Sprint – III

Model building

Date	**14 November2022**
Team ID	**PNT2022TMID46941**
Project Name	**Natural Disasters Intensity Analysis and Classification using Artificial Intelligence**
Maximum Marks	**20 Marks**

Extract zip file

ZIP is an archive file format that supports lossless data compression. By lossless compression, we mean that the compression algorithm allows the original data to be perfectly reconstructed from the compressed data.

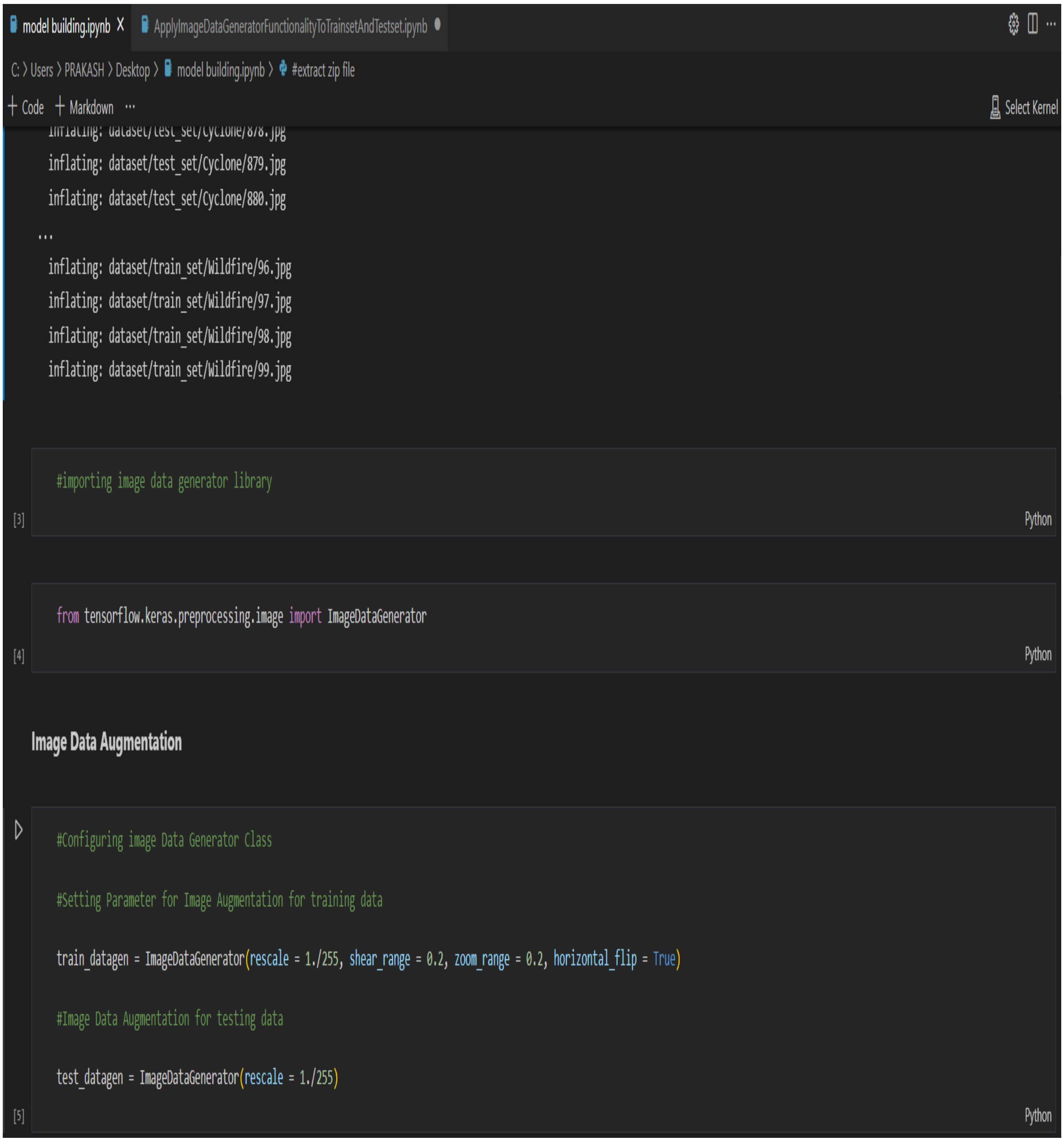
```
#extract zip file

!unzip '/content/drive/MyDrive/IBM/dataset.zip'
```

```
Output exceeds the size limit. Open the full output data in a text editor
Archive:  /content/drive/MyDrive/IBM/dataset.zip
replace dataset/readme.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: dataset/readme.txt
replace dataset/test_set/Cyclone/867.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: dataset/test_set/Cyclone/867.jpg
replace dataset/test_set/Cyclone/868.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: dataset/test_set/Cyclone/868.jpg
replace dataset/test_set/Cyclone/869.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: dataset/test_set/Cyclone/869.jpg
replace dataset/test_set/Cyclone/870.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: dataset/test_set/Cyclone/870.jpg
replace dataset/test_set/Cyclone/871.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: dataset/test_set/Cyclone/871.jpg
replace dataset/test_set/Cyclone/872.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: dataset/test_set/Cyclone/872.jpg
replace dataset/test_set/Cyclone/873.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: dataset/test_set/Cyclone/873.jpg
replace dataset/test_set/Cyclone/874.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: ALL yes
  inflating: dataset/test_set/Cyclone/874.jpg
  inflating: dataset/test_set/Cyclone/875.jpg
  inflating: dataset/test_set/Cyclone/876.jpg
  inflating: dataset/test_set/Cyclone/877.jpg
  inflating: dataset/test_set/Cyclone/878.jpg
  inflating: dataset/test_set/Cyclone/879.jpg
```

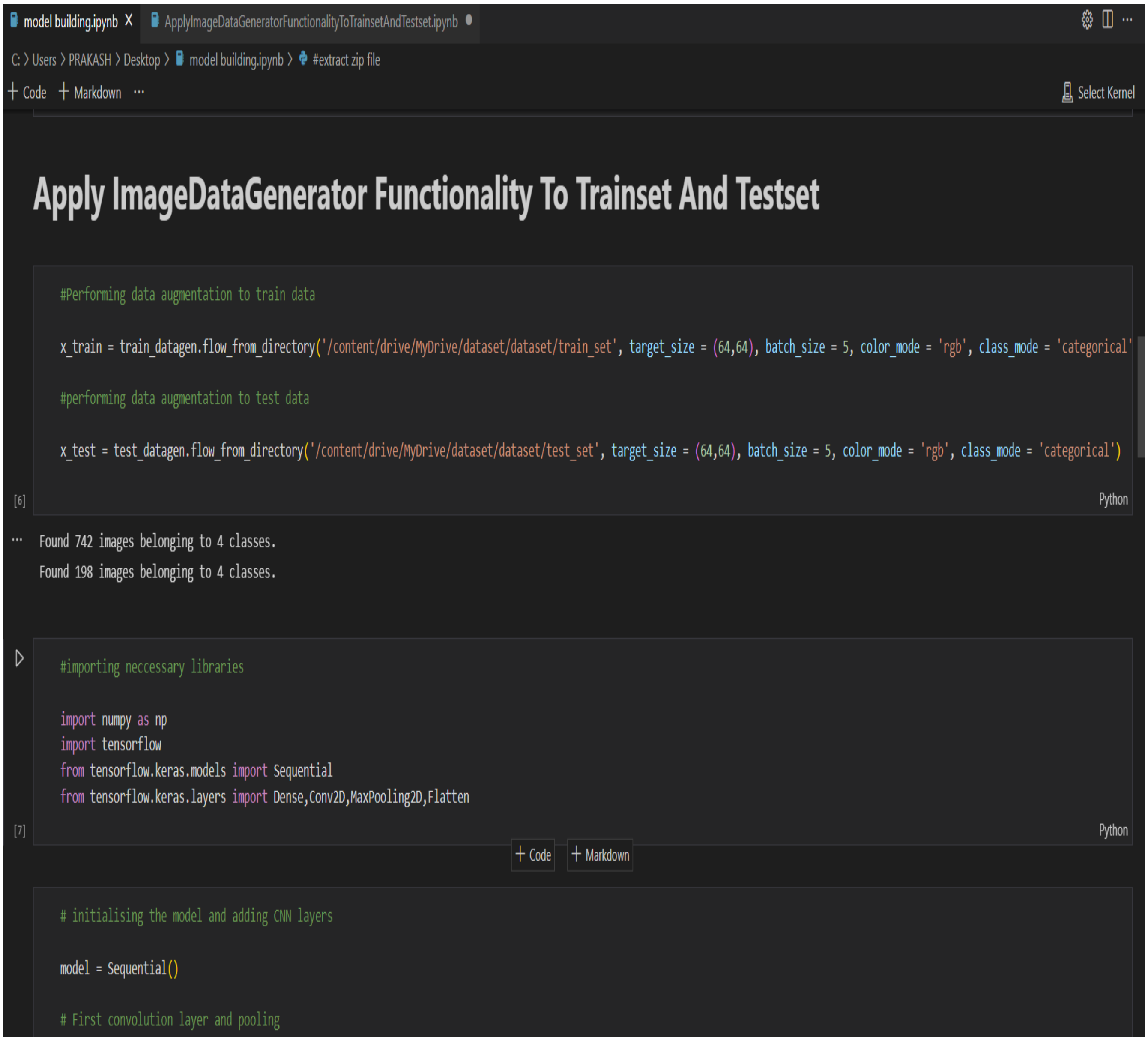
Importing image data generator library/Image data Augmentation

Keras Image Data Generator is used for getting the input of the original data and further, it makes the transformation of this data on a random basis and gives the output resultant containing only the data that is newly transformed.



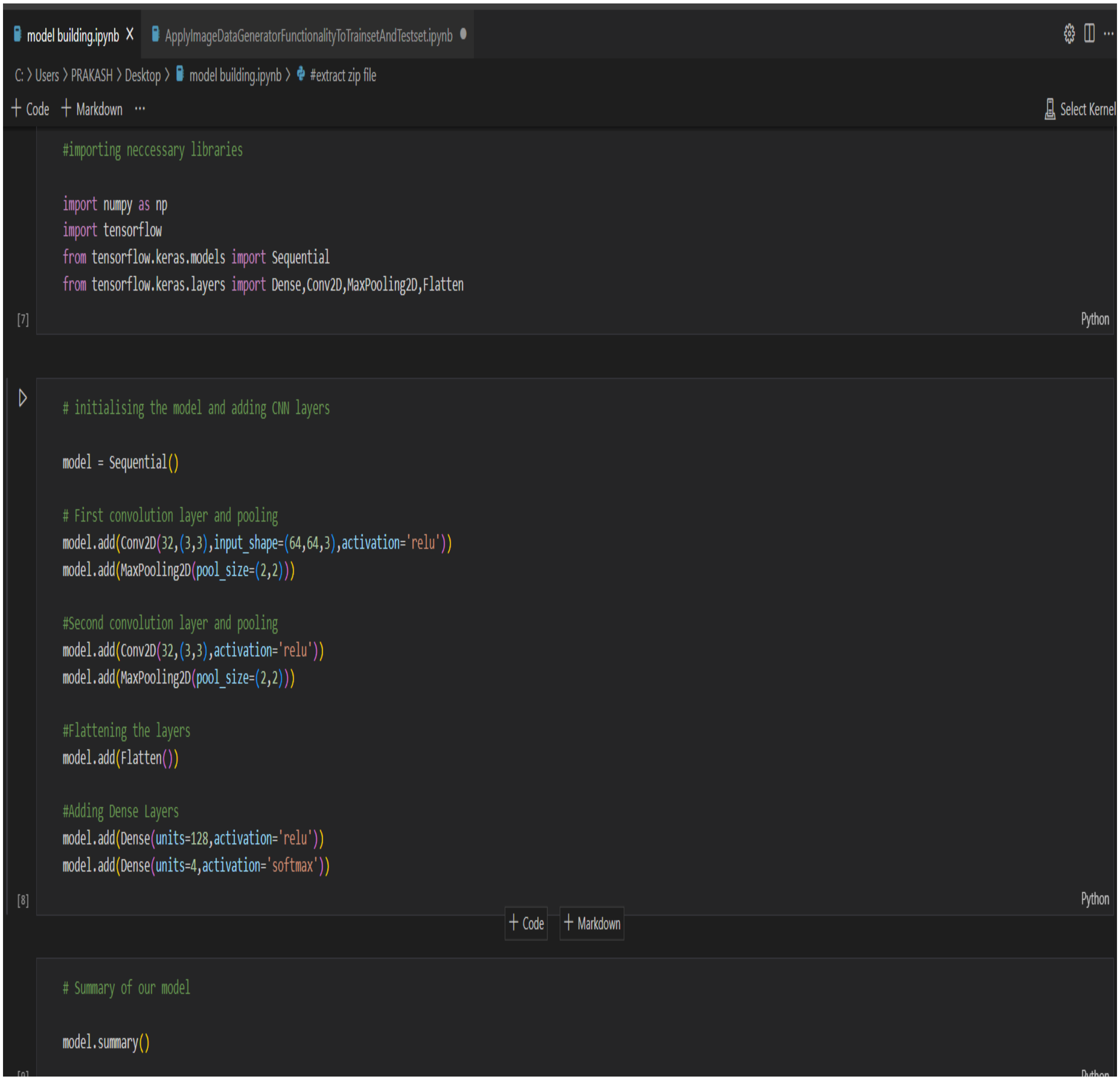
Apply Image Data Generator Functionality to trainset and test set

You probably encountered a situation where you try to load a dataset but there is not enough memory in your machine. As the field of machine learning progresses, this problem becomes more and more common. Today this is already one of the challenges in the field of vision where large datasets of images and video files are processed



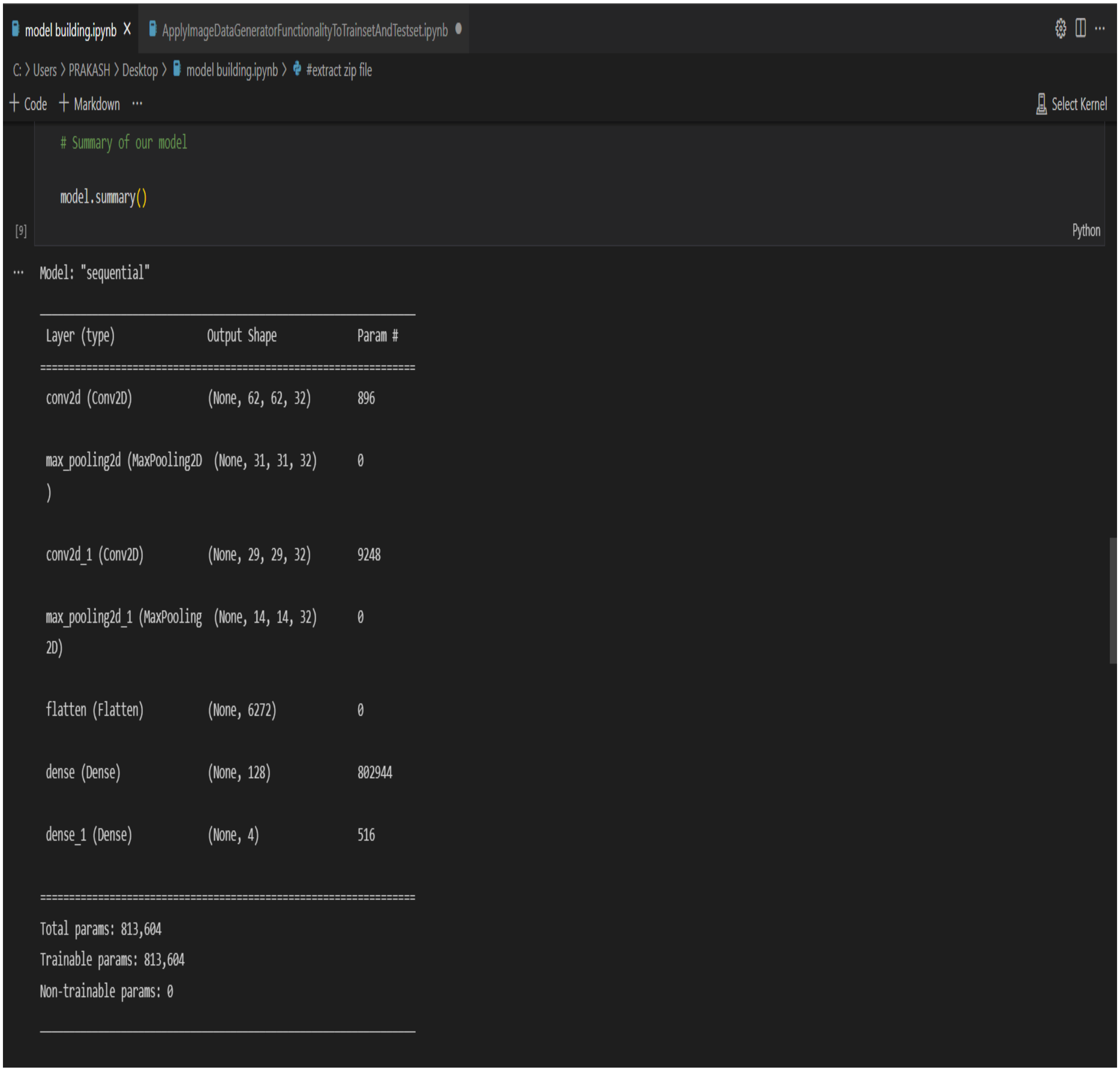
Importing necessary libraries/Initializing the model and adding CNN layers

TensorFlow is a popular deep learning framework. In this tutorial, you will learn the basics of this Python library and understand how to implement these deep, feed-forward artificial neural networks with it.



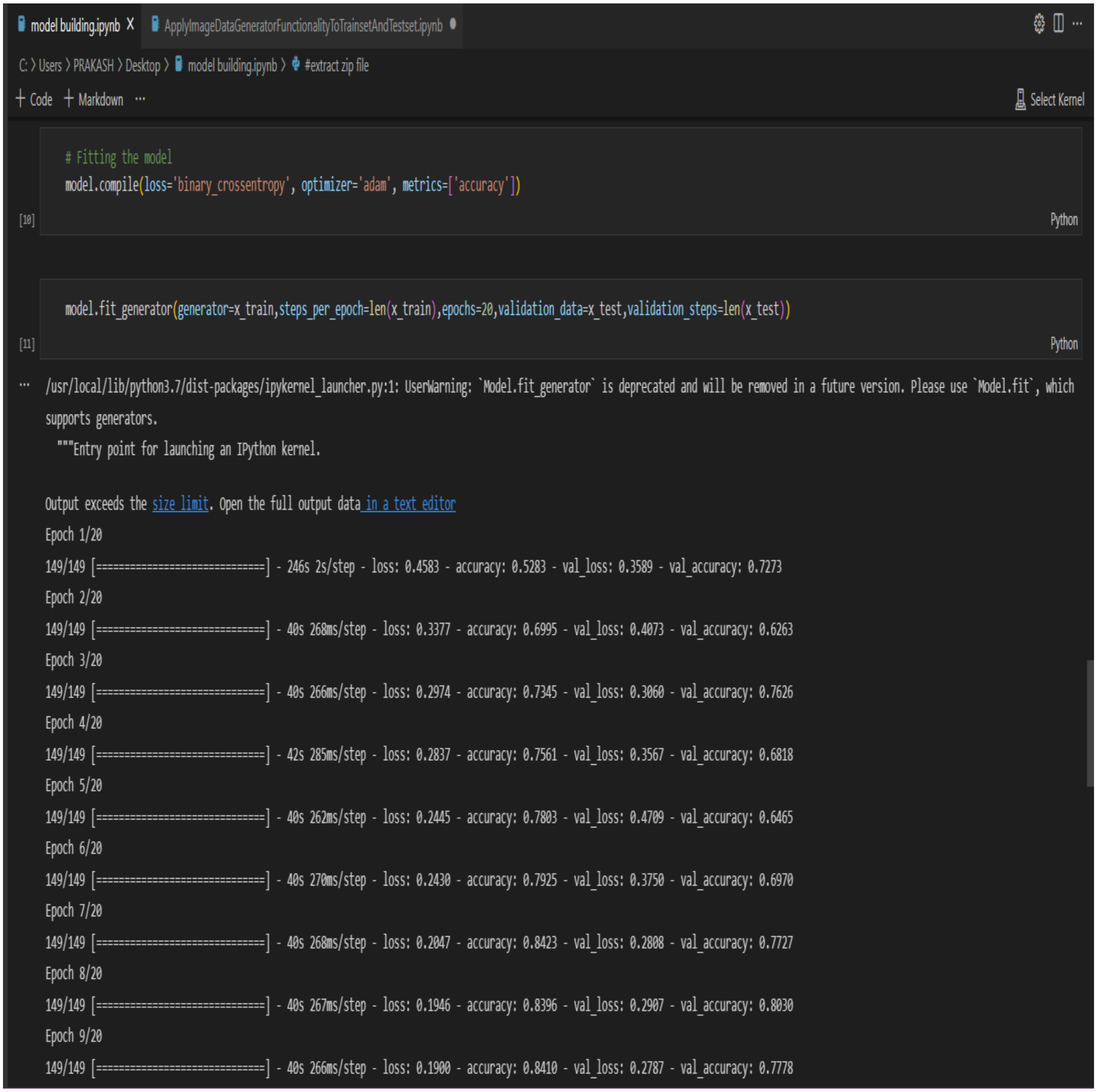
Summary of our model

The model summary gives us a fine visualization of our model and the aim is to provide complete information that is not provided by the print statement.



Fitting the model

We'll define the Keras sequential model and add a one-dimensional convolutional layer. Input shape becomes as it is confirmed above We'll add Dense, MaxPooling1D, and Flatten layers into the model. The output layer contains the number of output classes and 'SoftMax' activation.



Save the model/Load the saved model/Taking image as input

The Saved Model format is another way to serialize models. Models saved in this format can be restored using and are compatible with TensorFlow Serving. The Saved Model goes into detail about how to serve/inspect the Saved Model. The section below illustrates the steps to save and restore the model.

model building.ipynb X ApplyImageDataGeneratorFunctionalityToTrainsetAndTestset.ipynb

C: > Users > PRAKASH > Desktop > model building.ipynb > M↓Apply ImageDataGenerator Functionality To Trainset And Testset > model.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs=20,validation_data=x_test,validation_ste

+ Code + Markdown ··· Select Kernel

```python
# Save the model

model.save('disaster.h5')
model_json = model.to_json()
with open("/content/drive/MyDrive/IBM/model-bw.json", "w") as json_file:
    json_file.write(model_json)
```

[12] Python

```python
# Load the saved model

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model('disaster.h5')
```

[13] Python

```python
x_train.class_indices
```

[14] Python

```
{'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}
```

```python
# taking image as input

img = image.load_img('/content/drive/MyDrive/dataset/dataset/test_set/Wildfire/1040.jpg',target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])
```

[15] Python