

## Project Development Phase

### Model Performance Test

Date	14 November 2022
Team ID	PNT2022TMID29664
Project Name	Real time communication powered by AI for specially abled
Maximum Marks	10 Marks

### PERFORMANCE METRICS:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values																					
1.	Model Summary	<div>-Model: "sequential"</div> <div><table><thead><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr></thead><tbody><tr><td>conv2d (Conv2D)</td><td>(None, 62, 62, 32)</td><td>896</td></tr><tr><td>max_pooling2d (MaxPooling2D)</td><td>(None, 31, 31, 32)</td><td>0</td></tr><tr><td>flatten (Flatten)</td><td>(None, 30752)</td><td>0</td></tr><tr><td>dense (Dense)</td><td>(None, 200)</td><td>6150600</td></tr><tr><td>dense_1 (Dense)</td><td>(None, 200)</td><td>40200</td></tr><tr><td>dense_2 (Dense)</td><td>(None, 9)</td><td>1809</td></tr></tbody></table></div> <div>Total params: 6,193,505 Trainable params: 6,193,505 Non-trainable params: 0</div>	Layer (type)	Output Shape	Param #	conv2d (Conv2D)	(None, 62, 62, 32)	896	max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0	flatten (Flatten)	(None, 30752)	0	dense (Dense)	(None, 200)	6150600	dense_1 (Dense)	(None, 200)	40200	dense_2 (Dense)	(None, 9)	1809
Layer (type)	Output Shape	Param #																					
conv2d (Conv2D)	(None, 62, 62, 32)	896																					
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0																					
flatten (Flatten)	(None, 30752)	0																					
dense (Dense)	(None, 200)	6150600																					
dense_1 (Dense)	(None, 200)	40200																					
dense_2 (Dense)	(None, 9)	1809																					

2.	Accuracy	<p>Training Accuracy - 0.9942</p> <p>Validation Accuracy - 0.9778</p>
----	----------	---

## MODEL SUMMARY:

- Each layer has an output and its shape is shown in the “Output Shape” column. Each layer’s output becomes the input for the subsequent layer.
- The “Param #” column shows you the number of parameters that are trained for each layer.
- The total number of parameters is shown at the end, which is equal to the number of trainable and non-trainable parameters. In this model, all the layers are trainable.

## Accuracy:

- "loss" refers to the loss value over the training data after each epoch. This is what the optimization process is trying to minimize with the training so, the lower, the better.

"accuracy" refers to the ratio between correct predictions and the total number of predictions in the training data. The higher, the better. This is normally inversely correlated with the loss.

## accuracy image:

```

In [18]: model.fit(x_train, epochs=15, validation_data=x_test, steps_per_epoch=len(x_train), validation_steps=len(x_test))

Epoch 1/15
525/525 [=====] - 105s 192ms/step - loss: 0.3236 - accuracy: 0.8910 - val_loss: 0.3572 - val_accuracy: 0.9551
Epoch 2/15
525/525 [=====] - 91s 173ms/step - loss: 0.0708 - accuracy: 0.9767 - val_loss: 0.3196 - val_accuracy: 0.9658
Epoch 3/15
525/525 [=====] - 90s 172ms/step - loss: 0.0363 - accuracy: 0.9888 - val_loss: 0.4368 - val_accuracy: 0.9640
Epoch 4/15
525/525 [=====] - 89s 169ms/step - loss: 0.0304 - accuracy: 0.9908 - val_loss: 0.2606 - val_accuracy: 0.9649
Epoch 5/15
525/525 [=====] - 90s 171ms/step - loss: 0.0289 - accuracy: 0.9912 - val_loss: 0.3375 - val_accuracy: 0.9751
Epoch 6/15
525/525 [=====] - 88s 168ms/step - loss: 0.0148 - accuracy: 0.9947 - val_loss: 0.3050 - val_accuracy: 0.9818
Epoch 7/15
525/525 [=====] - 89s 170ms/step - loss: 0.0216 - accuracy: 0.9933 - val_loss: 0.2263 - val_accuracy: 0.9831
Epoch 8/15
525/525 [=====] - 88s 168ms/step - loss: 0.0194 - accuracy: 0.9945 - val_loss: 0.3935 - val_accuracy: 0.9729
Epoch 9/15
525/525 [=====] - 89s 170ms/step - loss: 0.0131 - accuracy: 0.9961 - val_loss: 0.3293 - val_accuracy: 0.9800
Epoch 10/15
525/525 [=====] - 89s 170ms/step - loss: 0.0109 - accuracy: 0.9965 - val_loss: 0.4748 - val_accuracy: 0.9707
Epoch 11/15
525/525 [=====] - 91s 173ms/step - loss: 0.0122 - accuracy: 0.9964 - val_loss: 0.4442 - val_accuracy: 0.9787
Epoch 12/15

```

The screenshot shows a Jupyter Notebook titled "Test The Model (2)" with a last checkpoint from Sunday at 8:54 PM. The notebook is running Python 3 (ipykernel). The code cell displays the training progress for epochs 7 through 15. Each epoch's output includes the time taken, loss, accuracy, validation loss, and validation accuracy. The training is using a sequential model with a convolutional layer, max pooling, and dense layers.

```
y: 0.9618
Epoch 7/15
525/525 [=====] - 89s 170ms/step - loss: 0.0216 - accuracy: 0.9933 - val_loss: 0.2263 - val_accuracy: 0.9831
y: 0.9831
Epoch 8/15
525/525 [=====] - 88s 168ms/step - loss: 0.0194 - accuracy: 0.9945 - val_loss: 0.3935 - val_accuracy: 0.9729
y: 0.9729
Epoch 9/15
525/525 [=====] - 89s 170ms/step - loss: 0.0131 - accuracy: 0.9961 - val_loss: 0.3293 - val_accuracy: 0.9800
y: 0.9800
Epoch 10/15
525/525 [=====] - 89s 170ms/step - loss: 0.0109 - accuracy: 0.9965 - val_loss: 0.4748 - val_accuracy: 0.9707
y: 0.9707
Epoch 11/15
525/525 [=====] - 91s 173ms/step - loss: 0.0122 - accuracy: 0.9964 - val_loss: 0.4442 - val_accuracy: 0.9787
y: 0.9787
Epoch 12/15
525/525 [=====] - 90s 171ms/step - loss: 0.0101 - accuracy: 0.9970 - val_loss: 0.1999 - val_accuracy: 0.9693
y: 0.9693
Epoch 13/15
525/525 [=====] - 93s 177ms/step - loss: 0.0091 - accuracy: 0.9971 - val_loss: 0.4493 - val_accuracy: 0.9804
y: 0.9804
Epoch 14/15
525/525 [=====] - 96s 183ms/step - loss: 0.0129 - accuracy: 0.9963 - val_loss: 0.4011 - val_accuracy: 0.9613
y: 0.9613
Epoch 15/15
525/525 [=====] - 92s 175ms/step - loss: 0.0179 - accuracy: 0.9942 - val_loss: 0.5153 - val_accuracy: 0.9778
y: 0.9778

Out[18]: <keras.callbacks.History at 0x1b1cc0be50>

In [27]: model.save('C:/Users/welcome/Downloads/sign-language-recognition-project/signlanguage-new.h5')
```

Model summary image:

The screenshot shows the same Jupyter Notebook, but now displaying the model summary. The code cell shows the output of the `model.summary()` method, which provides a detailed overview of the model's architecture, including the number of layers, output shapes, and the total number of parameters.

```
Epoch 14/15
525/525 [=====] - 96s 183ms/step - loss: 0.0129 - accuracy: 0.9963 - val_loss: 0.4011 - val_accuracy: 0.9613
y: 0.9613
Epoch 15/15
525/525 [=====] - 92s 175ms/step - loss: 0.0179 - accuracy: 0.9942 - val_loss: 0.5153 - val_accuracy: 0.9778
y: 0.9778

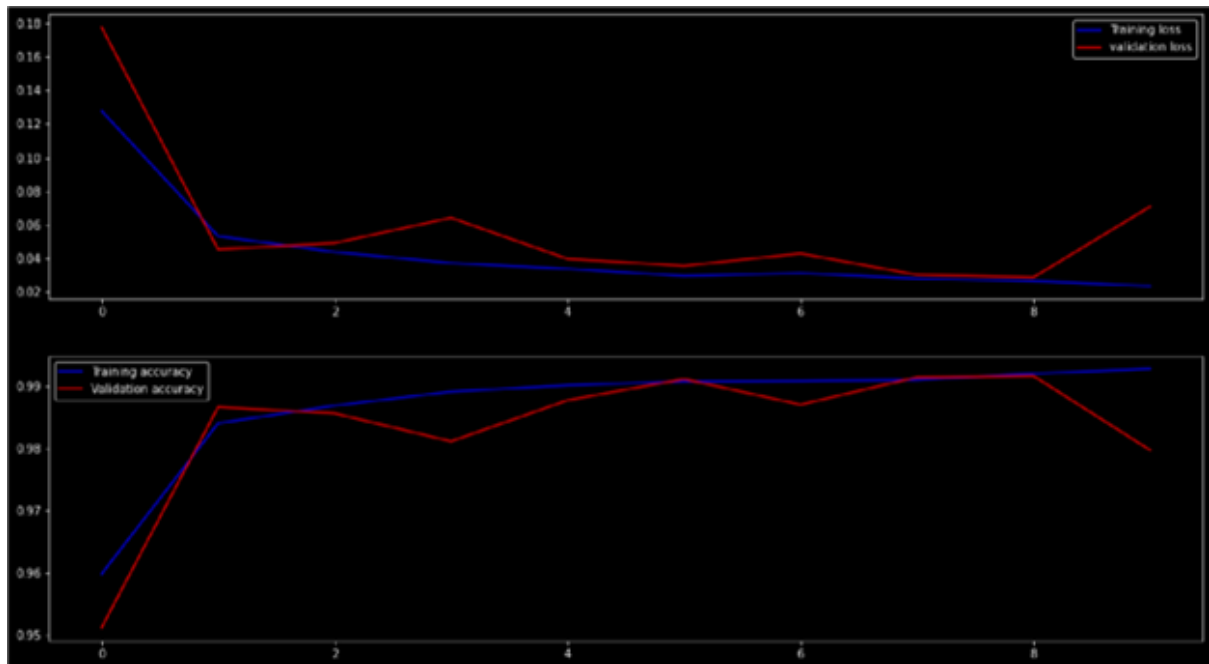
Out[18]: <keras.callbacks.History at 0x1b1cc0be50>

In [27]: model.save('C:/Users/welcome/Downloads/sign-language-recognition-project/signlanguage-new.h5')

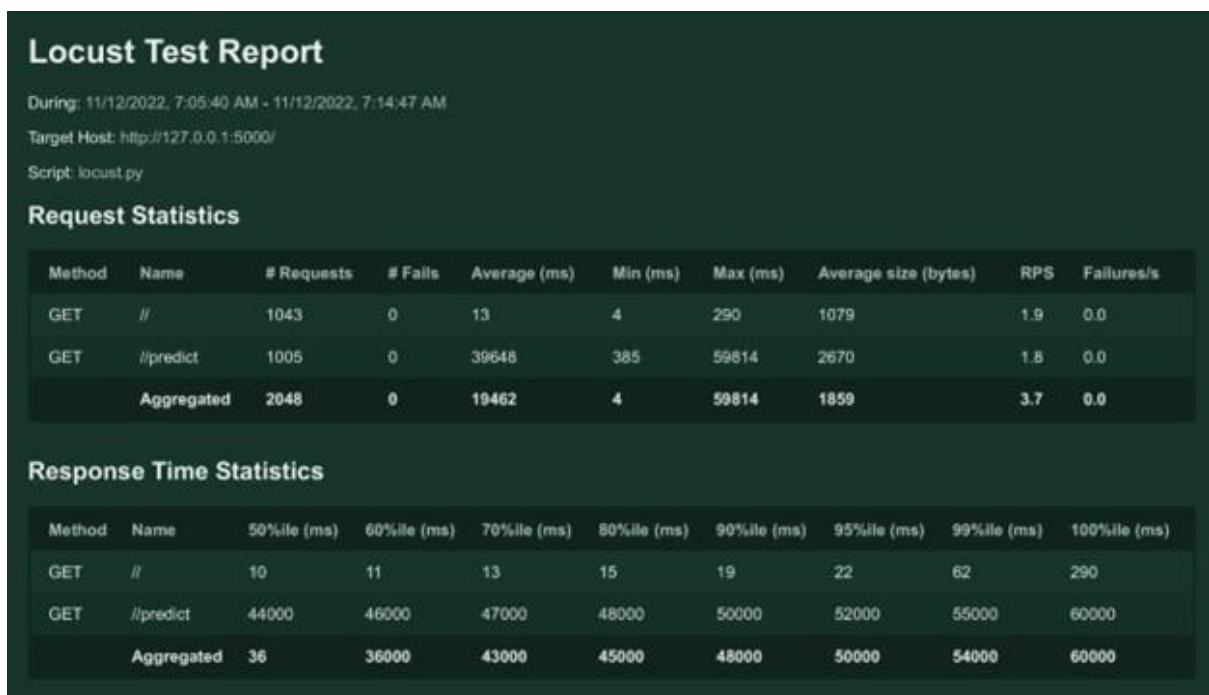
In [19]: model.summary()

Model: "sequential"
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)              (None, 62, 62, 32)        896
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)         0
flatten (Flatten)            (None, 30752)              0
dense (Dense)                 (None, 200)                6150600
dense_1 (Dense)               (None, 200)                40200
dense_2 (Dense)               (None, 9)                  1809
Total params: 6,193,505
Trainable params: 6,193,505
Non-trainable params: 0
```

## ACCURACY TESTING:



## PERFORMANCE TESTING USING LOCUST TEST :

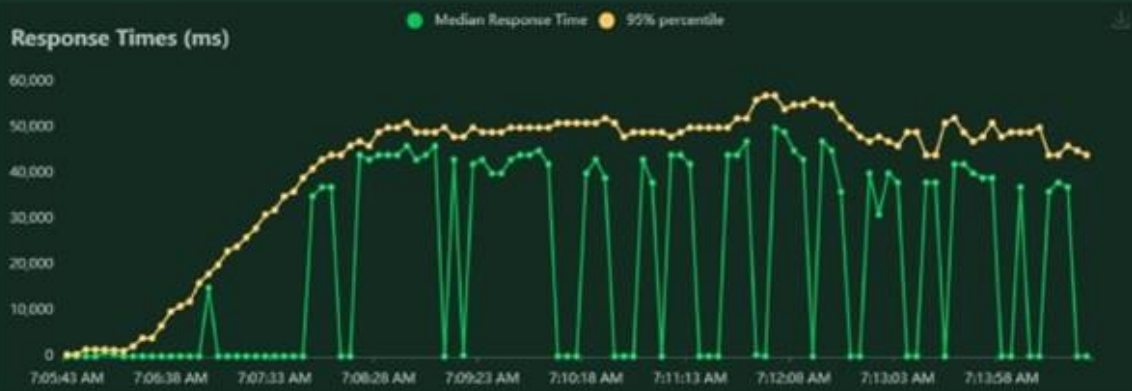


## Charts

### Total Requests per Second



### Response Times (ms)



### Number of Users

