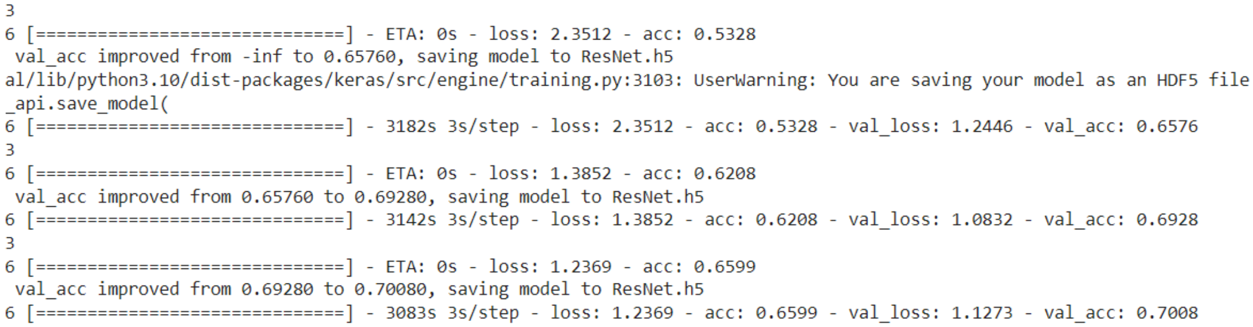
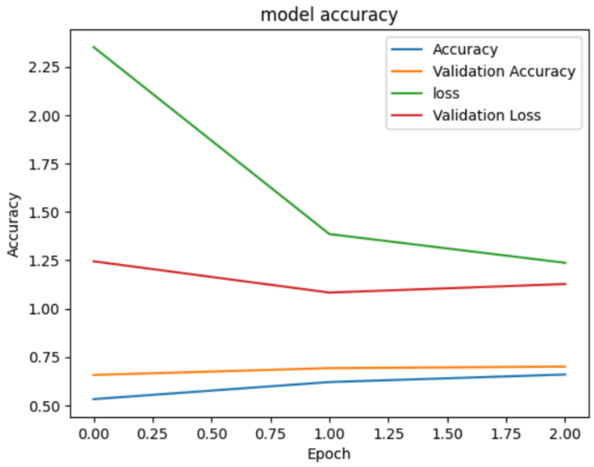
Results

Note: To check the results manually, follow the same procedure as shown in vgg16.ipynb and as already described in Readme.md, change your choose_model variable to the desired architecture in vgg16.ipynb to get the following results.

ResNet50:



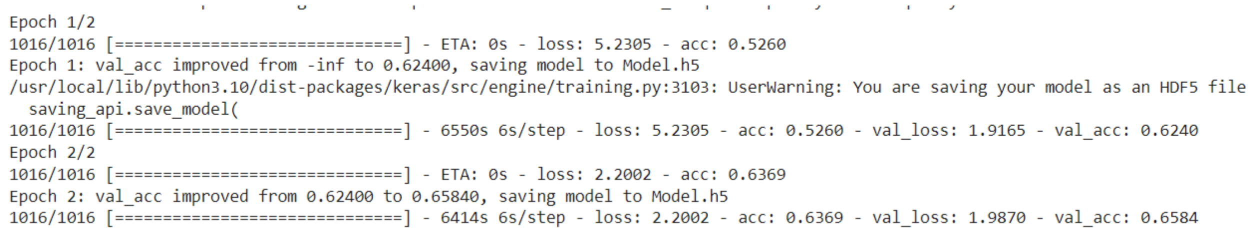
```
3
6 [==============================] - ETA: 0s - loss: 2.3512 - acc: 0.5328
 val_acc improved from -inf to 0.65760, saving model to ResNet.h5
al/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file
_api.save_model(
6 [==============================] - 3182s 3s/step - loss: 2.3512 - acc: 0.5328 - val_loss: 1.2446 - val_acc: 0.6576
3
6 [==============================] - ETA: 0s - loss: 1.3852 - acc: 0.6208
 val_acc improved from 0.65760 to 0.69280, saving model to ResNet.h5
6 [==============================] - 3142s 3s/step - loss: 1.3852 - acc: 0.6208 - val_loss: 1.0832 - val_acc: 0.6928
3
6 [==============================] - ETA: 0s - loss: 1.2369 - acc: 0.6599
 val_acc improved from 0.69280 to 0.70080, saving model to ResNet.h5
6 [==============================] - 3083s 3s/step - loss: 1.2369 - acc: 0.6599 - val_loss: 1.1273 - val_acc: 0.7008
```



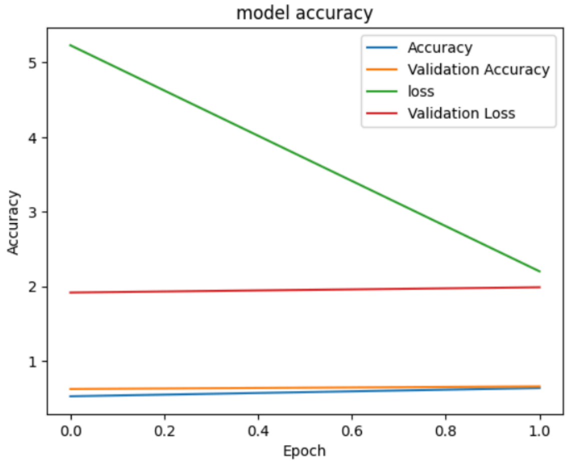
Under 3 epochs and above specified parameters, we notice decrease in training loss, decrease in validation loss from 1st epoch to 2nd epoch and a slight increase from 2nd epoch to 3rd epoch, Increase in both validation and training accuracy.

Therefore by using ResNet50, we have achieved pretty good accuracy in just 3 epochs. The accuracy will increase more, as we increase the number of epochs

VGG16:



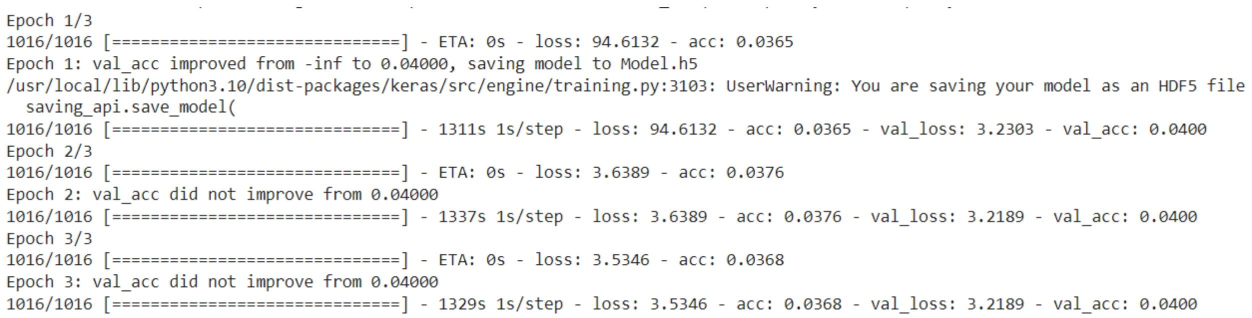
```
Epoch 1/2
1016/1016 [==============================] - ETA: 0s - loss: 5.2305 - acc: 0.5260
Epoch 1: val_acc improved from -inf to 0.62400, saving model to Model.h5
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file
  saving_api.save_model(
1016/1016 [==============================] - 6550s 6s/step - loss: 5.2305 - acc: 0.5260 - val_loss: 1.9165 - val_acc: 0.6240
Epoch 2/2
1016/1016 [==============================] - ETA: 0s - loss: 2.2002 - acc: 0.6369
Epoch 2: val_acc improved from 0.62400 to 0.65840, saving model to Model.h5
1016/1016 [==============================] - 6414s 6s/step - loss: 2.2002 - acc: 0.6369 - val_loss: 1.9870 - val_acc: 0.6584
```



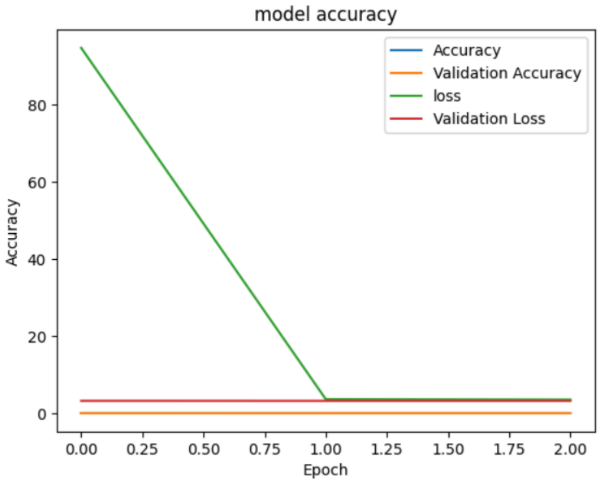
Under 3 epochs and above specified parameters, Training loss significantly decreases from 5.23 to 2.20 and Training accuracy significant increase from 0.52 to 0.63 . The Validation loss increases by about 0.07 approximately and validation accuracy increases from 0.62 to 0.65

Therefore by using vgg16, we have achieved pretty good accuracy in just 2 epochs. The accuracy will increase more, as we increase the number of epochs

EfficientNet:

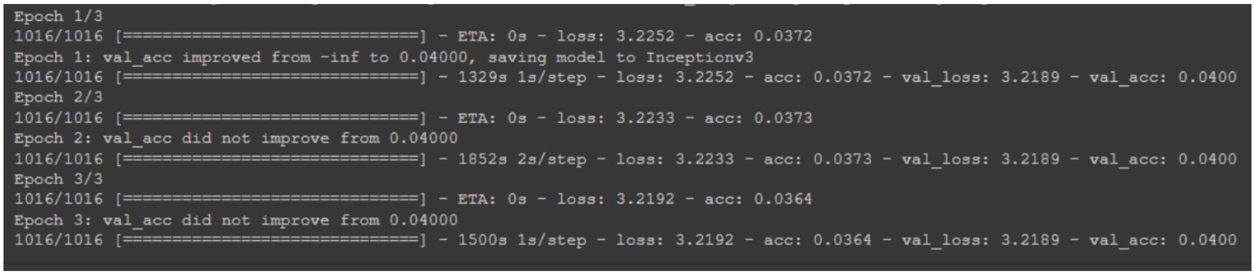


```
Epoch 1/3
1016/1016 [==============================] - ETA: 0s - loss: 94.6132 - acc: 0.0365
Epoch 1: val_acc improved from -inf to 0.04000, saving model to Model.h5
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file
  saving_api.save_model(
1016/1016 [==============================] - 1311s 1s/step - loss: 94.6132 - acc: 0.0365 - val_loss: 3.2303 - val_acc: 0.0400
Epoch 2/3
1016/1016 [==============================] - ETA: 0s - loss: 3.6389 - acc: 0.0376
Epoch 2: val_acc did not improve from 0.04000
1016/1016 [==============================] - 1337s 1s/step - loss: 3.6389 - acc: 0.0376 - val_loss: 3.2189 - val_acc: 0.0400
Epoch 3/3
1016/1016 [==============================] - ETA: 0s - loss: 3.5346 - acc: 0.0368
Epoch 3: val_acc did not improve from 0.04000
1016/1016 [==============================] - 1329s 1s/step - loss: 3.5346 - acc: 0.0368 - val_loss: 3.2189 - val_acc: 0.0400
```

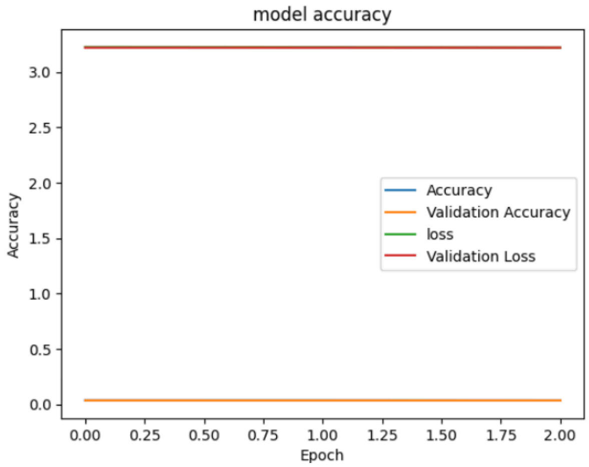


Under 3 epochs and above specified parameters, Training loss - Significant decrease from 94.61 to 3.53. Training accuracy - Significant increase from 0.0365 to 0.0376 and 0.0376 to 0.0368. Validation loss - Decrease in validation loss by about 3.23 to 3.21 approximately while validation accuracy remains constant at 0.0400. Thus, we need to change the hyperparameters, to obtain better results compared to this, even if the model fails to show improvement after 3 epochs.

InceptionV3:



```
Epoch 1/3
1016/1016 [==============================] - ETA: 0s - loss: 3.2252 - acc: 0.0372
Epoch 1: val_acc improved from -inf to 0.04000, saving model to Inceptionv3
1016/1016 [==============================] - 1329s 1s/step - loss: 3.2252 - acc: 0.0372 - val_loss: 3.2189 - val_acc: 0.0400
Epoch 2/3
1016/1016 [==============================] - ETA: 0s - loss: 3.2233 - acc: 0.0373
Epoch 2: val_acc did not improve from 0.04000
1016/1016 [==============================] - 1852s 2s/step - loss: 3.2233 - acc: 0.0373 - val_loss: 3.2189 - val_acc: 0.0400
Epoch 3/3
1016/1016 [==============================] - ETA: 0s - loss: 3.2192 - acc: 0.0364
Epoch 3: val_acc did not improve from 0.04000
1016/1016 [==============================] - 1500s 1s/step - loss: 3.2192 - acc: 0.0364 - val_loss: 3.2189 - val_acc: 0.0400
```



Under 3 epochs and above specified parameters, both validation accuracy and loss remains constant for the above specified parameters. Whereas training loss has decreased from 3.2252 to 3,2192 and training accuracy increased from 0.0372 to 0.0373, at 1st epoch to 2nd epoch and has decreased from 0.0373 to 0.0364, at 2nd epoch to 3rd epoch

Summarization:

Therefore, from the above, It can be concluded that training various algorithms with parameters such as, Adam optimizer, batch_size = 32, lr=0.001,epochs=3, dropout=0.5 and target_size=(128,128). We can arrange the performance of the models in ascending order as follows:

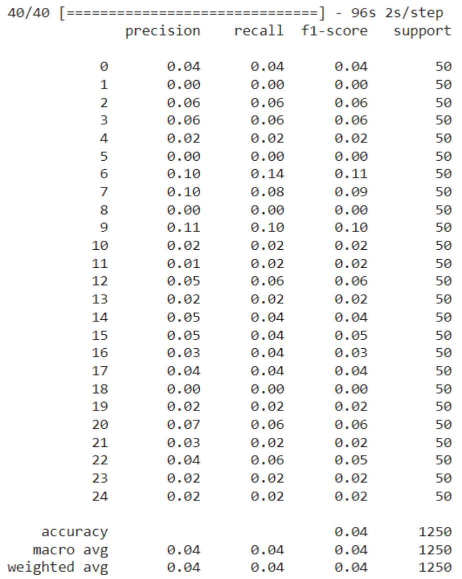
1. ResNet50
2. VGG16
3. EfficientNet
4. InceptionV3

For 3 epochs, ResNet50 and VGG16 has shown good performance. Whereas, EfficientNet and Inceptionv3 has shown poor performance. To improve the performance, hyperparameters should be changed if the performance is found to be poor even after 3 epochs . These hyperparameters can be changed to obtain pretty good results. Hyperparameter tuning can be done manually or using algorithms such as gridsearchcv/randomsearchcv**.**

Evaluation Metrics:

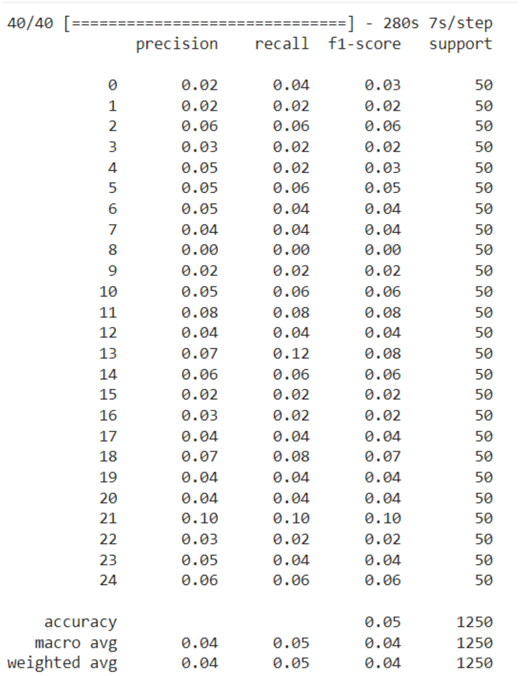
Precision, recall, F1 Score :

ResNet50:



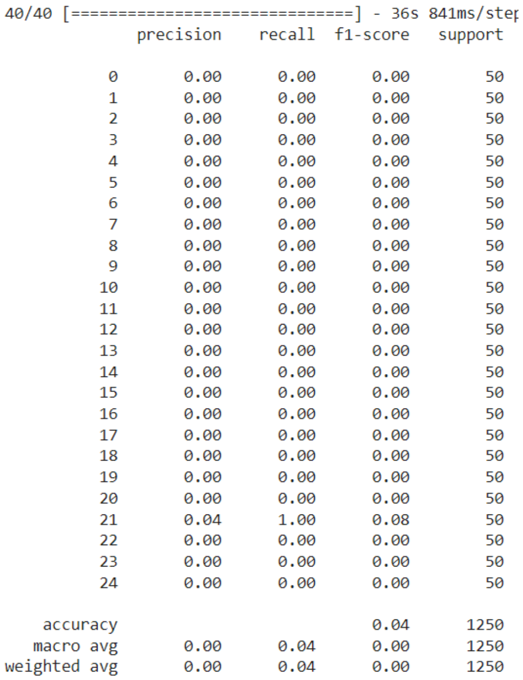
```
40/40 [==============================] - 96s 2s/step
              precision    recall  f1-score   support

           0       0.04      0.04      0.04        50
           1       0.00      0.00      0.00        50
           2       0.06      0.06      0.06        50
           3       0.06      0.06      0.06        50
           4       0.02      0.02      0.02        50
           5       0.00      0.00      0.00        50
           6       0.10      0.14      0.11        50
           7       0.10      0.08      0.09        50
           8       0.00      0.00      0.00        50
           9       0.11      0.10      0.10        50
          10       0.02      0.02      0.02        50
          11       0.01      0.02      0.02        50
          12       0.05      0.06      0.06        50
          13       0.02      0.02      0.02        50
          14       0.05      0.04      0.04        50
          15       0.05      0.04      0.05        50
          16       0.03      0.04      0.03        50
          17       0.04      0.04      0.04        50
          18       0.00      0.00      0.00        50
          19       0.02      0.02      0.02        50
          20       0.07      0.06      0.06        50
          21       0.03      0.02      0.02        50
          22       0.04      0.06      0.05        50
          23       0.02      0.02      0.02        50
          24       0.02      0.02      0.02        50

    accuracy                           0.04      1250
   macro avg       0.04      0.04      0.04      1250
weighted avg       0.04      0.04      0.04      1250
```

VGG16:



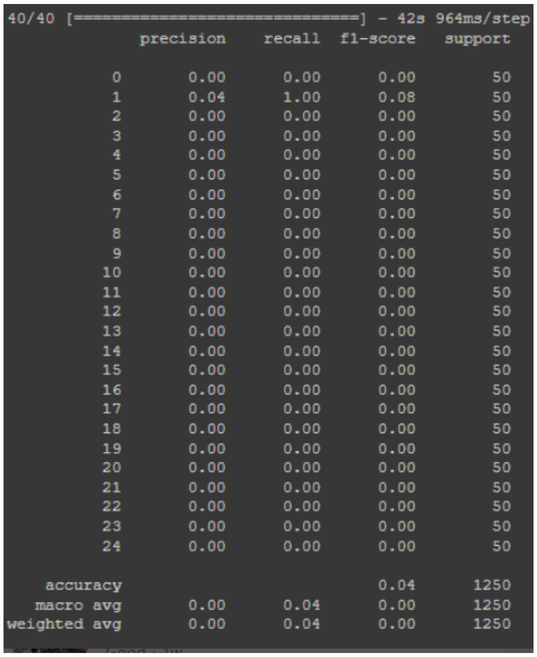
```
40/40 [==============================] - 280s 7s/step
              precision    recall  f1-score   support

           0       0.02      0.04      0.03        50
           1       0.02      0.02      0.02        50
           2       0.06      0.06      0.06        50
           3       0.03      0.02      0.02        50
           4       0.05      0.02      0.03        50
           5       0.05      0.06      0.05        50
           6       0.05      0.04      0.04        50
           7       0.04      0.04      0.04        50
           8       0.00      0.00      0.00        50
           9       0.02      0.02      0.02        50
          10       0.05      0.06      0.06        50
          11       0.08      0.08      0.08        50
          12       0.04      0.04      0.04        50
          13       0.07      0.12      0.08        50
          14       0.06      0.06      0.06        50
          15       0.02      0.02      0.02        50
          16       0.03      0.02      0.02        50
          17       0.04      0.04      0.04        50
          18       0.07      0.08      0.07        50
          19       0.04      0.04      0.04        50
          20       0.04      0.04      0.04        50
          21       0.10      0.10      0.10        50
          22       0.03      0.02      0.02        50
          23       0.05      0.04      0.04        50
          24       0.06      0.06      0.06        50

    accuracy                           0.05      1250
   macro avg       0.04      0.05      0.04      1250
weighted avg       0.04      0.05      0.04      1250
```

EfficientNet:



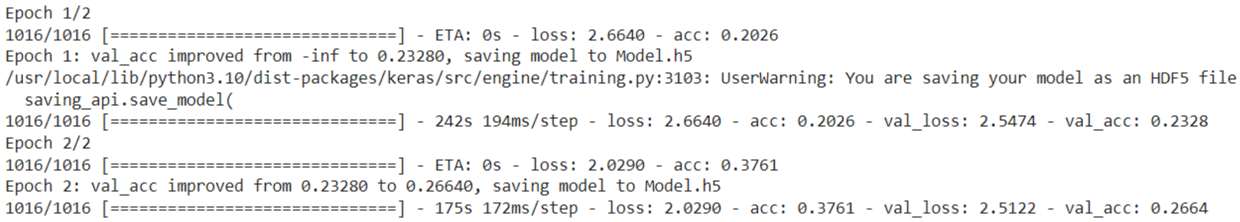
```
40/40 [==============================] - 36s 841ms/step
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        50
           1       0.00      0.00      0.00        50
           2       0.00      0.00      0.00        50
           3       0.00      0.00      0.00        50
           4       0.00      0.00      0.00        50
           5       0.00      0.00      0.00        50
           6       0.00      0.00      0.00        50
           7       0.00      0.00      0.00        50
           8       0.00      0.00      0.00        50
           9       0.00      0.00      0.00        50
          10       0.00      0.00      0.00        50
          11       0.00      0.00      0.00        50
          12       0.00      0.00      0.00        50
          13       0.00      0.00      0.00        50
          14       0.00      0.00      0.00        50
          15       0.00      0.00      0.00        50
          16       0.00      0.00      0.00        50
          17       0.00      0.00      0.00        50
          18       0.00      0.00      0.00        50
          19       0.00      0.00      0.00        50
          20       0.00      0.00      0.00        50
          21       0.04      1.00      0.08        50
          22       0.00      0.00      0.00        50
          23       0.00      0.00      0.00        50
          24       0.00      0.00      0.00        50

    accuracy                           0.04      1250
   macro avg       0.00      0.04      0.00      1250
weighted avg       0.00      0.04      0.00      1250
```

Inceptionv3

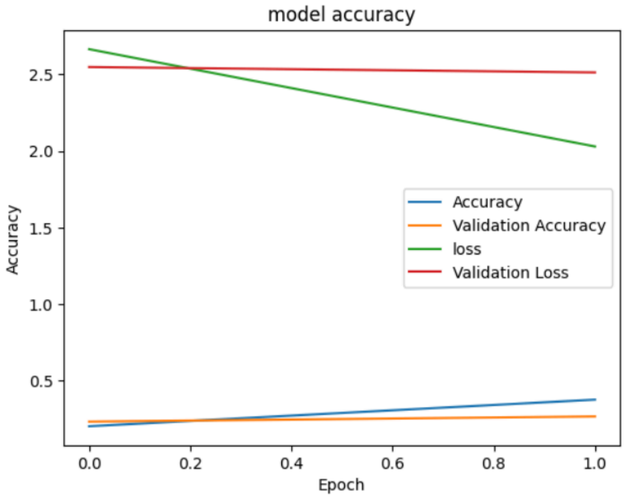


```
40/40 [==============================] - 42s 964ms/step
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        50
           1       0.04      1.00      0.08        50
           2       0.00      0.00      0.00        50
           3       0.00      0.00      0.00        50
           4       0.00      0.00      0.00        50
           5       0.00      0.00      0.00        50
           6       0.00      0.00      0.00        50
           7       0.00      0.00      0.00        50
           8       0.00      0.00      0.00        50
           9       0.00      0.00      0.00        50
          10       0.00      0.00      0.00        50
          11       0.00      0.00      0.00        50
          12       0.00      0.00      0.00        50
          13       0.00      0.00      0.00        50
          14       0.00      0.00      0.00        50
          15       0.00      0.00      0.00        50
          16       0.00      0.00      0.00        50
          17       0.00      0.00      0.00        50
          18       0.00      0.00      0.00        50
          19       0.00      0.00      0.00        50
          20       0.00      0.00      0.00        50
          21       0.00      0.00      0.00        50
          22       0.00      0.00      0.00        50
          23       0.00      0.00      0.00        50
          24       0.00      0.00      0.00        50

    accuracy                           0.04      1250
   macro avg       0.00      0.04      0.00      1250
weighted avg       0.00      0.04      0.00      1250
```

Since Inception showed a very poor performance, I used hyperparameter tuning to get better results comparatively. Follow the procedure in inception_hp.ipynb to get the following results. Also, do the necessary changes as described in Readme.md

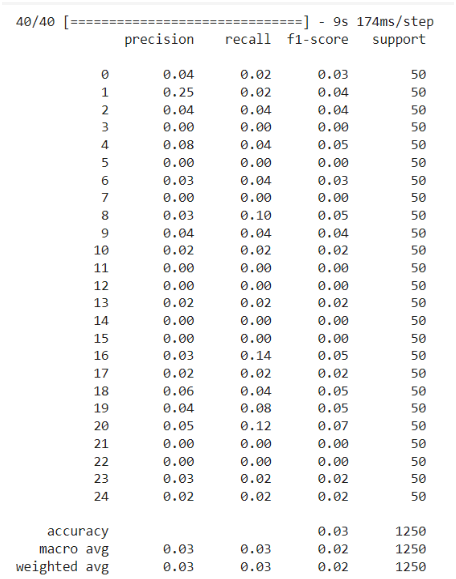


```
Epoch 1/2
1016/1016 [==============================] - ETA: 0s - loss: 2.6640 - acc: 0.2026
Epoch 1: val_acc improved from -inf to 0.23280, saving model to Model.h5
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file
  saving_api.save_model(
1016/1016 [==============================] - 242s 194ms/step - loss: 2.6640 - acc: 0.2026 - val_loss: 2.5474 - val_acc: 0.2328
Epoch 2/2
1016/1016 [==============================] - ETA: 0s - loss: 2.0290 - acc: 0.3761
Epoch 2: val_acc improved from 0.23280 to 0.26640, saving model to Model.h5
1016/1016 [==============================] - 175s 172ms/step - loss: 2.0290 - acc: 0.3761 - val_loss: 2.5122 - val_acc: 0.2664
```



We see that, the Accuracy increased from 20.2% to 37.6% and the Loss decreased from 2.6 to 2.0. Also the Validation accuracy increased from 23% to 26% and Validation loss dropped from 2.54 to 2.51.

Prediction:



```
40/40 [==============================] - 9s 174ms/step
              precision    recall  f1-score   support

           0       0.04      0.02      0.03        50
           1       0.25      0.02      0.04        50
           2       0.04      0.04      0.04        50
           3       0.00      0.00      0.00        50
           4       0.08      0.04      0.05        50
           5       0.00      0.00      0.00        50
           6       0.03      0.04      0.03        50
           7       0.00      0.00      0.00        50
           8       0.03      0.10      0.05        50
           9       0.04      0.04      0.04        50
          10       0.02      0.02      0.02        50
          11       0.00      0.00      0.00        50
          12       0.00      0.00      0.00        50
          13       0.02      0.02      0.02        50
          14       0.00      0.00      0.00        50
          15       0.00      0.00      0.00        50
          16       0.03      0.14      0.05        50
          17       0.02      0.02      0.02        50
          18       0.06      0.04      0.05        50
          19       0.04      0.08      0.05        50
          20       0.05      0.12      0.07        50
          21       0.00      0.00      0.00        50
          22       0.00      0.00      0.00        50
          23       0.03      0.02      0.02        50
          24       0.02      0.02      0.02        50

    accuracy                           0.03      1250
   macro avg       0.03      0.03      0.02      1250
weighted avg       0.03      0.03      0.02      1250
```

Randomsearch hyperparameter tuning can be implemented for other architectures too, by just changing the choose_model variable in inception_hp.ipynb to any one of the following, such as vgg16, inception, efficientnet and resnet50. Based on the best hyperparameters obtained, the model can be trained to achieve a better accuracy rate. The complete code implementation can be found in inception_hp.ipynb.