Voice Enable Blind Assistance System -Real time Object Detection

Jigar Parmar[1], Vishal Pawar[2], Babul Rai[3], Prof. Siddhesh Khanvilkar[4]

[1,2,3] Student, Dept. of Information Technology, Pillai HOC College of Engineering & Technology, Rasayani, Maharashtra, India
[4] Prof, Dept. of Information Technology, Pillai HOC College of Engineering & Technology, Rasayani, Maharashtra, India

---***---

Abstract - *Real-time object detection is a difficult operation since it requires more computing power to recognise the object in real time. However, the data created by any real-time system is unlabeled, and effective training frequently necessitates a huge quantity of labelled data. Single Shot Multi-Box Detection is a quicker detection approach for real-time object detection, based on a convolution neural network model proposed in this paper (SSD). The feature resampling stage was eliminated in this work, and all calculated results were merged into a single component. Still, a light-weight network model is required for places with limited processing capability, such as mobile devices (eg: laptop, mobile phones, etc). In this suggested study, a light-weight network model called MobileNet is adopted, which uses depth-wise separable convolution. The usage of MobileNet in conjunction with the SSD model increases the accuracy level in detecting real-time household objects, according to the results of the experiments.*

Words:- **Object Detection, TensorFlow object detection API, SSD with MobileNet**

1. INTRODUCTION

In today's advanced hi-tech environment, the need for self-sufficiency is recognised in the situation of visually impaired people who are socially restricted [3]. Visually impaired people encounter challenges and are at a disadvantage as a result of a lack of critical information in the surrounding environment, as visual information is what they lack the most [1]. The visually handicapped can be helped with the use of innovative technologies. The system can recognise items in the environment using voice commands and do text analysis to recognise text in a hard copy document. It may be an effective approach for blind persons to interact with others and may aid with their independence. Those who are wholly or partially blind are considered visually impaired. According to the World Health Organization (WHO), 285 million people worldwide suffer from vision impairment, 39 people are blind, and around 3% of the population of all ages is visually impaired [1][4]. Visually impaired people go through a lot and encounter a lot of difficulties in their daily lives, such as finding their way and directions, as well as going to places they don't go very often.

2. Existing System

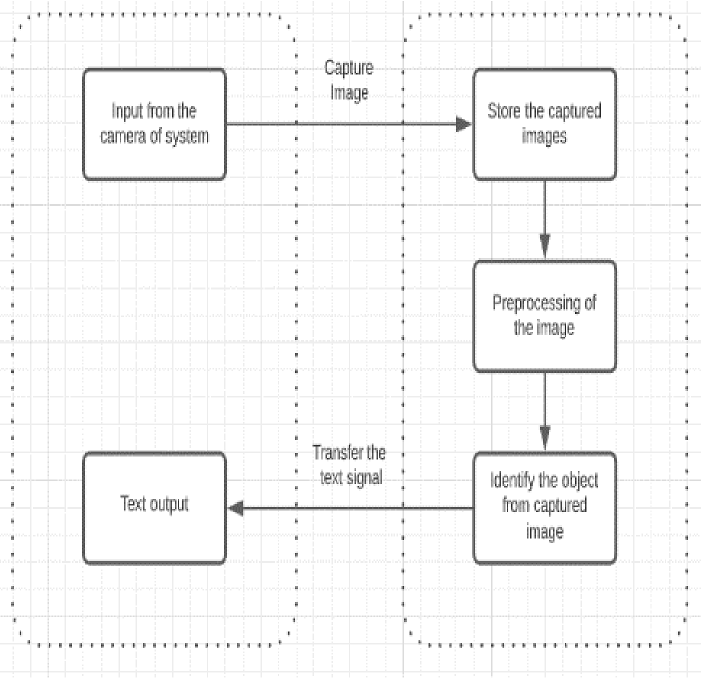

Input from the camera of system
Capture Image
Store the captured images
Preprocessing of the image
Transfer the text signal
Text output
Identify the object from captured image

Fig -1: Flow Chart Of Existing System

In existing system (Fig.1.), system take surrounding information with help of webcam and then store the captured images. These images under goes preprocessing step and then identify the objects from the captured image and after that system will give output in text format [1][3].

3. Working

Visually challenged people, on the other hand, cannot readily go outside for work. They are completely reliant on others. As a result, when they wish to walk outside, they will want assistance [12]. Our system's (Fig.2.) proposed design is based on the recognition of objects in the environment of a blind person. The proposed object/obstacle detection technology works in such a way that it requires various steps from frame extraction to output recognition. To detect items in each frame, a comparison between query frame and database objects is performed [3][8]. We present a system for recognizing and locating objects in photographs and videos. An audio file carrying information about each object detected is activated. As a result, both object detection and identification are addressed at the same time.

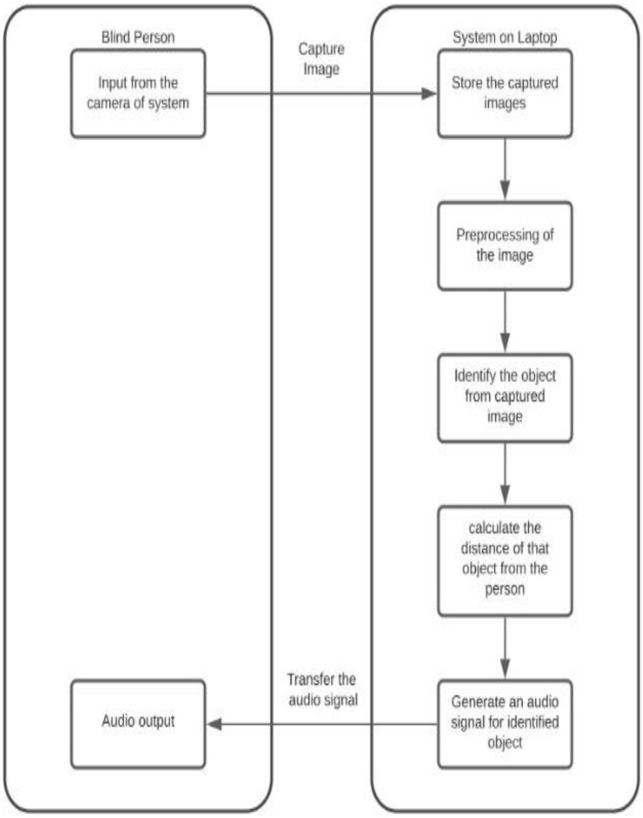

Blind Person
Input from the camera of system
Capture Image
System on Laptop
Store the captured images
Preprocessing of the image
Identify the object from captured image
calculate the distance of that object from the person
Transfer the audio signal
Audio output
Generate an audio signal for identified object

Fig -2: Flow Chart Of System

3.1 Methodology

[1] The system is set up in such a way where an system will capture real-time frames.

[2] The Laptop Based Server will be using a pre-trained SSD detection model trained on COCO DATASET [12]. It will then test and the output class will get detected with an accuracy metrics.

[3] 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.

[4] 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.

3.2 Tensor Flow

TensorFlow APIs were used to implement it. The benefit of using APIs is that they give a collection of common operations [5][9]. As a result, we don't have to write the program's code from start. They are both helpful and efficient, in our opinion. APIs are time savers since they give us with convenience. The TensorFlow object detection API is essentially a mechanism for building a deep learning network that can solve object detection challenges [5][11]. Their framework includes trained models, which they refer to as Model Zoo [3]. This contains the COCO dataset, the KITTI dataset, and the Open Images Dataset, among others. COCO DATASETS are the primary focus here.

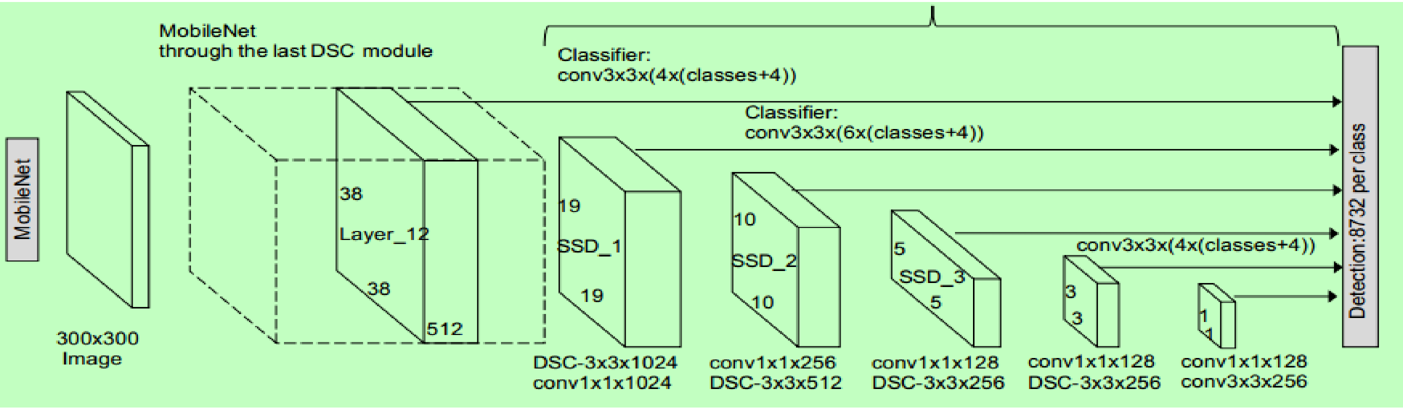


Fig -3: Architecture Of SSD

3.3 SSD

The SSD consists of two parts: an SSD head and a backbone model.

As a feature extractor, the backbone model is essentially a trained image classification network. This is often a network trained on ImageNet that has had the final fully linked classification layer removed, similar to ResNet [3][1].

The SSD head is just one or more convolutional layers added to the backbone, with the outputs read as bounding boxes and classifications of objects in the spatial position of the final layer activations [3].

As a result, we have a deep neural network that can extract semantic meaning from an input image while keeping its spatial structure, although at a lesser resolution.

In ResNet34, the backbone produces 256 7x7 feature maps for an input picture. SSD divides the image into grid cells, with each grid cell being in charge of detecting things in that region [1][7]. Detecting objects entails anticipating an object's class and placement inside a given region.

3.4 MobileNet

This model is based on the MobileNet model's idea of depthwise separable convolutions and generates a factorised Convolutionsv [7]. The depthwise convolutions are created by converting a basic conventional convolution into a depthwise convolution. Pointwise convolutions are another name for these 1 * 1 convolutions. These depthwise convolutions apply a general single filter based notion to each of the input channels for MobileNets to work. These pointwise convolutions use 1 * 1 convolution to combine the depthwise convolutions' outputs. Both filters, like a typical convolution, combine the inputs into a new set of outputs in a single step [1][7]. The depthwise identifiable convolutions partition this into two layers — one for filtering and the other for mixing. This approach of factorization has the effect of dramatically lowering computing time and model size.

3.5. VOICE GENERATION MODULE

Following the detection of an object, it is critical to inform the person on his or her way of the presence of that object. PYTTSX3 is a crucial component of the voice generation module. Pyttsx3 is a Python conversion module for converting text to speech [8]. This library is compatible with Python 2 and 3. Pyttsx3 is a simple tool for converting text to speech.

This technique works in the following way: everytime an item is identified, an approximate distance is calculated, and the texts are displayed on the screen using the cv2 library and the cv2.putText() function. We utilise Python-tesseract for character recognition to find buried text in an image. OCR recognises text content on images and encodes it in a format that a computer can understand [7]. The text is detected by scanning and analysing the image. As a result, Python-tesseract recognises and "reads" text encoded in images. These texts are also linked to a pyttsx.

As an output, audio commands are generated. "Warning: The object (class of object) is too close to you," it says if the thing is too close. And if the object is at a safe distance, a voice is generated that states, "The object is at a safe distance." This is accomplished using libraries like as pytorch, pyttsx3, pytesseract, and engine.io.

Pytorch is a machine learning library first and foremost [7][8]. Pytorch is primarily used in the audio field. Pytorch aids with the loading of the voice file in mp3 format.

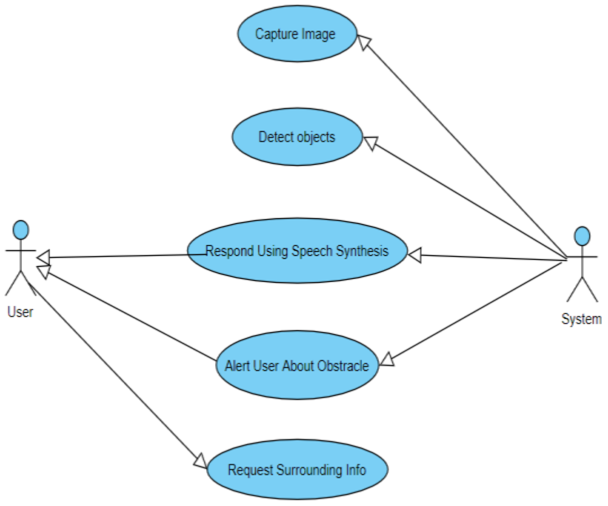


Fig -4: Use Case Diagram Of System

In above Fig.4. as you can see, there are two character, user and system. User will send surrounding information system. And then system will capture the image, detect the object & respond using a speech synthesis. And in last system will give an alert to user about the particular obstracles.

3.6 Image-Processing:

Image processing is a technique for performing operations on a picture in order to improve it or extract relevant information from it [6].

"Image processing is the study and manipulation of a digital image, notably in order to increase its quality," says the fundamental definition of image processing.

3.7 OPENCV:

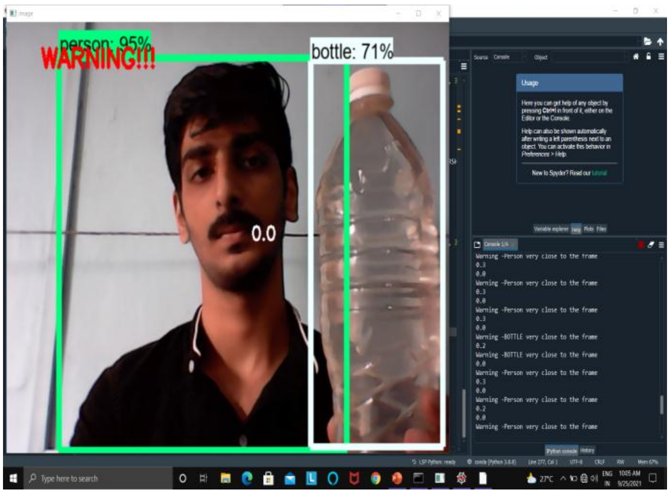
OpenCV is a free, open-source computer vision library. This library includes functions and algorithms for motion tracking, facial identification, object detection, segmentation and recognition, and a variety of other tasks [7]. This library allows you to edit images and real-time video streams to meet your specific needs.

3.8 What is COCO?

The image dataset was built with the objective of improving image recognition, therefore COCO stands for Common Objects in Context. The COCO dataset offers demanding, high-quality visual datasets for computer vision, with the majority of the datasets containing state-of-the-art neural networks [12].

There are a total of 90 predefined objects in this data set.

4. Result



Usage

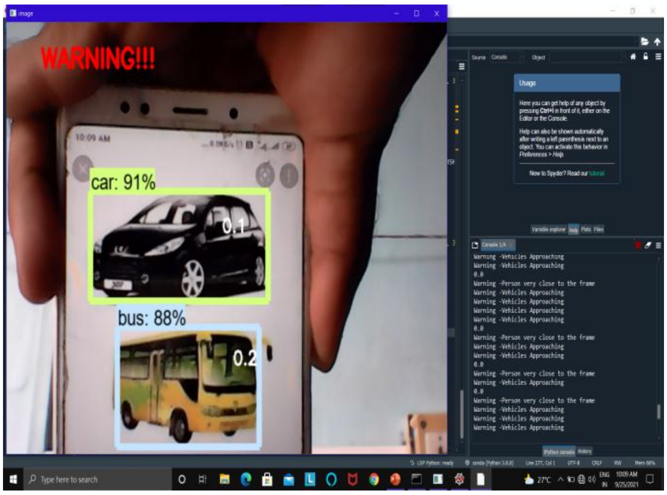
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

```
Warning -Person very close to the frame
0.3
0.0
Warning -Person very close to the frame
0.3
0.0
Warning -Person very close to the frame
0.3
0.0
Warning -BOTTLE very close to the frame
0.2
Warning -BOTTLE very close to the frame
0.0
Warning -Person very close to the frame
0.3
0.0
Warning -Person very close to the frame
0.2
0.0
Warning -Person very close to the frame
```

Fig -5 :- Result 1



Usage

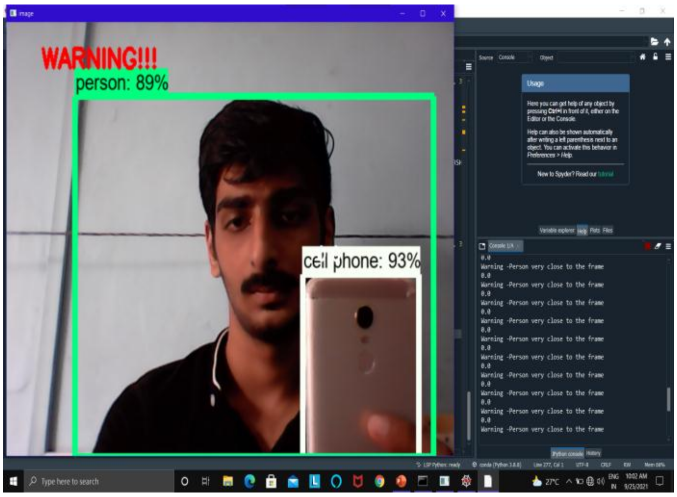
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

```
Warning -Vehicles Approaching
Warning -Vehicles Approaching
0.0
Warning -Person very close to the frame
Warning -Vehicles Approaching
Warning -Vehicles Approaching
Warning -Vehicles Approaching
Warning -Vehicles Approaching
0.0
Warning -Person very close to the frame
Warning -Vehicles Approaching
Warning -Vehicles Approaching
0.0
Warning -Person very close to the frame
Warning -Vehicles Approaching
0.0
Warning -Person very close to the frame
Warning -Vehicles Approaching
Warning -Vehicles Approaching
Warning -Vehicles Approaching
```

Fig -6 :- Result 2



Usage

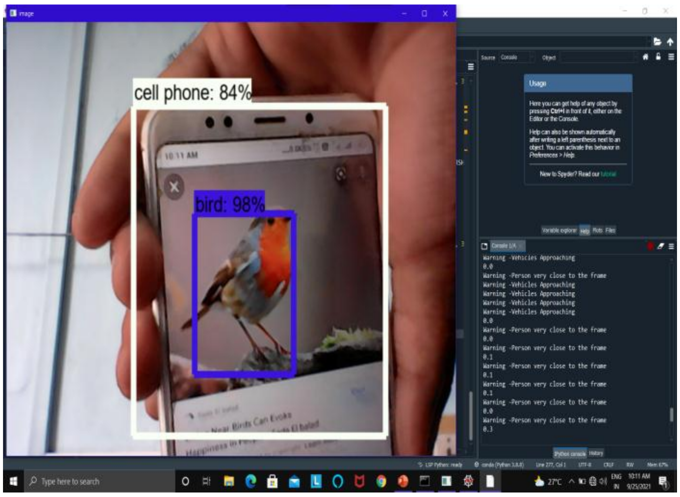
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

```
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
```

Fig -7 :- Result 3



Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

```
Warning -Vehicles Approaching
0.0
Warning -Person very close to the frame
Warning -Vehicles Approaching
Warning -Vehicles Approaching
Warning -Vehicles Approaching
Warning -Vehicles Approaching
0.0
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.1
Warning -Person very close to the frame
0.1
Warning -Person very close to the frame
0.1
Warning -Person very close to the frame
0.0
Warning -Person very close to the frame
0.1
```

Fig -8 :- Result 4

5. Conclusion & Feature

In this research, we attempted to recognise an object that was displayed in front of a webcam. TensorFlow Object Detection API frameworks were used to test and train the created model. Reading a frame from a web camera generates numerous problems, so a good frames per second solution is required to reduce Input / Output concerns [3. As a result, we focused on threading methodology, which considerably improves frames per second and hence greatly reduces processing time for each item. Despite the fact that the application accurately identifies each thing in front of the webcam, it takes roughly 3-5 seconds for the object detected box to move over the next object in the video.

Using this study, we will be able to recognise and track objects in a sports field, allowing the computer to learn deeply, which is a Deep Learning application.

We can manage the traffic signals by detecting the Ambulance Vehicle in the traffic using the Public Surveillance Camera. The technique could be beneficial in a variety of fields in the future, such as reading literature, currency checking, and language translation, among others.

As a result, the project will be beneficial in detecting and tracking items, making life easier.

Acknowledgement

Authors thanks Prof. Siddhesh Khanvilkar, Associate professor at University of Mumbai and Head of Department, Dr. Divya Chirayil at Pillai HOC College of Engineering and Technology, Rasayani, Panvel.

6. Reference

[1] Qianjun Shuai, Xingwen Wu “Object detection system based on SSD algorithm ”. In IEEE , Oct 2020

[2] Harish Adusumalli, D. Kalyani, R.Krishna Sri, M.Pratapteja, P V R D Prasada Rao “Face Mask Detection Using OpenCV”. In IEEE, 2021

[3] Kanimozhi S , Gayathri G , Mala T “Multiple Real-time object identification using Single shot Multi-Box detection”. In IEEE, 2021

[4] World Health Organization, “Visual Impairment and Blindness,” WHO Factsheet no. FS282 , Dec. 2014.

[5] B N Krishna Sai; T. Sasikala “Object Detection and Count of Objects in Image using Tensor Flow Object Detection API”. In IEEE, 10 February 2020

[6] Fares Jalled, “Object Detection Using Image Processing”. In arXiv:1611.07791v1 [cs.CV] 23 Nov 2016

[7] Mr. Harshal Honmote, Mr. Pranav Katta, Mr. Shreyas Gadekar, Mr. Shreyas Gadekar, “Real Time Object Detection and Recognition using MobileNet-SSD with OpenCV”. In IJERT ISSN: 2278-0181 Vol. 11 Issue 01, January-2022

[8] Ayushi Sharma, Jyotsna Pathak, Muskan Prakash, J N Singh, “Object Detection using OpenCV and Python”. In IEEE, 09 March 2022

[9] Priyal Jawale, Hitiksha Patel, Nivedita Rajput, Prof. Sanjay Pawar , “Real-Time Object Detection using TensorFlow”. In IRJET, Aug 2020

[10] Shreyas N Srivatsa, Amruth, Sreevathsa , Vinay , Mr. Elaiyaraja, “Object Detection using Deep Learning with OpenCV and Python”. In IRJET, JAN 2021

[11] Tufel Ali Qureshi, Mahima Rajbhar, Yukta Pisat, Prof. Vijay Bhosale, “AI Based App for Blind People” . In IRJET, March 2021

[12] Dr. S.V. Viraktamath, Madhuri Yavagal, Rachita Byahatti, “Object Detection and Classification using YOLOv3”. In IJERT, February-2021