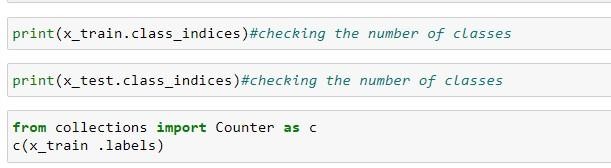
# PROJECT DEVELOPMENT PHASE SPRINT-III

|  |  |
| --- | --- |
| Date | 19 November 2022 |
| Team ID | PNT2022TMID28260 |
| Project Name | Natural Disaster Intensity Analysis and  Classification using Artificial Intelligence |

***DETECTION AND ANALYSIS OF DATA:***

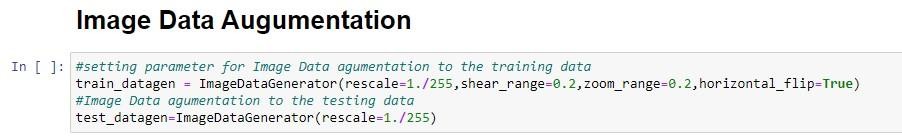
After Testing and Training the model, data which given in dataset are analysed and visualised effectively to detect the Disaster Type. Using webcam, it can capture image or video stream of Disaster, to detect and analyse the type of Disaster.



# IMAGE PREPROCESSING:

Image Pre-processing was done for Disaster intensity analysis and classification with three main tasks which includes for pre-processing of Images,

* Import ImageDataGenerator Library.
* Configure ImageDataGenerator Class.
* Applying ImageDataGenerator functionality to the trainset and test set.



# IMPORTING THE IMAGEDATAGENERATOR LIBRARY:

By importing the ImageDataGenerator Library can expand the train\_set data size using modified versions of dataset.

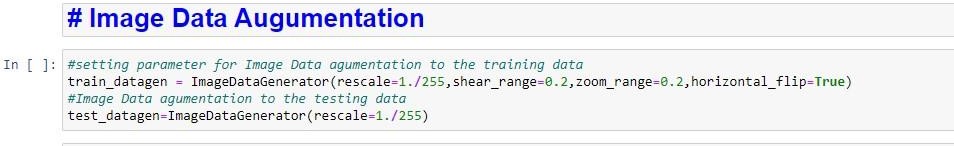
ImageDataGenerator class were importing from keras.



***CONFIGURE IMAGEDATAGENERATOR CLASS:***

ImageDataGenerator class is instantiated and the configuration for the types of data augmentation.

An instance of the ImageDataGenerator class can be constructed for train and test dataset by ImageDataGenerator class.

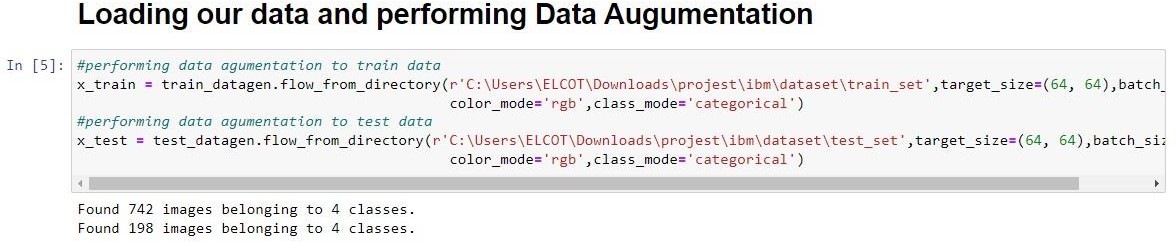


# APPLYING IMAGEDATAGENERATOR FUNCTIONALITY TO TRAINSET AND TESTSET

***:***

ImageDataGenerator functionality was applied to Trainset and Testset by using the following code,

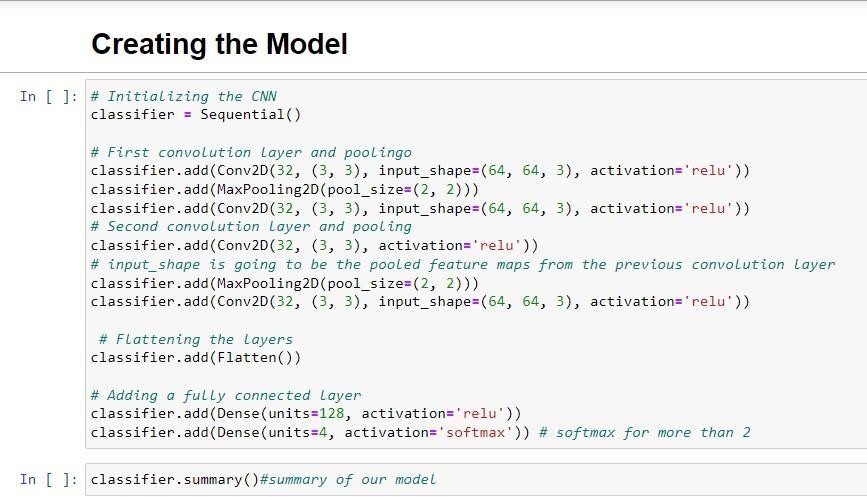
“For Training set using flow\_from\_directory function”.



# MODEL BUILDING:

Building a Model with web application named “FLASK”, model building process consist several steps like,

* + Import the model building Libraries
  + Initializing the model
  + Adding CNN Layers
  + Adding Hidden Layer
  + Adding Output Layer
  + Configure the Learning Process
  + Training and testing the model all the above processes are done and saved in a model.





# CREATING app.py:

