

**Project Development Phase**  
**Model Performance Test**

Date	17 November 2022
Team ID	PNT2022TMID49866
Project Name	Project - Digital Naturalist - AI Enabled tool for Biodiversity Researchers
Maximum Marks	10 Marks

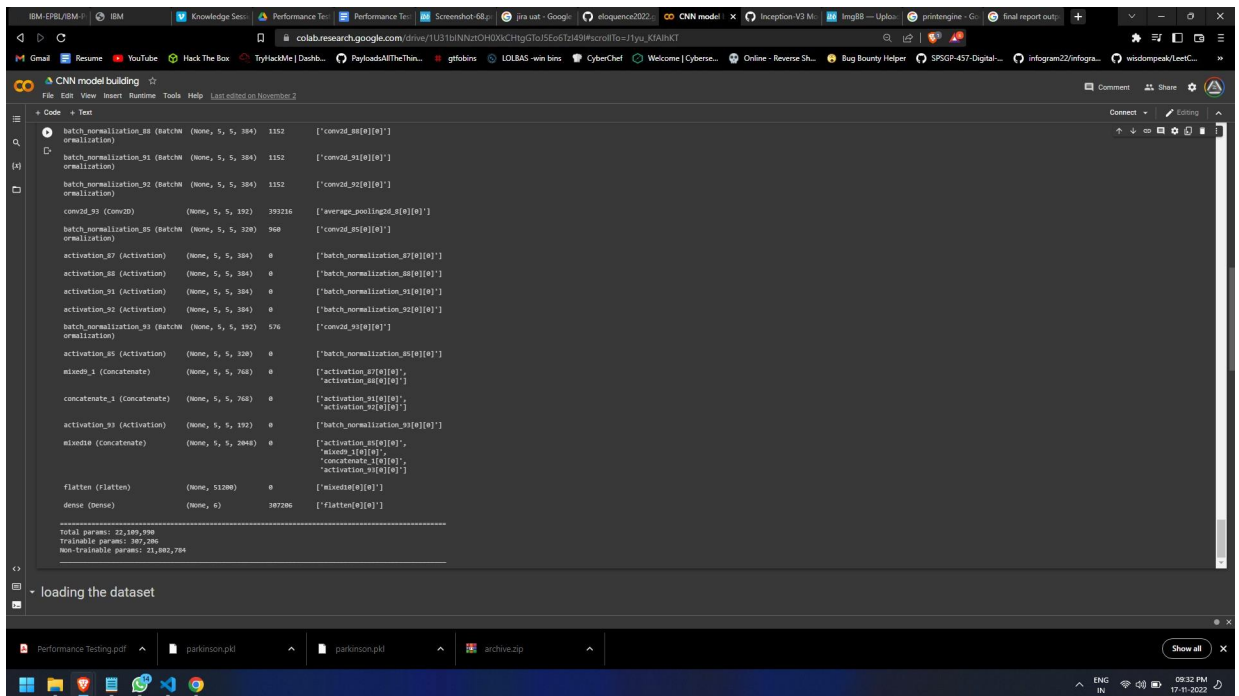
**Model Performance Testing:**

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 - 92.8%  Validation Accuracy - 85.6%	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 displaying a Google Colab link. The notebook is titled "CNN model building" and is in "Code" mode. The code defines a sequential CNN model with the following layers:

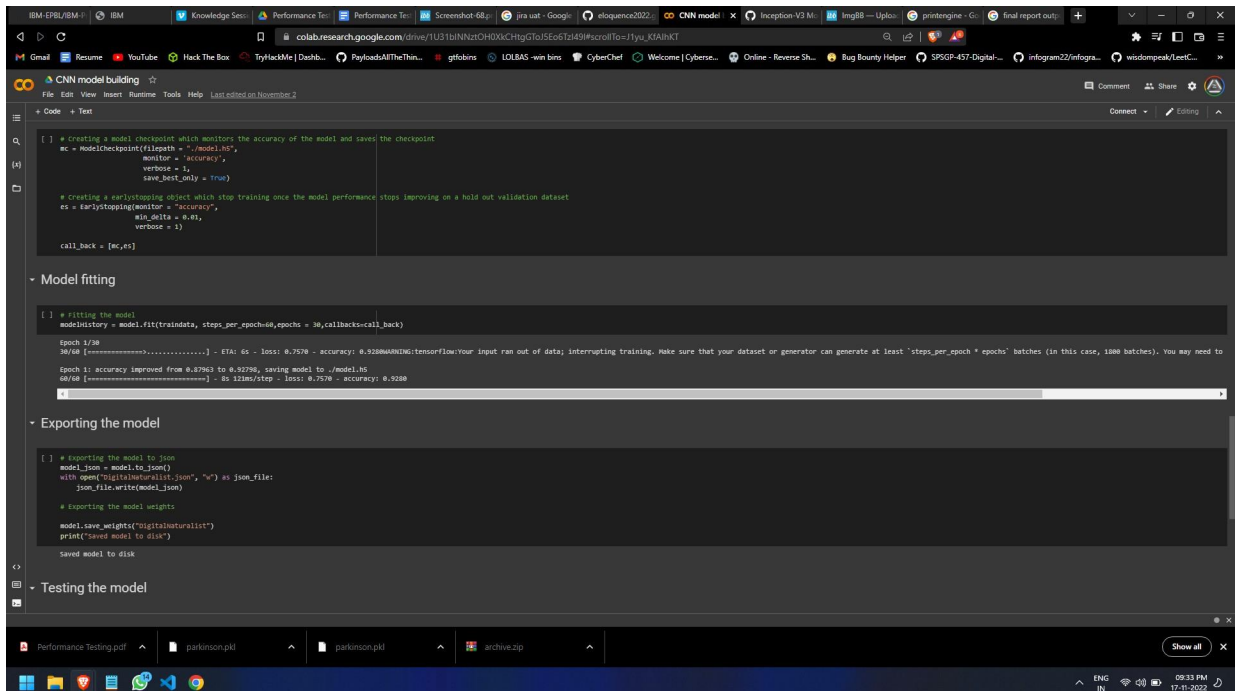
- batch\_normalization\_88 (Batch Normalization)
- batch\_normalization\_91 (Batch Normalization)
- batch\_normalization\_92 (Batch Normalization)
- conv2d\_31 (Conv2D)
- batch\_normalization\_95 (Batch Normalization)
- activation\_87 (Activation)
- activation\_88 (Activation)
- activation\_94 (Activation)
- activation\_92 (Activation)
- batch\_normalization\_93 (Batch Normalization)
- activation\_89 (Activation)
- mixadd\_1 (Concatenate)
- concatenate\_1 (Concatenate)
- activation\_91 (Activation)
- mixadd8 (Concatenate)
- flatten (Flatten)
- dense (Dense)

The model summary at the bottom indicates:

- Total params: 22,189,798
- Trainable params: 387,286
- Non-trainable params: 21,802,512

The notebook is currently on the "loading the dataset" cell.

## Screenshot 2:



The screenshot shows the same Jupyter Notebook, now displaying the "Model fitting" and "Exporting the model" sections.

**Model fitting:**

```
[ ] # fitting the model
modelHistory = model.fit(traindata, steps_per_epoch=64, epochs = 30, callbacks=call_back)

Epoch 1/30
30/0 [=====] - ETA: 6s - loss: 0.7570 - accuracy: 0.530868826: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 0.53087 to 0.53276; saving model to ./model.h5
60/0 [=====] - 5s 12ms/step - loss: 0.7570 - accuracy: 0.5308
```

**Exporting the model:**

```
[ ] # exporting the model to json
model_json = model.to_json()
with open('model.json', 'w') as json_file:
    json_file.write(model_json)

# Exporting the model weights
model.save_weights('weights.h5')
print("saved model to disk")

saved model to disk
```

The notebook is currently on the "Testing the model" cell.