

IBM-EPBL/IBM-Project-20074-1659711991

Natural Disasters Intensity Analysis and Classification using Artificial Intelligence

TEAM MEMBERS:

411419104010-LAKSHMIPRIYA D

411419104029-SWETHA A

411419104015-NIVETHA P

411419104004-GAYATHRI M

TEAM ID :PNT2022TMID37914

Project Flow:-

- ◆ The user interacts with the UI (User Interface) to open the integrated webcam.
- ◆ The video frames are captured and analyzed by the model which is integrated with flask application.
- ◆ Once model analyzes the video frames, the prediction is showcased on the UI and OpenCV window

To accomplish this, we have to complete all the activities and tasks listed below

→ Data Collection:

Data collection or data gathering is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes

Collect the dataset or Create the dataset

Downloaded the dataset from the given link given by the ibm.

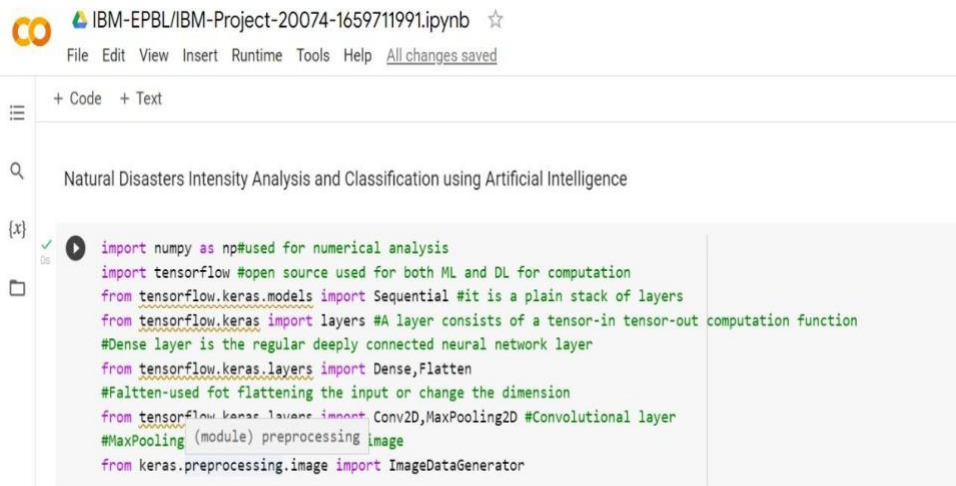
→ Data Preprocessing:

Data preprocessing can refer to manipulation or dropping of data before it is used in order to ensure or enhance performance, and is an important step in the data mining process.

Import the ImageDataGenerator library

Configure ImageDataGenerator class

Apply ImageDataGenerator functionality to Trainset and Testset



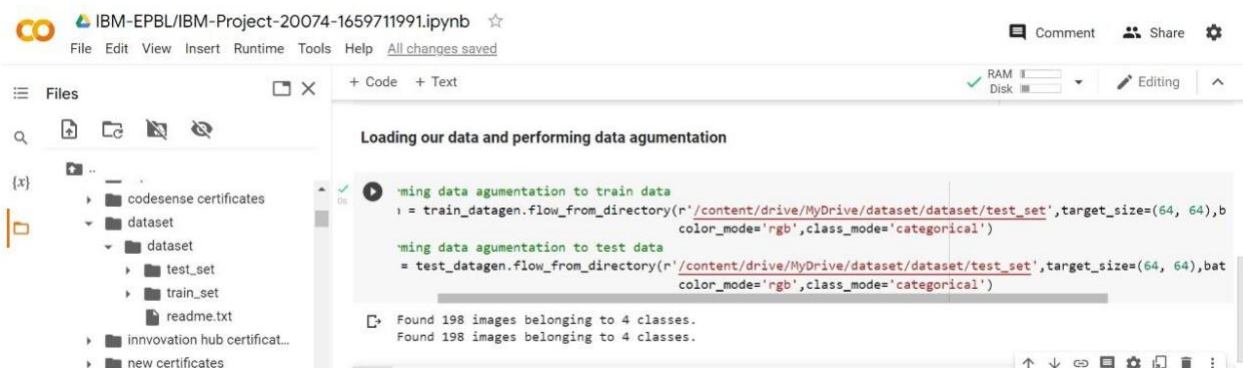
IBM-EPBL/IBM-Project-20074-1659711991.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Natural Disasters Intensity Analysis and Classification using Artificial Intelligence

```
import numpy as np #used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense, Flatten
#Flatten-used for flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D, MaxPooling2D #Convolutional layer
#MaxPooling (module) preprocessing image
from keras.preprocessing.image import ImageDataGenerator
```



IBM-EPBL/IBM-Project-20074-1659711991.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

RAM Disk Editing

Files

- codesense certificates
- dataset
 - test_set
 - train_set
 - readme.txt
- innovation hub certificat...
- new certificates

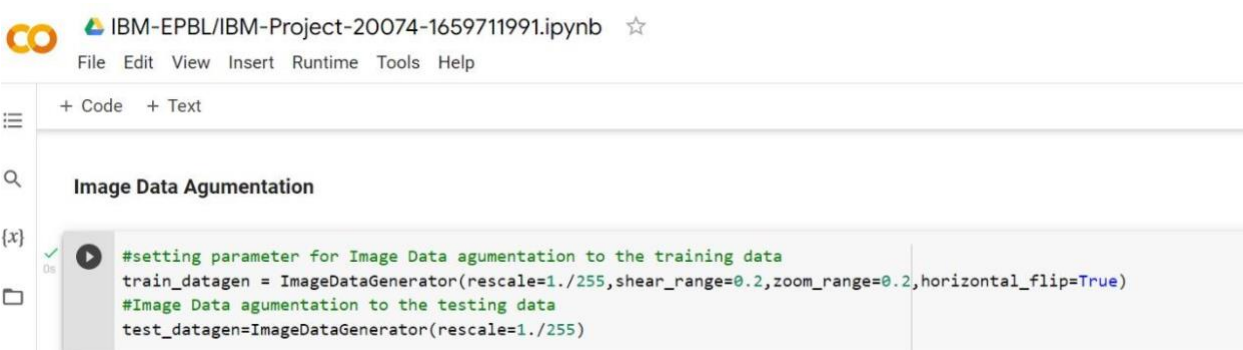
+ Code + Text

Loading our data and performing data agumentation

```
#adding data agumentation to train data
train_datagen = train_datagen.flow_from_directory(r'/content/drive/MyDrive/dataset/dataset/test_set', target_size=(64, 64), batch_size=32, color_mode='rgb', class_mode='categorical')

#adding data agumentation to test data
test_datagen = test_datagen.flow_from_directory(r'/content/drive/MyDrive/dataset/dataset/test_set', target_size=(64, 64), batch_size=32, color_mode='rgb', class_mode='categorical')
```

Found 198 images belonging to 4 classes.
Found 198 images belonging to 4 classes.



IBM-EPBL/IBM-Project-20074-1659711991.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Image Data Agumentation

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

#Image Data agumentation to the testing data
test_datagen = ImageDataGenerator(rescale=1./255)
```

→ Model Building:

- ◆ The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions.
- ◆ Import the model building Libraries

