**Convert Text to Speech Recognition**

Code:

import pyttsx3  
text\_speak = pyttsx3.init()  
  
answer = input("What do you want me to say to you ? :\n")  
text\_speak.say(answer)  
text\_speak.runAndWait()

**Convert text to Audio player response**

Code:

from playsound import playsound  
  
  
name = input("How are you :")  
if(name)=="Good":  
 playsound(r"C:\\Users\\Lenovo\Documents\\Sound recordings\\exc.mp3")  
  
elif (name) == "not good":  
 playsound(r"C:\Users\Lenovo\Documents\Sound recordings\all.mp3")  
elif (name)=="no":  
 playsound(r"C:\Users\Lenovo\Documents\Sound recordings\sorry.mp3")  
else:  
 playsound(r"C:\Users\Lenovo\Documents\Sound recordings\not.mp3")

**Object detection using yolo with voice:**

Code:

import cv2  
from time import sleep  
import numpy as np  
from playsound import playsound  
import pyttsx3  
text\_speak = pyttsx3.init()  
  
net = cv2.dnn.readNet('yolov3.weights','yolov3.cfg')  
classes = []  
with open('coco.names','r') as f:  
 classes = f.read().splitlines()  
  
#print(classes)  
cap = cv2.VideoCapture(0)  
#img = cv2.imread('dog.jpg')  
sleep(2)  
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)  
 net.setInput(blob)  
 output\_layers\_names = net.getUnconnectedOutLayersNames()  
 layerOutputs = net.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 > 0.5:  
 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)  
  
 #print(len(boxes))  
 indexes = cv2.dnn.NMSBoxes(boxes,confidences,0.5,0.4)  
 #print(indexes.flatten())  
 font = cv2.FONT\_HERSHEY\_PLAIN  
 colors = np.random.uniform(0,255,size=(len(boxes),3))  
   
 for i in indexes.flatten():  
 x,y,w,h = boxes[i]  
 label = str(classes[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+20),font,2,(255,255,255),2)  
 print(label)  
 if (label) == "person":  
 text\_speak.say(label)  
 text\_speak.runAndWait()  
 if (label) == "key":  
 text\_speak.say(label)  
 text\_speak.runAndWait()  
 if (label) == "cell phone":  
 text\_speak.say(label)  
 text\_speak.runAndWait()  
 if (label) == "box":  
 text\_speak.say(label)  
 text\_speak.runAndWait()  
  
 cv2.imshow('Image',img)  
 key = cv2.waitKey(1)  
 if key == 27:  
 break  
   
cap.release()   
cv2.destroyAllWindows()