INTERNET OF THINGS

TEAM - 08

TEAM MEMBERS	ROLL NO
ASHWANTHRAM A C	CB.EN.U4CCE21009
GAUTAM R S	CB.EN.U4CCE21020
DEENADAYALAN S	CB.EN.U4CCE21013
AKAASHABIMANYU	CB.EN.U4CCE21004

Amrita School Of Engineering

Amrita Vishwa Vidhyapeetham

Coimbatore-641112

DECEMBER- 2023

SMART GLASS FOR VISUALLY CHALLENGED PEOPLE

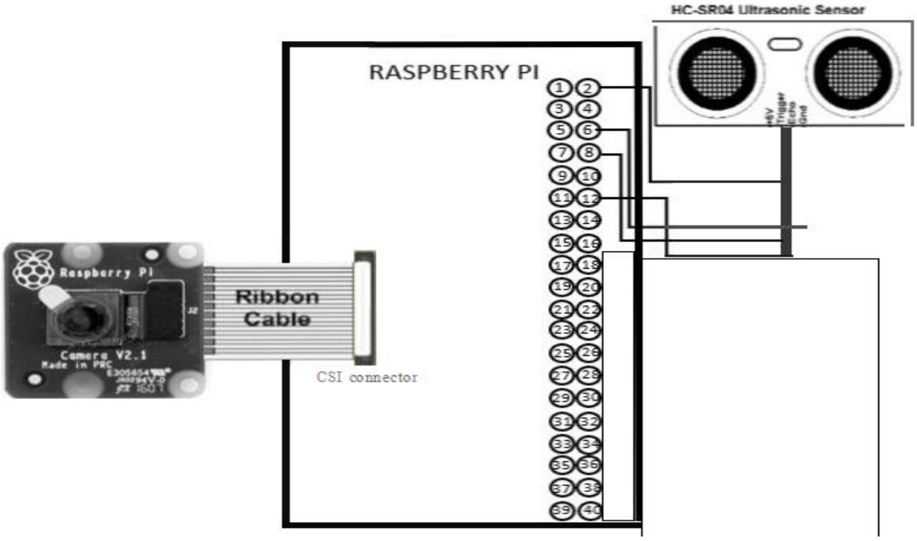
Abstract:

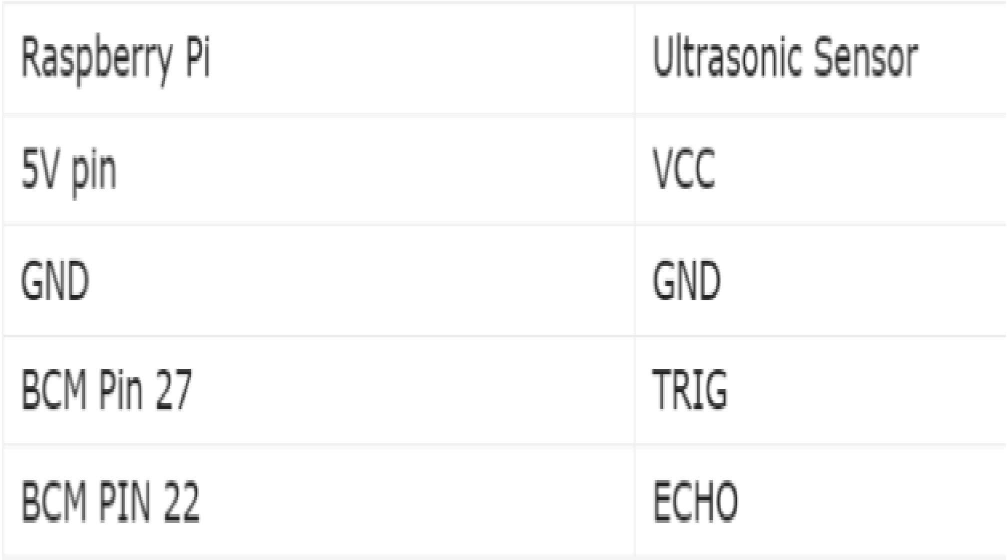
Introducing Facial Recognition Smart Glasses for the visually challenged—a cutting-edge solution blending advanced facial recognition technology with real-time audio feedback. These glasses empower users by providing instant, discreet verbal descriptions of individuals in their vicinity, fostering independence and social inclusive. With a user-friendly interface, stylish design, and cloud connectivity, this innovation marks a significant leap towards enhancing accessibility and autonomy for the visually impaired.

Code:

```
import picamera
import time
import cv2
import RPi.GPIO as GPIO
import pyttsx3
import numpy as np


def detect_face(frame):
    face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 
'haarcascade_frontalface_default.xml')
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)
    return faces


def measure_distance(trigger_pin, echo_pin):
    GPIO.output(trigger_pin, True)
    time.sleep(0.00001)
    GPIO.output(trigger_pin, False)
    pulse_start = time.time()
    pulse_end = time.time()

    while GPIO.input(echo_pin) == 0:
        pulse_start = time.time()

    while GPIO.input(echo_pin) == 1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150
    distance = round(distance, 2)

    return distance

def speak(text):
    engine = pyttsx3.init()
    engine.say(text)
    engine.runAndWait()

# Set up GPIO
GPIO.setmode(GPIO.BCM)
TRIG = 2
ECHO = 3
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.output(TRIG, GPIO.LOW)

picam = picamera.PiCamera()

try:
    picam.rotation = 180
    picam.resolution = (640, 480)
    picam.framerate = 30

    ultrasonic_active = False

    with picam as camera:
        raw_capture = np.empty((640 * 480 * 3,), dtype=np.uint8)


        camera.start_preview(alpha=200)
        time.sleep(2)

        while True:
            camera.capture(raw_capture, format="bgr", use_video_port=True)
            frame = raw_capture.reshape((480, 640, 3))

            faces = detect_face(frame)
            print("detected faces",len(faces))

            if len(faces) > 0:
                if not ultrasonic_active:
                    ultrasonic_active = True
                    print("Ultrasonic sensor activated.")

                distance = measure_distance(TRIG, ECHO)

                if distance <= 300:
                    print(f"human detected within {distance} centimeters. Please be cautious(voice
command).")
                    speak(f"human detected within {distance} centimeters. Please be cautious.")
                else:
                    print(f"human detected beyond 3 meters. No voice command.")
            else:
                if ultrasonic_active:
                    ultrasonic_active = False
                    print("Ultrasonic sensor deactivated.")

                distance = measure_distance(TRIG, ECHO)

                if distance <= 300:
                    print(f"Obstacle blocking within {distance} centimeters. Please be
cautious(voice command).")
                    speak(f"Obstacle blocking within {distance} centimeters. Please be cautious.")
                else:
                    print(f"No obstacle detected beyond 3 meters.")

except KeyboardInterrupt:
    pass

finally:
    picam.stop_preview()
    picam.close()
    GPIO.cleanup()
```

Circuit Diagram:





Raspberry Pi	Ultrasonic Sensor
5V pin	VCC
GND	GND
BCM Pin 27	TRIG
BCM PIN 22	ECHO

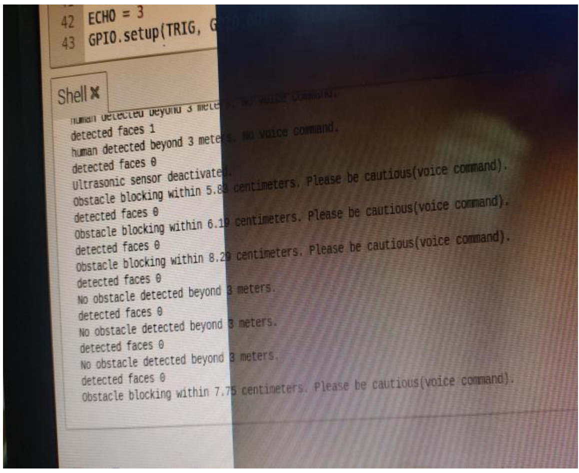
Working Principle:

The Facial Recognition Smart Glasses operate by capturing facial features through a discreet camera, utilizing advanced algorithms to identify individuals from a per-loaded database.

Upon recognition, real-time audio feedback delivers concise verbal descriptions to the user.

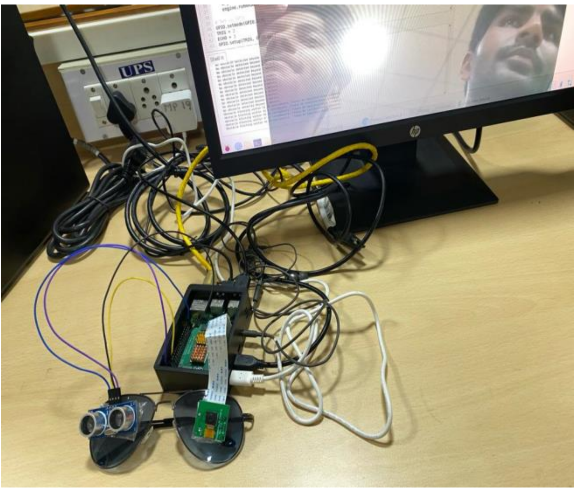
The device, with a user-friendly interface and stylish design, aims to enhance accessibility and independence for visually challenged individuals by fostering immediate and discreet communication in social interactions. Cloud connectivity ensures continuous updates to the recognition database, improving accuracy over time.

Output:



```
ECHO = 3
GPIO.setup(TRIG, G
```

```
human detected beyond 3 meters. No voice command.
detected faces 1
human detected beyond 3 meters. No voice command.
detected faces 0
Ultrasonic sensor deactivated.
Obstacle blocking within 5.83 centimeters. Please be cautious(voice command).
detected faces 0
Obstacle blocking within 6.19 centimeters. Please be cautious(voice command).
detected faces 0
Obstacle blocking within 8.29 centimeters. Please be cautious(voice command).
detected faces 0
No obstacle detected beyond 3 meters.
detected faces 0
No obstacle detected beyond 3 meters.
detected faces 0
No obstacle detected beyond 3 meters.
detected faces 0
Obstacle blocking within 7.75 centimeters. Please be cautious(voice command).
```



Result:

It's incredible how technology is being harnessed to enhance the lives of visually challenged individuals, empowering them to navigate the world more independently.Thus Successfully implemented the Smart glass for visually challenged people.