

## Project Development Phase Model Performance Test

Date	17 November 2022
Team Members	Shanmugapriya P Dhanushiya Y Monika B Aruna A Sindhumeenal T
Project Name	Project - 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	<b>Total params: 22,109,990</b> <b>Trainable params: 307,206</b> <b>Non-trainable params: 21,802,784</b>	Screenshot 1
2.	Accuracy	Training Accuracy - 90.9%  Validation Accuracy - 96.3%	Screenshot 2

Screenshots - Please refer to the next page:

## Screenshot 1 :

The screenshot shows a Google Colab notebook interface. The browser tabs at the top include 'Inbox (2,273)', 'IBM-Project-463', 'Performance Test', 'IBM', 'IBM-Project-192', 'IBM-19239-166', and 'CNN\_model\_bui'. The notebook title is 'CNN\_model\_building.ipynb' and it was last edited on November 15. The main content area displays a summary of a CNN model architecture with the following layers and parameters:

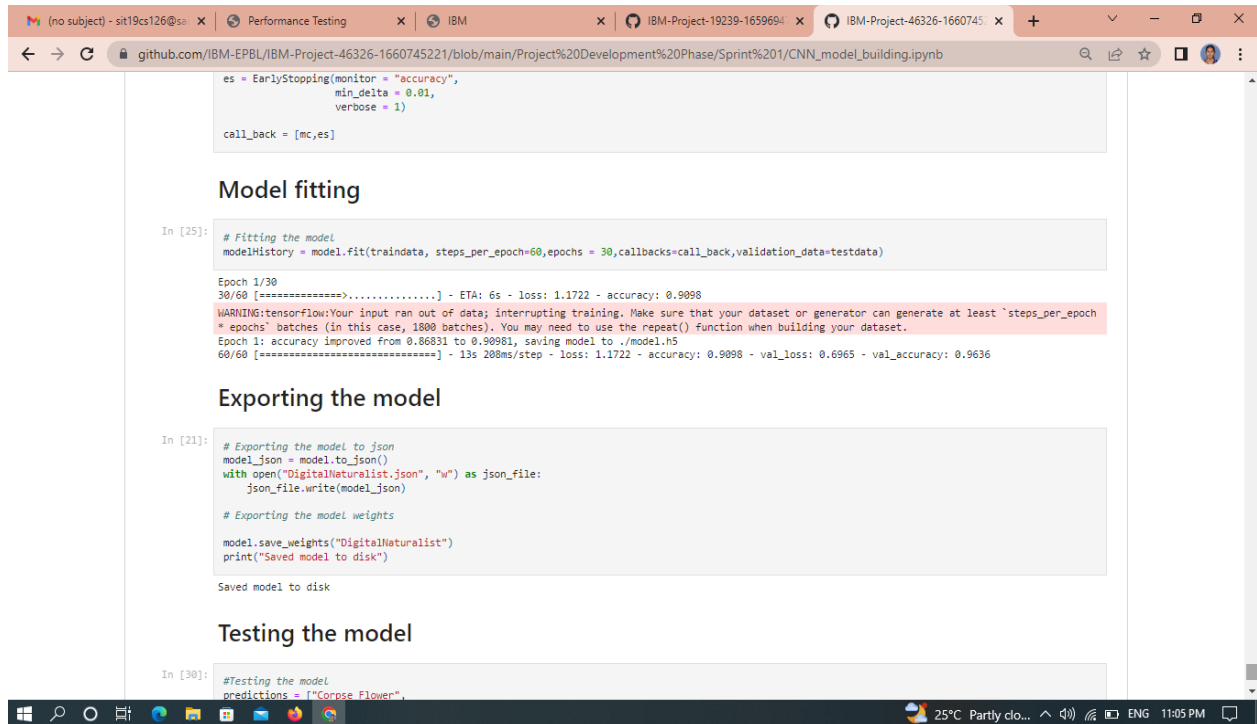
Layer Name	Layer Type	Input Shape	Output Shape	Connections
activation_85	(Activation)	(None, 5, 5, 320)	0	['batch_normalization_85[0][0]']
mixed9_1	(Concatenate)	(None, 5, 5, 768)	0	['activation_87[0][0]', 'activation_88[0][0]']
concatenate_1	(Concatenate)	(None, 5, 5, 768)	0	['activation_91[0][0]', 'activation_92[0][0]']
activation_93	(Activation)	(None, 5, 5, 192)	0	['batch_normalization_93[0][0]']
mixed10	(Concatenate)	(None, 5, 5, 2048)	0	['activation_85[0][0]', 'mixed9_1[0][0]', 'concatenate_1[0][0]', 'activation_93[0][0]']
flatten	(Flatten)	(None, 51200)	0	['mixed10[0][0]']
dense	(Dense)	(None, 6)	307206	['flatten[0][0]']

Summary of parameters:

- Total params: 22,109,990
- Trainable params: 307,206
- Non-trainable params: 21,802,784

Below the summary, a section titled 'loading the train and test dataset' is visible, with a code cell containing the comment '#initializing the datageneration'.

## Screenshot 2:



The screenshot displays a Jupyter Notebook interface within a web browser. The browser's address bar shows the GitHub repository path: `github.com/IBM-EPBL/IBM-Project-46326-1660745221/blob/main/Project%20Development%20Phase/Sprint%201/CNN_model_building.ipynb`. The notebook contains three main sections of code:

```
es = EarlyStopping(monitor = "accuracy",
                  min_delta = 0.01,
                  verbose = 1)

call_back = [mc, es]
```

### Model fitting

```
In [25]: # Fitting the model
modelHistory = model.fit(traindata, steps_per_epoch=60, epochs = 30, callbacks=call_back, validation_data=testdata)

Epoch 1/30
30/60 [=====>.....] - ETA: 6s - loss: 1.1722 - accuracy: 0.9098
WARNING: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 use the repeat() function when building your dataset.
Epoch 1: accuracy improved from 0.86831 to 0.90981, saving model to ./model.h5
60/60 [=====] - 13s 208ms/step - loss: 1.1722 - accuracy: 0.9098 - val_loss: 0.6965 - val_accuracy: 0.9636
```

### Exporting the model

```
In [21]: # Exporting the model to json
model_json = model.to_json()
with open("DigitalNaturalist.json", "w") as json_file:
    json_file.write(model_json)

# Exporting the model weights
model.save_weights("DigitalNaturalist")
print("Saved model to disk")

Saved model to disk
```

### Testing the model

```
In [30]: #Testing the model
predictions = ["Corpse Flower"]
```

The bottom of the image shows a Windows taskbar with various application icons and a system tray displaying the temperature as 25°C, the weather as 'Partly clo...', and the time as 11:05 PM.