

Virtual AI Assistant for Person with Partial Vision Impairment

Rohith Raghavan^{1*}, Vishodhan Krishnan¹, Hitesh Nishad¹, and Bushra Shaikh¹

¹Department of Information Technology, SIES Graduate School of Technology, Navi Mumbai, India

Abstract: Smartphones help us with almost every activity and task nowadays. The features and hardware of the phone can be leveraged to make apps for online payment, content consumption, and creation, accessibility, etc. These devices can also be used to help and assist visually challenged and guide them in their daily activities. As the visually challenged sometimes face difficulty in sensing the objects or humans in the surroundings, they require guidance or help in recognizing objects, human faces, reading text, and other activities. Hence, this Android application has been proposed to help and assist people with partial vision impairment. The application will make use of technologies like face detection, object and text recognition, barcode scanner, and a basic voice-based chatbot which can be used to execute basic commands implemented through Deep Learning, Artificial Intelligence, and Machine Learning. The application will be able to detect the number of faces, recognize the object in the camera frame of the application, read out the text from newspapers, documents, etc, and open the link detected from the barcode, all given as output to the user in the form of voice.

1 Introduction

A Normal person without any disabilities have no issues with daily work in their life. But, on the other hand, it is difficult for a partially blind person to carry out daily tasks. Actions like reading texts, identifying objects cannot be performed by them due to their disability. Making Braille versions of every text is an expensive and tedious task. Also, recognizing objects from a distance is not possible for a visually challenged person. Although there are several applications to help and assist the visually challenged, they offer only some features, making the person install a handful of applications for that. So, to overcome the current issues faced by a visually challenged person, we have developed this application that offers convenience and assistance to the visually challenged person. The application offers text, object recognition, and face detection to identify text, objects, and humans. It also offers a chatbot so that the visually challenged person can interact with the bot for basic information and activities

2 Literature Review

We studied and went through the following research papers listed below to get more knowledge and ideas about the implementation of our project.

Tosun et al [1] discussed the process and the algorithms involved for real-time object detection. They also compared the various algorithms like YOLOv2, SSD, and faster R- CNN in terms of accuracy. The paper explained the ML algorithms in brief. YOLOv2 provided better accuracy and ran on even low fps with a GPU processor.

Tembhurne et al. [2] studied the implementation of a voice assistant for visually challenged. The paper discussed the various modules which can be implemented in the voice assistant like calls, messages, TTS, OCR, etc. The paper also talks about using Maps API for navigation.

*Corresponding author: rohith.raghavan17@siesgst.ac.in

Dahiya et al. [3] elaborates on the R-CNN algorithm in detail and also compares the accuracy and computational time of R- CNN and faster R-CNN combined with resnet-50. The paper also discusses the data preprocessing steps required for feeding the data into the machine learning model. The framework proposed in the paper claims an accuracy of 92%.

Ahmed et al. [4] discussed using RNN (recurrent neural network) and CNN (convolutional neural network) for obstacle avoidance and way-finding. Their work using CNN proved helpful to implement object detection using CNN-based algorithms.

Gianani et al [5] described real-time object detection implemented using OpenCV and also determining the position of the object using Euclidean distance. The paper also guides the user to the objects through voice output. The paper explains object detection using the SSD framework and MobileNet architecture which has an accuracy of 99.61%. This system is designed to work in an indoor environment.

Kukade et al. [6] focused on Speech-to-Text, Text-to-Speech, Optical Character Recognition, and voice assistance and the proposed system to implement the same. The paper also discussed the ways of it

Shishir et al. [7] explained object recognition using Tensorflow ML API along with the implementation of it. They included informative flowcharts for understanding the process behind it. They also explained the working of OCR and object recognition. This implementation provided accuracy of over 80%.

Karthik et al. [8] provided an overview of the OCR algorithm and the hindrances faced while the text is being extracted. They also share the idea of using Raspberry Pi instead of a mobile phone to capture images. The paper also talks about the future scope of using a GPS location tracker for guidance.

Singh et al. [9] proposed an Android application which offers text recognition, speech recognition, image recognition and a chatbot for the user to interact with the application. The paper proposed using Google Cloud APIs (various APIs which can be used to automate tasks) and Google Dialogflow (a natural language understanding platform on which chatbot can be implemented) to implement various modules instead of training deep learning models to perform various activities.

Sharma et al. [10] focuses on implementing a system offering face recognition, text-to-speech and object recognition on a web browser which can be opened on a mobile device. The paper also talks about adding a feature to add unknown faces to the database at the tap of a button for future reference. The proposed system

also has a fairly simple and user-friendly UI designed specifically for the visually impaired.

Jakhete et al. [11] discussed about using Single Shot Detector (SSD) Algorithm to implement Object Detection in an Android Application. The paper lists other object recognition algorithms and mentioned the steps to implement SSD algorithm on an Android application.

3 Existing System

In this section, we are discussing the features of certain applications available on the Play Store

Supersense[12]– it is an application that assists the visually challenged and the features provided by it are Object recognition, Face recognition, and text recognition Sullivan+[13]–This application also serves the same purpose this provides Object Recognition to describe images, Face recognition and text recognition.

Envision AI [14]– This application also serves the same purpose and provides the features that are Face recognition, object recognition

LetSeeApp[15]– This application is also for the same purpose and provides the features of text recognition to read visiting cards as well as credit and debit cards

The above-mentioned applications provide more or less similar features (the links to these applications are provided in the references section)

4 Proposed System

An Android-based application based on technology and innovation promises to academically empower visually challenged by freeing them of their dependence on visuals by providing the information through an app.

This application aims to provide better functionality in an app that makes a partially blind user use it for navigation, identification, recognition, and also gaining information of the outer world. Some of them are listed below:

- The app will contain a chatbot such that we will be asking questions about time, weather, or any other kind to obtain information or asking to perform certain actions the user desires.
- It will detect the objects in real-time and provide the necessary information to the user.
- The app will also contain a barcode scanner which will help the user to get information about certain products.
- The app can also help the user detect human faces so that the user can understand human presence in the surrounding and also the number of people in the room.

- This application will have a text reader which will be used to read the text out loud to the user.

Using the app, the person can get help and guidance in day-to-day tasks and activities.

APP NAME	FR`	OR`	TR`	CHATBOT
SUPERSENSE	YES	YES	NO	NO
SULLIVAN +	NO	YES	YES	NO
LETSEEAPP	NO	YES	YES	NO
ENVISON AI	YES	YES	NO	NO
OUR APPLICATION	YES	YES	YES	YES

Table 1 Comparison of features provided by each application

NOTE- FR`- FACE RECOGNITION; OR`- OBJECT RECOGNITION; TR`-TEXT RECOGNITION

NOTE- As the algorithms used in the other applications are unknown to us .We have contacted their developers but haven't got any responses yet. So, we have done the comparison based on features provided

5 Methods

Face Detection:

Face detection is a computer technology that is used to detect human faces in images, videos, or in real-time video. Face detection is a broad technology that just marks or labels the human face identified by the application. The key difference between face detection and recognition is that face detection just identifies the face whereas face recognition will also label the person's name, gender, age, or other attributes. Face detection can be implemented in various fields - security, biometrics, entertainment, law enforcement, etc.

Basic face detection can be achieved through OpenCV whereas real-time face detection or face detection in different conditions can be achieved using machine learning or deep learning. The face detection algorithms start searching for human eyes in the frame as it is the easiest to detect.

It then searches for other factors like eyebrows, nose, ears, and iris. When the algorithm finds the factors in the image in the frame, it then applies additional tests and then confirms the detection of the face by labelling the face with a rectangular box.

Real-time face detection involves motion; hence traditional algorithms cannot be applied. So, advanced machine learning and deep learning algorithms are used to create models which can detect faces in real-time in various scenarios.

Object Recognition:

Object recognition is the technique to recognize and label an object detected in an image, video, or real-time. Object recognition is achieved using machine learning and deep learning. Object recognition algorithms take the frame from the camera as the input and then apply a bounding box of a specific size to the image and check for the object in the image. If the object is found in the image, the algorithm will recognize the object. There are two steps to object recognition –image classification and object localization. Image classification predicts the class of the object in an image. Whereas object localization identifies one or more objects in the image and drawing the bounding boxes. The object detection algorithm will combine both of the tasks and will classify the objects in the image.

Text Recognition:

Text recognition is the technique to detect and identify the text which is in printed, handwritten or digital format. Text recognition technology converts the text in different forms to digital form. It is also called OCR (Optical Character Recognition). Several APIs exist for various platforms which can be used to implement OCR.

For recognizing typed or printed text on objects or books, the user has to open the application on his smartphone and then select the required option. The application will identify the text and convert it to digital form. The text will then be read out to the user.

Chatbot:

Chatbots are AI-based computer programs that can simulate a human conversation. They are also called digital assistants as the chatbots can be used to do actions and commands given by the user. A chatbot can process the human conversation, reply to commands and queries or can solve user FAQs as well.

The key modules behind a chatbot are artificial intelligence, natural language processing, user-defined rules, and machine learning which are required to process the commands or messages sent by the user and deliver the required feedback.

Chatbots are of two types- task-oriented and data-driven. Task-oriented chatbots are designed for a single purpose and only generate automated responses. Their interaction is specific and restricted to only FAQs or basic questions.

The answers to the queries are already defined in task-oriented chatbots. Hence, they can only handle and process basic queries and are the most commonly used in websites and apps for user queries.

Data-driven chatbots or virtual assistants are more interactive, sophisticated, and advanced than data-driven ones.

These chatbots use NLP, NLU, and ML to learn from the user's queries and responses. These chatbots analyze and use past user interaction data and behavior to provide responses or feedback to the user's queries. Hence, data-driven chatbots get better, efficient, and precise over time. Amazon Alexa, Google Assistant, and Apple's Siri are examples of data-driven chatbots.

Implementation

For implementing object recognition and face detection, we have chosen to use Tensorflow Lite (Google's open-source deep learning framework designed for on-device processing) framework in our proposed system.

TensorFlow Lite was chosen as other frameworks like Keras (an open-source library that provides a Python interface for artificial neural networks) and PyTorch (an open-source ML library designed for NLP and Computer Vision) do not offer Lite versions for low-end devices like smartphones.

TF (Tensorflow Lite) also offers various pre-trained models with commonly used algorithms and datasets for out-of-the-box usage in projects and applications. Several algorithms like You Only Look Once (YOLO) algorithm [16], Single Shot Detector (SSD) [17] and Region-based Convolutional Neural Network (R-CNN) [18] among others can be used to implement realtime object recognition.

We chose the SSD algorithm for our project as it offers a fair trade-off between speed and accuracy over other algorithms which offered either of these parameters.

The following table shows the speed and accuracy comparisons.

Table 2. Speed and Accuracy comparison among object detection algorithms.

Method	mAP	FPS	Size	Boxes	Input
Fast R-CNN	73.2	7	1	6000	1000X 600
Fast YOLO	52.7	155	1	98	448X 448
YOLO-VGG16	66.4	21	1	98	448X 448
SSD300	74.3	46	1	8732	300X 300
SSD512	76.8	19	1	24564	512X 512
SSD300	74.3	59	8	8732	300X 300
SSD512	76.8	22	8	24564	512X 512

The SSD algorithm also performs better when it comes to detecting objects of different shapes and sizes. This is evident from the comparison graph which shows the difference.

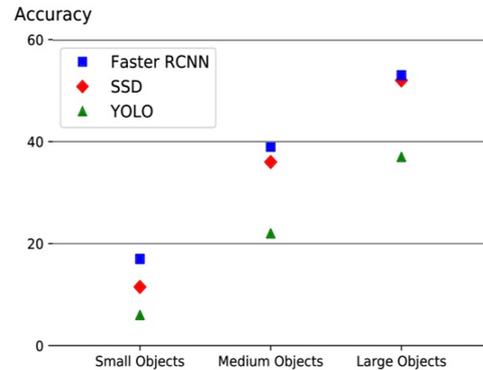


Fig 1 Algorithm's performance over objects of different sizes.

We have used the Tensorflow Object Detection API model which uses SSD mobilenet v1. This model is trained over the MS-COCO [19] dataset. The COCO dataset is a massive object detection dataset which has 330,000 images with over 200,000 labelled images consisting of 80 various object categories.

Real-time face detection can be implemented using algorithms like Multi-Task Cascaded Convolutional Neural Network (MT-CNN) [20], Google FaceNet [21] algorithm, and using the OpenCV Haar Cascade [22] and OpenCV Dlib [23] toolkits.

The FaceNet algorithm performed better among the others as it had a maximum accuracy of 99.63%. So, we chose to implement the FaceNet algorithm designed for low-power devices, the MobileFaceNet [24] model for face detection. The MobileFaceNet model offered better speed over others as is evident from the below graph.

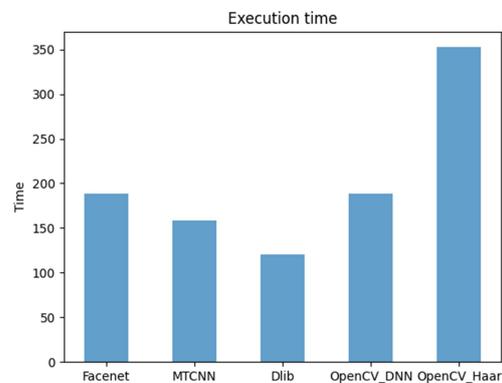


Fig 2 Comparison between face detection algorithms w.r.t time

The reason behind the fast performance is that the global average pooling layer has been replaced by a depth-wise convolutional layer which improves performance on face detection and recognition.

For the proposed system, we have used the MobileFaceNet model trained over the Labelled Faces in the Wild (LFW)^[25] dataset. The LFW dataset contains over 13,000 human faces captured in various angles and orientations.

Real-time text recognition has been implemented using Text recognition API from Google's ML Kit (Google's machine learning for Android devices in the form of a mobile SDK) which provides various libraries for implementing computer vision-related recognitions. Google's ML Kit website offers prebuilt APIs and packages which can be imported into our application to implement text recognition.

For implementing the chatbot, we have used AIML chatbot which uses Python packages like Pyttsx3 (An offline Python Text to Speech conversion library (TTS)), nltk (Natural Language Toolkit, a package of libraries and programs written in Python for processing natural language), chatterbot to provide feedback to the user as per the queries asked. Implementing the chatbot requires natural language processing and artificial intelligence for it to give replies and perform actions. The chatbot will read the command from the user, detect the keywords in the command, and then will perform the action as programmed by the developer.

The barcode scanner has been implemented using Google ML Kit's Barcode API. The API can directly be used in the application by importing the app dependencies and package.

6 Results

Object Recognition

The object recognition module has been implemented successfully

- Accuracy of 90 %
- Average run time of 1.3 seconds.



Fig 3 Detecting objects

Text Recognition

Text Recognition module has been implemented successfully

- Accuracy of 90 %
- Average run time of 1.4 seconds

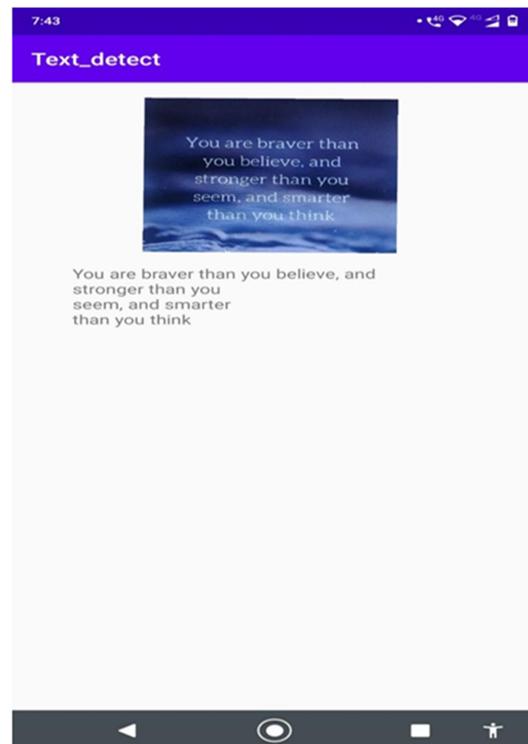


Fig 4 Text Detection

Face Recognition

Face Recognition module has been implemented successfully

- Accuracy of 85%
- Average run time of 1.2 seconds

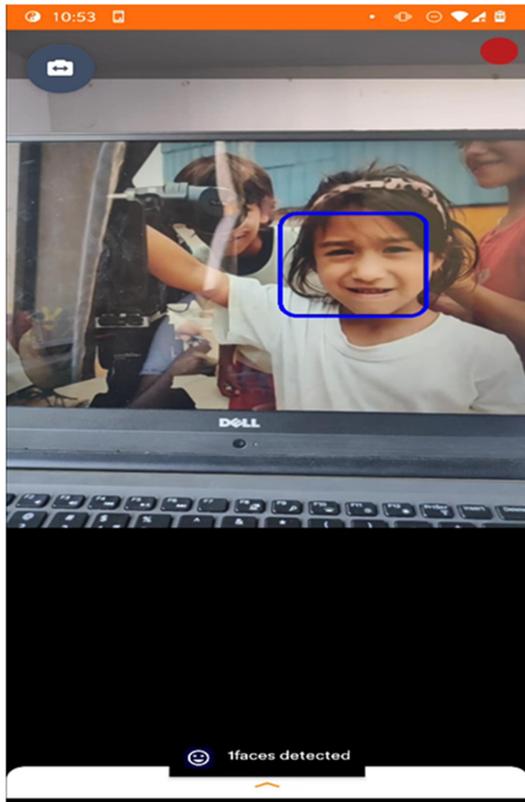


Fig 5 Detecting human face

Chatbot

Voice-based chatbot has been implemented successfully

- Offline feature's like calling, asking for date and time are working properly
- Online features like asking for weather, temperature, information on certain products are working properly if provided suitable internet connectivity

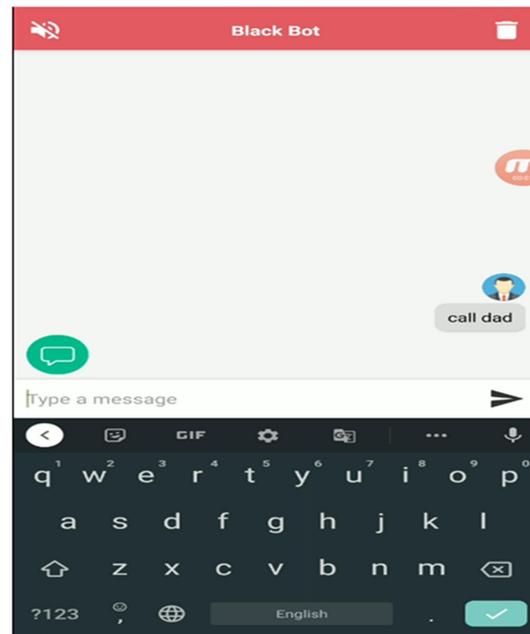


Fig 6 User command to chatbot via speech

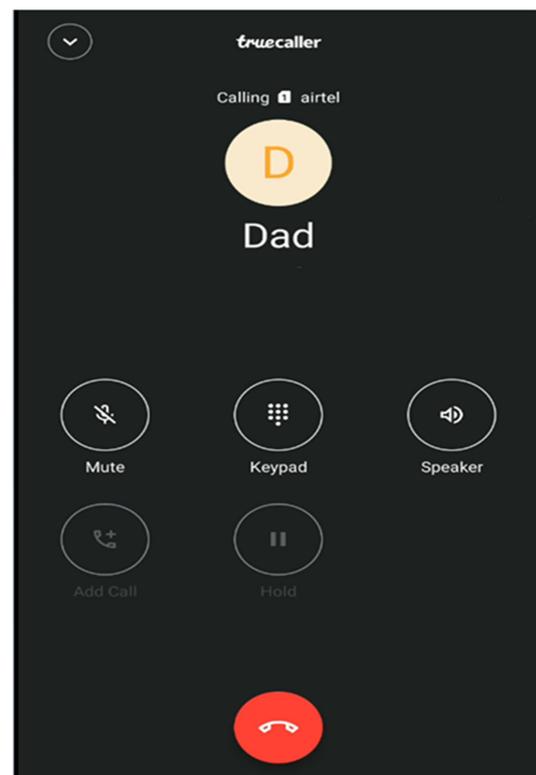


Fig 7 Chatbot performs the required action

Barcode Scanner

Barcode Scanner has been integrated successfully

- The information on product is provided correctly with a sufficient internet connectivity

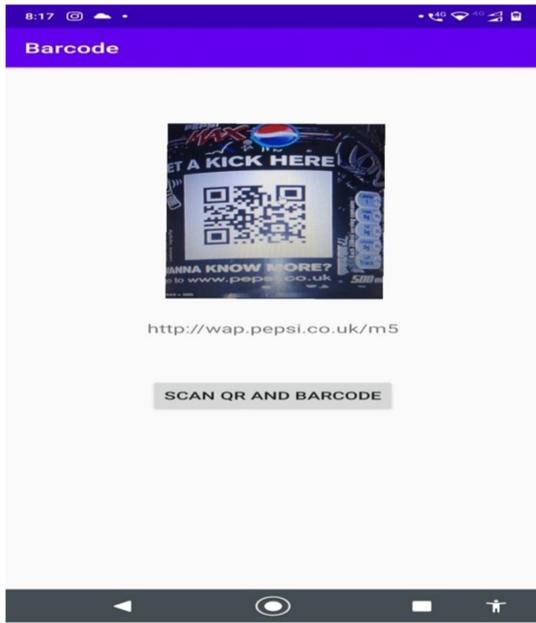


Fig 8 Barcode detection

Performance of our application compared to others

A. OBJECT RECOGNITION

1. Average time taken to perform one task(tested 20 times)

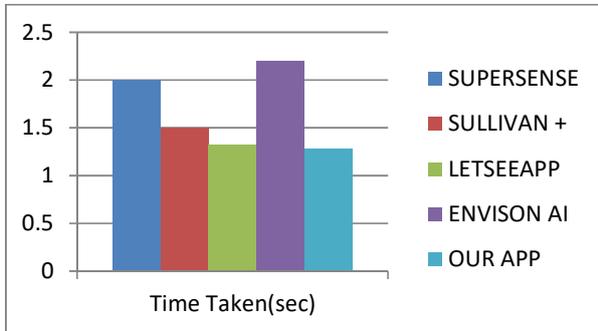


Fig 9 –Time taken by each application for object recognition

2. Accuracy- providing correct output (tested 20 times)

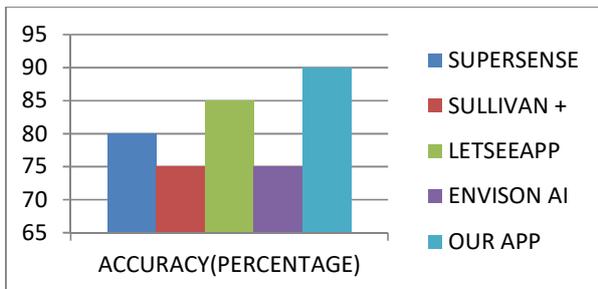


Fig 10 –Accuracy of each application for object recognition

B. FACE RECOGNITION

1. Average time taken to perform one task(tested 20 times)

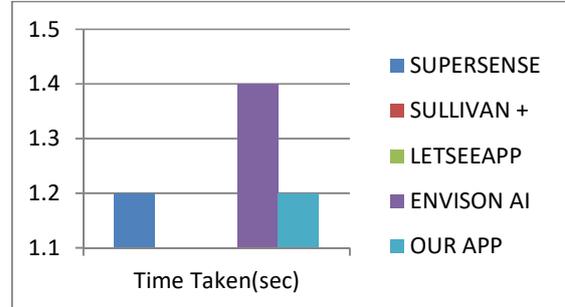


Fig 11 –Time taken by each application for face recognition

2. Accuracy- providing correct output (tested 20 times)

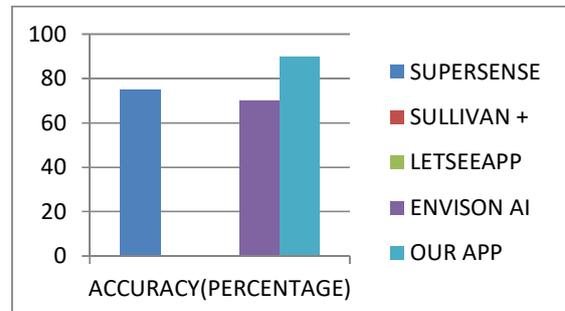


Fig 12 –Accuracy of each application for face recognition

C. TEXT RECOGNITION

1. Average time taken to perform one task(tested 20 times)

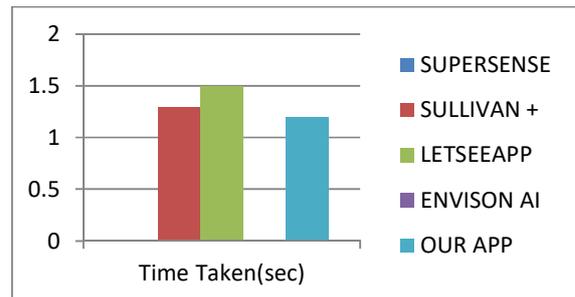


Fig 13 –Time taken by each application for text recognition

2. Accuracy- providing correct output (tested 20 times)

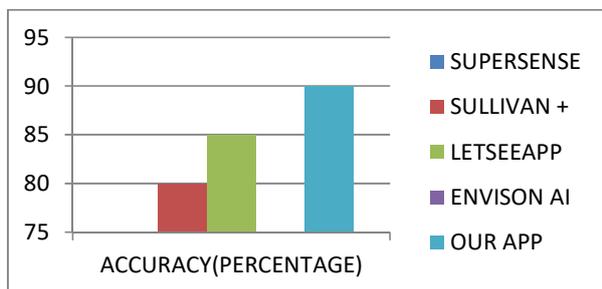


Fig 14- Accuracy of each application for text recognition

Note – the testing was done indoor under tube light (unnatural but ample brightness) and may vary according to the surrounding environmental condition

7 Discussion

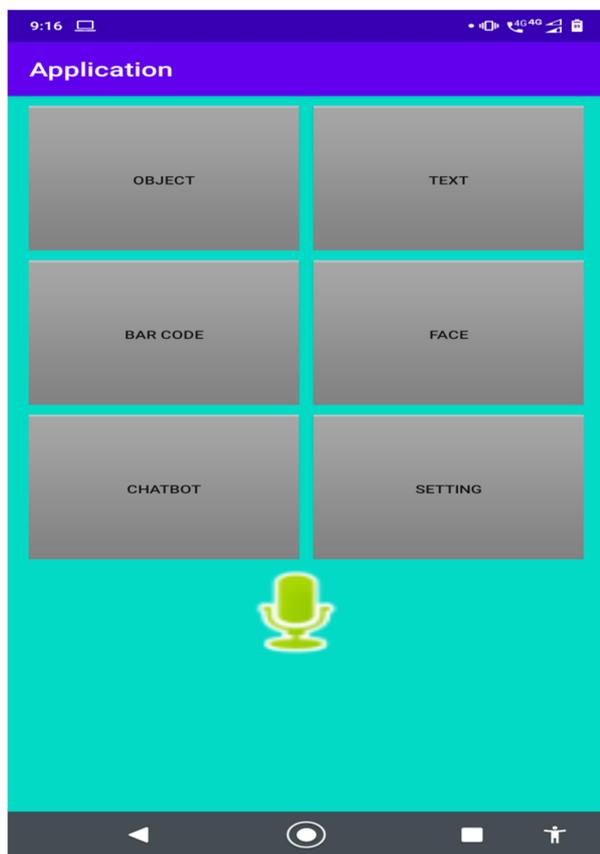


Fig 15 Layout of the application

The system (referred to as the android app hereafter) consists of 5 modules-- real-time face detection, real-time object and text recognition, barcode scanner, and chatbot. Each of these modules can be easily accessed from the android app with the click of a button. The UI of the application has been designed to be user-friendly for partially blind.

The working of the android app and its modules are explained below.

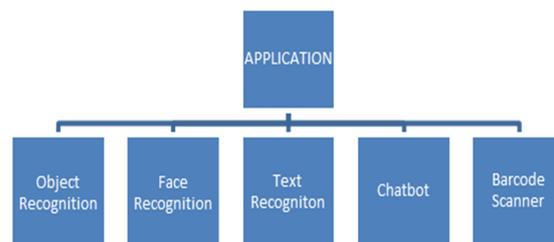


Fig 16 Application working flowchart

Modules used in ChatBot component of the Application.

The chatbot only requires the smartphone's microphone and Internet access. It offers some useful functionalities achieved through techniques and libraries mentioned below:

Pysttsx3 – It is a Python text-to-speech convertor that even works offline. We have implemented this module in our project to provide offline text-to-speech conversion.

This module provides us many features like

- TTS conversion without Internet
- Option to choose different voices
- Change speed or pitch of speech
- Easy-to-use and feature-rich API

Speech Recognition – A technique that is used to identify the queries of the user and convey it to the application. Which in turn will start the process it is requested to perform. This works in such a way that a keyword is associated with a particular action and when the keyword is spoken by the user the action will take place. Google Speech-to-Text has been used for speech recognition. We have implemented this for Natural Language Processing in our project

Natural Language Processing (NLP) – It is broadly defined as the automatic manipulation of natural language, like speech and text, by software. Natural language refers to the way humans normally communicate with each other.

This module is used in our project so that user can communicate with their device as they communicate with fellow human beings.

Datetime – This module is used to provide the date and time to the chatbot. This module works offline as its working on the data received from the device on which it runs. We have implemented this in our Project so that the user can ask the device for the time and date whenever needed the data from this module is given to another module (Pyttxs3)

Web Browser- The user can browse through the web using only voice commands given to the chatbot. We have configured this module in such a way that it can be used to gain information, play music (via an API to access YouTube), provide us with weather report(via an API to access The Weather Channel), and get news updates(via an API to access Times of India) and to get information on various topics we have also linked it with Wikipedia Module.

Working of Text Recognition:

Text Recognition API: Google ML Kit is a set of APIs and tools which can be used to deploy and automate certain applications like text recognition, barcode scanning, pose detection, etc. We have used the text recognition API in our project. The API will first use OCR to detect the text shown on the camera frame. It will then split the text into lines and the lines will be split into words.

These words will then be sent to the API for recognition and the recognized words will be spoken to the user using Google Text-to-Speech (TTS). Google TTS is available by default on all Android devices. Text recognition only requires the smartphone's camera access.

Working of Barcode Scanner:

Barcode API: Google ML Kit also offers a barcode API which can be used to scan barcodes and QR codes. The API will detect for any QR code/barcode displayed on the camera preview frame.

After detection, the QR Code/barcode will be read by the API to detect the embedded information or URL. The app will automatically open the URL link or will read out the information from the barcode using Google TTS. The barcode API only requires the smartphone's camera access.

Working of Face Recognition

For implementing face detection, we have used the MobileFaceNet model, which is an extremely efficient CNN model. The model is just 4.0MB in size and is designed for smartphones and embedded systems. The face detection process starts with detecting the human's faces in the real-time camera preview frame. The image is then warped using the detected landmarks like eyes, nose, jaws, eyebrows, etc and the face is captured. This image of the face is then processed and resized to be fed as input to the Deep Learning model.

The application is designed to capture preview frames at a resolution of 800*600px. The preview frame, if horizontal in orientation is rotated vertically and is cropped to 400*300px which removes the background and retains only the human body.

This image is then rescaled to 112*112px to be used as input for the MobileFaceNet model. On feeding the image, the model looks for the face in the image by matching the face features. It then creates a bounding box when it detects the face and is highlighted. The number of faces detected in the frame by the app is then outputted to the user orally using TTS functionality. This can be useful for the user to know the number of people in the room or a certain place. Face detection requires only the smartphone's camera access.

Working of Object Recognition

The android app has the object recognition feature where the user can point at an object and the app will recognize the object in frame and will output the object name to the user using Text-to-Speech. Object recognition has been implemented using SSD neural network. When the user points to an object, the frame is cropped to 600* 800 and is inputted into the model till the whole frame is covered. Based on the confidence level set by the user, the model creates multiple boxes with different aspect ratios throughout the image and tries to detect the object.

The accuracy of detection of the object depends on the confidence level. Once the object is detected, it then creates a box over the detected object with a label. The name of the object is then read to the user using TTS. The object recognition requires only the smartphone's camera access.

8 Conclusion

The proposed android application is designed to help and guide the partially blind in their daily tasks when needed. The application has 5 main components, namely- text recognition, object recognition, face detection, chatbot, and barcode scanner. The text and object recognition, barcode scanner, face detection, and chatbot are working as proposed and intended. Several changes in the text-to-speech module and the output are yet to be implemented which will be added in the coming months. This application is intended to work in indoor and outdoor conditions provided there is a good lighting condition.

Acknowledgment

This work is supported by the Department of Information Technology, SIES Graduate School of Technology. This Project is also supported by the Head of the Department Dr. K. Lakshmisudha and Project Guide Prof. Bushra Shaikh(co-author).

References

1. S. Tosun and E. Karaarslan, "Real-Time Object Detection Application for Visually Impaired People: Third Eye," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey
2. N. M. Tembhurne, S. V. Vaidya, A. Shiekh, S. Dravyakar, "Voice Assistant for Visually Impaired People", International Research Journal of Engineering and Technology (IRJET).
3. D. Dahiya, H. Gupta and M. K. Dutta, "A Deep Learning based Real-Time Assistive Framework for Visually Impaired," 2020 International Conference on Contemporary Computing and Applications (IC3A), Lucknow, India, 2020
4. F. Ahmed, M. S. Mahmud, and M. Yeasin, "RNN and CNN for Way-Finding and Obstacle Avoidance for Visually Impaired," 2019 2nd International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 2019
5. S. Gianani, A. Mehta, T. Motwani, and R. Shende, "JUVO - An Aid for the Visually Impaired", 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai
6. R. Kukade, R. Fengse, K. Rodge, S. Ransing, V. Lomte, "Virtual Personal Assistant for the Blind", 2018 International Journal of Computer Science and Technology (IJCTST), Bali, Indonesia.
7. M. A. Khan Shishir, S. Rashid Fahim, F. M. Habib, and T. Farah, "Eye Assistant: Using a mobile application to help the visually impaired," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh
8. A.Karthik, V.K.Raja and S.Prabakaran, "Voice Assistance for Visually Impaired People," 2018 International Conference on Communication, Computing and Internet of Things , Chennai, India
9. G. Singh, K. Takhtani, O. Kandale, N. Dadhwal," A Smart Personal AI Assistant for Visually Impaired People", Vol 7, Issue 6, pg.1450-54, International Research Journal of Engineering and Technology (IRJET)
10. V. Sharma, V. M. Singh, S. Thanneeru," Virtual Assistant for Visually Impaired", (April 19, 2020). Available at SSRN:<https://ssrn.com/abstract=3580035>
11. S. A. Jakhete, P. Bagmar, A. Dorle, A. Rajurkar and P. Pimplikar, "Object Recognition App for Visually Impaired," 2019 IEEE Pune Section International Conference (PuneCon), Pune, India, 2019
12. Supersense: https://play.google.com/store/apps/details?id=com.mediate.supersense&hl=en_IN&gl=US
13. Sullivan+: https://play.google.com/store/apps/details?id=tuat.k.r.sullivan&hl=en_IN&gl=US
14. EnvisionAI: https://play.google.com/store/apps/details?id=com.letsenvisionai&hl=en_IN&gl=US
15. LetSeeApp: https://play.google.com/store/apps/details?id=com.letseeapp&hl=en_IN&gl=US
16. "You Only Look Once: Unified, Real-Time Object Detection": [1506.02640] You Only Look Once: Unified, Real-Time Object Detection (arxiv.org)
17. SSD: Single Shot MultiBox Detector": [1512.02325] SSD: Single Shot MultiBox Detector (arxiv.org)
18. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks":[1506.01497] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (arxiv.org)
19. "COCO - Common Objects in Context":COCO - Common Objects in Context (cocodataset.org)
20. " Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks" [1604.02878] Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks (arxiv.org)
21. "FaceNet: A Unified Embedding for Face Recognition and Clustering": [1503.03832] FaceNet: A Unified Embedding for Face Recognition and Clustering (arxiv.org)
22. GitHub - opencv/opencv: Open Source Computer Vision Library
23. GitHub - ageitgey/face_recognition: The world's simplest facial recognition API for Python and the command line
24. " MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices": [1804.07573] MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices (arxiv.org)
25. LFW Face Dataset: <http://vis-www.cs.umass.edu/lfw>