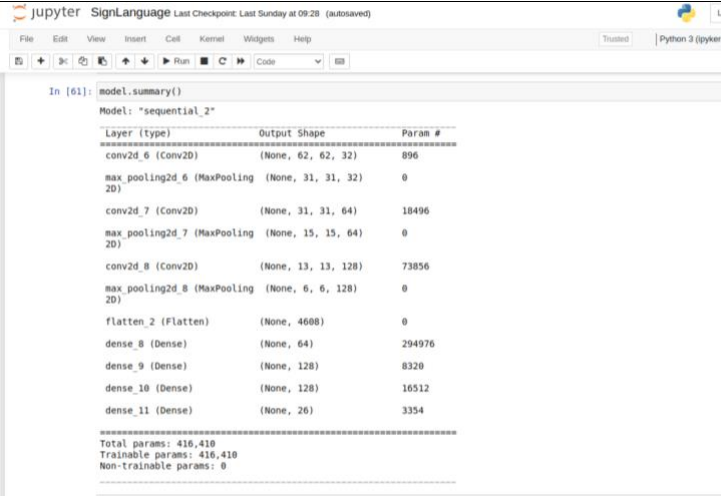
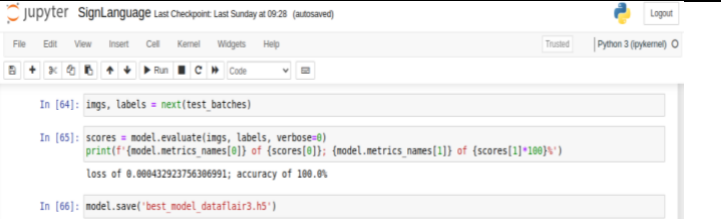


Project Development Phase Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID30205
Project Name	Project – Real-Time Communication System Powered by AI for Specially Abled
Maximum Marks	10 Marks

Model Performance Testing:

S.No	Parameter	Values	Screenshot
1.	Model Summary	Total params:416, 410 Trainable params:416, 410 Non- Trainable params:0	 <pre> In [61]: model.summary() Model: "sequential_2" ----- Layer (type) Output Shape Param # ----- conv2d_6 (Conv2D) (None, 62, 62, 32) 896 max_pooling2d_6 (MaxPooling2D) (None, 31, 31, 32) 0 conv2d_7 (Conv2D) (None, 31, 31, 64) 18496 max_pooling2d_7 (MaxPooling2D) (None, 15, 15, 64) 0 conv2d_8 (Conv2D) (None, 13, 13, 128) 73856 max_pooling2d_8 (MaxPooling2D) (None, 6, 6, 128) 0 flatten_2 (Flatten) (None, 4608) 0 dense_8 (Dense) (None, 64) 294976 dense_9 (Dense) (None, 128) 8320 dense_10 (Dense) (None, 128) 16512 dense_11 (Dense) (None, 20) 3354 ----- Total params: 416,410 Trainable params: 416,410 Non-trainable params: 0 </pre>
2.	Accuracy	Training Accuracy – 100.0%	 <pre> In [64]: imgs, labels = next(test_batches) In [65]: scores = model.evaluate(imgs, labels, verbose=0) print(f'{model.metrics_names[0]} of {scores[0]}; {model.metrics_names[1]} of {scores[1]*100}%') loss of 0.0004329237563869991; accuracy of 100.0% In [66]: model.save('best_model_dataflair3.h5') </pre>

Validation Accuracy – 1.0

```
In [8]: model.compile(optimizer=SGD(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
        reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.2, patience=1, min_lr=0.0005)
        early_stop = EarlyStopping(monitor='val_loss', min_delta=0, patience=2, verbose=0, mode='auto')

        model.compile(optimizer=adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
        reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.2, patience=1, min_lr=0.0005)
        early_stop = EarlyStopping(monitor='val_loss', min_delta=0, patience=2, verbose=0, mode='auto')

In [9]: history2 = model.fit(train_batches, epochs=10, callbacks=[reduce_lr, early_stop], validation_data = test_batches)

Epoch 1/10
3111/3111 [=====] - 76s 24ms/step - loss: 0.1676 - accuracy: 0.9638 - val_loss: 0.0613 - val accuracy: 0.9863 - lr: 0.0010
Epoch 2/10
3111/3111 [=====] - 73s 24ms/step - loss: 0.0426 - accuracy: 0.9929 - val_loss: 1.6384e-05 - val accuracy: 1.0000 - lr: 0.0010
Epoch 3/10
3111/3111 [=====] - 76s 24ms/step - loss: 0.0218 - accuracy: 0.9966 - val_loss: 1.1261e-06 - val accuracy: 1.0000 - lr: 0.0010
Epoch 4/10
3111/3111 [=====] - 75s 24ms/step - loss: 6.0457e-05 - accuracy: 1.0000 - val_loss: 0.0001 - val accuracy: 0.9959 - lr: 2.0000e-04
Epoch 5/10
3111/3111 [=====] - 73s 23ms/step - loss: 3.1256e-04 - accuracy: 0.9999 - val_loss: 1.2166e-07 - val accuracy: 1.0000 - lr: 1.0000e-04
Epoch 6/10
3111/3111 [=====] - 73s 23ms/step - loss: 3.4253e-07 - accuracy: 1.0000 - val_loss: 4.2295e-08 - val accuracy: 1.0000 - lr: 1.0000e-04
Epoch 7/10
3111/3111 [=====] - 74s 24ms/step - loss: 1.0593e-07 - accuracy: 1.0000 - val_loss: 9.9613e-09 - val accuracy: 1.0000 - lr: 1.0000e-04
Epoch 8/10
3111/3111 [=====] - 72s 23ms/step - loss: 2.5110e-08 - accuracy: 1.0000 - val_loss: 9.7900e-10 - val accuracy: 1.0000 - lr: 1.0000e-04
Epoch 9/10
3111/3111 [=====] - 73s 23ms/step - loss: 6.1941e-09 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val accuracy: 1.0000 - lr: 1.0000e-04
Epoch 10/10
3111/3111 [=====] - 75s 24ms/step - loss: 1.7900e-09 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val accuracy: 1.0000 - lr: 1.0000e-04

In [10]: imgs, labels = next(test_batches)
```

