ASSISTANCE FOR IMPAIRED PEOPLE USING ARTIFICIAL INTELLIGENCE

A MINI PROJECT REPORT

Submitted by

R.DHANVARSHINI [REGISTER NO:211419104060]
N.DIVYA [REGISTER NO:211419104071]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

JUNE 2022

ASSISTANCE FOR IMPAIRED PEOPLE USING ARTIFICIAL INTELLIGENCE

A MINI PROJECT REPORT

Submitted by

DHANVARSHINI.R [REGISTER NO: 211419104060]

DIVYA.N [REGISTER NO: 211419104071]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

JUNE 2022

BONAFIDE CERTIFICATE

Certified that this mini project report **"ASSISTANCE FOR IMPAIRED PEOPLE USING ARTIFICIAL INTELLIGENCE"** is the bonafide work of **"N.DIVYA[211419104071], DHANVARSHINI.R [211419104060]"** who carried out the mini project work under **Mrs.P.VIJAYALAKSHMI, M.C.A, M.Phil , M.Tech, (Ph.D)** supervision.

SIGNATURE 15/6/22

SIGNATURE

Dr. S.MURUGAVALLI, M.E, Ph.D..,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Mrs.P.VIJAYALAKSHMI,
M.C.A, M.Phil , M.Tech, (Ph.D)
SUPERVISOR
ASSOCIATE PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that this candidates were examined in the Mini Project Viva- Voce Examination held on....15/6/22........

INTERNAL EXAMINER

15/6/22
EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **DIVYA.N(211419104071), DHANVARSHINI.R(211419104060),** hereby declare that this mini project report titled **"ASSITANCE FOR IMPAIRED PEOLE"**, under the guidance of **Mrs.P.VIJAYALAKSHMI, M.C.A, M.Phil , M.Tech, (Ph.D)** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

N. Divya

1.DIVYA.N

R Dhanvarshini

2.DHANVARSHINI.R

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A, Ph.D..,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this mini project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C.VIJAYARAJESWARI**, **Dr. C.SAKTHIKUMAR, M.E., Ph.D.** and **Tmt. SARANYASREE SAKTHIKUMAR, B.E., M.B.A.,** for providing us with the necessary facilities for completion of this mini project.

We also express our gratitude to our Principal **Dr. K.Mani, M.E., Ph.D..,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E, Ph.D..,** for the support extended throughout the project.

We would like to thank my mini project guide **Mrs.P.VIJAYALAKSHMI, M.C.A, M.Phil , M.Tech, (Ph.D).,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the mini project.

N.DIVYA

R.DHANVARHINI

ABSTRACT

Detection of objects can be accomplished in a variety of ways. According to the World Health Organisation(WHO), over 40 million people worldwide are blind, with another 250 million who have some vision impairment. They face a lot of trouble and constant challenges in Navigation, especially when they are on their own. They need to often depend on someone to get their fundamental daily needs met.Developing accurate Machine Learning Models capable of identifying and localizing multiple objects in a single image has long been a significant challenge in computer vision. The practice of detecting real-world object instances in still photos or videos, such as a car, bike, TeleVision, flowers, and humans, is known as object detection. It lets us recognize, localize, and detect many things inside an image, giving us a better overall understanding of the scene. The existing system consists of object detection,device sensor,braille system, voiceover. The existing system does help the visually impaired people to navigate easily. But they aren't aware of the objects description and their distance.Proposed systems- depth analysis, voice feedback. However, thanks to recent advances in Deep Learning, developing Object Detection applications is now easier than ever. TensorFlow's Object Detection API is an open-source framework built on top of TensorFlow that makes building, training, and deploying object detection models simple.One of our project's goals is to create an integrated Machine Learning Framework. This will allow visually impaired persons to recognize and classify daily things with voice help. Which in turn calculate distance and issue warnings if the person is too close or too far away from the object. Obstacle detection devices can be built using the same framework. It's a complex process, it is yet implemented using mobile phones. Warning messages are sensitive and shows many warnings.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

INTRODUCTION

1.1 OVERVIEW

We came up with an Integrated Machine Learning System which allows the Blind Victims to identify and classify Real Time Based Common day-to-day Objects and generate voice feedbacks and calculates distance which produces warnings whether he/she is very close or far away from the object. The same system can be used for Obstacle Detection Mechanism.

The system is set up in such a way where a camera will capture real-time frames and will send it to a laptop based Networked Server where all the computations take place.

The Laptop Based Server will be using a pre-trained SSD detection model trained on **COCO DATASETS**. It will then test and the output class will get detected with an accuracy metrics.

After testing with the help of voice modules the class of the object will be converted into a default voice notes which will then be sent to the blind victims for their assistance.

Along with the object detection, we have used an alert system where approximate will get calculated. If that Blind Person is very close to the frame or is far away at a safer place, it will generate voice-based outputs along with distance units.

1.2 PROBLEM DEFINITION

The lack of visual capabilities has limited these individuals from completely perceiving their immediate surroundings which has potential safety concerns and also lowers their quality of life since they must rely on some sort of aid to get around. Currently, in order for visually impaired individuals to get around, they rely on walking canes, guide dogs, and/or personal human aids for assistance. While these walking canes and guide dogs may allow the individual to get around independently, they each have a common drawback. These aids lack the intelligence to provide directions to unvisited locations and cannot completely warn individuals of obstruent objects in their vicinity. A human aid provides this intelligence but makes the visually impaired individual very dependent on the human aid. A good solution will be a device that is portable and is able to provide directions to new locations and alert the user of obstacles in their path when the user is walking. For the purpose of this project, we streamlined the problem defined above to assisting blind individuals with getting around outdoors. We believe a project like this will create the case for further investment in creating smarter electronic devices to assist visually impaired individuals with getting around. By smarter we mean that the device is able to provide directions to unvisited locations, while ensuring safe navigation. Additionally, the device is not intended to replace the use of walking canes or guide dogs.

Design a portable device for visually impaired individuals that will provide direction to new locations and alert the user of obstacles in their path during outdoor navigation.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

2.1 INTRODUCTION

There are a lot of devices which assist the visually challenged for navigation indoor and outdoor.

2.2 LITERATURE SURVEY

2.2.1 Wearable Navigation Assistance System for the Blind and Visually Impaired

Author: Ali Khan; Aftab Khan; Muhammad Waleed

Year: 2019

An assistance system was developed in this research work which is based on ultrasonic sensors for obstacle detection. The ultrasonic sensors along with a vibration device and a buzzer are placed on multiple places in a wearable jacket. Sensors scan the environment of user and inform them through vibration and buzzer sound when the sensor detects any obstacle.

Methodology used:

The above paper proposes a system utilizes stereo vision, image processing methodology and a sonification procedure to support blind navigation. The developed system includes a wearable computer, stereo cameras as vision sensor and stereo earphones, all molded in a helmet.

Pros:

- It provides high sensitive sensors.

Cons:

- Size of hardware required is extensive and voluminous.
- Musical stereo sound for the blind’s understanding of the scene in front.

2.2.2 Blind Path Obstacle Detector using Smartphone Camera and Line Laser Emitter.

Authors: Rimon Saffoury* , Peter Blank* , Julian Sessner**, Benjamin H. Groh* , Christine F. Martindale* , Eva Dorschky* , Joerg Franke** and Bjoern M. Eskofier*

Year: 2018

Two things are prominently used in this paper i.e. Mobile camera and laser. The laser and the mobile is kept at static distance. The image is captured from the camera and along with it the laser is also observed. Using the static distance and the angle between the laser point and the camera the distance is measured.

Methodology:

The method of information transfer to the user about obstacles was efficient and useful as almost all subjects did not collide with an obstacle. However, some of them knew immediately how to handle the appearing obstacle, whereas other firstly needed to rethink. Nonetheless, an acoustic signal as feedback may reduce the natural use of the visually impaired person’s sense of hearing. Transmitting the environmental information, about obstacles, by other means should also be explored.

Pros:

High accuracy with fast object detection.

Cons:

- Distance between the camera and the laser must be constant.
- May not work efficiently on shiny surface as laser intensity may decrease.

2.2.3 Visual Assistance for Blind using Image Processing

Author: S.Mahalakshmi1, Veena N2, Anisha Kumari3 1,2Assistant Professor, Dept. of ISE, BMSIT&M, Karnataka 3UG Student, Dept. of ISE, BMSIT&M, Karnataka

Year: 2020

The machine acts by means of supplying speech instructions to the user through interface. A microphone captures the speech as input. The acquired input is diagnosed with the aid of the use of Google API. Also, it makes use of image processing as its primary approach to identify objects and signal boards. The video is captured by the Pi camera that is attached to the CSI port, which is then transformed into data frames for further processing. These frames are preprocessed for higher results. Now, Image processing algorithms for the item detection is carried out on these statistics frames and the object is identified. Audio commands from headsets are given to the user to inform him/her approximately the current position of the item.

Methodology used:

This system gives them a sense of visualization as it also helps them visualize their nearby environment based on their voice commands. This system consists of a simple architecture which makes it complexity free and user friendly

Pros:

Voice feedback with high accuracy and high quality.

Cons:

Environment cover is less in area.

2.2.4 Obstacle Detector for Blind Peoples

Author: M. Maragatharajan, G. Jegadeeshwaran, R. Askash, K. Aniruth, A. Sarath

Year: 2019

To make the life to be as a normal one for the blind peoples this may be very helpful project for them. By making this as a gadget or a device in their hand they can easily judge an object by their own by knowing the buzzer sound. The system uses ultrasonic sensor as a wide range of field to detect an object with its higher detection range. Based on this project we take survey in our institution.

Methodology used:

This project Arduino based obstacle detector for blind people is a new method to resolve their problems. A less complex portable, cost efficient, easy to manage an effective system with many more amazing properties and advantages are proposed to provide support for the blind. The system will be very easy to find the distance between the objects and the sensor. It can detect the objects in every directions th the blind person. Without the help of others the blind person can move from one place to other and lead their regular lives independently.

Pros:

High accuracy object detection.

Cons:

The description of the object is not notified to the user.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISITING SYSTEM

- **Object detection**

Object detection is a computer vision technique that allows us to identify and locate objects in an image or video. With this kind of identification and localization, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labeling them.

- **Device sensor**

It turns something about the physical world into data upon which a system can act. Traditionally, sensors have filled well defined, single-purpose roles: A thermostat, a pressure switch, a motion detector, an oxygen sensor, a knock detector, a smoke detector, a voltage arrestor. Measure one thing, and transmit a very simple message about that one thing. This thinking stems from several hundred years of physical engineering of devices and persists today in part because of the convenience of modular thinking in system design.

- **Braille system**

Braille is a reading and writing system for blind and vision impaired people, made up of raised dots that can be 'read' by touch. The most popular form of braille is Grade 2, which uses the alphabet as well as abbreviations and contractions.

- **Voiceover**

Using AI voice makers simplifies the process of creating voice overs. It gives you complete control over your process, allows you to directly convert your home recordings or scripts into voiceovers. AI text to speech is time and cost-effective while retaining the quality of your voice overs.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

- It doesn’t gives the description of the object detected
- It doesn’t not give high accuracy
- It is difficult for the blind people to understand the environment.

3.2 PROPOSED SYSTEM

- **DEPTH ANALYSIS**

It gives the description of the object, distance between the objects and the person

- **VOICE FEEDBACK**

It converts the object which is detected text into its respective voice feedback and is sent to the person.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

- It alerts the user with the distance of the object and the person
- It gives a feedback about the object description

3.3 HARDWARE ENVIRONMENT

- **LAPTOP**

It is a plateform where the code is executed and the output is shown.

- **CAMERA**

It capture the real time environment and sends it to the laptop.

3.4 SOFTWARE ENVIRONMENT

- STACKOVERFLOW
- ANACONDA
- SYDER
- PYTHON
- TENSORFLOW APIs

CHAPTER 4

SYSTEM DESIGN

SYSTEM DESIGN

4.1 ER DIAGRAM

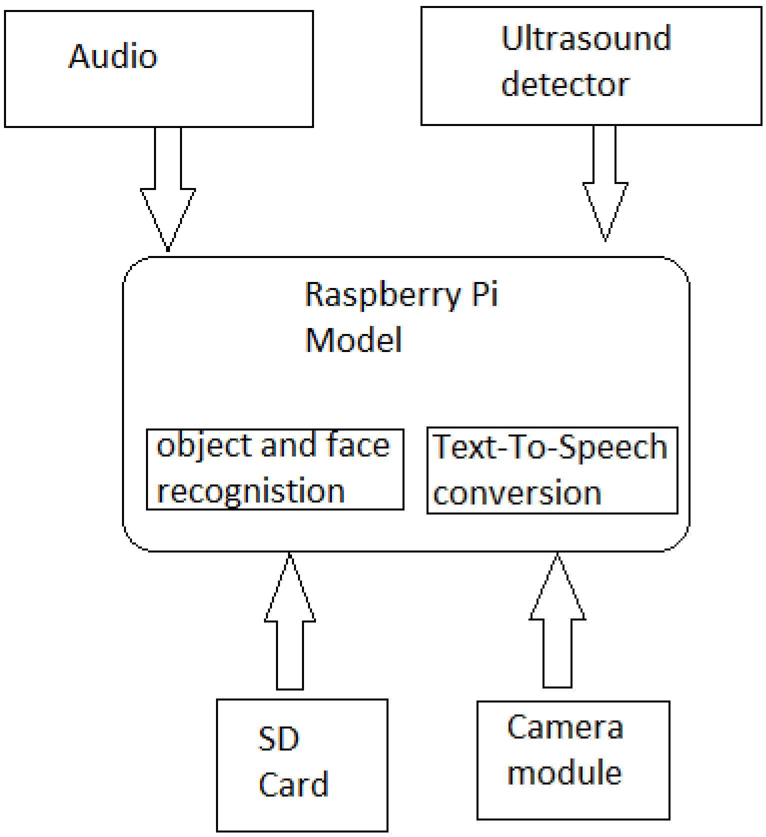

Fig.4.1. ER DIAGRAM

Description:

Entity relationship(ER) Diagram using Raspberry Pi Model it does object detection on the objects present infront of the user and takes it as an input by using Raspberry pi and converts the obtained input into voice output.

4.2 DATAFLOW DIAGRAM

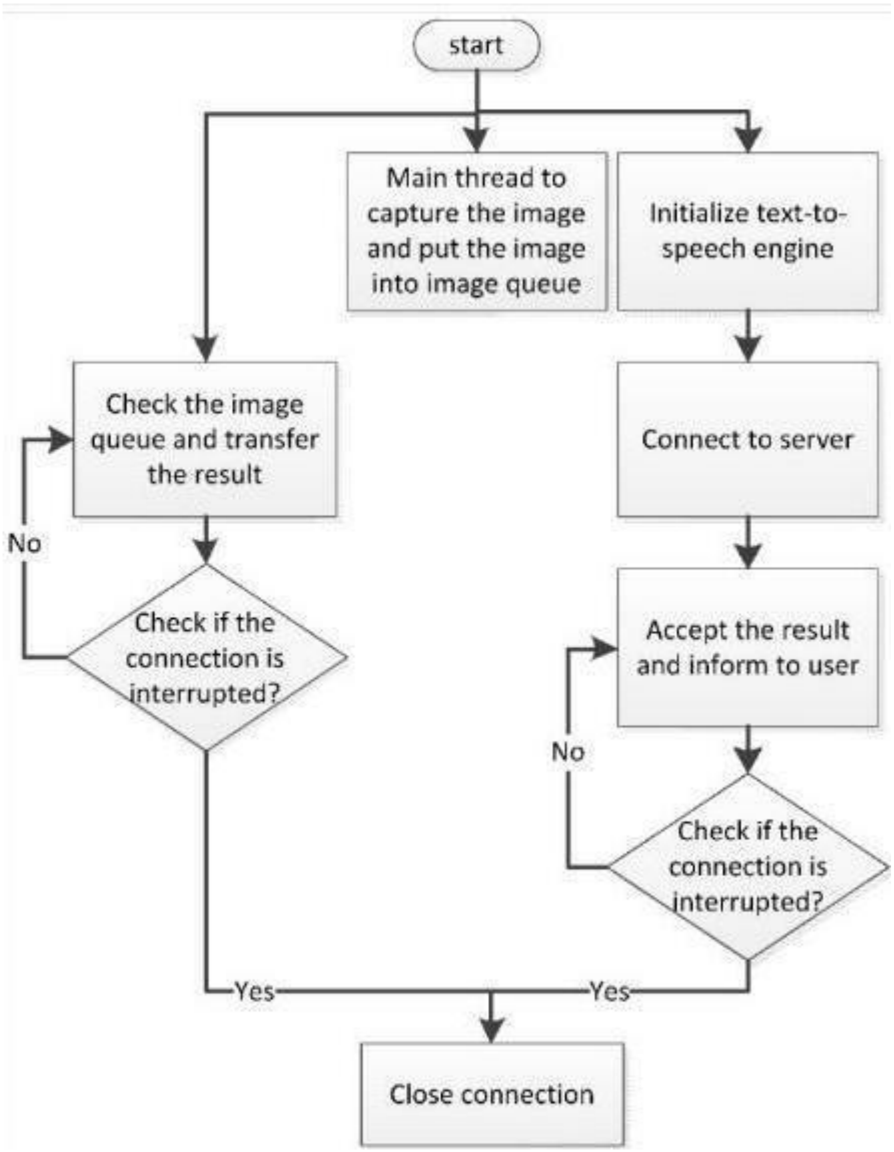

Fig.4.2. Data Flow diagram.

Description:

The detected images or the objects will go through a process where it is grouped according to its observation and the respected description of the image obtained converted into voice accordingly.

4.3 UML DIAGRAM

4.3.1 USECASE DIGRAM

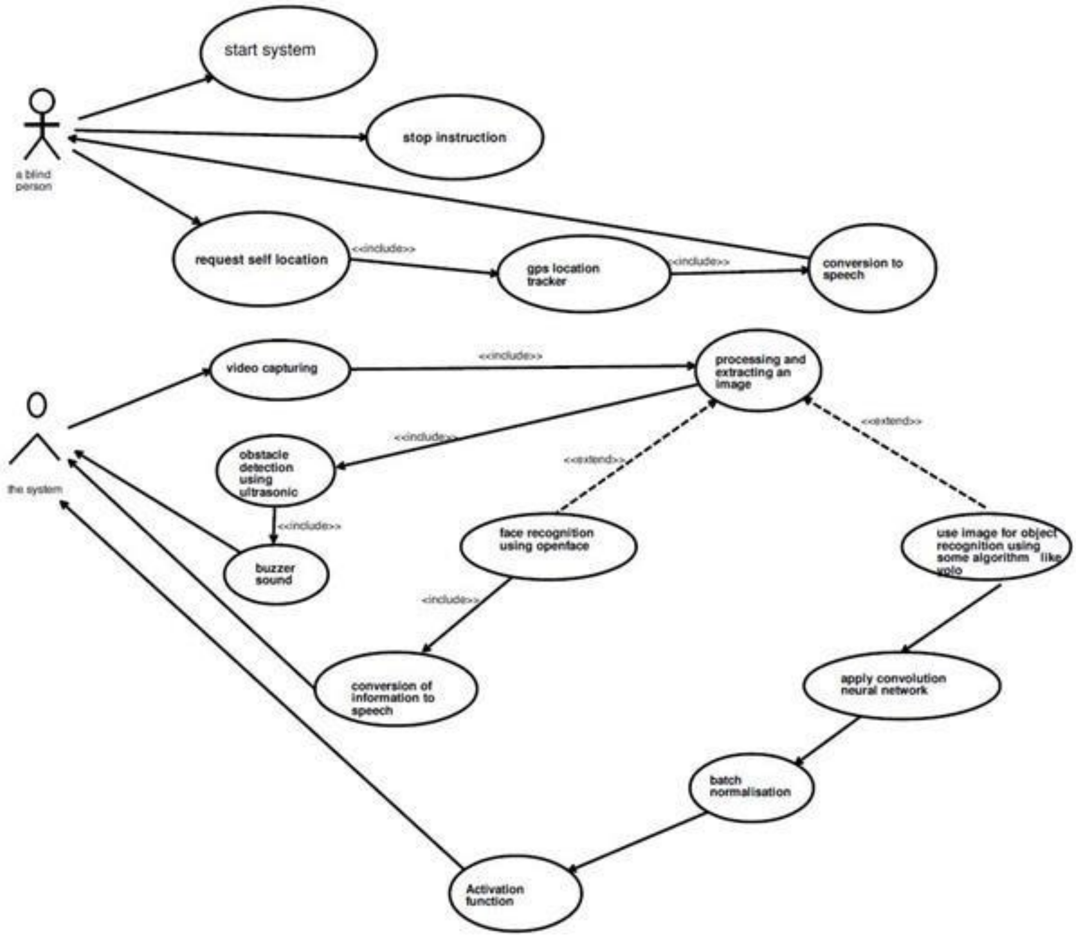

Fig.4.3.1. Usecase Diagram.

Description:

The use case diagram after detection of the object will go through test cases where its appropriate object will get mapped and its output is given by using voice detection.

4.3.2 CLASS DIAGRAM

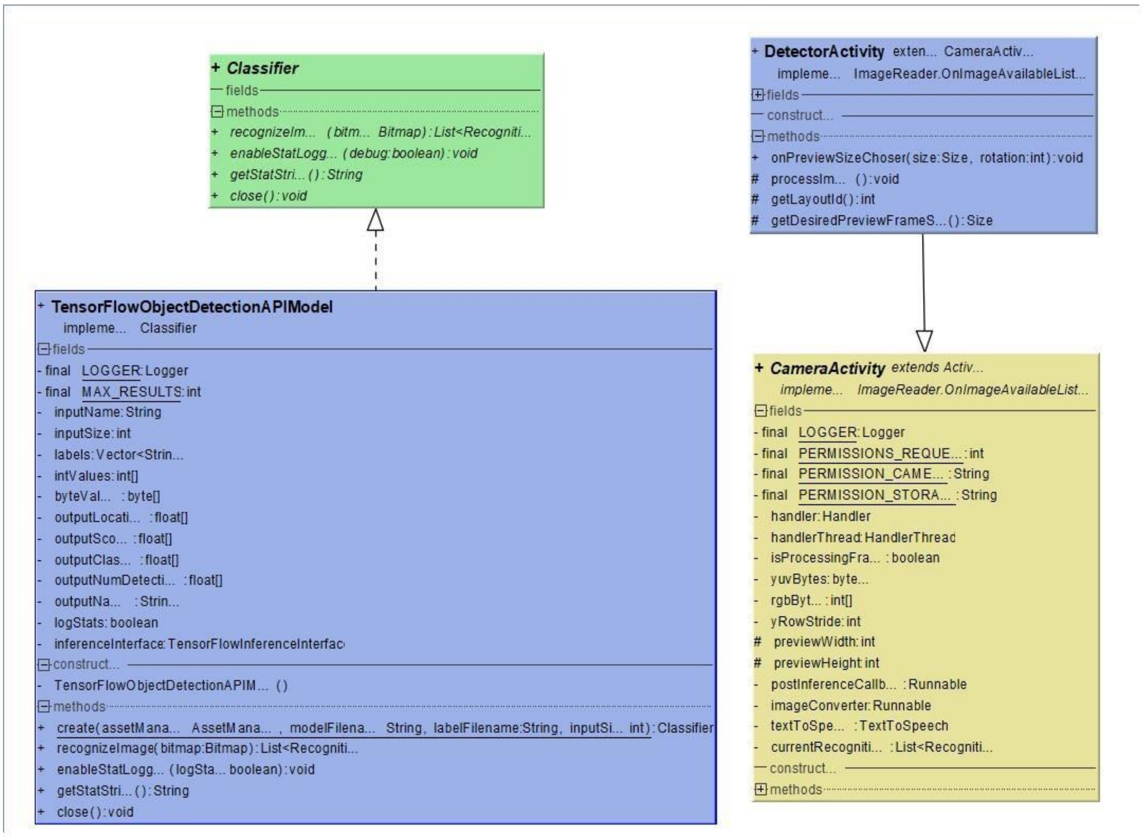

Fig.4.3.2. Class Diagram.

Description:

There is a connection between laptop and the camera and the process takes place on the laptop by using the software Open-CV the respected object is identified and is voice output is given.

CHAPTER 5

SYSTEM ARCHITECHTURE

5.1 SYSTEM ARCHITEDCTURE

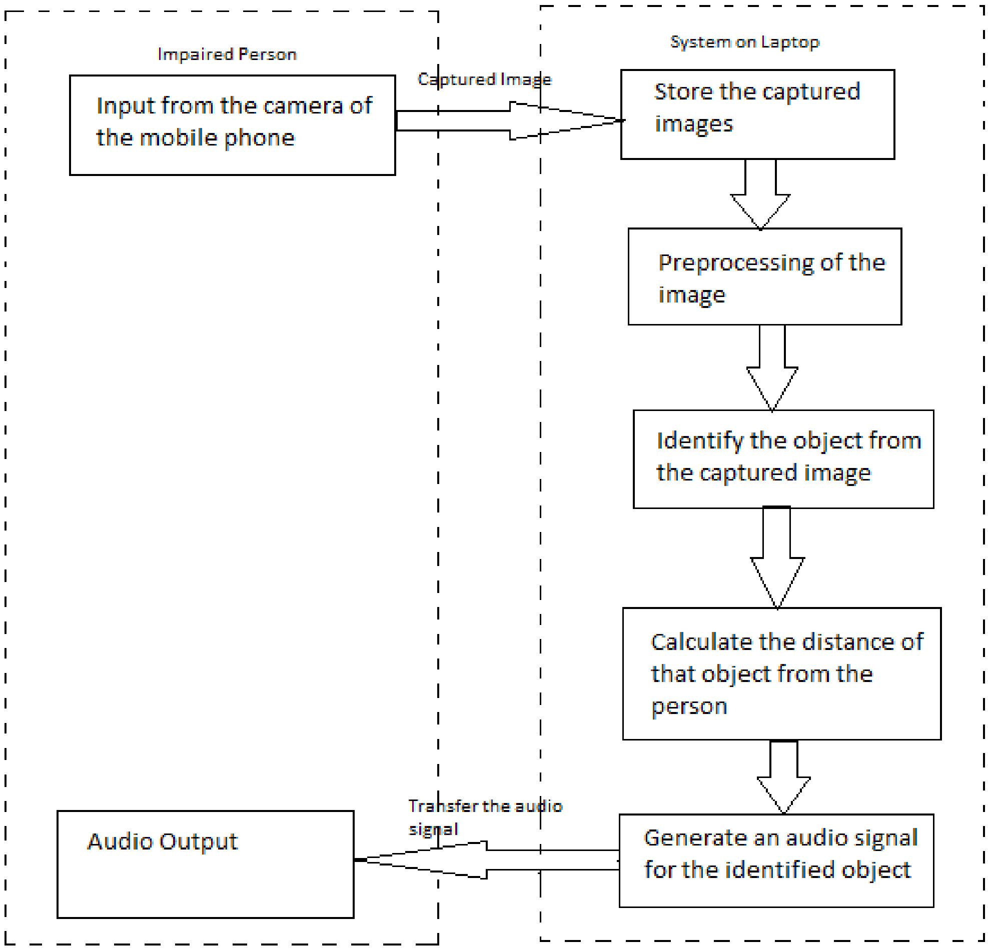

Fig.5.1. System Architecture.

5.2 MODULE DESCRIPTION

The Modules Involved Are

1.SSD ALGORITHM

SSD (Single Shot MultiBox Detector) is a popular algorithm in object detection. It's generally faster than Faster RCNN.

2.OBJECT DETECTION

Object Detection is a technology of deep learning, where things, human, building, cars can be detected as object in image and videos.

3.DEPTH ESTIMATION

Depth estimation or extraction feature is nothing but the techniques and algorithms which aims to obtain a representation of the spatial structure of a scene. In simpler words, it is used to calculate the distance between two objects.

4.VOICE GENERATION

After the detection of an object, it is utmost important to acknowledge the person about the presence of that object on his/her way. For the voice generation module PYTTSX3 plays an important role. Pyttsx3 is a conversion library in Python which converts text into speech.

5.MOBILENET

This model is based on the ideology of THE MobileNet model based on depthwise separable convolutions and it forms a factorized Convolutions. These converts a basic standard convolutions into a depthwise convolutions.This 1 × 1 convolutions are also called as pointwise convolutions. For MobileNets to work, these depthwise convolutions applies a general single filter based concept to each of the input channels.

CHAPTER 6

SYSTEM IMPLAMENTATION

SYSTEM IMPLEMENTATION:

```
import numpy as np
import os
import six.moves.urllib as urllib
import urllib.request as allib
import sys
import tarfile
import tensorflow as tf
import zipfile
import time
import pytesseract
import engineio
import torch
from torch.autograd import Variable as V
import models as models
from torchvision import transforms as trn
from torch.nn import functional as F
import pyttsx3
engine =pyttsx3.init()
from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image
arch = 'resnet18'
model_file = 'whole_%s_places365_python36.pth.tar' % arch
if not os.access(model_file, os.W_OK):
    weight_url =
```

```
'http://places2.csail.mit.edu/models_places365/' + model_file
    os.system('wget ' + weight_url)


#= label_map_util.create_category_index(categories)
pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files (x86)\\Tesseract-OCR\\tesseract'
from utils import label_map_util
#/object_detection/'   m2
from utils import visualization_utils as vis_util
MODEL_NAME = 'ssd_inception_v2_coco_2017_11_17'
MODEL_FILE = MODEL_NAME + '.tar.gz'
DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')
NUM_CLASSES = 90


if not os.path.exists(MODEL_NAME + '/frozen_inference_graph.pb'):
        print ('Downloading the model')
        opener = urllib.request.URLopener()
        opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
        tar_file = tarfile.open(MODEL_FILE)
        for file in tar_file.getmembers():
          file_name = os.path.basename(file.name)
          if 'frozen_inference_graph.pb' in file_name:
            tar_file.extract(file, os.getcwd())
        print ('Download complete')
else:
```

```
        print ('Model already exists')
detection_graph = tf.Graph()
with detection_graph.as_default():
  od_graph_def = tf.compat.v1.GraphDef()
  with tf.io.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)
#
url='http://10.67.208.240:8080//shot.jpg'
import cv2
cap = cv2.VideoCapture(0)

with detection_graph.as_default():
  with tf.compat.v1.Session(graph=detection_graph) as sess:
   ret = True
   while (ret):
      ret,image_np = cap.read()

      if cv2.waitKey(20) & 0xFF == ord('b'):

          cv2.imwrite('opencv'+'.jpg', image_np)
      model_file = 'whole_%s_places365_python36.pth.tar' % arch
```

```
    if not os.access(model_file, os.W_OK):
        weight_url = 'http://places2.csail.mit.edu/models_places365/' +
model_file
        os.system('wget ' + weight_url)
    useGPU = 1
    if useGPU == 1:
        model = torch.load(model_file)
    else:
        model = torch.load(model_file, map_location=lambda storage, loc:
storage) # model trained in GPU could be deployed in CPU machine like this!
    model.eval()
    centre_crop = trn.Compose([
        trn.Resize((256,256)),
        trn.CenterCrop(224),
        trn.ToTensor(),
        trn.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ])
    file_name = 'categories_places365.txt'
    if not os.access(file_name, os.W_OK):
        synset_url =
'https://raw.githubusercontent.com/csailvision/places365/master/categories_plac
es365.txt'
        os.system('wget ' + synset_url)
    classes = list()
    with open(file_name) as class_file:
        for line in class_file:
            classes.append(line.strip().split(' ')[0][3:])
    classes = tuple(classes)
    img_name = 'opencv.jpg'
```

```
if not os.access(img_name, os.W_OK):
    img_url = 'http://places.csail.mit.edu/demo/' + img_name
    os.system('wget ' + img_url)

img = Image.open(img_name)
input_img = V(centre_crop(img).unsqueeze(0), volatile=True)
logit = model.forward(input_img)
h_x = F.softmax(logit, 1).data.squeeze()
probs, idx = h_x.sort(0, True)

print('POSSIBLE SCENES ARE: ' + img_name)
engine.say("Possible Scene may be")
engine.say(img_name)
for i in range(0, 5):
    engine.say(classes[idx[i]])
    print('{}'.format(classes[idx[i]]))



# Expand dimensions since the model expects images to have shape: [1, None, None, 3]
image_np_expanded = np.expand_dims(image_np, axis=0)
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
# Each box represents a part of the image where a particular object was detected.
boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
# Each score represent how level of confidence for each of the objects.
# Score is shown on the result image, together with the class label.
scores = detection_graph.get_tensor_by_name('detection_scores:0')
```

```
    classes = detection_graph.get_tensor_by_name('detection_classes:0')
    num_detections = detection_graph.get_tensor_by_name('num_detections:0')
    # Actual detection.
    (boxes, scores, classes, num_detections) = sess.run(
      [boxes, scores, classes, num_detections],
      feed_dict={image_tensor: image_np_expanded})
# Visualization of the results of a detection.
    if cv2.waitKey(2) & 0xFF == ord('a'):
      vis_util.vislize_boxes_and_labels_on_image_array(
      image_np,
      np.squeeze(boxes),
      np.squeeze(classes).astype(np.int32),
      np.squeeze(scores),
      category_index,
      use_normalized_coordinates=True,
      line_thickness=8)
    else:
      vis_util.visualize_boxes_and_labels_on_image_array(
        image_np,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=8)
    if cv2.waitKey(2) & 0xFF == ord('r'):
      text=pytesseract.image_to_string(image_np)
```

```
        print(text)
        engine.say(text)
        engine.runAndWait()
      for i,b in enumerate(boxes[0]):



       #car    bus    truck
       if classes[0][i] == 3 or classes[0][i] == 6 or classes[0][i] == 8:
         if scores[0][i] >= 0.5:
           mid_x = (boxes[0][i][1]+boxes[0][i][3])/2
           mid_y = (boxes[0][i][0]+boxes[0][i][2])/2
           apx_distance = round(((1 - (boxes[0][i][3] - boxes[0][i][1]))**4),1)
           cv2.putText(image_np, '{}'.format(apx_distance),
(int(mid_x*800),int(mid_y*450)), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255,255,255), 2)


           if apx_distance <=0.5:
             if mid_x > 0.3 and mid_x < 0.7:
               cv2.putText(image_np, 'WARNING!!!', (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0,0,255), 3)
               print("Warning -Vehicles Approaching")
               engine.say("Warning -Vehicles Approaching")
               engine.runAndWait()


       if classes[0][i] ==44:
           if scores[0][i] >= 0.5:
               mid_x = (boxes[0][i][1]+boxes[0][i][3])/2
               mid_y = (boxes[0][i][0]+boxes[0][i][2])/2
```

```
            apx_distance = round(((1 - (boxes[0][i][3] - boxes[0][i][1]))**4),1)
            cv2.putText(image_np, '{}'.format(apx_distance),
(int(mid_x*800),int(mid_y*450)), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255,255,255), 2)
            print(apx_distance)
            engine.say(apx_distance)
            engine.say("units")
            engine.say("BOTTLE IS AT A SAFER DISTANCE")
             if apx_distance <=0.5:
               if mid_x > 0.3 and mid_x < 0.7:
                  cv2.putText(image_np, 'WARNING!!!', (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0,0,255), 3)
                  print("Warning -BOTTLE very close to the frame")
                  engine.say("Warning -BOTTLE very close to the frame")
                  engine.runAndWait()
      if classes[0][i] ==1:
         if scores[0][i] >= 0.5:
            mid_x = (boxes[0][i][1]+boxes[0][i][3])/2
            mid_y = (boxes[0][i][0]+boxes[0][i][2])/2
            apx_distance = round(((1 - (boxes[0][i][3] - boxes[0][i][1]))**4),1)
            cv2.putText(image_np, '{}'.format(apx_distance),
(int(mid_x*800),int(mid_y*450)), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255,255,255), 2)
            print(apx_distance)
            engine.say(apx_distance)
            engine.say("units")
            engine.say("Person is AT A SAFER DISTANCE")
            if apx_distance <=0.5:
               if mid_x > 0.3 and mid_x < 0.7:
```

```
                    cv2.putText(image_np, 'WARNING!!!', (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0,0,255), 3)
                    print("Warning -Person very close to the frame")
                    engine.say("Warning -Person very close to the frame")
                    engine.runAndWait()
#      plt.figure(figsize=IMAGE_SIZE)
#      plt.imshow(image_np)
     #cv2.imshow('IPWebcam',image_np)
     cv2.imshow('image',cv2.resize(image_np,(1024,768)))
     if cv2.waitKey(2) & 0xFF == ord('t'):
        cv2.destroyAllWindows()
        cap.release()
        break
#{1: {'id': 1, 'name': 'person'}, 2: {'id': 2, 'name': 'bicycle'}, 3: {'id': 3, 'name':
'car'}, 4: {'id': 4, 'name': 'motorcycle'}, 5: {'id': 5, 'name': 'airplane'}, 6: {'id': 6,
'name': 'bus'}, 7: {'id': 7, 'name': 'train'}, 8: {'id': 8, 'name': 'truck'}, 9: {'id': 9,
'name': 'boat'}, 10: {'id': 10, 'name': 'traffic light'}, 11: {'id': 11, 'name': 'fire
hydrant'}, 13: {'id': 13, 'name': 'stop sign'}, 14: {'id': 14, 'name': 'parking
meter'}, 15: {'id': 15, 'name': 'bench'}, 16: {'id': 16, 'name': 'bird'}, 17: {'id': 17,
'name': 'cat'}, 18: {'id': 18, 'name': 'dog'}, 19: {'id': 19, 'name': 'horse'}, 20: {'id':
20, 'name': 'sheep'}, 21: {'id': 21, 'name': 'cow'}, 22: {'id': 22, 'name': 'elephant'},
23: {'id': 23, 'name': 'bear'}, 24: {'id': 24, 'name': 'zebra'}, 25: {'id': 25, 'name':
'giraffe'}, 27: {'id': 27, 'name': 'backpack'}, 28: {'id': 28, 'name': 'umbrella'}, 31:
{'id': 31, 'name': 'handbag'}, 32: {'id': 32, 'name': 'tie'}, 33: {'id': 33, 'name':
'suitcase'}, 34: {'id': 34, 'name': 'frisbee'}, 35: {'id': 35, 'name': 'skis'}, 36: {'id':
36, 'name': 'snowboard'}, 37: {'id': 37, 'name': 'sports ball'}, 38: {'id': 38, 'name':
'kite'}, 39: {'id': 39, 'name': 'baseball bat'}, 40: {'id': 40, 'name': 'baseball glove'},
41: {'id': 41, 'name': 'skateboard'}, 42: {'id': 42, 'name': 'surfboard'}, 43: {'id': 43,
'name': 'tennis racket'}, 44: {'id': 44, 'name': 'bottle'}, 46: {'id': 46, 'name': 'wine
glass'}, 47: {'id': 47, 'name': 'cup'}, 48: {'id': 48, 'name': 'fork'}, 49: {'id': 49,
'name': 'knife'}, 50: {'id': 50, 'name': 'spoon'}, 51: {'id': 51, 'name': 'bowl'}, 52:
{'id': 52, 'name': 'banana'}, 53: {'id': 53, 'name': 'apple'}, 54: {'id': 54, 'name':
'sandwich'}, 55: {'id': 55, 'name': 'orange'}, 56: {'id': 56, 'name': 'broccoli'}, 57:
{'id': 57, 'name': 'carrot'}, 58: {'id': 58, 'name': 'hot dog'}, 59: {'id': 59, 'name':
```

'pizza'}, 60: {'id': 60, 'name': 'donut'}, 61: {'id': 61, 'name': 'cake'}, 62: {'id': 62, 'name': 'chair'}, 63: {'id': 63, 'name': 'couch'}, 64: {'id': 64, 'name': 'potted plant'}, 65: {'id': 65, 'name': 'bed'}, 67: {'id': 67, 'name': 'dining table'}, 70: {'id': 70, 'name': 'toilet'}, 72: {'id': 72, 'name': 'tv'}, 73: {'id': 73, 'name': 'laptop'}, 74: {'id': 74, 'name': 'mouse'}, 75: {'id': 75, 'name': 'remote'}, 76: {'id': 76, 'name': 'keyboard'}, 77: {'id': 77, 'name': 'cell phone'}, 78: {'id': 78, 'name': 'microwave'}, 79: {'id': 79, 'name': 'oven'}, 80: {'id': 80, 'name': 'toaster'}, 81: {'id': 81, 'name': 'sink'}, 82: {'id': 82, 'name': 'refrigerator'}, 84: {'id': 84, 'name': 'book'}, 85: {'id': 85, 'name': 'clock'}, 86: {'id': 86, 'name': 'vase'}, 87: {'id': 87, 'name': 'scissors'}, 88: {'id': 88, 'name': 'teddy bear'}, 89: {'id': 89, 'name': 'hair drier'}, 90: {'id': 90, 'name': 'toothbrush'}}

```
#open("yolo-coco/coco.names").read().strip().split("\n")
```

CHAPTER 7

SYSTEM TESTING

SYSTEM TESTING

7.1 SYSTEM TESTING

At first, we are capturing real time images from the rear camera of the mobile handset of blind people and a connection is established between camera and system in laptop and then those images are sent from the camera to laptop.

This connection is done by a laptop software which is installed in the laptop of the person. All the real time images which get captured by the rear camera are first transferred to the in laptop where they are processed for some further conclusions.

The system in laptop will test it using its APIs and SSD ALGORITHM and it detects the confidence accuracy of the image which it is testing. We reached 98% accuracy for certain classes like books, cups, remote.

After testing the images we are generating an output on the laptop based system and its prediction is being translated into voice with voice modules and sent to the blind person with the help of wireless audio support tools.

7.2 TEST CASES

S.NO	TEST CASES	OUTPUT	EXPECTED OUTPUT	STATUS
1	Object detected– accuracy 45<=100	Detected object	Detected object with accuracy 60<=100	Pass/fail.
2	Distance of the person from the object	Distance isestimated	Warning if its close else safe toproceed	Pass/fail.
3	Voice feed back which is obtained fromobject detection	Object is converted toaudio feedback	Audio feedbackis obtained	Pass/fail.

Table 7.2 Test cases

CHAPTER 8

RESULT AND DISCUSSION

8.1 RESULT AND DISCUSSION

Object detection is the emphasis of the proposed system. Portable technology has been developed. The person's chest is hooked to the system. Using TENSORFLOW, the laptop camera captures video of the scene, which is subsequently translated into frames by the processor. The auditory output from the system directs the user to the object. The system displays the object's name its probability as a percentage. As a result, the system will only detect items with a probability more significant than the set threshold.. It shows the object's name as well as its probability. The program also informs the user of the class identification and the distance between the object and the camera over the device's speakers.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENTS

9.1 CONCLUSION AND FUTURE ENHANCEMENTS

Several technologies have been created to aid visually impaired persons. One such attempt is that we would wish to make an Integrated Machine Learning System that allows the blind victims to identify and classify real-time objects generating voice feedback and distance. Which also produces warnings whether they are very close or far away from the thing. For visually blind folks, this technology gives voice direction. This technique has been introduced specifically to assist blind individuals. The precision, on the other hand, can be improved. Furthermore, the current system is based on the Android operating system; it can be altered to work with any device that is convenient.

In future GPRS is installed for better outdoor navigation of the user.

CHAPTER 10

APPENDICES

APPENDICES

10.1 SCREENSHOTS

Fig.10.1.Object (Remote) is detected.

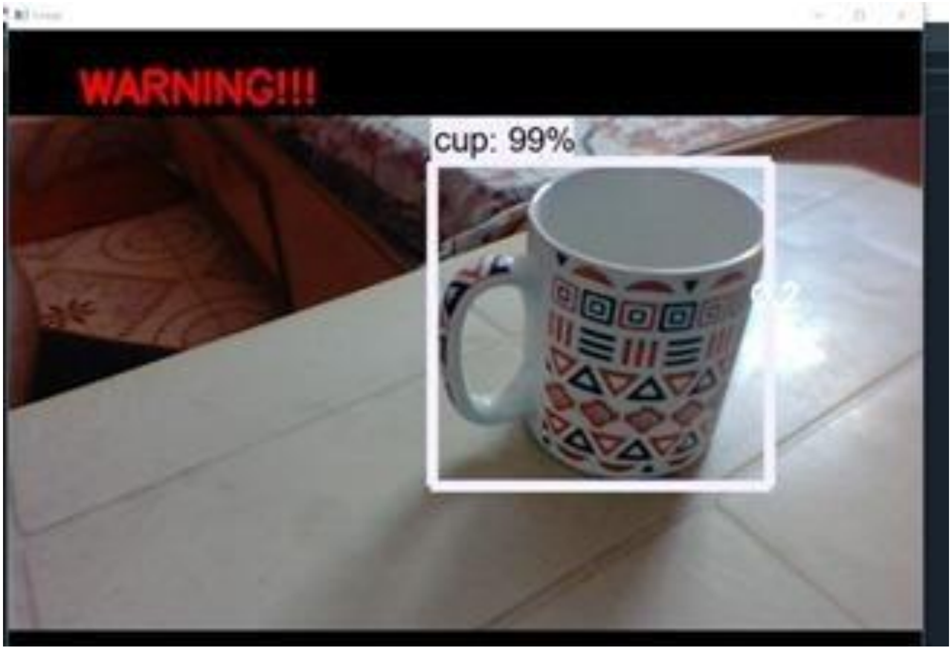

Fig.9.1.1Object(Cup)is detected.

Fig.9.1.2.Object (Bed) is detected.

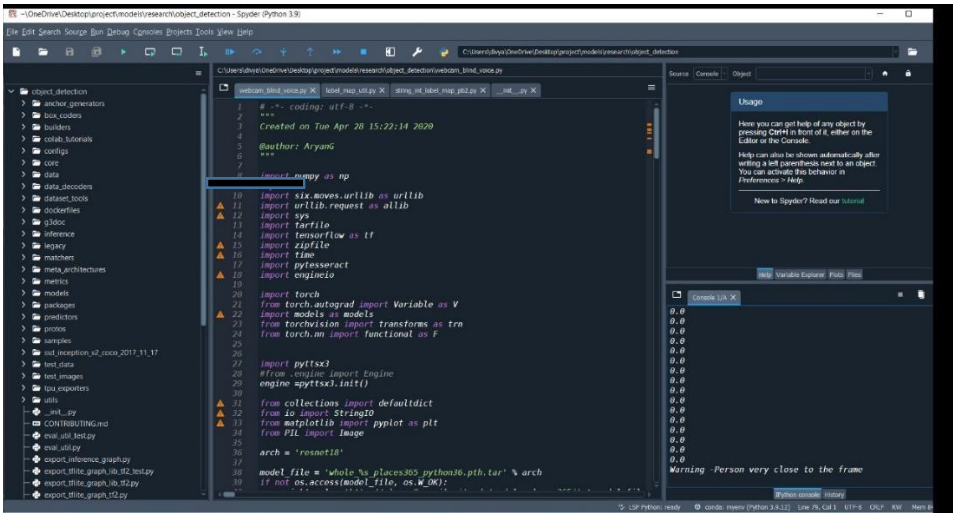

Fig.9.1.3.Program on Spyder.

CHAPTER 10

REFERENCES

REFERENCES

[1] Choi D., and Kim M. (2018). Trends on Object DetectionTechniques Based on Deep Learning, Electronics and Telecommunications Trends, 33(4): 23-32.

[2] Dai Jet al., (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. Conf. NeuralInform. Process. Syst.,Barcelona,Spain,Dec. 4-6, p. 379-387.

[3] Dalal N. and Triggs B., Histograms of oriented gradients for Human Detection (2015). IEEE Comput. Soc.Conf. Comput. Vision Pattern Recogn.,San Diego, CA, USA, June 20-25, p. 886-893.

[4] Russakovsky O et al., (2015). ImageNet Large Scale Visual Recognition Challenge, Int..J.Comput. Vision, 115(3): 211-252.

[5] Rajeshvaree Ravindra Karmarkar (2021). Object Detection System for the Blind with Voice Guidance, International Journal of Engineering Applied Sciences and Technology,6(2): 67-70.

[6] A. Ali and M. A. Ali, "Blind navigation system for visually impaired using windowing-based mean on Microsoft Kinect camera," in *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, pp. 1–4, Beirut, Lebanon, 2017.

[7] H. Takizawa, A. Kitagawa, and M. Aoyagi, "Stereovision cane system: obstacle detection and seat recognition for the visually impaired," *IIEEJ Transactions on Image Electronics and Visual Computing*, vol. 6, no. 2, pp. 74–81, 2018.

[8] T. Schwarze, M. Lauer, M. Schwaab, M. Romanovas, S. Bohm, and T. Jurgensohn, "An intuitive mobility aid for visually impaired people based on stereo vision," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 409–417, Santiago, Chile, 2015.

[9]Q. Nguyen, H. Vu, T. Tran, and Q. Nguyen, "A vision-based system supports mapping services for visually impaired people in indoor environments," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pp. 1518–1523, Singapore, 2014.