

Project Development Phase
Model Performance Test

Date	17 November 2022
Team ID	PNT2022TMID52124
Project Name	Digital Naturalist - AI Enabled tool for Biodiversity Researchers
Maximum Marks	10 Marks

Model Performance Testing:

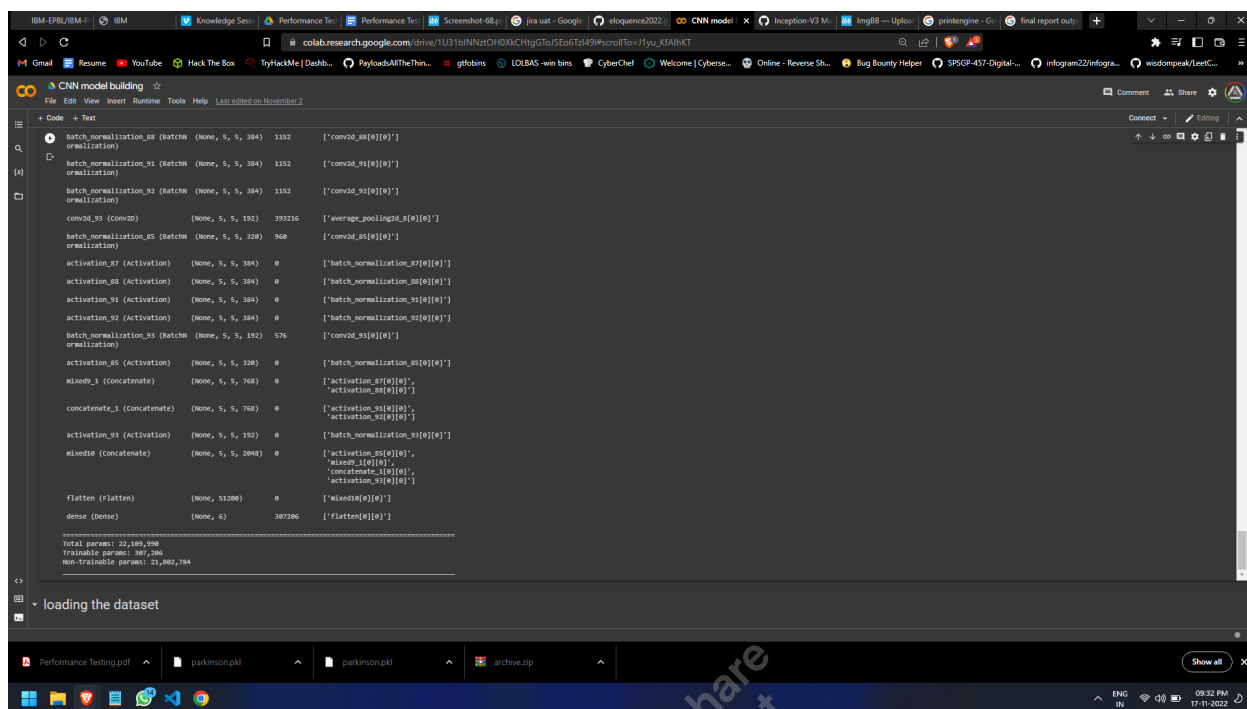
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 22,109,990 Trainable params: 307,206 Non-trainable params: 21,802,784	Screenshot 1
2.	Accuracy	Training Accuracy - 92.8% Validation Accuracy - 85.6%	Screenshot 2

Screenshots - Please refer to the next page:

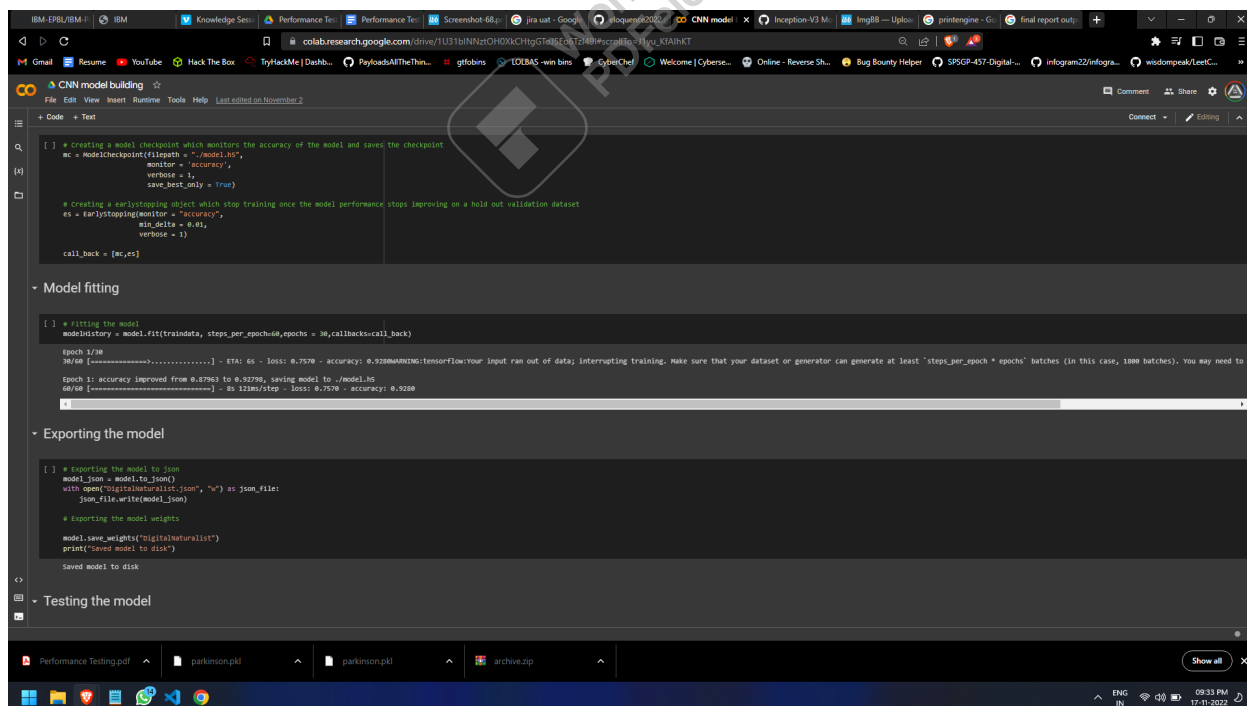


Screenshot 1 :



```
File Edit View Insert Runtime Tools Help Last edited on November 7
CNN model building
+ Code + Text
batch_normalization_B8 (Batch Normalization) (None, 5, 5, 384) 1152 ['convd_8[0][0]']
batch_normalization_B1 (Batch Normalization) (None, 5, 5, 384) 1152 ['convd_9[0][0]']
batch_normalization_B2 (Batch Normalization) (None, 5, 5, 384) 1152 ['convd_10[0][0]']
convd_3 (Conv2D) (None, 5, 5, 192) 193216 ['average_poolingd_1[0][0]']
batch_normalization_B5 (Batch Normalization) (None, 5, 5, 320) 960 ['convd_15[0][0]']
activation_7 (Activation) (None, 5, 5, 384) 0 ['batch_normalization_B7[0][0]']
activation_8 (Activation) (None, 5, 5, 384) 0 ['batch_normalization_B8[0][0]']
activation_9 (Activation) (None, 5, 5, 384) 0 ['batch_normalization_B9[0][0]']
activation_10 (Activation) (None, 5, 5, 384) 0 ['batch_normalization_B10[0][0]']
batch_normalization_B3 (Batch Normalization) (None, 5, 5, 192) 576 ['convd_3[0][0]']
activation_15 (Activation) (None, 5, 5, 128) 0 ['batch_normalization_B5[0][0]']
mixed_1 (Concatenate) (None, 5, 5, 768) 0 ['activation_7[0][0]', 'activation_8[0][0]']
concatenate_1 (Concatenate) (None, 5, 5, 768) 0 ['activation_9[0][0]', 'activation_10[0][0]']
activation_19 (Activation) (None, 5, 5, 192) 0 ['batch_normalization_B3[0][0]']
mixed_18 (Concatenate) (None, 5, 5, 2048) 0 ['activation_15[0][0]', 'mixed_1[0][0]', 'concatenate_1[0][0]', 'activation_19[0][0]']
flatten (Flatten) (None, 51200) 0 ['mixed_18[0][0]']
dense (Dense) (None, 10) 107200 ['flatten[0][0]']
Total params: 22,189,398
Trainable params: 187,096
Non-trainable params: 21,882,784
loading the dataset
```

Screenshot 2:



```
File Edit View Insert Runtime Tools Help Last edited on November 7
CNN model building
+ Code + Text
# creating a model checkpoint which monitors the accuracy of the model and saves the checkpoint
mc = ModelCheckpoint(filepath = "../model.h5",
                    monitor = 'accuracy',
                    verbose = 1,
                    save_best_only = True)

# creating a earlystopping object which stop training once the model performance stops improving on a hold out validation dataset
es = EarlyStopping(monitor = "accuracy",
                  min_delta = 0.01,
                  verbose = 1)

call_back = [mc, es]

Model fitting

# fitting the model
modelHistory = model.fit(traindata, steps_per_epoch=40, epochs = 30, callbacks=call_back)

Epoch 1/30
30/30 [-----] - ETA: 6s - loss: 0.7578 - accuracy: 0.9288WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least 'steps_per_epoch * epochs' batches (in this case, 1800 batches). You may need to
Epoch 1: accuracy improved from 0.87963 to 0.92798, saving model to ../model.h5
60/60 [-----] - 5s 112ms/step - loss: 0.7578 - accuracy: 0.9288

Exporting the model

# exporting the model to json
model_json = model.to_json()
with open("digitalsubrealist.json", "w") as json_file:
    json_file.write(model_json)

# exporting the model weights
model.save_weights("digitalsubrealist")
print("Saved model to disk")

Saved model to disk

Testing the model
```