

# Assistive Device for Visually Impaired People



N. Sriram, Anirudh Hosur, Akash Reshan, and P. Vetrivelan

**Abstract** In this project, we intend to create an assistive system for the blind, which will help them identify the person in front of them, read text written on signboards, etc., and assist them while crossing a road based on the traffic signal color. All the output will be in the form of audio instruction that will be played via a headphone that is connected to the system. The Raspberry Pi controller will be used to integrate the various components and control the processing of the images and also to train and store the model, using OpenCV and Tesseract; for the audio output, we will be using Google-text-to-Speech (gTTS). This system will assist the blind and partially sighted people in the known and unknown environment without much help from other people. This device can improve mobility as well as the safety for the visually impaired people.

**Keywords** gTTS · TTS · Recognition system · Assistive system · OpenCV.

## 1 Introduction

Moving through an unknown environment is a real challenge for the visually impaired or blind people, although they have to rely on their other senses. According to the World Health Organization, there are roughly about quarter billion visually impaired people around the world, out of which 39 million are blind [1]. The World Health Organization has estimated the number of blind people will double by 2022. They usually use a white-colored cane or Seeing Eye dog for their assistance.

However, these methods are not effective because they do not help in saving blind people from risks [2].

Assistance systems that were designed previously for visually impaired people consist of ultrasonic canes, voice-assisted navigation canes, laser-based walkers,

---

N. Sriram · A. Hosur · A. Reshan · P. Vetrivelan (✉)  
School of Electronics Engineering, Vellore Institute of Technology, Chennai, India  
e-mail: [vetrivelan.p@vit.ac.in](mailto:vetrivelan.p@vit.ac.in)

A. Hosur  
e-mail: [anirudh.hosur2017@vitstudent.ac.in](mailto:anirudh.hosur2017@vitstudent.ac.in)

infrared canes etc. [3]. There are some benefits of these systems, but they also have some disadvantages. For instance, infrared cane assistance infrared has a short-range for detecting obstacles, produces not so favorable results in the dark. When an obstacle is detected, the ultrasonic sensors which are present on top of the cane start vibrating signaling obstacles nearby or in that surrounding. As the cane can only scan the nearby surroundings compared to a person, it is not so effective in its function [4]. In case of the laser-based walking assistance, obstacles are detected by scanning the surrounding with lasers.

A laser is fixed and another is rotated by a DC motor. There is one belt of five motors that give feedback to users when an obstacle is detected [5]. In this assistance system, lasers are used which may harm other people if it strikes them in their eyes or skin.

An assistance system is suggested by the team in this research work which is aimed at assisting blind people, which will be entirely safe for people and the environment as it uses a camera and a headphone or a speaker to alert the user if there is an obstacle in front of him and to notify him to change his/her direction thus eliminating the use of the cane and laser-based projects as it can harm people vigorously and can cause untreatable damage, whereas our project is very cost-effective, efficient and safe to use and user friendly.

## 2 Design and Methodology of Blind Assistive Device

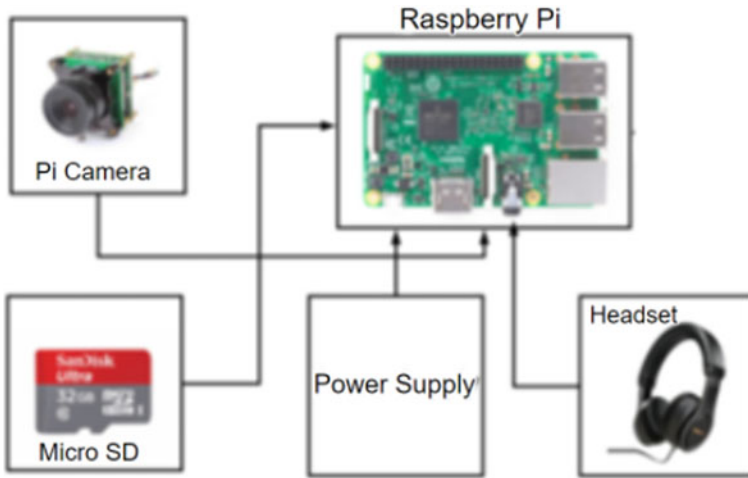
The idea implemented in this project is a voice assistive system for visually impaired people. It will have multiple functionalities that help the blind in different situations.

The text-to-speech (TTS) module is used to generate audio output so that the person can hear the instructions. To implement this, we use the Google text-to-speech (gTTS) module; it creates an mp3 file of the audio based on the string that is input, and this can be played using the default audio player on your device.

The face recognition module is used to identify the name of the person that is in front of you. For this, we will use Python OpenCV to process the image. We create a dataset by adding images of a person's face. The features are stored in the dataset along with the name of the person. Now, when a new image is an input, using Haar Cascade, we try to identify the name of the person by matching the features of the images. This name is then sent to the TTS module for audio.

The traffic signal detection system is used to help the blind while crossing roads, etc.; it uses Python and NumPy to convert and store the image into arrays. We initially set the RGB values for the colors in the traffic signal. So, based on the RGB values in the array, we can identify the signal color and hence generate an instruction string that is sent to the TTS module for audio output.

The sign-board text recognition module is used to read the signboard and provide audio output. For this, we use the PyTesseract module to extract the text present in the image and convert it into a string and send it to the TTS.



**Fig. 1** Block diagram of the assistive device for the visually impaired

## 2.1 Block Diagram

Figure 1 depicts the schematic block diagram of the proposed model of the assistive device for the visually impaired. Various components involved in this project diagram are explained in the following sections. The interface between user and system will be controlled by using buttons.

## 2.2 Raspberry Pi 3B

The third generation of the Raspberry Pi is called a Raspberry Pi 3B. It is a compact single-board computer that is widely used in various fields. It has a Broadcom system on a chip and processing unit which is central (CPU) and a graphics processing unit (GPU) which is on chip which are ARM-compatible. Its subsequent features:

- Broadcom BCM2387
- 802.11bgn Wireless LAN
- Bluetooth 4.1
- 1.2 GHz Quad-Core ARM
- 4xUSB ports
- 1 GB RAM, 64 Bit CPU [6].

The purpose of Raspberry Pi in this project is to interface all the components, process the image for the various modules, and to provide the audio output through the 3 mm audio port.

### **2.3 Raspberry Pi Camera**

The RPi camera module has a resolution of 5MP. It is a CMOS camera that has a fixed focus lens that helps in capturing still images and also for high-definition video. By default, the still images have a resolution of  $2592 \times 1944$ , whereas the video has a resolution of 1080p at 30 FPS or 720p at 60 FPS, and  $640 \times 480$  at 60 or 90 FPS. The Raspberry Pi's preferred operating system is the Raspbian OS, and it helps us in interfacing the camera module. [7]. It is used to capture the real-time images and provide the input to the different modules.

### **2.4 Headphones**

Headphones are a form of miniature loudspeaker-like drivers that are worn over or inside a user's ears/head, and thus enables them to listen to audio in a more private environment. They are a type of electro-acoustic transducers that mainly convert electrical signals to their respective sound waves. It is used to provide an audio result of the various modules; we can use any mobile headphones. A better headphone produces a better sound quality.

### **2.5 Power Supply**

An external power supply such as a power bank is required to power the Raspberry Pi and hence power the entire system.

## **3 Algorithm**

This section discusses the platform and major modules used to implement the programming of the algorithm used to test the images for the various functionalities of the project.

### **3.1 OpenCV**

OpenCV is a collection of functions that enables users to apply real-time computer vision in their code [8]. Since our project involves vision, this module is of utmost importance. It has ways of feature extraction, model training, etc.

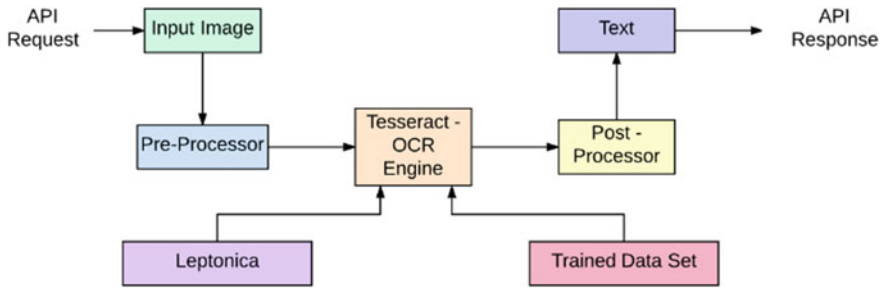


Fig. 2 Text recognition using Tesseract

Digital image processing is the use of computer algorithms to process, communicate and display digital images. Digital image processing algorithms are often used to: improve resolution and eliminate noise and other artifacts.

### 3.2 Tesseract

Tesseract engine in Python contains a vast set of functions for optical character recognition. Optical character recognition refers to the process of detecting text in images. Tesseract is provided to us using apache 2. This can be used directly or through a set of functions/APIs so that it can be used in languages like Python [9]. However, it is a line-based command and does not have a GUI.

Using wrappers (classes that allow you to use certain functions), we can use Tesseract in a huge set of languages. This has two ways of implementation. It can be used directly to detect text in a document, or it can be combined with an image detector to recognize text from an image. This is the application that we use in our project.

Tesseract uses a convolution neural network that recognizes the image with the character. Inside this, there is another neural network which is set to be a text line detector. However, the problem of detecting the length of the text is solved using LSTM which is a type of recurrent neural network (Fig. 2) [10].

### 3.3 Image Pre-Processing

Preprocessing is used to eliminate unwanted distortions and the correct image is used Next is that the process. Preprocessing steps within the image include increasing the contrast, resizing the RGB to form changes (Fig. 3).



**Fig. 3** Preprocessing of the test image

### 3.4 Segmentation

This segment is usually supported by image pixel structures. These edges can define locations. Other methods separate the image from areas based on color.

Color image segmentation supported the color characteristic of the image pixels assumes that similar colors within the image imply different objects. Hence, object classifications happen inside an image. This basically means that each group has a category of pixels that have similar color characteristics.

In this project, we use color image separation to extract the properties of the test image. Partitioning stages include morphological operations on RGB to HSV (color, saturation and value), binary conversion, binary black and white image and clear boundaries of the infected area.

### 3.5 Feature Extraction

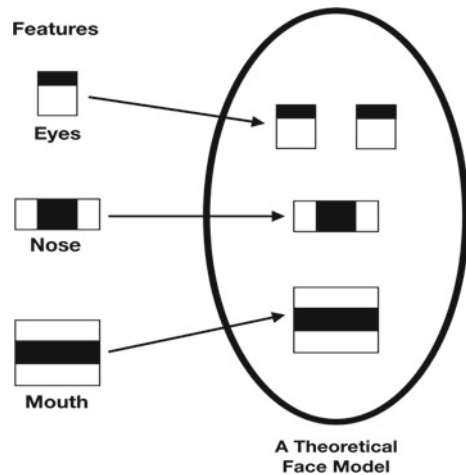
Feature extraction describes the process of obtaining information about the shape that we are interested in. This process helps it easier for the recognition function to recognize the pattern [11]. In image processing, pattern recognition primarily occurs in the form of dimension reduction. Many algorithms make this possible.

If the data fed to the algorithm is too big, then we convert it to a reduced set called the feature vector. The features that make it here go through a feature selection process, and the features obtained after this selection process are the ones that have the useful information from the input data. We can use this set in the algorithm as it will be faster and easier as it is a smaller set to process.

In our project, we use Haar Cascades. Haar Cascades is a machine learning technique in which the model is trained using several positive images that help in improving the model and negative images which help in removing errors [12]. This process is used to find out facial features.

Figure 4 describes the various features of the face that are extracted and compared in the face recognition module. The most prominent features of the face are extracted; the eyes, nose and mouth determine a theoretical model of the face. This information is then carried out towards the output modules.

**Fig. 4** Extracted features of the face



### 3.6 NumPy

NumPy is the primary extension in Python that is used for advanced mathematical operations as well as simple processing of multidimensional arrays [13]. Since most images are multi-dimensional arrays, this is of prime importance.

In our project, by storing the images read as a NumPy array, various image processing can be performed using NumPy functions. NumPy allows acquisition and rewriting of pixel values, trimming by the slice, concatenating, etc.

In our traffic light detection module, we store the pixel values of the image as a NumPy array, and using this, we compare the values with the standard values set to identify the region with the required colors.

### 3.7 Bilateral Filter

A bilateral filter is a very common filter used in image processing. The main reason for this is that it keeps the edges sharp while removing noise effectively [14]. Another option we had was the Gaussian filter. The advantage of the Gaussian filter is that it is faster. However, the Gaussian filter algorithm of finding the weighted averages of nearby pixels is a flaw for our project as our focus is the features that may get ignored while using the Gaussian filter. The Gaussian filter also blurs the edges which can cause problems for the model to recognize [15].

This is used in our signboard text detection, the edges of the letters can be clearly identified, while the other regions of the images are blurred (Figs. 5 and 6).

As you can see in the above images, the edges are preserved, and this helps to identify the image text in our project.



Fig. 5 The application of a bilateral filter is seen in the above pair of images

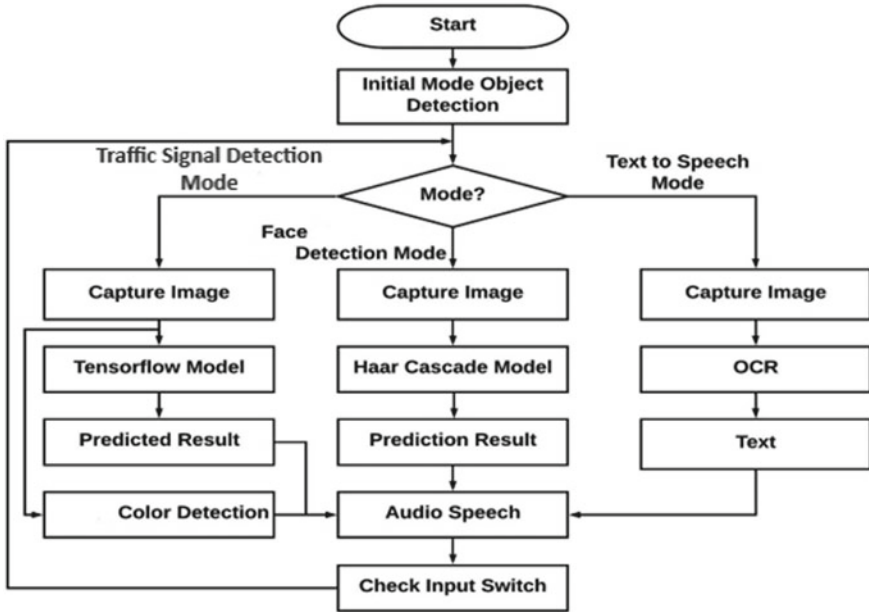


Fig. 6 Flow chart of our system

### 4 Results and Discussion

The traffic light detection module identifies the color of the signal, and if it is red, we instruct the user to cross as all the vehicles would have come to a halt; similarly, if it is yellow or green, we ask them to wait through the audio output.

Here are a few sample images where the colors are identified (Figs. 7 and 8).

The face recognition system stores the sample images of the face and then compares the input image with the available dataset to classify the person. Some of the output images of the module are provided below in Fig. 9. A classification





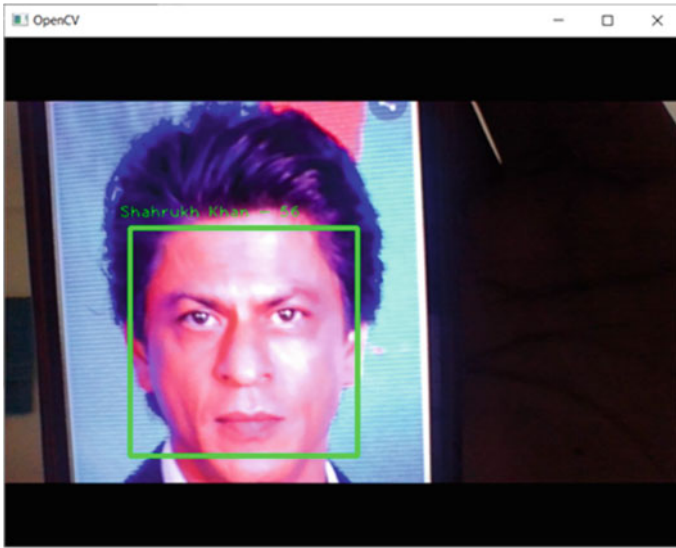
Fig. 7 Snapshot of the detection of red signal color



Fig. 8 Snapshot of the detection of green signal color



Fig. 9 Snapshot of the sample dataset



**Fig. 10** Snapshot of the command window during runtime

algorithm then classifies the image based on the training data (Fig. 10).

The signboard text detection module is capable of reading short phrases and very small sentences, like those on the signboards. The text-to-speech is capable of converting any English string to an mp3 file that can be played out loud using any of the default audio player present in the device. The interface between the user and the system is done using buttons to choose which module is to be used.

## 5 Conclusion and Future Work

This paper combines the use of Python and OpenCV for image processing and recognition. This paper proposes an efficient and cost-effective system for the blind. System design involves using the Raspberry Pi and the Pi camera module as a hardware device. The Pi camera captures the images of the face/traffic light/signboard, using image processing. Using OpenCV and Python, we can recognize faces and detect traffic lights and signboards. The output audio is played through the headphone and provides necessary instruction/information to the user.

The system allows the blind to act independently to a certain extent. The user can choose the mode of operation using buttons, depending upon the assistance that is required. He/she has three modes to choose from, which are the 3 modules in our project.

The future work on this system can be as follows:

- We can improve the signboard recognition module to help read books, etc., and hence serve as a book reader.
- We can enhance the model by retraining the model on a cloud GPU and use it to recognize a wide number of faces, signs, etc.
- We could add a GPS module to our device which will send the information about the current location of the user to their care-taker, etc.
- The information could be received on the care-takers mobile as a text message or also we could develop an app using Android Studio, which could show us the real-time location of the blind person (like Google maps).

## References

1. Pascolini D, Mariotti SP (2011) Global estimates of visual impairment: 2010. *British J Ophthalmology* 96(5):614–618
2. Dakopoulos D, Bourbakis N (2010) Wearable obstacle avoidance electronic travel aids for blind: A survey. *IEEE Trans Syst, Man, Cybern Part C (Appl Rev)* 40(1):25–35
3. Varpe KM, Wankhade MP (2013) Visually impaired assistive system. *Int J Comput Appl* 77(16):5–10
4. Elmannai W, Elleithy K (2017) Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions. *Sensors* 17(3):565
5. Agarwal AWP, Zink M, Adame MR, Möller K, Burgard W (2016) Navigating blind people with walking impairments using a smart walker. *Auton Robot* 41(3):555–573
6. Gay WW (2014) The Raspberry Pi. In: *Mastering the Raspberry Pi*. Apress, pp 5–8.
7. Gay WW (2018) Pi camera. In: *Advanced Raspberry Pi*. Apress, pp 493–499.
8. Pulli K, Baksheev A, Korniyakov K, Eruhimov V (2012) Real-time computer vision with {OpenC}. *Queue* 10(4):40–56
9. Geetha C (2019) Optical character recognition with tesseract. *J Mech Contin Math Sci* 1(2)
10. Google Open Source Google, [Online]. Available: <https://opensource.google/projects/tesseract>
11. Setiono R, Liu H (1998) Feature extraction via Neural networks. In: *Feature extraction, construction, and selection*. Springer, pp 191–204
12. Automatic gate using Face recognition technique using HAAR cascade algorithm. *Int J Eng Adv Technol* 9(3):1302–1305.
13. Harris CR, Millman KJ (2020) Array programming with NumPy. *Nature* 585:357–362
14. Tomasi C, Manduchi R, Bilateral filtering for gray and color images. In: *Sixth International Conference on Computer Vision*
15. Sagar V (2015) Gaussian blur filter effect for CAPTCHA security. *Int J Comput Appl* 124(3):29–31



**N. Sriram** is a final year student at Vellore Institute of Technology pursuing his B.Tech in Electronics and Computer Engineering. His research work is concentrated in the fields of computer vision and machine learning. He has a strong sense of problem-solving and strives to solve real world problems using state-of-the-art technology.



**Anirudh Hosur** is a final year student at Vellore Institute of Technology pursuing his B.Tech in Electronics and Computers. He's research work is concentrated in the fields of deep learning and computer vision. He has a strong sense of problem-solving and a data-driven enthusiast.



**Akash Reshan** a final year student at Vellore institute of technology pursuing his B.Tech in Electronics and Communication. His research work is concentrated in the fields of deep learning and computer vision. He has a strong sense of problem-solving and wants to improve the current systems using newer technology.



**Vetrivelan Pandu** is an Associate Professor Senior and Head of the Department (HoD) for Bachelor of Technology (Electronics and Communication Engineering) in School of Electronics Engineering at Vellore Institute of Technology (VIT), Chennai, India. He has completed Bachelor of Engineering from the Madras University, Chennai and both Master of Engineering in Embedded Systems Technologies and Doctor of Philosophy in Information and Communication Engineering from Anna University, Chennai. He has around 17 years of teaching experience altogether in CSE and ECE Departments in both private Engineering Colleges in Chennai (affiliated to Anna University, Chennai) and Private Engineering University in Chennai respectively. He is the author of five book chapters and one proceeding in lecture notes published by reputed springer publisher, and has authored 30+ Scopus indexed Journal papers and few other papers published in reputed international conferences. He was member in Board of Studies, doctoral committee, doctoral thesis Examiner, doctoral oral Examiner in both private and government Universities. He worked as a reviewer for reputed International Journals and International Conferences. His research interests include Wireless Networks, Adhoc, VANETs, Embedded Systems and Internet of Things (IoT) with Machine Learning.