[[1]](#footnote-1)

REAL TIME OBJECT DETECTION WITH VOICE AND DISTANCE CALCULATION

PROF. RANJANA JADHAV , DIVSEHAJ SINGH ANAND , ARYAN KUMAR GUPTA , SHREYAS KHARE , DHEERAJ SHARMA , PRACHI TAPADIYA

DEPARTMENT OF INFORMATION TECHNOLOGY

*Abstract* — *There are about 39 million people who are blind and about 246 million people having low vision across the world. Without the vision its quite difficult to do the daily jobs, and due to which one is always dependent on someone. Our project helps these people who are visually impaired. This paper talks about a Blind assistance system. We have created a wireless system which consists of laptop as server and an android mobile phone as a client. We have created an application which will connect the laptop and mobile phone and has very user friendly interface.*

***Keywords*** *— (OpenCV, Object detection, Pytorch, Computer Vision )*

# INTRODUCTION

In this beautiful world where most of us are blessed with a good vision, there are many people who are visually impaired. These people are always dependent on other people to help them out in everything. Hence to overcome this problem we have created this system.

Our system is called Blind Assistance System. An overview of our system is , there is a laptop which acts a server and processing machine, and a mobile phone which captures the images. The mobile phone captures the real time images through its camera and sends these images to the laptop. The laptop analyizes these images, and sends a voice message to the person. It also tells the distance at which the object is located. To use this efficiently we have created an application which connects the laptop and the mobile. Through this

application one captures the real time images. In our project one can identify objects as well as person in front of it.

Our project uses single shot detection algorithm for detecting multiple objects. In our project we have total 90 classes of objects which are trained and then converted into API’s. by using this SSD algorithm , we are acquiring accuracy of 98% in testing these models.

# Literature Review

Object detection and Navigation for Visually impaired people. This paper gives a system in which there are two cameras on the blind person’s gogles.

Real time obeject detection for blind people.

This paper proposes a application for object detection and creating a 3-D voice of object.

# proposed system

## Proposed Architecture

**Resnet** which stands for Residual Network is basically a convolutional neural network that is designed for hundreds or thousands of convolutional layers. It stores the identity mappings. It skips over the layers , which don’t do anything initially. It reuses the activations from the previous layers. Due to skipping layers , it compresses the network which makes it possible to learn fast.

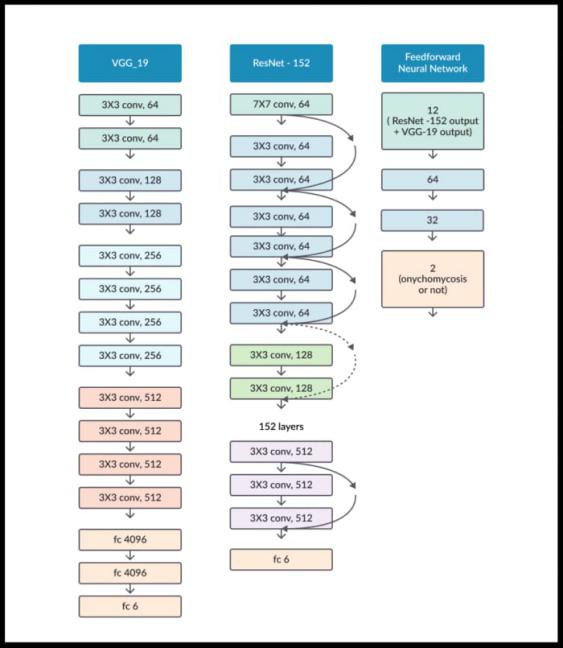


Fig no1: representation of model architecture from resnet.

## System

Our project’s general flow is , first the real images are captured through the mobile phone, these are sent to system, then the processing and testing takes place and finally the output of object detected is given in audio. It also gives the distance at which the object is placed. The given below are flowcharts and an overview of our project.

A screenshot of a cell phone

Description automatically generated

# Methodology/Experimental

## Libraries Used.

1. **OpenCV**

OpenCV (Open Source Computer Vision Library). As name suggests, it is a machine learning and computer vision library. It is mainly used for person, faces, objects detection. It can also be used to track or detect moving objects, it also classifies human action in videos,etc.

We have used this for detecting objects and person through camera.

1. **Matplotlib.**

Matplotlib is a library that produces a graphical representation of the results. It works similar to matlab. This library allows one to do changes in figure by plotting lines, decorating the plot by labels,etc.

1. **Tensorflow**

TensorFlow is mainly used to built deep learning models. It is the library that provides a collection of workflows to develop and train models of deep learning.

1. **Numpy**

Numpy is a library that library supports large amount of matrices. It is used in pyton programming. It also supports multi-dimensional arryas, and also many high-level functions. The foundation of machine learning stack is formed by this library.

1. **Pyttsx3**

This is the library that is used in python for converting text into speech. This library is compatible with python 2 and 3 both and it also works offline.

We have used this library to convert the predicted image's text to speech.

1. **Pytesseract**

Python-tesseract is a library that is same as OCR tool for python. OCR is optical character recognition. It recognizes the words written in the image. It extracts the text from the image and converts it into a editable file.

1. **Pytorch**

Pytorch is used for implementing high level algorithms. Such as CNN, RNN,Resnet, etc. It is a deep learning framework.

we are using this library since resnet architecture Is being used.

1. **Torchvision**

Torchvision consists of various model architectures and datasets. It is also used to transform common images for computer vision.

## TENSORFLOW APIs

Application Programming Interface is the full form of API.The advantage one have by using APIs is it provids us with a set of common operations. Because of which we dont have to write the code for program from scratch.We can say they are quite helpfull as well as efficient.APIs provides us convienenceand hence they are time saver.The TensorFlow object detection API is bascially a structure bulid for creating a deep learning network that solves the problems for object detection.There are trained models in their framework and they refer it as Model Zoo .This includes a collection of COCO dataset, the KITTI dataset, and the Open Images Dataset.These models can be used if we want restrict our model to the categories in these dataset.

## DATASETS

The datsets we have used here is called the COCO datasets. COCO stands for Common Objects in Context. The word context says that the images in the COCO dataset are from day to day life only.COCO was an initiative to collect images which reflect day to day life in short everyday scene and provides contextual information. In everyday life we come accross multiple objects their multiple types,an object can make multiple appereances and multiple objects can be found and each should be labeled as a different object in order to differentiate and segmented properly. Proper labeling and segmentation of the objects in the images is provided by COCO dataset providers.We have taken an advantage of the labeled as well as segmented images to create a better and a relaible object detection model.COCO is a large-scale object detection, segmentation, and captioning dataset. COCO 2017 version contains images, bounding boxes " and labels for the 2017 version.Some images from the train and validation sets don't have annotations. There's a difference in train/val/test splits else Coco 2014 and 2017 uses the same images.There are no notations to test split. (only images).Only 80 classes are used by data though Coco defines 91 classes. Panotptic annotations defines 200 classes but only uses 133.The values of Test/train and validation are 40,670; 118,287 and 5000 respectively.

## D. OBJECT DETECTION ALGORITHM

## D.1 THE SSD ARCHITECTURE

SSD has two components: SSD head and a backbone model. Backbone model basically is a trained image classification network as a feature extractor. Like ResNet this is typically a network trained on ImageNet from which the final fully connected classification layer has been removed. The SSD head is just one or more convolutional layers added to this backbone and the outputs are interpreted as the bounding boxes and classes of objects in the spatial location of the final layers activations.We are hence left with a deep neural network which is able to extract semantic meaning from the input image while preserving the spatial structure of the image albeit at a lower resolution. For an input image ,the backbone results in a 256 7x7 feature maps in ResNet34 . Grid cellInstead of using sliding window, SSD divides the image using a grid and have each grid cell be responsible for detecting objects in that region of the image. Detection objects basically means predicting the class and location of an object within that region. Background class is considered if no object is present and the location is ignored. For instance, we could use a 4x4 grid in the example below diagram.

A close up of a map

Description automatically generated

Each grid cell is able to output shape of the object it and the position.Anchor box and receptive field come into play are used when there are multiple objects in one grid cell or we need to detect multiple objects of different shapes. Anchor box:Multiple anchor/prior boxes can be assigned to each grid cell in SSD. These assigned anchor boxes are pre-defined and each one is responsible for a size and shape within a grid cell.Matching phase is used by SSD while training, so that there's an appropriate match to anchor box with the bounding boxes of each ground truth object within an image. For predicting that object’s class and its location the anchor box with the highest degree of overlap with an object is responsible.Once the network has been trained,this property is used for training the network and for predicting the detected objects and their locations. Practically, each anchor box is specified with an aspect ratio and a zoom level. Well,we know that all objects are not square in shape. Some are shorter ,some are longer and some are wider, by varying degrees. The SSD architecture allows pre-defined aspect ratios of the anchor boxes to account for this.The different aspect ratios can be specified using ratios parameter of the anchor boxes associated with each grid cell at each zoom/scale level.

ZOOM LEVEL

It is not mandatory for the anchor boxes to have the same size as that of the grid cell.The user might be intrested in finding both smaller or larger objects within a grid cell. In order to specify how much the anchor boxes need to be scaled up or down with respect to each grid cell ,the zooms parameter is used.

*D.2 MOBILENET*

The MobileNet model is based on depthwise separable convolutions which are a form of factorized convolutions. These factorize a standard convolution into a depthwise convolution and a 1 × 1 convolution called a pointwise convolution. For MobileNets, the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1 × 1 convolution to combine the outputs of the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers – a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.

A picture containing table

Description automatically generated

*E. DEPTH ESTIMATION*

Depth estimation or extraction refers to the set of techniques and algorithms which aims to obtain a representation of the spatial structure of a scene. In other words, it is used to calculate distance between two real time objects. Our prototype is used to assist the blind people which aims to issue warning to the blind people about the hurdles coming on their way. In order to do this, we need to find that at how much distance the obstacle and person are located in any real time situation. After the object is detected rectangular box is generated around that object. If that object occupies most of the frame then with respect to some constraints the approximate distance of the object from the particular person is calculated.Following code is used to recognize objects and to return the information for the locations and confidence:

(boxes, scores, classes, num\_detections) = sess.run( [boxes, scores, classes, num\_detections], feed\_dict={image\_tensor: image\_np\_expanded})

So, here, for further analysis we can iterate through the boxes. Boxes are an array, inside of an array. So, for iteration we need to define the following conditions.

for i,b in enumerate(boxes[0]):

boxes[0][i][0] – y axis upper start coordinates

boxes[0][i][1] – x axis left start coordinates

boxes[0][i][2] – y axis down start coordinates

boxes[0][i][3] – x axis right start coordinates

Index of box in boxes array is represented by i. Analysis of the score of the box is done by index. It is also used to access class.Now, the width of the detected object is measured. This is done by asking the width of object in terms of pixels.

apx\_distance = round(((1 - (boxes[0][i][3] - boxes[0][i][1]))\*\*4),1)

We get centre of two by subtracting same axis start coordinates and dividing them by two. In this way centre of our detected rectangle is calculated. And at the last, a dot is drawn in the centre. The default parameter for drawing boxes is a score of 0.5. if scores[0][i] >= 0.5 (i.e. equal or more than 50 percent) then we assume that the object is detected. 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)

where mid\_x is centre of X axis and mid\_y is centre of y axis If the distance apx\_distance < 0.5 and if mid\_x > 0.3 and mid\_x < 0.7 then it can be concluded that the object is too close from the particular person.With this code, relative distance of the object from a particular person can be calculated. After the detection of object the code is used to determine the relative distance of the object from the person. If the object is too close then signal or a warning is issued to the person through voice generation module.

*F. VOICE GENERATION MODULE*

Voice generation module plays a crucial role in the proposed system. After the detection of object, it is quite necessary to acknowledge the person about the presence of that object on his/her way. This is done by voice generation module which warns or signs the blind people by generating audio commands which are easily understood by them. After detection of object and its relative distance from the person we have to give voice commands about objects in the path of blind people. Also, if the object is very close then a warning is also issued to the blind person through voice generation module. Audio commands are generated as output. If the object is too close then it states “Warning: The object (class of object) is very close to you. Stay alert!”. Else if the object is at safer distance then then a voice is generated which says that “The object is at safer distance”. This is achieved with the help of certain libraries like pytorch, pyttsx3, pytesseract and engine. For voice generation module PYTTSX3 plays an important role. Pyttsx3 is a text-to-speech conversion library in Python. This library is compatible with both Python 2 and 3 an it works offline. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3 Engine instance. Pyttsx3 is a tool which converts text to speech easily. Two voice modules are supported by pyttsx3. First is female and the second is male which is provided by “sapi5” for windows. An application invokes the pyttsx.init() factory function to get a reference to a pyttsx.Engine instance. During construction, a pyttsx.driver.DriverProxy object is initialized by engine which is responsible for loading a speech engine driver implementation from the pyttsx.drivers module. After construction, an engine object is used by the application to register and unregister event callbacks; produce and stop speech; get and set speech engine properties; and start and stop event loops. Sometimes there is a need to identify the hidden text in the image. For this purpose Python-tesseract is used. Python-tesseract an optical character recognition (OCR) tool for python. OCR detects the text content on images and encodes the text into language which is understood by the computer. This text detection is done by scanning and analysis of the image. Thus, the text embedded in images are recognized and “read” using Python-tesseract. Suppose there are some danger boards on road then the text and symbols hidden in the image of that board are identified and using voice generation module, warning is issued to the person. Pytorch it is primarily a machine learning library. Pytorch is mainly applied to the audio domain. Pytorch helps in loading the voice file in standard mp3 format. It also regulates the rate of audio dimension. Thus, it is used to manipulate the properties of sound like frequency, wavelength and waveform. The numerous availability of options for audio synthesis can also be verified by taking a look at the functions of Pytorch

*G. TESTING*

Third Party App

Third Party App provides ease and freedom in the field of app development. It brings efficiency and also helps in speedy delivery of the output. Third Party App allows you to divide your work in parts and helps you to focus on the core part of app or any system. This strategy helps in the development of good and quality software. We can pass on the Features of the Third Party App to the system.

1) 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 mobile phone and system in laptop and then those images are send from mobile phone to laptop.

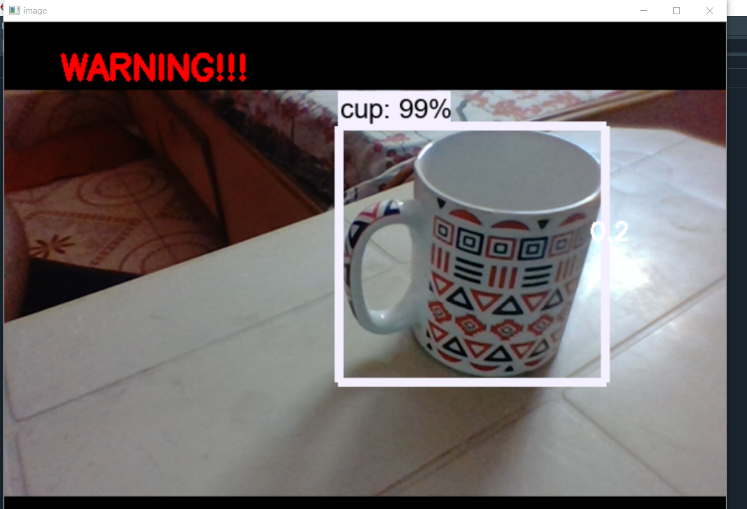
2)This connection is done by a Third party app which is installed in the mobile phone of the person. All the real time images which get captured by the rear camera of mobile phone are first transferred to the Third party app in the mobile phone and then those images are sent in laptop where they are processed for some further conclusions.

3) 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.

4) 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.

## H.OUTPUTS

*H.1 CUP*



***THE ACCURACY OF CUP IS 99%***

***FINAL DISTANCE IS 0.2 UNITS FROM FRAME AND WARNING IS GENERATED WRT DISTANCE AS IT IS TOO CLOSER AND VOICE OUTPUT HEARD THAT IT’S CLASS NAME IS CUP. IT STATES (WARNING -CUP VERY CLOSE TO THE FRAME)***

*H.2 REMOTE*

******

***THE FINAL ACCURACY OF REMOTE IS 98%***

***FINAL DISTANCE IS 0.8 UNITS FROM FRAME AND NO DISTANCE BASD WARNING IS GENERATED AS IT IS AT A SAFER DISTANCE AND CLASS RECOGNITION VOICE CAN BE HEARD AND THE CLASS IS REMOTE.***

*H.3 BED*

******

***THE FINAL ACCURACY OF BED IS 98%***

***FINAL DISTANCE IS 0.9 UNITS FROM FRAME AND NO DISTANCE BASED WARNING AS IT IS AT A SAFER DISTANCE RATHER CLASS RECOGNITION VOICE IS GENERATED AND THE NAME OF THE OBJECT CAN BE HEARD AS IT IS A BED.***

*H.4 CHAIR*

**

***THE FINAL ACCURACY OF CHAIR IS 96%***

***FINAL DISTANCE IS 0.7 UNITS FROM FRAME AND NO DISTANCE BASD WARNING IS GENERATED AS IT IS AT A SAFER DISTANCE AND CLASS RECOGNITION VOICE CAN BE HEARD AND IT’S CLASS IS CHAIR.***

***IT CAN DETECT MULTIPLE FRAMES***

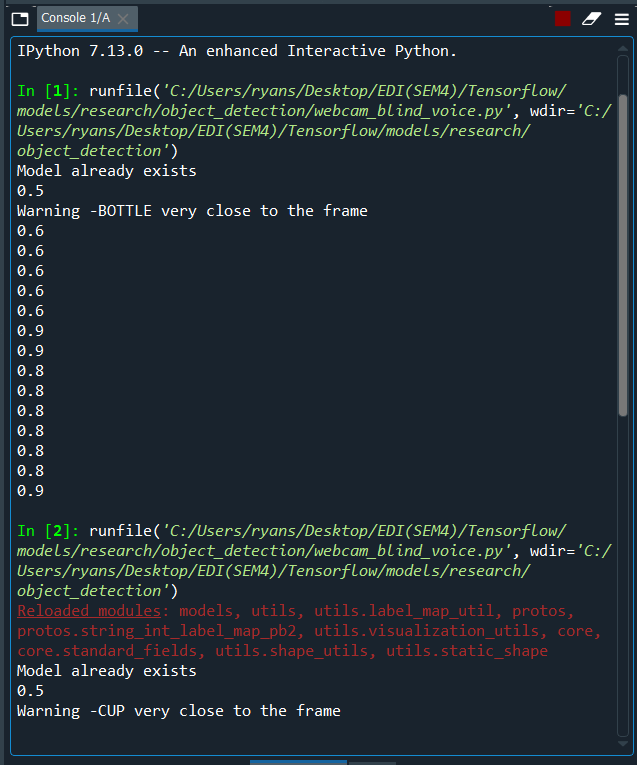
*H.5 TV*

**

***THE FINAL ACCURACY OF TV IS 96%***

***FINAL DISTANCE IS 0.8 UNITS FROM FRAME AND NO DISTANCE BASED WARNING AS IT IS AT A SAFER DISTANCE RATHER CLASS RECOGNITION VOICE IS GENERATED AND THE NAME OF THE OBJECT CAN BE HEARD AS IT IS A TV.***

*H.6 CONSOLES VIEW*

**

***IT CALCULATES DISTANCE AND SHOWS ITS OUTPUT ON CONSOLES AND THE BLIND PERSON CAN HEAR THESE DISTANCE VALUES ALONG WITH OBJECT BASED WARNING.***

# RESULT

The proposed system successfully detects 90 objects,labels them and also shows its accuracy.The model also calculates the distance from the object to the camera and gives a voice feedback as when the person with the camera is approaching the object.The dataset was trained on two different models, SSD Mobilenet V1 and SSD Inception V2.However the SSD Mobilenet V1 model showed less latency and was faster in detecting objects.

# CONCLUSION

The proposed system is an initiative to solve the problems of the blind people.Many devices like ultrasonic sensors, the traditional white cane with sensors are currently being used to aid the blind.The proposed system eliminates many limitations of the old systems for the blind.The system gets the real time image from an application in the mobile and sends it to the model on the laptop which then detects the object, calculates the distance between the person and the object and gives an audio feedback when the person comes near to the object.

# FUTURE SCOPE

Future scope of the proposed system is aimed at getting the real time image from a small camera which would be mounted on a pair of glasses which the blind person would be wearing.This would make the system very feasible and comfortable to use.Moreover we would be making a voice controlled application which would make the user experience better and a feature of adding custom images of objects or people with labels to the dataset.

1. [↑](#footnote-ref-1)