SIGNARA

GRADUATION PROJECT

REAL-TIME ARABIC SIGN LANGUAGE TRANSLATOR
FOR BARRIER FREE COMMUNICATION AMONG DEAF AND HEARING INDIVIDUALS

TECHNICAL REPORT

Table of Contents

Team Members

Peter Boshra Aziz

dev.peterboshra@gmail.com
linkedin.com/in/peterboshra/
https://github.com/PeterBushra

Education

Computer Engineering @ Modern Academy

Aya Tarek Elghannam

ayatarek9832@gmail.com
linkedin.com/in/aya-elghannam
github.com/AyaElghannam

Education

Electronics & Communications Engineering
@ Alexandria University

Introduction

Signing has always been part of human communications. Newborns use gestures as a primary means of communication until their speech muscles are mature enough to articulate meaningful speech. For thousands of years, deaf people have created and used signs among themselves. These signs were the only form of communication available for many deaf people. Within the variety of cultures of deaf people all over the world, signing evolved to form complete and sophisticated languages.

Sign language is a form of manual communication and is one of the most natural ways of communication for most people in deaf community. There has been a re-surging interest in recognizing human hand gestures. The aim of the sign language recognition is to provide an accurate and convenient mechanism to transcribe sign gestures into meaningful text or speech so that communication between deaf and hearing society can easily be made

The significance of using hand gestures for communication becomes clearer when sign language is considered. Sign language is a collection of gestures, movements, postures, and facial expressions corresponding to letters and words in natural languages, so the sign language has more than one form because of its dependence on natural languages.

The sign language is the fundamental communication method between people who suffer from hearing impairments. In order for an ordinary person to communicate with deaf people, an interpreter is usually needed to translate sign language into natural language and vice versa.

Human-Computer Interaction (HCI) is getting increasingly important as a result of the increasing significance of computer's influence on our lives. Researchers are trying to make HCI faster, easier, and more natural. To achieve this, Human-to-Human Interaction techniques are being introduced into the field of Human-Computer Interaction. One of the richest Human-to-Human Interaction fields is the use of hand gestures in order to express ideas

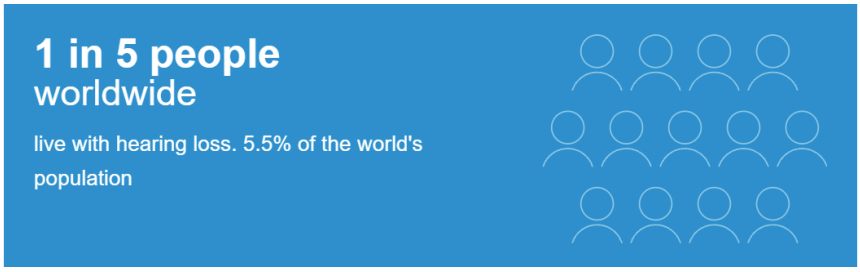
Problem Statement

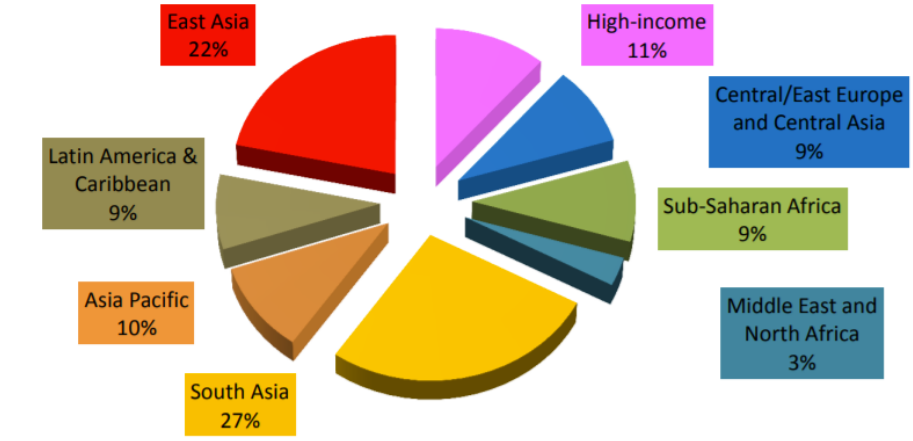
Overview

A person is said to have hearing loss if they are not able to hear as well as someone with normal hearing, meaning hearing thresholds of 20 dB or better in both ears. It can be mild, moderate, moderately severe, severe or profound, and can affect one or both ears. Major causes of hearing loss include congenital or early-onset childhood hearing loss, chronic middle ear infections, noise-induced hearing loss, age-related hearing loss, and ototoxic drugs that damage the inner ear.

The impacts of hearing loss are broad and can be profound. They include a loss of the ability to communicate with others delayed language development in children, which can lead to social isolation, loneliness, and frustration, particularly among older people with hearing loss. Many areas lack sufficient accommodations for hearing loss, which affects academic performance and options for employment. Children with hearing loss and deafness in developing countries rarely receive any schooling.

WHO estimates that unaddressed hearing loss costs the global economy US$ 980 billion annually due to health sector costs (excluding the cost of hearing devices), costs of educational support, loss of productivity, and societal costs.





MBD, WHO, 2012 DHL estimates; DHL adult threshold is ≥41 dB, adults of 15 years or older.

Prevalence

Deafness and hearing loss are widespread and found in every region and country. An estimated 466 million people worldwide – 5.5% of the population – have disabling hearing loss, and this number is expected to rise to 1 in 4 by 2050.

Low- and middle-income countries bear a disproportionate burden from hearing loss. WHO estimates that global hearing aid production covers just 3% of the need in these countries.

Difficulties the Hearing Impaired face every day:

1. Sign language misunderstandings

Sign language is far from universal, and different standards exist in different countries (for example, the differences between Egyptian and Libyan Sign Language are quite significant). In addition, regional areas have their specific variations—just like accents or slang—leading to further difficulty. There are many instances of professional interpreters using the wrong words due to the variations across regions and countries; while this may not seem like a big deal, it has led to lasting harm, such as in legal situations or miscommunication during hospital visits.

2. Job applications and interviews

Job interviews are already stressful situations; now consider being hearing impaired. Those who are hard of hearing or deaf may sometimes feel completely ignored when they reveal their hearing status on application forms, possibly because recruiters see it as too much extra work to accommodate them. When they do reach the interview stage, more complications arise. Telephone interviews are nearly impossible without an interpreter, and in-person interviews can be difficult to carry out if an interviewer is unprepared for the situation.

3. Depression and anxiety

Studies reveal that deaf people are around twice as likely to suffer from psychological problems such as depression and anxiety. Research suggests this stems from feelings of isolation. Making matters worse

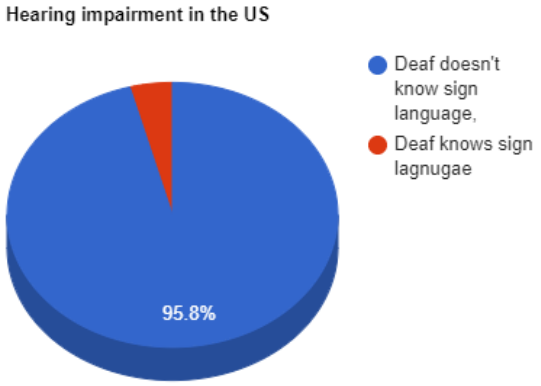
4. Social Isolation

Nine in 10 deaf children are born to hearing parents, yet less than a third have family members who sign regularly.

You can imagine the emotional and psychological toll of not being able to communicate with those closest to you, let alone others at school or work. For many deaf individuals living in rural areas, they might be the only deaf person in their community or school, making it extremely challenging to build relationships.

In addition to social isolation, some research shows that deaf children, in particular, are more vulnerable to abuse, neglect, and sexual assault than their hearing peers—the results of which can have a lasting impact on both mental and physical health.

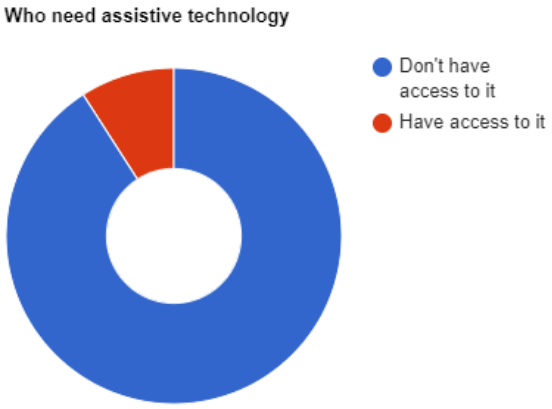
Another problem that faces the hearing impairment community is lack of sign language knowledge even among hearing-impaired themselves, example in the US even though there are 48,000,000 deaf individuals only 2,000,000 are Sign language Speakers.



The Seventy-first World Health Assembly 26 May 2018

Having considered the report on improving access to assistive technology
Considering that one billion people need assistive technology and that, as the global population ages and the prevalence of noncommunicable diseases increases, this figure will rise to more than two billion by 2050.

Noting that assistive technology enables and promotes the inclusion, participation and engagement of persons with disabilities, aging populations, and people with co-morbidities in the family, community, and all areas of society, including the political, economic, and social spheres; Recalling that 90% of those who need assistive technology do not have access to it and that this has a significant adverse impact on the education, livelihood, health, and well-being of individuals, and on families, communities, and societies.



Proposed Solution

We used the power of assistive technology to increase the quality of life by opening up new opportunities to disabled people and increasing the range of options open to them.

Our team developed an artificial intelligence (AI) powered system for the deaf and mute people, Implementing our solution on mobile and web applications will offer a low-cost superior approach to translating Arabic sign language into text and vice versa in real-time.

Introducing SIGNARA the easy-to-use innovative digital interpreter dubbed as "Google Translate for the deaf and mute" works by placing a smartphone or webcam in front of the user while the app translates gestures or sign language into text. SIGNARA uses neural networks and computer vision to recognize the video of a sign language speaker and then smart algorithms translate it into speech. And the vise versa works by animating the input text into sign language.

The proposed Arabic Sign Language Translator system does not rely on using any gloves or visual markings to accomplish the recognition job. As an alternative, it deals with images of bare hands, which allows the user to interact with the system in a natural way.

In this technical report, we will show how we tackled our problem giving a detailed way of thinking and design approaches that allowed us to build this prototype

Deep Learning Approaches

Sign to text

The goal of SIGNARA is to develop software that is capable of real-time translation of Arabic sign language into text. Due to resources and time constraints, the scope has been limited to completing 9 Words and the whole Arabic alphabet.

We achieved the words sector using Mediapipe - a framework for building multimodal, cross-platform, applied ML pipelines. While the alphabets sector Uses transfer learning. Now we will discuss technical details of the 2 sectors:

Words

Challenges we faced while doing this part for having a dynamic data that can be represented as series of motion we had many questions during the research process

- How long does a word sign take ? is there deviation ?
- How to deal with FPS variance ?
- When to use dynamic neural network and when to use static (Alphabet) ?

- How long does a word sign take ? is there deviation ?

We agreed to implement the basic language words on SIGNARA which we found that it takes about 1 second, with a large deviation So we implemented our code to collect a 30 fps for the dataset. to capture the variations during the movement.

- How to deal with FPS Variance

We dealt with our approach using LSTMs which requires a static number of inputs.let's say our LSTM model accpets the last 60 frames If the FPS of the device is 10, the LSTM will process the last 6 seconds of video data - while if the FPS were 30, the LSTM would process the last 2 seconds of video data. It's straightforward to control the last (x) seconds of frames, but it's less so to control the number of frames in the last (x) seconds

- When to use dynamic neural network and when to use static (Alphabet) ?

Our system has two neural network (one for static alphabet will be discussed later and one for dynamic words), So we decided to make a virtual button in our screen that will change the mode to either word mode or character mode by clicking with both index and middle fingers combined.

Dataset

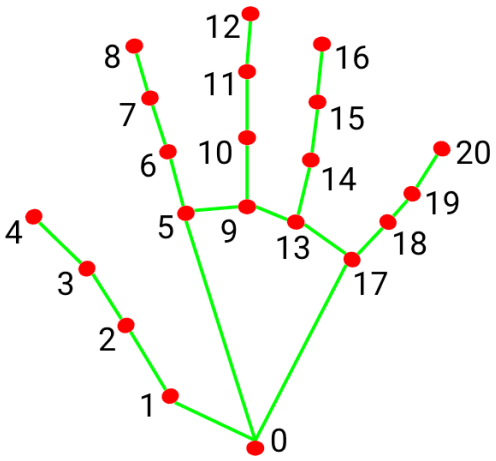
Step one: Data Collection

After doing online research we found that there is no public dataset for the Arabic Sign Language words which contain continuous motion. So we started collecting our dataset by capturing video that takes 30 and 60 Frames of the word motion. We Captured 120 videos of 2 different data collectors to have a variety.

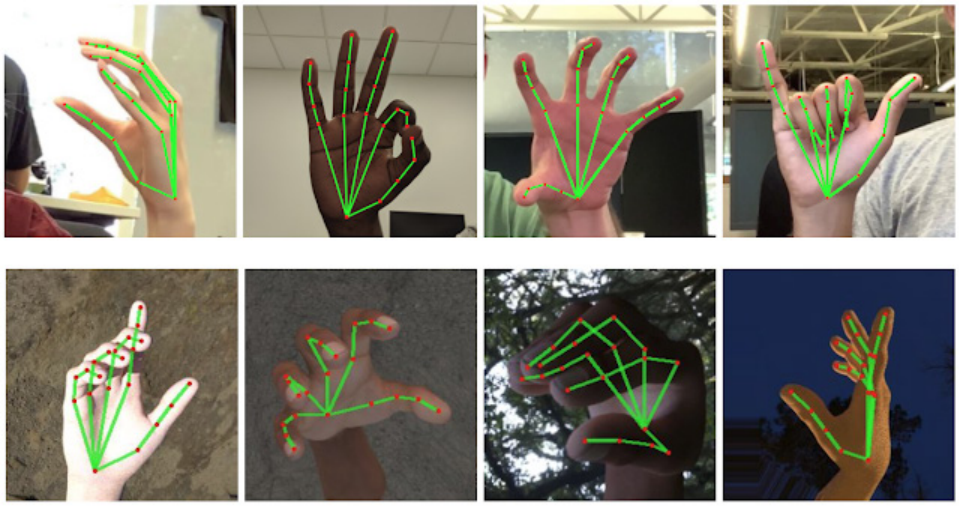
Step Two: Transforming Images to Keypoints

The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. For example, it can form the basis for sign language understanding and hand gesture control, and can also enable the overlay of digital content and information on top of the physical world in augmented reality. While coming naturally to people, robust real-time hand perception is a decidedly challenging computer vision task, as hands often occlude themselves or each other (e.g. finger/palm occlusions and handshakes) and lack high contrast patterns.

MediaPipe Hand is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame. Whereas current state-of-the-art approaches rely primarily on powerful desktop environments for inference, Mediapipe method achieves real-time performance on a mobile phone, and even scales to multiple hands.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP



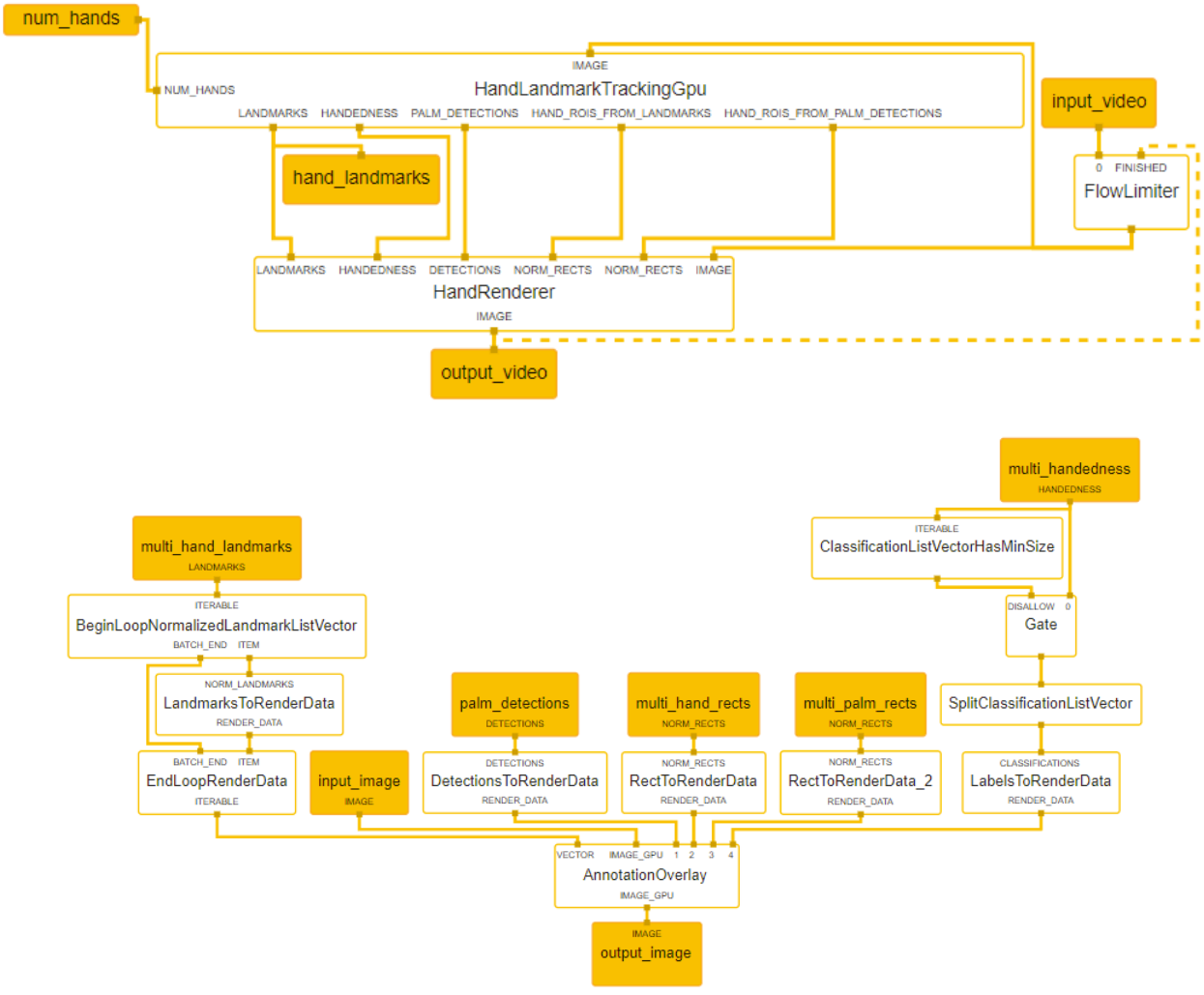
Mediapipe Pipeline

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: A palm detection model that operates on the full image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand keypoints.

Providing the accurately cropped hand image to the hand landmark model drastically reduces the need for data augmentation (e.g. rotations, translation, and scale) and instead allows the network to dedicate most of its capacity towards coordinate prediction accuracy. In addition, in Mediapipe's pipeline, the crops can also be generated based on the hand landmarks identified in the previous frame, and only when the landmark model could no longer identify hand presence is palm detection invoked to relocalize the hand.

Collection of detected/tracked hands, where each hand is represented as a list of 21 hand landmarks, and each landmark is composed of x, y, and z. x and y are normalized to [0.0, 1.0] by the image width and height respectively. z represents the landmark depth with the depth at the wrist being the origin, and the smaller the value the closer the landmark is to the camera.

Normalized Keypoints landmarks provide a scale Invariant property to our system, So after extracting the key points from the hands we transformed each location and saved it in a .npy file as an array of Xs and Ys.



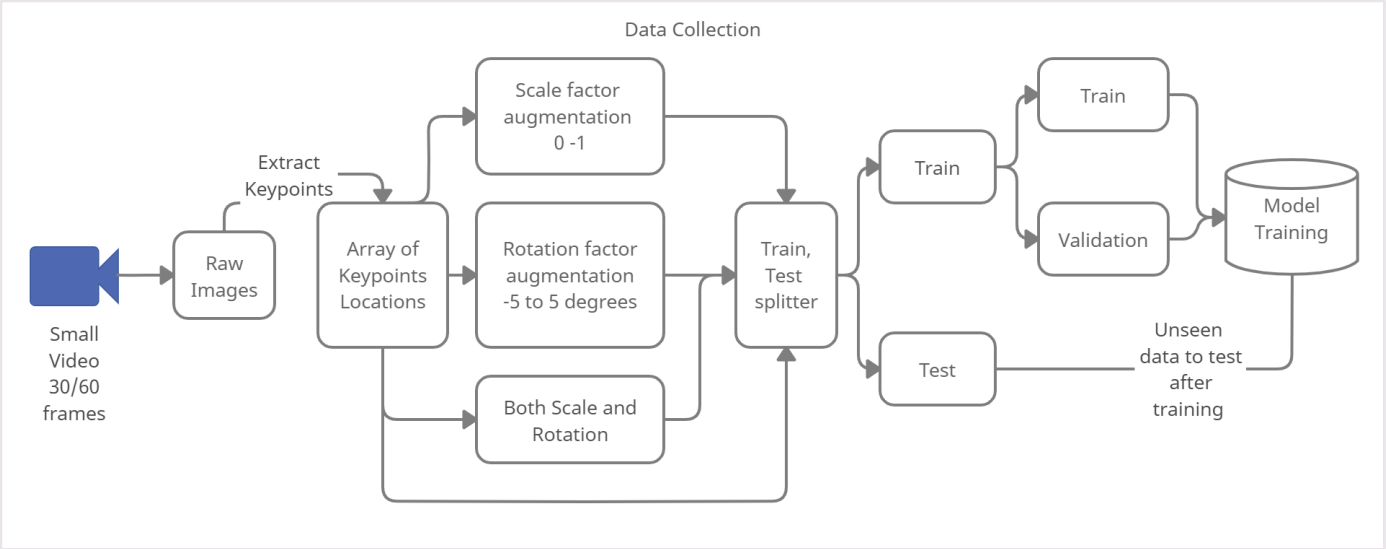
Step Three: Landmarks Augmentation

Data augmentation we used 2 types of augmentation that are normally variable with respect to the user during system usage the scale (how close is the user to the camera) and the orientation (what angle is the camera held). That's how we obtained our dataset in a variety of conditions. We accounted for these situations by training our Neural Network by adding synthetically modified data

We used 2 augmentation

1- by adding a random scale factor from 0 to 1.
2- by adding a random rotation factor from -5 degrees to 5 degrees.
3- by adding both 1 and 2.

Class	Original	Augmented	Total
مرحباً	120	360	480
انا	120	360	480
كيف حالك	120	360	480
اخ	120	360	480
اخت	120	360	480
بابا	120	360	480
ماما	120	360	480
مصر	120	360	480
امام	120	360	480



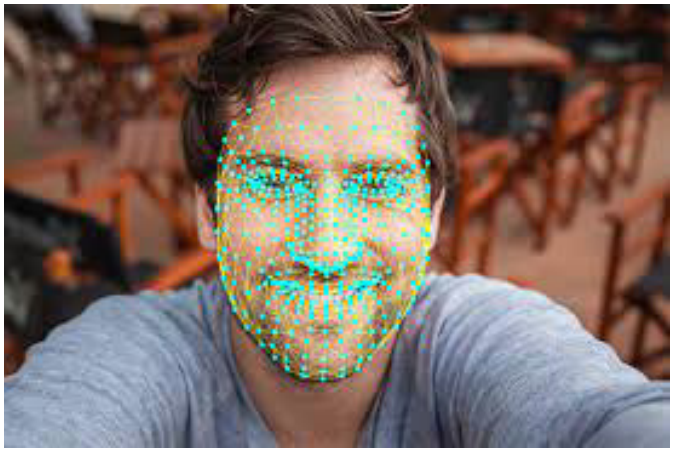
Feature Engineeing

Before the neural network it was challenging for us to decide what are the features that will mainly affect the motion without confusing the model we tried 4 different features.

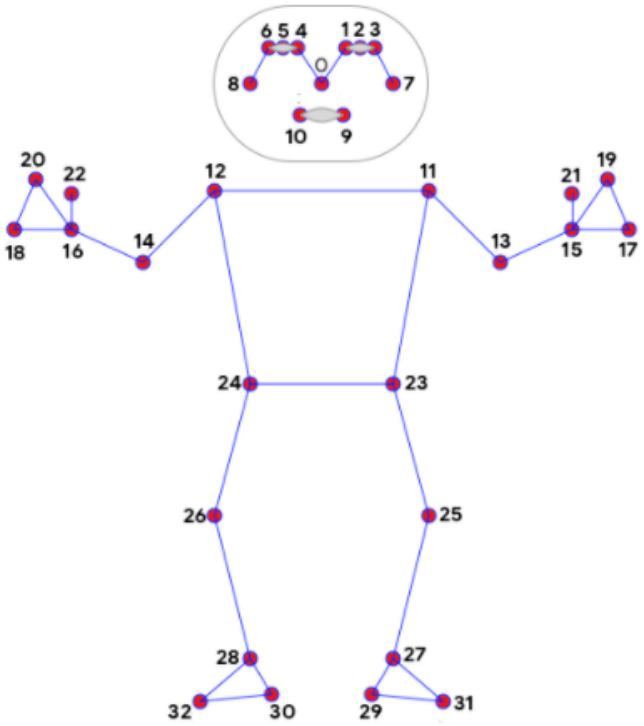
1- Pose + Hands + face Landmarks Coordinates in each frame.
2- Difference between X,Y of pose + hands landmarks between 2 consecutive frames.
3- Distance between Hand Landmarks and specific points of human pose.
4 - Hands Landmark Only X,Y Coordinates in each frame.

1- Pose + Hands + face Landmarks X,Y Coordinates in each frame

Using Mediapipe face Mesh - a face geometry solution that estimates 468 3D face landmarks - which retrieved the X,Y,Z Coordinates of the 468 points of the face with support of multiple faces.



Using MediaPipe Pose - a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks in addition to visibility factor and background segmentation mask on the whole body from RGB video frames. with only upper body taken in consideration 23 3D landmarks



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

In addition to previously mentioned Mediapipe Hands - Retrieves 21 3D landmarks for each hand.

Number of features was huge 468*3 + 23*4 + 21*2*3 = 1622 it was huge number of features leading to noise and given the dataset which wasn't big.

2- Difference between X,Y of pose + hands landmarks between 2 consecutive frames.

Same as first approach but instead of using X,Y,Z Coordinates we used the difference between each 2 consecutive frames and removed the Z and visibility Coordinates to decrease the number of features and make evey frame dependant on previous one. In addition to removing face mesh.

Number of features was huge 23*2 + 21*2*2 = 130 fewer number of features than first approach that decreased the noise in a very good way and improved the performance but still the accuracy is about 50-60%.

3- Distance between Hand Landmarks and specific points of human pose.

This time we thought that the noise coming from the 2 previous approaches was due to different locations of the user within the frame he may be doing the same Sign but he is located in the left or right of the frame not in the middle therefore we thought about what is constant in the user movement that we may take the distance between his hands and it?

We came up with an approach by which we calculate the distance between hands landmarks and his shoulders and nose landmarks from pose landmarks (11,12,0) since the shoulders are unchanged while applying the sign. This approach enhanced the model accuracy but was noisy as the distance varies from body to body.

Number of features was huge 3*2 + 21*2*2 = 90 fewer number of features than first two approaches.

4 - Hands Landmark Only X,Y Coordinates in each frame.

After noticing the output Coordinates change in our videos we found out that the only obvious changes happens in the hands coordinates and taking advantgae of the normalized output by the frame dimensions. we thought that decreasing the number of features may help in our case.

Number of features was huge 21*2*2 = 84 fewer number of features than first two approaches.

Approaches we thought about but didn't apply

- Relative angles of the Finger joints may be a very useful feature to generalize with all the signs out there but since our model is only based on 9 words we didn't apply it.

- Lips from Mediapipe face Mesh, in some words there are some different lips movement asocciated with the sign that may be a helpful feature.

- Doing algorithm instead of the pure X,Y Coordinates after doing research we foud that calculating the z-score of each coordinate may help improving the accuracy and decreasing the Noise

Model

Human activity recognition is the problem of classifying sequences of accelerometer data recorded by specialized harnesses or smart phones into known well-defined movements.

Classical approaches to the problem involve hand crafting features from the time series data based on fixed-sized windows and training machine learning models, such as ensembles of decision trees. The difficulty is that this feature engineering requires strong expertise in the field.

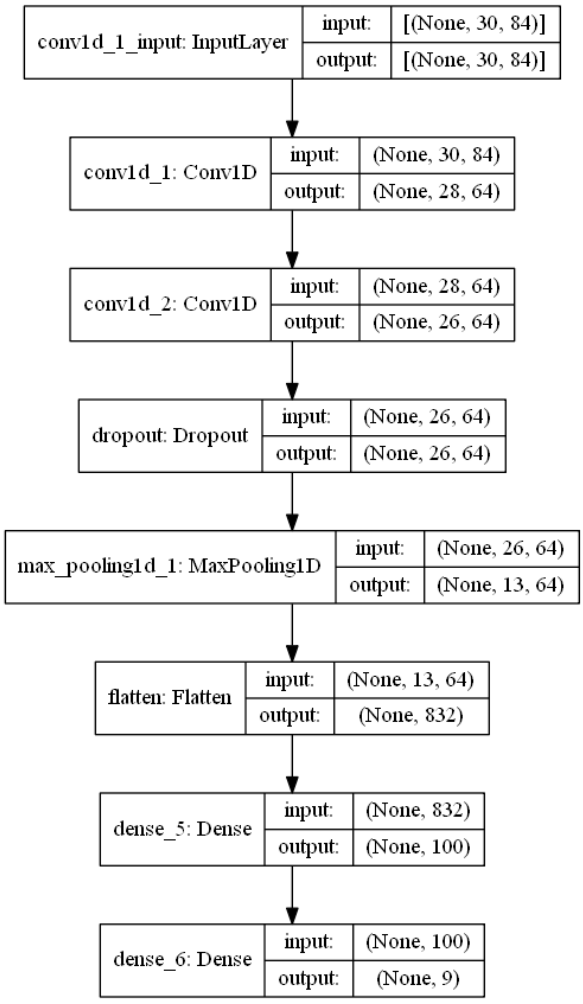
Deep learning methods such as recurrent neural networks like as LSTMs and variations that make use of one-dimensional convolutional neural networks or CNNs have been shown to provide state-of-the-art results on challenging activity recognition tasks with little or no data feature engineering, instead using feature learning on raw data.

We applied our knowledge and tried many deep learning algorithms in order to get insghts of the performance by comparing different approaches.

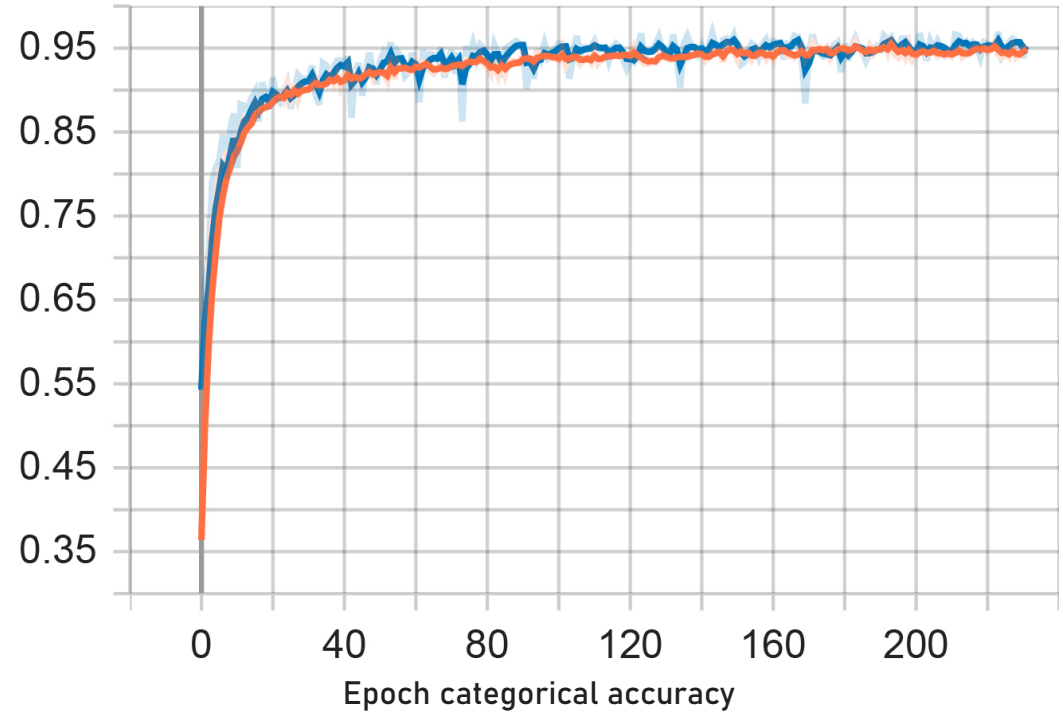
CONV1D

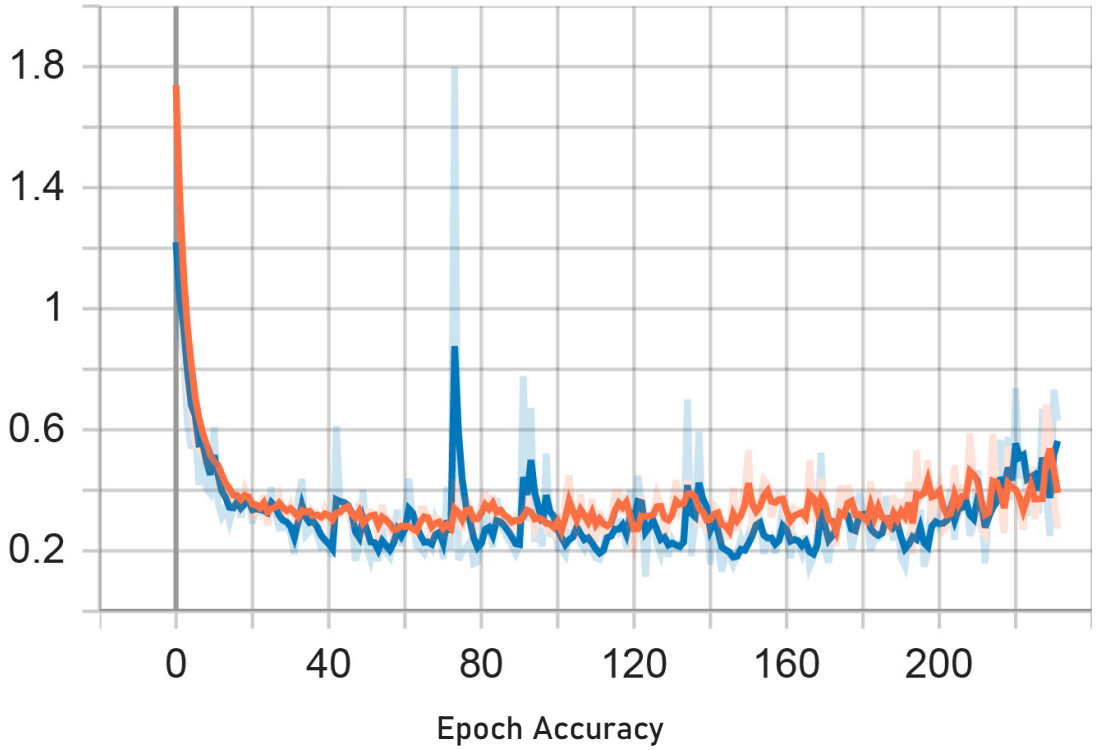
When working with images, the best approach is a CNN (Convolutional Neural Network) architecture. The image passes through Convolutional Layers, in which several filters extract important features. After passing some convolutional layers in sequence, the output is connected to a fully-connected Dense network.

Model Architecture



Training Results



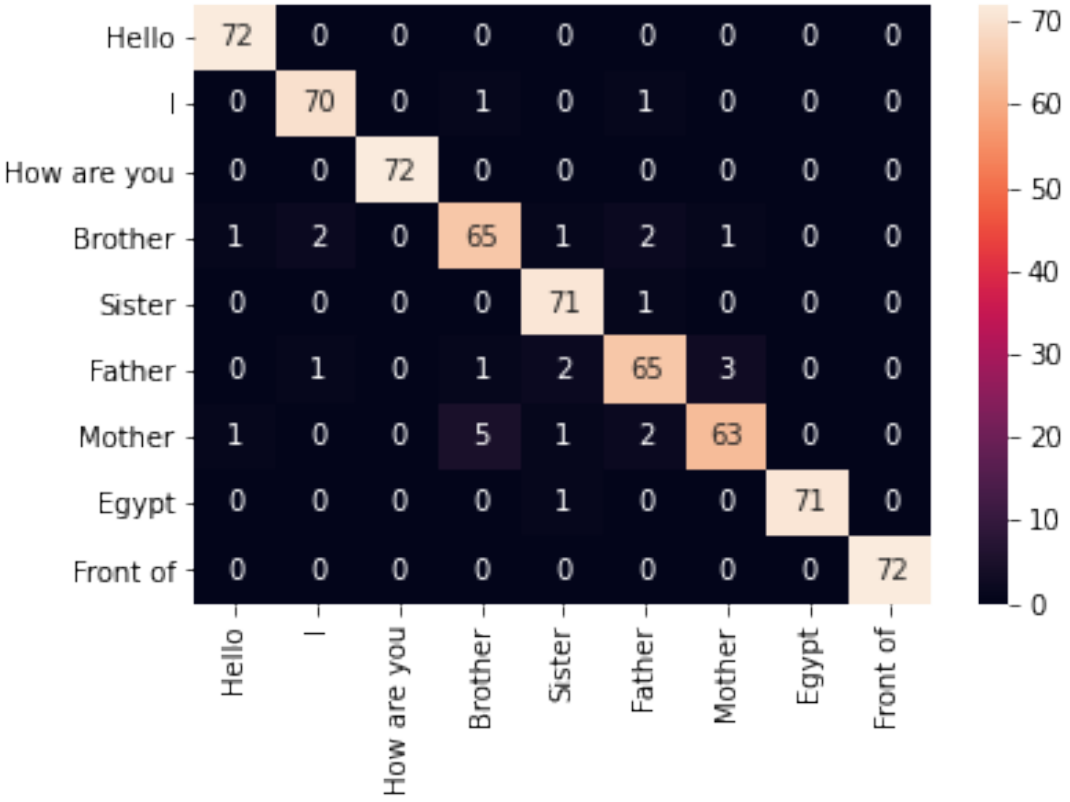


Results

Training Accuracy = 0.9248
Training Loss = 0.265

Validation Accuracy = 0.93981
Validation Loss = 0.3305

Confusion Matrix

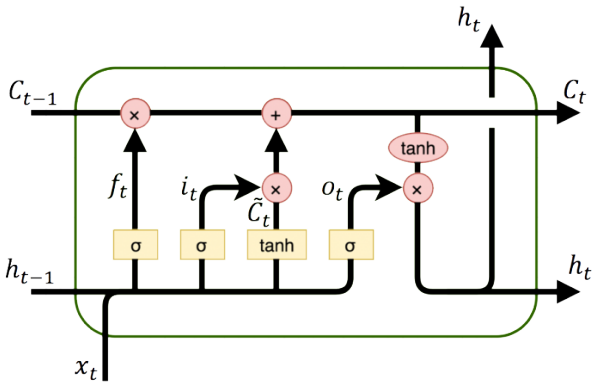


LSTM

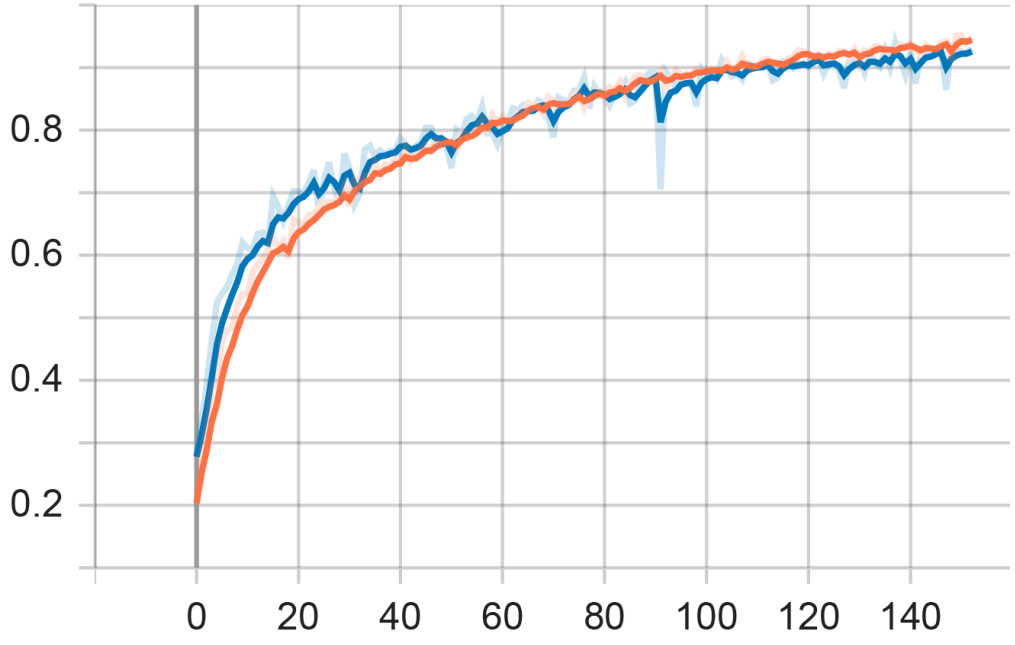
Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

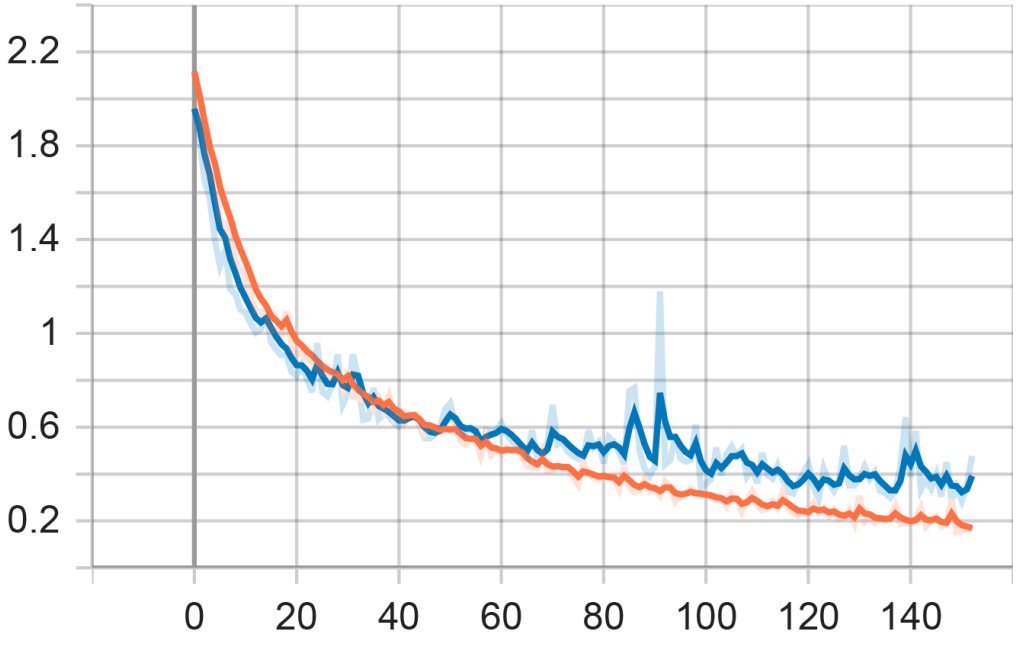
In this kind of architecture, the model passes the previous hidden state to the next step of the sequence. Therefore holding information on previous data the network has seen before and using it to make decisions. In other words, the data order is extremely important.



Training Results



Epoch categorical accuracy



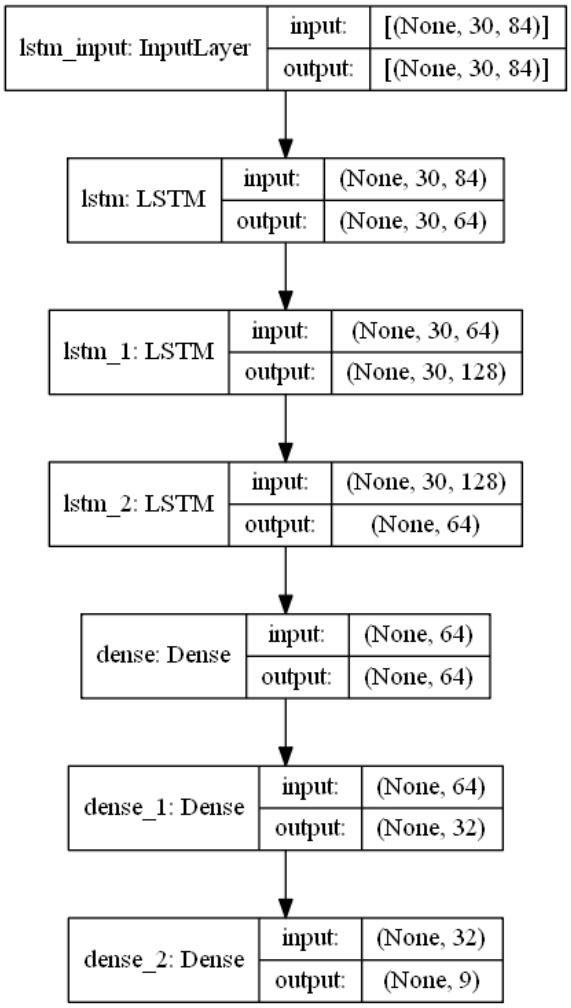
Epoch accuracy

Results

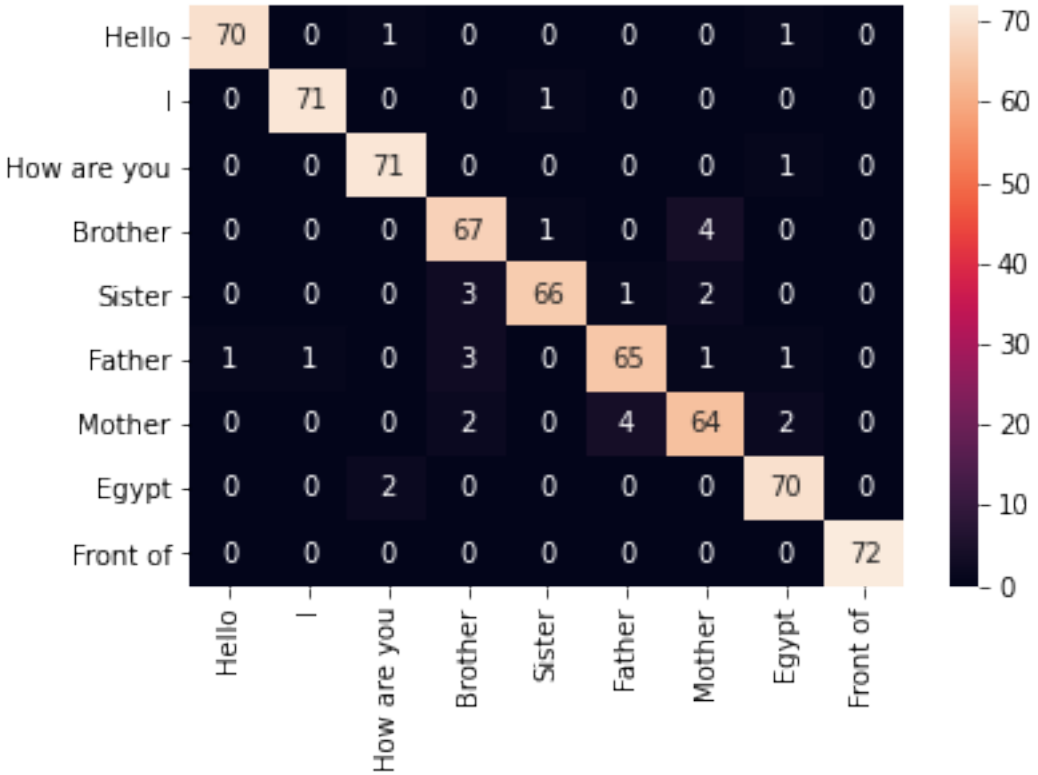
Training Accuracy = 0.9892
Training Loss=0.0272

Validation Accuracy = 0.9738
Validation Loss =0.1387

Model Architecture



Confusion Matrix



CONV1DLSTM

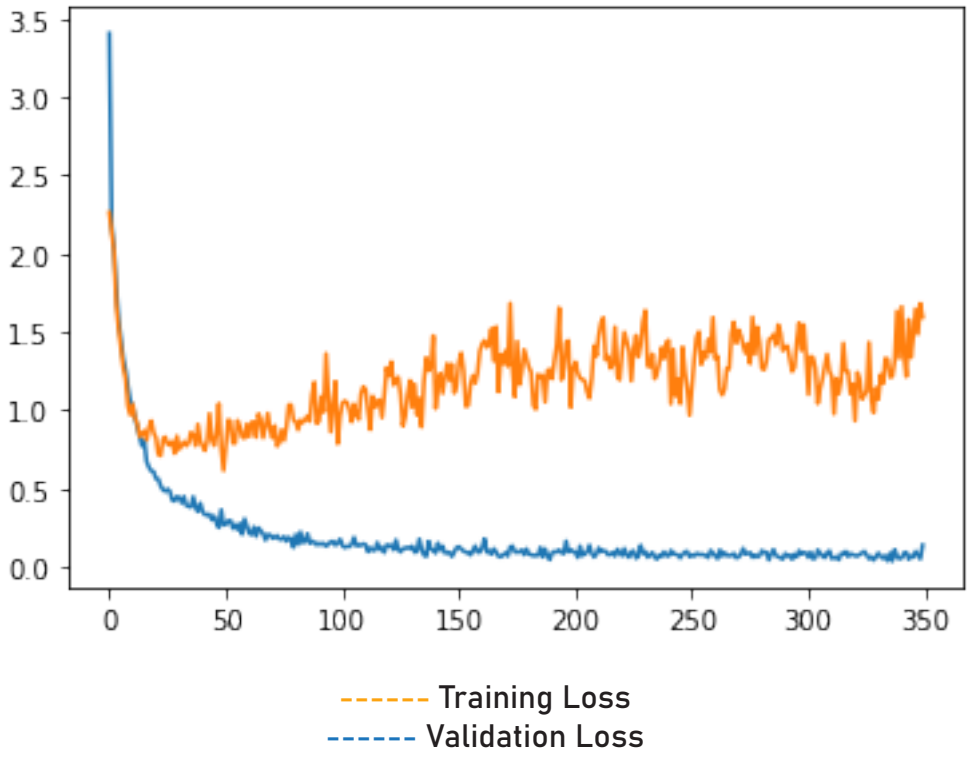
This Approach is mainly takibg best of both worlds from the first approach, In our case, sequencial images, one approach is using ConvLSTM layers. It is a Recurrent layer, just like the LSTM, but internal matrix multiplications are exchanged with convolution operations.
As a result, the data that flows through the ConvLSTM cells keeps the input dimension instead of being just a 1D vector with features.

During Implementation it ran slowly we spend time to tune the parameters but giving time constraints and good results from previous models we skipped it.

LSTM With CTC Loss

A Connectionist Temporal Classification Loss, or CTC Loss, is designed for tasks where we need alignment between sequences, but where that alignment is difficult - e.g. aligning each character to its location in an audio file. It calculates a loss between a continuous (unsegmented) time series and a target sequence. It does this by summing over the probability of possible alignments of input to target, producing a loss value which is differentiable with respect to each input node. The alignment of input to target is assumed to be “many-to-one”, which limits the length of the target sequence such that it must be <= the input length.

In our case we don’t know how fast the user will do the sign! some users are slower than others. during collecting the dataset we did each sign in one second but this won’t be the case during the use

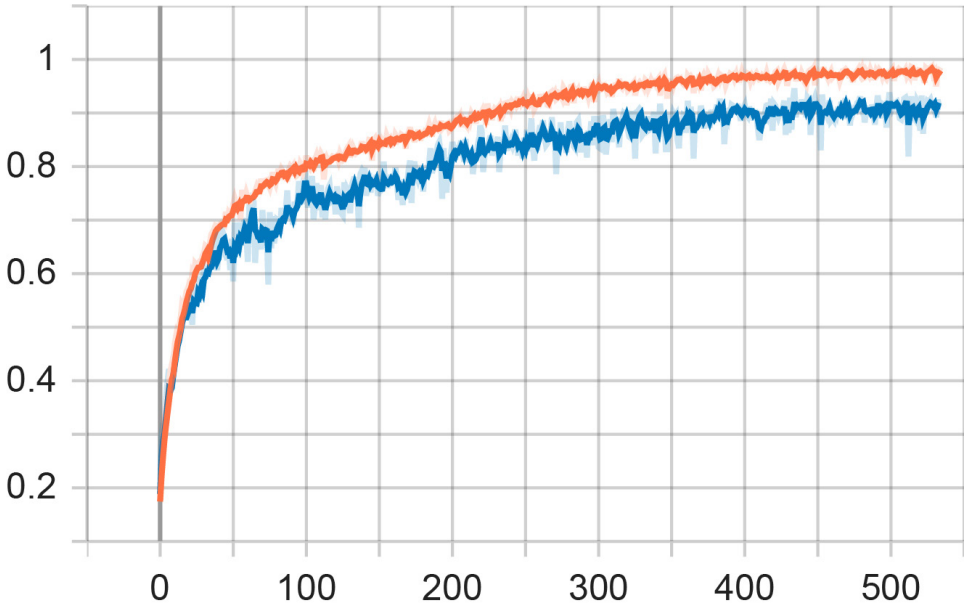


Transformer

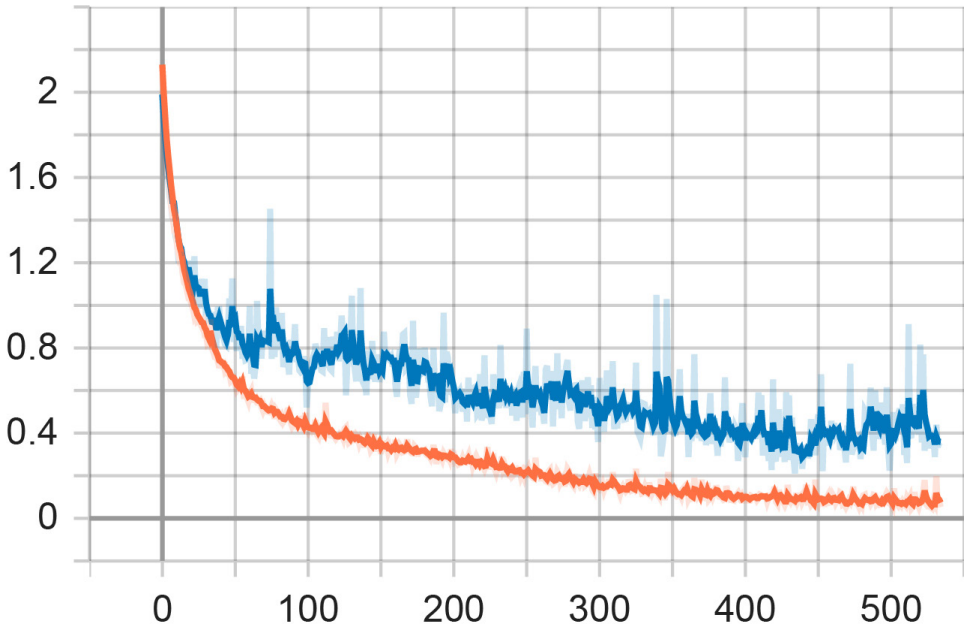
A transformer is a deep learning model that adopts the mechanism of attention, differentially weighting the significance of each part of the input data. It is used primarily in the field of natural language processing (NLP) and in computer vision (CV).

Like recurrent neural networks (RNNs), transformers are designed to handle sequential input data, such as natural language, for tasks such as translation and text summarization. However, unlike RNNs, transformers do not necessarily process the data in order. Rather, the attention mechanism provides context for any position in the input sequence. This feature allows for more parallelization than RNNs and therefore reduces training times.

Training Results

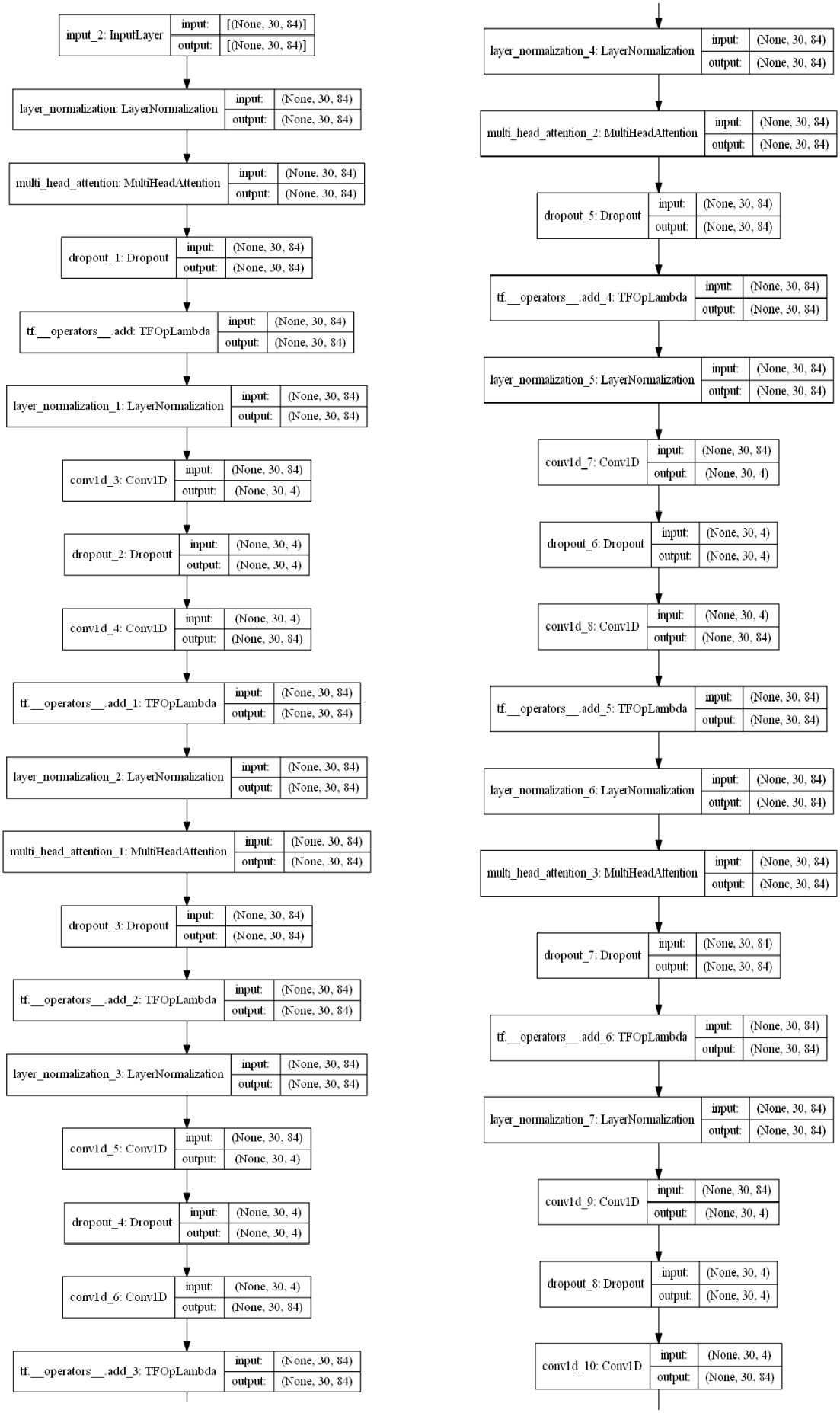


Epoch categorical accuracy

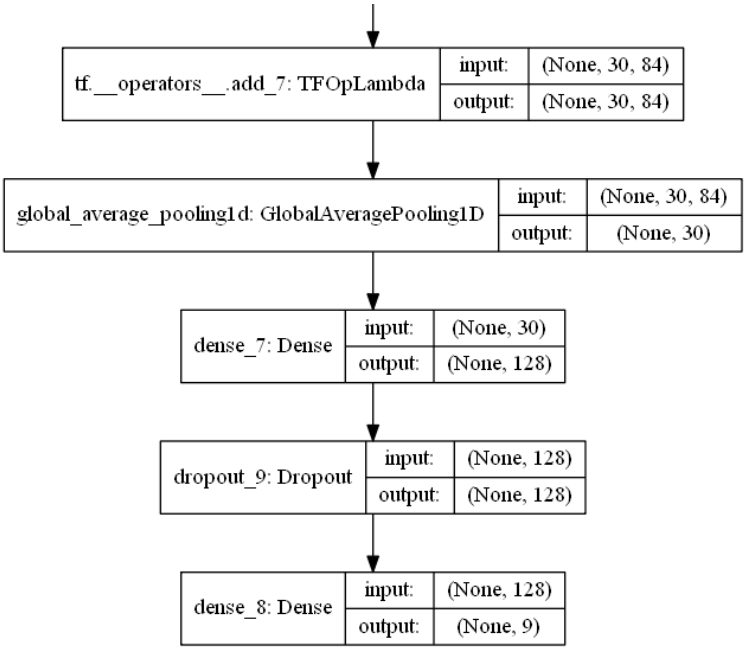


Epoch Loss

Model Architecture



Model Architecture Cont.

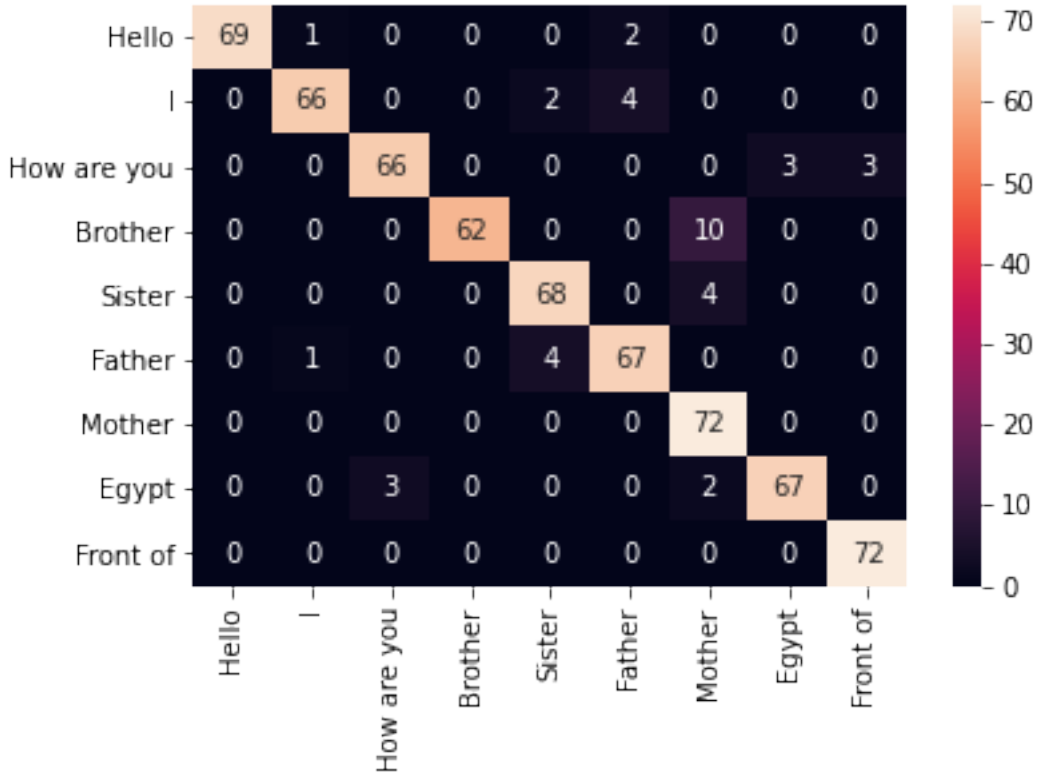


Results

Training Accuracy = 0.9865
Training Loss = 0.0432

Validation Accuracy = 0.948
Validation Loss = 0.242

Confusion Matrix



Characters

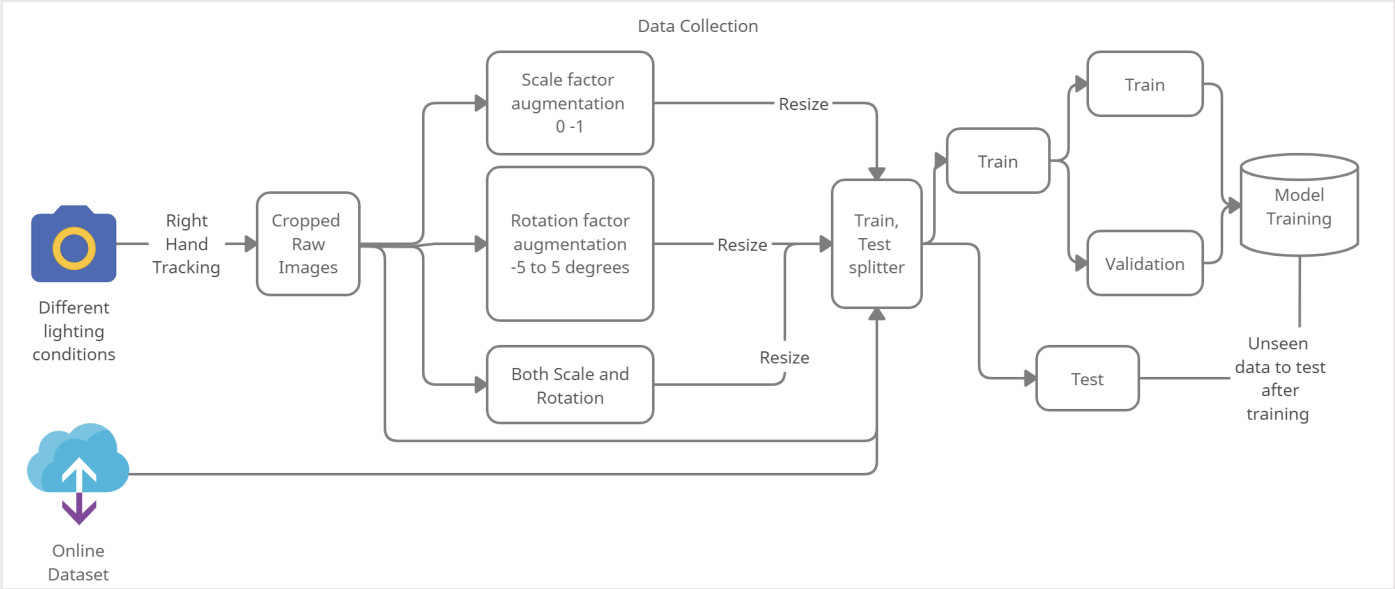
This part is simpler as the charatcers are static unlike words also we found some resources online which made it easier to get insgight of the problem

Dataset

Data Collection

We found dataset consists of 54,049 images of ArSL alphabets performed by more than 40 people for 32 standard Arabic signs and alphabets. The number of images per class differs from one class to another. The dataset gathered are of size 64 * 64 Pixels of grayscale.

We also added to it our feed by collecting extra dataset manually of 200 images for each class gathered by two persons of different hand sizes RGB but this time by using hand-tracking function that crop the hand bounding box changing the dimensions based on the sign itself.



Model

Deep convolutional neural network models may take days or even weeks to train on very large data-sets.

A way to short-cut this process is to re-use the model weights from pre-trained models that were developed for standard computer vision benchmark datasets, such as the ImageNet image recognition tasks. Top performing models can be downloaded and used directly, or integrated into a new model for your own computer vision problems. This way is called Transfer Learning

In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest.

Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error.

Models for Transfer Learning

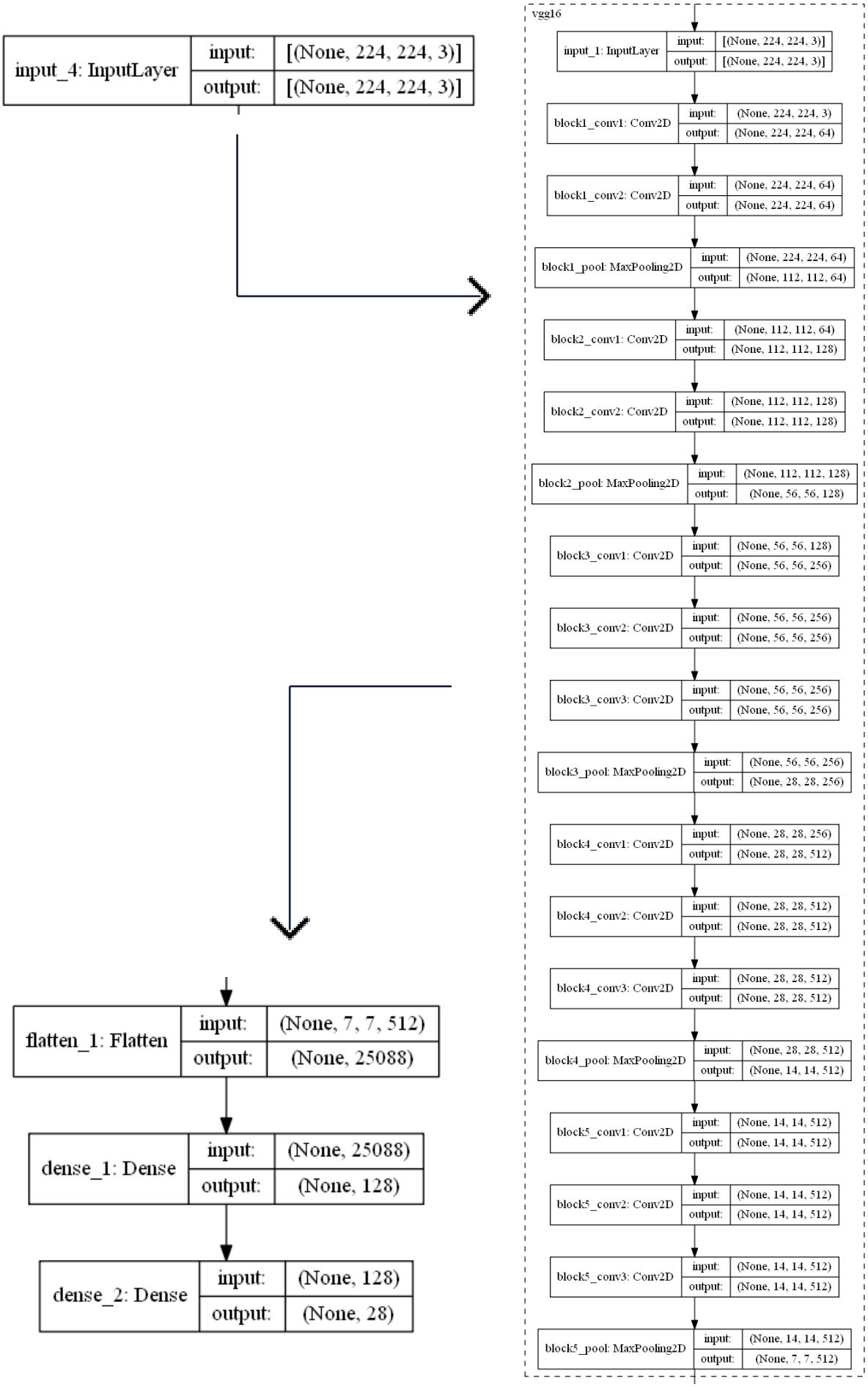
There are perhaps a dozen or more top-performing models for image recognition that can be down-loaded and used as the basis for image recognition and related computer vision tasks.

Perhaps three of the more popular models are as follows:

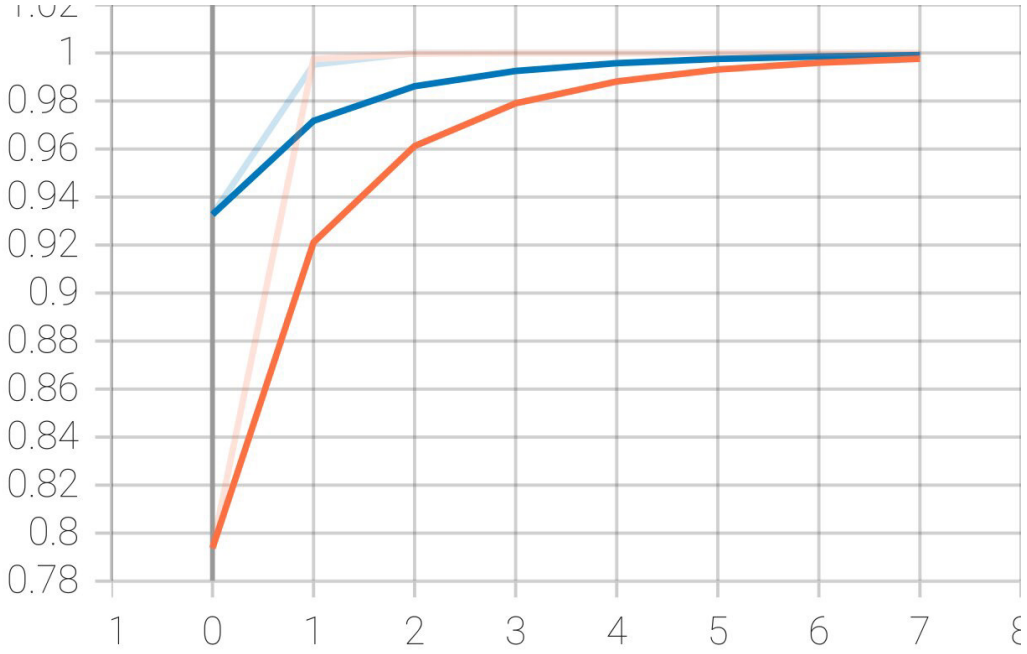
1 - VGG (e.g. VGG16 or VGG19).
2 - GoogLeNet (e.g. InceptionV3).
3 - Residual Network (e.g. ResNet50).

In our Approach we used VGG16 Model ,The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

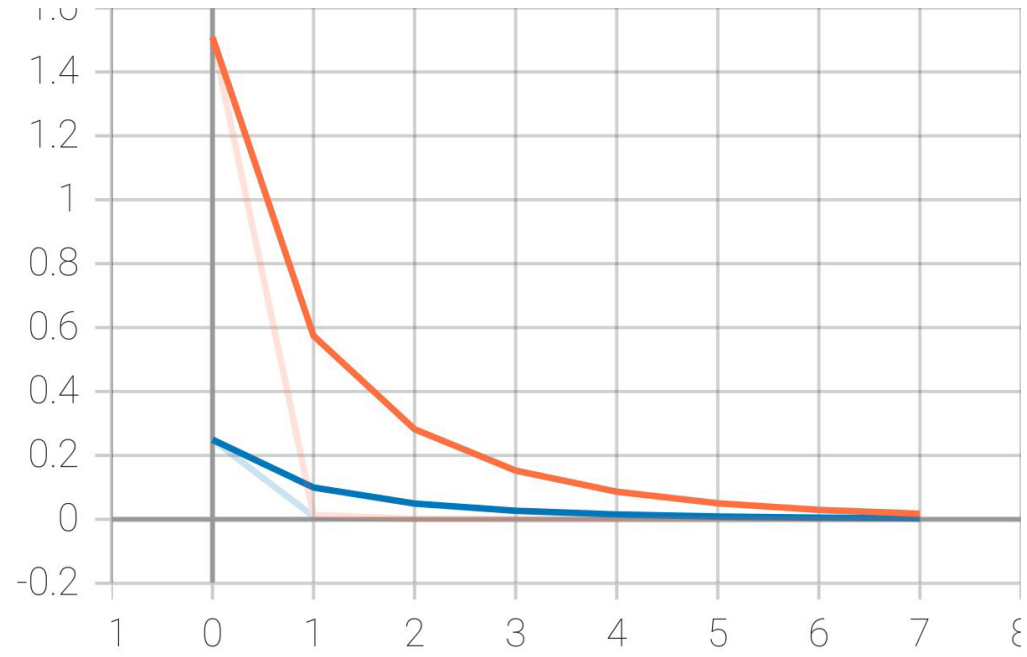
Model Architecture



Training Results



Epoch categorical accuracy



Epoch Loss

Results

Training loss: 1.0453e-07
Validation loss: 1.3472e-06

Training accuracy : 1.0000
Validation acc: 1.0000

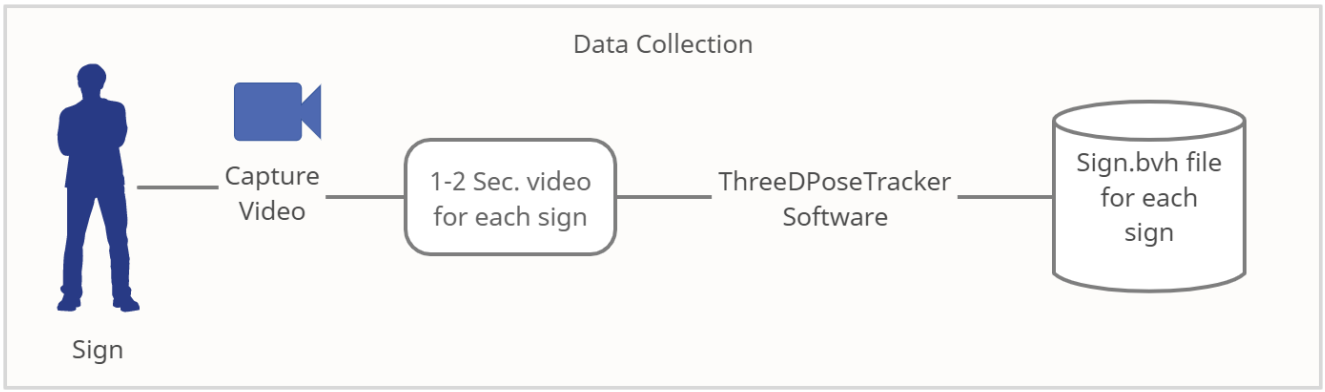
Text to Sign

A critical part of animating a sign language using virtual avatar is to display a sign gesture having multiple rotational arm poses to identify a word instead of a single static arm pose. Sequencing a group of gestures related to a sentence requires each gesture in the middle of a sentence to be animated using different initial arm positions.

Sequencing pre-captured arm videos, ordering preset animations compiled by 3D animations, and ordering motion capture data are the widely used techniques used by sign language animators presently. The transition from one word to another is not smooth as the initial and the terminating positions of each animation is not the same.

Meet SIGNARA's Avatar

Data Collection diagram



ThreeDPoseTracker

It is an application for Windows 10 that detects the 3D coordinates of the joints of the whole body from the image from the USB camera or any video.
Information on the 3D coordinates of the joints of the whole body can be processed at high speed in real time using AI technology (deep learning).

.bvh file

ASCII file that contains motion capture data for three-dimensional characters; used by 3ds Max's Character Studio and other 3D animation programs to import rotational joint data; developed by Biovision as a standard format to save biped character motion data.

After having all the required .bvh file we created an avatar in Blender to export these files it.

Blender

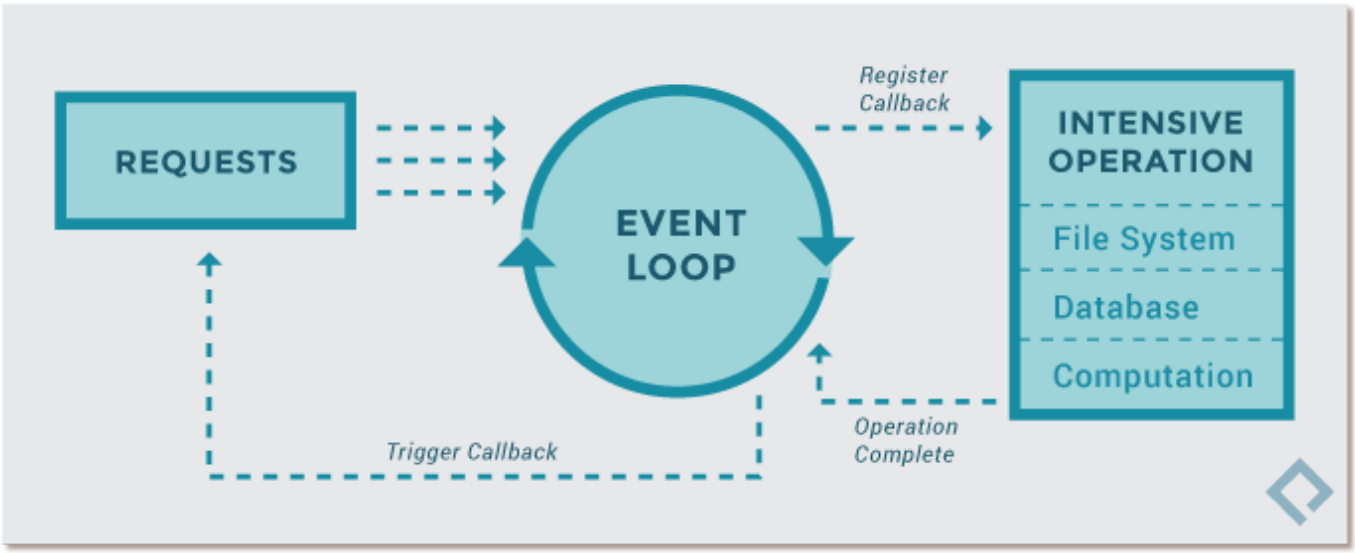
Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality, and computer games. Blender's features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, rendering, motion graphics, video editing, and compositing.

After having all the code for building the scentence we created a python script to link all the codes together and to send the data from python to Blender we used asyncio to have the advantage of accepting data all the time without freezing the program and so achieve our goal which is real-time

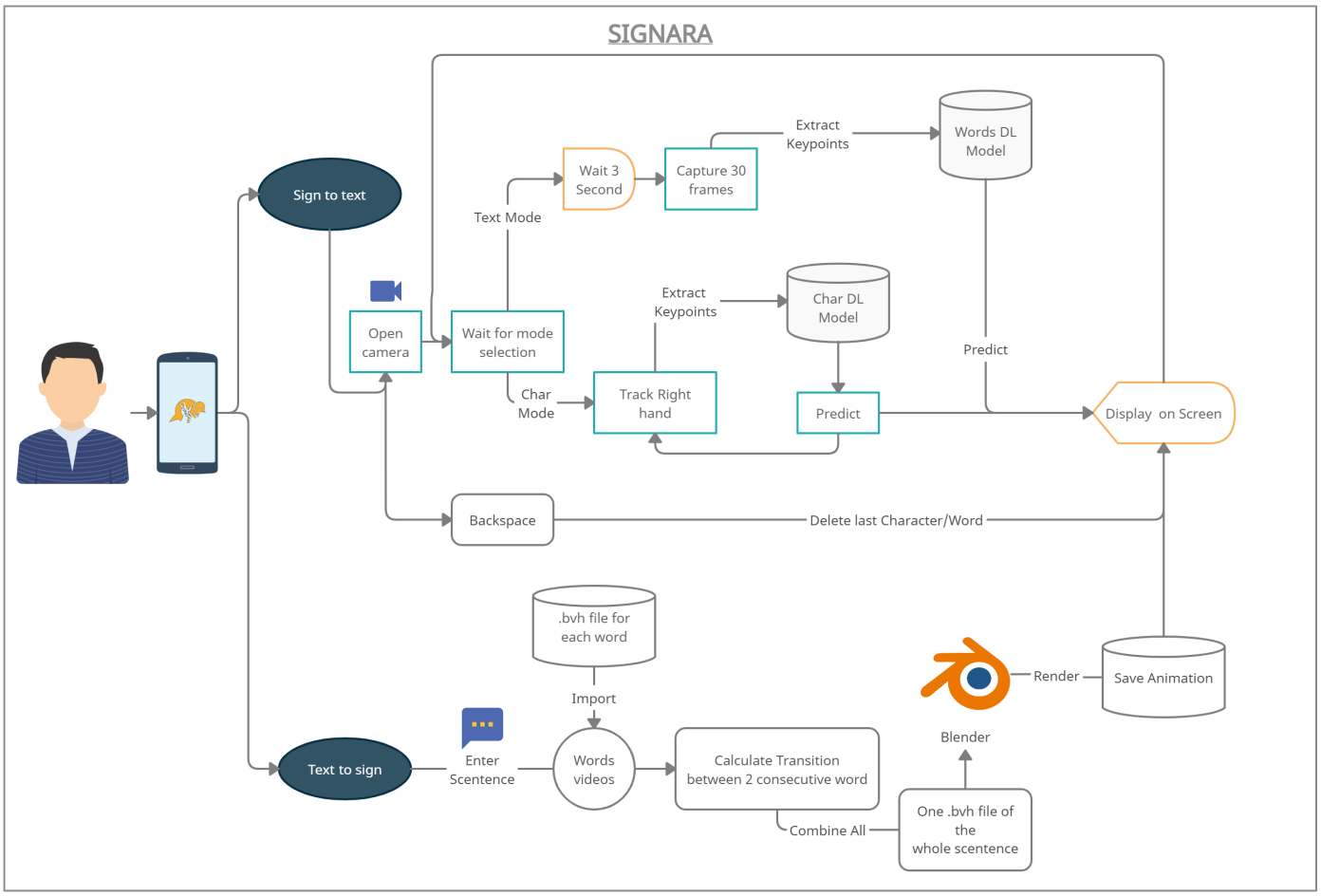
Asyncio

asyncio is a library to write concurrent code using the async/await syntax. asyncio is used as a foundation for multiple Python asynchronous frameworks that provide high-performance network and web-servers, database connection libraries, distributed task queues, etc. asyncio is often a perfect fit for IO-bound and high-level structured network code.

if you've scheduled a number of tasks to execute and you want to wait for all of them to finish, -which is our case here - use asyncio.



SIGNARA System



Use Case

During this project, our main focus was to create a proof of concept for our proposed solution. We managed to solve all the problems we encountered during our implementation. To Develop a real-time computationally inexpensive system for the edge devices for everyday use so that our main target users be able to interact in daily life without needing extra accessories only by using their mobile phone camera.

Even though we solved all the problems we faced but during the usage of the system, Every day new ideas came to our minds to make a more effective and user-friendly system. But, due to time constraints, we documented our ideas as future improvements for our system.

Challenges and solutions

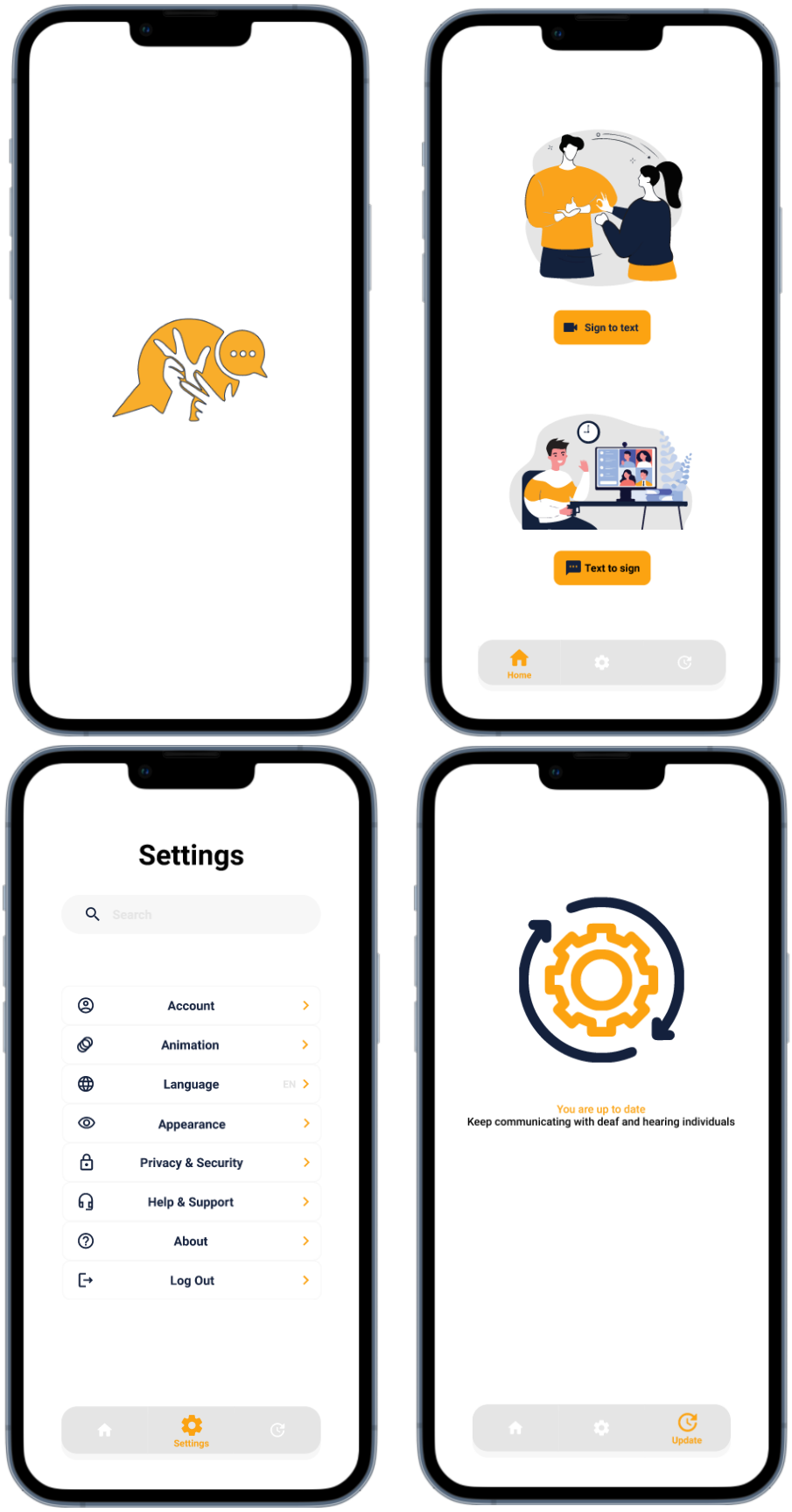
- Wrong predicted word : we invented virtual Backspace button in the screen to delete the last prediction whether it's a character or word.
- Frame overlapping: we invented a virtual button for words mode and after it we started a countdown so that the clicking action is not captured as a sequence.

Positive Impact on Sustainability

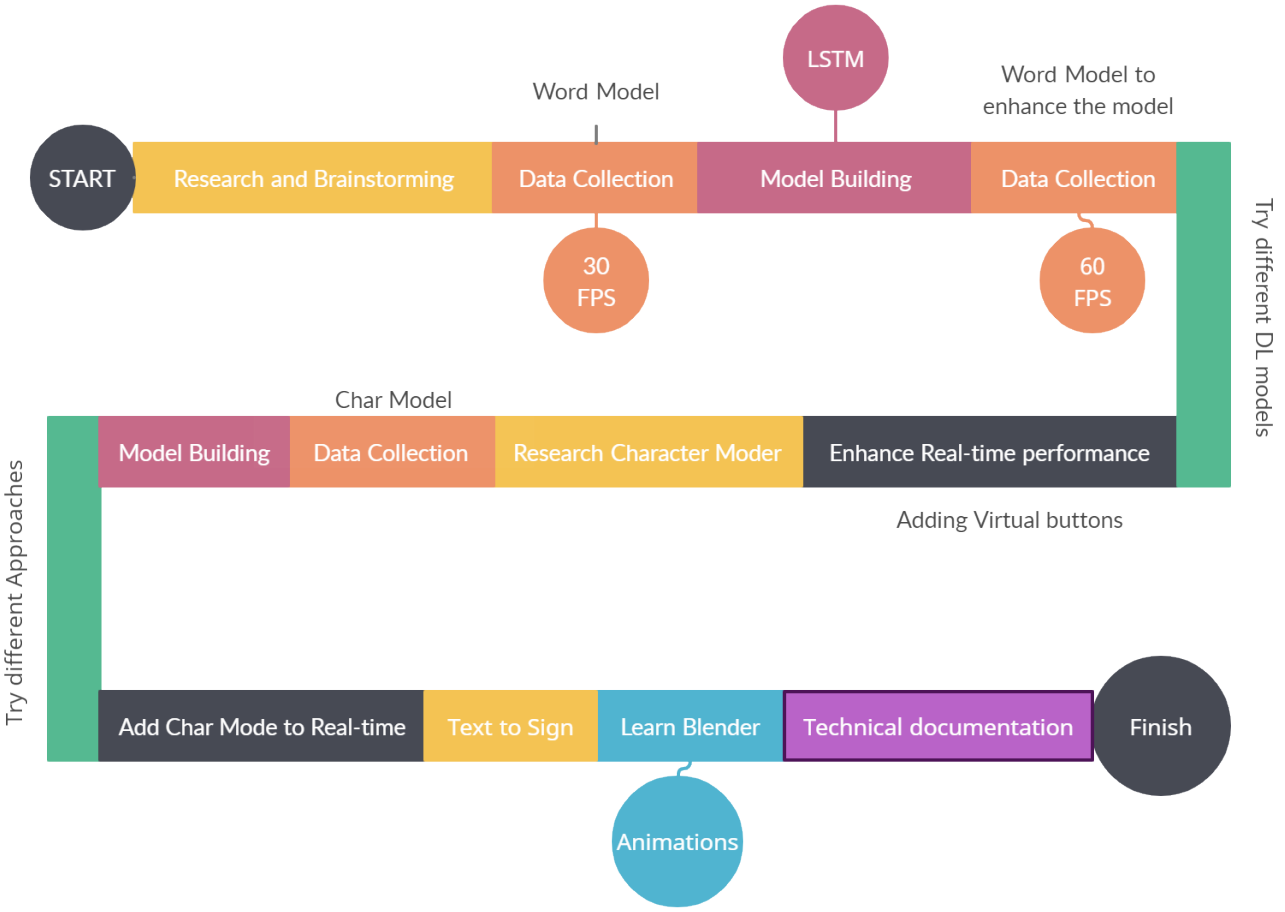
The UN 2030 GOALS for Sustainable Development emphasizes the importance of Disability-Inclusion, to maintain sustainability across the world, we have to strive to give everyone equal opportunities, and this means focusing on being Disability-Inclusive, and as engineers we strive to design solutions and devices to help facilitate the integration of Persons with disabilities into society.

Emphasizing the need for a comprehensive, sustainable and multisectoral approach to improving access to assistive technology that fulfils the safety and quality standards established by national and international regulations, at the national and subnational levels

Mobile Application UI



Workflow



Key Results and Summary

Real-time translation of sign language is a computationally difficult task that may not be possible on most consumer-grade hardware. However, libraries such as MediaPipe are making mobile real-time sign language translation possible .

Real-time translation of sign language is a computationally difficult task that may not be possible on most consumer-grade hardware. However, new libraries such as MediaPipe are starting to make mobile real-time sign language translation possible.

The collection, management, and processing of training data is a task which cannot feasibly be done manually, and should be streamlined

Words are “time-series” signs that requires the use of LSTM and complex data management infra-structure unlike Characters which are static image.

Future Improvements

- The most obvious improvement for our system is to increase both numbers of classes and dataset and it's a time-consuming task that requires many data collectors under the supervision of sign language experts.

- Use Tflite to convert our models to be used on-device ML solutions for mobile and edge use cases.

- Meet our target users from the hearing impaired and mute community to understand their needs and get an insight into what kind of problems may face them while using our system.

- Work on the performance of the animation.
- Implement the sound feature that also may accept a sound input and then change it into sign language and vice versa may be a very good solution to facilitate the usage of the system

References

https://www.who.int/health-topics/hearing-loss
https://apps.who.int/gb/ebwha/pdf_files/WHA71/A71_R8-en.pdf?ua=1
https://www.disabilityexpertsfl.com/blog/difficulties-the-deaf-face-every-day

https://www.verywellhealth.com/what-challenges-still-exist-for-the-deaf-community-4153447

https://www.who.int/pbd/deafness/WHO_GE_HL.pdf
https://mediapipe.dev/
https://google.github.io/mediapipe/solutions/hands
https://link.springer.com/content/pdf/10.1007%2F978-3-319-11656-3_25.pdf

https://medium.com/towards-data-science/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc

https://www.tensorflow.org/api_docs/python/tf/all_symbols

https://www.researchgate.net/publication/224197735_ArSLAT_Arabic_Sign_Language_Alphabets_Translator#pf2

https://data.mendeley.com/datasets/y7pckrw6z2/1
https://github.com/digital-standard/ThreeDPoseTracker

https://www.researchgate.net/publication/281434972_3D_Animation_framework_for_sign_language