Technology for Special Needs Education

Speak Up!

A step towards making communication easier....

Team Members

Sarita Singh (20CS10053)

Nikhil Saraswat (20CS10039)

Amit Kumar (20CS30003)

Jatin Gupta (20CS10087)

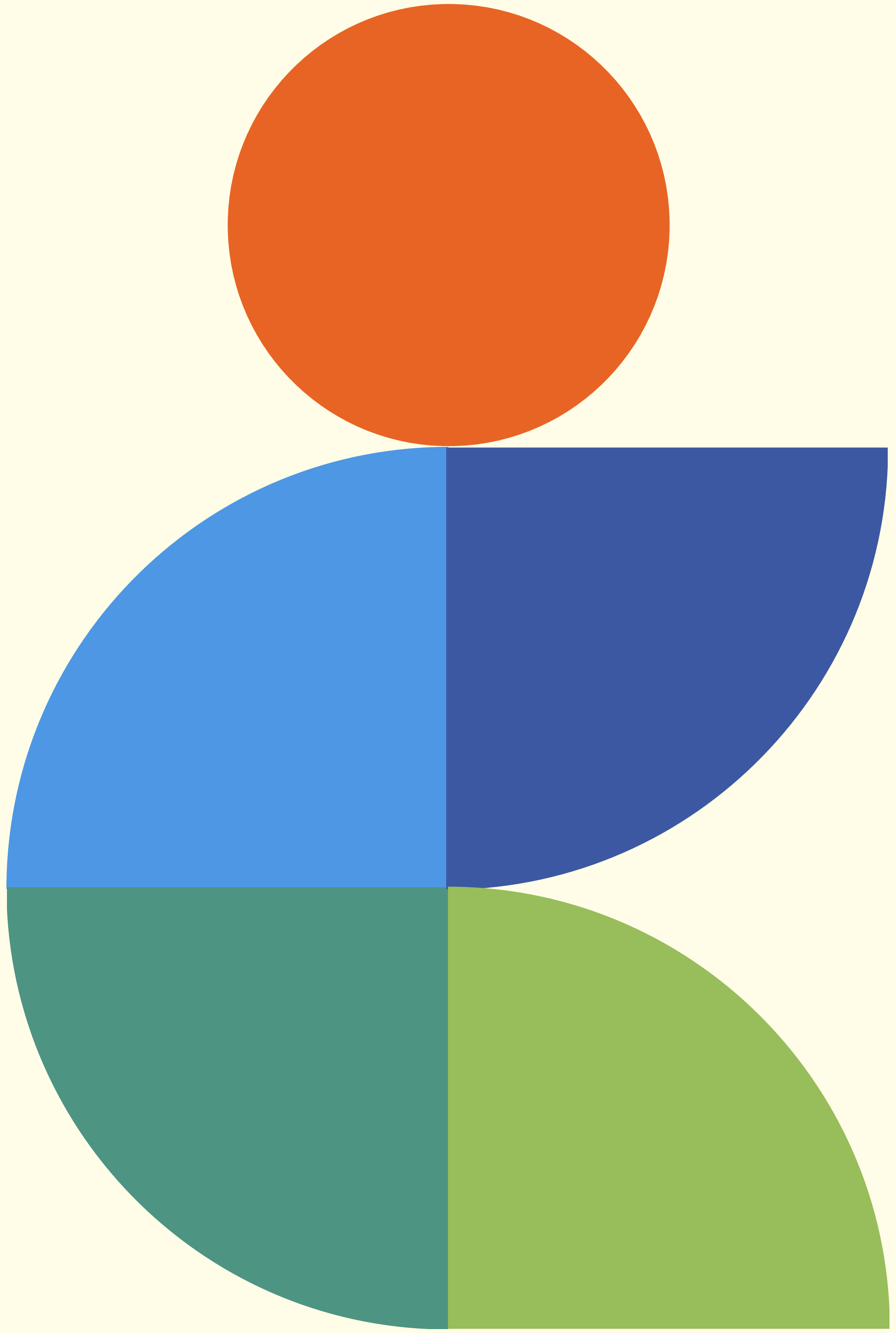
Chirag Ghosh (20CS10020)

Manami Mondal (20CS10033)

Voice to Text and Text to Braille Converter

Braille is a crucial and lively form of communication, education, and community building for persons who are blind or visually impaired.The direct attention kids require to achieve correctness and fluency in Braille is frequently difficult to deliver with a limited number of skilled teachers. There is a scarcity of skilled teachers who can teach Braille. There aren't enough tools to make learning Braille as easy as learning print.

Also, How do Deaf-Blind People Communicate? We need a text to Braille or Voice to Braille converter. This can be a tiny, portable gadget that allows them to converse with individuals who can see or speak. The gadget will feature a braille display. By putting his or her fingertips on the braille display, the deaf-blind person can read the written text.

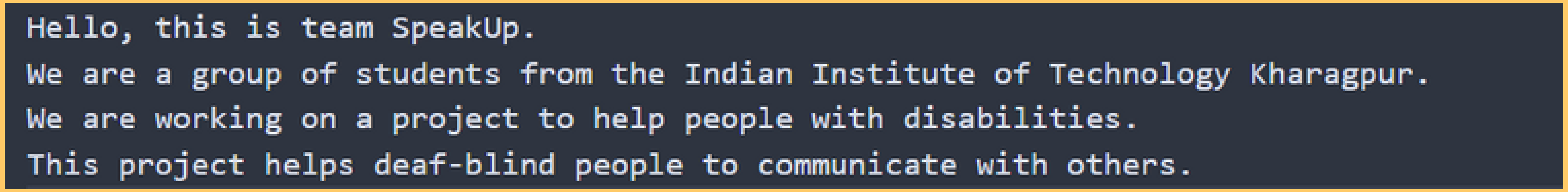


So, we are developing a Web application which can convert voice to Text and Text to Braille (We are not developing the hardware model for it). Our project covers the software part of such a hardware prototype.

General features which we are going to implement :

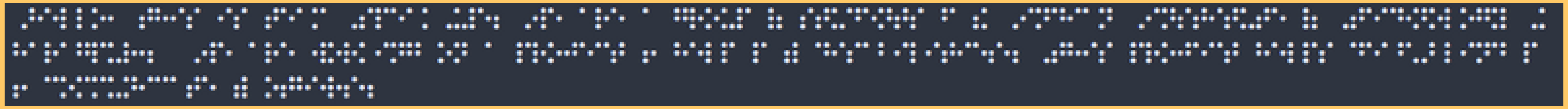
1. Voice to text - we will be implementing this by available free APIs .
2. Text to Braille - by general known algorithm to convert text to grade 2 braille (given below)
3. Braille to text - by general known algorithm to convert grade 2 braille to text

Text:



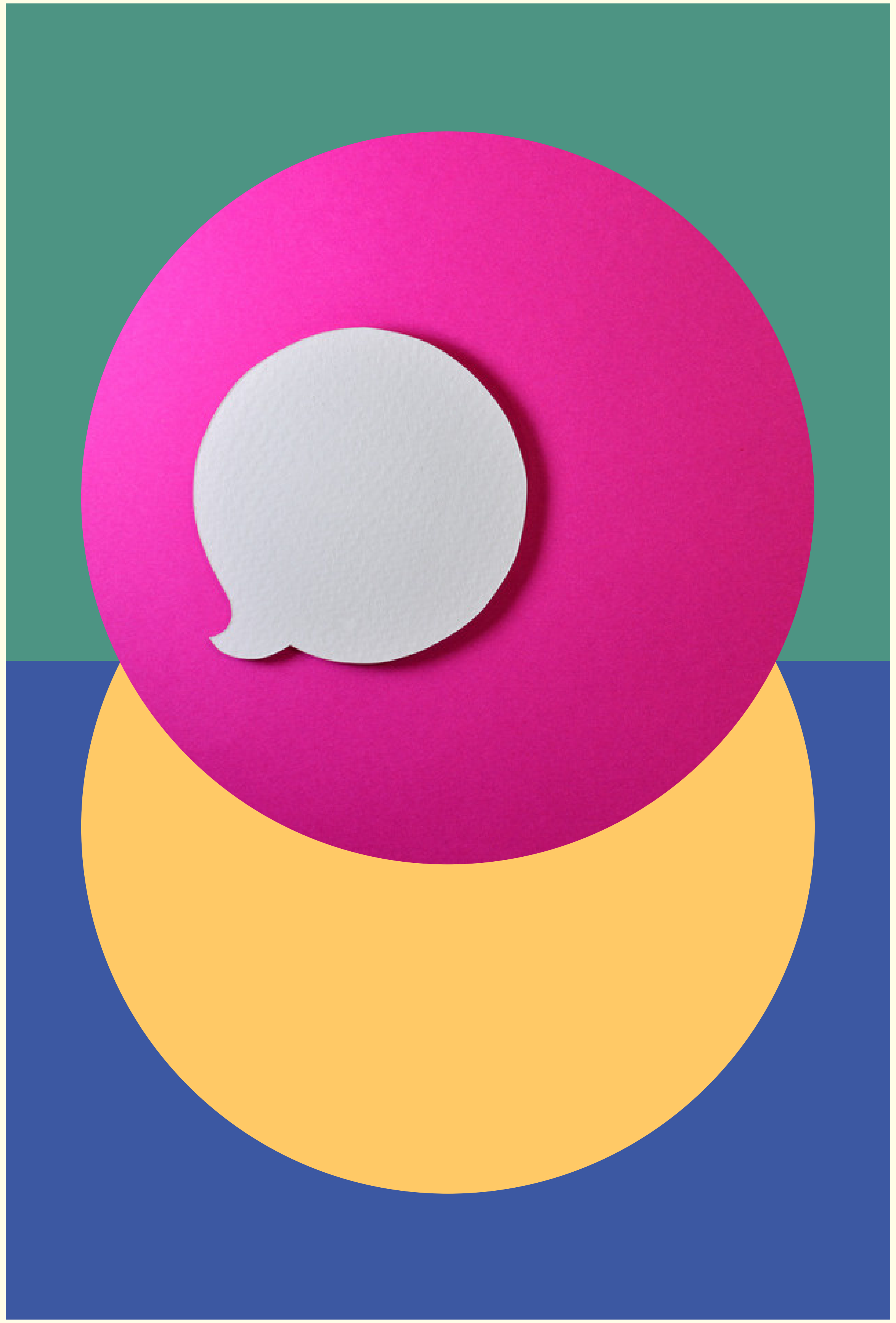
```
Hello, this is team SpeakUp.
We are a group of students from the Indian Institute of Technology Kharagpur.
We are working on a project to help people with disabilities.
This project helps deaf-blind people to communicate with others.
```

Braille:



The Algorithm

1. Whitespace includes spaces (' ') and new lines ('\n')
2. For each word, handle the numbers first.
   1. Numbers in braille use the same symbols as the first 10 letters of the alphabet.
      - The number '7' and the letter 'g' are both represented by '⠛'.
      - To differentiate between numbers and letters, an escape code (⠼) is placed before groups of numbers.
      - Therefore '7' is actually '⠼⠛' whereas 'g' is only '⠛'.
   2. In this step, only the numbers are dealt with, so there will be a mix of both braille and Alphabet symbols.
      - Example: "123-456-JUNK" becomes "⠼⠁⠃⠉-⠼⠙⠑⠋-JUNK"
3. Handle the capitals.
   1. Similarly to numbers in braille, capital letters need an escape code (⠠).
   2. The escape code (⠠) is added to the beginning of each capital letter and the letter is changed to lowercase.
      - Example 1: "⠼⠁⠃⠉-⠼⠙⠑⠋-JUNK" becomes "⠼⠁⠃⠉-⠼⠙⠑⠋-⠠j⠠u⠠n⠠k". The dashes still remain.
      - Example 2: "Sweet" becomes "⠠sweet". The non-capital letters remain untouched.
4. Trim the word.
   1. Sometimes the words extracted contain punctuation attached to them such as commas or brackets.
   2. Words need to be trimmed so that they can be converted to contractions.
      - Example: The word "the" is represented by a single braille symbol (⠮).
      - If the word "the" has punctuation around it ("the!") then it will not be interpreted correctly.
      - This is also why capitals are converted to lowercase in step 3 because "The" would not work either.
   3. The characters that are trimmed off are called "shavings".
      - Example: In the word "!where?", the shavings are "!?" and the trimmed word is "where".
5. Build the translation.
   1. Check to see if the trimmed word can be contracted.
      - This includes common words like "the", "in", "you" etc...
   2. Translate the remaining characters that are still alphabetic.
   3. Translate the shavings (this will mostly just be punctuation).
      - Exceptions to be mindful of:
        - There is no braille symbol for a generic quote (")
        - There is only open quotation (“) and closed quotation (”).
        - Therefore we must keep track of what the last quotation was to translate it correctly.



Instant Messaging

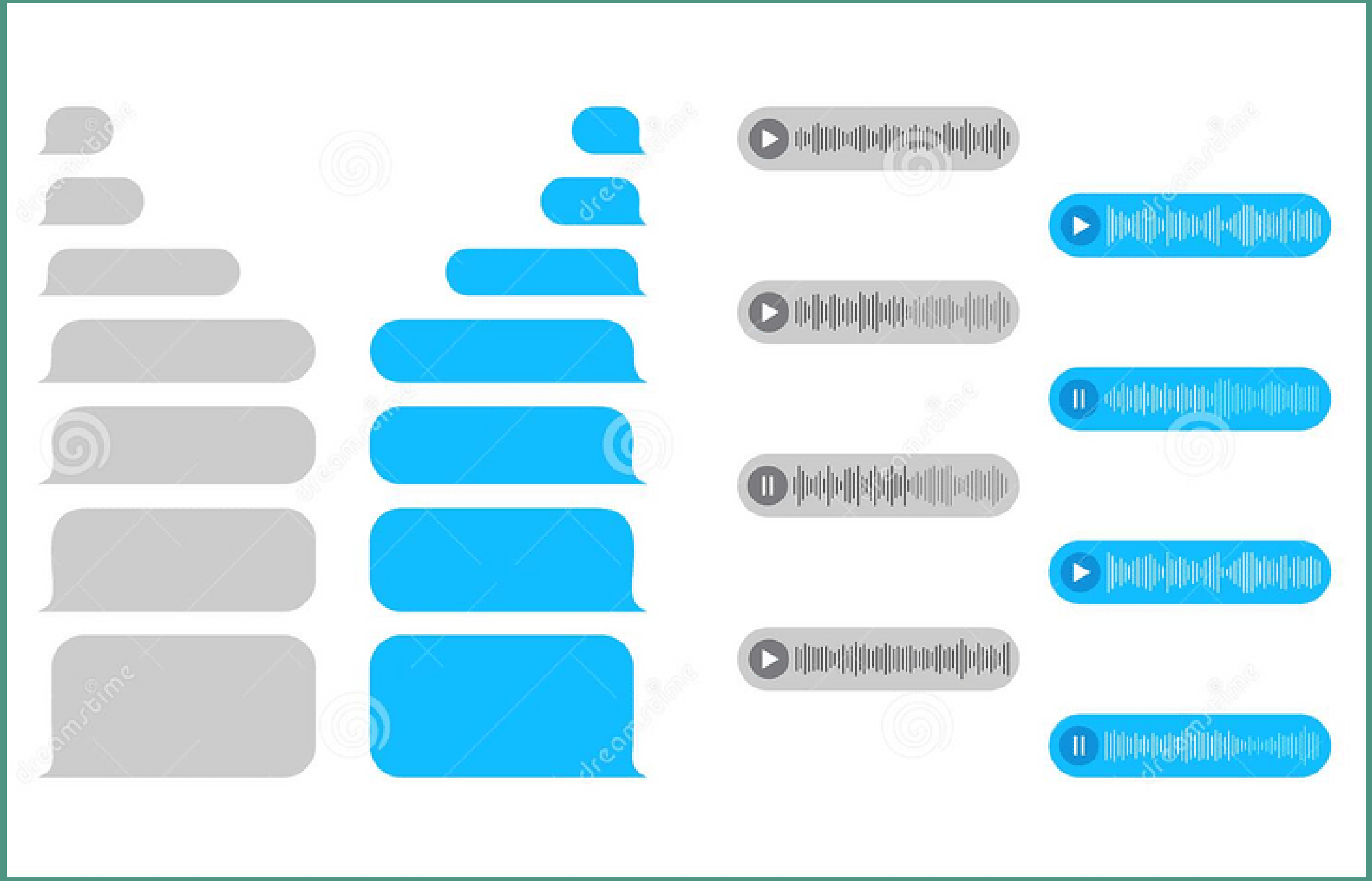
Chatting and messaging is one of the most popular methods of communication nowadays.

They can be much more than ways to communicate. Apart from communicating, one can learn new perspectives and skills via messaging and chatting with people. Communication is a great way to socially interact with one another and improve the overall personality.

People with certain disabilities find it difficult to message and chat with other people and because of this, they are not able to use this amazing facility to communicate. They might feel lonely at some time and may want to chat with someone, maybe a stranger. People with visual impairment cannot read and write text. For them, chatting can be done via audio files. Audio files can also be used by people who find it difficult to read and write. People with hearing impairment cannot listen to these audio files. So they can stick to the traditional text communication for chatting.

We thought of solving this problem and creating a feature in our software that will act as messaging software for everyone. In this software, specially-abled ones can also have a real-time lag-free chat with the other person. The software will have features that will help people with certain disabilities to chat. This feature will be developed using a socket.io server. Along with this, we'll be using Google cloud speech API for incorporating text-to-speech and speech-to-text in our software.

Our Approach



- We will establish a socket.io server for client-server communication. This server will provide real-time communication. After getting this server ready, the users will join this server for a chat.
- The user will have the option to write the message or record the message.
- If we have the input message as audio, then we will use the google cloud speech API to convert the speech to text. If we have the input message in text, then we'll be using the text-to-speech feature to convert the text message to audio. So, now we will have the input message in both audio as well as text formats.
- The socket.io server will now publish this input message to another user in both the audio format as well as the text format. The other user can then read or listen to this message and respond accordingly.
- This acts as an all-in-one real-time lag free chatting feature.

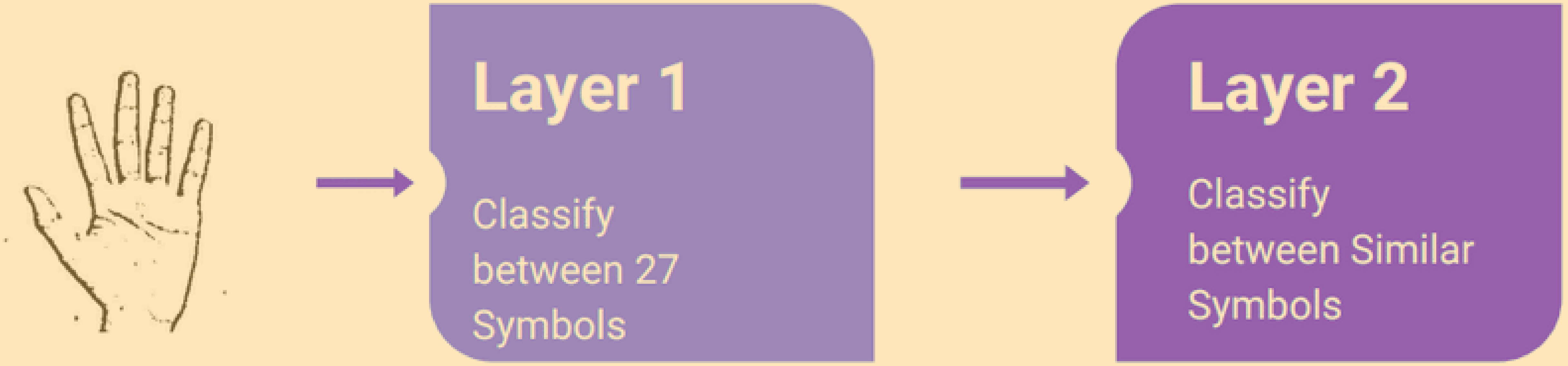
Sign Language to Text Convertor

Communication is the process of exchange of thoughts and messages through speech, signaling, behavior, and visual cues.The disability that Deaf and Mute people have is communication related and they cannot use spoken languages.

Sign language is one of the oldest and most natural forms of language for communication. But the majority of people do not know sign language and interpreters are very hard to find. Hence we have planned to develop a real-time method using neural networks for Sign language.

We will implement a Vision based approach. In our approach, computer camera will be the input device for observing the information of hands or fingers. The approach will require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision by describing 7 artificial vision systems that are implemented in software and/or hardware. The main challenge of vision-based hand detection will be to cope with the large variability of human hand's appearance due to a huge number of hand movements as well as variations in view points.

Our approach



Layer 1

Classify between 27 Symbols

Layer 2

Classify between Similar Symbols

- We will implement 26 symbols(A-Z) and blank of Indian Sign Language (ISL).
- To predict the final symbol of the user we will use two layers of algorithm.
- In algorithm Layer 1 we will apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.
- This processed image will be passed to the CNN (Convolution Neural Network) model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
- We will consider the space between words using the blank symbol.
- In algorithm Layer 2 we will detect various sets of symbols which show similar results on getting detected.
- Using the data we will then classify between those sets using classifiers made for those sets. We convert sign language to text, aiding those unable to comprehend sign language.

Book Narrator

Since as we all know books play a crucial role in development of a human being, but due to visual impairment many people who are book-lovers face difficulties to read books unlike a normal person.

Audio-books might help such people a lot who have visual impairments. By using audio-books reading book is more like story-telling, which makes a person easy to understand the book content. Audio-books are not only helpful for a person with visual-imparment but this is also useful for a normal person.
We can develop a feature which takes the book (pdf) as input, parse it and then convert it to audio and narrate it.

Our Approach

- For this feature, we need to obtain text from a pdf, which can be done using a pdf parsing tool.

- Sometimes in a book there are also some images (screenshots of some text) which also need to be converted to text, so using Optical Character Recognition (OCR) (Optical Character Recognition (OCR) is the process that converts an image of text into a machine-readable text format.)
- We also need a tool to stitch the audio files of each page to make a single audio file of the whole book, which can be done easily using audio merging tools.
- We can use PyPDF python module to parse pdfs of book.
- We can use OCR and audio APIs of Azure Services for OCR and audio requirements.

Handwriting Reader

We have also planned to work up a feature into our app that lets you scan handwritten documents or short notes that can then be fed to our book reading feature to be read aloud, i.e. it takes in handwritten notes and processes a text based document that can be read and further processed as required.

This will help the people who are visually impaired to understand the content of handwritten documents.

A Neural Network (NN) which is trained on word-images from the IAM dataset will be built. As the input layer (and therefore also all the other layers) can be kept small for word-images, NN-training is feasible on the CPU (of course, a GPU would be better). It consists of convolutional NN (CNN) layers, recurrent NN (RNN) layers and a final Connectionist Temporal Classification (CTC) layer. The NN consists of 5 CNN and 2 RNN layers.

The text is recognized on character-level, therefore words or texts not contained in the training data can be recognized too (as long as the individual characters get correctly classified).
The popular Long Short-Term Memory (LSTM) implementation of RNNs is used, as it is able to propagate information through longer distances and provides more robust training-characteristics than vanilla RNN.
The word beam search decoder is used instead of the two decoders shipped with TF. Words are constrained to those contained in a dictionary, but arbitrary non-word character strings (numbers, punctuation marks) can still be recognized.

Thank you!