Chakshu-Eye for The Blind

Abhishek Bhardwaj (B17072)*, Aashima (B17031)†, Akash Deep Batham (B17074)‡, Anurag Garg (B17077)§, Aryan Singh (B17010)¶, Arun Kumar Bairwa (B17078)‖, Deshraj Meena (B17080)**
{*b17072, †b17031, ‡b17074, §b17077, ¶b17010, ‖b17078, **b17080}@students.iitmandi.ac.in

***Abstract*—Chakshu-Eye for the Blind empowers the blind to recognise faces and objects by voice control. The Blind recognise the people around them by their voice or feeling their touch. But, with changing voices and so many names to memorise with their characteristic voices, it is really difficult and confusing. With Chakshu, the user can ask for identification of the known entities or store names of the new by voice input. Also, it can tell the user the distance of closest entity i.e. it gives the user basic idea of the surroundings. Thus, it aims at making the blind independent and confident using technology.**

I. Apparatus Required

Serial no.	Materials Used	Quantity
1	Raspberry Pi 3 Model B	1
2	Raspberry camera module V2	1
3	Ultrasonic Sensor HC-SR04	2
4	Resistor 4.7 kΩ	2
5	Resistor 10 kΩ	2
6	Sound Card	1
7	Mic	1
8	Earphones	1
9	Jumper Wires	16
10	Connecting Wires	8
11	Hat	1

II. Motivation

For people with low or no vision, it becomes difficult to recognise people during several social interactions. They tend to remember voices, fragrances or specific touch of people and the sound, smell and sensation of objects. The inability to identify people in large groups is a drawback for the blind in professional and social gatherings. Sole reliance on hearing and retaining ability makes them dependent on others but, always they might not find someone to help. Chakshu is a solution to these problems.

III. Introduction

Chakshu is a proposed prototype which extracts images using Raspberry pi. Images taken using camera module are processed using OpenCV. It does the following:

1) Face Recognition
2) Object Recognition
3) Speech Recognition
4) Speech Output
5) Distance Measurement

The hat/cap consists of a raspberry camera module, the mic and the earphones, all connected to the Raspberry pi where the camera and mic provide input and the earphones provide output. The hat is command driven. It takes voice input from the user which is then converted into python string to be taken as command by the program. The program functions in a loop which can be ended by the user. The program takes command, acts accordingly, finishes its task and then again the loop continues asking for input. It has five commands:

1) **Face:** The face command is for identifying faces. It activates the camera to capture image which is then processed using OpenCV (Open Source Computer Vision). The program finds a face in the image and compares it with the stored faces. If it finds a match, it returns the name of the first matched face, else it asks for name and stores the face in its database with the respective input.
2) **Object:** The object command is for identifying objects. It activates the camera to capture image which is then processed using DNN (Deep Neural Networks). If its a human, it returns person, if a match is find with any of the objects present in the library, it returns the name else it returns nothing and the loop continues.
3) **Sleep:**"Sleep x" command delays the program i.e. after finishing its task, the next iteration of loop occurs after x seconds.
4) **Find:**The find command is used to calculate the distance of nearest entity. It activates the ultrasonic sensor which calculates distance x of closest entity. It returns its output in the form, "There is something at x cm (centimetres)".
5) **Cancel:** The program ends.

IV. How Things Work

A. *Hardware/Devices Implementation*

1) Pi 3 Model B: Raspberry Pi 3 Model B is the third generation Raspberry Pi. It is a powerful small programmable single-board computer. Raspberry Pi works in Open Source platform. It uses Broadcom BCM2387 chipset with a 1.2GHz 64-bit Quad Core ARM cortex A53 processor. Equipped with Camera Interface and Display Interface, it has 1GB RAM and micro SD port for storing data and loading bigger programs. Pi 3 contains 4 USB ports and 40pin extended GPIO (General Purpose Input Output) enhancing real world project. With inbuilt Wi-Fi and Bluetooth, it supports wireless internet and we have used Ubuntu Mate as the Operating system [1].

2) Raspberry Camera Module V2: It features a Sony IMX219 8-megapixel image sensor. Featuring a fixed focal

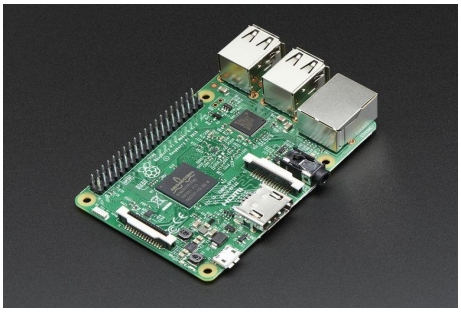

Figure 1. Raspberry Pi 3 Model B

lens, its capable of taking high-definition videos and static photographs. It can be attached to the Pi through a small socket, dedicated with CSI Interface specially for cameras [2].

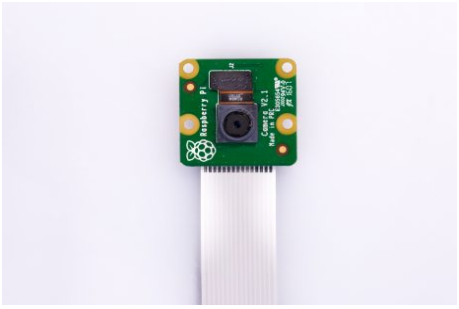

Figure 2. Raspberry Camera Module V2

3) USB Sound Card: A USB sound card is a simple input/output device. It takes power from the USB port and has a microphone in and audio out (3.5mm) jack.

4) Ultrasonic Sensor HC-SR04: The sensor transmits a frequency of 40Hz and can measure distances accurately over a theoretical range of 2cm to 400cm. It works on an input of 5V.To initiate the sensor, a pulse of 5V is needed for at least $10\mu s$ [12].

B. *Facial Recognition*

Face Recogniiton has two stages:

1) Face Detection: The image is searched for a face.
2) Face Recognition: The detected face is compared to the ones stored in database.

A facial recognition system uniquely identifies a person from an image or frame from a live video. In general, it maps a persons face using selected facial features and compares the given image from the images stored in its database by analysing patterns [3]. We have used *OpenCV* and *Face Recognition library* provided by Python for the same. OpenCV is a cross platform library that enables real-time image processing. Face Recognition library makes use of machine learning algorithms for face detection. It is built using dlibs state-of-art face recognition with deep learning. The model can have an accuracy up to 99.38% on the labelled faces. Faces are complicated, thus one simple test isnt enough for proper identification. There are thousands of small features, more particularly patterns to be recorded and matched.

The face recognition algorithms work by breaking the complete task into several thousand smaller, bite-sized tasks, which are easier to solve. These smaller tasks are termed *classifiers*. A face might have more than 6,000 classifiers, each one having a face to detect (within error limits). The algorithm starts from the top left of an image and moves down across small blocks of data, analysing each block. With more than 6,000 tests per block, there are millions of calculations to be done.

To make it easier, face recognition and dlib use *cascades*. Cascade divides the process of face recognition (detection and matching) into several stages. Per block, a rough and quick test is performed. If it passes, only then a slightly more detailed test is performed and so on. Majority of the images return negative in the earlier tests, thus saving time. Thus, face recognition is performed in real time [4].

C. *Object Recognition*

An object recognition algorithm takes an image as input and gives a class label as output. The algorithm has to be trained with thousands of images to differentiate between different classes/objects. It can only recognise objects it has learnt.

Images are in the form of 2-D Matrices. They are perceived either in the form of raster (sequence of pixels) or vector images (set of colour-annotated polygons). The geometrical encoding of the image is converted into physical features and objects, called *constructs*. These constructs are then logically analysed for classification and feature extraction.

The first step is to simplify the image by extracting useful information and discarding the rest. Haar-like features, HOG (Histogram of Oriented Gradients), SIFT (Scale-Invariant Feature Transform), and SURF (Speeded Up Robust Feature) are few well-known features to be used. Algorithm treats feature vectors to be points in dimensional space and find planes/surfaces such that all objects from same class belong to the same side of plane.

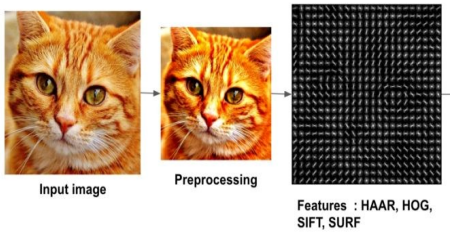

Figure 3. Image Analysis by program

But, we are using more advanced *DNN* which does the work really efficiently. Neural networks are inspired by the human brain. Like nerve cells, they are made up of layers that are interconnected, more the layers, deeper is the network. The connection between any two units has weight assigned to it.

For an object "abc", they are trained with thousands of images of "abc" and "not abc". Neural Networks self-learn i.e. if they recognise object incorrectly they note the errors and adjust their weightings [5],[6].

D. *Speech Recognition*

It is the process of converting speech to text. We have used *Google Speech API (Application Programming Interface)* and *Speech Recognition Library* provided by Python. It requires an active internet connection. The program records audio from mic, sends it to the speech API for processing and returns a string in Python3.

Audio recording is done with the help of speech recognition module. Speech is a physical sound and is converted into electrical signal with the mic, and then to digital data using an *analog-to digital converter*. *HMM (Hidden Markov Model)* is used for further process. The input signal is viewed for a frequent short times intervals, known as fragments and is approximated to be a stationary process (no change in statistical properties). The digitised sound is filtered to cancel out unwanted noise and may be separated into different frequency bands. Humans may not speak at constant speed, and volume, so the system adjusts and aligns the sound to match with the already stored sample sound templates. The output of HMM is a sequence of vectors. These vectors are then matched with the known *phonemes*-a representations of different types of sounds we make. A special algorithm is used to compare this output to all known words, phrases and sentences. The resultant is a python string which may be used as a computer command [7]-[9].

E. *Speech Output*

To convert text into speech, *gTTS (Google Text to Speech) API* is used. It converts the given text into audio, which can be stored as a mp3 file. It allows unlimited length of text as input having note of proper intonation, decimals, abbreviations and much more. It supports many languages including English, Hindi, Tamil, French and many more [10].

F. *Distance Measurement*

Ultrasonic Sensor is a 4 pin module that measures distance using ultrasonic waves. The pins are as follows:

1) Ground: The pin is connected to the common ground of the system.
2) Vcc: The Vcc pin is to power the sensor, with 5V.
3) Echo: Echo pin is an output pin. It remains at high for a time period equal to total time taken by ultrasonic wave to go and return back to the sensor.
4) Trigger: Trigger pin is an input pin which initialises the sensor. Therefore, it has to be kept at high (5V) for at least $10\mu s$.

The sensor uses *SONAR (Sound Navigation and Ranging)*. The module has two eyes like projections in the front called the *transmitter* and the *receiver* [12]. The transmitter sends an ultrasonic wave, which is reflected by the object and received by the receiver. Distance (in m) is calculated as follows:

$$\begin{aligned} Distance &= Speed \times Time \\ &= (3 \times 10^8) \times 1/2 \times totaltime \end{aligned} \tag{1}$$

V. Results and Discussions

Our proposed prototype looks as in Figure 4. It has been tested and found to correctly recognise faces, objects like: tree, car, bicycle, bottle, potted plant etc. and measure accurate distances up to 200 cm. The user is independent and doesn't need a helper, thus, it succeeds in what we aimed to make.

The prototype has been made in a budget of Rs.4600. If produced on large scale, it will cost lesser and more affordable.

Figure 4. User wearing Chakshu

VI. Further Improvements

The project can be improved by adding/replacing various components:

- Replacing Raspberry Pi by Nano micro controller, with greater RAM, cache and better performance processor. Powerful processor will run program efficiently and faster. Also, face and object recognition could be processed simultaneously.
- Using Nano camera module with high resolution, auto focus and high frame rate will give more precise and faster image processing.
- Addition of GPS will make it easy for the user to reach his destination without seeking much of others help.
- Adding GSM to implement a panic button for safety. The panic button will send an alert and location to important contacts.
- Recognising objects like currencies, visa cards, tickets, sign boards etc. along with text recognition will be very helpful to the blind.
- It can be integrated with Google Assistant, making daily life tasks like calling, playing songs etc. easier.
- With compact devices, everything can be incorporated in a spectacle, being more convenient than a hat. Bluetooth

can be used to remove wired connections (if any), making the complete set portable an easy to carry.

VII. Conclusion

The device consists of a hat with a camera installed over it. It is voice controlled and wired connection with the camera provides faster access. The Blind need not to guess people and objects. He can identify them as the camera captures entities to process, upon his command. Audio feedback is spontaneous. Facial Recognition is non-contacting i.e. does not require the user to go or interact with the person beyond a distance. These features make it simple, reliable and user friendly.

VIII. Acknowledgement

We would like to thank the following people for their expertise and suggestions in this project.

- Dr. Hitesh Shrimali, Assistant Professor, IIT Mandi
- Suryavanshi Virendrasingh, B.Tech 2^{nd} year

References

[1] Figure 1. Available: https://www.zdnet.com/article/what-is-the-raspberry-pi-3-everything-you-need-to-know-about-the-tiny-low-cost-computer/
[2] Figure 2. Available: https://www.raspberrypi.org/products/camera-module-v2/
[3] Figure 3. Available: https://www.learnopencv.com/image-recognition-and-object-detection-part1/
[4] https://en.wikipedia.org/wiki/Facial_recognition_system
[5] https://realpython.com/face-recognition-with-python/
[6] https://cosmosmagazine.com/technology/what-is-deep-learning-and-how-does-it-work
[7] http://www.explainthatstuff.com/introduction-to-neural-networks.html
[8] https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/
[9] https://pythonspot.com/speech-recognition-usinggoogle-speech-api/
[10] https://realpython.com/python-speech-recognition/
[11] https://pypi.org/project/gTTS/
[12] https://www.acmesystems.it/HC-SR04