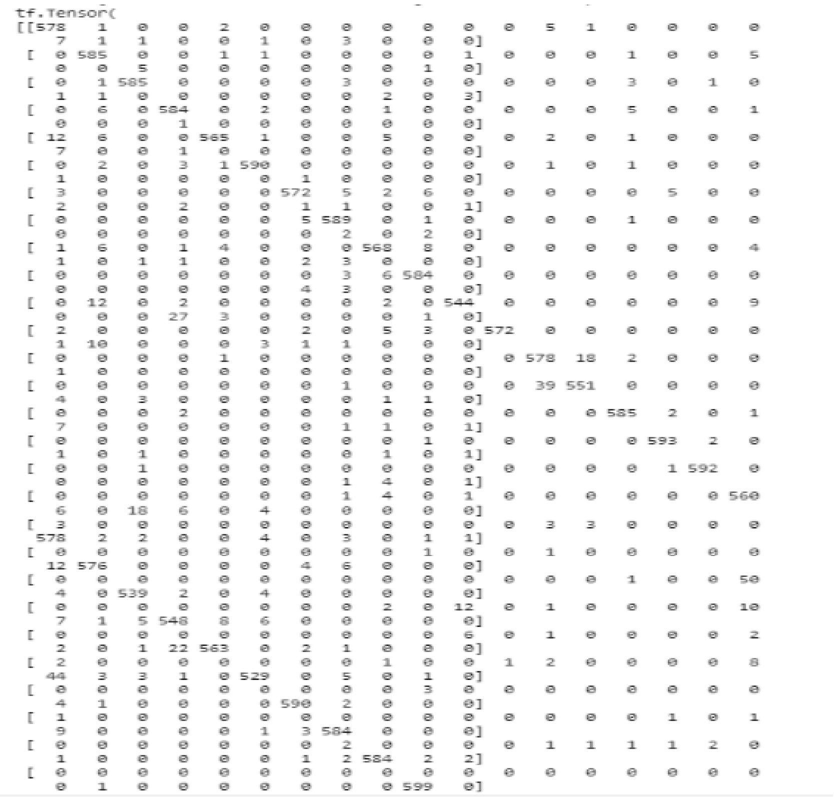
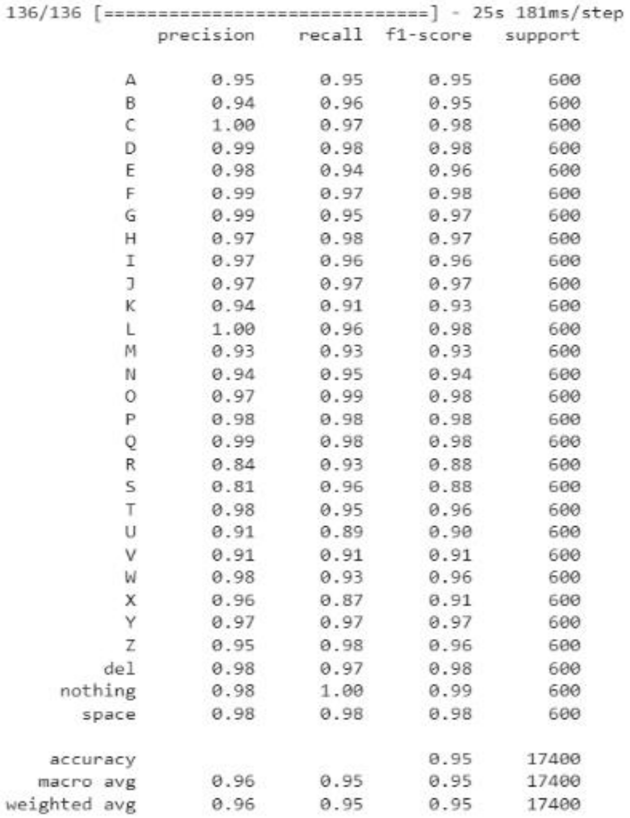
# PERFORMANCE AND FINAL SUBMISSION PHRASE

| | |
|---|---|
| Date | 21 November 2023 |
| Team ID | Team- 592184 |
| Project Name | ASL - Alphabet image recognition |
| Maximum Marks | 10 Marks |

**Model Performance Testing:**

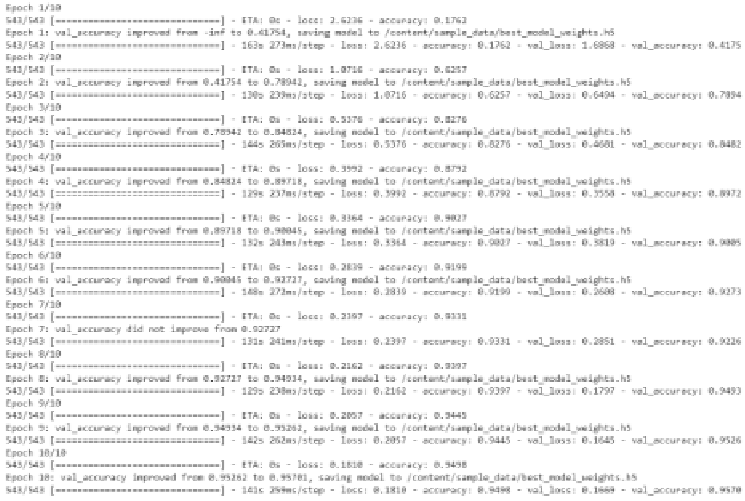
| S.No. | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Metrics | **Classification Model:**<br>VGG16 model<br>**Confusion Matrix , Classification Report &**<br>**Accuracy Scores-**<br>Training accuracy:- 94.98%<br>Testing accuracy:- 95.03% | Confusion Matrix:-<br><br>Classification report:- |

```
tf.Tensor(
[[578   1   0   0   2   0   0   0   0   0   0   0   5   1   0   0   0   0
    7   1   1   0   0   1   0   3   0   0   0]
 [  0 585   0   0   1   1   0   0   0   0   1   0   0   0   1   0   0   5
    0   0   5   0   0   0   0   0   0   1   0]
 [  0   1 585   0   0   0   0   3   0   0   0   0   0   0   3   0   1   0
    1   1   0   0   0   0   0   0   2   0   3]
 [  0   6   0 584   0   2   0   0   1   0   0   0   0   0   5   0   0   1
    0   0   0   1   0   0   0   0   0   0   0]
 [ 12   6   0   0 565   1   0   0   5   0   0   0   2   0   1   0   0   0
    7   0   0   1   0   0   0   0   0   0   0]
 [  0   2   0   3   1 590   0   0   0   0   0   0   1   0   1   0   0   0
    1   0   0   0   0   0   1   0   0   0   0]
 [  3   0   0   0   0   0 572   5   2   6   0   0   0   0   0   5   0   0
    2   0   0   2   0   0   1   1   0   0   1]
 [  0   0   0   0   0   0   5 589   0   1   0   0   0   0   1   0   0   0
    0   0   0   0   0   0   0   2   0   2   0]
 [  1   6   0   1   4   0   0   0 568   8   0   0   0   0   0   0   0   4
    1   0   1   1   0   0   2   3   0   0   0]
 [  0   0   0   0   0   0   0   3   6 584   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   4   3   0   0   0]
 [  0  12   0   2   0   0   0   0   2   0 544   0   0   0   0   0   0   9
    0   0   0  27   3   0   0   0   0   1   0]
 [  2   0   0   0   0   0   2   0   5   3   0 572   0   0   0   0   0   0
    1  10   0   0   0   3   1   1   0   0   0]
 [  0   0   0   0   1   0   0   0   0   0   0   0 578  18   2   0   0   0
    1   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   1   0   0   0   0  39 551   0   0   0   0
    4   0   3   0   0   0   0   0   1   1   0]
 [  0   0   0   2   0   0   0   0   0   0   0   0   0   0 585   2   0   1
    7   0   0   0   0   0   0   1   1   0   1]
 [  0   0   0   0   0   0   0   0   0   1   0   0   0   0   0 593   2   0
    1   0   1   0   0   0   0   0   1   0   1]
 [  0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   1 592   0
    0   0   0   0   0   0   0   1   4   0   1]
 [  0   0   0   0   0   0   0   1   4   0   1   0   0   0   0   0   0 560
    6   0  18   6   0   4   0   0   0   0   0]
 [  3   0   0   0   0   0   0   0   0   0   0   0   3   3   0   0   0   0
  578   2   2   0   0   4   0   3   0   1   1]
 [  0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0
   12 576   0   0   0   0   4   6   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0  50
    4   0 539   2   0   4   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   2   0  12   0   1   0   0   0   0  10
    7   1   5 548   8   6   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   6   0   1   0   0   0   0   2
    2   0   1  22 563   0   2   1   0   0   0]
 [  2   0   0   0   0   0   0   0   1   0   0   1   2   0   0   0   0   8
   44   3   3   1   0 529   0   5   0   1   0]
 [  0   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0
    4   1   0   0   0   0 590   2   0   0   0]
 [  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1
    9   0   0   0   0   1   3 584   0   0   0]
 [  0   0   0   0   0   0   0   2   0   0   0   0   1   1   1   1   2   0
    1   0   0   0   0   0   1   2 584   2   2]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   1   0   0   0   0   0   0   0 599   0]
```

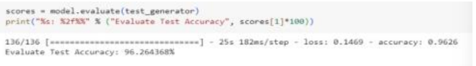


```
136/136 [==============================] - 25s 181ms/step
              precision    recall  f1-score   support

           A       0.95      0.95      0.95       600
           B       0.94      0.96      0.95       600
           C       1.00      0.97      0.98       600
           D       0.99      0.98      0.98       600
           E       0.98      0.94      0.96       600
           F       0.99      0.97      0.98       600
           G       0.99      0.95      0.97       600
           H       0.97      0.98      0.97       600
           I       0.97      0.96      0.96       600
           J       0.97      0.97      0.97       600
           K       0.94      0.91      0.93       600
           L       1.00      0.96      0.98       600
           M       0.93      0.93      0.93       600
           N       0.94      0.95      0.94       600
           O       0.97      0.99      0.98       600
           P       0.98      0.98      0.98       600
           Q       0.99      0.98      0.98       600
           R       0.84      0.93      0.88       600
           S       0.81      0.96      0.88       600
           T       0.98      0.95      0.96       600
           U       0.91      0.89      0.90       600
           V       0.91      0.91      0.91       600
           W       0.98      0.93      0.96       600
           X       0.96      0.87      0.91       600
           Y       0.97      0.97      0.97       600
           Z       0.95      0.98      0.96       600
         del       0.98      0.97      0.98       600
     nothing       0.98      1.00      0.99       600
       space       0.98      0.98      0.98       600

    accuracy                           0.95     17400
   macro avg       0.96      0.95      0.95     17400
weighted avg       0.96      0.95      0.95     17400
```

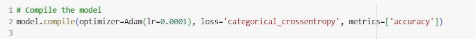
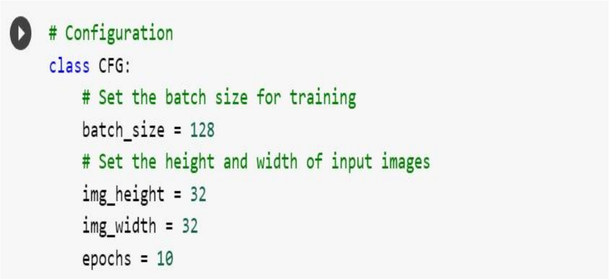
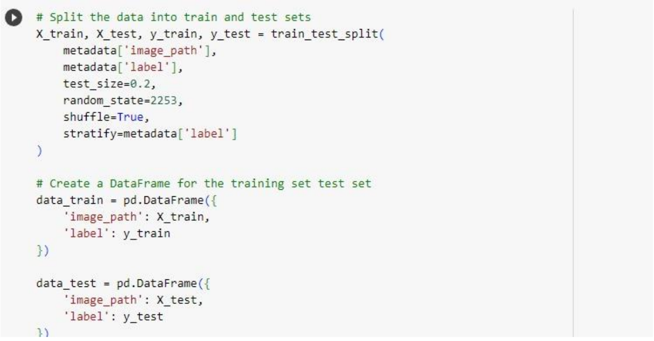
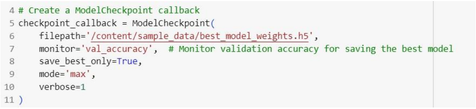
Accuracy Score:-



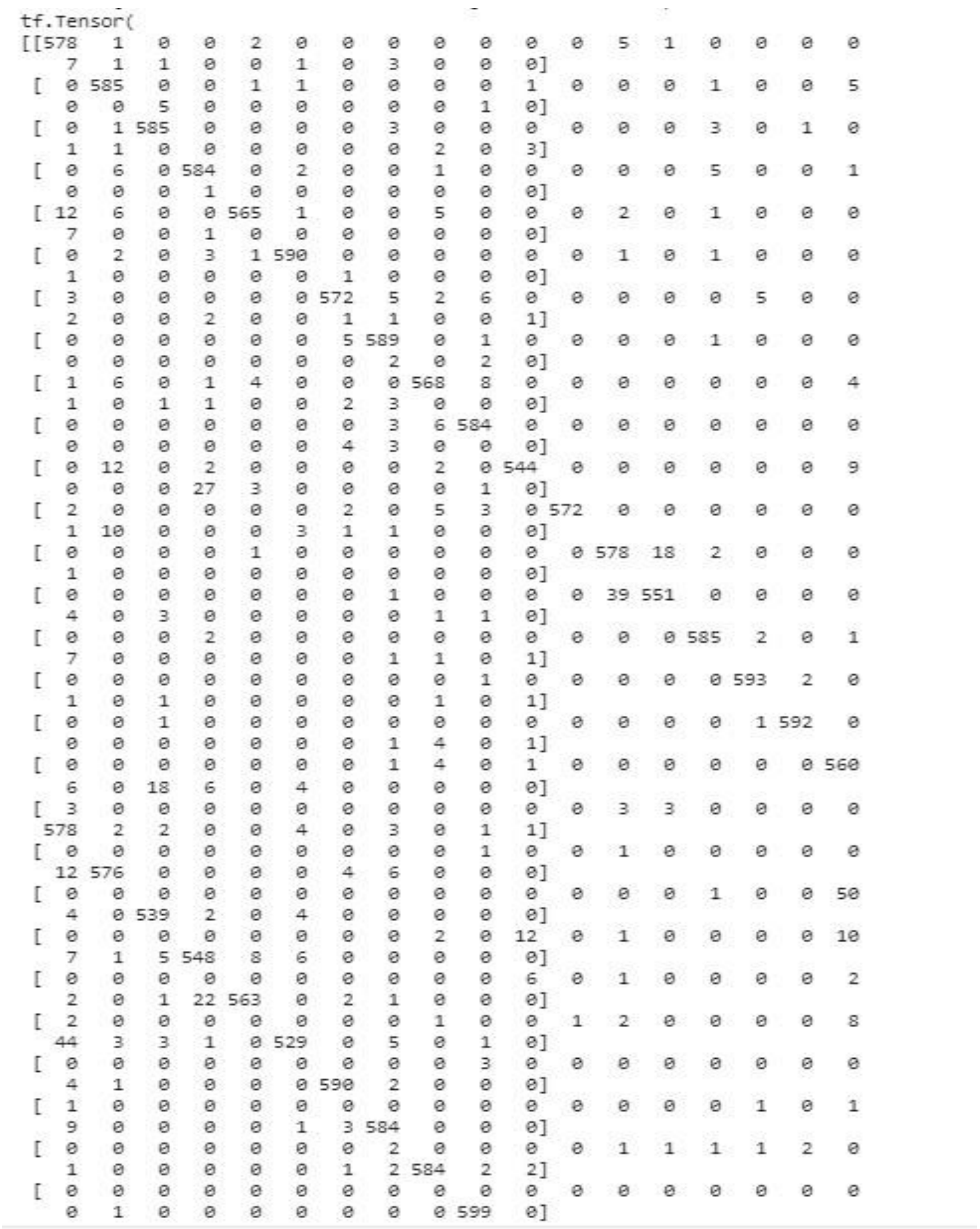
```
Epoch 1/10
543/543 [==============================] - ETA: 0s - loss: 2.6236 - accuracy: 0.1762
Epoch 1: val_accuracy improved from -inf to 0.41754, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 163s 273ms/step - loss: 2.6236 - accuracy: 0.1762 - val_loss: 1.6868 - val_accuracy: 0.4175
Epoch 2/10
543/543 [==============================] - ETA: 0s - loss: 1.0716 - accuracy: 0.6257
Epoch 2: val_accuracy improved from 0.41754 to 0.78942, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 130s 239ms/step - loss: 1.0716 - accuracy: 0.6257 - val_loss: 0.6494 - val_accuracy: 0.7894
Epoch 3/10
543/543 [==============================] - ETA: 0s - loss: 0.5376 - accuracy: 0.8276
Epoch 3: val_accuracy improved from 0.78942 to 0.84824, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 144s 265ms/step - loss: 0.5376 - accuracy: 0.8276 - val_loss: 0.4681 - val_accuracy: 0.8482
Epoch 4/10
543/543 [==============================] - ETA: 0s - loss: 0.3992 - accuracy: 0.8792
Epoch 4: val_accuracy improved from 0.84824 to 0.89718, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 129s 237ms/step - loss: 0.3992 - accuracy: 0.8792 - val_loss: 0.3558 - val_accuracy: 0.8972
Epoch 5/10
543/543 [==============================] - ETA: 0s - loss: 0.3364 - accuracy: 0.9027
Epoch 5: val_accuracy improved from 0.89718 to 0.90045, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 132s 243ms/step - loss: 0.3364 - accuracy: 0.9027 - val_loss: 0.3819 - val_accuracy: 0.9005
Epoch 6/10
543/543 [==============================] - ETA: 0s - loss: 0.2839 - accuracy: 0.9199
Epoch 6: val_accuracy improved from 0.90045 to 0.92727, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 148s 272ms/step - loss: 0.2839 - accuracy: 0.9199 - val_loss: 0.2608 - val_accuracy: 0.9273
Epoch 7/10
543/543 [==============================] - ETA: 0s - loss: 0.2397 - accuracy: 0.9331
Epoch 7: val_accuracy did not improve from 0.92727
543/543 [==============================] - 131s 241ms/step - loss: 0.2397 - accuracy: 0.9331 - val_loss: 0.2851 - val_accuracy: 0.9226
Epoch 8/10
543/543 [==============================] - ETA: 0s - loss: 0.2162 - accuracy: 0.9397
Epoch 8: val_accuracy improved from 0.92727 to 0.94934, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 129s 238ms/step - loss: 0.2162 - accuracy: 0.9397 - val_loss: 0.1797 - val_accuracy: 0.9493
Epoch 9/10
543/543 [==============================] - ETA: 0s - loss: 0.2057 - accuracy: 0.9445
Epoch 9: val_accuracy improved from 0.94934 to 0.95262, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 142s 262ms/step - loss: 0.2057 - accuracy: 0.9445 - val_loss: 0.1645 - val_accuracy: 0.9526
Epoch 10/10
543/543 [==============================] - ETA: 0s - loss: 0.1810 - accuracy: 0.9498
Epoch 10: val_accuracy improved from 0.95262 to 0.95701, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 141s 259ms/step - loss: 0.1810 - accuracy: 0.9498 - val_loss: 0.1669 - val_accuracy: 0.9570
```



```
scores = model.evaluate(test_generator)
print("%s: %2f%%" % ("Evaluate Test Accuracy", scores[1]*100))

136/136 [==============================] - 25s 182ms/step - loss: 0.1469 - accuracy: 0.9626
Evaluate Test Accuracy: 96.264368%
```

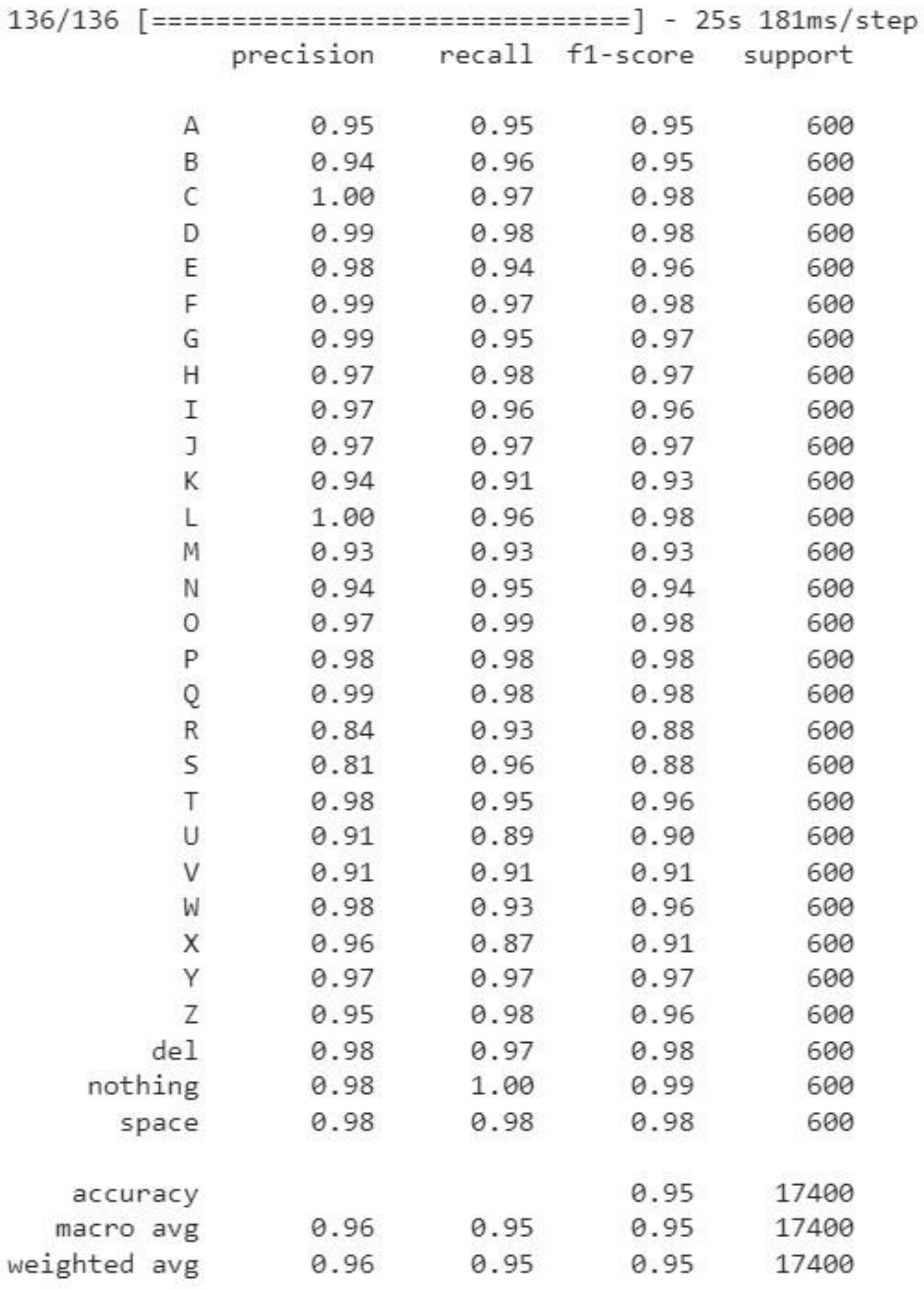
| | | | |
|---|---|---|---|
| 2. | Tune the Model | **Hyperparameter Tuning:**<br>The model is tuned with following hyper parametersOptimizer - Adam<br>Learning rate - 0.0001<br>Loss - Categorical cross entropy<br>Batch size - 128<br>EPOCHS - 10<br><br>**Validation Method:**<br>The validation of the model is done through the validation data , which is set to 20% of training data. Data augmentation and callbacks are also used to validate performance. Accuracy is the validation parameter that we have monitored | Hyperparameter Tuning:-<br><br># Compile the model<br>model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])<br><br># Configuration<br>class CFG:<br>    # Set the batch size for training<br>    batch_size = 128<br>    # Set the height and width of input images<br>    img_height = 32<br>    img_width = 32<br>    epochs = 10<br><br>Validation Method:-<br><br># Split the data into train and test sets<br>X_train, X_test, y_train, y_test = train_test_split(<br>    metadata['image_path'],<br>    metadata['label'],<br>    test_size=0.2,<br>    random_state=2253,<br>    shuffle=True,<br>    stratify=metadata['label']<br>)<br><br># Create a DataFrame for the training set test set<br>data_train = pd.DataFrame({<br>    'image_path': X_train,<br>    'label': y_train<br>})<br><br>data_test = pd.DataFrame({<br>    'image_path': X_test,<br>    'label': y_test<br>})<br><br># Create a ModelCheckpoint callback<br>checkpoint_callback = ModelCheckpoint(<br>    filepath='/content/sample_data/best_model_weights.h5',<br>    monitor='val_accuracy',  # Monitor validation accuracy for saving the best model<br>    save_best_only=True,<br>    mode='max',<br>    verbose=1<br>) |

**CONFUSION MATRIX:-**



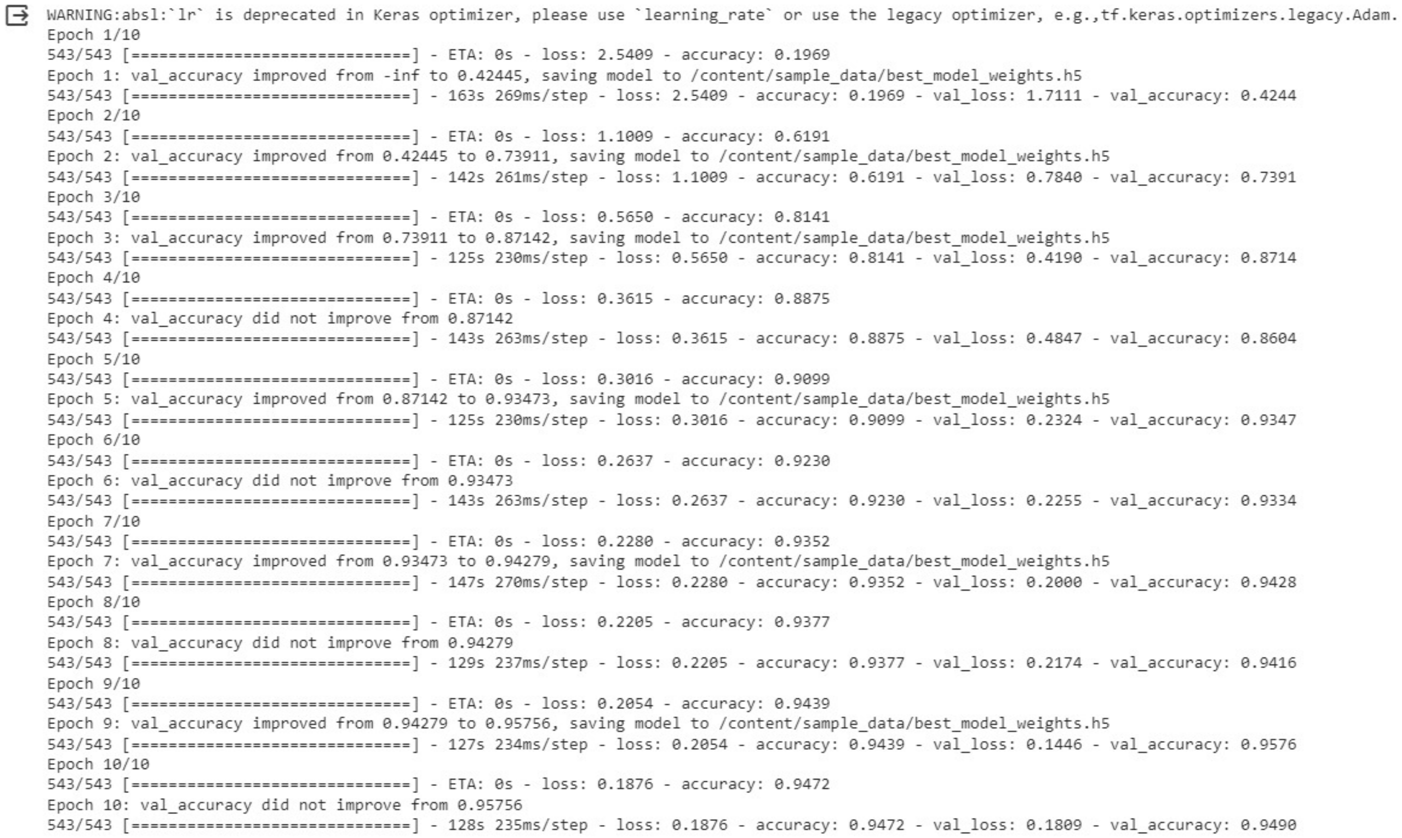
```
tf.Tensor(
[[578   1   0   0   2   0   0   0   0   0   0   0   5   1   0   0   0   0
    7   1   1   0   0   1   0   3   0   0   0]
 [  0 585   0   0   1   1   0   0   0   0   1   0   0   0   1   0   0   5
    0   0   5   0   0   0   0   0   0   1   0]
 [  0   1 585   0   0   0   0   3   0   0   0   0   0   0   3   0   1   0
    1   1   0   0   0   0   0   0   2   0   3]
 [  0   6   0 584   0   2   0   0   1   0   0   0   0   0   5   0   0   1
    0   0   0   1   0   0   0   0   0   0   0]
 [ 12   6   0   0 565   1   0   0   5   0   0   0   2   0   1   0   0   0
    7   0   0   1   0   0   0   0   0   0   0]
 [  0   2   0   3   1 590   0   0   0   0   0   0   1   0   1   0   0   0
    1   0   0   0   0   0   1   0   0   0   0]
 [  3   0   0   0   0   0 572   5   2   6   0   0   0   0   0   5   0   0
    2   0   0   2   0   0   1   1   0   0   1]
 [  0   0   0   0   0   0   5 589   0   1   0   0   0   0   1   0   0   0
    0   0   0   0   0   0   0   2   0   2   0]
 [  1   6   0   1   4   0   0   0 568   8   0   0   0   0   0   0   0   4
    1   0   1   1   0   0   2   3   0   0   0]
 [  0   0   0   0   0   0   0   3   6 584   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   4   3   0   0   0]
 [  0  12   0   2   0   0   0   0   2   0 544   0   0   0   0   0   0   9
    0   0   0  27   3   0   0   0   0   1   0]
 [  2   0   0   0   0   0   2   0   5   3   0 572   0   0   0   0   0   0
    1  10   0   0   0   3   1   1   0   0   0]
 [  0   0   0   0   1   0   0   0   0   0   0   0 578  18   2   0   0   0
    1   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   1   0   0   0   0  39 551   0   0   0   0
    4   0   3   0   0   0   0   0   1   1   0]
 [  0   0   0   2   0   0   0   0   0   0   0   0   0   0 585   2   0   1
    7   0   0   0   0   0   0   1   1   0   1]
 [  0   0   0   0   0   0   0   0   0   1   0   0   0   0   0 593   2   0
    1   0   1   0   0   0   0   0   1   0   1]
 [  0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   1 592   0
    0   0   0   0   0   0   0   1   4   0   1]
 [  0   0   0   0   0   0   0   1   4   0   1   0   0   0   0   0   0 560
    6   0  18   6   0   4   0   0   0   0   0]
 [  3   0   0   0   0   0   0   0   0   0   0   0   3   3   0   0   0   0
  578   2   2   0   0   4   0   3   0   1   1]
 [  0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0
   12 576   0   0   0   0   4   6   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0  50
    4   0 539   2   0   4   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   2   0  12   0   1   0   0   0   0  10
    7   1   5 548   8   6   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   6   0   1   0   0   0   0   2
    2   0   1  22 563   0   2   1   0   0   0]
 [  2   0   0   0   0   0   0   0   1   0   0   1   2   0   0   0   0   8
   44   3   3   1   0 529   0   5   0   1   0]
 [  0   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0
    4   1   0   0   0   0 590   2   0   0   0]
 [  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1
    9   0   0   0   0   1   3 584   0   0   0]
 [  0   0   0   0   0   0   0   2   0   0   0   0   1   1   1   1   2   0
    1   0   0   0   0   0   1   2 584   2   2]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   1   0   0   0   0   0   0   0 599   0]
```

**CLASSIFICATION REPORT:-**



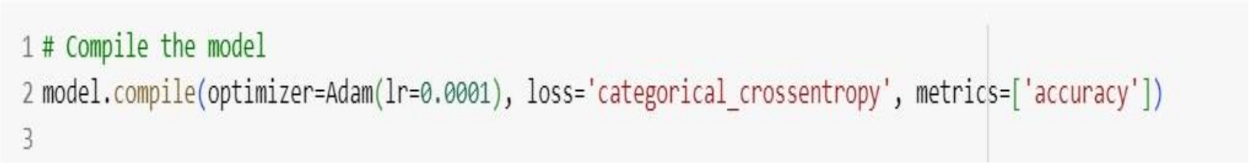
```
136/136 [==============================] - 25s 181ms/step
              precision    recall  f1-score   support

           A       0.95      0.95      0.95       600
           B       0.94      0.96      0.95       600
           C       1.00      0.97      0.98       600
           D       0.99      0.98      0.98       600
           E       0.98      0.94      0.96       600
           F       0.99      0.97      0.98       600
           G       0.99      0.95      0.97       600
           H       0.97      0.98      0.97       600
           I       0.97      0.96      0.96       600
           J       0.97      0.97      0.97       600
           K       0.94      0.91      0.93       600
           L       1.00      0.96      0.98       600
           M       0.93      0.93      0.93       600
           N       0.94      0.95      0.94       600
           O       0.97      0.99      0.98       600
           P       0.98      0.98      0.98       600
           Q       0.99      0.98      0.98       600
           R       0.84      0.93      0.88       600
           S       0.81      0.96      0.88       600
           T       0.98      0.95      0.96       600
           U       0.91      0.89      0.90       600
           V       0.91      0.91      0.91       600
           W       0.98      0.93      0.96       600
           X       0.96      0.87      0.91       600
           Y       0.97      0.97      0.97       600
           Z       0.95      0.98      0.96       600
         del       0.98      0.97      0.98       600
     nothing       0.98      1.00      0.99       600
       space       0.98      0.98      0.98       600

    accuracy                           0.95     17400
   macro avg       0.96      0.95      0.95     17400
weighted avg       0.96      0.95      0.95     17400
```

## ACCURACY SCORE:-



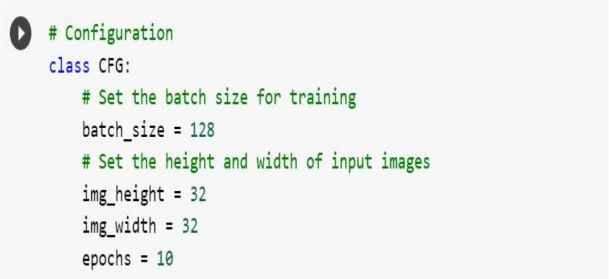
```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.Adam.
Epoch 1/10
543/543 [==============================] - ETA: 0s - loss: 2.5409 - accuracy: 0.1969
Epoch 1: val_accuracy improved from -inf to 0.42445, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 163s 269ms/step - loss: 2.5409 - accuracy: 0.1969 - val_loss: 1.7111 - val_accuracy: 0.4244
Epoch 2/10
543/543 [==============================] - ETA: 0s - loss: 1.1009 - accuracy: 0.6191
Epoch 2: val_accuracy improved from 0.42445 to 0.73911, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 142s 261ms/step - loss: 1.1009 - accuracy: 0.6191 - val_loss: 0.7840 - val_accuracy: 0.7391
Epoch 3/10
543/543 [==============================] - ETA: 0s - loss: 0.5650 - accuracy: 0.8141
Epoch 3: val_accuracy improved from 0.73911 to 0.87142, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 125s 230ms/step - loss: 0.5650 - accuracy: 0.8141 - val_loss: 0.4190 - val_accuracy: 0.8714
Epoch 4/10
543/543 [==============================] - ETA: 0s - loss: 0.3615 - accuracy: 0.8875
Epoch 4: val_accuracy did not improve from 0.87142
543/543 [==============================] - 143s 263ms/step - loss: 0.3615 - accuracy: 0.8875 - val_loss: 0.4847 - val_accuracy: 0.8604
Epoch 5/10
543/543 [==============================] - ETA: 0s - loss: 0.3016 - accuracy: 0.9099
Epoch 5: val_accuracy improved from 0.87142 to 0.93473, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 125s 230ms/step - loss: 0.3016 - accuracy: 0.9099 - val_loss: 0.2324 - val_accuracy: 0.9347
Epoch 6/10
543/543 [==============================] - ETA: 0s - loss: 0.2637 - accuracy: 0.9230
Epoch 6: val_accuracy did not improve from 0.93473
543/543 [==============================] - 143s 263ms/step - loss: 0.2637 - accuracy: 0.9230 - val_loss: 0.2255 - val_accuracy: 0.9334
Epoch 7/10
543/543 [==============================] - ETA: 0s - loss: 0.2280 - accuracy: 0.9352
Epoch 7: val_accuracy improved from 0.93473 to 0.94279, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 147s 270ms/step - loss: 0.2280 - accuracy: 0.9352 - val_loss: 0.2000 - val_accuracy: 0.9428
Epoch 8/10
543/543 [==============================] - ETA: 0s - loss: 0.2205 - accuracy: 0.9377
Epoch 8: val_accuracy did not improve from 0.94279
543/543 [==============================] - 129s 237ms/step - loss: 0.2205 - accuracy: 0.9377 - val_loss: 0.2174 - val_accuracy: 0.9416
Epoch 9/10
543/543 [==============================] - ETA: 0s - loss: 0.2054 - accuracy: 0.9439
Epoch 9: val_accuracy improved from 0.94279 to 0.95756, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 127s 234ms/step - loss: 0.2054 - accuracy: 0.9439 - val_loss: 0.1446 - val_accuracy: 0.9576
Epoch 10/10
543/543 [==============================] - ETA: 0s - loss: 0.1876 - accuracy: 0.9472
Epoch 10: val_accuracy did not improve from 0.95756
543/543 [==============================] - 128s 235ms/step - loss: 0.1876 - accuracy: 0.9472 - val_loss: 0.1809 - val_accuracy: 0.9490
```

## HYPERPARAMETER TUNING:-

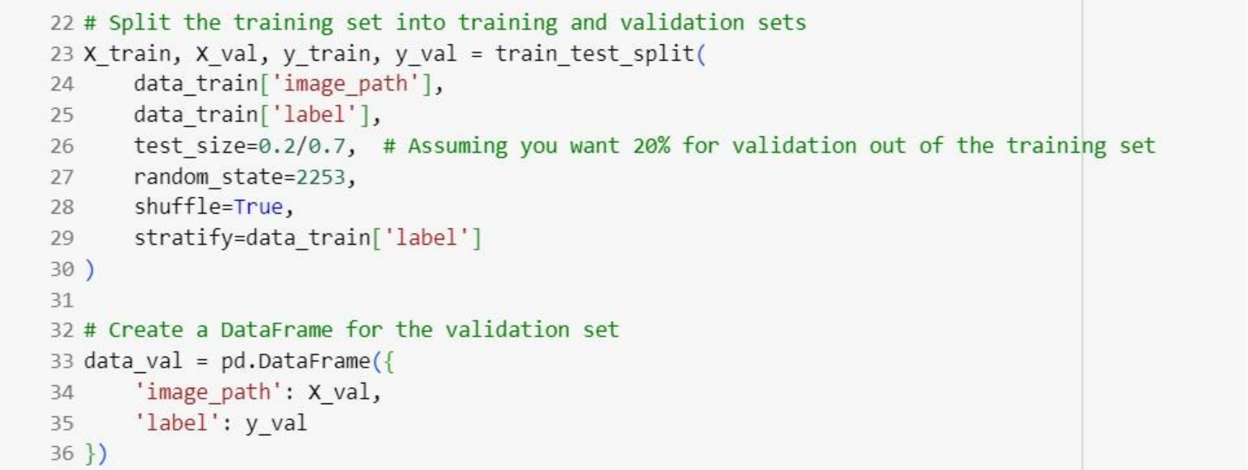


```python
# Compile the model
model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])

```

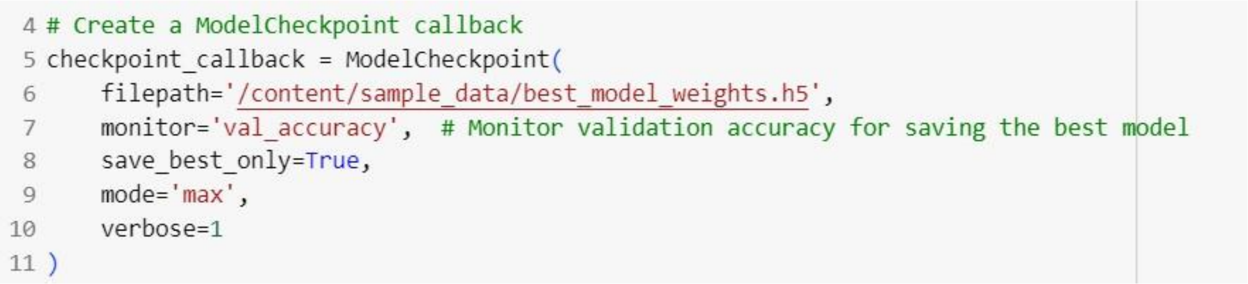
## VALIDATION METHOD:-



```python
# Configuration
class CFG:
    # Set the batch size for training
    batch_size = 128
    # Set the height and width of input images
    img_height = 32
    img_width = 32
    epochs = 10
```



```python
# Split the training set into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(
    data_train['image_path'],
    data_train['label'],
    test_size=0.2/0.7,  # Assuming you want 20% for validation out of the training set
    random_state=2253,
    shuffle=True,
    stratify=data_train['label']
)

# Create a DataFrame for the validation set
data_val = pd.DataFrame({
    'image_path': X_val,
    'label': y_val
})
```



```python
# Create a ModelCheckpoint callback
checkpoint_callback = ModelCheckpoint(
    filepath='/content/sample_data/best_model_weights.h5',
    monitor='val_accuracy',  # Monitor validation accuracy for saving the best model
    save_best_only=True,
    mode='max',
    verbose=1
)
```