Sign Language Recognition

Project members:

Bijesh Patel Vachanni

Suman Pandey

Salim Jivani

Divya Geethanjali

Mohammed Johny

Workflow:

The project will be hosted in Github, so we can collaborate and work on separate parts of the application together. We will be building our own CNN using Tensorflow, and getting the data from Kaggle. The UI for the project will either be built with TKinter or in a Web Application using Flask. We are seeing which will work better for our current application.

Communication via Discord

- Text Channels
- Voice Channels

Workflow application

- Github in Visual Studio Code

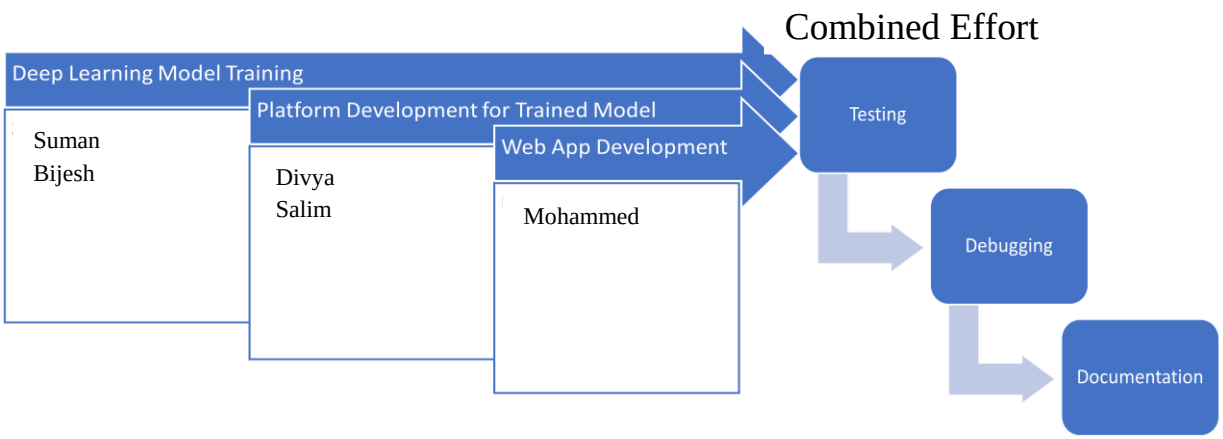

Fig 1: Project role distribution

Project Abstract:

We want to build an application that will live stream a person's hand while they are using sign language and be able to decipher what they are trying to communicate. First, we will start with simple letters, and put the words together. Then, move to grabbing gestures for a full conversation.

Eventually, we could see this becoming part of an AR headset application. Where in real time, you will be able to see the sign language and communicate with that person.

Project background and objectives:

Project usage:

- Sign Language Recognition helpful in communication between mute people with people who do not understand sign language.
- It breaks off the communication barrier and inbuilds confidence among the people who are unable to hear.

Figure 2: Estimated result of project

Project Design:

- The front end will be done with TKInter or Web Application using Flask, which detects sign language alphabetic letters: A-Z represented with hand gestures.
- We will use the MNIST Dataset that is stored in CSV file format from kaggle website that contains train and test dataset.
- Generate training, validation batches by using data generators and then design the CNN model to train and evaluate the model on the created data set.
- Getting the train and validation accuracy and losses.
- Load the model for prediction using Keras.
- Python, Numpy, Kera, Tensorflow will be used in the project.
-

Milestone:

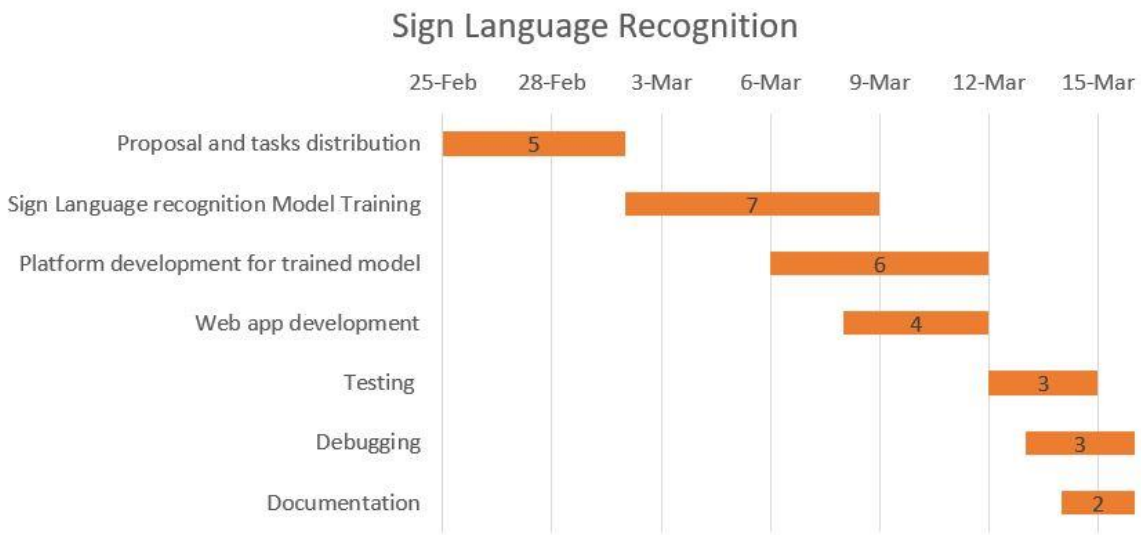

Fig 3: Gantt chart of Sign Language Recognition

Resources and Related Projects:

"Sign Language Recognition Using Python and OpenCV." *DataFlair*, 15.Sept.2020,data-flair.training/blogs/sign-language-recognition-python-ml-opencv/.

Tecperson. "Sign Language MNIST." *Kaggle*, 20 Oct. 2017, www.kaggle.com/datamunge/sign-language-mnist.