

Edge Intelligence

D. Sai Mohith

25MML0032

Task-1:

Key Take aways:

1. Analysing an image dataset-MNIST and applying basic preprocessing operations,
2. Applied Artificial neural network on the following dataset.
3. Saving the model using pickle model.

Code:

```
Jupyter D. Sai Mohith 25MML0032 Last Checkpoint: 12 minutes ago
File Edit View Run Kernel Settings Help
JupyterLab Python [conda env:ml_lab]

[23]: import tensorflow as tf
      from tensorflow.keras import layers, models
      from tensorflow.keras.utils import to_categorical
      import numpy as np
      (x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()

[24]: x_train=x_train.astype("float32")/255.0
      x_test=x_test.astype("float32")/255.0

      x_train=x_train.reshape(-1,784)
      x_test=x_test.reshape(-1,784)

      y_train=to_categorical(y_train,10)
      y_test=to_categorical(y_test,10)

[25]: model=models.Sequential([
      layers.Dense(64,activation='relu',input_shape=(784,)),
      layers.Dense(32,activation='relu'),
      layers.Dense(10,activation='softmax')
      ])

model.summary()

Model: "sequential_3"

Layer (type)                 Output Shape                 Param #
dense_9 (Dense)               (None, 64)                   50,240
dense_10 (Dense)              (None, 32)                   2,080
dense_11 (Dense)              (None, 10)                   330

Total params: 52,650 (205.66 KB)

Trainable params: 52,650 (205.66 KB)

Non-trainable params: 0 (0.00 B)

model.compile(
  optimizer='adam',
  loss='categorical_crossentropy',
  metrics=['accuracy']
)
```

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

```
history=model.fit(
    x_train,y_train,
    epochs=5,
    batch_size=128,
    validation_split=0.1,
    verbose=1)
```

```
Epoch 1/5
422/422 ————— 4s 7ms/step - accuracy: 0.7617 - loss: 0.8098 - val_accuracy: 0.9472 - val_loss: 0.1925
Epoch 2/5
422/422 ————— 3s 6ms/step - accuracy: 0.9378 - loss: 0.2158 - val_accuracy: 0.9602 - val_loss: 0.1494
Epoch 3/5
422/422 ————— 2s 6ms/step - accuracy: 0.9525 - loss: 0.1650 - val_accuracy: 0.9643 - val_loss: 0.1239
Epoch 4/5
422/422 ————— 2s 6ms/step - accuracy: 0.9644 - loss: 0.1222 - val_accuracy: 0.9662 - val_loss: 0.1083
Epoch 5/5
422/422 ————— 2s 6ms/step - accuracy: 0.9704 - loss: 0.0998 - val_accuracy: 0.9697 - val_loss: 0.0987
```

```
test_loss,test_acc=model.evaluate(x_test,y_test)
print(test_acc)
```

```
313/313 ————— 1s 4ms/step - accuracy: 0.9645 - loss: 0.1203
0.9674999713897705
```

```
import pickle
from joblib import Parallel, delayed
import joblib
```

```
# Save the model as a pickle in a file
joblib.dump(model, 'Model.pkl')
```

```
['Model.pkl']
```

```
#Loading model
knn_from_joblib = joblib.load('Model.pkl')
```

```
# Using mn
knn_from_joblib.predict(x_test)
```

```
313/313 ————— 1s 2ms/step
array([[1.1633743e-06, 4.0897052e-07, 8.9151312e-05, ..., 9.9886167e-01,
        5.9455942e-06, 3.5978155e-05],
       [2.1035637e-06, 3.1101084e-04, 9.9802101e-01, ..., 1.1188741e-07,
        1.9130403e-04, 1.7233789e-09],
       [1.5596344e-05, 9.9899894e-01, 2.3897045e-04, ..., 4.3203219e-04,
        1.1988523e-04, 4.3392272e-05],
       ...,
       [2.7945777e-09, 5.4175042e-10, 1.8290272e-09, ..., 1.6284190e-05,
        9.7806937e-05, 5.4433109e-04],
       [2.6654772e-07, 4.2919970e-07, 3.7009013e-08, ..., 4.3409457e-07,
        6.0357887e-04, 2.7900020e-09],
       [2.4362751e-07, 8.7505470e-10, 3.7320308e-07, ..., 7.2588663e-12,
        1.5195829e-07, 2.0652326e-08]], dtype=float32)
```

Size of file:

<input type="checkbox"/> Saved Games	9 months ago	
<input type="checkbox"/> scikit_learn_data	5 months ago	
<input type="checkbox"/> Searches	9 months ago	
<input type="checkbox"/> Videos	9 months ago	
<input type="checkbox"/> 25MML0032_Lab-1_Image_Processing_MNIST.ipynb	28 days ago	5 KB
<input type="checkbox"/> 25MML0032_LogData.ipynb	21 days ago	33.2 KB
<input type="checkbox"/> Animal_ImageDataAnalysis_25MML0032.ipynb	21 days ago	9.3 KB
<input type="checkbox"/> AnimalDataAnalysis.ipynb	27 days ago	9.3 KB
<input type="checkbox"/> API and NLTK.ipynb	2 months ago	8.2 KB
<input type="checkbox"/> API_Dataset.ipynb	22 days ago	2.4 KB
<input type="checkbox"/> CarsDataEngineering.ipynb	3 months ago	167.3 KB
<input type="checkbox"/> clustering.ipynb	2 months ago	25.5 KB
<input type="checkbox"/> clusteringProject.ipynb	2 months ago	68.5 KB
<input type="checkbox"/> Customer_segmentation_unsupervised.ipynb	2 months ago	452.7 KB
<input checked="" type="checkbox"/> D. Sai Mohith 25MML0032.ipynb	now	13.7 KB

Task-2

Steps to collect the data are listed below in detail:

- 1. Make an Edge Impulse account**

Initially, go to the Edge Impulse website and register for a new user account using your email address or one of the accepted login credentials. After registering successfully, access the Edge Impulse Studio dashboard. Projects, datasets, and model training all require this account.

- 2. Visit the section on data acquisition**

Create a new project or open an existing one after logging in. Go to the Data Acquisition tab on the project dashboard. The raw data (pictures, audio, or sensor data) that will be used to train the machine learning model is gathered and uploaded in this step.

- 3. Select the "Connect Data" option.**

Choose Connect Data from the Data Acquisition page. This enables you to directly gather data from Edge Impulse by connecting external devices like a computer, development board, or cell phone.

- 4. Use your phone to scan the QR code.**

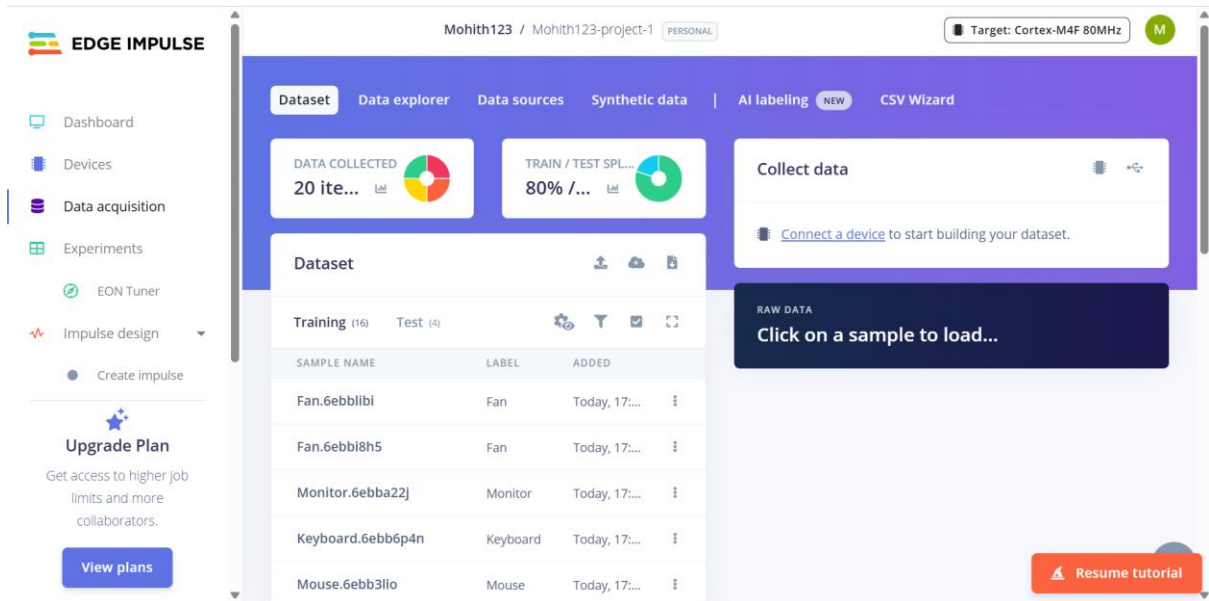
Edge Impulse shows a QR code on the screen after choosing the connect option. Scan this code using your smartphone's camera or QR scanner. This allows you to take and upload data straight from your phone by connecting it to the Edge Impulse project.

- 5. Prior to clicking a picture, choose Label Data**

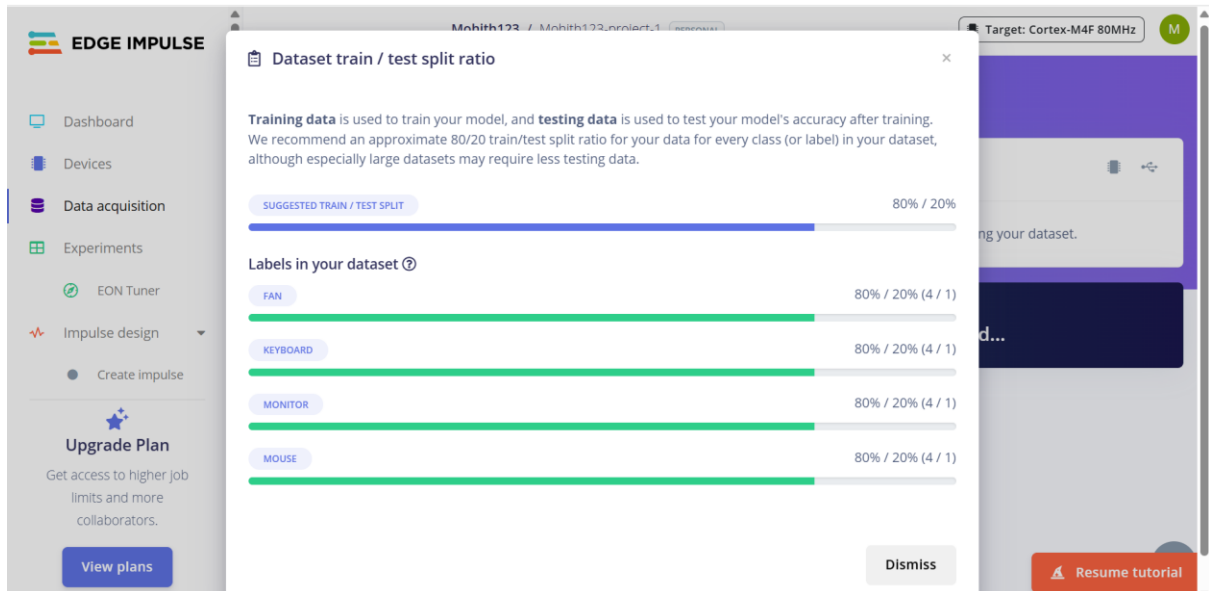
Select the proper label (class name) for the data before taking any pictures, such as "cat," "dog," or "car." Accurately labelling the data at the time of collection is crucial since it aids the machine learning model in comprehending the meaning of each image.

- 6. Divide the recorded images into training and testing sets**

Once the photos have been gathered, separate the dataset into training and testing sets.



Total pictures are 20 out of which 16 are for training and 4 are for testing ie 80:20 ratio



EDGE IMPULSE

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

Create impulse

Upgrade Plan

Get access to higher job limits and more collaborators.

View plans

20 ite...80% /...


Dataset

Training (16)Test (4)

SAMPLE NAME	LABEL	ADDED
Fan.6ebb1ibi	Fan	Today, 17:...
Fan.6ebb18h5	Fan	Today, 17:...
Monitor.6ebba22j	Monitor	Today, 17:...
Keyboard.6ebb6p4n	Keyboard	Today, 17:...
Mouse.6ebb3lio	Mouse	Today, 17:...
Mouse.6ebb3kom	Mouse	Today, 17:...
Mouse.6ebb2sab	Mouse	Today, 17:...
Mouse.6ebb2ekb	Mouse	Today, 17:...

RAW DATA

Monitor.6ebba22j



Resume tutorial