

Eye(I) Still Know! – An App for the Blind Built using Web and AI

Akshata Sangwai¹, Shriya Deshmukh², Vardaan Sathe³, Rishika Agarwal⁴ and Rakhi Kalantri⁵

¹Student, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Mumbai, INDIA

²Student, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Mumbai, INDIA

³Student, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Mumbai, INDIA

⁴Student, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Mumbai, INDIA

⁵Assistant Professor, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Mumbai, INDIA

⁵Corresponding Author: rakhikalantri2007@gmail.com

ABSTRACT

This paper proposes eye(I) still know!, a voice control solution for the visually impaired people. The main purpose is even though the blind cannot see they can still know where to go and what to do! Nearly 60% of total blind population across the world is present in India. In a time where no one likes to rely on anyone, this is a small effort to make the blind independent individuals. This can be achieved using wireless communication, voice recognition and image scanning. The application with the use of object identification will priorly inform about the barriers in the path.

The software will use the camera of the device and scan all the obstacles with their corresponding distances from the user. This will be followed by audio instructions through audio output of the device.

This will efficiently direct the user through his/her way.

Keywords-- Speech-to-Text, Text-to-Speech, Voice Recognition, Image Identification, Visually Impaired People, Audio Instructions, Wireless Communication

I. INTRODUCTION

1.1 Introduction Background

India is home to the world's largest number of blind people. Of the 37 million people across the globe who are blind, over 15 million are from India. Science and technology have been making continuous advancements in every field. With the use of recent developments in computer technology we can try to make the world a better place for the blind.

1.2 Motivation

The digital revolution has not reached everyone in the same way. In a world where the next great invention is expected to appear on mobile phone screens, blind and visually impaired people have been left behind. The biggest challenge for a visually challenged person, especially the one with the complete loss of vision, is to navigate around places. They have to depend on their sighted colleagues to navigate through their path. The

recent developments in Artificial Intelligence and Machine Learning can help in guiding visually challenged people through their path.

1.3 Problem Statement and Objectives

Problem Statement: Build an application which will help visually challenged people to navigate through their path with the help of minute-to-minute voice commands.

Voice commands are supported by our Android App "Eye(I) Still Know." When the user launches the app, the app will gain access to the camera and audio settings with the user's permission. A visually impaired person can freely configure and run a set of applications available on a mobile device using touch, gestures, voice commands, and audio feedback. Then the camera will scan the obstacles in a radius of a specific distance and their respective distances. This data will be in form of text which will be converted into audio and using the audio output it will be given to the user through headphones or mic. This will make it easier for them to travel around places on their own. This will boost their self-confidence and make them independent.

1.4 Organization of the Report

This is a report of the paper 'I(eye) still know'. Section 2 has three subsections which inform the reader about the existing papers and systems. It also specifies why the mentioned application is better than the existing systems by listing out their limitations.

Section 3 has six subsections which explain in detail the proposed system. It includes Introduction to proposed system, architecture of the system, algorithm and process design, details of hardware and software, experiments, results, conclusion and future work. Section 3.5 include the implementation screenshots.

II. LITERATURE SURVEY

2.1 Survey of Research Papers

1. *Object detection and identification by Mukkamala Rohith Sri Sai KL University and Sainageshveeravalli*

In various fields, there is a requirement to detect an object and also track them effectively while handling occlusions and other included complexities. Many researchers (Almeida and Guting 2004, Hsiao-Ping Tsai 2011, Nicolas Papadakis and Aure lie Bugeau 2010) attempted for various approaches in object tracking. The nature of the techniques largely depends on the application domain. Some of the research works which made the evolution to proposed work in the field of object tracking. By using this thesis and based on experimental results we are able to detect object more precisely and identify the objects individually with exact location of an object in the picture in x, y axis. This paper also provides experimental results on different methods for object detection and identification and compares each method for their efficiencies.

2. Be My Eyes by Hansjörgenwiberg

This project demonstrates the concept of a messaging and calling system for people who are visually impaired. It enables the removal of environmental barriers for people with a wide range of disabilities. We presented the system designs and use cases for the application "BE MY EYES" in this project. On the Android operating system, it is a universal voice control assistant. The application improves all applications that run on the mobile system. BE MY EYES can provide universal eyes-free and hands-free voice control of mobile devices to a large number of users.

3. A Smart App for Visually Impaired People By Miss. Gowri, Harish Kumar, Gokulkrishna, Bovaz Prabhu

This paper about android-based application helps the blind person to access and use the library resources. It also guides the user to travel from the current location to the destination with the help of GPS navigation using Bluetooth. The objective of this project is to provide the visually impaired with the ease to travel on their own. This application has a specific set of features for object and face recognition and navigation. It also includes speech-to-text converter. This application mainly focuses on the visually impaired people and helps them to live their lives easier by bringing an initiative on android based platform.

4. Face Recognition: A Literature Review By A.S.Tolba, A.H. El-Baz And A.A. El-Harby

This paper provides a current overview of major human face recognition research. It provides an overview of the major human face recognition techniques that are mostly applicable to frontal faces, as well as the benefits and drawbacks of each method. Eigenfaces (eigen features), neural networks, dynamic link architecture, hidden Markov model, geometrical feature matching, and template matching are among the methods considered. The approaches are evaluated based on the facial representations they employ. It discusses two main methods: LEM, SVM, and MCSs. A Line Edge Map

(LEM) method extracts lines as features from a face edge map. This method can be thought of as a hybrid of template matching and geometrical feature matching. The LEM approach not only has the advantages of feature-based approaches, such as invariance to illumination and low memory requirement, but it also has the advantage of high template matching recognition performance. SVM is a learning technique that is widely used for general-purpose pattern recognition due to its high generalisation performance without the need for additional knowledge. A SVM, given a set of points from two classes, finds the hyperplane that separates the greatest possible fraction of points from the same class on the same side while minimising the distance from either class to the hyperplane.

In the field of face recognition, MCSs based on the combination of outputs from a set of different classifiers have been proposed as a method of developing high performance classification systems.

2.2 Existing Systems

1. Blindtool

The features provided by this application are:

- **Object Recognition:** This application scans the objects in the pathway and tells the user what the object is.
- **Image Recognition:** It also provides the feature to scan and identify a particular image and inform the user about the same.
- **Text-To-Speech:** This feature of the application after scanning a particular object or image enables the user to understand what the object or image is by speech.

It does not provide any feature for face and expression recognition. Face and expressions cannot be identified by the features of object and image recognition. Hence, this application lacks one of the important features i.e. face and expression recognition. Also navigation is not taken care of in this application.

2. Sullivan+

The features provided by this app are:

AI Mode:

It uses Camera - Use camera API to control the camera and take photos.

With only one shot, you can get the help that suits the situation.

AI mode-AI automatically finds the best results that fit the taken photos.

Text Recognition

Text Recognition-Finds a text and tells you by sound. Point the camera on the place you want text recognition.

Face Recognition

Face Recognition- It recognizes a person photographed with a camera and predicts the person's age and gender.

Image Recognition

Image Description - Identifies objects around you and creates a sentence that describes the identified scene.

This application has no feature of navigation which is a very important problem for visually impaired people. It doesn't store data for face recognition and identifies everyone as an unnamed person.

3. Be My Eyes

Be My Eyes is a free smartphone app available for both iOS and Android. Through a live video connection, Be My Eyes connects blind and low vision users with sighted volunteers or company representatives for visual assistance. In simpler terms, it enables users to have a sighted person available on demand via video chat. There is no limit to how frequently a user can access Be My Eyes, and the service is completely free. The app can also be accessed from any location in the world, with no language restrictions. People who use Be My Eyes have low vision or are blind. It's difficult to define a typical user because people with vision impairments come from all walks of life and from all corners of the globe.

However, I'll give you some examples of users you might encounter as a volunteer. During calls, both the user and the volunteer remain anonymous.

1. No resources are available for Be My Eyes volunteers to learn how to assist vision-impaired users.
2. One cannot choose a particular assistant.
3. The app is to be used at one's own risk, i.e., you are depending on "helpers" to navigate the world around you.

4. Supersense

The features provided by this app are:

Text Recognition

It reads the text seen by the camera and also gives directional instructions to move the camera in order to find text.

Explores Environment

It explores the environment and verbally tells the objects it scanned. It recognizes different objects and people.

Object Detection

The different options available in the app for find option are seat, door, stairs, trashcan, vehicle, bathroom, animals, etc. Once the option is selected it searches for the object and once spotted tells the user.

The app does not allow navigation. Also, it finds the object needed but for finding one needs to select the option which is not possible for a visually impaired person. Voice

commands are not taken as input. After locating that a particular object is present in the environment it does not provide any information to reach the object.

2.3 Mini-Project Contribution**Rishika Agarawal**

- Object Detection in Pycharm using Python, Tensorflow, Opencv, Numpy (Implementation).
- Making The Presentation Slides Including Literature Survey, Proposed System And Proposing Project Idea

Shriya Deshmukh

- Object Detection In Pycharm Using Python, Tensorflow, Opencv, Numpy (Implementation).
- Making The Presentation Slides Including Abstract, Introduction And Literature Survey.

Akshata Sangwai

- Face Recognition In Pycharm Using Python, Opencv, Numpy And In Making Of The Basic Website Using Django (Implementation).
- Making The Presentation Slides Including Proposed System, Technologies Used And Flowchart.

Vardaan Sathe

- Face Recognition In Pycharm Using Python, Opencv, Numpy (Implementation).

Making the presentation slides including Existing survey.

III. PROPOSED SYSTEM**3.1 Introduction**

Build a web App which will try to help visually challenged people by giving minute to minute updates about the surrounding. Looking forward to help the visually impaired public with this idea. If this turns out to be a success it will surely prove to be helpful to the blind as well as the society.

3.2 Flowchart

Following figure 3.1 is the flowchart for the system. The system performs mainly two functions i.e. Face Identification and Object Identification.

As shown in figure 3.1 as the app starts it reaches the user to ask for camera access and user authentication. It reaches to system for further process.

If an object is identified user is informed. If a person is identified the database is accessed. If the face is present in the database the name is informed to the user but if not a new user is created.

After creating the person the user is informed.

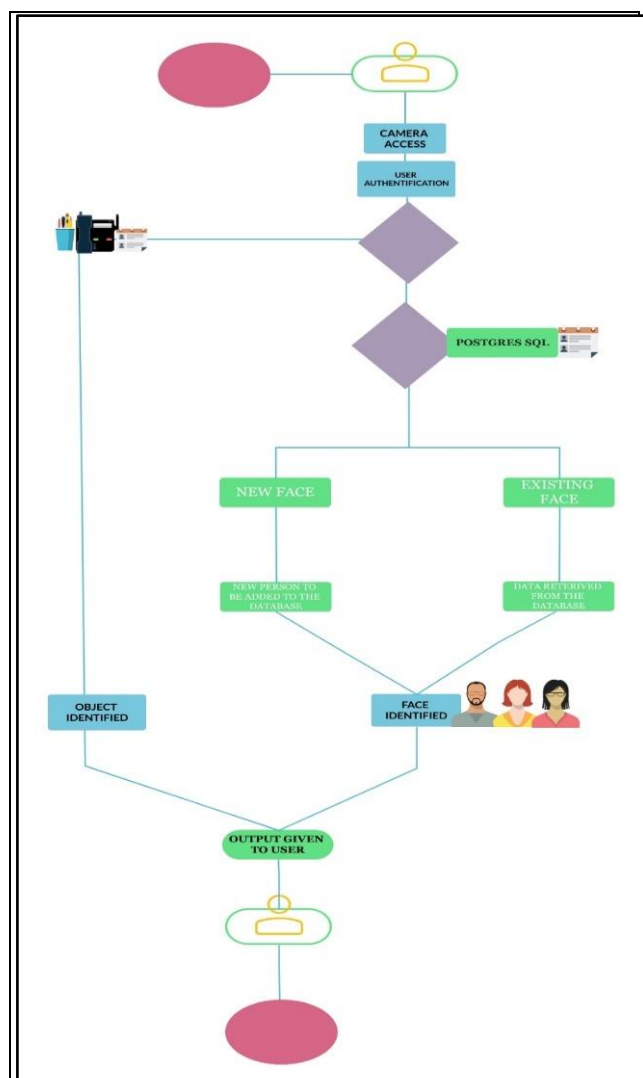


Figure 1: Flowchart

3.3 Architecture/Framework

3.3.1 DJANGO

Django is a high-level Python Web framework that promotes rapid development as well as clean, pragmatic design. It was created by experienced developers to take care of much of the hassle of Web development, allowing you to focus on writing your app instead of reinventing the wheel. It is both free and open source.

- **Ridiculously Fast**

Django was created to assist developers in moving applications from concept to completion as quickly as possible.

- **Reassuringly Secure**

Django prioritises security and assists developers in avoiding many common security blunders.

- **Exceedingly Scalable**

Some of the busiest websites on the Internet make use of Django's ability to scale quickly and flexibly.

3.3.2 Tensorflow

Tensorflow is Google's Open Source Machine Learning Framework for dataflow programming across multiple tasks. The graph's nodes represent mathematical operations, while the graph's edges represent the multidimensional data arrays (tensors) that are communicated between them.

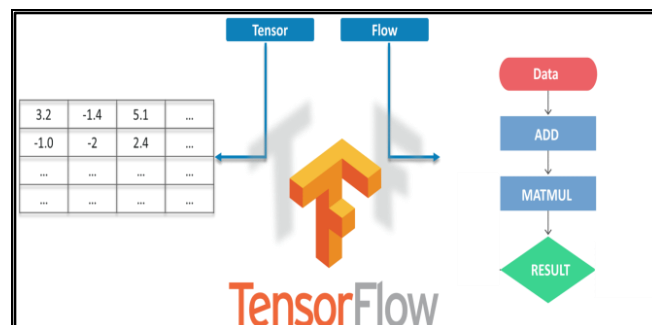


Figure 2: TensorFlow

Tensors are simply multidimensional arrays, which are 2-dimensional tables extended to data with a higher dimension. TensorFlow has many features that make it suitable for Deep Learning. So, without further ado, let's look at how we can use TensorFlow to implement Object Detection.

3.4 Algorithm and Process Design

3.4.1 Object Detection

Every Object Detection Algorithm works in a different way, but they all follow the same principle.

Feature Extraction: They extract features from the input images and use these features to determine the image's class. Whether using MatLab, OpenCV, Viola Jones, or Deep Learning.

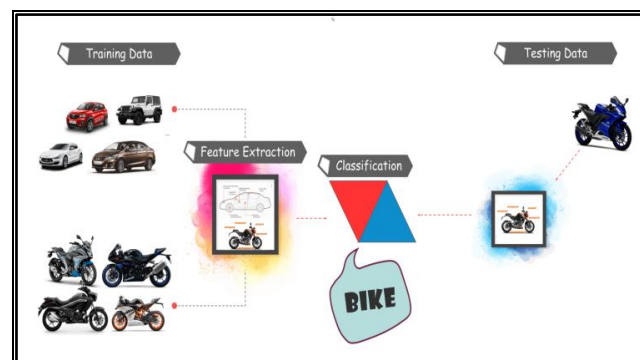


Figure 3: Feature Extraction

Getting Prerequisites

Before working on the Demo, let's have a look at the prerequisites. We will be needing:

- Python

- TensorFlow
- Tensorboard
- Protobuf v3.4 or above

Setting up the Environment

Now to Download TensorFlow and TensorFlow GPU you can use pip or conda commands:

- # For CPU
- pip install tensorflow
- # For GPU
- pip install tensorflow-gpu

For all the other libraries we can use pip or conda to install them. The code is provided below:

1. pip install --user Cython
2. pip install --user contextlib2
3. pip install --user pillow
4. pip install --user lxml
5. pip install --user jupyter
6. pip install --user matplotlib

Then there's Protobuf: Protocol Buffers (Protobuf) are Google's language-neutral, platform-neutral, extensible mechanism for serialising structured data – think of it as a smaller, faster, and simpler version of XML. For this demo, you must download and extract Protobuf version 3.4 or higher.

You must now clone or download TensorFlow's Model from Github. After downloading and extracting, rename the “models-masters” folder to simply “models”.

For the sake of simplicity, we'll group “models” and “protobuf” into a single folder called “Tensorflow.”

Next, we must navigate to the Tensorflow folder, then to the research folder, and run protobuf from there using the following command:

```
1 "path_of_protobuf's bin" ./bin/
  protocobject_detection/protos/
```

To check whether this worked or not, you can go to the **protos** folder inside `models>object_detection>protos` and there you can see that for every proto file there's one python file created.

3.4.2 Face Detection

Face detection is a computer vision problem that involves finding faces in photos. Face detection is a non-trivial computer vision problem for identifying and localizing faces in images and has been solved reasonably well by classical feature-based techniques, such as the Deep Neural Network (DNN).

Deep Neural Network (DNN)

A deep neural network (DNN) is an ANN with multiple layers between the input and output layers. A deep neural network (DNN), or deep net for short, is a neural network with some level of complexity, usually at least two layers. Deep nets use sophisticated math modelling to process data in complex ways. Deep Neural Networks (DNNs) are a powerful class of machine learning algorithms that are built by stacking layers of

neutral networks along the depth and width of smaller architectures. It is based on the Single-shot-multibox detector and employs the ResNet-10 Architecture as its backbone. The DNN-based detector overcomes all of the shortcomings of the Haarcascade-based detector. The progression to DNN is as follows:

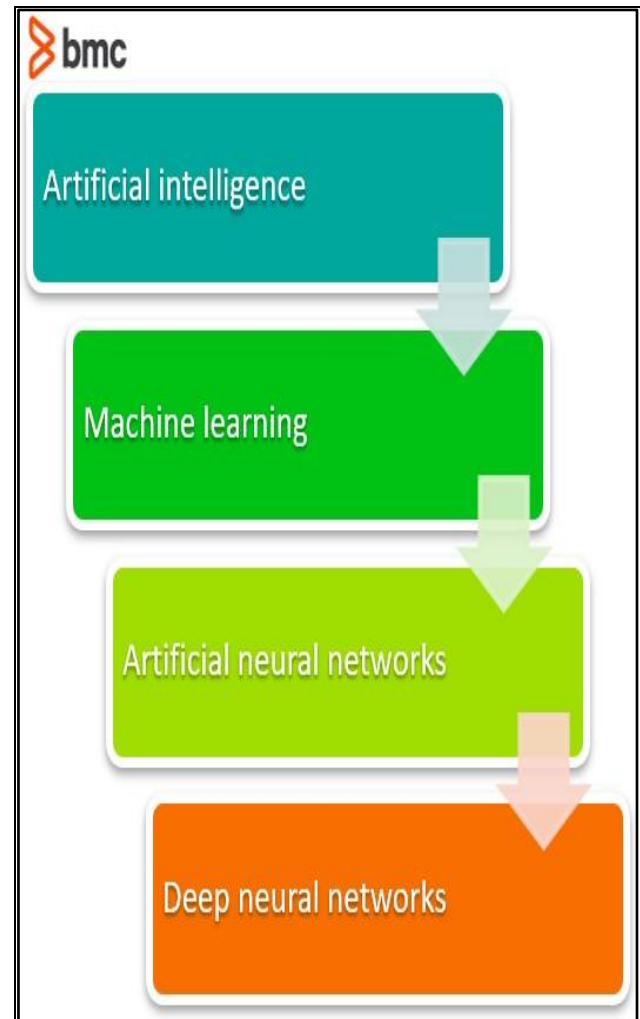


Figure 4: DNN

Deep Neural Network (DNN) models consume a lot of resources, especially when ensembles of multiple models are involved. The deep learning face detector in OpenCV is built on the Single Shot Detector (SSD) framework and a ResNet base network. OpenCV (Open Source Computer Vision) is a library that contains functions aimed primarily at real-time computer vision. Caffe, Tensorflow, and Torch/PyTorch are Deep Learning frameworks supported by OpenCV. Face detection with OpenCV is possible thanks to the library's pre-trained deep learning face detector model. The following libraries are used:

- OpenCV, which supports a lot of algorithms related to Computer Vision and Machine Learning. It is a built in Deep Neural Network.
- Numpy, which is a package used for scientific computing with python.
- OS, which provides a portable way of using operating system dependent functionality.

To use OpenCV Deep Neural Network module with Caffe models two files will be needed:

- prototxt file which defines model architecture
- caffe model file which contains the weights for the actual layers

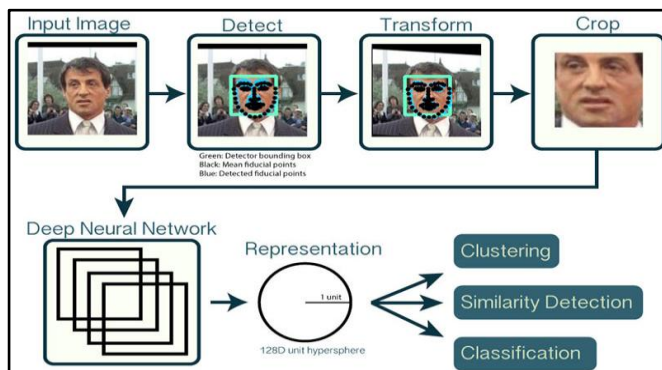


Figure 5: CNN feature extractor

The cascade classifier is made up of several stages, each of which is made up of a group of weak learners. The weak learners are decision stumps, which are simple classifiers. Boosting is a technique used to train each stage. By taking a weighted average of the decisions made by the weak learners, boosting allows you to train a highly accurate classifier.

Each stage of the classifier labels the region defined by the sliding window's current location as positive or negative. Positive indicates that an object was discovered, while negative indicates that no objects were discovered. If the label is negative, this region's classification is complete, and the detector moves the window to the next location. If the label is positive, the region is advanced to the next stage. When the final stage classifies the region as positive, the detector reports an object found at the current window location.

The stages are set up in such a way that negative samples are rejected as quickly as possible. The assumption is that the vast majority of windows do not contain the desired object. True positives, on the other hand, are uncommon and should be verified.

- A True Positive occurs when a positive sample is correctly classified.
- A False Positive occurs when a negative sample is mistakenly classified as positive.

- A False Negative occurs when a positive sample is mistakenly classified as negative.

Each stage of the cascade must have a low false negative rate in order to function properly. If a stage incorrectly labels an object as negative, the classification is halted and the error cannot be corrected. Each stage, however, has a high false positive rate. Even if the detector incorrectly labels a nonobject as positive, the error can be corrected in later stages. Increasing the number of stages lowers the overall false positive rate, but it also lowers the overall true positive rate.

Cascade classifier training necessitates the use of both positive and negative images. To be used as positive samples, you must provide a set of positive images with regions of interest specified. The Image Labeler can be used to label objects of interest with bounding boxes. The Image Labeler generates a table to be used with positive samples. You must also provide a set of negative images from which the function will automatically generate negative samples. Set the number of stages, feature type, and other function parameters to achieve acceptable detector accuracy.

3.5 Details of Hardware and Software

The Hardware requirements are:

- Laptop with integrated graphics card.

The Software requirements are:

- VsCode, PyCham.
- Technologies used:
- Machine Learning, Django and Postgres SQL.

The language used in backend is Python. Postgres SQL is used for managing databases. Machine Learning has been used for the codes of face detection and object detection. Django is a python-based framework. Django is used to retrieve and add data to database.

TensorFlow library is used for object detection. TensorFlow is an open-source ML framework for dataflow across a range of tasks. TensorFlow is a name made of two parts: Tensor which means multi-dimensional array and flow which means a graph of operations.

Pip installed are: Django, psypog, pillow, pylint, cv2, numpy, pycotool, keras, pyaudio

Keras is high level python library run on top of the tensorflow network. It provides an easy-to-use API to interface with various machine learning frameworks.

Cv2 is a library of python designed to solve computer vision problems. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

IV. RESULT AND DISSCUSSION

4.1 Experiment and Results

Screenshots of Implementation



Figure 6: Implementation of Object detection



Figure 7: Implementation of face detection

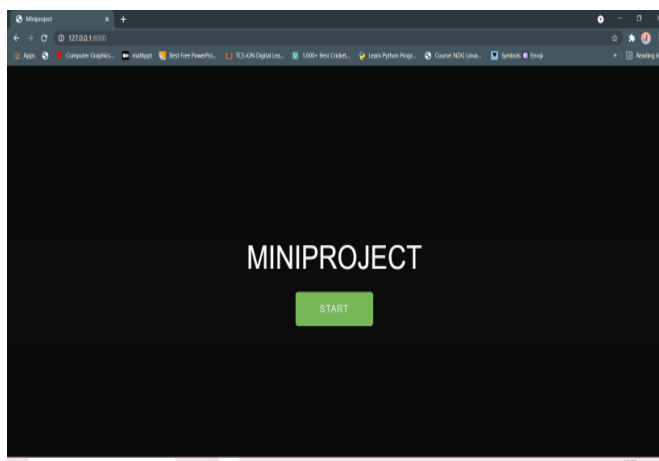


Figure 8: Implementation of Django website

V. CONCLUSION AND FUTURE SCOPE

Visually disabled people can be greatly benefitted from technology and its recent advancements. The major problem faced by such people is figuring out one's environment.

They have to rely on a third person to give them instructions and guide through the path which may lead to mishaps. Now, if the same commands are given by a machine that is trained to recognize objects i.e. the obstacles present on the path, faces of people whom the user knows and a functionality to add such person too, the user will be fully aware of the environment. Due to lack of sight the commands by the machine can be given verbally. This can be made possible using certain libraries in python as this project is using audio.

The implementation of object detection and face recognition is done using Django with the installation of some particular pip and libraries.

The implementation of a website using Django is done using VsCode and connection with database can be done using ORM function of django.

The text to speech is done by python text to speech library. Also, speech to text is done using pyaudio library of python. All the commands are taken as voice input.

This project can be taken to the next level by adding certain more functionalities as location teller, mood detector, ability to read signs, ability to read text(banners, etc.).

REFERENCES

- [1] Crossland, M. D., Silva, R., & Macedo, A. F. (2014). Smartphone, tablet computer, and e-reader use by people with vision impairment. *Ophthalmic and Physiological Optics*, 34(5), 552–557.
- [2] Dim, N. K. & Ren, X. (2014). Designing motion gesture interfaces in mobile phones for blind people. *Journal of Computer Science and Technology*, 29(5), 812–824.
- [3] Wong, M. E., & Tan, S. S. K. (2012). Teaching the benefits of smart phone technology to blind consumers: Exploring the potential of the iPhone. *Journal of Visual Impairment & Blindness*, 106(10), 646 – 650.
- [4] Venugopal, G. (2013). Android Note Manager application for people with visual impairment. *International Journal of Mobile Network Communications & Telematics*, 3(5), 13–18.
- [5] Yu Zhong1 , T.V. Raman2 , Casey Burkhardt2 , Fadi Biadisy2 and Jeffrey P. Bigham. (2014). *JustSpeak: Enabling universal voice control on android*. Available at: <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/41924.pdf>.