**Source Code**

**4.1 Implementation Code**

Fig.5: Object detection source Code

import cv2  
import numpy as np  
  
config\_file = r'config\_files//YOLO\_config.cfg'  
YOLO\_model = r'config\_files//yolov3.weights'  
model = cv2.dnn.readNet(YOLO\_model, config\_file)  
  
classLabels = []  
file\_name = 'label.txt'  
with open(file\_name, 'rt') as fpt:  
 classLabels = fpt.read().rstrip('\n').split('\n')  
  
# print(len(classLabels))  
  
cap = cv2.VideoCapture(0)  
cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH,854)  
cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 480)  
  
count = 0  
while True:  
 \_, img = cap.read()  
 height, width, \_ = img.shape  
  
 blob = cv2.dnn.blobFromImage(img, 1 / 255, (416, 416), (0, 0, 0), swapRB=True, crop=False)  
  
 model.setInput(blob)  
  
 output\_layers\_names = model.getUnconnectedOutLayersNames()  
 layerOutputs = model.forward(output\_layers\_names)  
  
 boxes = []  
 confidences = []  
 class\_ids = []  
  
 for output in layerOutputs:  
 for detection in output:  
 scores = detection[5:]  
 class\_id = np.argmax(scores)  
 confidence = scores[class\_id]  
 if confidence > 0.6:  
 center\_x = int(detection[0] \* width)  
 center\_y = int(detection[1] \* height)  
 w = int(detection[2] \* width)  
 h = int(detection[3] \* height)  
  
 x = int(center\_x - w / 2)  
 y = int(center\_y - h / 2)  
  
 boxes.append([x, y, w, h])  
 confidences.append((float(confidence)))  
 class\_ids.append(class\_id)  
  
 indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)  
  
 font = cv2.FONT\_HERSHEY\_COMPLEX  
 colors = np.random.uniform(0, 255, size=(len(boxes), 3))  
  
  
 if len(indexes) > 0:  
  
 for i in indexes.flatten():  
  
 x, y, w, h = boxes[i]  
 label = str(classLabels[class\_ids[i]])  
 confidence = str(round(confidences[i], 2))  
 color = colors[i]  
 cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)  
 cv2.putText(img, label + " " + confidence, (x, y), font, 1, (0, 255, 0), 1)  
 if(label != 'person'):  
 print(label)  
 count = 0  
 elif(count > 6):  
 print('bring object in the frame')  
 count += 1  
  
  
 cv2.imshow('Image', img)  
  
 key = cv2.waitKey(2)  
 if key==27:  
 break  
  
 if cv2.getWindowProperty("Image", cv2.WND\_PROP\_VISIBLE) < 1:  
 break  
  
cap.release()  
cv2.destroyAllWindows()

Fig.6: Speech code

import speech\_recognition as sr  
import pyttsx3  
  
engine = pyttsx3.init('sapi5')  
voices = engine.getProperty('voices')  
engine.setProperty('voice',voices[1].id)  
engine.setProperty('rate',128)  
engine.setProperty('volume',1)  
  
def speak(audio):  
 engine.say(audio)  
 engine.runAndWait()  
  
  
def record\_audio():  
 r = sr.Recognizer()  
 print("Starting recognition....")  
 with sr.Microphone() as source:  
 r.adjust\_for\_ambient\_noise(source=source,duration=0.5)  
 audio\_data = r.record(source,duration=5)  
 print("Recognizing....")  
 try:  
 text = r.recognize\_google(audio\_data)  
 print(text)  
 except sr.UnknownValueError:  
 print('Sorry, I did not get that')  
 except sr.RequestError:  
 print('Sorry, my speech service is down')  
 return text  
  
speak("Opening mini project on Object detection for blind")

**4.2 Labels of class IDs**

person  
bicycle  
car  
motorbike  
aeroplane  
bus  
train  
truck  
boat  
traffic light  
fire hydrant  
stop sign  
parking meter  
bench  
bird  
cat  
dog  
horse  
sheep  
cow  
elephant  
bear  
zebra  
giraffe  
backpack  
umbrella  
handbag  
tie  
suitcase  
frisbee  
skis  
snowboard  
sports ball  
kite  
baseball bat  
baseball glove  
skateboard  
surfboard  
tennis racket  
bottle  
wine glass  
cup  
fork  
knife  
spoon  
bowl  
banana  
apple  
sandwich  
orange  
broccoli  
carrot  
hot dog  
pizza  
donut  
cake  
chair  
sofa  
pottedplant  
bed  
diningtable  
toilet  
tvmonitor  
laptop  
mouse  
remote  
keyboard  
cell phone  
microwave  
oven  
toaster  
sink  
refrigerator  
book  
clock  
vase  
scissors  
teddy bear  
hair drier  
toothbrush