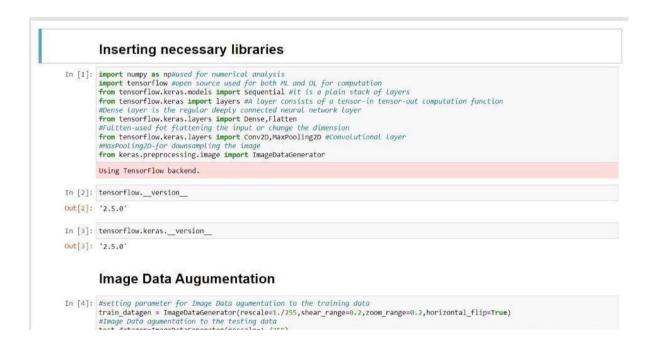
PROJECT DEVELOPMENT PHASE

SPRINT-III

Date	14 November 2022
Team ID	PNT2022TMID29508
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.



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.

```
In [1]: 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 its the regular deeply connected neural network layer from tensorflow.keras.layers import Dense, Flatten #foltten-used for flattening the input or change the dimension from tensorflow.keras.layers import ConvDs, MaxPoolingDb econvolutional layer #MAXPoolingDb for downsampling the image from keras.preprocessing.image import ImageDataGenerator

Using Tensorflow backend.

In [2]: tensorflow._version__

Out[2]: '2.5.0'

Image Data Augumentation

In [4]: #setting parameter for Image Data agumentation to the training data train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, horizontal_flip=True) #Image Data agumentation to the testing data test_datagen=maneshateGenerator(rescale=1./255, shear_range=0.2, horizontal_flip=True) #Image Data agumentation to the testing data test_datagen=maneshateGenerator(rescale=1./255, shear_range=0.2, horizontal_flip=True) #Image Data agumentation to the testing data test_datagen=maneshateGenerator(rescale=1./255)
```