

VIT Chennai Vandalur - Kelambakkam Road Chennai - 600 127 044 3993 1555

**OBJECT IDENTIFICATION FOR**

**VISUALLY IMPAIRED**

**IoT Fundamentals**

**20BCE1667 Tanvi Bahedia**

**GUIDED BY**

**Dr. Kanchana Devi V**

**ABSTRACT**

In exploring the field of Internet of Things (IoT), the solutions that arise in the realm of accessibility tend to have the most creative solutions. In developing technologies and coming up with new innovations, people with disabilities have become more included and given more access to information. A majority of the information presented to us on a daily basis is received with the use of our eyes. Webpages, books, images, and videos are some of the most common mediums of information sharing. With that fact it is important to realize the need to come up with alternative methods of sharing this information with individuals who are visually impaired. Throughout history efforts have been made to help people with these impairments and strides are still being made to this day to further make information accessible, many of which rely on computational technology.

With the help of this project, visually impaired persons will be given helpful assistance and support that they may utilize in their daily lives to help them function independently. Our primary goal will be to enhance the recognition capabilities of previous introduced models while also accelerating the processing rate. A Guiding assistant will also be developed to help perform the task of obstacle detection for the user. The service allows visually impaired users of tech devices to "see." With the help of a camera, the image will be captured, then it will be matched with the database that will be taken. If the object is identified in the given set of data, then it will activate the Audio IC which will be telling the user what object is in front of them or how far is the object in front of them. We will achieve these tasks with the help of Ultrasonic Sensors, Camera and Audio output, connected to the Arduino uno Board for communication.

**INTRODUCTION**

According to the information from the World Health Organization (WHO) in 2014, there are 285 million people estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. Human beings are highly dependent on vision sensor for daily tasks such as walking, eating, finding food, searching, driving vehicle, reading book etc., object recognition is the core algorithm in most of vision related task. Human outperform best computer vision algorithm to almost any measure and due to this main stream computer vision is always inspired from human vision, but visual neuroscience is limited to early vision. We do object recognition all the time for example while you are reading this thesis you are recognizing characters and hence words, object recognition is needed in navigation, tracking, automation and so on, from above discussion we can convince ourself that object recognition is highly important. The loss of vision often indicates loss of independence, lack of communication and human contact, which increase the limitations in mobility.

In this field, it has been a long tradition of concern of accessing computer systems by people with visual impairments. It is important to develop high quality user interfaces, accessible and usable by blind people with different skills, requirements, and preferences, in a variety of contexts of use, and through a variety of different technologies. In recent days, access for blind and visually impaired users to computer systems is gradually improving.

**EXISTING SYSTEMS**

[**Assistive Device for Mobility and Object Recognition for Visually Impaired People**](https://ieeexplore.ieee.org/document/8801898)

An innovative electronic assistive device known as the NavCane is proposed in this article. Its purpose is to empower visually impaired individuals with the ability to navigate and orient themselves independently. The gadget assists users in locating routes that are clear of obstructions, whether they are searching in an indoor or outdoor location. The NavCane is also useful for assisting with the identifying of things found inside an interior environment.

[**IoT Enabled Automated Object Recognition for the Visually Impaired**](https://www.sciencedirect.com/science/article/pii/S2666990021000148)

The paper addresses the difficulties that users face with mobility by providing methods for secure and risk-free movement in both indoor and outdoor settings. The proposed system provides assistance to people who are visually impaired in recognising objects that people who are visually impaired generally are unable to identify. The accuracy of the proposed system in the detection of objects is 99.31%, and its accuracy in recognising those objects is 98.43%. As a result of the proposed system's lighter weight in comparison to the weight of existing systems, an individual will have an easier time transporting the developed system.

[**Outdoor Navigation for Visually Impaired based**](https://ceur-ws.org/Vol-2514/paper102.pdf)

The ability of the proposed system to allow users to walk outside without stumbling while maintaining voice awareness will make it possible for the system to lower the risk of users colliding with obstacles. The proposed solution involves the construction of an independent application for outdoor navigation using a camera vision system that is based on mobile devices. In addition to this, the system has a high usability, making it possible for visually impaired people to navigate unfamiliar environments like parks, roads, and other places.

[**Assistive Object Recognition System for Visually Impaired**](https://www.ijert.org/assistive-object-recognition-system-for-visually-impaired)

This paper makes reference to a system that employs the Raspberry Pi computer and applies the You Only Look Once version 3 (YOLO v3) machine learning algorithm after it has been trained on the coco database. According to the findings of the experiment, YOLO v3 is capable of achieving state-of-the-art results with an overall performance of 85% to 95% and a recognition accuracy of 100% for persons, chairs, clocks, and mobile phones. Not only does this system make it possible for visually impaired people to move around more freely, but it also makes it possible for them to understand that an XYZ object rather than an obstacle lies ahead.

**PROBLEM STATEMENT**

Blind people have always historically struggled with interacting with the physical world. Being unable to see limits how much we can experience and act on our surroundings. Navigating through unknown environments is a challenging tricky task for individuals with vision impairments. Typically, this burden on the blind person was reduced by someone helping them with their daily chores. This, however, makes the blind person dependent on their helper and they lose certain independence due to their condition. People with visual impairments have to memorize the directions, as it is impossible to note them down. If the visually impaired get lost, the only way is to find someone to help them. Visually impaired people have little opportunity to find their way in an

unfamiliar place. This is not a very desirable situation for the blind person. There are two main

challenges that affect a person's ability to navigate. The first is obstacle avoidance, which addresses the objects and terrain in the person's immediate surroundings environment such as persons, stairs, walls, and tables. In addition, the inability to handle these types of situations themselves has an adverse impact on the feeling of independence of the visually impaired. There must be something that allows the blind person to act independently and have autonomy. We aim to solve that problem.

**INTRODUCTION TO PROPOSED MODEL**

The problem statement clearly states the objectives of this project. The proposed model will help the visually impaired to a certain level and will support them in everyday life. Object recognition is inherent part of our vision system to survive in this world, human and other creatures can perform this task instantly and effortlessly. To successfully detect surrounding objects, we investigate several existing detection systems that could classify objects and evaluate it at various locations in an image. This project tries to transform the visual world into the audio world with the potential to inform blind people objects as well as their spatial locations. In this project, we want to explore the possibility of using the hearing sense to understand visual objects. The sense of sight and hearing sense share a striking similarity: both visual object and audio sound can be spatially localized.

**PROPOSED SOLUTION**

The idea here is to design the systems which will be more efficient for visually impaired people. There are two objectives set for this project. The proposed solution has two goals to be achieved:

1. **Object Detection:**

The object detection system will help in avoiding any obstacle in the path for the user. The object detection system will also work as obstacle avoidance and guidance system for the visually impaired. The idea is to fix Ultrasonic sensors in shoes. There will be 2 Ultrasonic sensors fixed in the shoes. The sensors will read the distance between the obstacle and the user and will warn the user at a certain distance to stop and instruct the user to turn right or turn left based on the location of the object. The warnings and instructions will be given to the user by an audio output interface.

1. **Object Identification:**

The object identification system will help the visually impaired not only detect some object in a room but also realize and know what the object is in front of them. The object identification will utilize a camera to capture images of the live object and things present in real time. The object identification system will process the images and help the visually impaired know about the object. There will be a fixed database and information of objects present in the system to identify the basic objects and entities around the user.

**COMPONENTS OF THE MODEL**

**HARDWARE**

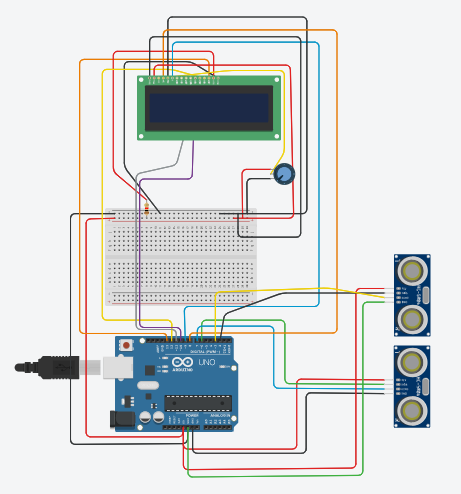
1. Ultrasonic sensors
2. Arduino Uno board
3. USB Cable
4. Jumper Wires
5. Laptop Webcam

**SOFTWARE**

1. Arduino IDE
2. Tinkercad
3. PyCharm
4. Python [CODING LANGUAGE]
5. OpenCV [CODING LANGUAGE]

**BLOCK DIAGRAM**

**1) OBJECT DETECTION:**

****

**Figure 1 – Circuit connection for Object detection using Arduino uno and ultrasonic sensors**

The circuit connects the two ultrasonic sensors with the Arduino uno board. The sensors read the distance between the object/obstacle and processes it in Arduino uno. The program will then calculate the distance and check whether the user is close to the object/obstacle. If very close then the system will warn the user and guide the user accordingly to turn right or turn left. The GND in sensors is connected to GND in Arduino. VCC in both the sensors is connected to 5V in Arduino uno. The trigger pin in sensors is set and connected to Pin 13 and 8 in Arduino uno respectively. Similarly, the echo pin in sensors is connected to Pin 12 and 7 in Arduino respectively.

The circuit is then connected to the laptop using USB cable to load programs and stimulate the circuit using the program. The Arduino IDE is used to load the programs into the Arduino uno

**DESCRIPTION OF THE COMPONENTS**

**HARDWARE**

1. **Ultrasonic sensors**

The HC-SR04 Ultrasonic Distance Sensor is a sensor used for detecting the distance to an object using sonar. The HC-SR04 uses non-contact ultrasound sonar to measure the distance to an object, and consists of two ultrasonic transmitters (basically speakers), a receiver, and a control circuit. The HC-SR04 sensor works best between 2cm – 400 cm (1" - 13ft) within a 30-degree cone, and is accurate to the nearest 0.3cm.

1. **Arduino Uno board**

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It is basically used in designing of basic circuits.

1. **USB Cable**

USB cables carry power as well as signals. This allows for "USB powered" gadgets as well as recharging batteries in cameras and other USB peripherals. USB cables are designed with several distinct connector types, making it easy to identify which plug goes into the computer.

1. **Jumper Wires**

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

**SOFTWARE**

1. **Arduino IDE**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

1. **Tinkercad**

Tinkercad is a free web app for 3D design, electronics, and coding. Tinkercad is an online collection of software tools from Autodesk that enable complete beginners to create 3D models. This CAD software is based on constructive solid geometry (CSG), which allows users to create complex models by combining simpler objects together.

1. **PyCharm**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

1. **Python [CODING LANGUAGE]**

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

1. **OpenCV [CODING LANGUAGE]**

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

**MODULES USED:**

1. **Serial :** This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named "serial" automatically selects the appropriate backend.
2. **Time :** The Python time module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.
3. **Pyttsx3 :** It is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3. init() factory function to get a reference to a pyttsx3. Engine instance.
4. **Cv2 :**  It is the module import name for opencv-python.

**DATASET**

**COCO Dataset:**

The Microsoft Common Objects in Context (MS COCO) dataset is an example of a large-scale dataset that includes object recognition, segmentation, key-point detection, and captioning. There are 328 thousand photos included in the collection.

Splits: The initial edition of the MS COCO dataset was made available for download in the year 2014. It includes 164 thousand photos that have been separated into training (83 thousand), validation (41 thousand), and test (41 thousand) sets. Additional test photographs totaling 81K were made available in 2015; these included all of the previously published test images as well as 40K brand-new images.

In 2017, the training/validation split was modified from 83K/41K to 118K/5K as a direct result of comments received from the community. The new division makes use of the same photos and annotations as the previous one. A total of 41,000 photos from the 2015 test set were used to compile the 2017 test set. In addition, the 2017 edition includes a brand new dataset of 123 thousand photos that has not been annotated.

**SSD\_mobilenet**

In order to accomplish the task of object detection, the mobilenet-ssd model functions as a Single-Shot multibox Detection (SSD) network. To put it another way, this file contains a Tensorflow model that has already been pre-trained and has been trained using the COCO dataset.

**Frozen-graph**

The act of identifying and saving all of the necessary things (graph, weights, etc.) in a single file that you can quickly utilize is referred to as "freezing." For making inferences using TensorFlow, frozen graphs are frequently employed, and they also serve as building blocks when making inferences with other frameworks.

**WORKING MECHANISM**

1. **Object Detection:**

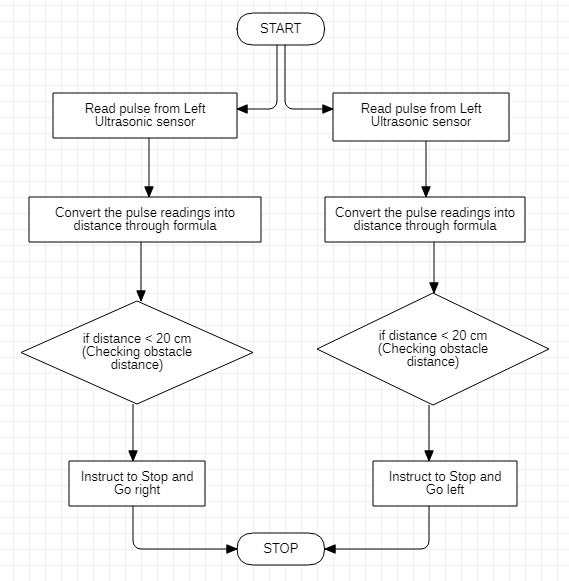
The circuit will be constructed as explained in the block diagram above. The two Ultrasonic sensors will read distance between the obstacle and the user. A certain warning distance will be set. If there is any obstacle ahead of the user, the user will be warned through an audio output interface by stopping him. The system will then guide the user to either continue going ahead left or right based on the location of the obstacle. This will also be achieved through audio output interface. The readings will be taken continuously and system will work accordingly to help the user navigate through his/her path.

1. **Object Identification:**

The object identification system setup is constructed as shown above in the report. The CAMERA captures the images and live photos and provides it to the system to process it. The OpenCV program then processes the image and tries to match with the existing database which has the basic everyday objects used by humans. The coco.name file consists of basic elements to match and identify the object. It will inform the user about the object identified through audio output interface.

**FLOW DIAGRAM / FLOW CHART**

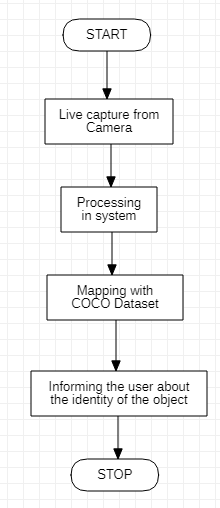
**OBJECT DETECTION:**



**Figure 2 – Flow Chart for Object detection using Arduino uno**

*The pulse is read with the assistance of ultrasonic sensors in accordance with a specific distance, which, when bordered, sends an alert-output to the Arduino indicating that the obstacle distance is getting closer and that the user needs to halt. The user is given instructions to either turn left or right, or to halt, depending on the direction in which the obstruction was discovered.*

**OBJECT IDENTIFCATION**



**Figure 3 – Flow Chart for Object Identification**

*The camera captures the data around the user and sends user the needed information by processing the image retrieved and then mapping it to the dataset included [COCO dataset] and proceeding with an audio output in user understandable format.*

**CODE**

**1. OBJECT DETECTION [ARDUINO]**

#define trigPin 13

#define echoPin 12

#define echoPin2 7

#define trigPin2 8

void setup()

{

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(trigPin2, OUTPUT);

pinMode(echoPin2, INPUT);

}

void loop()

{

long duration,duration2, cm, cm2;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

cm = (duration/2)/29.1;

digitalWrite(trigPin2, LOW);

delayMicroseconds(2);

digitalWrite(trigPin2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin2, LOW);

duration2 = pulseIn(echoPin2, HIGH);

cm2 = (duration2/2)/29.1;

if(cm < 20 || cm > 320)

{

stop();

delay(30);

go\_left();

delay(20);

}

if(cm2 < 20 || cm2 > 320)

{

stop();

delay(300);

go\_right();

delay(2000);

}

if(cm2 < 20 && cm <20)

{

stop();

delay(300);

}

else

{

delay(1000);

}

}

void stop()

{

Serial.println("STOP SOMETHING AHEAD ");

}

void go\_left()

{

Serial.println("Go LEFT");

}

void go\_right()

{

Serial.println("Go RIGHT");

}

**[PYTHON CODE]**

import serial

import time

import pyttsx3

# initialisation

engine = pyttsx3.init()

language = 'en'

ser = serial.Serial('COM5', 9600, timeout=1)

time.sleep(2)

for i in range(100):

line = ser.readline()

if line:

string = line.decode()

engine.say(string)

engine.runAndWait()

print(string)

ser.close()

**2. OBJECT IDENTIFICATION [OpenCV]**

import cv2

import time

import pyttsx3

cap = cv2.VideoCapture(0) #VIDEO CAPTURE OBJECT DEFINATION

cap.set(3,640)

cap.set(4,480)

engine = pyttsx3.inti()

classNames = [] #ARRAY DEFINED FOR DATABASE

classFile = 'coco.names'

with open(classFile,'rt') as x:

classNames = x.read().rstrip('\n').split('\n') #STORING NAMES IN ARRAY FROM COCO FILE

configpath = 'ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt'

weightspath = 'frozen\_inference\_graph.pb'

net = cv2.dnn\_DetectionModel(weightspath,configpath)

net.setInputSize(320,320)

net.setInputScale(1.0/ 127.5)

net.setInputMean(127.5)

net.setInputSwapRB(True)

while True:

success, img = cap.read()

classIDs, confs, bbox = net.detect(img,confThreshold = 0.5)

print(classIDs,bbox)

if len(classIDs) != 0:

for classIDs, confidence, box in zip(classIDs.flatten(),confs.flatten(),bbox): #GREEN BOX

cv2.rectangle(img,box,color=(0,255,0),thickness=2)

cv2.putText(img,classNames[classIDs-1].upper(),(box[0]+10, box[1]+30),

cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 255, 0), 2)

cv2.imshow("Output", img) #DISPLAYING IMAGE

engine.say(classNames[classIDs-1])

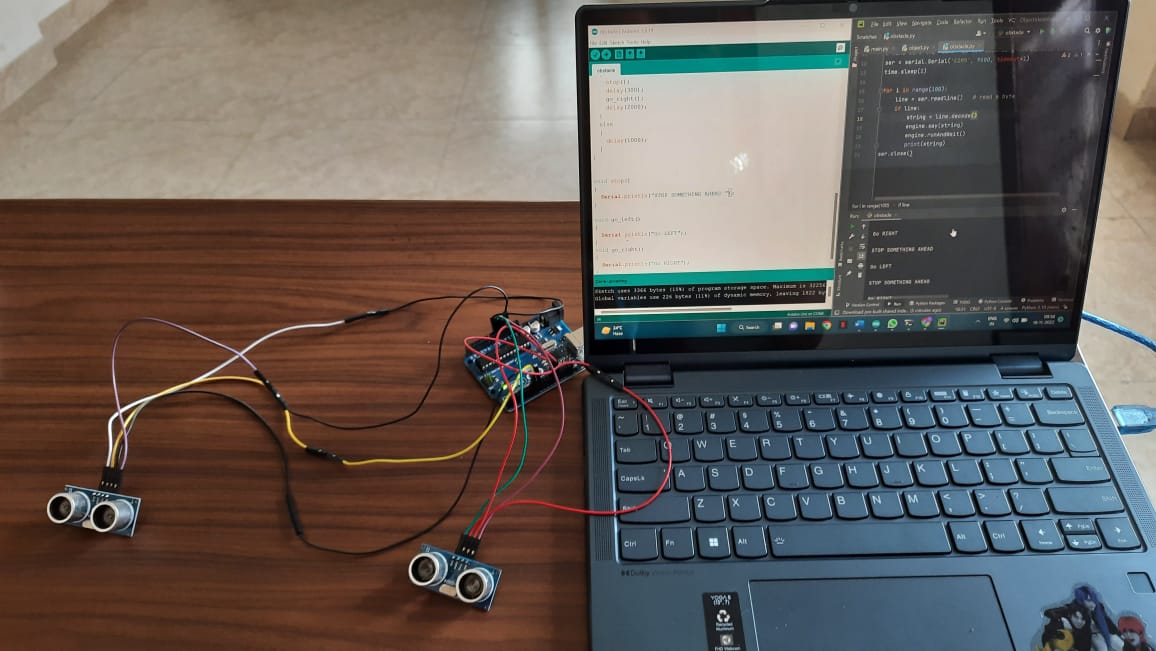
engine.runAndWait()

cv2.waitKey(1) #RUNS INFINITELY UNTIL KEY PRESSED

time.sleep(1)

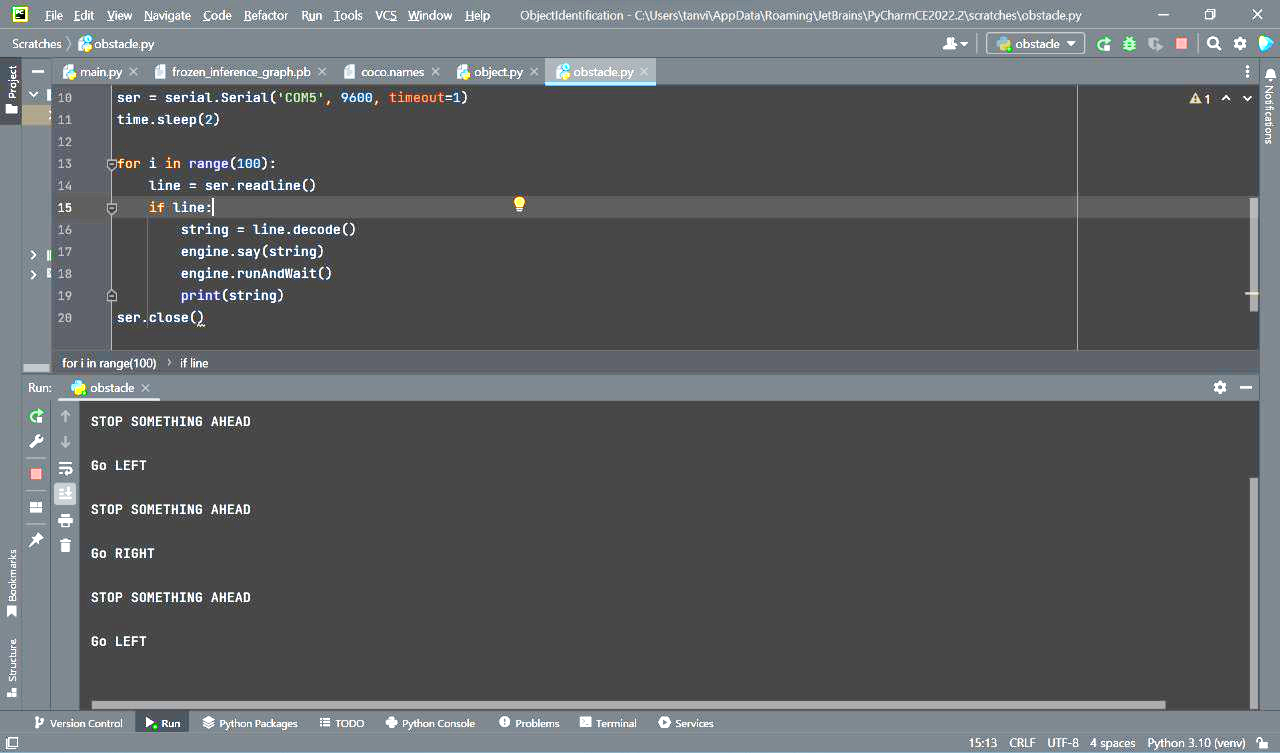
**OUTPUT SCREENSHOTS**

**1. OBJECT DETECTION**



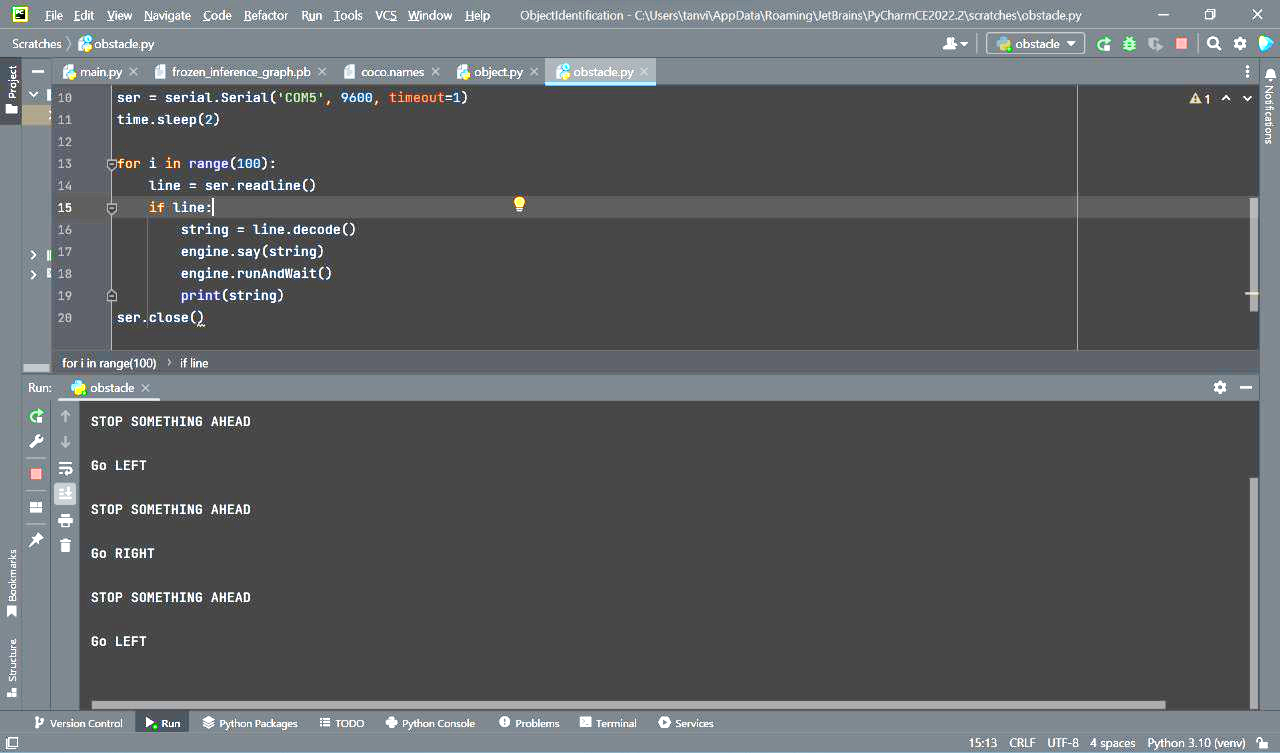
**Figure 4 – Output for Object Detection using Arduino Uno**

*This figure shows the set-up of ultrasonic sensors [left and right] connected to Arduino through jumping wires, which is connected to the device [Laptop] through USB cable. The Arduino IDE helps uploading the code to the Arduino which makes the sensors work and then the received output is then sent to python [PyCharm] to vocalize the result to be told to the user [visually impaired].*



**Figure 6 – Audio Output for Object Detection using Arduino Uno**

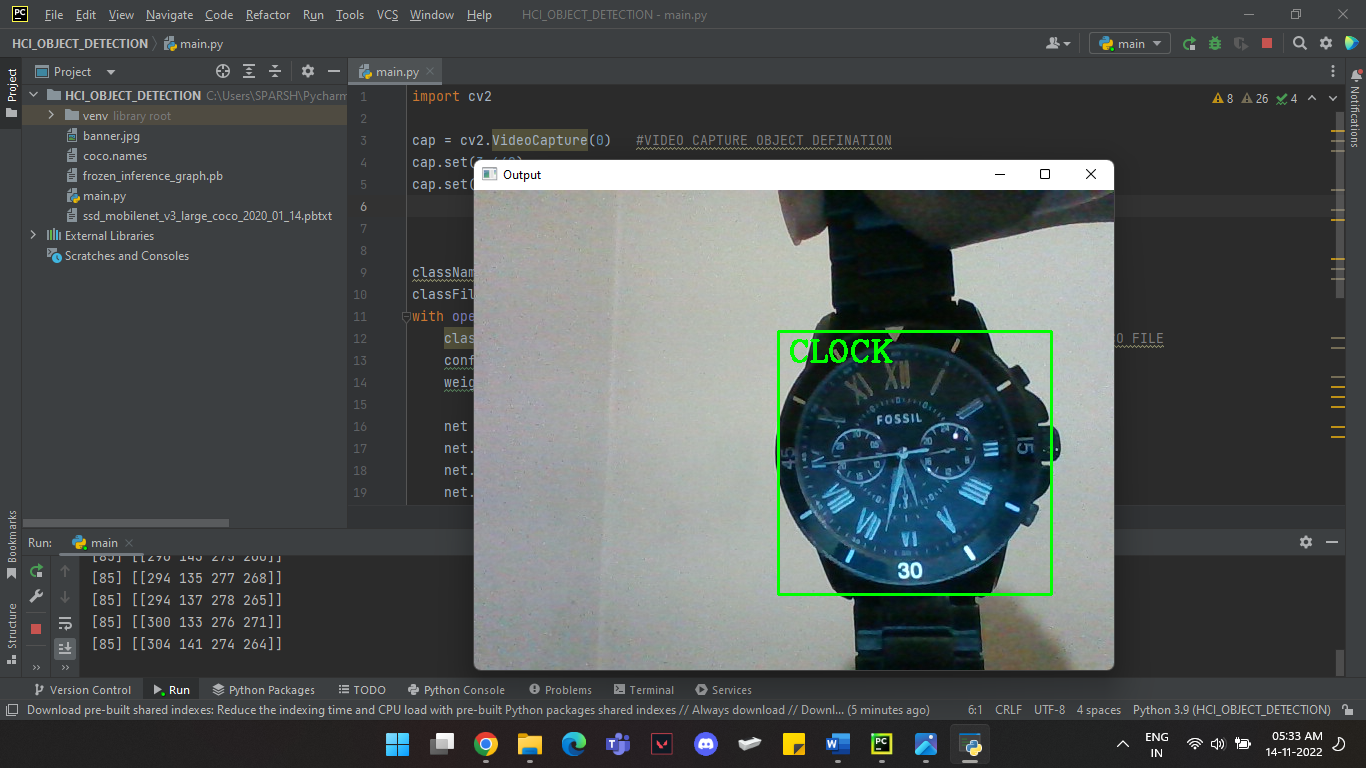
*The python code in PyCharm to collect information from Arduino-Ultrasonic sensors and send an audio output of the result obtained.*



**Figure 5 – Audio Output for Object Detection using Arduino Uno**

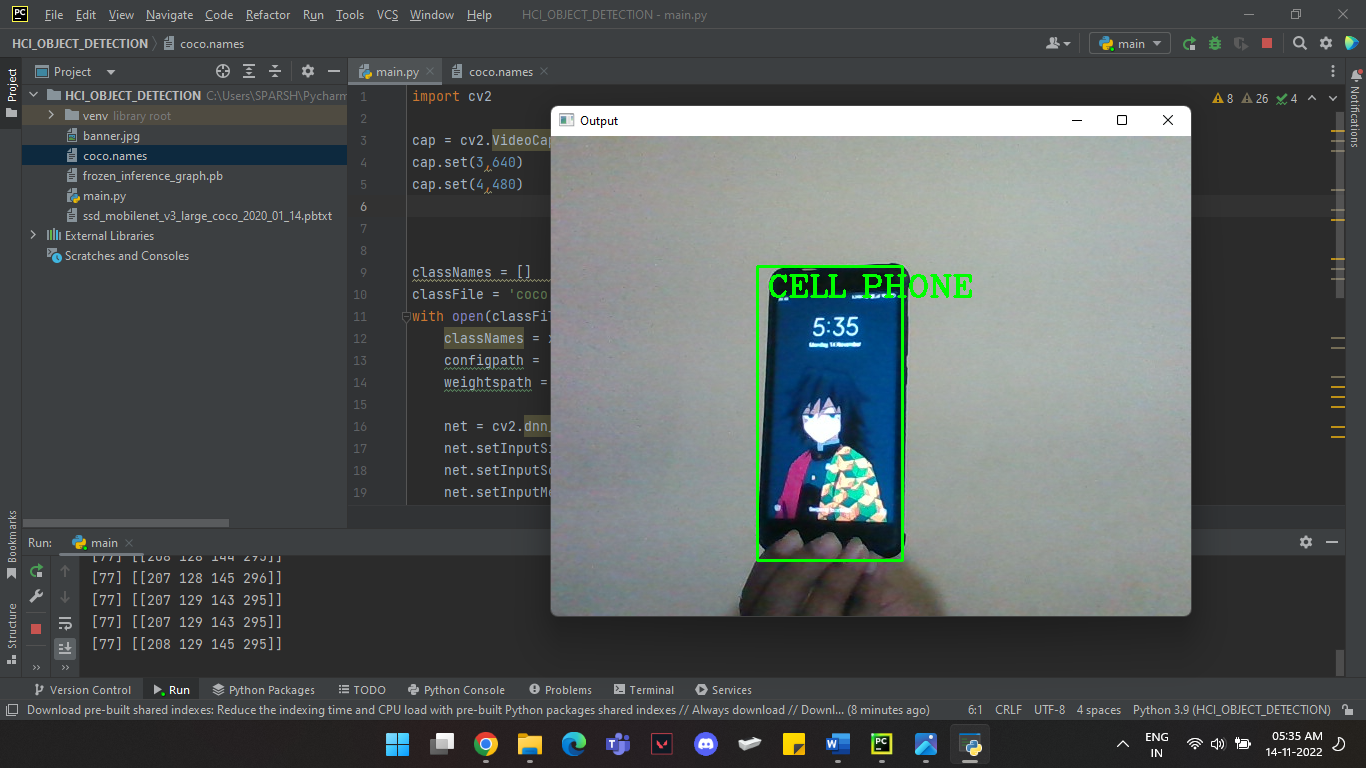
*The result obtained when an obstacle was found in the range of sensors to alert the user.*

1. **OBJECT IDENTIFICATION**



**Figure 6 – Output 1 for Object Identification**

*The result obtained after running the* ***Object Identification*** *code in PyCharm using CV Module and with the help of COCO Dataset, SSD\_Mobilenet and Frozen-Graphs. There is an audio-output incorporated within this system using python module.*



**Figure 7 – Output 2 for Object Identification**

*The result obtained after running the* ***Object Identification*** *code in PyCharm using CV Module and with the help of COCO Dataset, SSD\_Mobilenet and Frozen-Graphs. There is an audio-output incorporated within this system using python module.*

**CONCLUSION**

In this project (j component) a preliminary background review about HCI design for blind people was

presented. The proposed solution had two objectives. One is Object detection and other is Object Identification. These two objectives proposed as a solution for the visually impaired people will help and assist them at a certain level. The object detection module purpose is to warn the user about any obstacle ahead using sensors and guide which direction to turn. Whereas the object identification module uses the webcam and identifies the object using the images captured through webcam.

The objectives defined were achieved and hardware implementation was also completed successfully. The proposed systems are effective in helping visually impaired individuals in everyday environments. Assistive system using object recognition is in its very early stage and hence need lot of further efforts to make it feasible for end user, it may need industrial collaboration and funding. As far as future work is concern two main improvements would be useful, one is processing time reduction for object recognition module and second is improved number of categories in dataset,

**REFERENCES**

[1] Edward, Shirly. (2016). Object Identification for Visually Impaired. Indian Journal of Science and Technology. 9. 10.17485/ijst/2016/v9iS(1)/108446.

[2] <https://github.com/rtgdk/VizAid>

[3] <https://how2electronics.com/esp32-cam-based-object-detection-identification-with-opencv/>

[4] <http://ethesis.nitrkl.ac.in/7480/1/138.pdf>

[5] <https://www.intechopen.com/chapters/56916>

[6] <https://uxdesign.cc/human-computer-interaction-for-the-visually-impaired-a467ad61175b>