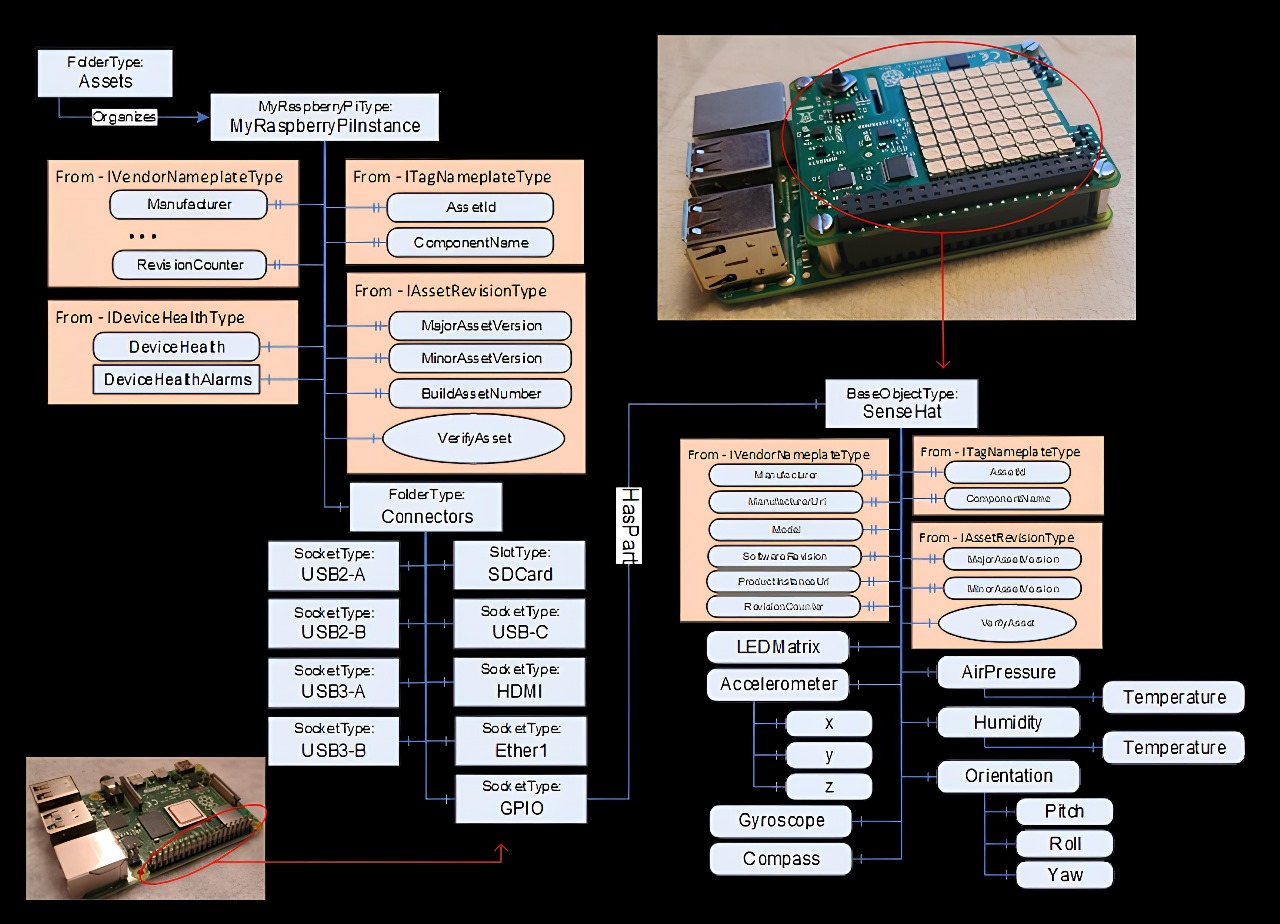
cv2.destroyAllWindows()

engine.say("Pi Vision exiting. Thank you for using.")

engine.runAndWait()

**CIRCUIT DIAGRAM:**

****



**WORKING PRINCIPLE:**

**Algorithm: Real-Time Text Recognition with Raspberry Pi Camera**

**Library Import:**

1. Import necessary libraries:

- OpenCV (cv2)

- Tesseract OCR (pytesseract)

- Text-to-speech (pyttsx3)

- Raspberry Pi Camera components (PiRGBArray and PiCamera)

**Initialize Text-to-Speech Engine:**

2. Initialize the text-to-speech engine using `pyttsx3.init()`.

**Set Camera Parameters:**

3. Initialize the PiCamera:

- Set resolution and framerate.

**Initialize PiRGBArray:**

4. Set up PiRGBArray for capturing frames.

**Notify User:**

5. Notify the user via text-to-speech that the application has started.

**Continuous Recognition Loop:**

6. Enter a continuous loop to capture frames from the camera.

- Display each frame using OpenCV.

**Key Press Handling:**

7. Wait for a key press.

- If 's' is pressed:

- Notify the user about image capture and text recognition.

- Use Tesseract OCR to recognize text from the captured image.

- Read aloud the recognized text using text-to-speech.

- Display the recognized text on the console.

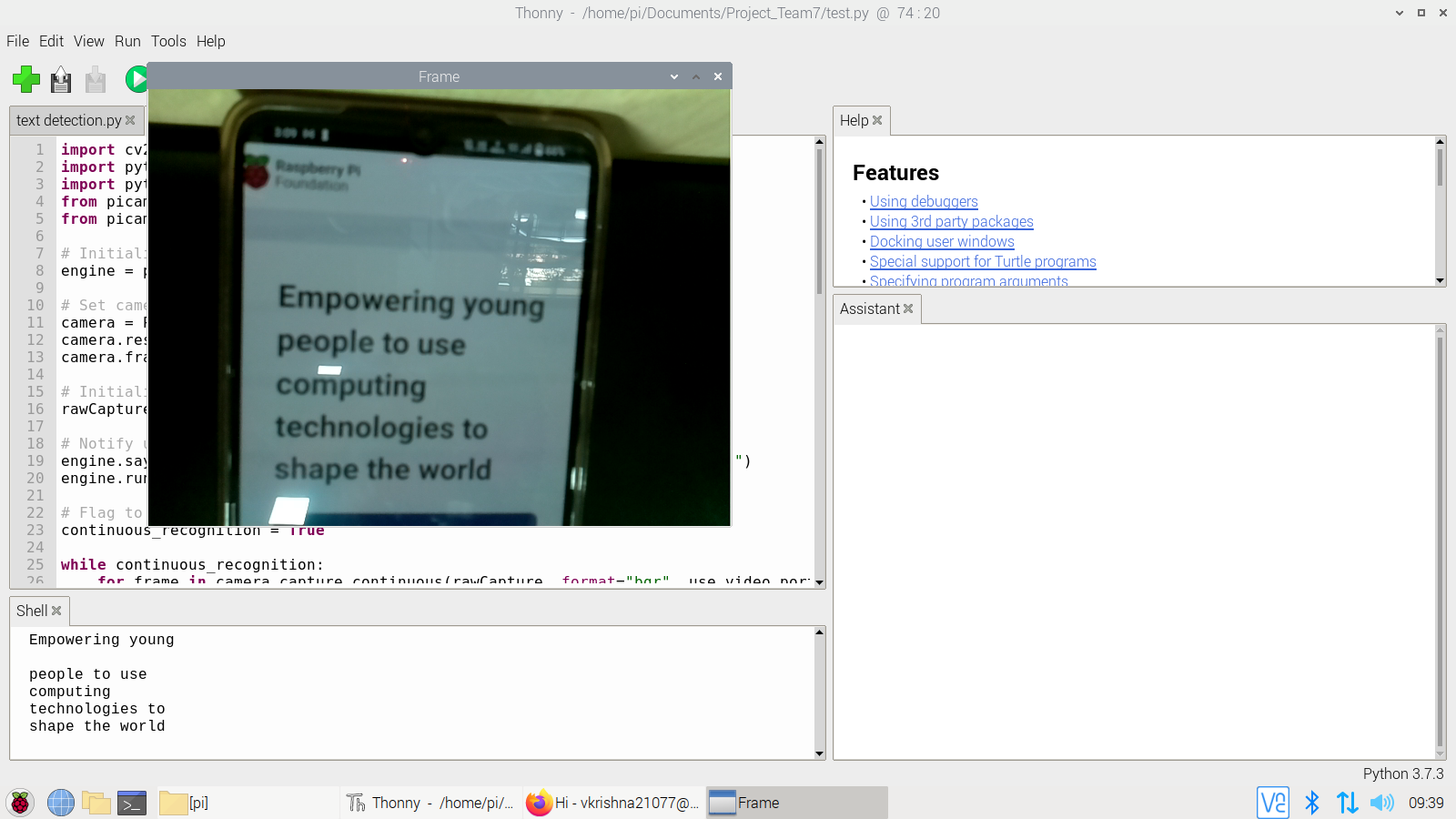
**Exit Handling:**

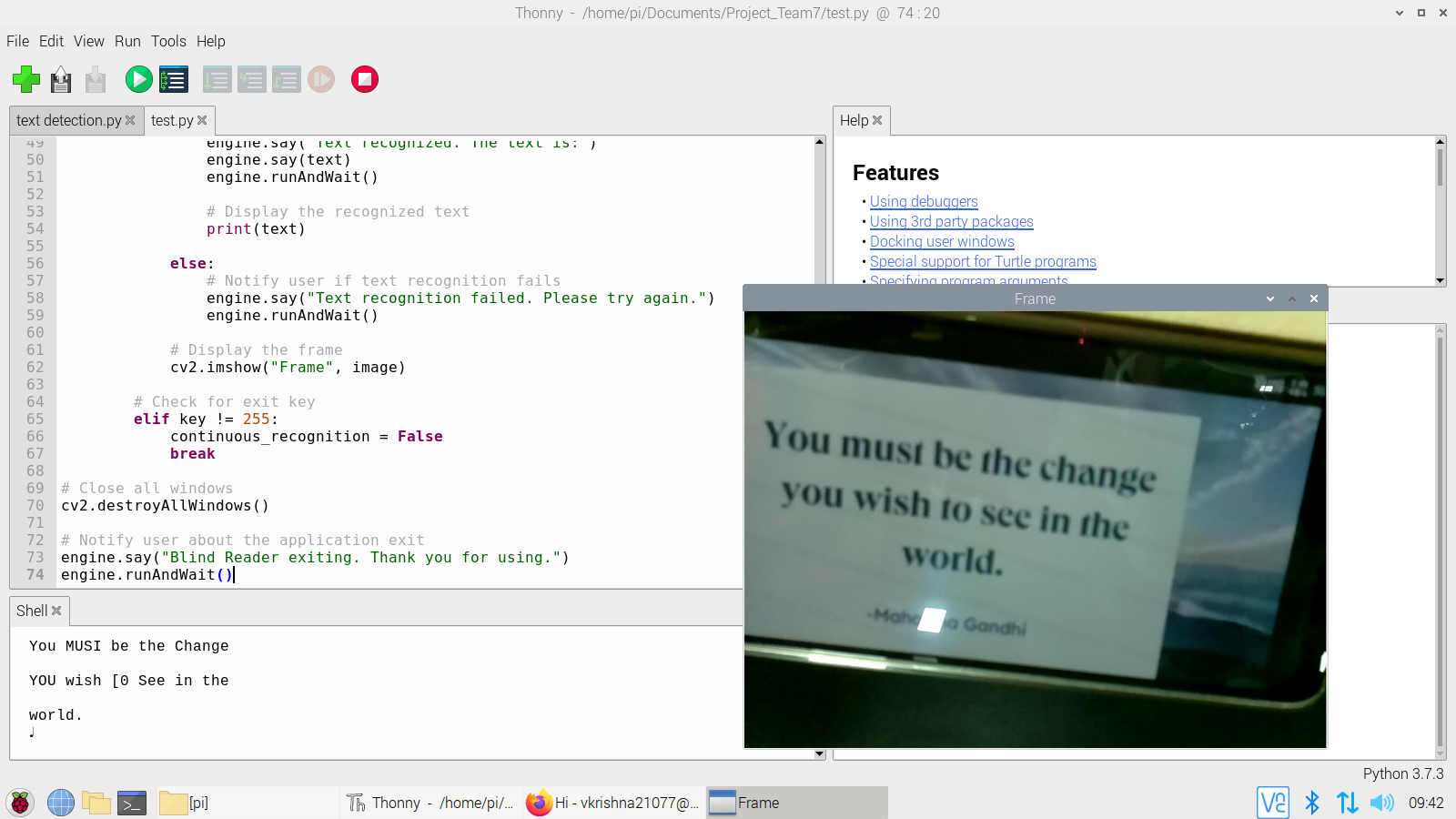
8. If the space bar is pressed:

- Break out of the continuous loop.

- Close all OpenCV windows, ending the real-time display.

**OUTPUT:**





**CONCLUSION:**

In conclusion, the developed real-time text recognition application effectively integrates Raspberry Pi Camera, OpenCV, Tesseract OCR, and pyttsx3. The user-friendly design allows easy initiation of text recognition, providing audible output. The simplicity, coupled with potential applications in accessibility and automation, highlights the practicality of such systems. As technology evolves, further enhancements and integrations can be explored to maximize the system's functionality and usability. Overall, the application demonstrates the successful convergence of hardware and software for practical, real-time solutions.