

Project Development Phase
Performance Metrics

Date	19 November 2022
Team ID	PNT2022TMID40321
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	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:

[illegible]

```
CNN model building
File Edit View Insert Run/Debug Tools Help Last edited on November 2

+ Code + Test

[ ] # 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
model.compile(optimizer='adam')
model.fit(train_data, steps_per_epoch=10, epochs = 10, callbacks=call_back)

Epoch 1/10
10/10 [=====]..... - ETA: 4s - loss: 0.7578 - accuracy: 0.3208WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch` epochs (in this case, 1000 batches). You may need to increase the number of samples provided.
Epoch 1: accuracy improved from 0.27963 to 0.32798, saving model to ../model.h5
10/10 [=====]..... - 4s 128K/step - loss: 0.7578 - accuracy: 0.3208

Exporting the model

[ ] # Exporting the model to json
model_json = model.to_json()
with open("digit recognizer1.json", "w") as json_file:
    json_file.write(model_json)

# Exporting the model weights
model.save_weights('digit recognizer1')
print("Saved model to disk")

Saved model to disk

Testing the model
```