

## Project Development Phase Model Performance Test

DATE	18 NOVEMBER 2022
TEAM ID	PNT2022TMID07192
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 - 90.9%  Validation Accuracy - 96.3%	Screenshot 2

Screenshots - Please refer to the next page:

## Screenshot 1 :

The screenshot shows a Jupyter Notebook interface with a browser window at the top. The notebook is titled "CNN\_model\_building.ipynb" and has a menu bar with options: File, Edit, View, Insert, Runtime, Tools, Help. The last edited date is November 15. The notebook content is in Code mode and displays a summary of a CNN model architecture. The summary is organized into a table with columns for layer names, shapes, and connections. Below the table, the total number of parameters is listed: 22,109,990. The number of trainable parameters is 307,206, and the number of non-trainable parameters is 21,802,784. The notebook also shows a section titled "loading the train and test dataset" with a code cell starting with "#initializing the datagenerator".

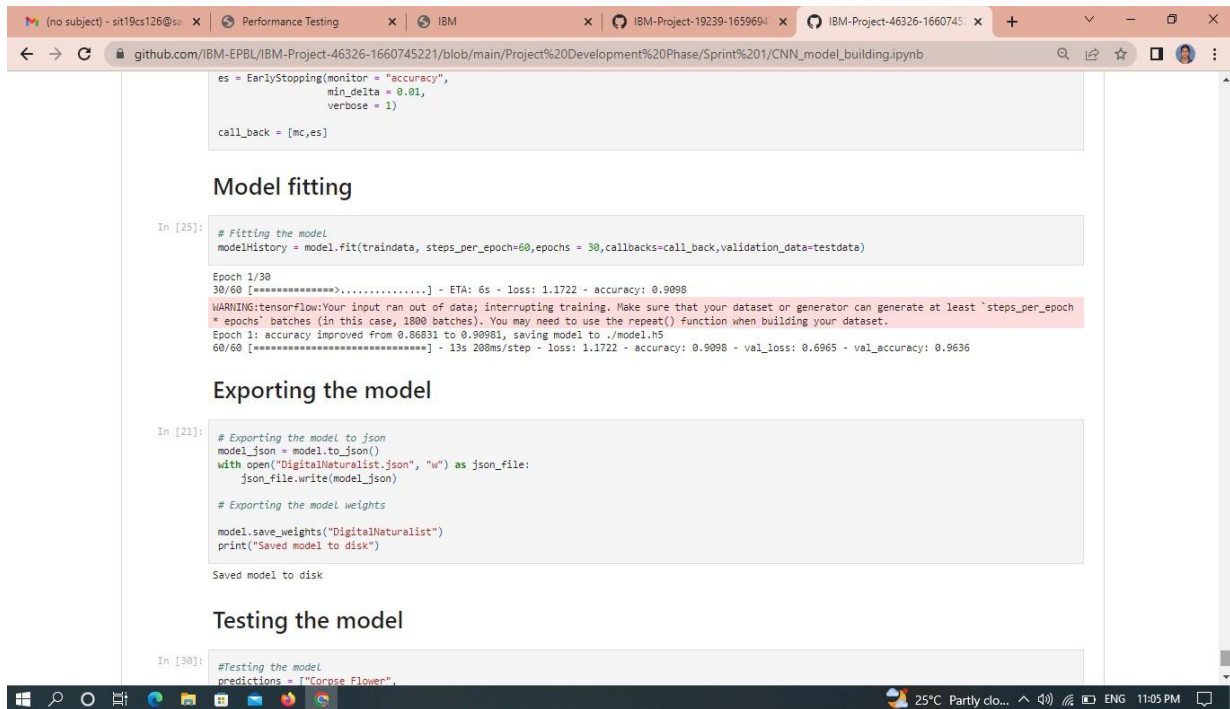
Layer Name	Shape	Connections
activation_85 (Activation)	(None, 5, 5, 320)	0
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0
concatenate_1 (Concatenate)	(None, 5, 5, 768)	0
activation_93 (Activation)	(None, 5, 5, 192)	0
mixed10 (Concatenate)	(None, 5, 5, 2048)	0
flatten (Flatten)	(None, 51200)	0
dense (Dense)	(None, 6)	307206

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

loading the train and test dataset

```
[ ] #initializing the datagenerator
```

## Screenshot 2:



The screenshot shows a Jupyter Notebook interface with three tabs: 'Performance Testing', 'IBM', and 'IBM-Project-19239-1659694'. The active tab is 'IBM-Project-46326-1660745', displaying a GitHub repository page for 'CNN\_model\_building.ipynb'. The notebook content includes:

```
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 = model.predict(testdata)
```

The bottom of the screen shows a Windows taskbar with the system clock at 11:05 PM and temperature at 25°C.