How Raspberry Pi Microcomputers and Computer Vision Artificial Intelligence Can Be Utilized to Provide Aid to the Visually Impaired Community

CT Science & Engineering Fair Project # 6030

Abstract

According to the World Health Organization, an estimated 2.2 million people globally are suffering from a visual impairment or total blindness. Although millions have an inability to see, there is a lack of a practical product on the market which is both cost effective and efficient in assisting the visually impaired population. The Object Detection Universal System Plus (O.D.U.S.+) was designed in order to be that solution, allowing and providing the wearer with freedom and safety in movement, as well as extending their capabilities with Computer Vision Artificial Intelligence. The device was programmed with the programming language Python, and utilized two Raspberry Pi microcomputers, 4 ultrasonic sensors, 4 vibration motors, and a camera. O.D.U.S.+ works by collecting distance data using ultrasonic sensors and parsing it to provide alerts if an object is within 3-feet of the wearer, through 3 methods. These alerts are vibrations created by vibration motors on the glove, with data being received over BLE, or Bluetooth Low Energy, a series of beeps from a buzzer, and a computer-generated voice produced through a text-to-speech program. Furthermore, by using the TensorFlow LITE database, coupled with the programming library, OpenCV, the device can accurately describe objects in front of the wearer, providing an added sense of awareness. Performance tests were done on both the ultrasonic sensors and OpenCV to determine their accuracy. More research and development is needed to expand the functionality of the device, including multiple modes for different applications of the Computer Vision AI for different situations.

Background

According to a 2010 study conducted by the World Health Organization, an estimated 2.2 million people globally are suffering from a visual impairment or total blindness. Although hundreds of millions have the inability to see, there is a lack of a practical product on the market which is both cost effective and efficient in assisting the visually impaired population. Devices such as Ray Electronic Mobility Aid for the Blind act as a cane replacement, but it does not allow full mobility for the user; their hand is still occupied with the device since the user still must move it from side to side.

Project Goals

- Create a wearable device with sensor-based technology to collect distance data
- Develop Python Programs to parse the distance data and provide alerts based on how close an object is
- Integrate Computer Vision Artificial Intelligence, for added safety and usability

Materials

Materials	Quantity
HC-SR04 Ultrasonic Sensors	4
Raspberry Pi 4 B+	1
Raspberry Pi Zero W	1
Coral Edge TPU	1
USB Camera	1
Vibration Motors	4
Power Banks	2
Jumper Cables	16
Momentary Switches (Push Buttons)	2
Hat (Fedora)	1
Fingerless Glove	1
3d Printer	1
Soldering Iron and Solder	1 (each)
Wire Stripper and Cutter	1
Electrical tape and Heat Shrink	16
Needle and thread	1
Zip ties	4

Methods: Design and Construction

Hat Transmitter Module:

Four 5-Volt Ultrasonic Sensors were wired to the Raspberry Pi 4 over the General-Purpose Input Output Pins. Each sensor was positioned in one of the four directions: front, back, left, and right. The Power and Ground pins were connected in a parallel circuit (See Figure 1). The Trigger and Echo Pins were connected to unique pins, to be referenced in the code. Peripherals were also added, such as two momentary switches (push buttons) and a buzzer, each wired to unique GPIO Pins. Each of the ultrasonic sensors are mounted onto the hat with custom 3-part 3-D printed housings, the case body, the back of the case body, and the hat mount (See Figure 2). The hat mount was sewn directly onto the hat. Two 3 x 5 mm neodymium magnets were embedded in both the back of the case body and the hat mount, which allows for the sensors to be removable, making it easier to service the unit, as well as being able to wash the hat as needed. The case body and the hat mount have matching braille lettering to ensure the correct placement of the sensors after removal. This is the same for the Raspberry Pi case and the Coral Edge TPU Case. A camera mount was also designed and 3-D printed in order to secure the USB Camera needed for the Computer Vision AI.

Glove Receiver Module:

Four vibration motors were soldered and wired to unique GPIO pins on the Raspberry Pi Zero W, one for grounding the motor, and one for interfacing with the motor through the code (See Figure 3). Each motor was embedded into a 3-D printed vibration motor case, with braille lettering to correspond with each direction that the Ultrasonic Sensors on the hat are facing (See Figure 4). A 3-D Printed case was designed for the Raspberry Pi Zero W and was sewn onto the glove.

Methods: Programming

Hat Transmitter Module:

A custom Python script was created to continuously collect distance data every second from each sensor, parse it, and provide alerts if an object is within 3-feet of the wearer (See Figure 5). Those outputs include a computer-generated voice from a text-to-speech library, a series of beeps generated from the buzzer, and vibrations generated by motors on the glove. Another function was created to transmit the distance data to the Raspberry Pi Zero W on the glove, over Bluetooth Low-Energy (BLE). This wireless connection ensures full mobility and freedom of movement for the wearer. The program on the Hat Transmitter Module also has Computer Vision capability, with the library, OpenCV. It references a dataset compiled in the TensorFlow Lite database, which has the capability to detect and describe objects. The function takes the description of the object, and converts it into a String, or a data type that the text-to-speech function is able to utilize, allowing the computer-generated voice to describe the object that is in front of the wearer, as well as how far away it is from the wearer.

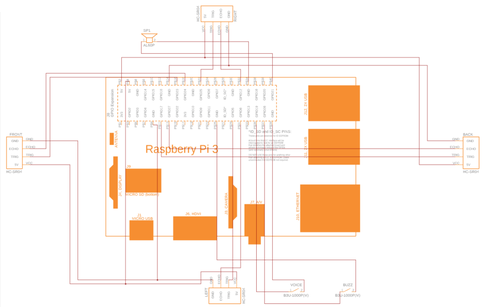

Figure 1: Ultrasonic Sensor Module Schematic

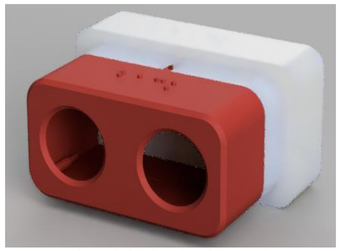

Figure 2: 3-D Render of the Back Ultrasonic Sensor Case

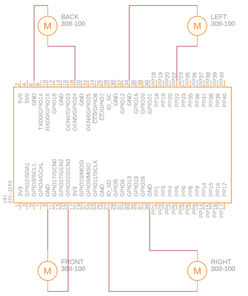

Figure 3: Vibration Motor Module Schematic

Figure 4: 3-D Render of the Front Vibration Motor Case

```python
def distance(GPIO_TRIGGER, GPIO_ECHO):

    global start_time
    global distance_alert

    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    while GPIO.input(GPIO_ECHO) == 0:

        start_time = time.time()

    while GPIO.input(GPIO_ECHO) == 1:

        time_elapsed = time.time() - start_time


    distance_cm = ((time_elapsed * 34300) / 2)
    distance_feet = round(distance_cm / 30.48)

    return distance_feet
```

Figure 5: Excerpt from the Ultrasonic Sensor program to continuously collect distance data

```python
def Distances():

    data_string = Receving_Data.split(",")

    Receving_Data = ""

    back_dist = int(data_string[1])
    front_dist = int(data_string[2])
    right_dist = int(data_string[3])
    left_dist = int(data_string[4])

    data_string = ""

    if back_dist <= 2 and back_dist > 0:

        print("Back Alert")

        for _ in range(1):

            GPIO.output(back_motor, GPIO.HIGH)
            time.sleep(0.3)
            GPIO.output(back_motor, GPIO.LOW)
            time.sleep(0.3)

        GPIO.output(back_motor, GPIO.LOW)

    if front_dist <= 2 and front_dist > 0:

        print("Front Alert")

        for _ in range(1):

            GPIO.output(front_motor, GPIO.HIGH)
            time.sleep(0.3)
            GPIO.output(front_motor, GPIO.LOW)
            time.sleep(0.3)

        GPIO.output(front_motor, GPIO.LOW)
```

Figure 6: Excerpt from Glove Receiver function to continuously receive distance data

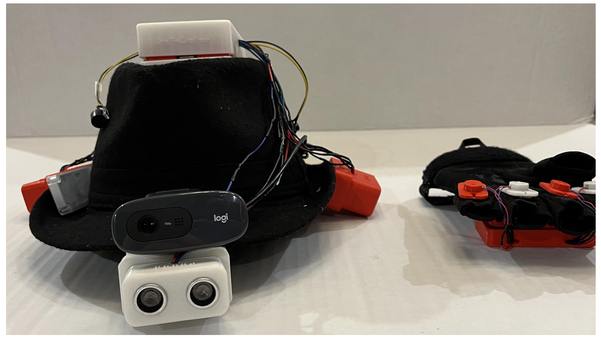

Figure 8: The completed O.D.U.S.+, with all the ultrasonic sensors, the vibration motors, the Raspberry Pi's, and the camera, all in working order

Methods: Programming Cont.

Glove Receiver Module:

The custom Python on the Raspberry Pi Zero W had two functions. One was to receive the distance data from the Hat Transmitter Module (See Figure 6), and the other was to interpret each distance and turn on the corresponding vibration motor on the glove, based on the smallest value.

Results

To ensure that every component of the O.D.U.S.+ was working properly, extensive testing was completed.

Ultrasonic Sensors:

The Ultrasonic Sensors are the most important part of the design, and it's important to test that each of them are working, as well as the accuracy of each sensor. This was done by collecting distance data in one program and creating a distance over time graph based on the data using the Matplotlib Python Library. This was done because it detailed the accuracy of how close an object was to the sensor. After each of the sensors were tested, it was determined that they all worked, and they were accurate.

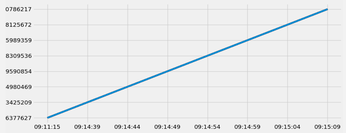

Figure 7: Distance vs. Time Graph for the Front Ultrasonic sensor

Computer Vision AI:

Programming the Computer Vision AI was the most complicated part of this project, because there were multiple parts to consider. The first test was making sure that the OpenCV commands from the library were executed fast enough to be effective, to help the visually impaired to identify objects quickly. This was done by completing a median-time performance test for the library, generating a chart of times, in milliseconds, that the commands were executed. After this experiment, it was determined that OpenCV was running fast enough, with the longest command taking 87.19 milliseconds. Finally, the last thing to test was the combination of OpenCV and the TensorFlow Lite Database, to test the framerate that the Raspberry Pi 4 would be able to detect objects. This was done by turning on the framerate overlay in the object detection preview window. After this experiment, it was determined that the framerate was much too low for Computer Vision to be effectively used, averaging at about 2 frames per second. In order to solve this problem, a Coral Edge TPU was utilized, as an Edge TPU is a device meant specifically to run artificially intelligent models. This drastically increased the frame rate by 4 times the original, at about 8 frames per second.

Conclusion

The Object Detection Universal System Plus is a device with the hopes of being a viable, comfortable, and functioning aid for the visually impaired. The four ultrasonic distance sensors, placed in the front, back, left, and right, provide a 360-degree protection radius for the wearer. The camera mounted on the front allowed for the Computer Vision AI to actively detect real-world objects, creating an added measure of safety and autonomy for the wearer. The wearer can be alerted about an object that is less than 3 feet away from them by three alerts, a vibration, a series of beeps, and a computer-generated voice. These alerts are user friendly, with the computer-generated voice stating what the object is, what direction the object is in and how far away it is. With the buzzer mode, the frequency of beeps change based upon how close the object is, and the vibration mode vibrates one of the four motors, representing front, back, left, and right for the closest object that is near the user. The vibration is continuous, and the data is sent from the Raspberry Pi 4 connected to the sensors to the Raspberry Pi Zero W, connected to the vibration motors. The data is being sent over the Bluetooth Low Energy link, and is acted upon by a custom Python program, on both the Raspberry Pi 4 and the Raspberry Pi Zero W.