ASL Document

Backend:

File 1: ASL-VS.ipynb

- Preprocessing of data(pictures roundabout 86000)
- Splitting into train test (20% test , 80% train)
- Applying CNN(Convolutional Neural Networks: tensorflow keras) model for training(on preprocessed data) on 29 classes(26 alphabets and 3 others), using activation function : **Relu** in hidden layers and **softmax** in outer layer.
- After training model has been saved in 'ASL_model' folder (it is in the files) *Do not need to run this file as model has been saved and it takes roundabout 9 to 10 hours*

File 2: ASL-Cam_handler.ipynb

- Using Computer vision for detection of hand gestures
- Using mediapipe library for for the preprocess of the image and to draw landmarks and connections on image
- Extracting key points in the form of numpy array from the detected landmarks
- loading model's architecture from a JSON file, then loading the corresponding model weights from an HDF5 file, and compiling the model using the Keras library.

  At cell no 8 change the path of the files of model.json and model_checkpoint.h5
  With your correct path : here changes are needed`with`

```
open('/Users/samannazir/model.json', 'r') as json_file:
   loaded_model_json = json_file.read()


# Load the ASL model architecture
ASL_model = model_from_json(loaded_model_json)


# Load ASL model weights
ASL_model.load_weights('/Users/samannazir/model_checkpoint.h5')
```

- videoCapturing using OpenCV
- Making live predictions using all things mentioned above

File3: asl_app.py

In it making a flask app as a python script, using the same code as mentioned In the ‘ASL-Cam_handler.ipynb’ file. The purpose of making a flask app is to define routes that map URLs to functions. These functions are responsible for handling HTTP requests and generating responses.

Basically it is creating a url.

change the path of the files of model.json and model_checkpoint.h5

With your correct path : here changes are needed line no 12 AND LINE NO 19

```
with open('model.json', 'r') as json_file:
    loaded_model_json = json_file.read()

# Load the ASL model architecture
ASL_model = model_from_json(loaded_model_json)

# Load ASL model weights
ASL_model.load_weights('model_checkpoint.h5')
```

Remember to install flask in your python setup ‘pip install flask’

Now, run your virtual environment through your command prompt and then run : asl_app.py **on it, do not close it**

This will create http url like given below,
http://192.168.100.11:5000/camera_feed_with_predictions
so replace this ip address ‘**192.168.100.11:5000**’
With your computer IP Address

If you want You can check this url whether working or not on your browser.

Frontend:

Open the ASLDetection folder, and run it on android studio
It is made in java.

The folder contain three xml files and 3 java files
For starting page : **activity_main.xml** and **MainActivty.java**
For click button page : **SecondActivity.java** and **second.xml**
For prediction camera page : **WebViewActivity.java** and **webview.xml**

In **WebViewActivity.java** file make sure to replace the url with your url.
And run it on your emulator

Make sure while detecting signs, for good prediction keep a blank background and hand signs properly visible to the camera

Link: https://drive.google.com/file/d/1_OhKjvy7mOajW4Q6cWD_s_b04xLR-Cpe/view