

A Helping Hand for The Visually Challenged, Mute and Deaf people (Team – 6)

Assessment - 3

Submitted by

RAMKRITHIK S (18BCE0528)

S A HARIPRASAD (18BCE0686)

N KISHORE KUMAR (18BCE2106)

KUNAL AGARWAL (18BCE2108)

A ANIRUDDHA (18BCI0029)

PARDHASARADHI TIRUMALASETTI (18BCE2068)

Prepared For

TECHNICAL ANSWERS FOR REAL WORLD PROBLEMS (CSE3999)

TC1 Slot

Submitted To

Dr. ARCHANA T

School of Computer Science and Engineering (SCOPE)



PROPOSED METHODOLOGY:

1. Sign Language Converter:

The language used for communication with deaf people is sign language. This is like their native language; in other words, it is like a mother tongue. As deaf people can't hear our voices in order to communicate with them, we use signs or symbols. But everyone may not know sign language as it is not the necessary one to be learned. so, to overcome this problem we proposed an application that takes live or recorded voice as input and communicates with the deaf through signs or symbols. In order, to make this application more interactive we have trained the data sets with images and also small GIFs.

This was implemented using Python. For our implementation we used libraries like speech recognition, NumPy, matplotlib.pyplot, matplotlib.pyplot.cv2 etc. Now, when the application is launched, we get there two option to begin our application like live recording and all done. All done is to close the application and the live recording is to get the input voice from the user through the microphone which is done using recognizer function which is python inbuilt library.

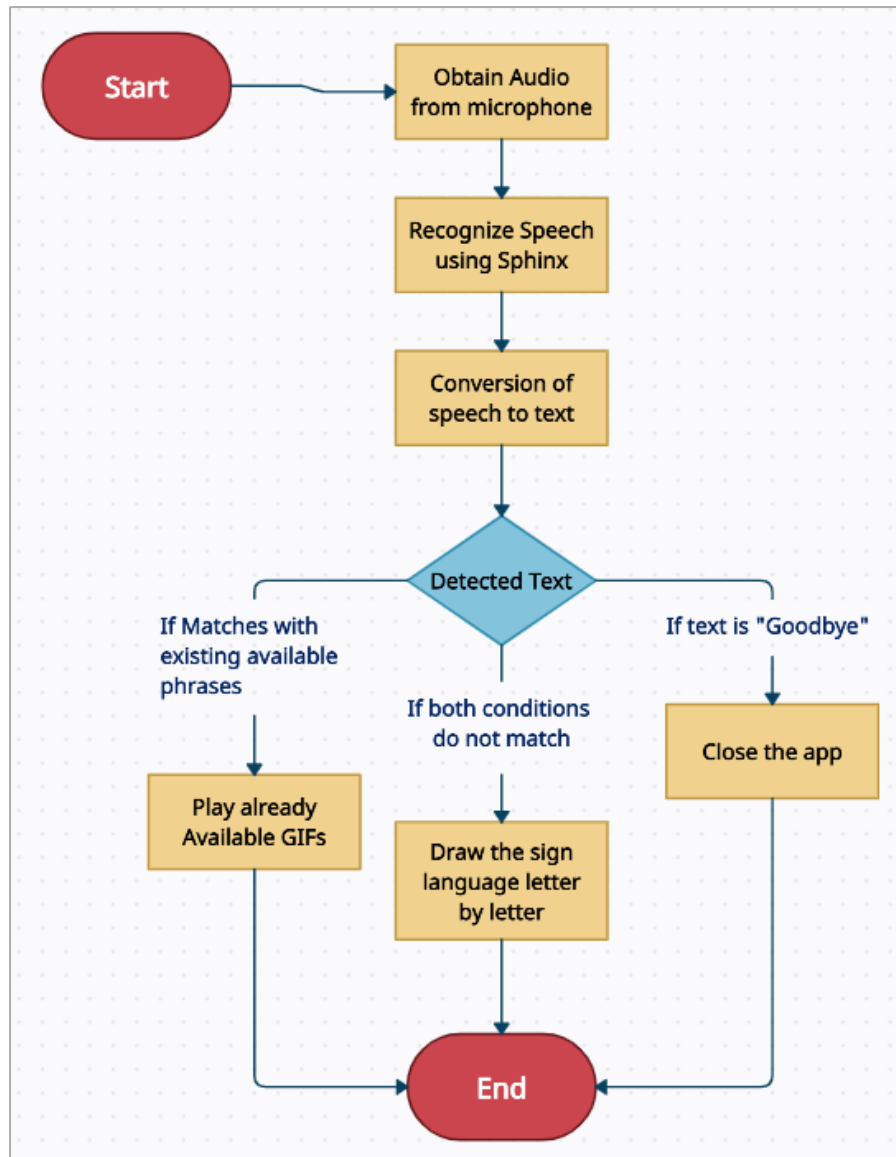
So store the recorded voice in temporary memory and we do text pre-processing using natural language processing techniques and if the voice is not able to be recorded we display a message called "could not listen" generally this is a user choice of message and for the text detected we perform dictionary based Machine translation that is to search the words in our trained dictionary if they were not found just spell the word using sign language which we have already trained from the alphabets A-Z.

Here to make our job easier and make the application more interactive we have added GIFs and some small video clips which are generally based on daily routines like "Hello", " Good Morning" ,"what is your name" etc. so that for these common questions and compliments our job gets done easier and here this is the main purpose of this application ,but for every application there must be a termination phase also so to achieve this we made a terminating command called "goodbye" so when the application receives this command during inputting audio phase it just get terminated.

Algorithm:

1. Get the input voice from the user through the microphone.
 - 1.1. Listen for a particular amount of time.
 - 1.2. Listened voice gets converted into text through speech recognition libraries.
2. After listening the voice is captured and saved in the temporary memory.
3. Through the temporary memory the voice is fetched and converted to text.
 - 3.1. To proceed with further manipulation, convert the entire string of text to lowercase letters
4. So, after getting the text each and every character of the inputted text is searched throughout the data set.
 - 4.1. If the text is "goodbye" then the application exits, as this is the existing command.
 - 4.2. And if the detected text is not "goodbye" then it first searches the word in the predefined dictionary images and GIFs.
 - 4.3. If it is not found there, then spells the word using symbols with some delay in the image display actions.
 - 4.4. Loops from step 2 till the speech ends
5. If an error occurs in Step 1, display "Could not listen."

Architecture Diagram:



2. **Braille Converter:**

Braille is a written language for blind people that is made up of patterns of raised dots that can be felt with the fingertips. Reading a book, typing, and determining what is written on a computer are next to impossible for visually impaired people because they cannot determine what is in front of them. The closest way for them to determine what is in front of them is to use their sense of touch. So, to overcome this problem, we designed a system that'll help these people by giving them a chance to connect with others. Our system will be responsible for many features like changing Text into braille format, Braille into text format and providing them with a simple keyboard made not of alphabets but with braille alphabet images. Our main motive is to provide them with a helping hand to make them as much part of the community as we are.

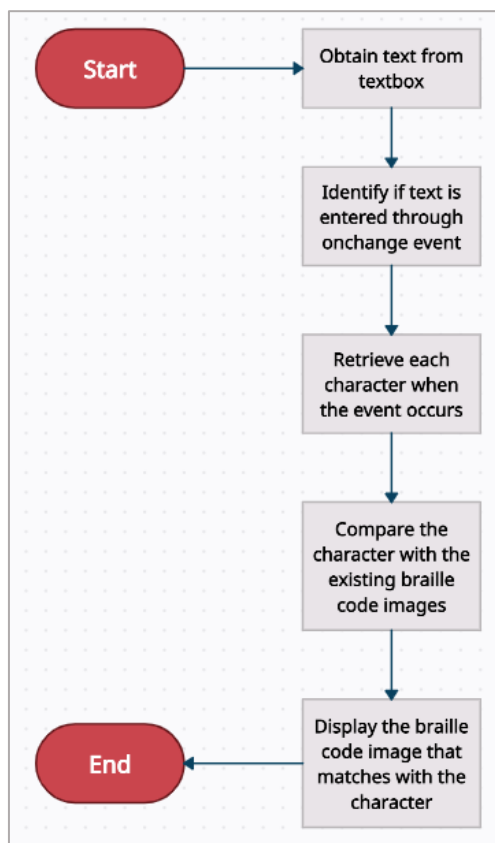
2.1. Text – Braille code Converter:

This is implemented using React-JS. When the application is launched the user will be given a text box to enter the text. Each text is converted to its lowercase and it is searched among the database for a match with our saved braille code images. The image is saved as “letter.png” form. So, when the user enters a text, the application searches for the text in the database and displays that image to the user. This is dynamic application as the output is displayed based on the text currently in the textbox and it is identified using the onchange event in the DOM.

Algorithm:

1. Get the input from the user through the textbox.
2. Identify if any change occurred in the textbox
3. Retrieve each character of the text from the textbox.
 - 3.1. Search the character with the images available in the database.
 - 3.2. If the character matches with the image name, display the image to the user.
 - 3.3. Loop from step 2 till the complete text is parsed.

Architecture Diagram:



2.2. Braille image – Text Converter:

The language used by blind people to communicate with others. These are their native alphabets used for forming sentence. As blind people cannot see or read the words and sentence, they use these symbols to communicate with others and also write sentence. But it's not very easy to read and understand all documents written in braille form hence we have developed an application using image processing to convert braille image to text so that people can read the braille text without understanding each alphabet and also help in communication between blind people and others.

This application is developed using python and libraries such as OpenCV, skimage and many more. We have the option of uploading the image and using react framework and process the work using flask API.

The process starts with uploading the image in react and passing the image to flask api using axiom library. After passing the image to flask API the image is processed and converted to OpenCV format for processing. After processing the image, the diameter and centers of circles are found and converted to matrix form. After conversion the contours are drawn and lines are drawn between letters for differentiating. After separation based on matrix value the alphabets are mapped and merged. After the sentence is formed it is passed back for printing in react app.

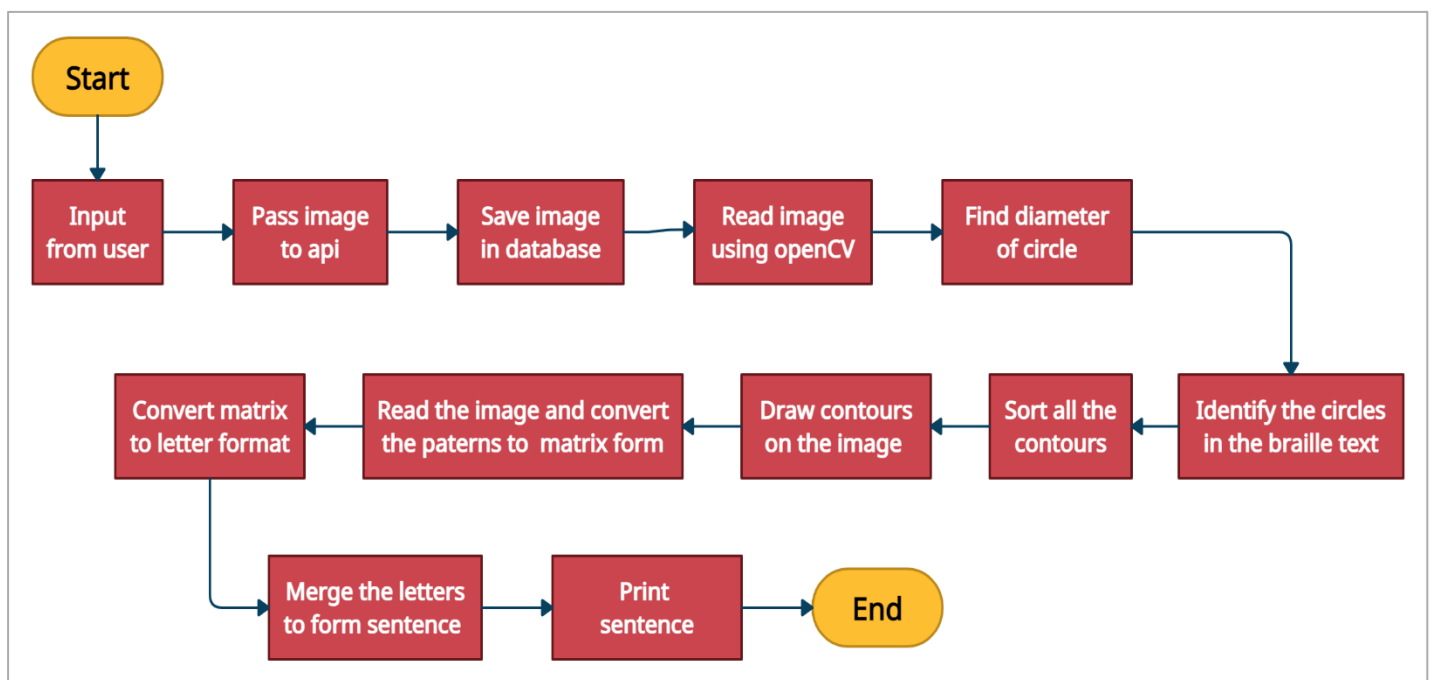
The main advantage of this application in comparison to ML or deep learning model is this process is comparatively faster and efficient in comparison to neural network method and instead of using huge dataset for training simple matrix and image processing methods can be used to achieve the task.

Algorithm:

1. Get the input file from the user in image format
 - 1.1. If no file uploaded message file not uploaded, please upload file
2. After uploading the image pass the image to the API
3. After uploading image save the image in the database
4. Read the image using OpenCV
 - 4.1. Convert image to black and white form

- 4.2. Blur image to remove noise
- 4.3. Get image edges
- 4.4. Getting contours for image
- 4.5. Group contours with a minimum of four points
- 4.6. Apply Otsu's thresholding method to binarize the image
- 4.7. Erode and dilate image to remove unnecessary noise
- 4.8. Return image details
5. Find the diameter of the circle
6. Find the circles of the braille text
7. Sort all the contours
8. Draw contours
9. Convert contour to matrix form of letters
10. Get all the letters
11. Group letters and print output and stop process the process

Architecture Diagram:



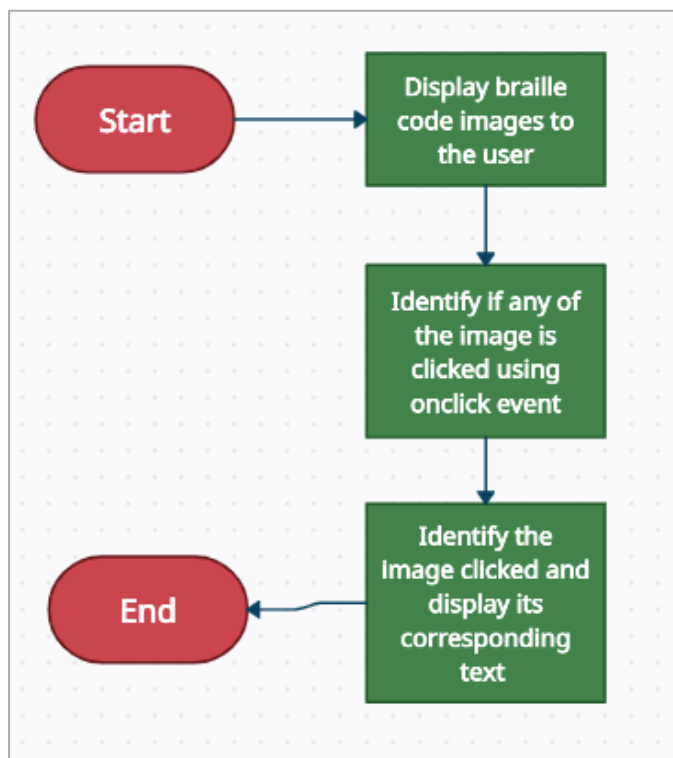
2.3. Braille Keyboard:

This is implemented using React-JS. When the application is launched, a braille code keyboard will be displayed to the user. The keyboard is set of braille code images which is displayed to the user from the database. When the user presses any braille code image, the event is captured using onclick event and the corresponding text for that braille code will be displayed. This is a dynamic application in which we can enter the modify the text inside the textbox and also convert the braille code to text and any click in the keyboard is captured through onclick event.

Algorithm:

1. The braille code images will be displayed to the user
2. Identify if the user clicks any image using the onclick event
3. Identify the image and display its corresponding text to the user in the textbox
 - 3.1. Loop from step 2.

Architecture Diagram:



Innovation / Novelty:

In the sign language converter, we have created an effective model which enables a proper communication between deaf and mute people. We have trained the neural network model in such a way that it can catch the phrases properly and display the correct GIF according to the given user input.

The braille keyboard can be used as an API for applications that can be used by blind people for better communication and interact with the rest of the community.

The text to braille converter can be used as an API for people to communicate with blind people in a better way by understanding the braille concepts even without any knowledge of braille characters. The applications that are already in the market are not equipped with this feature and moreover this application can be incorporated in many fields.

The braille code image to text conversion enables the users to understand braille text easily so that they don't have to do a tiring job of comparing each and every characters separately or have an understanding of the whole language to read braille sentences. The process using OpenCV image processing techniques is more accurate in comparison to neural network model as this reads the features in comparison to conversion of image to particular format for neural network model for better understanding.