Rajiv Gandhi Institute of Technology
Chola Nagar, Bengaluru, Karnataka - 560032

Sign Language Interpreter using Deep Learning

Under the guidance of : Asst. Prof. Pushplata Dubey

- Bahaduri Prachiti Jagdish - 1RG17CS009
- Mohammed Azim - 1RG17CS030
- Mohammed Touseef - 1RG17CS031
- Prajwal B Mani - 1RG17CS037

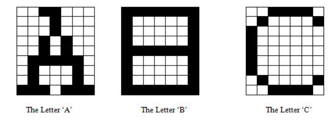
ABSTRACT

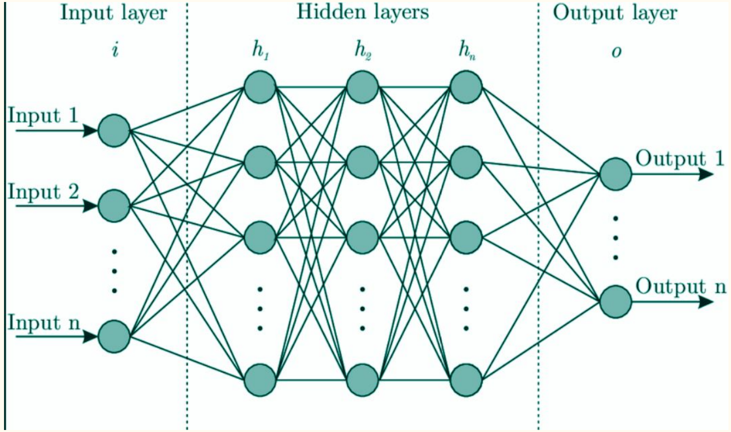
Speech Impairment is a disability, which affects an individual’s ability to communicate using speech and hearing. This brings about the difficulty for both the sign and non - sign language speakers to communicate with each other. With recent advances in deep learning and computer vision, the focus of our project is to create a vision of an end to end Convolutional Neural Network that will be trained on the ASL(American Sign Language) dataset then modeled on robust architectures like GoogLeNet/MobileNet architecture and deploy it on an android application so that it will have more accessibility and provides an ease of use, thus aiding communication between signers and non-signers. It is a challenging and interesting problem that if solved will bring a leap in social and technological aspects alike.

INTRODUCTION

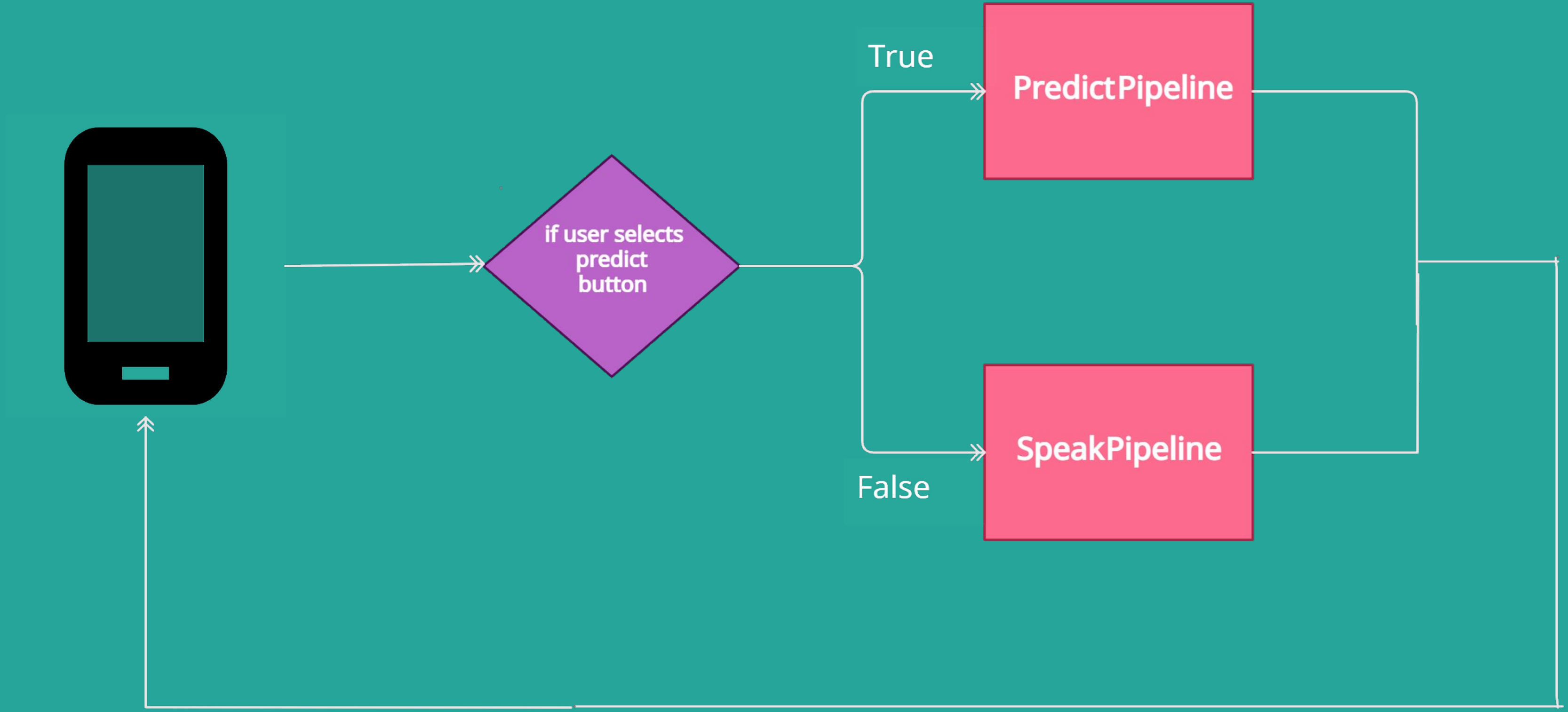
- A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data.
- A neural net consists of thousands or even millions of simple processing nodes that are densely interconnected.
- COMPONENTS OF NEURAL NETWORK:
  - Input Layer
  - Hidden Layer
  - Output Layer
- APPLICATIONS:
  - Speech Recognition

  - Character Recognition

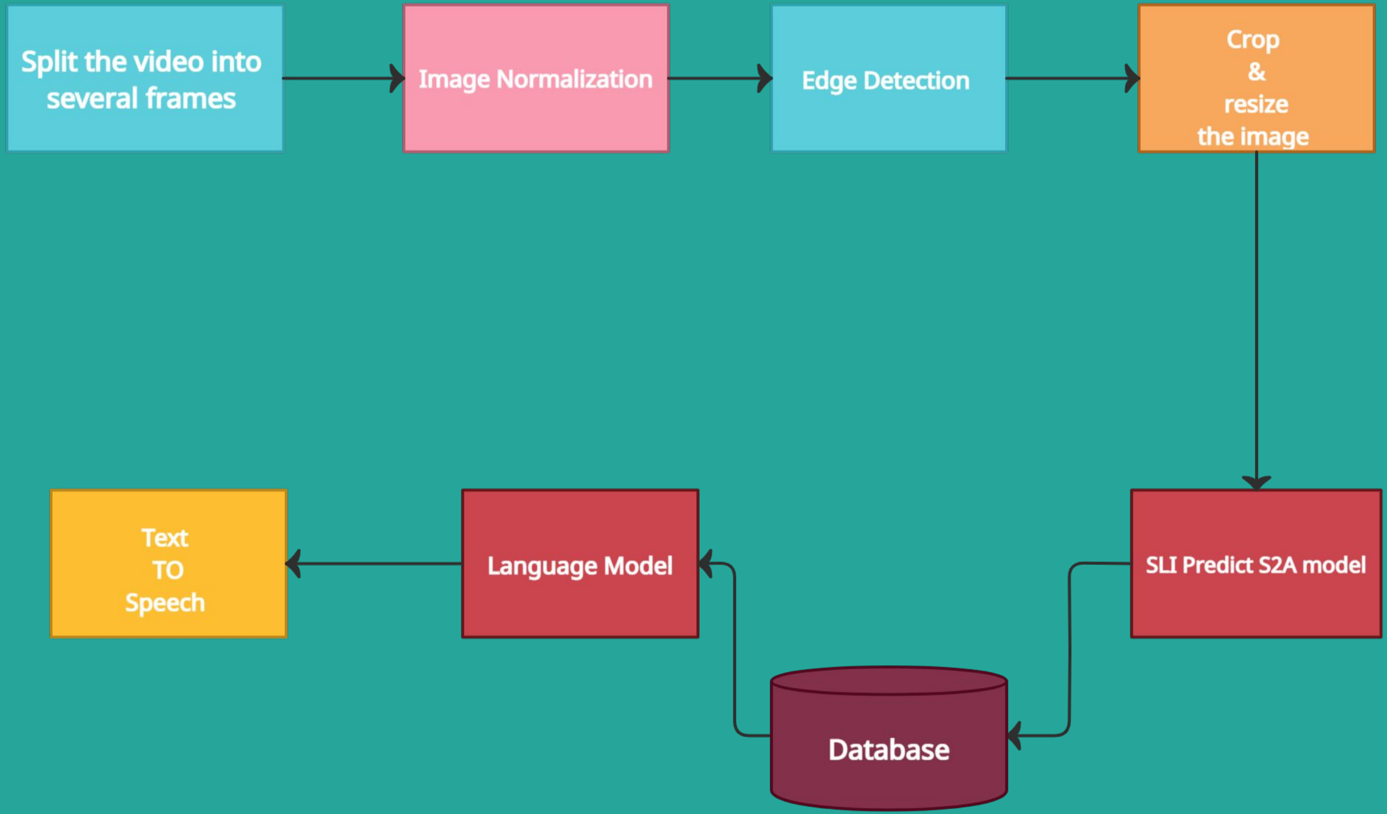




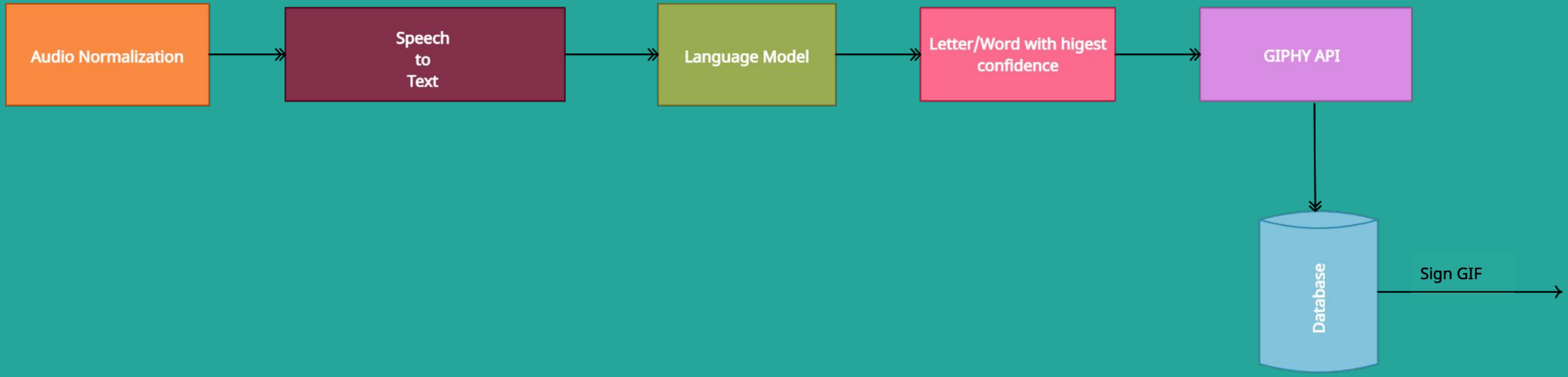
ARCHITECTURE



PREDICT PIPELINE

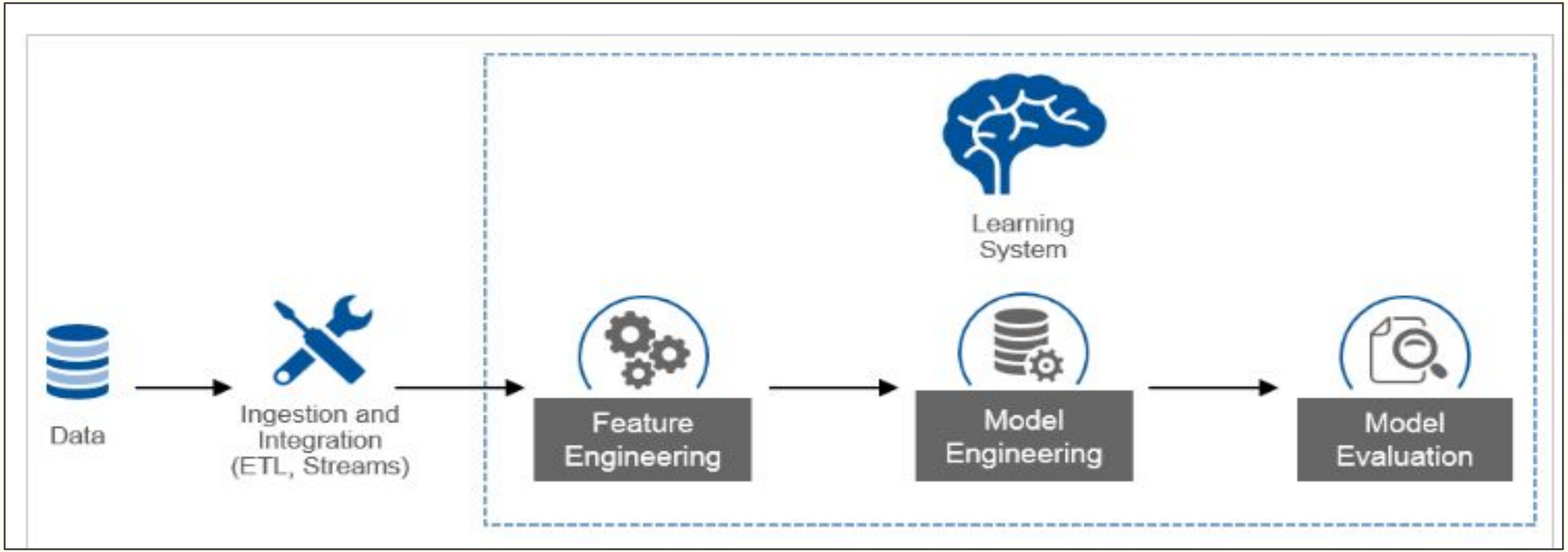


SPEAK PIPELINE



METHODOLOGY

Development life cycle of our project



IMPLEMENTATION COMPONENTS

Predict Module

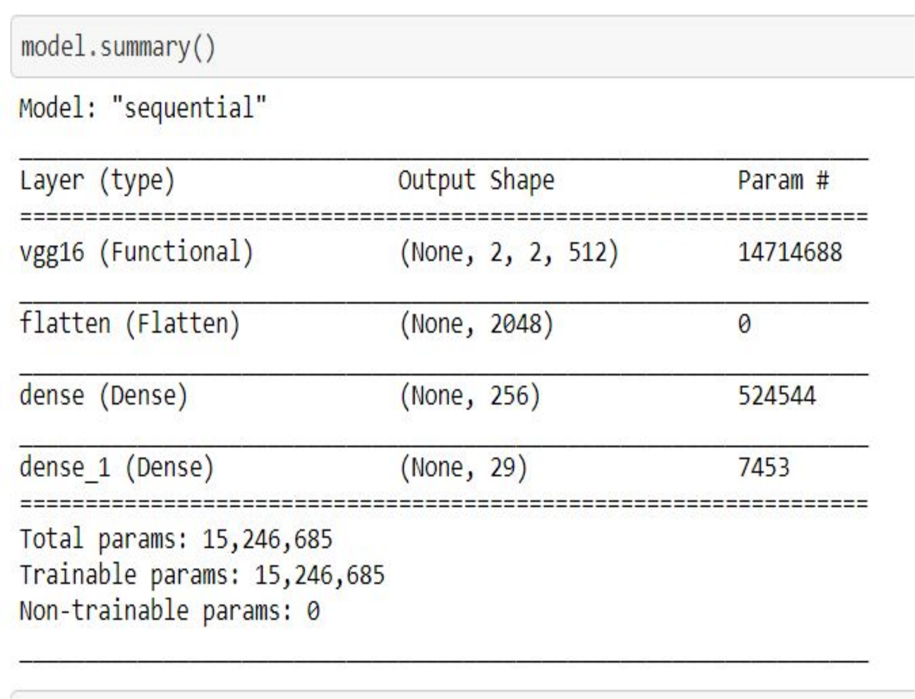
Image PreProcessing:

- We use computer algorithms to perform image processing on digital images so that our models has better image data to learn from.
- We have used OpenCV library for preprocessing
- Some of the preprocessing techniques are video to images, normalization, resizing and edge detection

CNN Model :

- We have a VGG16 CNN architecture model that is trained on ImageNet weights and our dataset using Tensorflow and Keras
- EarlyStopping keras callback is used so the model doesn’t overfit for the data

More about CNN model



```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 2, 2, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 256)               524544
_________________________________________________________________
dense_1 (Dense)              (None, 29)                7453
=================================================================
Total params: 15,246,685
Trainable params: 15,246,685
Non-trainable params: 0
_________________________________________________________________
```

- We have used two activation functions that is “relu” and “softmax”
- Adam optimizer
- Categorical_crossentropy loss function
- Early stopping callback is monitored on the validation loss

Language correction model(Language model):

- The main role of this model is to check for grammar error and spelling mistakes
- This is built on Phoneme and Symspell where Phoneme uses Soundex algorithm and Symspell uses edit-distance concept.

Text to Speech:

- We have used google text-to-speech(gtts) api to convert the text to audio which does its job perfectly.

Speak Module

Audio Preprocessing :

In order for the neural network to process the auditory information in the same manner as it does visual information so we do audio preprocessing.

Some of the python modules used in audio preprocessing pyAudioAnalysis and PYO

Few preprocessing steps are noise cancellation,normalization and noise injection

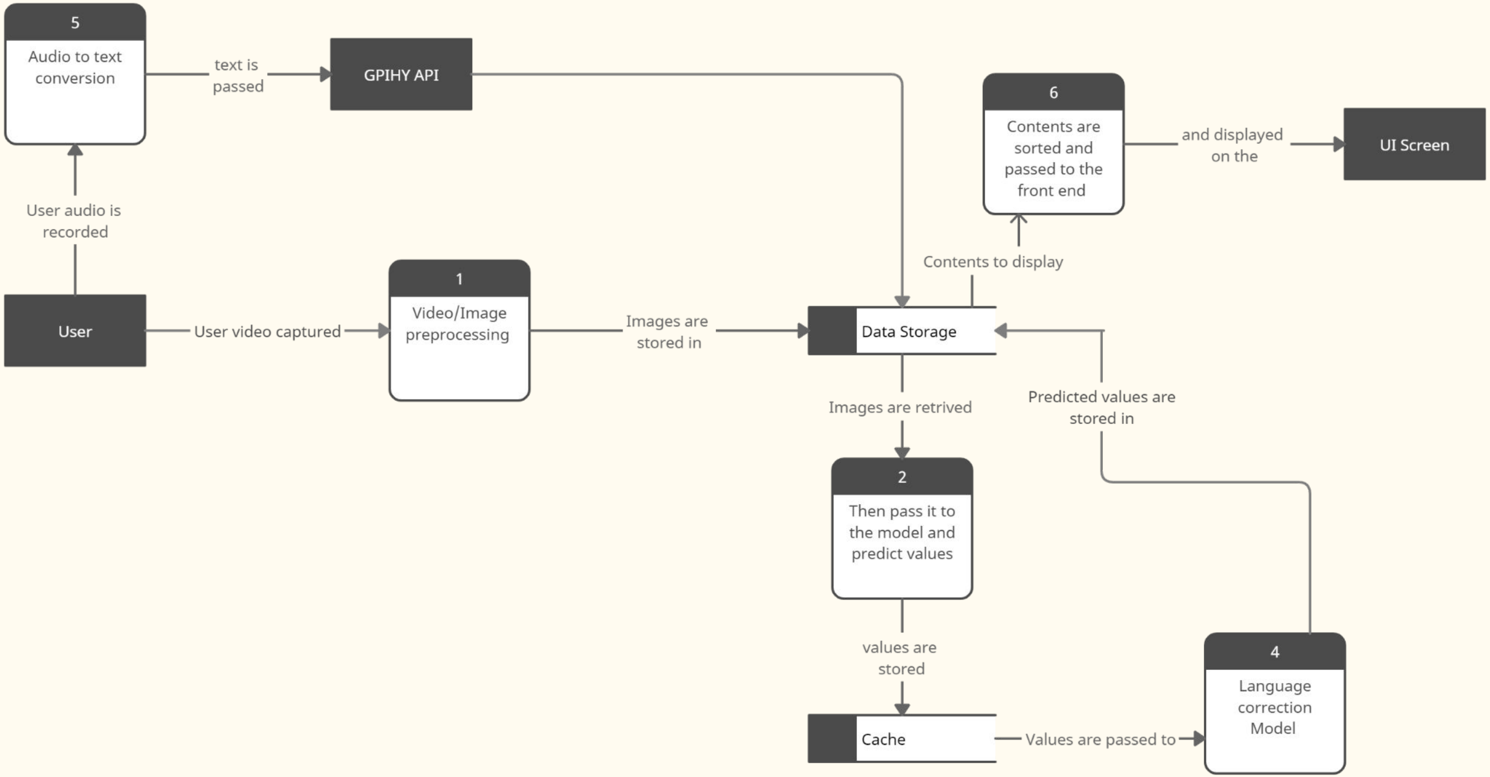
Speech to Text:

For speech to text we have used google-cloud-speech and Speech Recognition libraries

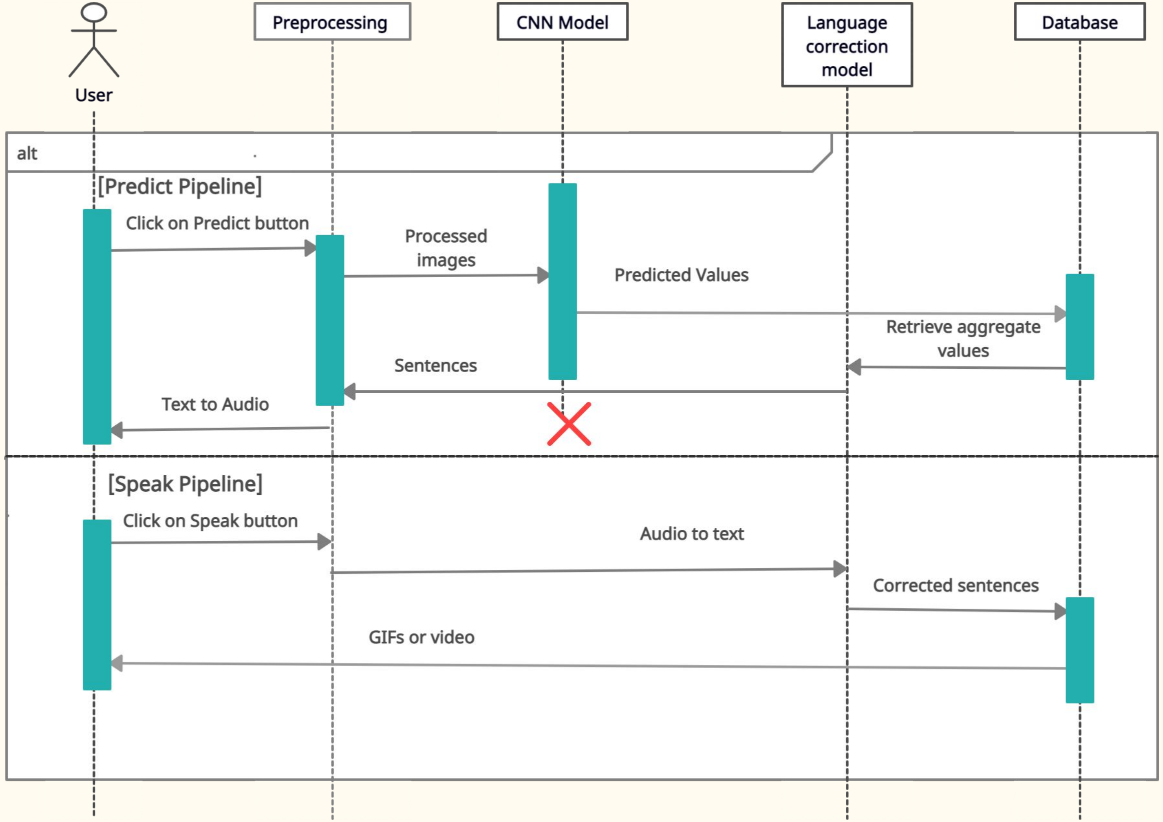
Giphy API:

- There is total of more than one million words in english
- Giphy is a online database and search engine that allows users to search for GIF(Graphics Interchange Format) and share all over the internet
- Giphy as a category for sign language which is recorded by Robert
- Using there Api to make our App simple, flexible and access there abundant resource
- At last all the gifs are stored in database and retrieved when it is needed and displayed to the user

DATA FLOW DIAGRAM



SEQUENCE DIAGRAM

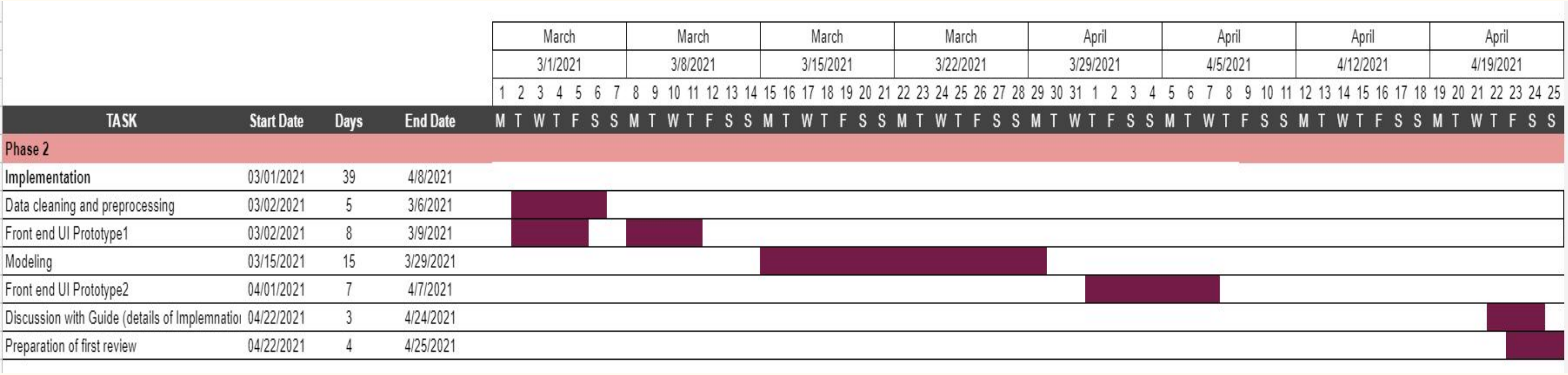


PROGRESS SO FAR

What did we build till now?

- A VGG16 Model that has capacity to identify the sign language
- A language correction model to helps to detect errors and fix it
- Image and Audio Pre-Processor
- Giphy script to handle gifs
- Front end UI

All the modules that are implemented as a separate entity



TASK	Start Date	Days	End Date
Phase 2			
Implementation	03/01/2021	39	4/8/2021
Data cleaning and preprocessing	03/02/2021	5	3/6/2021
Front end UI Prototype1	03/02/2021	8	3/9/2021
Modeling	03/15/2021	15	3/29/2021
Front end UI Prototype2	04/01/2021	7	4/7/2021
Discussion with Guide (details of Implemnatioi	04/22/2021	3	4/24/2021
Preparation of first review	04/22/2021	4	4/25/2021

CONCLUSION

A sign language is a natural mode of communication used by the deaf community. Building end to end model with pipelines make it faster and efficient. Tensorflow and kivy gives us the tools build this entire process with ease.

What's next?

We need to integrate all this pieces together and complete this puzzle and see the end picture.
There is always room for improvement and try out new cutting edge technology.