SMART GLASSES FOR BLIND PEOPLE USING RASPBERRY PI PICO

Introduction

The glasses with an obstacle detecting module in the centre, a processing unit and a power supply make up this device. The processing unit is coupled to the obstacle detecting module and the output device. The central processing unit receives power from the power supply. A ultrasonic sensor serves as the obstacle detection module, while a control module serves as the processing unit. The control unit activates the ultrasonic sensors, which gather information about the barrier or object in front of the person, analyses it, and then sends obtained data through serial connection. After sending data, in the form of voice letting the user know if obstacle is near or far away.The main feature is Indian currency detection , enabled in the smart glasses.where camera on the smart glasses detect the Indian currency notes and sends the detected currency as a voice to the person.

1. Components

1.1 Hardware

- Glasses
- Ultra sonic sensors
- Jumper wires
- W5100S-EVB-Pico
- Raspberry Pi 4
- Speaker
- Power Bank

1.2 Software

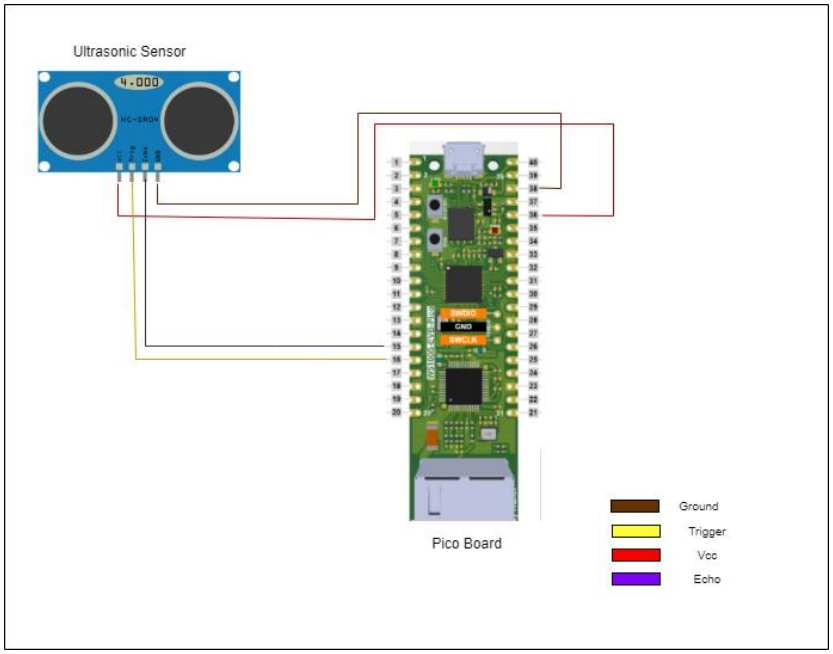
- Yolov5 Object detection
- Arduino IDE
- Google Colab
- Thonny - Python IDE
- Raspberry Pi IDE

2.Component Description

Device	Functions
Glass 	The glass used to setup the components Pico board,camera and ultrasonic sensors.
Ultrasonic Sensor 	The ultrasonic sensor is used to detect the distance between the objects.
USB Cam 	The USB cam is used to capture the currency for blind people..
Jumper Wire 	The jumper wires are used to connect the components.

Speaker 	The speaker is used to provide which note has been detected and voice recognition of distance of object in front.
Raspberry Pi 4 	Raspberry Pi 400 is your complete personal computer, built into a compact keyboard. Featuring a quad-core 64-bit processor, 4GB of RAM, wireless networking, dual-display output, and 4K video playback, as well as a 40-pin GPIO header, it's the most powerful and easy-to-use Raspberry Pi computer yet.
Power bank 	The power bank is used to provide power supply to Raspberry Pi 4.
W5100S-EVB-Pico 	The W5100S-EVB-Pico is micro controller evaluation board based on the Raspberry Pi RP2040 micro controller chip and full hardwired TCP/IP controller W5100S chip.

3.Circuit Diagram

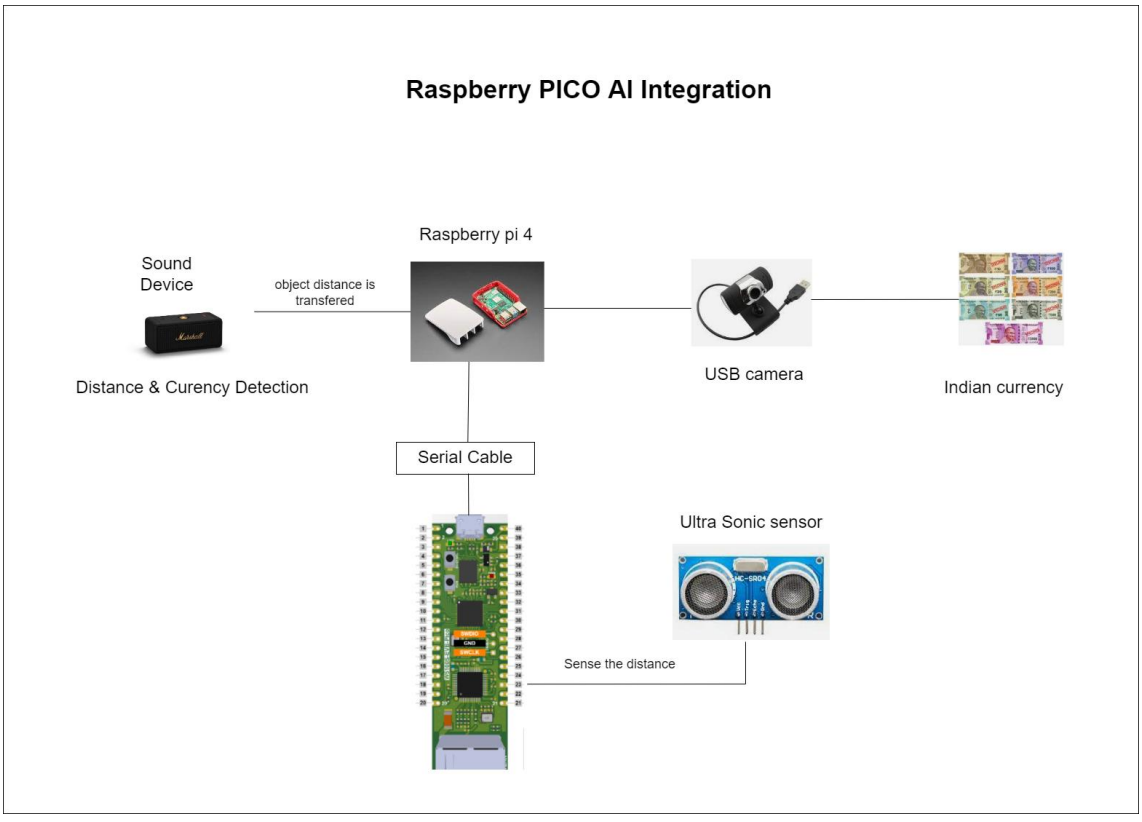


3.1 Pin Configuration

Ultrasonic sensor (HC-SR04) consists of 4 pins.

- GND - It is connected to GND of Pico Board.
- Trig - This pin is connected to GP12.
- Echo - Its connected to pin is connected to GP11.
- VCC - Its connected to 3V3(OUT).

4. AI diagram



4.1 AI implementation

4.1.1 Voice Module:

We have used Python eSpeak Module. eSpeak is a small open source software speech synthesizer for Linux and Windows that supports English and other languages. eSpeak employs a technique known as "format synthesis." This enables a large number of languages to be supplied in a little space. It basically converts text to speech format. There are several Voices to choose from, each with its own set of attributes that can be changed. In this Project we are using eSpeak for two purpose convert the distance calculated using ultrasonic sensor to speech. It will let user know how safer are they to move forward. The other purpose is currency detection. When any 100, 200 , 500 etc notes is bought in front of camera. Detected note will be speak out loud using eSpeak module.

4.1.2 Camera Module

The image is trained with yolov5 library using the mechanism of annotation. The Indian currencies are annotated using Roboflow software for the image training. Training is done by collecting Indian currencies. We then annotate these pictures differentiating between new Indian notes. Then feed it to the training algorithm. Ensure that the present working directory is the YOLOV5 directory and all dependencies are installed. The coco128.yaml file has to be updated with the dataset directory and also the classes used are classification of Indian currencies.Train using the command:

!python train.py --img 416 --batch 16 --epochs 100 --data coco128.yaml --weights yolov5s.pt --cache

A pertained weight has been assigned to the training [yolov5s.pt] and is running at 100 epochs. Once the training is done check in the “runs” folder to check the accuracy of predictions in the .jpg files. The real time detection can be used with the command :

python detect.py --weights best.pt --img 416 --conf 0.5 --source 0

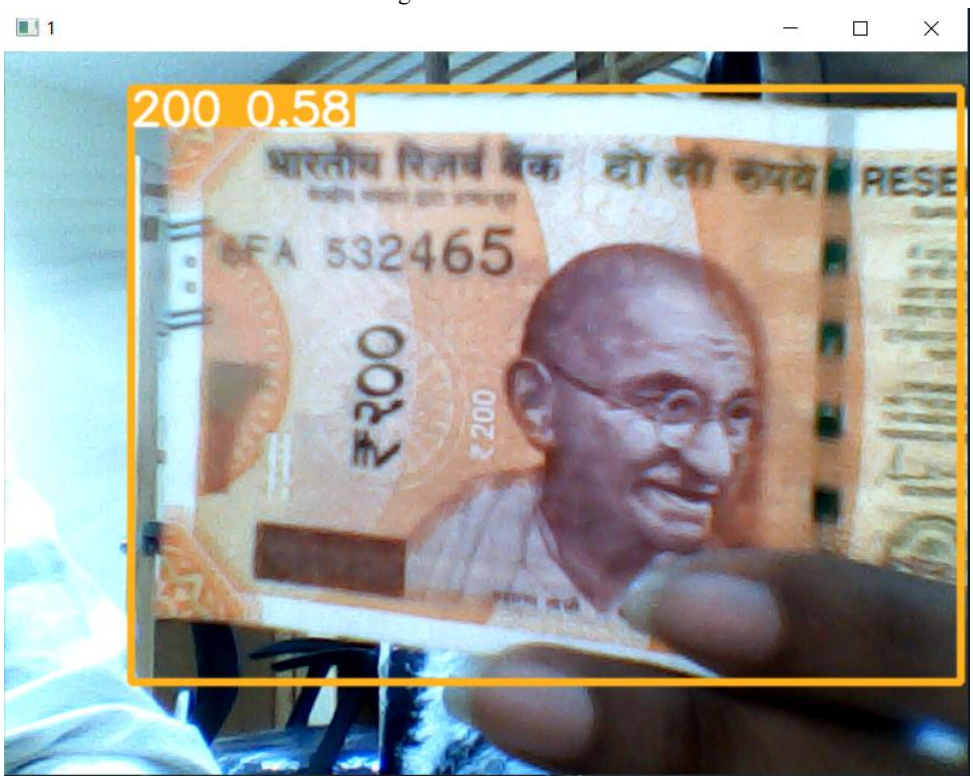
While running this command the present working directory should be the root of the yolov5 folder. The weight file ending in “.pt” file has to be given in the - -weight argument. The source can be set to 1 or 0 depending on the webcam used internal or external. On running this command a frame opens up with bounding box detection. In the detect.py we have made some edits so that the frames will only only be sent at a slower rate.

Indian Currency detection - Yolov5 folder has a in-built detect.py file. In the detect.py file where if conditions have been added for note detection and a command through voice. The following if conditions is been added in detedct.py files.

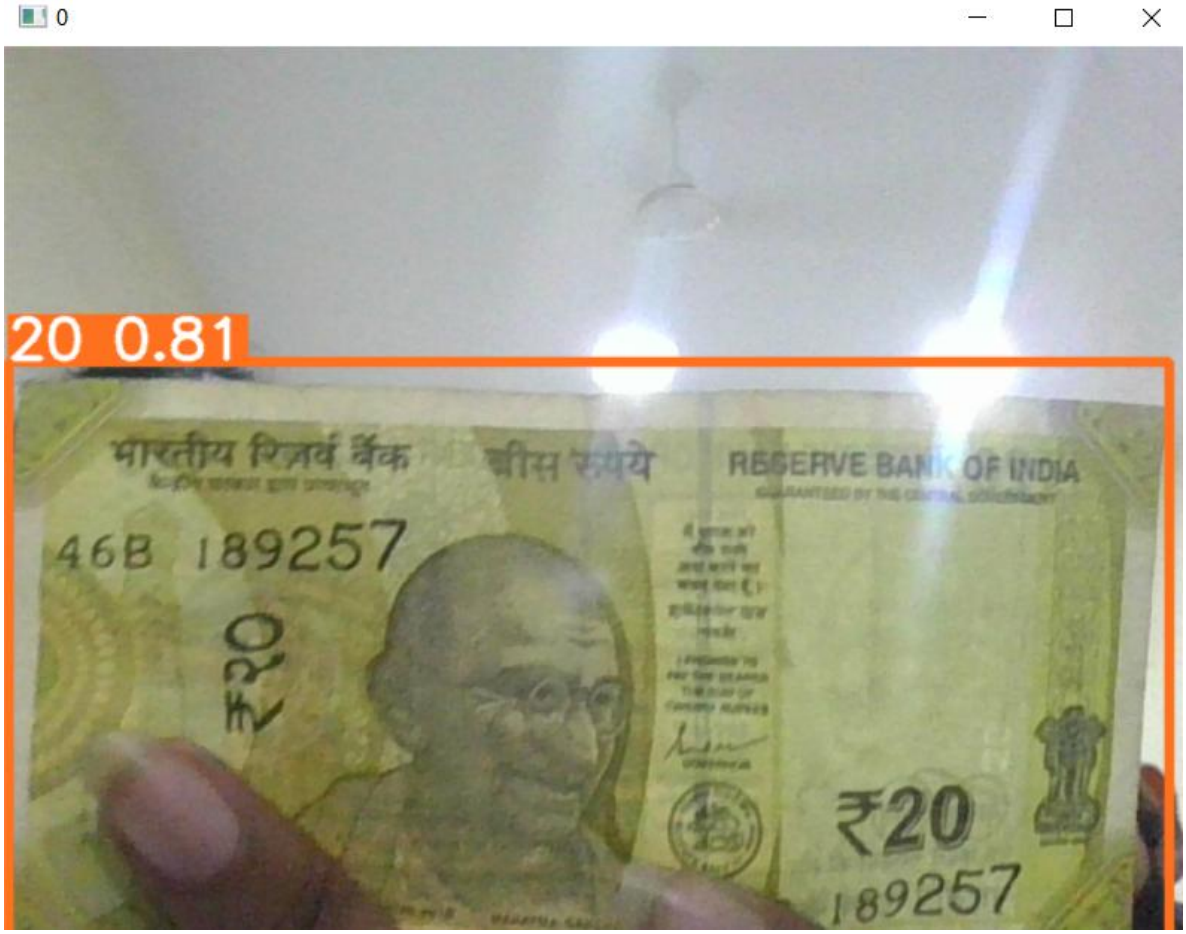
```
207
208         if "1 100" in s:
209             print(cc)
210             if(cc < 3):
211                 print("updating cc")
212                 a = "Its 100 rupee"
213                 os.system('espeak "{}"'.format(a))
214
215
```

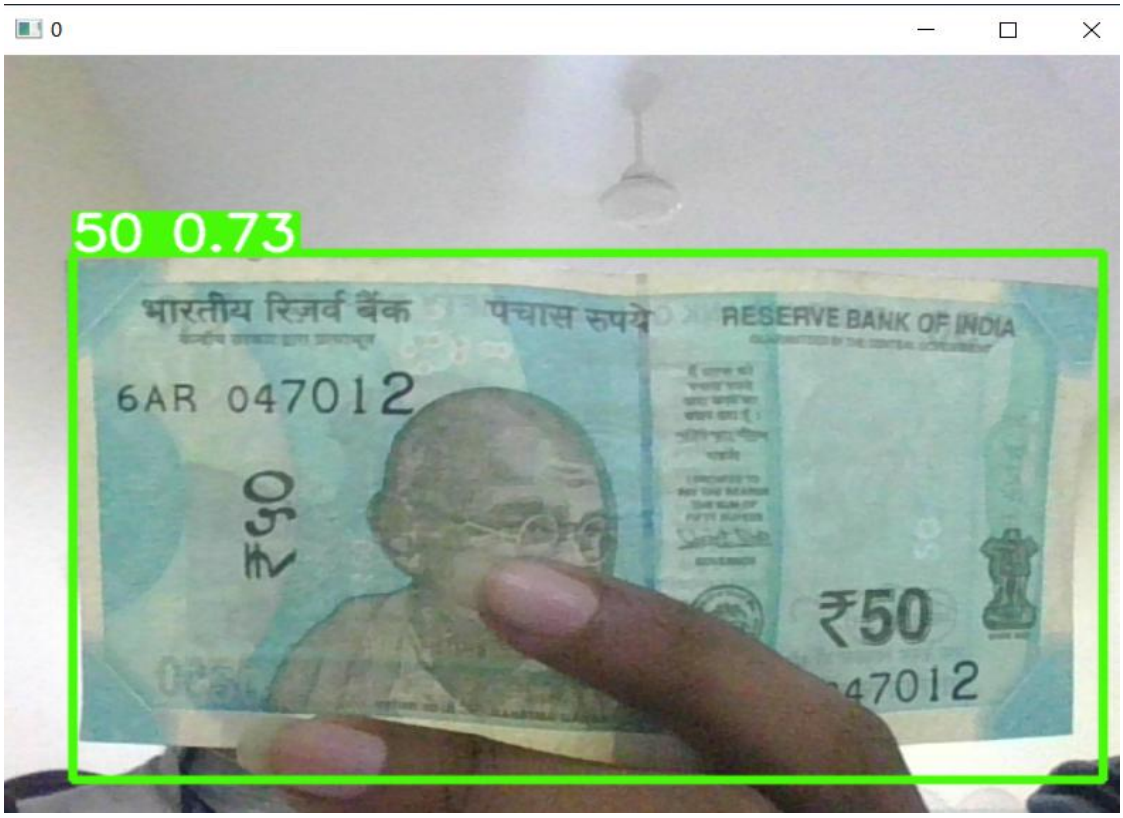
```
    elif "done" in s:
        do = "waiting"
        os.system('espeak "{}"'.format(do))

    elif "1 500" in s:
        b = "Its 500 rupee"
        os.system('espeak "{}"'.format(b))

    elif "1 2000" in s:
        c = "Its 2000 rupee"
        os.system('espeak "{}"'.format(c))

    elif "1 10" in s:
        d = "Its 10 rupee"
        os.system('espeak "{}"'.format(d))

    elif "1 200" in s:
        print(cc)
        if (cc < 3):
            print("updating cc")
            a = "Its 200 rupee"
            os.system('espeak "{}"'.format(a))

    elif "1 20" in s:
        e = "Its 20 rupee"
        os.system('espeak "{}"'.format(e))

    elif "1 50" in s:
        f = "Its 50 rupee"
        os.system('espeak "{}"'.format(f))

    else:
        cc=0
        g = "NO  rupee detected"
        os.system('espeak "{}"'.format(g))

```

The above code checks if the detected note is 100 then using eSpeak module voice will be appeared that “Its 100 rupee ” , in the same way it checks for all other notes.. These status messages include “100, 200, 10, 20, 50, 500, 2000” On receiving the status from camera. Once detected it gives the note detected as voice command.

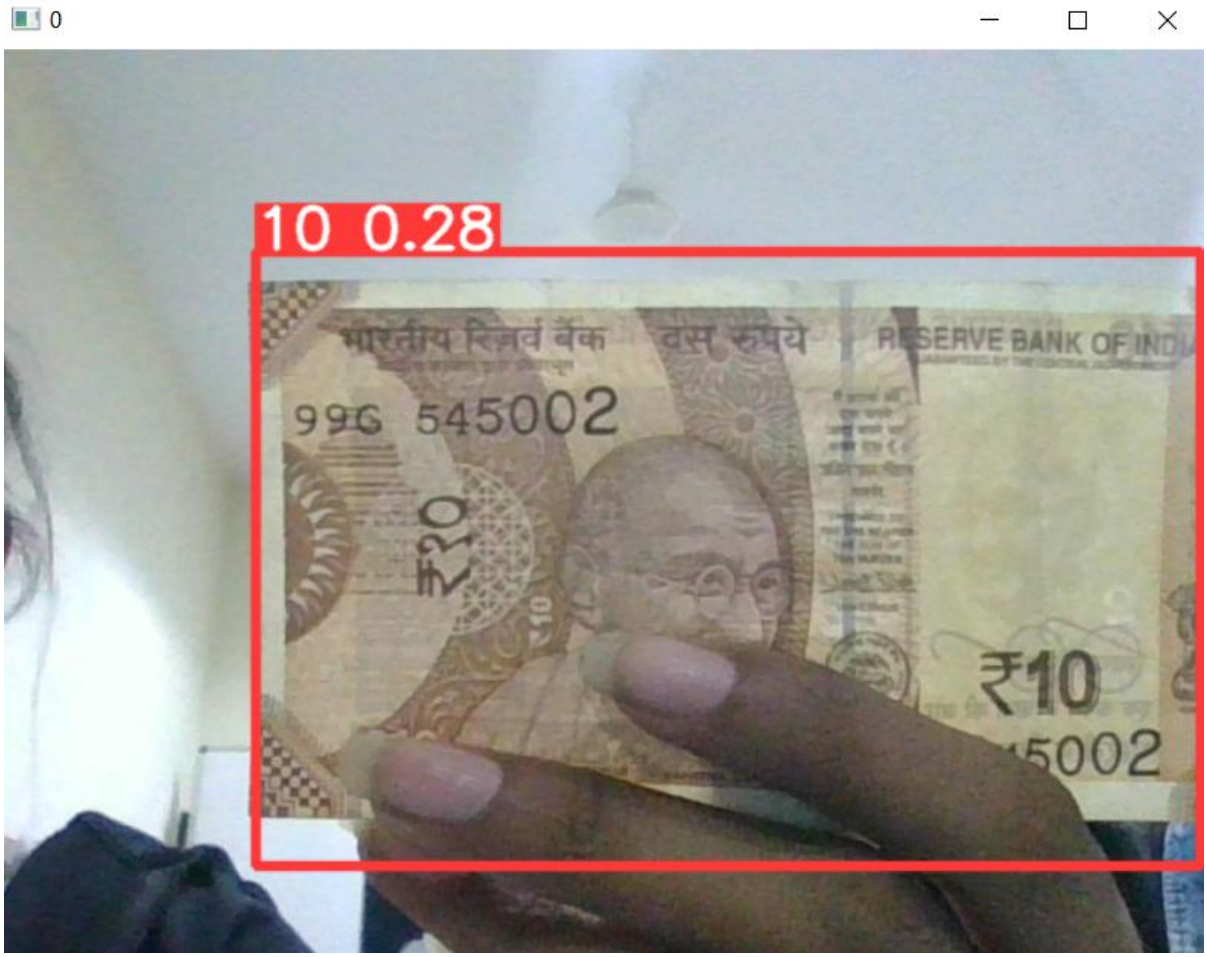



0
100 0.78
RESERVE BANK OF INDIA
₹100


0
20 0.81
RESERVE BANK OF INDIA
₹20


0
50 0.73
भारतीय रिज़र्व बैंक
पचास रुपये
RESERVE BANK OF INDIA
6AR 047012
₹50


0
500 0.42
भारतीय रिज़र्व बैंक
पाँच सौ रुपये
2LR 324699
₹500



5. Code

5.1 Arduino code

```
const int trig = 12; //Trigger pin of ultrasonic Sesnor
const int echo = 11; //Echo pin of ultrasonic Sesnor
long time_taken;
int dist,distance;

void setup()
{
  Serial.begin(9600);  //tells the arduino to get ready to exchange messages with the serial monitor  at a data rate of 9600 bits/sec.
  pinMode(trig, OUTPUT); //configures the specified pin to behave as output/input.
 pinMode(echo, INPUT);
}

/*###Function to calculate distance###*/
void calculate_distance(int trigger, int echo)
{
  digitalWrite(trigger, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger, LOW);
```

```
21
22   time_taken = pulseIn(echo, HIGH);
23   dist= time_taken*0.034/2;
24
25 }
26
27 void loop()
28 {
29   calculate_distance(trig, echo);
30   distance=dist; //getting distance of ultrasonic sensor
31   Serial.println(distance);
32   delay(6000);
33 }
```

Explanation:

Above code explains distance calculation using ultrasonic sensor. Line [1 - 2] defines the pin number for trig and echo pin of ultrasonic sensor. Line [3 - 4] initializing the the variables. Line [6 - 11] Void setup configures the specified pin to behave as output/input. Line [14 - 25] function to calculate distance. Formula used is dist= time_taken*0.034/2. Line [27 - 33] Void loop here getting the distance calculated by ultrasonic sensor from the function defined.

5.2 Python code

```
1 import serial
2 import time
3 import os
4
5 # make sure the 'COM#' is set according the Windows Device Manager
6 ser = serial.Serial('/dev/ttyACM0', 9700, timeout=1)
7 time.sleep(2)
8
9 for i in range(50):
10     line = ser.readline()   # read a byte
11     if line:
12         string = line.decode()  # convert the byte string to a unicode string
13         num = int(string) # convert the unicode string to an int
14         # os.system('espeak "{}"'.format(num))
15
16         if (num < 20):
17             a="StopStop, Close object in front."
18             os.system('espeak "{}"'.format(a))
19
```

```
20      elif (20 < num < 60):
21          b="object is quite a distance away."
22          os.system('espeak "{}"'.format(b))
23
24      elif (40 < num < 100):
25
26          c="object is 1 meter away."
27          os.system('espeak "{}"'.format(c))
28
29      else:
30          d="Object is far away."
31          os.system('espeak "{}"'.format(d))
32
33 ser.close()
```

Explanation:

Above code explains text to speech voice module. Python language is used. eSpeak module is used to convert the text to speech. The distance data sensed by ultrasonic sensor is passed to python code used serial communication. Line [1-3] importing libraries. Line [6] is where we give the COM port as per your device. Make sure it is correct. Here our device COM is /dev/ttyACM0 as per raspberry pi . Therefore, its used for transfer of data from pico board to python. Line [9] where it starts the for loop condition. Line [11 - 13] where we convert the byte string to a unicode string, then convert the unicode string to an int. Line [16 - 31] the condition for the command “stop there is a near by obstacle , if condition is greater than 20 and less than 60 then it commands “ object is quite a distance away. ” , if condition is greater than 40 and less than 100 then it says “ object is 1m distance away. ”, else it says “ object is far away ” .

6. Implementation of Raspberry pi 4

The Raspberry Pi 4 is mainly used to exclude the use of system interaction. Raspberry pi makes the use of portability compared to other computers. With the application of Balena Etcher it flash OS images to SD cards USB drivers. After flashing, the SD card has to be inserted in Raspberry pi computer in SD card slot.

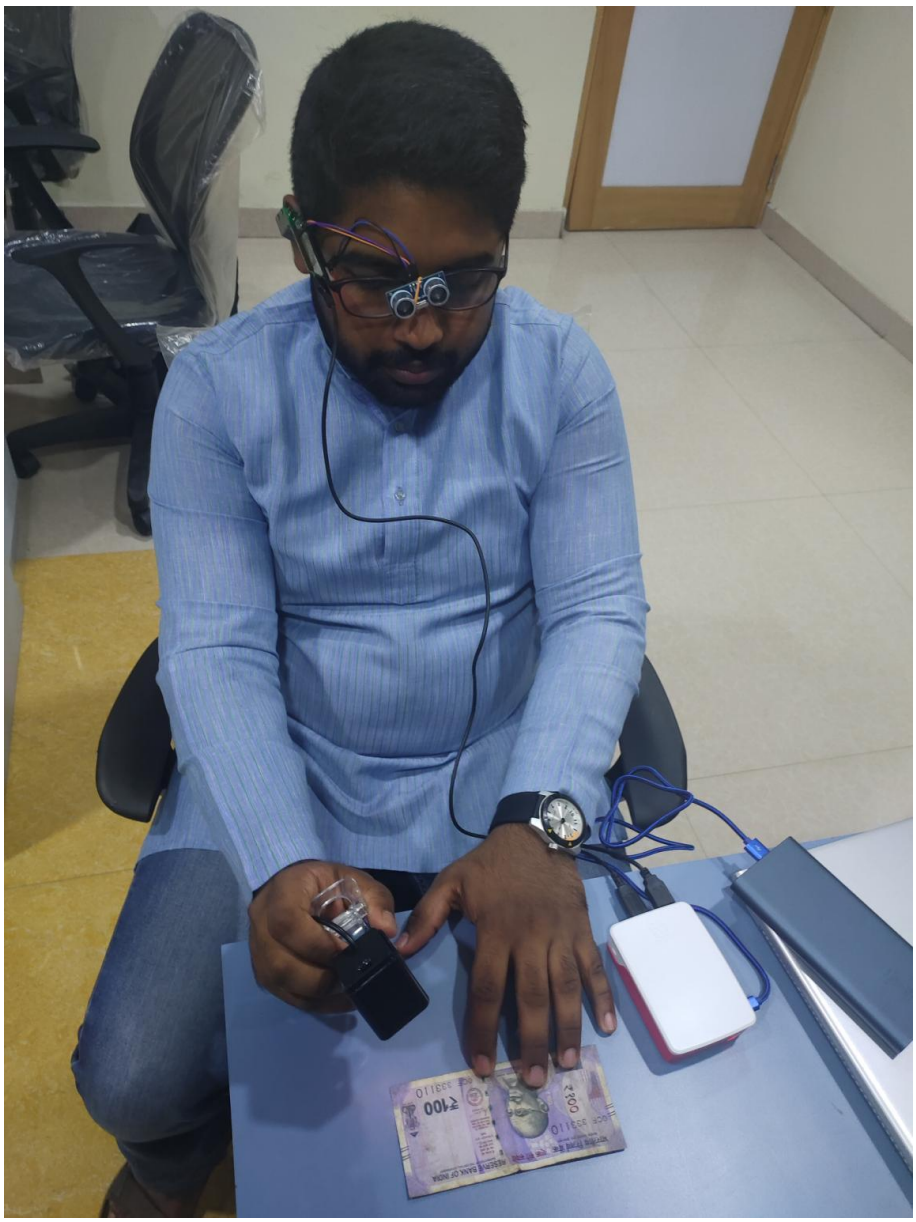
6.1 Features of Raspberry Pi 4

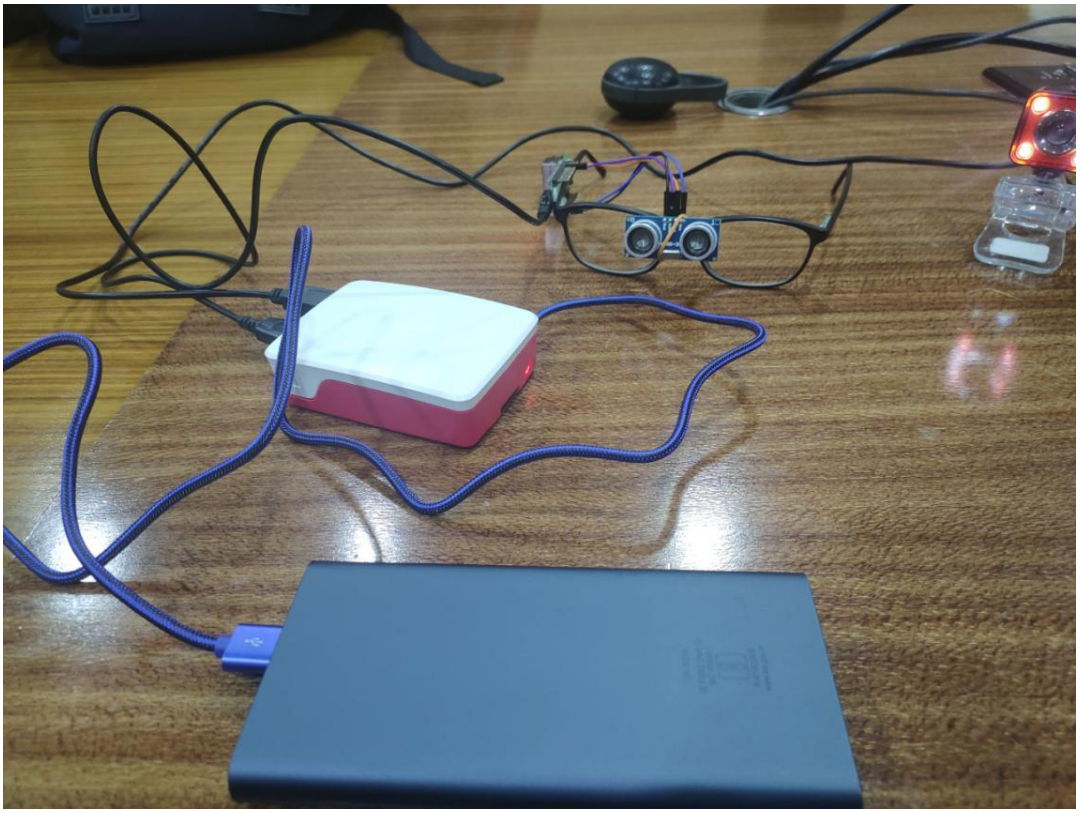
- A high-performance 64-bit quad-core processor
- Dual display support with resolutions up to 4K via a pair of micro-HDMI ports
- Hardware video decoding up to 4Kp60
- 4 GB of RAM
- A connection to the dual-band wireless local area network 2.4/5.0 GHz
- Bluetooth 5.0 / Gigabit Ethernet / USB 3.0 / PoE features (via a separate HAT PoE add-on module)

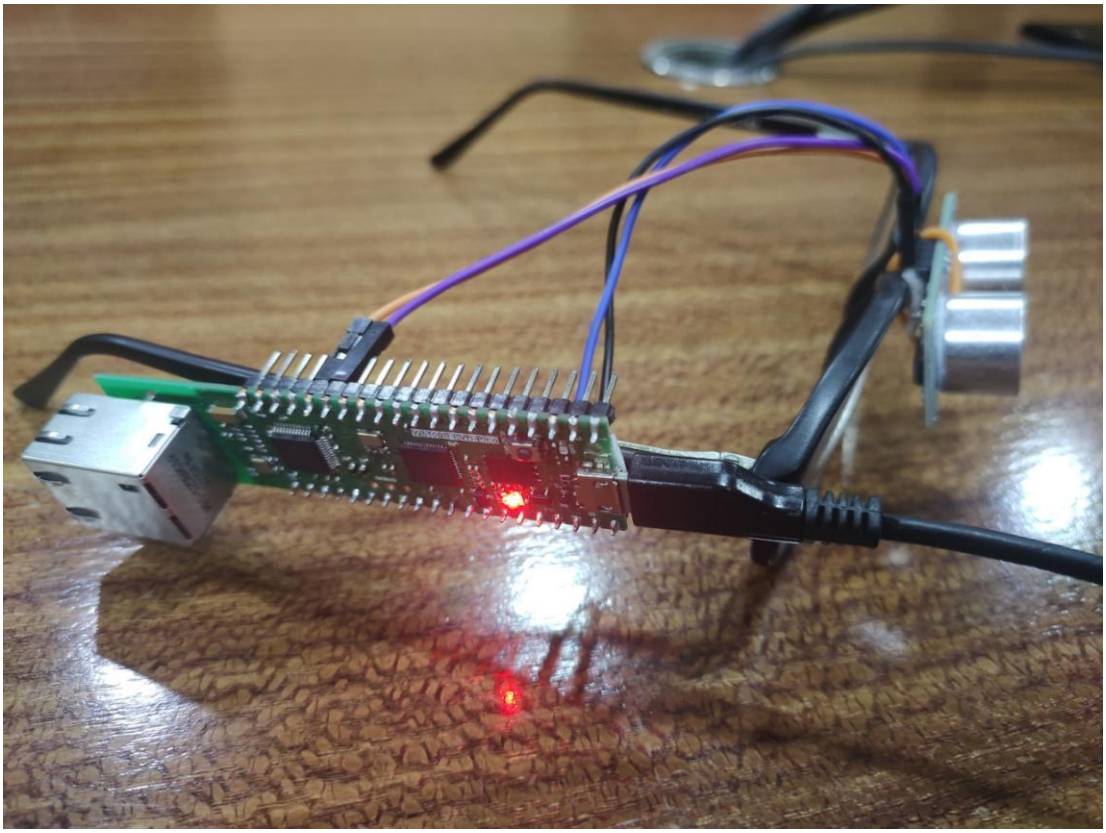
6.2 Setup of Raspberry Pi 4

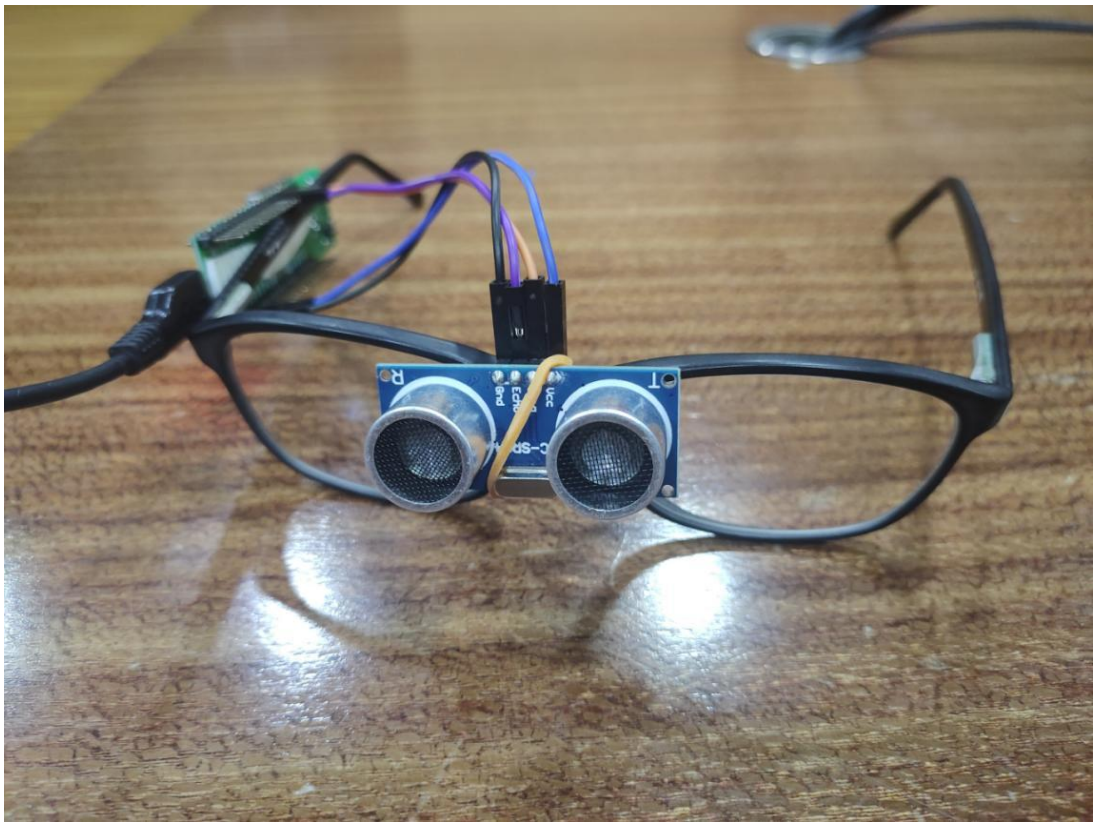
Connect monitor to Raspberry pi through HDMI port. Install all the required python libraries for the camera detection. Thonny software has been used to run python code. The power bank is used as the power source for Raspberry pi 4. The USB camera and Bluetooth Speaker is connected respectively for the currency detection and tells the currency detected through speaker to the user.

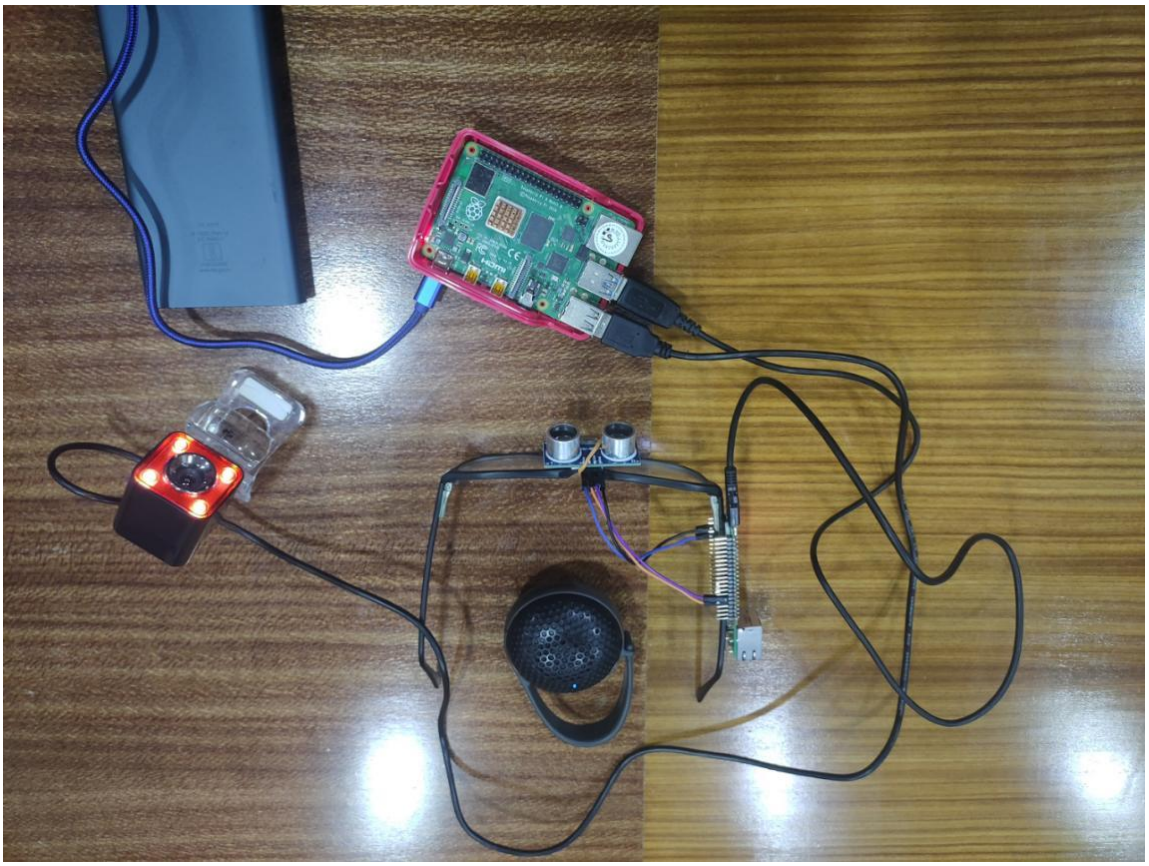
7.Images








R
T
Gnd
Vcc



8. References

[1]https://www.researchgate.net/publication/321288844_Low_cost_ultrasonic_smart_glasses_for_blind

[2] https://www.instructables.com/Talking-Smart-Glass-for-the-Blind/

[3] https://www.irjet.net/archives/V8/i4/PIT/ICIETET-50.pdf

[4]https://create.arduino.cc/projecthub/B45i/talking-smart-glass-for-the-blind-87d31e

[5] https://www.mdpi.com/2079-9292/11/7/1076/htm