SPRINT 1

Team ID : PNT2022TMID47032

Project Name : Natural Disaster Intensity Analysis and Classification Using Artificial Intelligence

Import Image Data Generator from keras

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

Image Data Augmentation

```
#setting parameter for Image Data augmentation to the training data
train_datagen = ImageDataGenerator
(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#Image Data augmentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

Loading our data and performing data augmentation

```
x_train = train_datagen.flow_from_directory
('../data/train_set',target_size=(64,64),batch_size=5,color_mode='rgh',
class_mode='categorical')
x_test = test_datagen.flow_from_directory
('../data/test_set',target_size=(64,64),batch_size=5,color_mode='rgh',
class_mode='categorical')
```

```
Found 742 images belonging to 4 classes.

Found 198 images belonging to 4 classes.
```

Importing the Required Libraries

```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
```

```
(x_train, y_train), (x_test, y_test) = disaster.load_data()
```

```
print (x_train.shape)
print (y_test.shape)
```

```
(60000, 28, 28)
(10000, 28, 28)
```

```
x_train[0]
```

```
array([[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 18, 18, 18, 126, 136,
         175, 26, 166, 255, 247, 127, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 30, 36, 94, 154, 170, 253, 253, 253,
       253, 253, 225, 172, 253, 242, 195, 64, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 49, 238, 253, 253, 253, 253, 253, 253,
       253, 253, 251, 93, 82, 82, 56, 39, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 18, 219, 253, 253, 253, 253, 253, 198,
       182, 247, 241, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 80, 156, 107, 253, 253, 205, 11, 0,
       43, 154, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 1, 154, 253, 90, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0,0,0,0,0,0,0,0,0,0,0,139,253, 190,2,0,0,0,0,0,0,0,0,0,0,0,
       0,0],
       [0,0,0,0,0,0,0,0,0,0,0,11,190, 253,70,0,0,0,0,0,0,0,0,0,0,0,
       0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,35, 241,225,160,108,1,0,0,0,0,0,0,0,0,
       0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0, 81,240,253,253,119,25,0,0,0,0,0,0,0,
         0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0,  0,45,186,253,253,150,27,0,0,0,0,0,0,
          0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0,    0,0,16,93,252,253,187,0,0,0,0,0,0,
          0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0,     0,0,0,0,249,253,249,64,0,0,0,0,0,
          0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0, 0,46,130,183,253,253,207,2,0,0,0,0,0,
          0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,39,
       148,229,253,253,253,250,182,0,0,0,0,0,0, 0,0],
       [0,0,0,0,0,0,0,0,0,0,24,114,221,
       253,253,253,253,201,78,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,0,0,0,0,23,66,213,253,253,
       253,253,198,81,2,0,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,0,0,18,171,219,253,253,253,253,
      195,80,9,0,0,0,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,55,172,226,253,253,253,253,244,133,
       11,0,0,0,0,0,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,136,253,253,253,212,135,132,16,0,
       0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0],
       [0,0,0,0,0,0,0,0,0,0,0,0,0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], dtype=uint8)
```

```
y_train[0]
```

```
5
```

Reshaping The Data

x_train=x_train.shape.reshape (60000,28,28,1).astype (‘float32’)

x_test=x_test.shape.reshape (10000,28,28,1).astype (‘float32’)

Applying One Hot Encoding

number_of_classes = 10

y_train=np.utils.to_categorical(y_train, number_of_classes)
y_test=np.utils.to_categorical(y_test, number_of_classes)

```
y_train[0]

array([0 . , 0 . , 0 . , 0 . , 0. , 1 . , 0 . , 0 . , 0 . , 0 .], dtype=float32
```