Section C Team 50

Samir Kaddoura(sk880), Ivy Liu(yl693), Guntae Park(gp150), Ranjeet Pawar(rdp44), Chhavi Sharma(cs764)

American Sign Language translator

1. Business understanding :

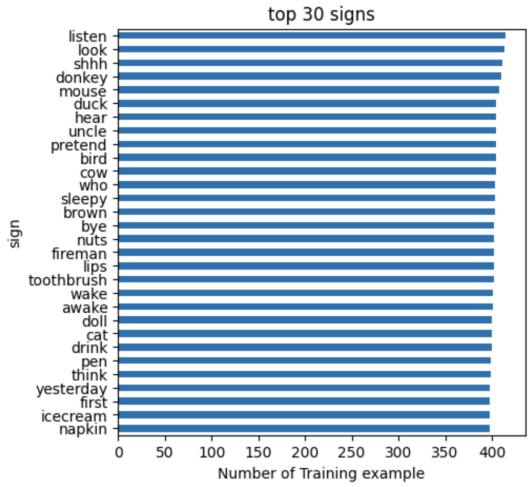
American Sign Language (ASL) is a visual language. With signing, the brain processes linguistic information through the eyes. The shape, placement, and movement of the hands, as well as facial expressions and body movements, all play important parts in conveying information. It is the primary language of many North Americans who are deaf and hard of hearing, and is used by many hearing people as well. Through this project, individuals with physical disabilities can have simple conversations with those who do not know sign language. We believe that, in a business context, it is necessary to provide a tool for automated sign language translation as it will allow a more mainstream inclusion of the deaf and hard of hearing communities. Though there may already exist resources aimed at inclusion, we aim to go beyond and close the gap between deaf/mute employees and those who aren't, so as to have their participation in the workforce appear as seamless as possible. Furthermore, we believe that such a tool would be attractive to companies as they could potentially reduce costs by avoiding hiring translators, with translation now automatic.

2. Data Understanding :

The dataset we are using contains 250 different signs from the sign language, with each sign having between 299 and 415 video examples, so as to account for variety in motion. Our dataset also has a participant_id column to identify the different people recording the videos. We believe that, although this will not take direct part in the model, it is crucial to have a variety of different participants record videos,

so as to have the model be flexible and account for different physical attributes such as skin tone, facial hair, face shapes etc…

Figure 1: top 30 ASL signs represented in the data



To better understand the data we are dealing with, this graph shows the frequency of the top 10 most frequent signs seen in the dataset. As we see, they mostly have the same number of examples. We notice that there are a variety of different kinds of words in the top 30. We will be able to train our model on verbs such as "listen", "look" and "hear", nouns such as "donkey", "mouse", "duck", adjectives such as "sleepy", "brown" and even onomatopoeia such as "shhh". As such, we believe our dataset, although limited in its current size, presents a good variety of types of words.

3. Data Preparation :

Our data therefore consists of two components we will make use of: the video files themselves, capturing the motions of face, hands and body (represented by the shoulders) as well as the label of the associated sign. We seek to convert the video files into numeric data so as to use it to train our deep learning model. To do so, we use the mediapipe package. This package allows us to extract "landmarks" from a video in numerical form. As such, we can "vectorize" the frames of a video and define the videos as such, which

will allow us to input them into our deep learning model. We extract landmarks from three body parts as defined by the mediapipe package: the pose (shoulders), left hand, and right hand. Each landmark identified by the package is defined with four values: the x-coordinate, the y-coordinate, the z-coordinate and a visibility value. We would then flatten those values and attribute a single vector of values to each video file. Due to file size constraints, we decided to restrict ourselves to a select few labels : hello, bye, listen, look, shhh, donkey, mouse, duck, hear, and uncle.

4. Model :

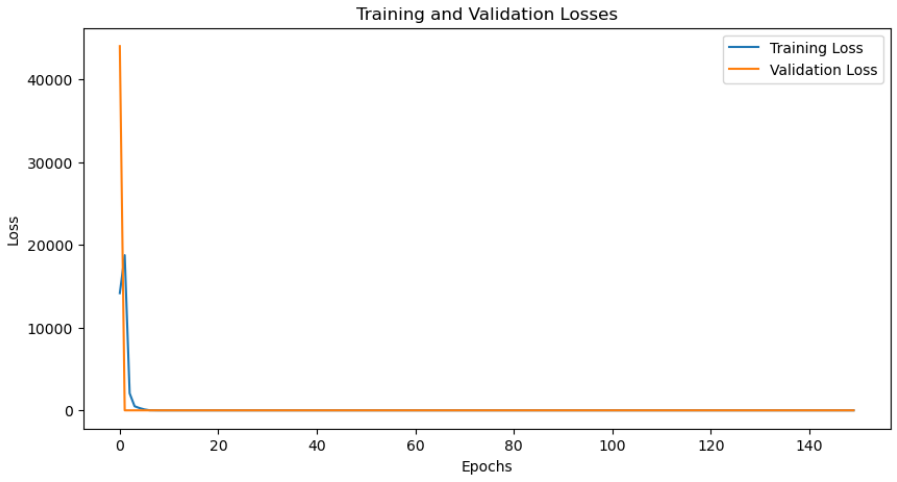
Firstly, to test our models, we have decided to split the data into a training and validation set, with the training set taking 95% of the entire data and the validation set the remaining 5%. For all the models we have tried, we used the Adam optimizer, as it is commonly recognized as the state of the art optimizer. For all models, as we return probabilities, we use a categorical cross-entropy loss to evaluate our models. In order to return the corresponding probabilities for each label, we first attempt a long short-term memory neural network with the following hyperparameters defined: a size 64 LSTM layer with ReLu activation, a size 128 LSTM layer with ReLu activation, and a size 64 LSTM layer with ReLu activation. Next, we have two regular layers, both size 64 and size 32, both with ReLu activation. Finally, we have a layer the size of the number of layers that passes a softmax activation. We train this model over 80 epochs and obtain the following training and testing losses:

Figure 2:

This model has a corresponding accuracy of 0.7. We chose 80 epochs as a result of an early stoppage, as we noticed that any further epochs simply overfit the model and led to a decrease in accuracy and increase in validation loss.

We then attempted another LSTM model with the same layers, but, to avoid overfitting, added three dropout layers in between the LSTM layers, each with a dropout rate of 0.2. Additionally, we defined a learning rate of 0.01 and chose to run this new model over 150 epochs. The following training and validation losses were produced:

Figure 3:

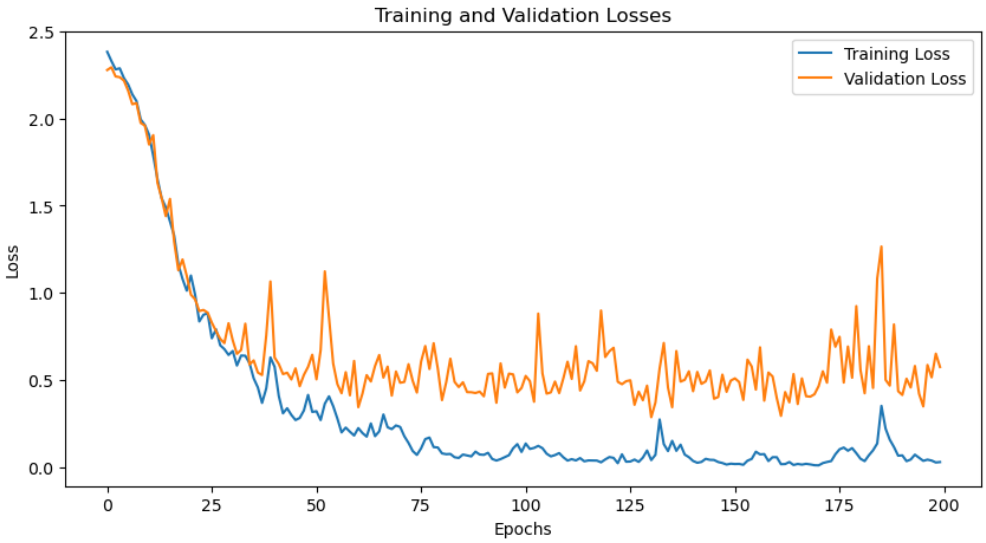


As we can see, both errors drastically drop very quickly, potentially the result of the learning rate. This model performed much worse than the previous LSTM model, with a corresponding accuracy of about 0.067.

Next, we decided to attempt a convolution neural network. This model is built using a size 64 convolution layer, with a filter size of 3 by 3 and a ReLu activation, followed by a max-pooling layer of size 2 and a dropout layer with dropout rate of 0.25. We then add the same structure with a size 128 convolution layer instead, everything else constant. We flatten the layer and transition to a regular size 128 ReLu layer followed by a size 64 ReLu layer and finally run the weights through a softmax activation. We pass a

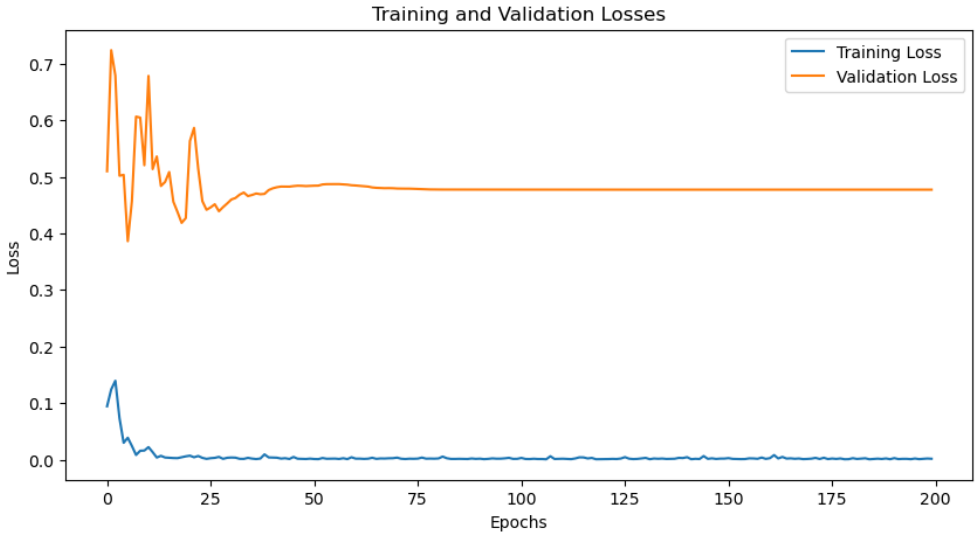
learning rate of 0.001 to the Adam optimizer and run 200 epochs to obtain the following training and validation losses:

Figure 4:



This model resulted in an accuracy level of 0.86, much better than the previous ones. Seeing success in this approach, we decided to build on it. We first attempted to add L2 regularization with a weight decay of 0.01, keeping all other hyperparameters constant, except for the number of epochs, which we increased to 300. Given that however, the model performed significantly worse, with an accuracy level of 0.73. The last modification we attempted was a learning rate scheduler. The scheduler works as follows as defined by our code: we initialize a learning rate of 0.001 for the first ten epochs and then exponentially decrease the learning rate every epoch following the tenth one by a factor of exp(-0.1) as this would allow us to avoid overfitting in the long run by greatly reducing the learning rate, while still having the model quickly train in the short run. We ran this model over 200 epochs and obtained the following training and validation loss:

Figure 5:



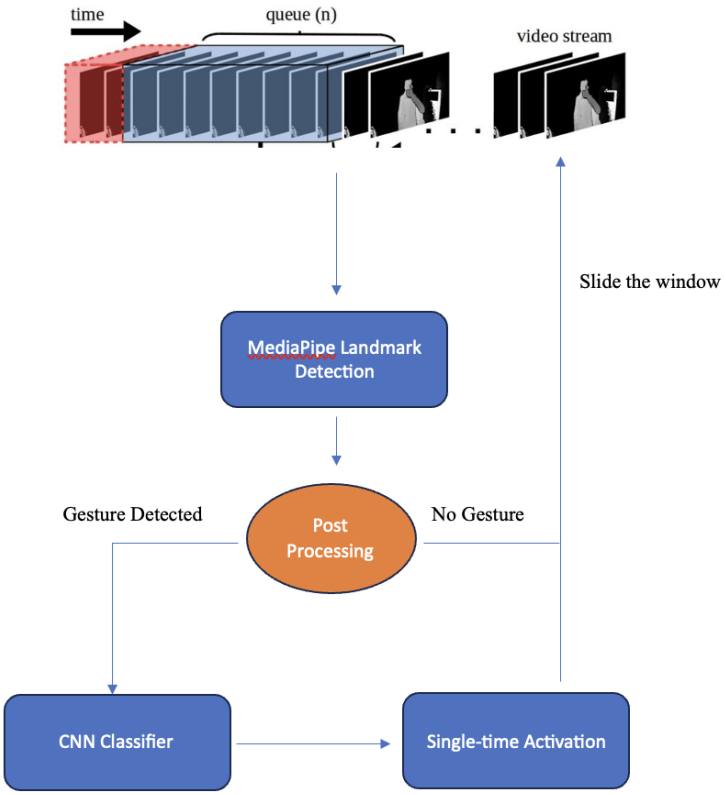
This model is the best of all the models we have attempted, with a corresponding level of accuracy of 0.8999. We'd finally attempted a transformer model but that one however underperformed relative to the CNN network with learning rate scheduler.

5. Implementation/Methodology : We elaborate on our model hierarchical architecture that enables the-state-of-the-art CNN model to be used in real-time gesture recognition applications as efficiently as possible. While the state-of-the-art model achieved an accuracy of 0.8929, our model outperformed this with an accuracy of 0.8999, despite being trained on a comparatively limited number of videos. The state-of-art model uses 1D CNN combined with transformer. Notably, our exploration didn't encompass the fusion of CNN with transformer models, an avenue of inquiry that remains untapped and holds potential for further advancements in this project.

In our model, handling variable-length input correctly was very crucial for ensuring train-test consistency and efficient inference , as we do not necessarily have to pad the short videos. During training, we used a

sequence_length=30, this approach provided sufficient inference speed and allowed the use of reasonably large models.

Architecture :



6. Deployment :

This innovative technology is poised to make a transformative impact by addressing communication barriers between the deaf and hearing communities. Below are key areas where the deployment of the ASL Translator model is set to be exceptionally beneficial:

a. Education: Classroom Integration; The ASL Translator can be integrated into educational settings, facilitating communication between deaf students, teachers, and hearing peers. It opens up avenues

for a more inclusive learning environment, enabling seamless participation and comprehension of course materials.

b. Employment: Workplace Communication; In professional environments, the ASL Translator can enhance communication between deaf and hearing colleagues. This can lead to increased workplace integration, improved collaboration, and a more inclusive corporate culture.
c. Healthcare: Doctor-Patient Interactions; The ASL Translator can be employed in healthcare settings to bridge communication gaps between deaf patients and medical professionals. This ensures that crucial information about health conditions, treatment plans, and post-treatment care is accurately conveyed.
d. Public Spaces: Public Announcements; Integration of the ASL Translator into public spaces, such as transportation hubs, government offices, and entertainment venues, ensures that individuals with hearing impairments receive timely and relevant information. This includes announcements, emergency alerts, and general public messages.
e. Social Interactions: The ASL Translator can be incorporated into social media platforms, allowing deaf individuals to participate in online conversations seamlessly. It opens up new avenues for expression and connection, breaking down digital communication barriers.
f. Emergency Services: During emergency situations, effective communication is paramount. The ASL Translator can be utilized to enhance communication with emergency services, ensuring that individuals with hearing impairments receive timely and accurate assistance.
g. Entertainment: Subtitles have traditionally been the primary means of making media accessible to the deaf community. The ASL Translator introduces a dynamic and expressive alternative, allowing individuals to enjoy content in their preferred mode of communication.

h. Accessibility Apps: Integration into mobile applications allows for on-the-go translation, providing real-time assistance for users navigating various situations, e.g., stores, public transportation, and other daily activities.

7. Potential risks:

Like all AI models, this one is prone to mistakes. Firstly, we should consider that, although our accuracy is high, we are unsure if we will be able to replicate this high level of precision at a large scale dataset, in a way to account for the ASL as a whole. Even then, assuming that our model persists in its efficiency at the large scale, it is worth noting that our model incorrectly translates the ASL around 10% of the time. This is a potential risk as the accurate and precise convection of information is key in a business setting, and an error such as this, which we can only reduce but not fully eliminate, may cause very detrimental issues to the function of a firm with deaf and hard of hearing employees. To counter such issues, we advise firstly continuously fine-tuning and improving the model, so as to provide more accurate translations. We also suggest explaining the functioning of the technology at hands to its users, so as to have them understand and be patient of potential translation issues, and be more wary of it. Overall however, as a translation tool, there are not many more ways to avoid the issues other than simply improving the model.

8. Appendix:

Contributions:

Chhavi: Performed Exploratory Data Analysis. Worked on Data preparation, Model development and Deployment.
Samir: Worked on Model development and writeup.
Ivy: Worked on the business understanding and data understanding write-up.
Ranjeet: Worked on the data preparation. Contributed to model deployment and write-up.
Guntae: Worked on the model evaluation and presentation.

References:

https://www.kaggle.com/competitions/asl-signs/overview

Paper- “Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks” by OkanKop¨ ukl¨u1, AhmetGunduz1, NeslihanKose2, GerhardRigoll1

https://github.com/anuragk240/Speech-to-Sign-Language-Translator

Used ChatGPT for hyperparameter tuning of Transformer model.