

EARLY DETECTION OF FOREST FIRE

MODEL BUILDING

INITIALIZING THE MODEL

TeamID	PNT2022TMID03558
ProjectName	Project-Early detection of forest fire using deeplearning

INITIALILIZINGTHEMODEL:

kerashas2waystodefineaneuralnetwork:

- Sequential
- FunctionAPI

The Sequential class is used to define linear initializations of network layers which then,collectively, constitute a model. In our example below, we will use the Sequential constructortocreateamodel,whichwill thenhavelayersadded toit using theadd () method.

Now,willinitializeourmodel.

11/7/22, 12:35 AM

Untitled8.ipynb - Colaboratory

▾ Importing Keras libraries

```
import keras
```

▾ Importing ImageDataGenerator from Keras

```
from keras.preprocessing.image import ImageDataGenerator
```

▾ Importing Keras libraries

```
[1] import keras
```

▾ Importing ImageDataGenerator from Keras

```
[13] from matplotlib import pyplot as plt
      from keras.preprocessing.image import ImageDataGenerator
```

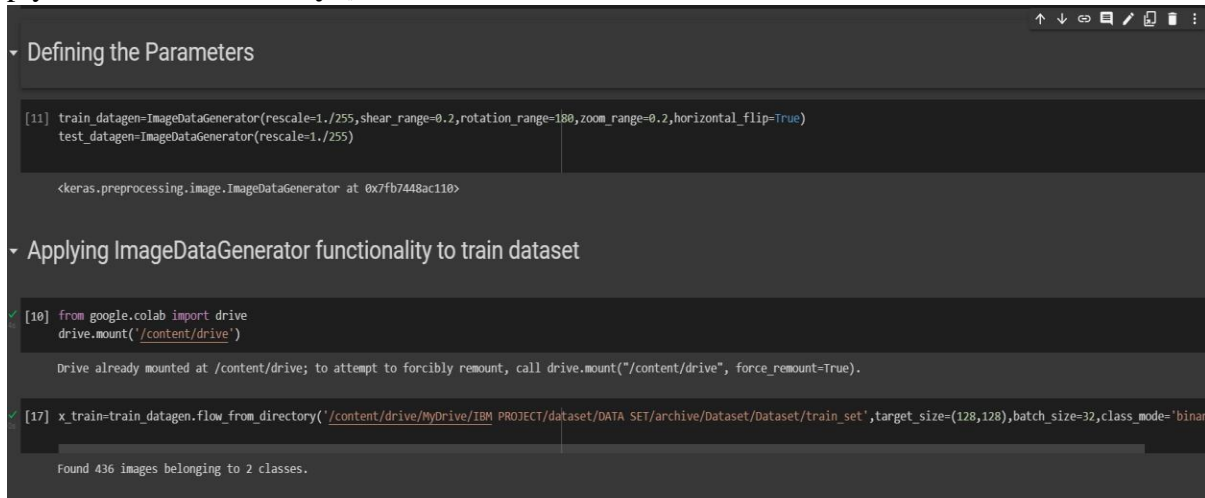
▾ Defining the Parameters

```
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, rotation_range=180, zoom_range=0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
<keras.preprocessing.image.ImageDataGenerator at 0x7fb7448ac110>
```

APPLYING ImageDataGenerator to train dataset:

ply `flow_from_directory()` method for Train folder.



The screenshot shows a Jupyter Notebook with two code cells. The first cell, titled 'Defining the Parameters', defines `train_datagen` and `test_datagen` using `ImageDataGenerator` with parameters: `rescale=1./255`, `shear_range=0.2`, `rotation_range=180`, `zoom_range=0.2`, and `horizontal_flip=True`. The second cell, titled 'Applying ImageDataGenerator functionality to train dataset', imports `drive` from `google.colab`, mounts the drive, and then uses `train_datagen.flow_from_directory()` on the path `"/content/drive/MyDrive/IBM PROJECT/dataset/DATA SET/archive/Dataset/Dataset/train_set"` with `target_size=(128,128)`, `batch_size=32`, and `class_mode='binary'`. The output of the second cell indicates that 436 images were found belonging to 2 classes.

```
[11] train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

<keras.preprocessing.image.ImageDataGenerator at 0x7fb7448ac110>

[10] from google.colab import drive
drive.mount('/content/drive')

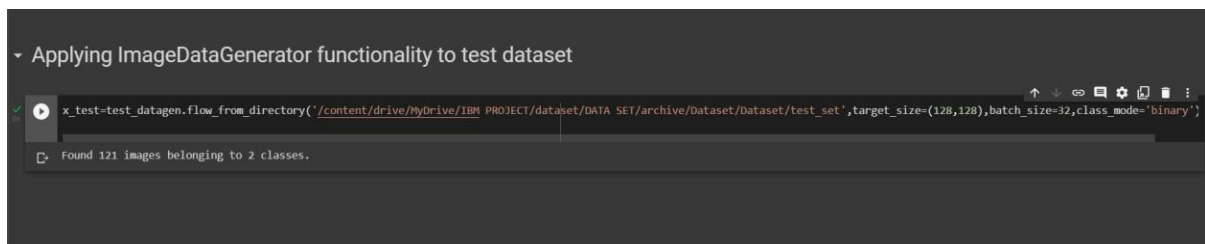
Drive already mounted at /content/drives; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[17] x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM PROJECT/dataset/DATA SET/archive/Dataset/Dataset/train_set',target_size=(128,128),batch_size=32,class_mode='binary')

Found 436 images belonging to 2 classes.
```

APPLYING ImageDataGenerator to test dataset:

Applying the `flow_from_directory()` method for test folder.



The screenshot shows a Jupyter Notebook with one code cell titled 'Applying ImageDataGenerator functionality to test dataset'. The cell uses `x_test=test_datagen.flow_from_directory()` on the path `"/content/drive/MyDrive/IBM PROJECT/dataset/DATA SET/archive/Dataset/Dataset/test_set"` with `target_size=(128,128)`, `batch_size=32`, and `class_mode='binary'`. The output indicates that 121 images were found belonging to 2 classes.

```
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IBM PROJECT/dataset/DATA SET/archive/Dataset/Dataset/test_set',target_size=(128,128),batch_size=32,class_mode='binary')

Found 121 images belonging to 2 classes.
```

IMPORTING MODEL BUILDING LIBRARIES:

11/8/22, 1:16 AM

Main code - Colaboratory

▼ Importing Model Building Libraries

```
#to define the linear Initialisation import sequential
from keras.models import Sequential
#to add layers import Dense
from keras.layers import Dense
#to create Convolutional kernel import convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

INITIALIZINGTHEMODEL:

▼ Initializing the model

```
model=Sequential()
```