Chapter 1:

INTRODUCTION

- AIM

  An aid to the blind by providing information about the surroundings, by detecting objects and emotions of the people around.

- SCOPE

  To use a camera and identify faces and give a feedback about the emotion of the face (either happy or sad). To identify objects in the surroundings and give a feedback about the objects. Also tell the accuracy with which it was able to identify the object.

- Organization of report

  Describe what there in 2.1, 2.2, 2.3 etc (in complete report)

  2.1 Features

  2.2 Technical Details

  2.3 Screenshots

  2.4 Hardware and Software Requirements

  2.4.1 Hardware Requirements

  2.4.2 Software Requirements

  3.1 Conclusion and Future Enhancement

  3.2 References

Chapter 2

Description of Project

2.1 Features

1. Training the network :

There is a GUI developed where we get data from olivetti_faces which is inbuilt in sklearn. We have given 2 options to train the network manually. We train it to either detect a happy or a sad face. We are training the network with 400 faces got from olivetti_faces,by which we will be able to identify if the person is either simling or is sad.

2. Emotion Detection :

The project uses a camera where it is able to retrieve the picture of the person in front of it. We use Haarcascade to plot the face in the image .It converts the image to gray-scale and then identifies the mouth of the person. By identifying the mouth it can tell if a person is smiling or if the person is sad. The project is trained to detect the teeth in the face of the person in the picture. This program also calculates the accuracy with which it identifies the emotion based on the dataset.

3. Object Detection :

The other part of the project is object detection. In this module we provide the program with a picture. With this picture the program can identify the object in the picture out of a few selected objects it is trained to identify . Datasets of these objects were used to train the program so that it will be able to identify the object in the image. Using this feature, the program can look into the surroundings and identify these specific objects and give a feedback to the user.

4. Accuracy of identification :

This project can also identify with what percentage accuracy it was able to identify the object in the image. The network has already been trained with the datasets which were obtained . Now that it was able to identify the object it is now trying to find what percentage of the object is actually related to the actual object. This helps in identifying the accuracy with which the program works.

2.2 Technical Details

It has Python interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

2.3 Code

Code in the following link..

- https://goo.gl/GSA1JR

2.4 Screen shot

Fig 1: Emotion Detection : Smiling

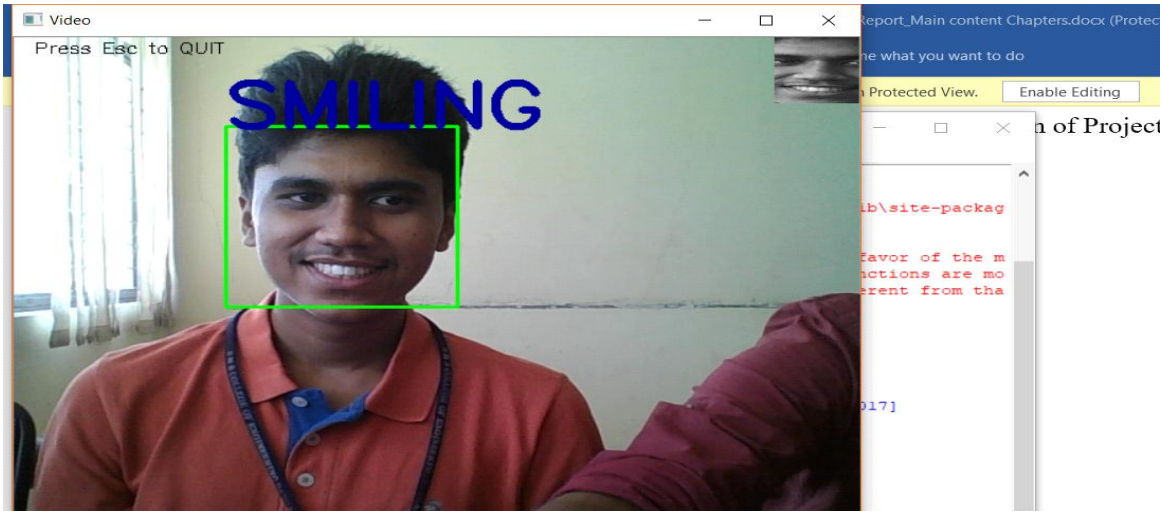
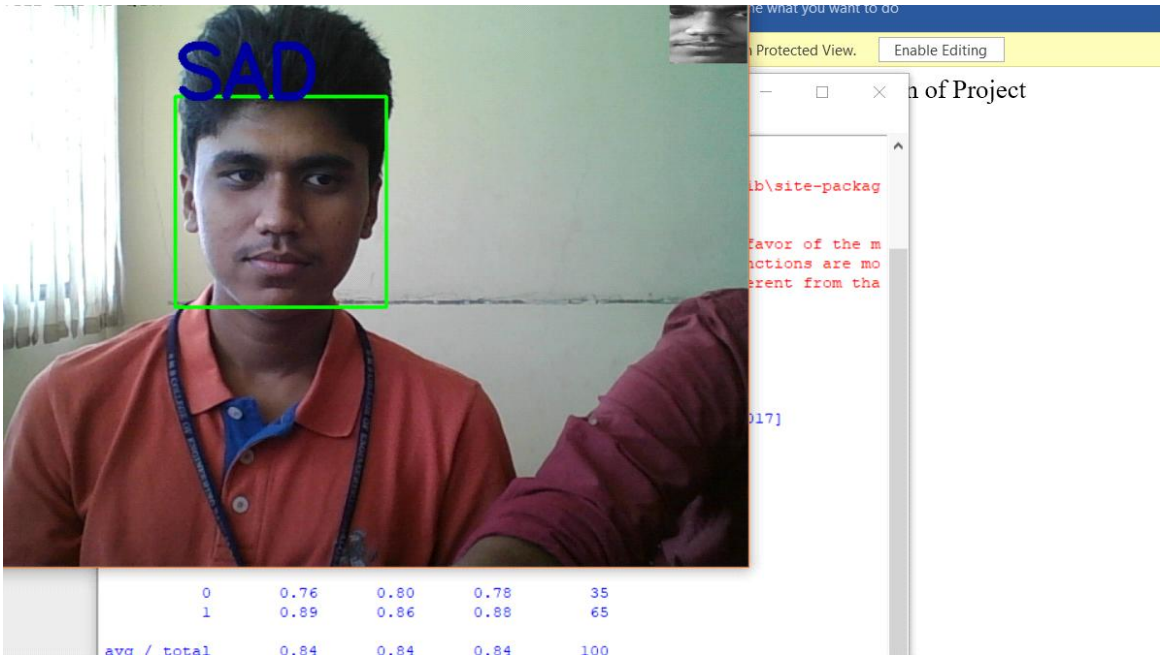


Fig 2 : Emotion Detection : Sad



This screenshot shows the Monitor displaying the emotion of the person in front of it, in the first case smiling, and in the second case sad .The program is trained with 400 image data with which it can identify if a person is smiling. Haarcascade returns the points on the frame where the face is detected. Then it is converted to a gray scale and the mouth is detected. With the trained data the program can predict if the person is smiling or is sad.

Fig 3 : Object Detection : object 1

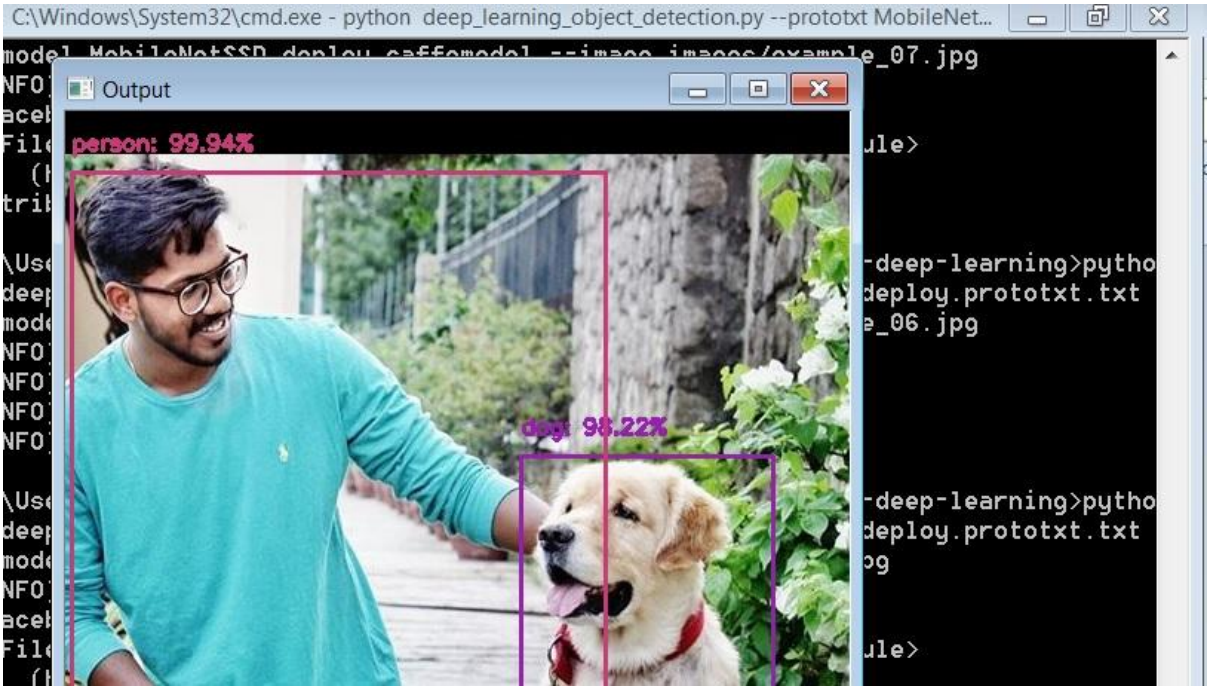


Fig 4 : Object Detection : object2

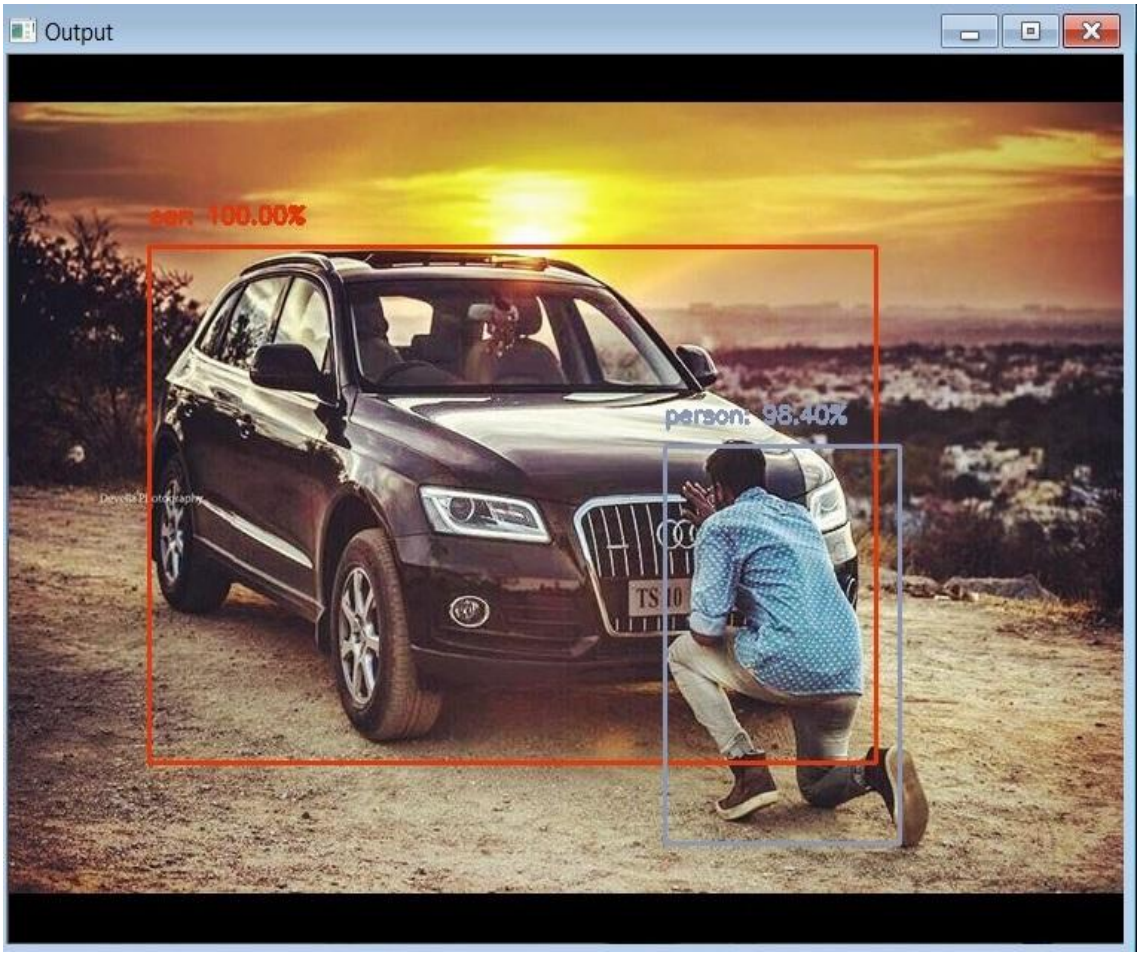


Fig 5 : Object Detection : object 3

The above Screenshots shows us the types of objects in the given images. It is Classified based on the training done. This also shows us the accuracy in the object detection.

- **Hardware & Software Requirement**

2.5.1 Hardware Requirement:

- RAM : 8gb and above
- Processor : 6th Gen i5 and above
- Input Devices : Camera / Webcam
- Output Device: Monitor

2.5.2 Software Requirement:

- Operating System : Windows xp and above
- Software Packages : OpenCV , NumPy ,Python 3.6
- Language : Python

Chapter 3:

Conclusion & Future Enhancement

This report introduced the way in which we can provide assistance to the blind by using the current technologies by which we were able to make our program understand about it is surroundings and give a feedback to the user. We presented information about the various features such as object identification and emotion detection.

As future enhancements to this project we plan on providing audio feedback to the user. We also plan on using the camera on a spectacle (Ex. google lens) to give input of the program about the surroundings.

References

1. OpenCV Tutorials
    - https://pythonprogramming.net/loading-images-python-opencv-tutorial/
2. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
    - https://arxiv.org/abs/1506.01497
3. Mobile Nets: Efficient Convolutional Neural Networks for Mobile Vision Applications
    - https://arxiv.org/abs/1704.04861
4. Object Detection API
    - https://research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html
    - https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html
5. Caffe Article
    - http://caffe.berkeleyvision.org/gathered/examples/mnist.html