MODEL IMPLEMENTATION

6.1 Method I: MediaPipe 3D Landmark

The first approach uses MediaPipe (Mediapipe, n.d.) pose and hands landmark coordinates. A total of 267 landmarks were extracted from each video frames which include 33*3 (x, y, z coordinates) from pose, 21*3 (x, y, z coordinates) from each hand and lastly 21 values representing the distances between the thumb to each fingertip, wrist to each fingertip, between each fingertip, within finger joints and between each fingertip to thumb joint. Before feeding the data into the model, the coordinates of each landmark were normalized to the frame size whereas the distance values were normalized to the shoulder width of the signer.

6.1.1 Model 1 Conv-LSTM

As shown in figure 16, the first model was built using the combination of Convolution and LSTM layers as the model's backbone. Specifically, the model was constructed using a 1D-Convolution layer with 32 filters, kernel size 3, stride of 1 and with padding, followed by two LSTM layers with 64 and 128 units respectively. In Convolution layer, 'relu' activation function was used. On the other hand, in LSTM layers, 'Tanh' activation function and 'sigmoid' recurrent activation function were used. Both activations help to control the flow of information within the cell and introduce non-linearities. Specifically, Tanh function squashes the values to -1 and 1. On the other hand, sigmoid transforms the values to range between 0 and 1. Both function helps the model to decide which information to forget and how much information to add to the cell state. Next, a self-attention layer with 'sigmoid' activation function was added to ensure the most relevant information was focused before passing the output to the fully connected layer with 15 units and softmax activation function. Finally, the softmax activation function transforms the output of the neural network into a probability distribution over all classes. The highest probability generated corresponds to the predicted class. Several measures were taken to avoid overfitting which include applying dropout layers and batch normalization as well as adding kernel L2 regularizer to the self-attention layer.

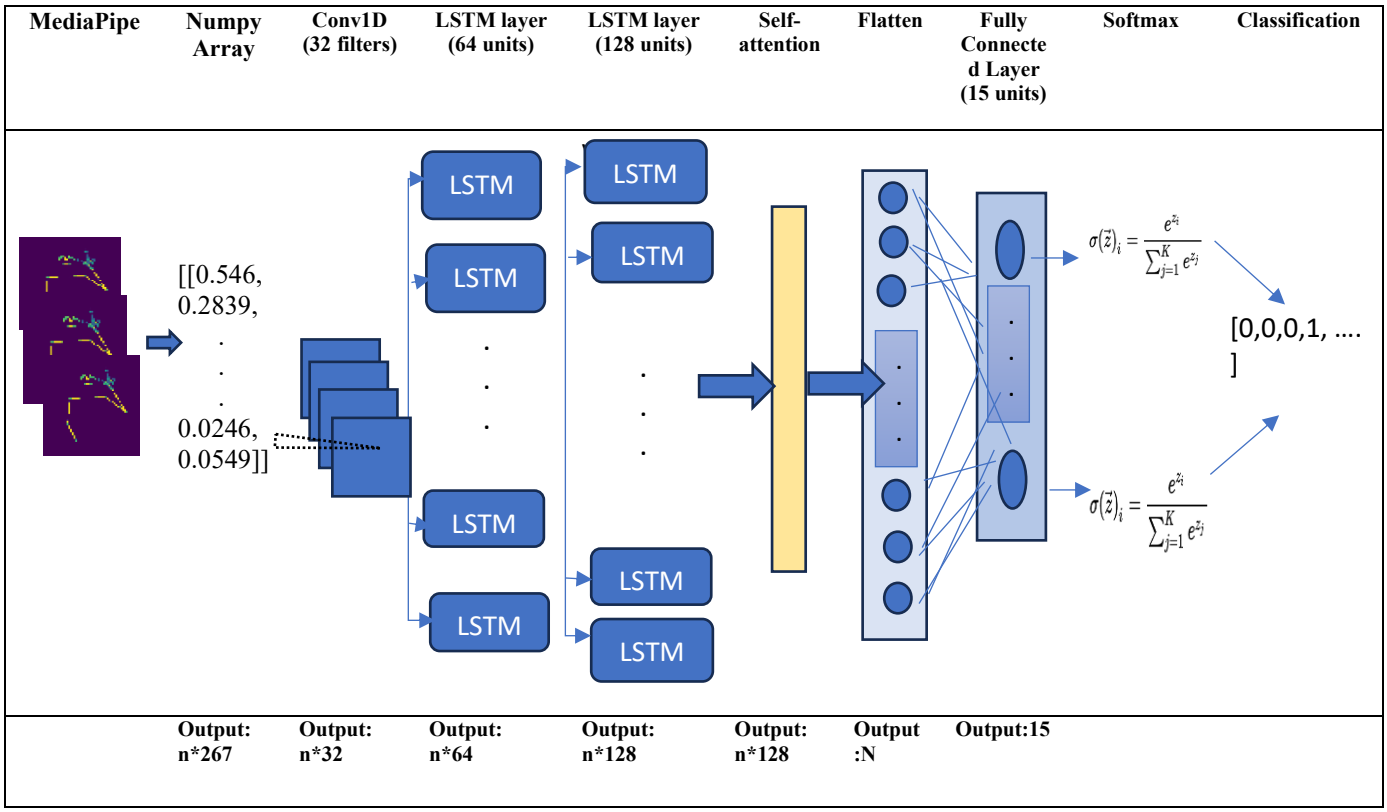

Figure 1 Model 1 Landmark-LSTM architecture

The same network architecture was trained and validated on LSA64 and WLASL datasets. As the maximum of video length differs in both datasets, the input dimension for LSA64 dataset was 190*267 while WSASL dataset was 145*267. As a result, the total number of trainable parameters of LSA64 was slightly higher, with 28,800 parameters more. Upon application of the Conv layer, the number of features were transformed from 267 to 32 reflecting the number of filters being applied. Followed by a max pooling layer which downsamples the input temporal dimensions by selecting the maximum value when an input window strides along each dimension. As 'same' padding option was used, the resulting output shape is determined by ((input shape – pool size) / strides) +1 (MaxPooling2D layer, n.d.). In convolution layer, the number of parameters is determined by multiplying kernel size with the number of features and adding one bias for every filter. Hence, each will have 3*267+1=802 params and 32 filters have 25,664 params.

As each LSTM cell consists of input gate, forget gate, cell state update and output gate, therefore it is computationally intensive compared to feedforward neural networks. The

number of parameters for a function within the LSTM unit is determined by the number of features, number of units in LSTM layer and the bias parameter, 1. Since LSTM has 4 functions, the total number of parameters is four times higher. For example, for the first LSTM layer on LSA64 dataset, the total number of parameters would be 4*(32(number of features) + 64 (LSTM units)+1(bias)*64(LSTM units)=24832 parameters.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 190, 32)           25664

 max_pooling1d (MaxPooling1  (None, 63, 32)            0
 D)

 dropout (Dropout)           (None, 63, 32)            0

 lstm (LSTM)                 (None, 63, 64)            24832

 dropout_1 (Dropout)         (None, 63, 64)            0

 batch_normalization (Batch  (None, 63, 64)            256
 Normalization)

 lstm_1 (LSTM)               (None, 63, 128)           98816

 dropout_2 (Dropout)         (None, 63, 128)           0

 batch_normalization_1 (Bat  (None, 63, 128)           512
 chNormalization)

 seq_self_attention (SeqSel  (None, 63, 128)           8257
 fAttention)

 flatten (Flatten)           (None, 8064)              0

 dense (Dense)               (None, 15)                120975

=================================================================
Total params: 279312 (1.07 MB)
Trainable params: 278928 (1.06 MB)
Non-trainable params: 384 (1.50 KB)
```

Figure 2 Mediapipe-model 1 summary on LSA64 dataset.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 145, 32)           25664

 max_pooling1d (MaxPooling1  (None, 48, 32)            0
 D)

 dropout (Dropout)           (None, 48, 32)            0

 lstm (LSTM)                 (None, 48, 64)            24832

 dropout_1 (Dropout)         (None, 48, 64)            0

 batch_normalization (Batch  (None, 48, 64)            256
 Normalization)

 lstm_1 (LSTM)               (None, 48, 128)           98816

 dropout_2 (Dropout)         (None, 48, 128)           0

 batch_normalization_1 (Bat  (None, 48, 128)           512
 chNormalization)

 seq_self_attention (SeqSel  (None, 48, 128)           8257
 fAttention)

 flatten (Flatten)           (None, 6144)              0

 dense (Dense)               (None, 15)                92175

=================================================================
Total params: 250512 (978.56 KB)
Trainable params: 250128 (977.06 KB)
Non-trainable params: 384 (1.50 KB)
_________________________________________________________________
```

Figure 3 MediaPipe model 1 summary on WLASL dataset

6.1.1.1 Performance

LSA64 dataset

Model fitness

The model was trained for 200 epochs. From the accuracy and loss graph, the model performance showed a good fit. Both training and validation accuracy as well as loss were improving in the first 25 epochs, reaching 95% accuracy and flatten until the end of training. At epoch 200, the training accuracy was 99.86% with 0.0186 loss whereas validation accuracy was 99.58% with 0.0319 loss.

```
Epoch 200/200
136/136 [==============================] - 43s 315ms/step - loss: 0.0186 - accuracy: 0.9986 - val_loss: 0.0319 - val_accurac
y: 0.9958
```

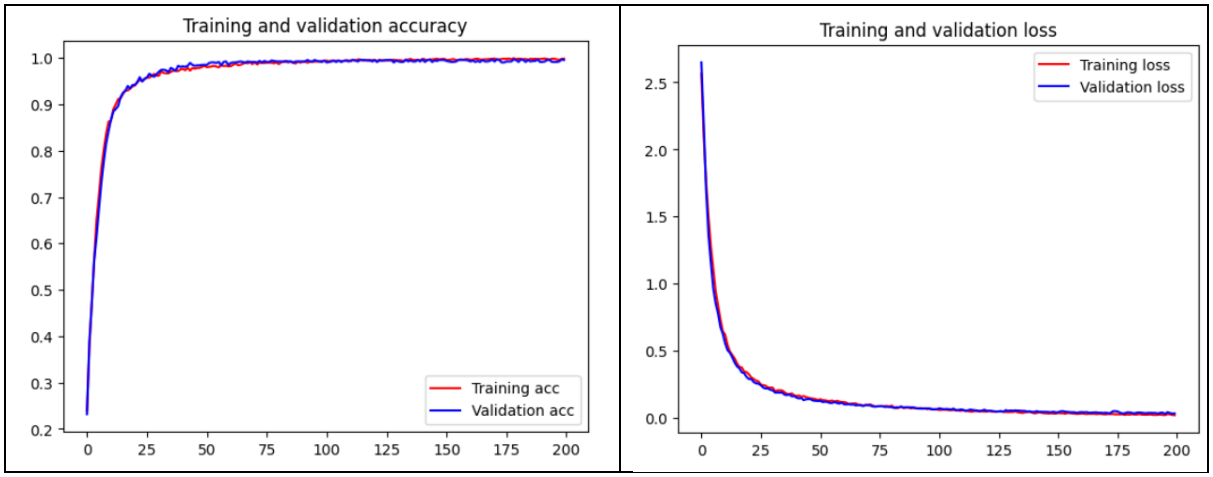

Figure 4 Training, Validation Accuracy and Loss across 200 epochs.

Test Set

The model achieved 98% accuracy and 0.0799 loss on test set. Based on the confusion matrix, the model failed to classify all instances of 'appear' and 'birthday correctly. Two instances from 'appear' was classified as 'accept' whereas an instance of 'birthday' was classified as 'buy'. Based on the classification report, 'accept' and 'buy' each has a precision of 0.83 and 0.91 respectively, which implies that of the positive classification the model made with respective to each of these classes, 83% and 91% are truly positive. On the other hand, for both words 'appear' and 'birthday', the model has correctly predicted 80% and 90% of the actual positive instances of these two classes, respectively.

```
5/5 [==============================] - 1s 87ms/step - loss: 0.0799 - accuracy: 0.9800
```

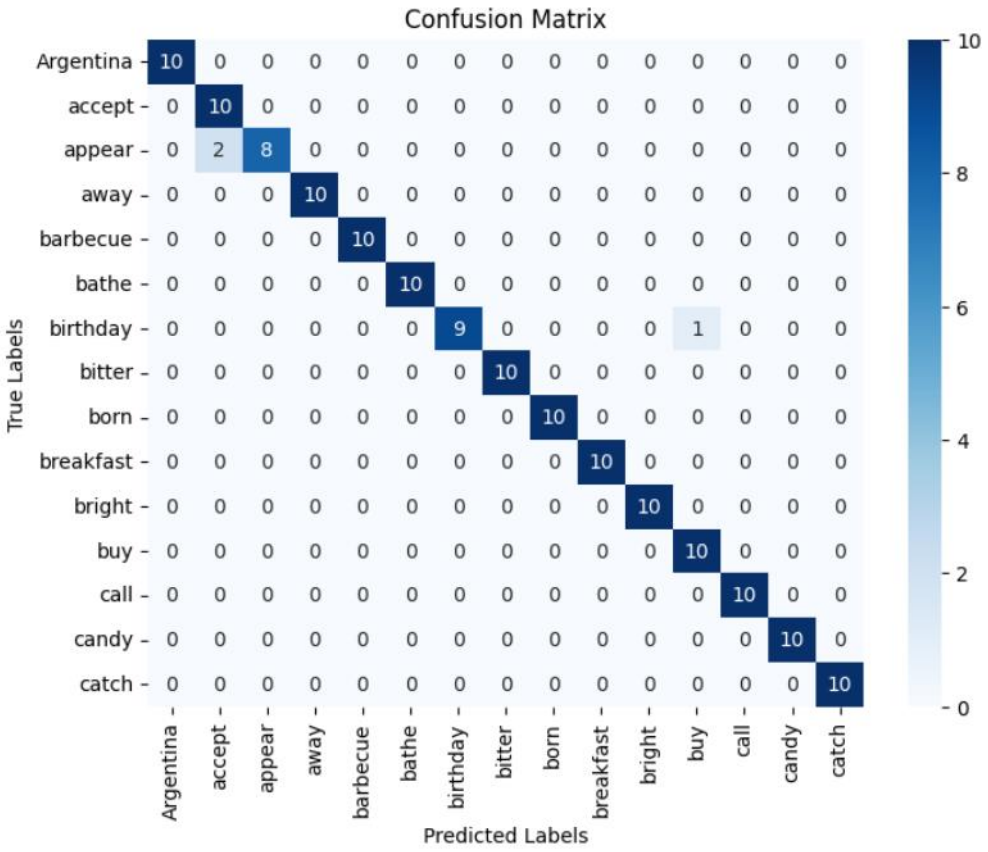

Figure 5 Confusion Matrix of Mediapipe-Model 1 on LSA64

	precision	recall	f1-score	support
Argentina	1.00	1.00	1.00	10
accept	0.83	1.00	0.91	10
appear	1.00	0.80	0.89	10
away	1.00	1.00	1.00	10
barbecue	1.00	1.00	1.00	10
bathe	1.00	1.00	1.00	10
birthday	1.00	0.90	0.95	10
bitter	1.00	1.00	1.00	10
born	1.00	1.00	1.00	10
breakfast	1.00	1.00	1.00	10
bright	1.00	1.00	1.00	10
buy	0.91	1.00	0.95	10
call	1.00	1.00	1.00	10
candy	1.00	1.00	1.00	10
catch	1.00	1.00	1.00	10
accuracy			0.98	150
macro avg	0.98	0.98	0.98	150
weighted avg	0.98	0.98	0.98	150

Figure 6 Classification report of Mediapipe-model 1 on LSA64

WLASL dataset

Model Fitness

The model was trained for 200 epochs. From the accuracy and loss graph, the model performance showed overfitting starting from epoch 50. Both training and validation accuracy as well as loss were improving in the first 50 epochs, the model achieved 95% accuracy and flatten till epoch 200 in the training set. However, the validation accuracy achieved close to 60% accuracy at epoch 50 and stops improving despite some small fluctuations until epoch 200. At the end of training, the training accuracy was 99.46% with 0.0281 loss, whereas validation accuracy was 57.50% with higher loss 2.3904.

```
Epoch 200/200
245/245 [==============================] - 44s 178ms/step - loss: 0.0281 - accuracy: 0.9946 - val_loss: 2.3904 - val_accurac
y: 0.5750
```

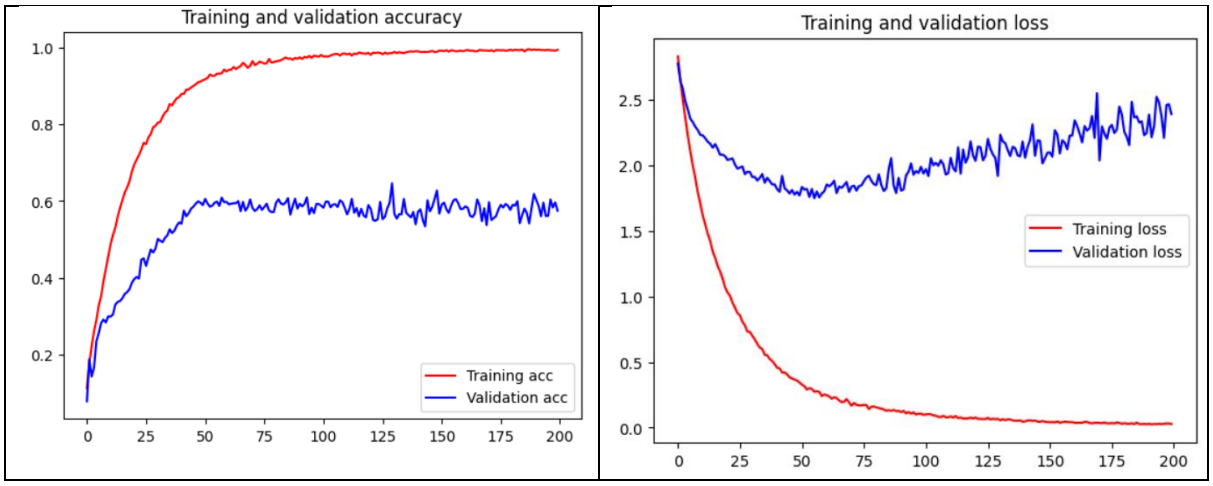

Figure 7 Training, Validation Accuracy and Loss across 200 epochs of Model 1 WLASL dataset.

Test Set

The model achieved 60% accuracy and 5.2784 loss on test set. Based on the confusion matrix, the model failed to recognize ‘africa’, ‘again’, ‘ago’, ‘all day’, ‘and’, and ‘baseball’ correctly, which is close to 50% of the total class. As this dataset is imbalance and all classes share equal level of importance, macro average of the precision, recall and f1-score are evaluated. Based on the classification report, the macro average f1-score is 0.53, 0.5 precision and 0.6 recall. ‘Bacon’, ‘benefit’, and ‘birthday’ has precision of 0.50 respectively, which implies that of the positive classification the model made with respective to each of these classes, 50% are truly positive.

```
1/1 [==============================] - 0s 106ms/step - loss: 5.2784 - accuracy: 0.6000
```

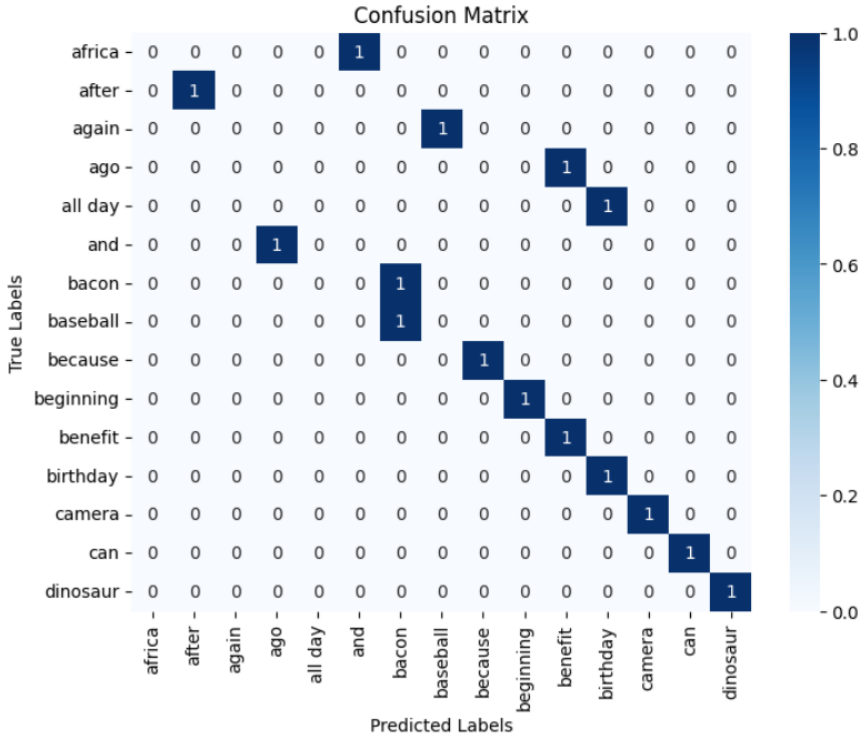

Figure 8 Confusion Matrix of Model 1 on WLASL dataset

	precision	recall	f1-score	support
africa	0.00	0.00	0.00	1
after	1.00	1.00	1.00	1
again	0.00	0.00	0.00	1
ago	0.00	0.00	0.00	1
all day	0.00	0.00	0.00	1
and	0.00	0.00	0.00	1
bacon	0.50	1.00	0.67	1
baseball	0.00	0.00	0.00	1
because	1.00	1.00	1.00	1
beginning	1.00	1.00	1.00	1
benefit	0.50	1.00	0.67	1
birthday	0.50	1.00	0.67	1
camera	1.00	1.00	1.00	1
can	1.00	1.00	1.00	1
dinosaur	1.00	1.00	1.00	1
accuracy			0.60	15
macro avg	0.50	0.60	0.53	15
weighted avg	0.50	0.60	0.53	15

Figure 9 Classification Report of Model 1 on WLASL dataset

6.1.2 Model 2 Conv-GRU

In general, the architecture of model 2 is similar to model 1. The only changes made was to replace LSTM layers with GRU layers while holding every hyperparameter constant. As GRU is an improved version of LSTM, this model was less computation expensive than the LSTM model.

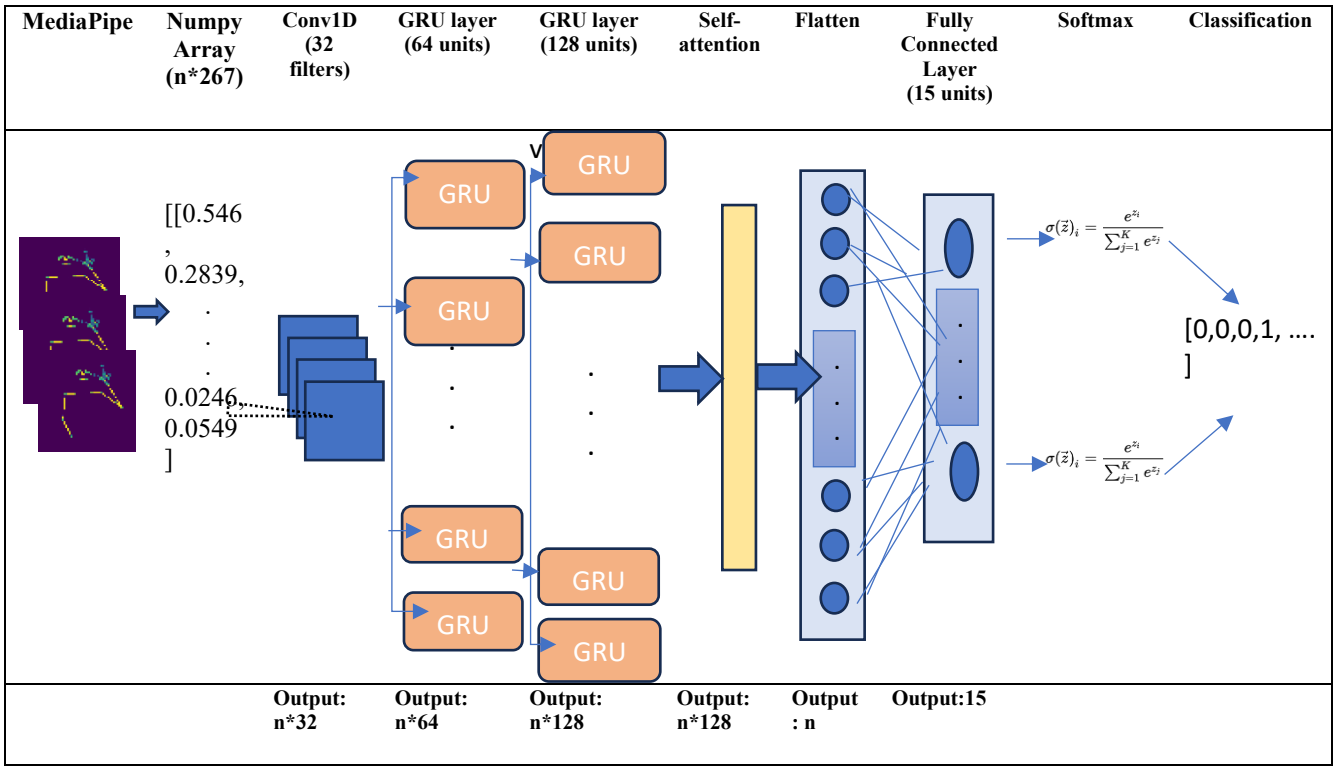

Figure 10 MediaPipe 3D Landmark Model 2 Architecture

Similarly, due to the difference in the sequence length. LSA64 dataset has higher number of trainable parameters than WLASL dataset. As a result, the total number of trainable parameters of LSA64 was slightly higher, with 28,800 parameters more. As each GRU cell consists fewer gates compared to LSTM, therefore it is less computationally intensive. GRU model has 30,336 parameters lesser than LSTM model. The number of parameters for a function within the GRU unit is determined by the number of features and number of units in GRU. Since GRU has 3 functions, the total number of parameters is three times higher. For example, for the first GRU layer on LSA64 dataset, the total number of parameters would be ((32(depth) + 64 (GRU units))*64(GRU units) +64unit+64unis)*3=18,816. Besides that, by applying batch

normalization, there were 384 non-trainable parameters. By using batch normalization, there were also 384 non-trainable parameters in both models.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 190, 32)           25664

 max_pooling1d (MaxPooling1  (None, 63, 32)            0
 D)

 dropout (Dropout)           (None, 63, 32)            0

 gru (GRU)                   (None, 63, 64)            18816

 dropout_1 (Dropout)         (None, 63, 64)            0

 batch_normalization (Batch  (None, 63, 64)            256
 Normalization)

 gru_1 (GRU)                 (None, 63, 128)           74496

 dropout_2 (Dropout)         (None, 63, 128)           0

 batch_normalization_1 (Bat  (None, 63, 128)           512
 chNormalization)

 seq_self_attention (SeqSel  (None, 63, 128)           8257
 fAttention)

 flatten (Flatten)           (None, 8064)              0

 dense (Dense)               (None, 15)                120975

=================================================================
Total params: 248976 (972.56 KB)
Trainable params: 248592 (971.06 KB)
Non-trainable params: 384 (1.50 KB)
```

Figure 28: Mediapipe 3D-Landmark-Model 2 Summary on LSA64 dataset

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_1 (Conv1D)           (None, 145, 32)           25664

 max_pooling1d_1 (MaxPoolin  (None, 48, 32)            0
 g1D)

 dropout_3 (Dropout)         (None, 48, 32)            0

 gru (GRU)                   (None, 48, 64)            18816

 dropout_4 (Dropout)         (None, 48, 64)            0

 batch_normalization_2 (Bat  (None, 48, 64)            256
 chNormalization)

 gru_1 (GRU)                 (None, 48, 128)           74496

 dropout_5 (Dropout)         (None, 48, 128)           0

 batch_normalization_3 (Bat  (None, 48, 128)           512
 chNormalization)

 seq_self_attention_1 (SeqS  (None, 48, 128)           8257
 elfAttention)

 flatten_1 (Flatten)         (None, 6144)              0

 dense_1 (Dense)             (None, 15)                92175

=================================================================
Total params: 220176 (860.06 KB)
Trainable params: 219792 (858.56 KB)
Non-trainable params: 384 (1.50 KB)
```

Figure 29: Mediapipe 3D-Landmark-Model 2 Summary on WLASL dataset

6.1.2.1 Performance

LSA64 dataset

Model Fitness

The model was trained for 200 epochs. From the accuracy and loss graph, the model performance showed a good fit. Both training and validation accuracy as well as loss were improving in the first 20 epochs, reaching 95% accuracy and flatten until the end of training. This model converges faster than model 1, at epoch 151, the training accuracy was 99.35% with 0.0461 loss whereas validation accuracy was 100% with 0.0320 loss.

```
Epoch 151/200
136/136 [==============================] - 40s 291ms/step - loss: 0.0461 - accuracy: 0.9935 - val_loss: 0.0320 - val_accurac
y: 1.0000
```

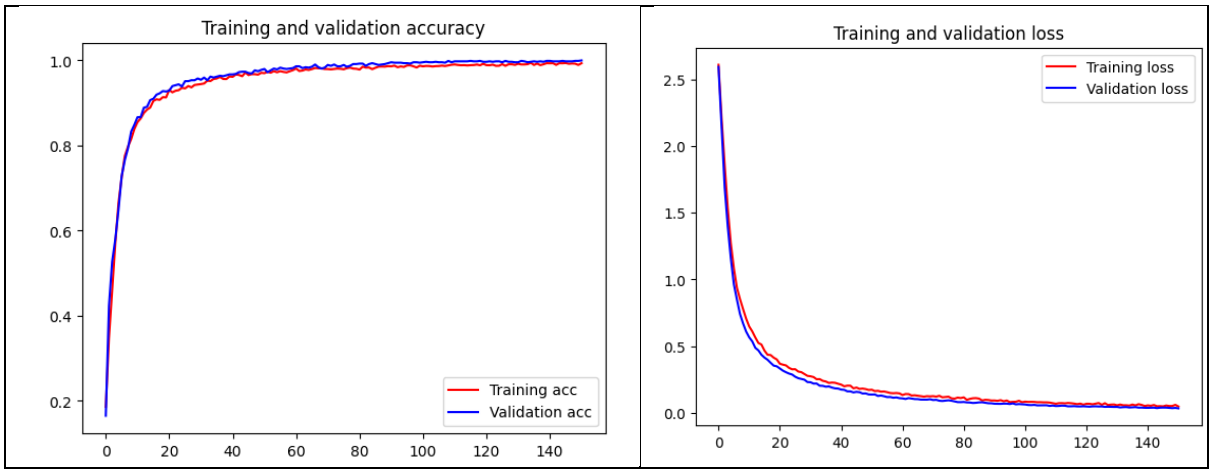

Figure 11 Mediapipe-Model 2 Training and Validation Accuracy and Loss on LSA64 dataset.

Test set

The model achieved 97.33% accuracy and 0.2267 loss on test set. Based on the confusion matrix, the model failed to classify all instances of 'appear' and 'buy' correctly. Similar to model 1, two instances from 'appear' was classified as 'accept' whereas two instances of 'buy' was classified as 'birthday'. Based on the classification report, 'accept' and 'birthday' each has a precision of 0.83 respectively, which implies that of the positive classification the model made with respective to each of these classes, 83% was truly positive. On the other hand, for both words 'appear' and 'buy', the model has correctly predicted 80% of the actual positive instances of these two classes.

```
5/5 [==============================] - 0s 75ms/step - loss: 0.2267 - accuracy: 0.9733
```

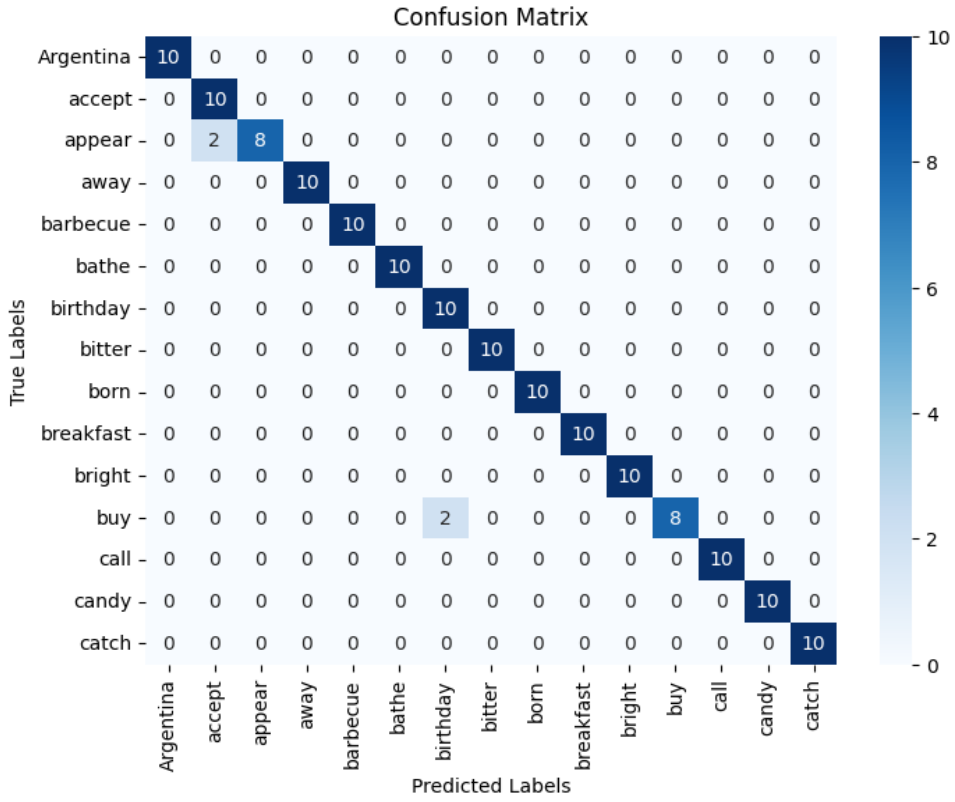

Figure 12 : Mediapipe 3D-Landmark-Model 2 Confusion Matrix on LSA64 dataset.

	precision	recall	f1-score	support
Argentina	1.00	1.00	1.00	10
accept	0.83	1.00	0.91	10
appear	1.00	0.80	0.89	10
away	1.00	1.00	1.00	10
barbecue	1.00	1.00	1.00	10
bathe	1.00	1.00	1.00	10
birthday	0.83	1.00	0.91	10
bitter	1.00	1.00	1.00	10
born	1.00	1.00	1.00	10
breakfast	1.00	1.00	1.00	10
bright	1.00	1.00	1.00	10
buy	1.00	0.80	0.89	10
call	1.00	1.00	1.00	10
candy	1.00	1.00	1.00	10
catch	1.00	1.00	1.00	10
accuracy			0.97	150
macro avg	0.98	0.97	0.97	150
weighted avg	0.98	0.97	0.97	150

Figure 13 Mediapipe 3D-Landmark-Model 2 Classification Report on LSA64 dataset.

WLASL Dataset

Model Fitness

The model was trained for 200 epochs. From the accuracy and loss graph, the model performance showed overfitting starting from epoch 25. Both training and validation accuracy as well as loss were improving in the first 50 epochs, the model achieved 95% accuracy and flatten till epoch 200 in the training set. However, the validation accuracy achieved close to 50% accuracy at epoch 50 and continuously having small improvement until epoch 200 reaching more than 65% of accuracy. However, validation loss did not decrease further after epoch 25, maintaining at around 2.0. At the end of training, the training accuracy was 98.74% with 0.0547 loss, whereas validation accuracy was 65.69% with higher loss 1.8550.

```
Epoch 200/200
245/245 [==============================] - 45s 184ms/step - loss: 0.0547 - accuracy: 0.9874 - val_loss: 1.8550 - val_accurac
y: 0.6569
```

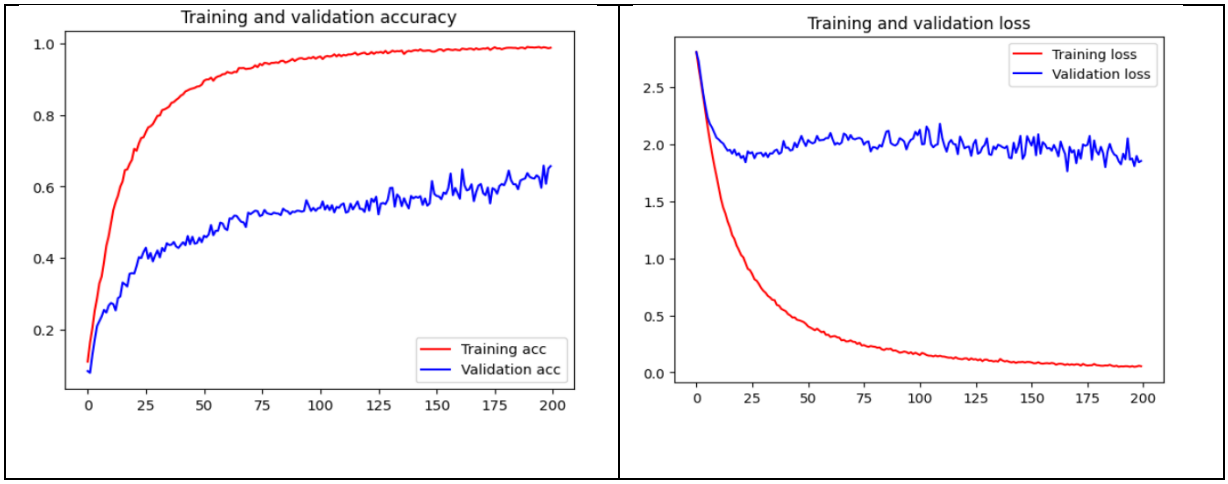

Figure 14 Mediapipe-Model 2 Training and Validation Accuracy and Loss on WLASL Dataset

Test set

The model achieved 66.67% accuracy and 2.8769 loss on test set. Based on the confusion matrix, the model failed to recognize ‘again’, ‘ago’, ‘all day’, ‘and’, and ‘baseball’ correctly. As this dataset is imbalance and all classes share equal level of importance, macro average of the precision, recall and f1-score are evaluated. Based on the classification report, the macro average f1-score is 0.62, 0.6 precision and 0.67 recall. In general, of the positive classification

the model made with respective to each of these classes, 60% was truly positive. On the other hand, the model has correctly predicted 67% of the actual positive instances of the classes.

```
1/1 [==============================] - 0s 82ms/step - loss: 2.8769 - accuracy: 0.6667
```

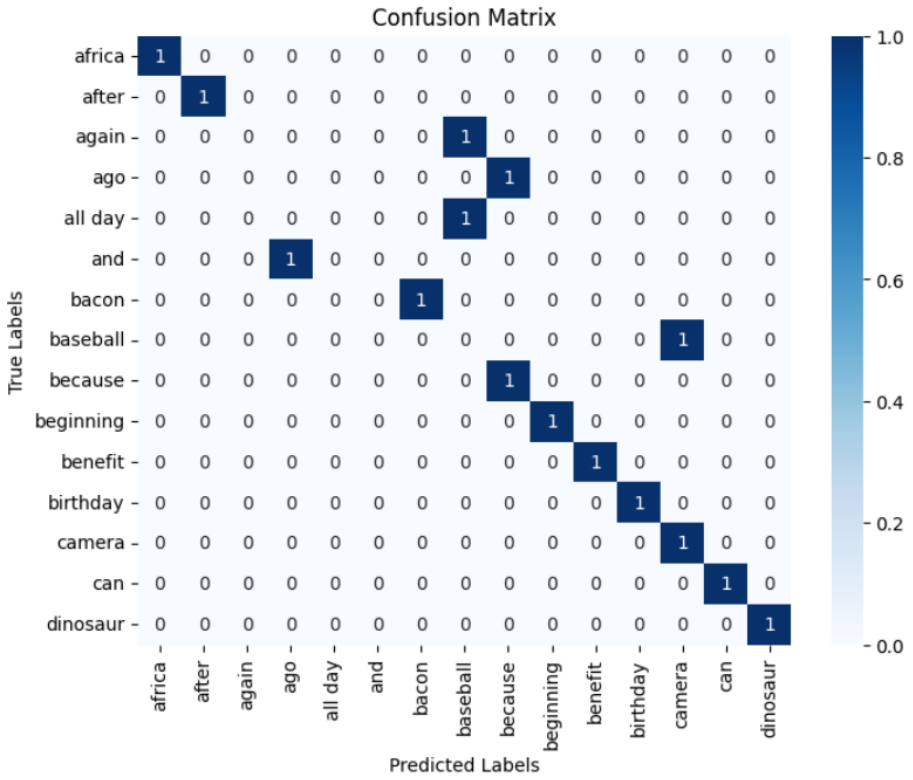

Figure 15 MediaPipe3D-Landmark-Model 2 Confusion Matrix on WLASL dataset

	precision	recall	f1-score	support
africa	1.00	1.00	1.00	1
after	1.00	1.00	1.00	1
again	0.00	0.00	0.00	1
ago	0.00	0.00	0.00	1
all day	0.00	0.00	0.00	1
and	0.00	0.00	0.00	1
bacon	1.00	1.00	1.00	1
baseball	0.00	0.00	0.00	1
because	0.50	1.00	0.67	1
beginning	1.00	1.00	1.00	1
benefit	1.00	1.00	1.00	1
birthday	1.00	1.00	1.00	1
camera	0.50	1.00	0.67	1
can	1.00	1.00	1.00	1
dinosaur	1.00	1.00	1.00	1
accuracy			0.67	15
macro avg	0.60	0.67	0.62	15
weighted avg	0.60	0.67	0.62	15

Figure 16 MediaPipe 3D-Landmark -Model 2 Classification Report on WLASL dataset.

6.2 Method II: Frames Sequence with MediaPipe Landmark

In the second approach, the raw videos were applied with MediaPipe Holistic Landmarker. MediaPipe Holistic Landmarker allows to combine the pose, face, and hand landmarks. Next, from each processed video, starting at 500mili seconds, 8 frames with 12 frame steps apart were extracted from each video from LSA64 dataset whereas 6 frames with 10 frame steps part for WLASL dataset. To reduce computation cost, these frames were further transformed to grayscale images and reduced to 100*100 resolution. Finally, these frame sequences were then passed into the model. Due to memory constraints, all models of method 2 were trained on Google Colab notebook. Google Colab notebooks have absolute timeout of 24 hours. Hence, the number of epochs were hold constant as 20 for all models, this is to ensure that the number of epochs training can be completed in time.

6.2.1 Model 1 3D-Conv-LSTM

The first model combines two 3D-Conv layers, each followed by a maxpooling and dropout layer at 0.5 rate. From the first layer until the flatten layer, the temporal dimension was hold constant. It is observed that upon application of the first CNN layer, the number of channels changed from 1 channel to 32, reflecting the number of filters being applied. Hence, the depth of the image was elongated to 32 output feature maps in this process. Similar observations were seen for the second 3D Conv layer with 64 output feature maps. The output shape of the 3D-Conv layer depends on the kernel size, padding and stride.

$$output = \frac{input - kernel\_size + 2 * padding}{stride} + 1$$

Figure 6.19:17 Equation for calculating the output shape of convolutional layer. (Ding, 2020)

Followed by a max pooling layer which down samples the input spatial dimensions by selecting the maximum value when an input window strides along each dimension. As ‘valid’ padding option was used, the resulting output shape is determined by ((input shape – pool size) / strides) +1 (MaxPooling2D layer, n.d.). Hence the output shape was 50*50*32. Similarly, after the maxpooling layer was added to the second convolution layer, the output spatial dimension was

downsized to 25*25. Each maxpooling layer was followed by rectified linear unit activation function (ReLU), an activation layer. Before feeding the data into LSTM layer, the input data was reshaped to a 2-dimensional array. A LSTM layer with 4 units was added followed by a self-attention layer to focus more on relevant information. Finally, the model was ended with a fully-connected layer with 15 units. In 3D-CNN layer, 'ReLU' activation function was applied while 'tanh' activation function was used in the GRU layer. On the other hand, 'sigmoid' activation function was used in the self-attention layer and lastly, 'softmax' was used in the fully-connected layer.

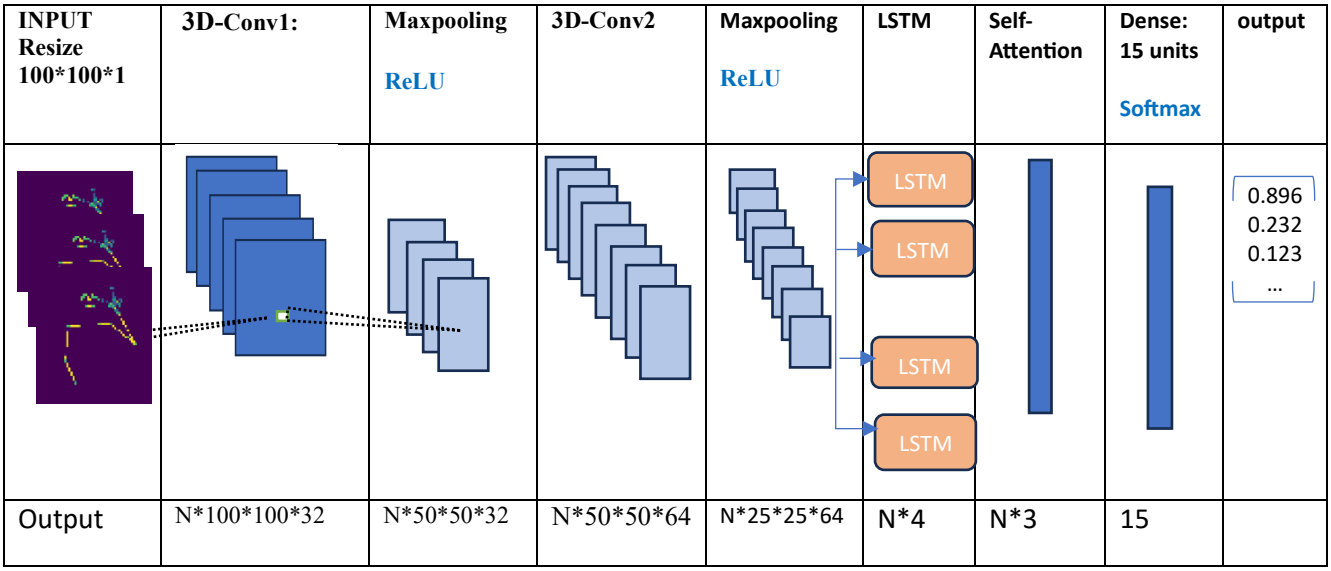

Figure 18 Frame-sequence Model 1 Architecture

Since 8 frames with 12 frame steps apart were extracted from each video from LSA64 dataset whereas 6 frames with 10 frame steps part for WLASL dataset, the model for LSA64 dataset has a slightly higher number of parameters, 120 compared to WLASL dataset. Both dataset has approximately 697,000 trainable parameters. Note that with maxpooling layer, the spatial resolution of the video frames were reduced to half of the input resolution.

```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_6 (InputLayer)        [(None, 8, 100, 100, 1)]  0

 conv3d_10 (Conv3D)          (None, 8, 100, 100, 32)   896

 max_pooling3d_9 (MaxPooling  (None, 8, 50, 50, 32)    0
 3D)

 re_lu_9 (ReLU)              (None, 8, 50, 50, 32)     0

 dropout_11 (Dropout)        (None, 8, 50, 50, 32)     0

 conv3d_11 (Conv3D)          (None, 8, 50, 50, 64)     55360

 max_pooling3d_10 (MaxPoolin  (None, 8, 25, 25, 64)    0
 g3D)

 re_lu_10 (ReLU)             (None, 8, 25, 25, 64)     0

 dropout_12 (Dropout)        (None, 8, 25, 25, 64)     0

 reshape_4 (Reshape)         (None, 8, 40000)          0

 lstm_3 (LSTM)               (None, 8, 4)              640080

 seq_self_attention_4 (SeqSe  (None, 8, 4)             321
 lfAttention)

 flatten_4 (Flatten)         (None, 32)                0

 dense_4 (Dense)             (None, 15)                495

=================================================================
Total params: 697,152
Trainable params: 697,152
Non-trainable params: 0
```

Figure 37 3D-Conv LSTM model summary for LSA64 dataset

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 6, 100, 100, 1)   0
                             ]

 conv3d_2 (Conv3D)           (None, 6, 100, 100, 32)   896

 max_pooling3d_2 (MaxPoolin  (None, 6, 50, 50, 32)     0
 g3D)

 re_lu_2 (ReLU)              (None, 6, 50, 50, 32)     0

 dropout_2 (Dropout)         (None, 6, 50, 50, 32)     0

 conv3d_3 (Conv3D)           (None, 6, 50, 50, 64)     55360

 max_pooling3d_3 (MaxPoolin  (None, 6, 25, 25, 64)     0
 g3D)

 re_lu_3 (ReLU)              (None, 6, 25, 25, 64)     0

 dropout_3 (Dropout)         (None, 6, 25, 25, 64)     0

 reshape_1 (Reshape)         (None, 6, 40000)          0

 lstm_1 (LSTM)               (None, 6, 4)              640080

 seq_self_attention_1 (SeqS  (None, 6, 4)              321
 elfAttention)

 flatten_1 (Flatten)         (None, 24)                0

 dense_1 (Dense)             (None, 15)                375

=================================================================
Total params: 697032 (2.66 MB)
Trainable params: 697032 (2.66 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 38: 3D-Conv-LSTM model summary for WLASL dataset.

6.2.1.1 Performance

LSA64 Dataset

From the accuracy and loss graph, the model performance showed overfitting starting from epoch 5. Both training and validation accuracy as well as loss were improving gradually from epoch 0 to 12. The model achieved more than 95% accuracy and started to flatten at epoch 15 in the training set. However, the validation accuracy achieved close to 70% accuracy at epoch 5 and continuously having small improvement until epoch 13 reaching more than 75% of accuracy. However, validation loss did not decrease further after epoch 10, maintaining at around 1.0. At the end of training, the training accuracy was 97.01% with 0.1626 loss, whereas validation accuracy was 78.23% with higher loss 0.9523.

```
Epoch 20/20
120/120 [==============================] - 1223s 10s/step - loss: 0.1626 - accuracy: 0.9701 - val_loss: 0.9523 - val_accuracy: 0.7823
```

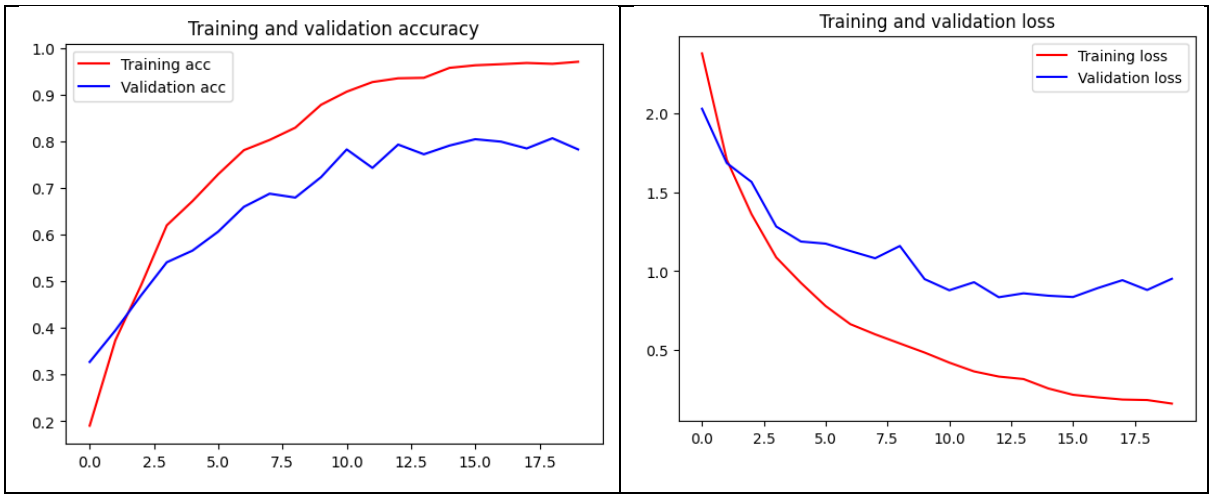

Figure 19 Frame Sequence-Model 1 Training and Validation Accuracy and Loss on LSA64 Dataset

Test set

The model achieved 86% accuracy and 0.4753 loss on test set. Based on the confusion matrix, the model failed to recognize 'birthday', 'born' and 'breakfast' signs correctly. Based on the classification report, the model achieved an f1-score of 0.91. Based on the classification report, 'bathe', 'call' and 'candy' each has a precision of 0.5, 0.75 and 0.5 respectively, which implies that of the positive classification the model made with respective to each of these classes, 50% or 75% was truly positive. On the other hand, for 'birthday'. 'born', 'breakfast' signs, the model

has correctly predicted 50%, 67% and 50% of the actual positive instances from these three classes.

```
#evaluate on test set
test_loss, test_accuracy = model.evaluate(test_ds)
```

```
5/5 [==============================] - 139s 26s/step - loss: 0.4753 - accuracy: 0.8600
```

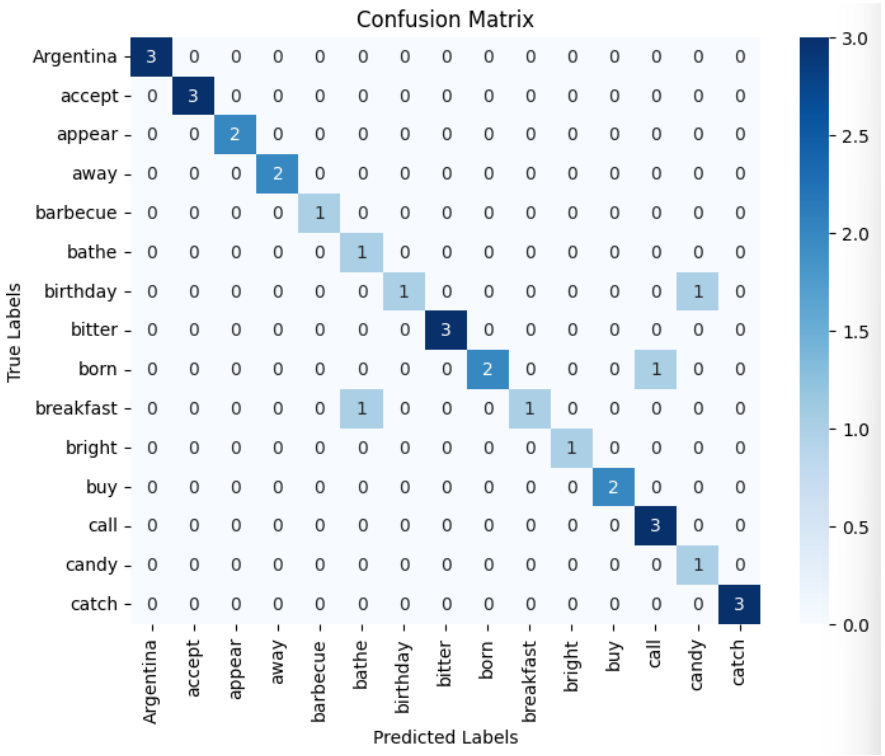

Figure 20 Frame Sequence--Model 1 Confusion Matrix on LSA64 dataset

	precision	recall	f1-score	support
Argentina	1.00	1.00	1.00	3
accept	1.00	1.00	1.00	3
appear	1.00	1.00	1.00	2
away	1.00	1.00	1.00	2
barbecue	1.00	1.00	1.00	1
bathe	0.50	1.00	0.67	1
birthday	1.00	0.50	0.67	2
bitter	1.00	1.00	1.00	3
born	1.00	0.67	0.80	3
breakfast	1.00	0.50	0.67	2
bright	1.00	1.00	1.00	1
buy	1.00	1.00	1.00	2
call	0.75	1.00	0.86	3
candy	0.50	1.00	0.67	1
catch	1.00	1.00	1.00	3
accuracy			0.91	32
macro avg	0.92	0.91	0.89	32
weighted avg	0.95	0.91	0.91	32

Figure 21 Frame Sequence-Model 1 Classification report on LSA64 dataset.

WLASL dataset

Overfitting is significant when using this model on the WLASL dataset. It is observed that the training accuracy and loss had been improving throughout the 20 epoch and reaching 94.92% accuracy with 0.4427 loss at epoch 20. Conversely, the validation accuracy was almost flatten while validation loss was increasing gradually throughout the 20 epochs. At epoch 20, the validation accuracy was 17% with 3.6310 loss.

```
Epoch 20/20
30/30 [==============================] - 191s 6s/step - loss: 0.4427 - accuracy: 0.9492 - val_loss: 3.6310 - val_accuracy: 0.17
57
```

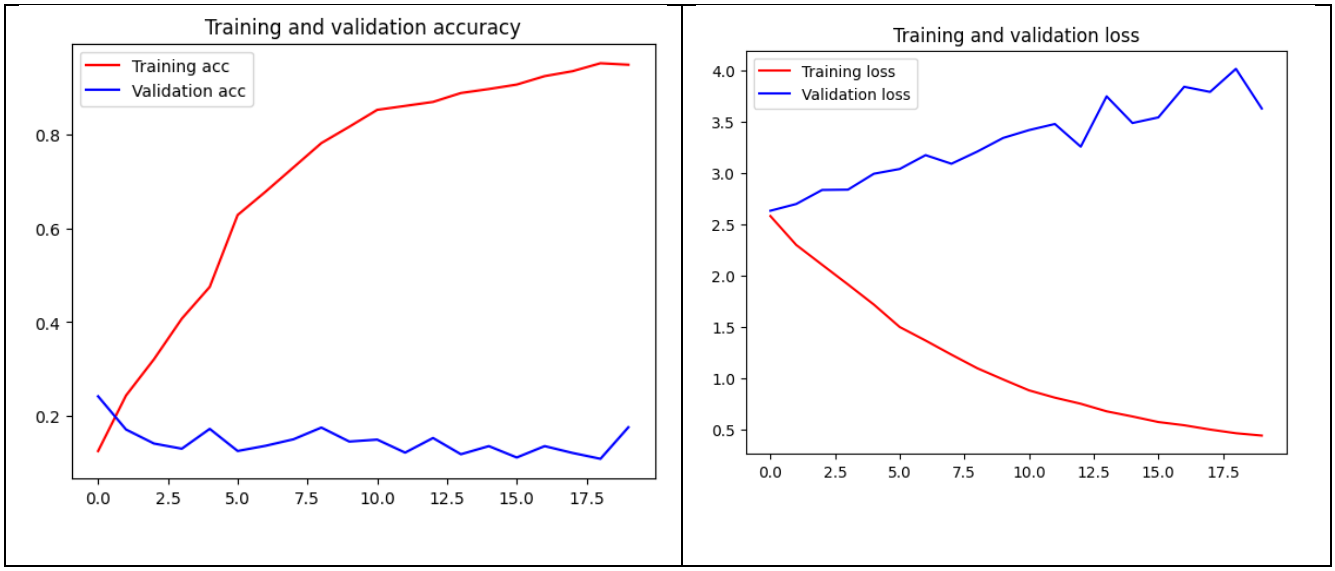

Figure 22 Frame Sequence-Model 1 Training and Validation Accuracy and Loss on WLASL Dataset

Test set

The model achieved 20% accuracy and 4.2496 loss on test set. Based on the confusion matrix, the model classified only 4 signs correctly which are ‘beginning’, ‘birthday’ and ‘can’. As this dataset is imbalance and all classes share equal level of importance, macro average of the precision, recall and f1-score are evaluated. Based on the classification report, the macro average f1-score is 0.17, 0.16 precision and 0.2 recall. In general, of the positive classification the model made with respective to each of these classes, 16% was truly positive. On the other hand, the model has correctly predicted 20% of the actual positive instances of the classes.

```
1/1 [==============================] - 0s 236ms/step - loss: 4.2496 - accuracy: 0.2000
```

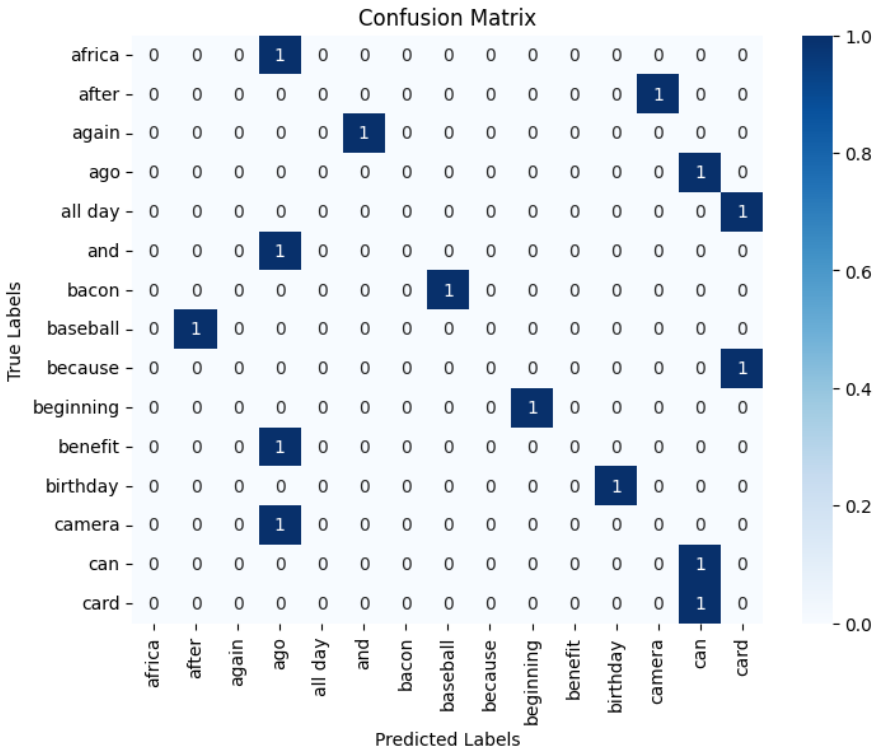

Figure 23 Frame Sequence--Model 1 Confusion Matrix on WLASL dataset

	precision	recall	f1-score	support
africa	0.00	0.00	0.00	1
after	0.00	0.00	0.00	1
again	0.00	0.00	0.00	1
ago	0.00	0.00	0.00	1
all day	0.00	0.00	0.00	1
and	0.00	0.00	0.00	1
bacon	0.00	0.00	0.00	1
baseball	0.00	0.00	0.00	1
because	0.00	0.00	0.00	1
beginning	1.00	1.00	1.00	1
benefit	0.00	0.00	0.00	1
birthday	1.00	1.00	1.00	1
camera	0.00	0.00	0.00	1
can	0.33	1.00	0.50	1
card	0.00	0.00	0.00	1
accuracy			0.20	15
macro avg	0.16	0.20	0.17	15
weighted avg	0.16	0.20	0.17	15

Figure 24 Frame Sequence-Model 1 Classification report on WLASL dataset

.

6.2.2 Model 2 3DConv-GRU

In general, the architecture of model 2 is similar to model 1. While holding every hyperparameter constant, the only changes made was to replace LSTM layers with GRU layers. As GRU is an improved version of LSTM, this model was less computation expensive than the LSTM model.

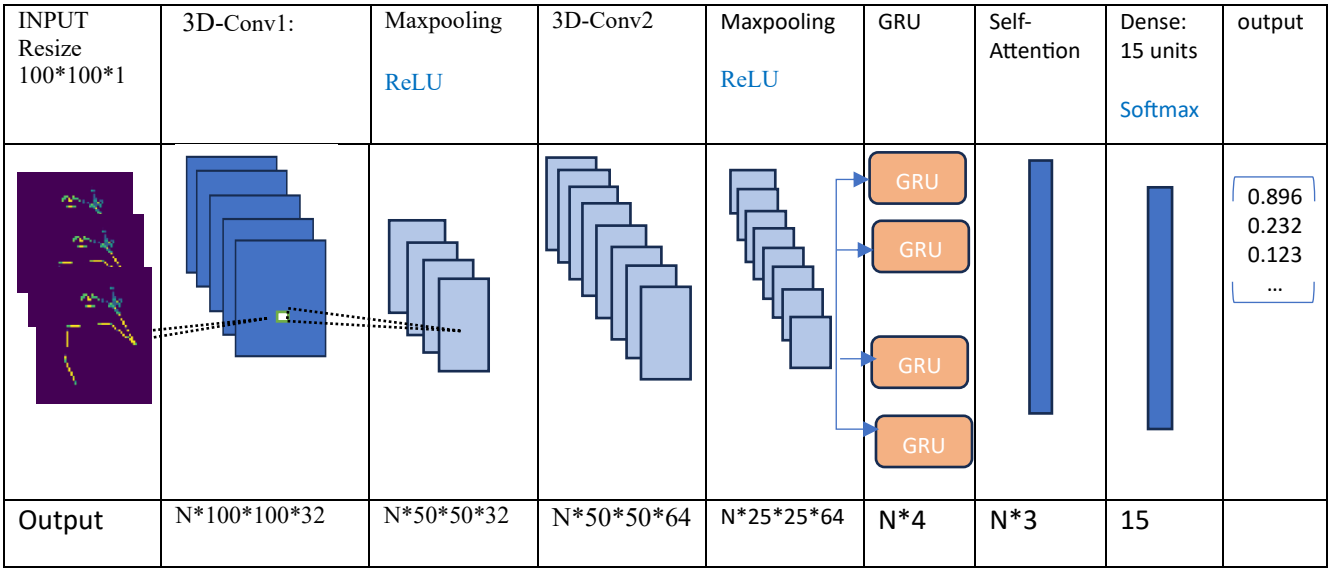

Figure 25 3D-Conv-GRU Architecture

Similarly, due to the difference in the number of frames extracted, the sequence length differs in both datasets. As a results, the number of parameters is slightly higher for LSA64 dataset, by 90 more parameters.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 8, 100, 100, 1)]  0

 conv3d (Conv3D)             (None, 8, 100, 100, 32)   896

 max_pooling3d (MaxPooling3D  (None, 8, 50, 50, 32)    0
 )

 re_lu (ReLU)                (None, 8, 50, 50, 32)     0

 dropout (Dropout)           (None, 8, 50, 50, 32)     0

 conv3d_1 (Conv3D)           (None, 8, 50, 50, 64)     55360

 max_pooling3d_1 (MaxPooling  (None, 8, 25, 25, 64)    0
 3D)

 re_lu_1 (ReLU)              (None, 8, 25, 25, 64)     0

 dropout_1 (Dropout)         (None, 8, 25, 25, 64)     0

 reshape (Reshape)           (None, 8, 40000)          0

 gru (GRU)                   (None, 8, 4)              480072

 seq_self_attention (SeqSelf  (None, 8, 4)             321
 Attention)

 flatten (Flatten)           (None, 32)                0

 dense (Dense)               (None, 15)                495

=================================================================
Total params: 537,144
Trainable params: 537,144
Non-trainable params: 0
_________________________________________________________________
```

Figure 46 Frame sequence 3D-ConvGRU Model Summary on LSA64 Dataset.

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 6, 100, 100, 1)]  0

 conv3d_2 (Conv3D)           (None, 6, 100, 100, 32)   896

 max_pooling3d_2 (MaxPooling  (None, 6, 50, 50, 32)    0
 3D)

 re_lu_2 (ReLU)              (None, 6, 50, 50, 32)     0

 dropout_2 (Dropout)         (None, 6, 50, 50, 32)     0

 conv3d_3 (Conv3D)           (None, 6, 50, 50, 64)     55360

 max_pooling3d_3 (MaxPooling  (None, 6, 25, 25, 64)    0
 3D)

 re_lu_3 (ReLU)              (None, 6, 25, 25, 64)     0

 dropout_3 (Dropout)         (None, 6, 25, 25, 64)     0

 reshape_1 (Reshape)         (None, 6, 40000)          0

 gru_1 (GRU)                 (None, 6, 4)              480072

 seq_self_attention_1 (SeqSe  (None, 6, 4)             321
 lfAttention)

 flatten_1 (Flatten)         (None, 24)                0

 dense_1 (Dense)             (None, 15)                375

=================================================================
Total params: 537,024
Trainable params: 537,024
Non-trainable params: 0
```

Figure 47 Frame sequence 3D-ConvGRU Model Summary on WLASL Dataset

6.2.2.1 Performance

LSA64 dataset

The model was trained with 0.0001 learning rate for 30 epochs. Overfitting was observed from both accuracy and loss graphs. The training accuracy increases gradually throughout 30 epochs and reaching close to 100% whereas validation accuracy increases at a lower rate, stopping at 77%. On the other hand, it was noted that the training loss increases gradually while validation loss flattens at 1.2.

```
Epoch 20/20
120/120 [==============================] - 1712s 14s/step - loss: 0.0700 - accuracy: 0.9937 - val_loss: 1.2999 - val_accuracy: 0.7719
```

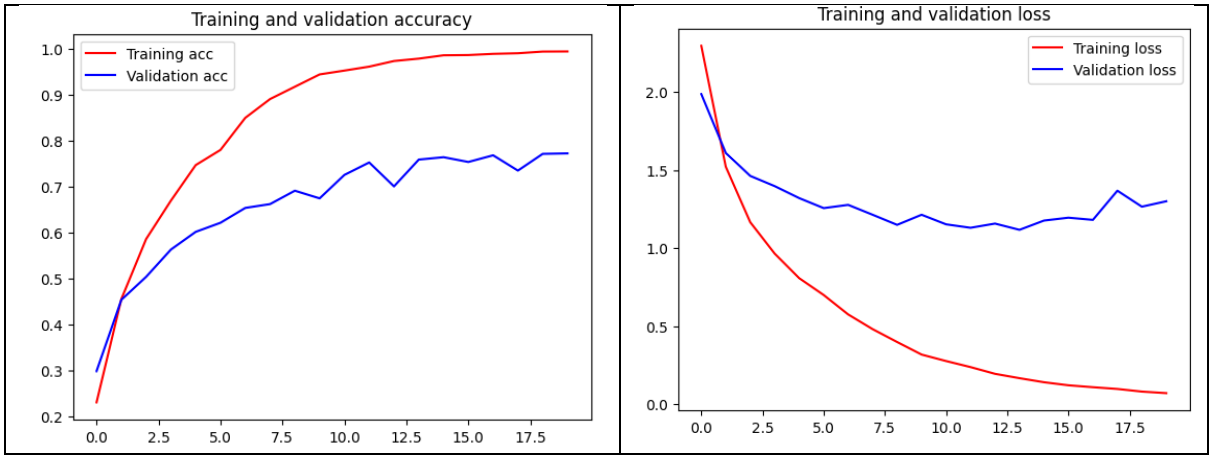

Figure 26 Frame sequence-Model 2 Training, Validation Accuracy and Loss on LSA64 dataset

.

Test set

The model achieved 92.67% accuracy and 0.2995 loss on test set. Based on the confusion matrix, the model failed to recognize all instances of 3 out of 15 signs correctly. Based on the classification report, the model achieved an f1-score of 0.91. Based on the classification report, 'bathe' and 'candy' has a precision of 0.75 and 0.67 respectively, which implies that of the positive classification the model made with respective to each of these classes, 75% or 67% was truly positive, while the model failed to recognize the sign for 'bitter' and 'catch' completely. On the other hand, for 'appear' and 'buy' has a recall of 0.5 and 0.67, the model has correctly predicted 50% and 67% of the actual positive instances from these classes.

```
5/5 [==============================] - 171s 34s/step - loss: 0.2995 - accuracy: 0.9267
```

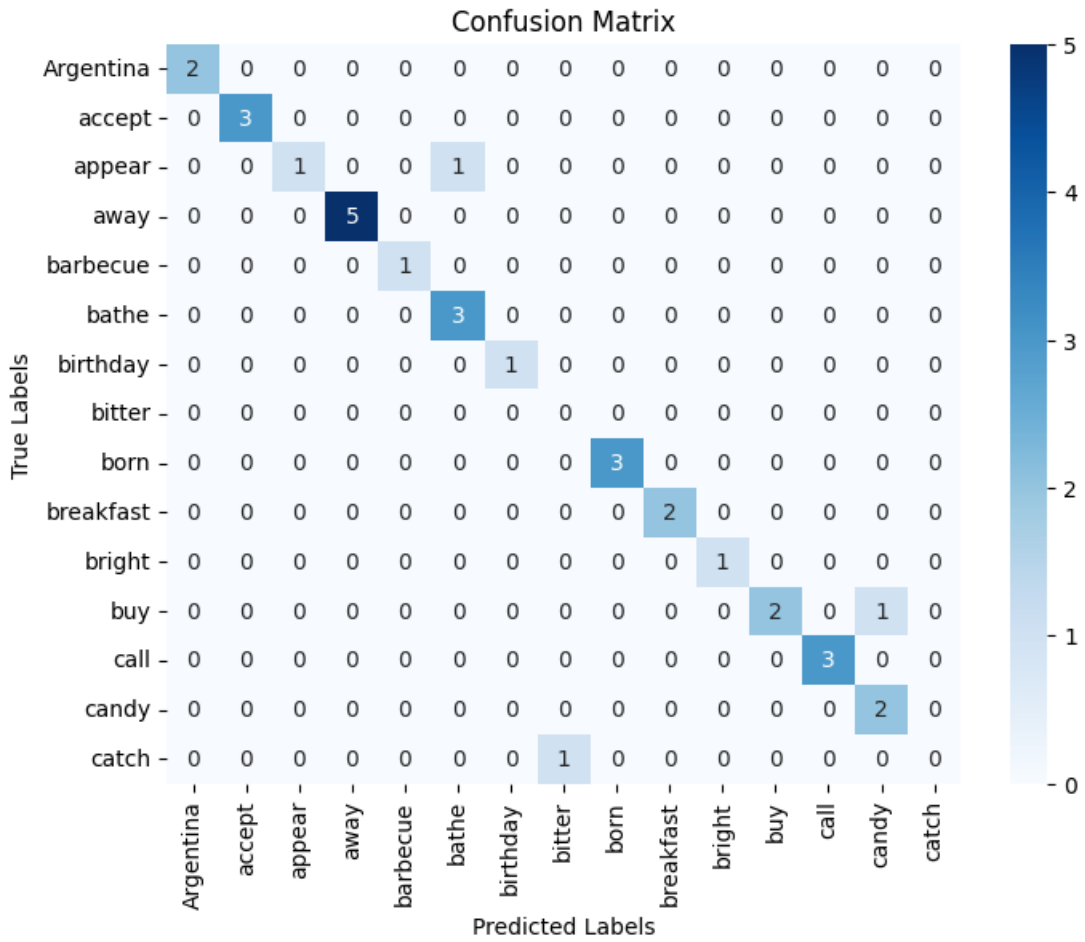

Figure 27 Frame Sequence--Model 2Confusion Matrix on LSA64 dataset

	precision	recall	f1-score	support
Argentina	1.00	1.00	1.00	2
accept	1.00	1.00	1.00	3
appear	1.00	0.50	0.67	2
away	1.00	1.00	1.00	5
barbecue	1.00	1.00	1.00	1
bathe	0.75	1.00	0.86	3
birthday	1.00	1.00	1.00	1
bitter	0.00	0.00	0.00	0
born	1.00	1.00	1.00	3
breakfast	1.00	1.00	1.00	2
bright	1.00	1.00	1.00	1
buy	1.00	0.67	0.80	3
call	1.00	1.00	1.00	3
candy	0.67	1.00	0.80	2
catch	0.00	0.00	0.00	1
accuracy			0.91	32
macro avg	0.83	0.81	0.81	32
weighted avg	0.92	0.91	0.90	32

Figure 28 Frame Sequence-Model 2 Classification report on LSA64 dataset.

WLASL Dataset

The model was trained for 20 epochs using this dataset. Based on the model performance, overfitting was obvious. The training accuracy and validation improves dramatically and reaches its peak at epoch 3 then flatten throughout the entire training session. On the other hand, despite some small fluctuations in validation accuracy, the curve was flatten throughout the 20 epochs. On the other hand, the validation increases gradually throughout the 20 epochs. It is suggested that the model fail to generalize to unseen data.

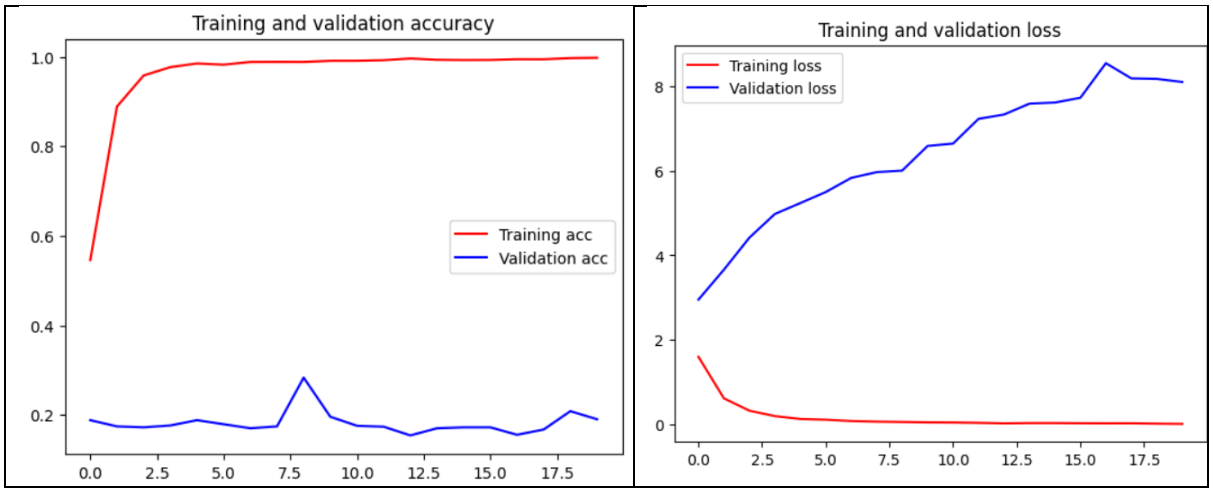

Figure 29 Frame Sequence-Model 2 Training, Validation Accuracy and Loss on WLASL dataset

Test set

The model achieved 13.3% accuracy and 7.2542 loss on test set. Based on the confusion matrix, the model only managed to recognize all instances of 2 out of 15 signs correctly. Based on the classification report, the model achieved 0.13 f1-score, precision and recall respectively. . In general, of the positive classification the model made with respective to each of these classes, 13% was truly positive. On the other hand, the model has correctly predicted 13% of the actual positive instances of the classes.

```
1/1 [==============================] - 0s 147ms/step - loss: 7.2542 - accuracy: 0.1333
```

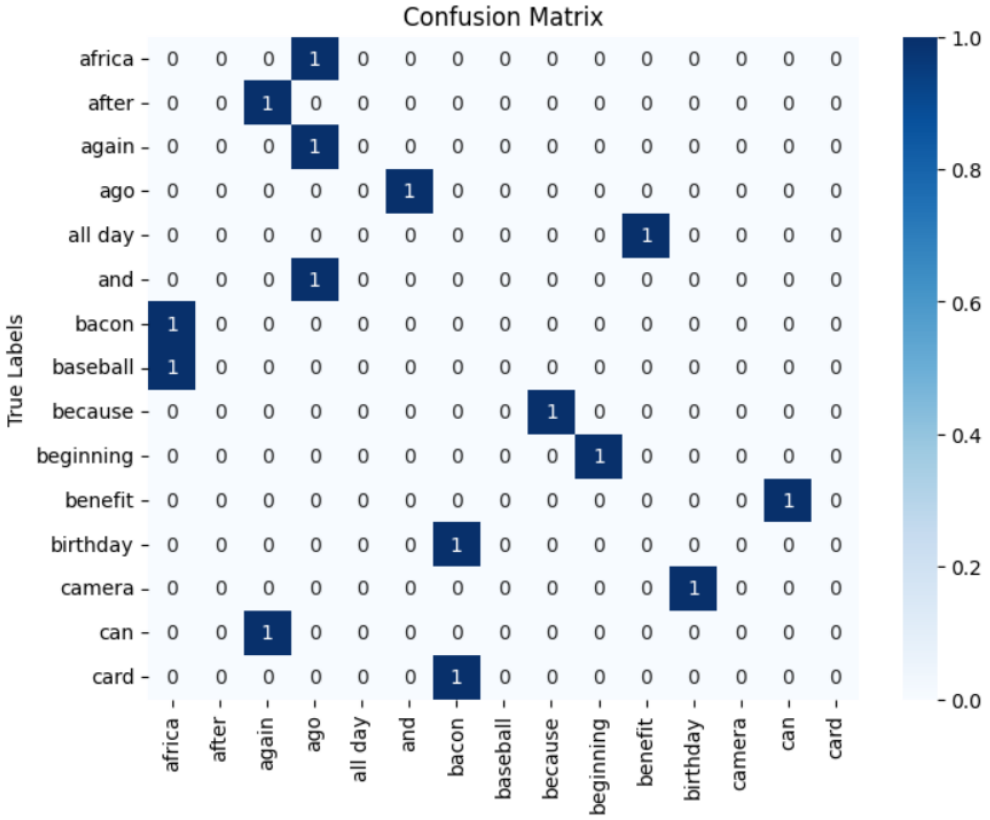

Figure 30 Frame Sequence--Model 2Confusion Matrix on WLASL dataset

	precision	recall	f1-score	support
africa	0.00	0.00	0.00	1
after	0.00	0.00	0.00	1
again	0.00	0.00	0.00	1
ago	0.00	0.00	0.00	1
all day	0.00	0.00	0.00	1
and	0.00	0.00	0.00	1
bacon	0.00	0.00	0.00	1
baseball	0.00	0.00	0.00	1
because	1.00	1.00	1.00	1
beginning	1.00	1.00	1.00	1
benefit	0.00	0.00	0.00	1
birthday	0.00	0.00	0.00	1
camera	0.00	0.00	0.00	1
can	0.00	0.00	0.00	1
card	0.00	0.00	0.00	1
accuracy			0.13	15
macro avg	0.13	0.13	0.13	15
weighted avg	0.13	0.13	0.13	15

Figure 31 Frame Sequence--Model 2 Classification Report on WLASL dataset