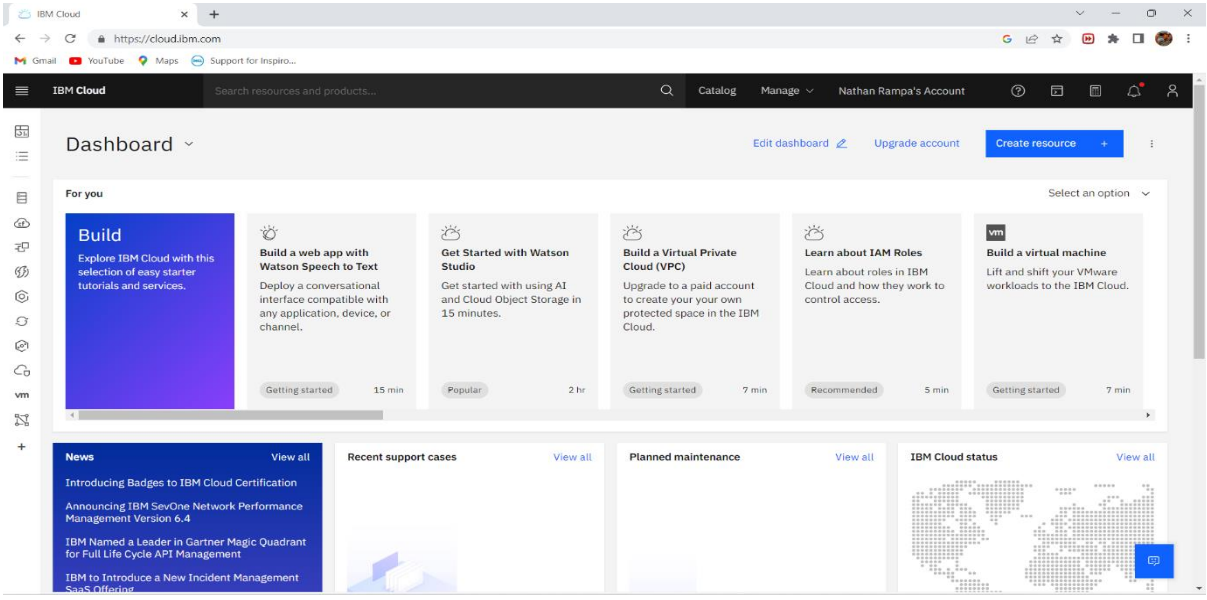
Train Model On IBM

TEAM ID: PNT2022TMID16547

PROJECT NAME: Natural Disasters Intensity Analysis and Classification using Artificial Intelligence



Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Build a web app with Watson Speech to Text

Deploy a conversational interface compatible with any application, device, or channel.

Getting started 15 min

Get Started with Watson Studio

Get started with using AI and Cloud Object Storage in 15 minutes.

Popular 2 hr

Build a Virtual Private Cloud (VPC)

Upgrade to a paid account to create your your own protected space in the IBM Cloud.

Getting started 7 min

Learn about IAM Roles

Learn about roles in IBM Cloud and how they work to control access.

Recommended 5 min

Build a virtual machine

Lift and shift your VMware workloads to the IBM Cloud.

Getting started 7 min

News View all

Introducing Badges to IBM Cloud Certification

Announcing IBM SevOne Network Performance Management Version 6.4

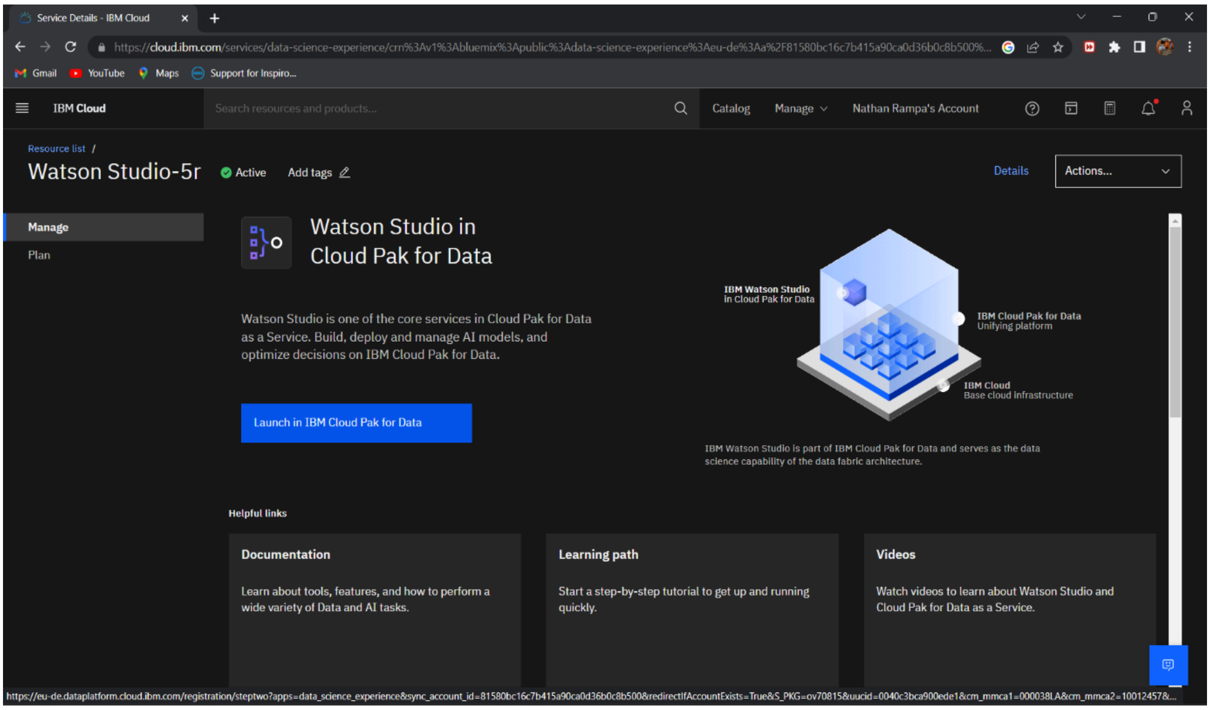
IBM Named a Leader in Gartner Magic Quadrant for Full Life Cycle API Management

IBM to Introduce a New Incident Management SaaS Offering

Recent support cases View all

Planned maintenance View all

IBM Cloud status View all


Service Details - IBM Cloud
https://cloud.ibm.com/services/data-science-experience/crn%3Av1%3Abluemix%3Apublic%3Adata-science-experience%3Aeu-de%3Aa%2F81580bc16c7b415a90ca0d36b0c8b500%...
Gmail
YouTube
Maps
Support for Inspiro...
IBM Cloud
Search resources and products...
Catalog
Manage
Nathan Rampa's Account
Resource list /
Details
Actions...
Manage
Plan
IBM Watson Studio
in Cloud Pak for Data
IBM Cloud Pak for Data
Unifying platform
IBM Cloud
Base cloud infrastructure

Watson Studio-5r ✓ Active Add tags

Watson Studio in Cloud Pak for Data

Watson Studio is one of the core services in Cloud Pak for Data as a Service. Build, deploy and manage AI models, and optimize decisions on IBM Cloud Pak for Data.

Launch in IBM Cloud Pak for Data

IBM Watson Studio is part of IBM Cloud Pak for Data and serves as the data science capability of the data fabric architecture.

Helpful links

Documentation

Learn about tools, features, and how to perform a wide variety of Data and AI tasks.

Learning path

Start a step-by-step tutorial to get up and running quickly.

Videos

Watch videos to learn about Watson Studio and Cloud Pak for Data as a Service.

https://eu-de.dataplatform.cloud.ibm.com/registration/steptwo?apps=data_science_experience&sync_account_id=81580bc16c7b415a90ca0d36b0c8b500&redirectIfAccountExists=True&S_PKG=ov70815&uucid=0040c3bca900ede1&cm_mmca1=000038LA&cm_mmca2=10012457&...

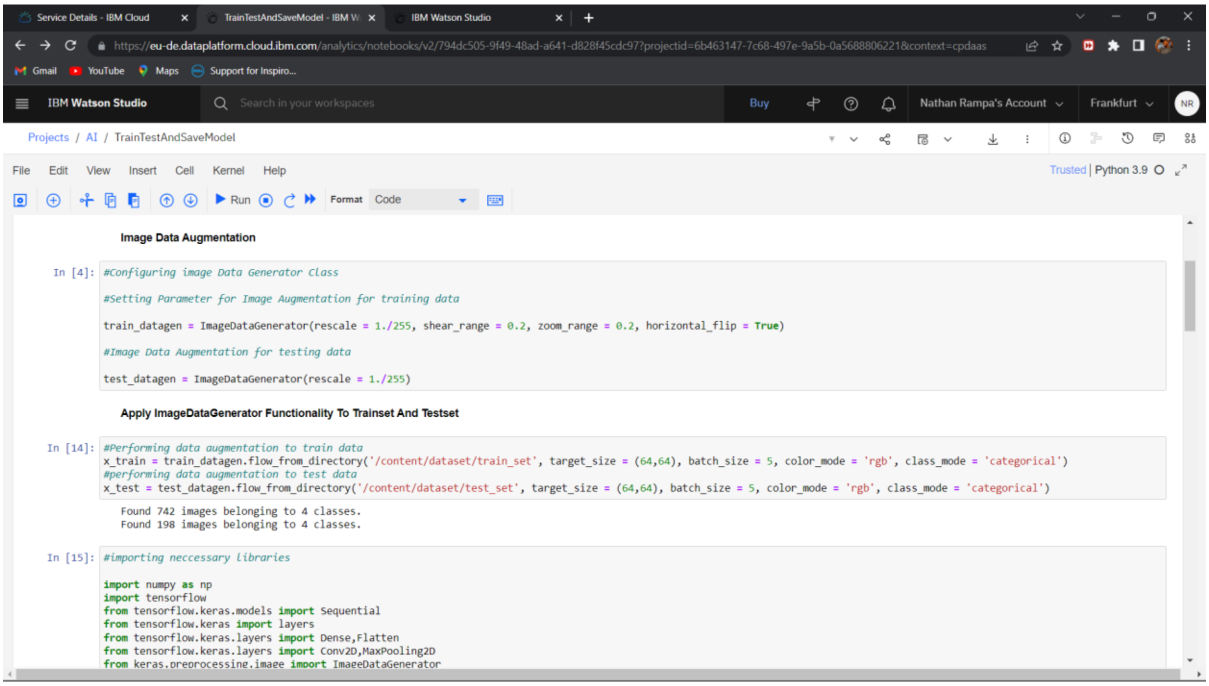

Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code

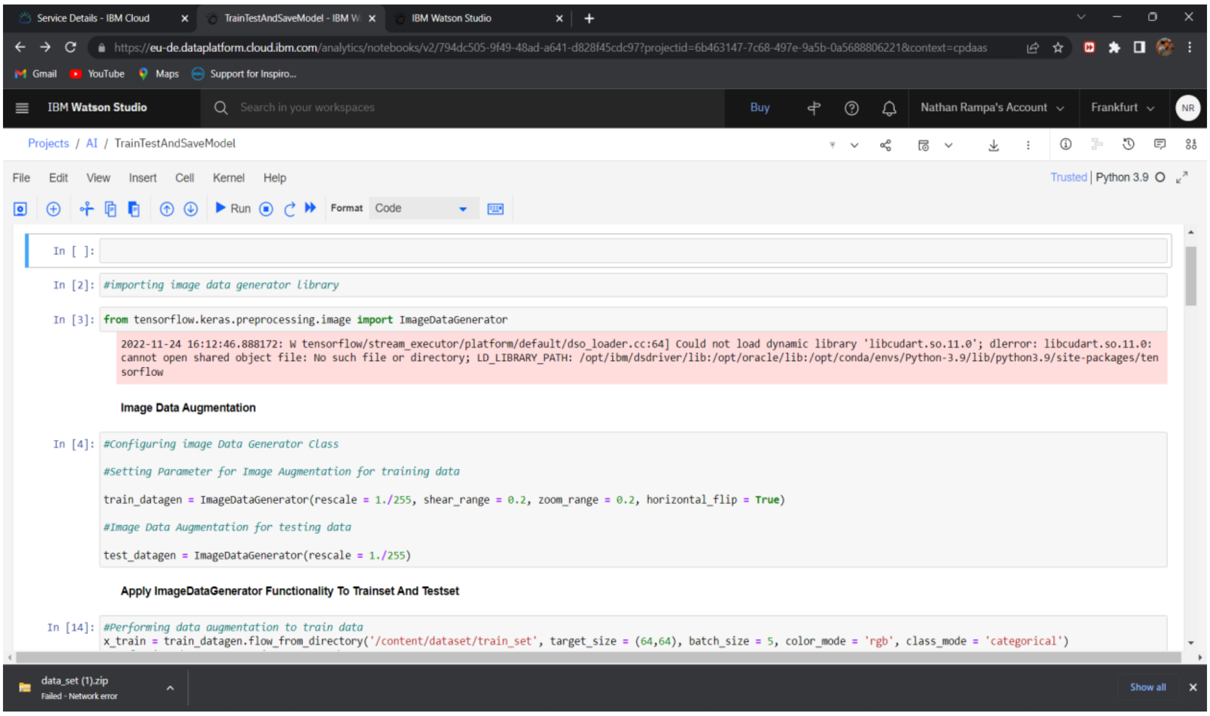
Image Data Augmentation

```
In [4]: #Configuring image Data Generator Class

        #Setting Parameter for Image Augmentation for training data

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

        #Image Data Augmentation for testing data

        test_datagen = ImageDataGenerator(rescale = 1./255)
```

Apply ImageDataGenerator Functionality To Trainset And Testset

```
In [14]: #Performing data augmentation to train data
         x_train = train_datagen.flow_from_directory('/content/dataset/train_set', target_size = (64,64), batch_size = 5, color_mode = 'rgb', class_mode = 'categorical')
         #performing data augmentation to test data
         x_test = test_datagen.flow_from_directory('/content/dataset/test_set', target_size = (64,64), batch_size = 5, color_mode = 'rgb', class_mode = 'categorical')

           Found 742 images belonging to 4 classes.
           Found 198 images belonging to 4 classes.
```

```
In [15]: #importing neccessary 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,MaxPooling2D
         from keras.preprocessing.image import ImageDataGenerator
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code
data_set (1).zip
Failed - Network error
Show all

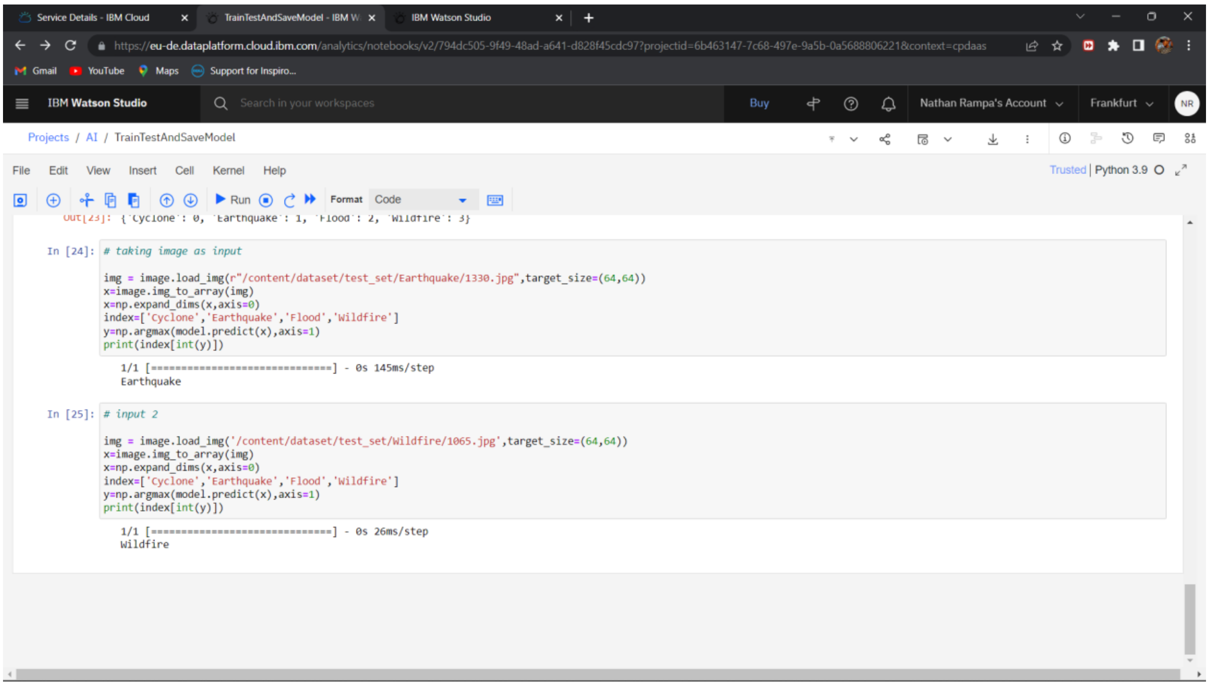
```
In [ ]:

In [2]: #importing image data generator library

In [3]: from tensorflow.keras.preprocessing.image import ImageDataGenerator

          2022-11-24 16:12:46.888172: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0:
          cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/ibm/dsdriver/lib:/opt/oracle/lib:/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ten
          sorflow
```

Image Data Augmentation

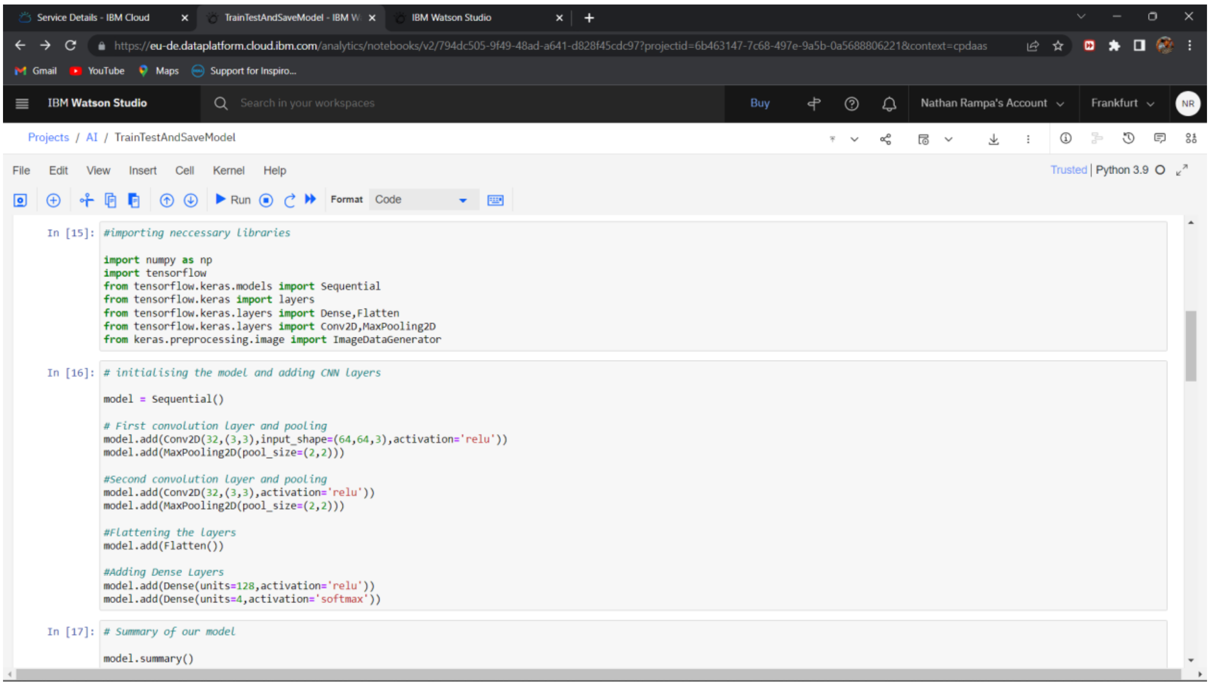
```
In [4]: #Configuring image Data Generator Class

        #Setting Parameter for Image Augmentation for training data

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

        #Image Data Augmentation for testing data

        test_datagen = ImageDataGenerator(rescale = 1./255)
```

Apply ImageDataGenerator Functionality To Trainset And Testset

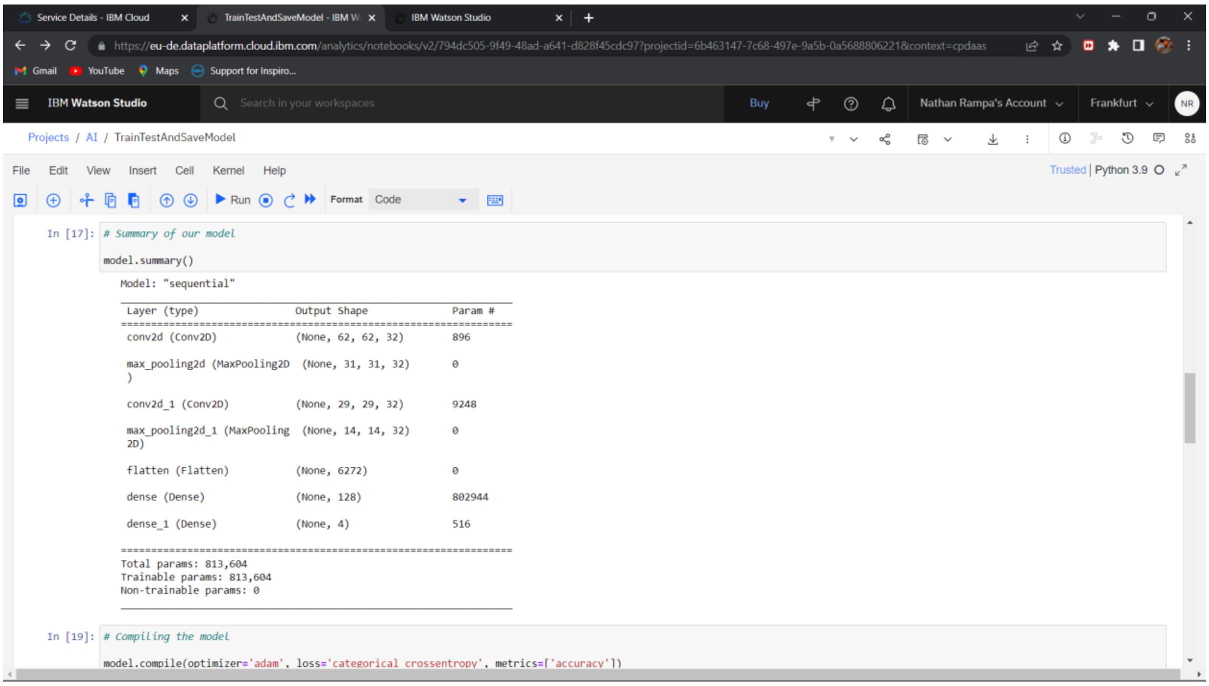
```
In [14]: #Performing data augmentation to train data
         x_train = train_datagen.flow_from_directory('/content/dataset/train_set', target_size = (64,64), batch_size = 5, color_mode = 'rgb', class_mode = 'categorical')
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted | Python 3.9
Run
Format
Code

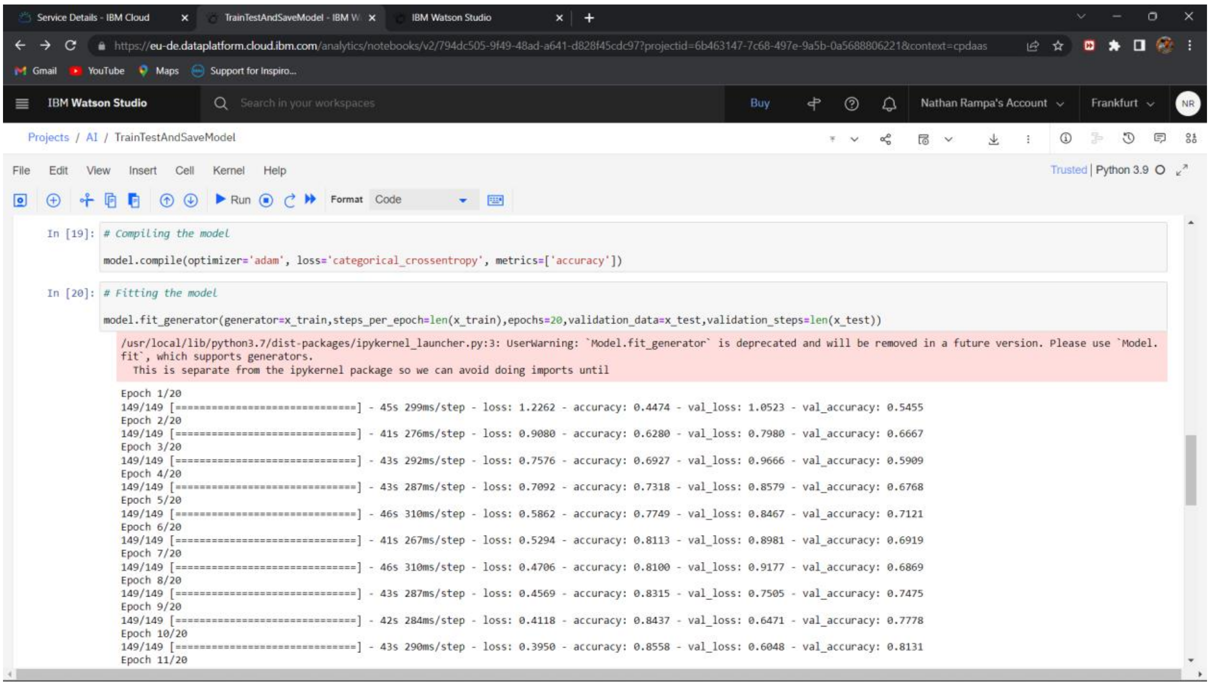
```
Out[23]: {'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}

In [24]: # taking image as input

         img = image.load_img(r"/content/dataset/test_set/Earthquake/1330.jpg",target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         index=['Cyclone','Earthquake','Flood','Wildfire']
         y=np.argmax(model.predict(x),axis=1)
         print(index[int(y)])

           1/1 [==============================] - 0s 145ms/step
           Earthquake

In [25]: # input 2

         img = image.load_img('/content/dataset/test_set/Wildfire/1065.jpg',target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         index=['Cyclone','Earthquake','Flood','Wildfire']
         y=np.argmax(model.predict(x),axis=1)
         print(index[int(y)])

           1/1 [==============================] - 0s 26ms/step
           Wildfire
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted | Python 3.9
Run
Format
Code

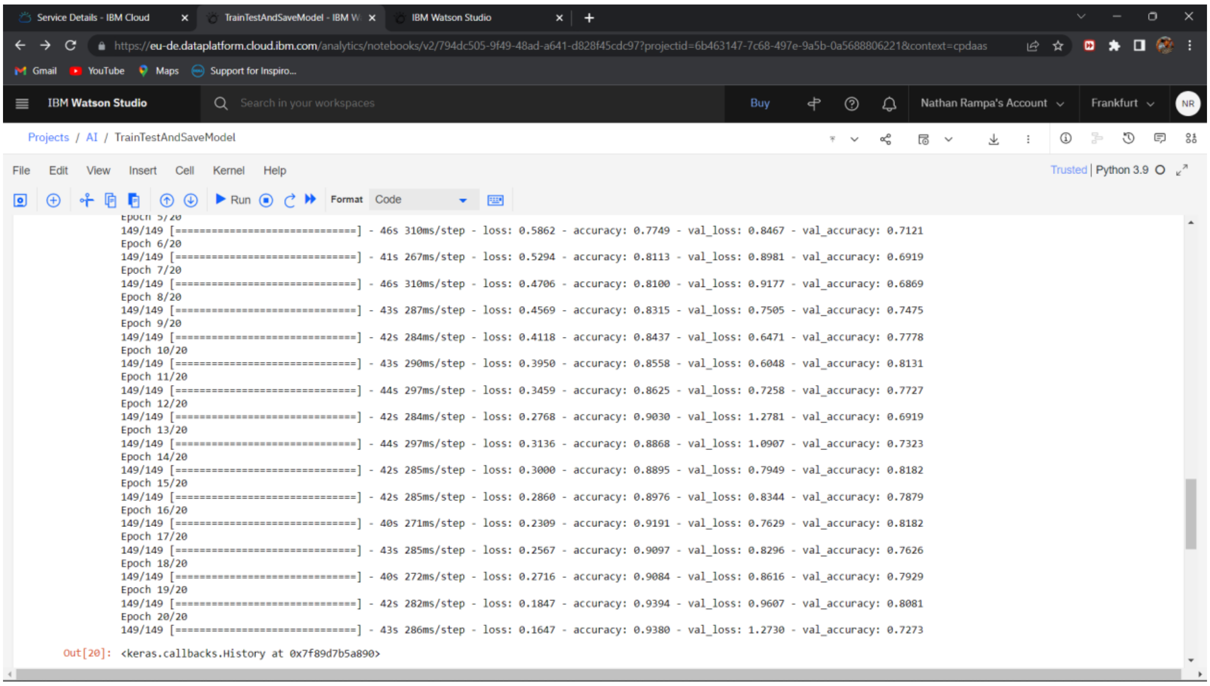
```
In [15]: #importing neccessary libraries

         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,MaxPooling2D
         from keras.preprocessing.image import ImageDataGenerator

In [16]: # initialising the model and adding CNN layers

         model = Sequential()

         # First convolution layer and pooling
         model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
         model.add(MaxPooling2D(pool_size=(2,2)))

         #Second convolution layer and pooling
         model.add(Conv2D(32,(3,3),activation='relu'))
         model.add(MaxPooling2D(pool_size=(2,2)))

         #Flattening the layers
         model.add(Flatten())

         #Adding Dense Layers
         model.add(Dense(units=128,activation='relu'))
         model.add(Dense(units=4,activation='softmax'))

In [17]: # Summary of our model

         model.summary()
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code

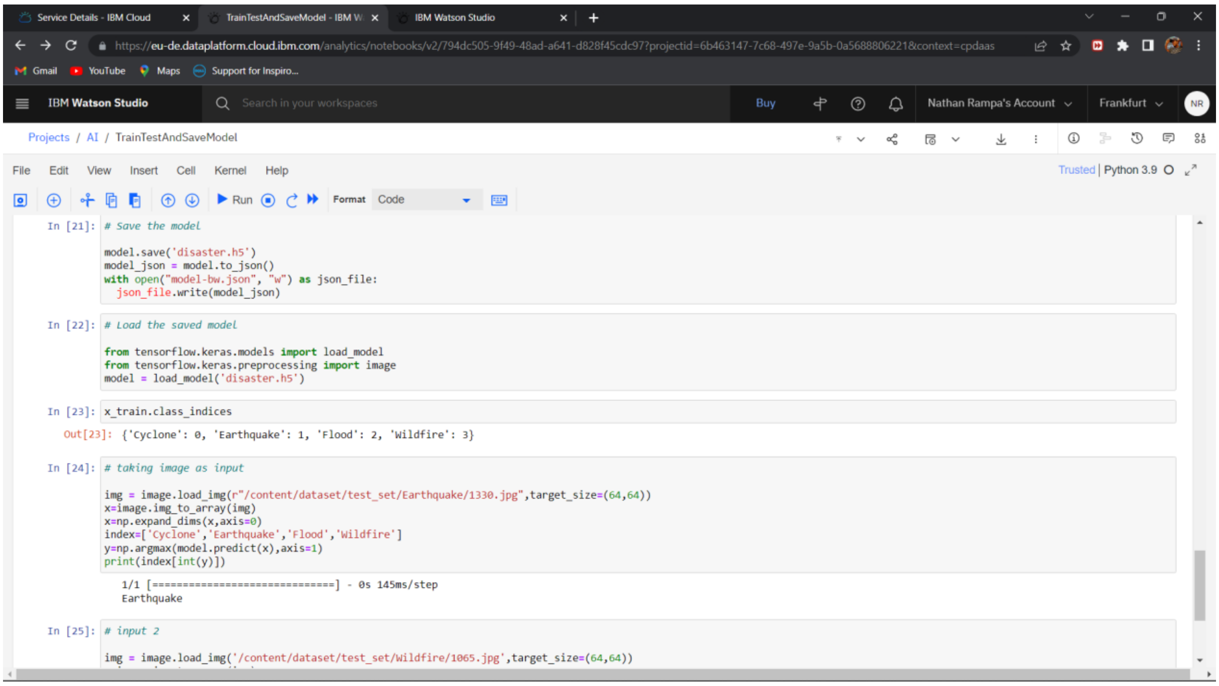
```
In [17]: # Summary of our model

         model.summary()

           Model: "sequential"
           _________________________________________________________________
            Layer (type)                Output Shape              Param #
           =================================================================
            conv2d (Conv2D)             (None, 62, 62, 32)        896

            max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
            )

            conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

            max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
            2D)

            flatten (Flatten)           (None, 6272)              0

            dense (Dense)               (None, 128)               802944

            dense_1 (Dense)             (None, 4)                 516

           =================================================================
           Total params: 813,604
           Trainable params: 813,604
           Non-trainable params: 0
           _________________________________________________________________

In [19]: # Compiling the model

         model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code

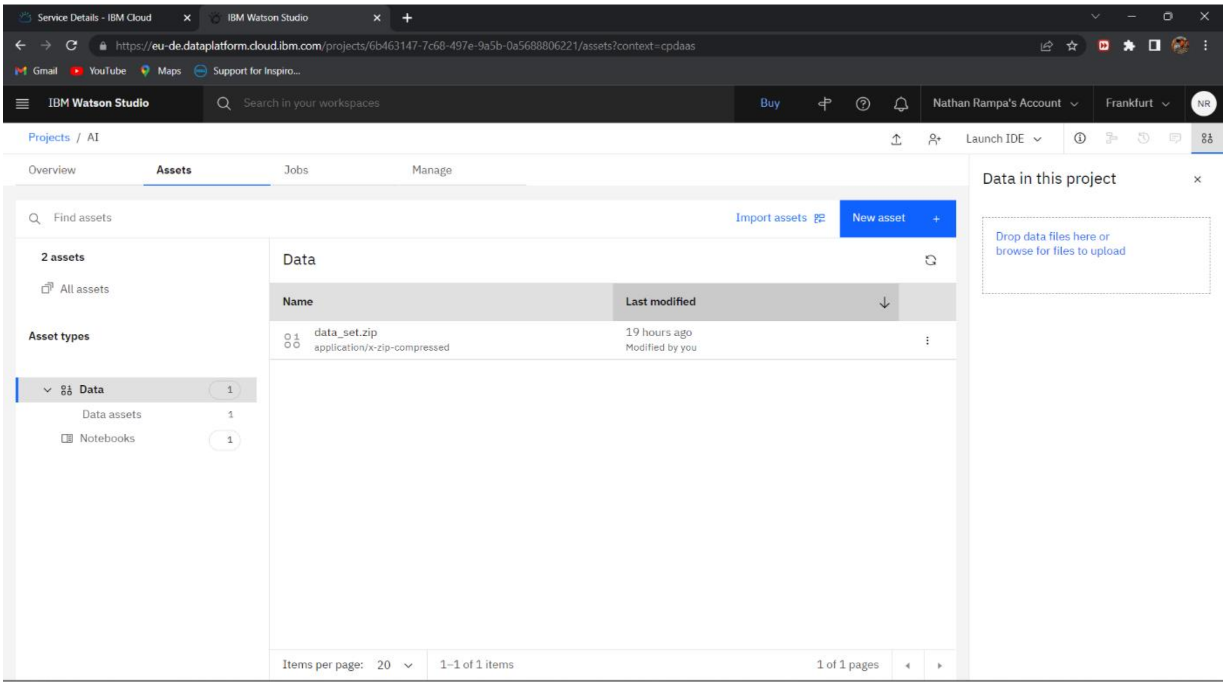
```
In [19]: # Compiling the model

         model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

In [20]: # Fitting the model

         model.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs=20,validation_data=x_test,validation_steps=len(x_test))

           /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.
           fit`, which supports generators.
             This is separate from the ipykernel package so we can avoid doing imports until

           Epoch 1/20
           149/149 [==============================] - 45s 299ms/step - loss: 1.2262 - accuracy: 0.4474 - val_loss: 1.0523 - val_accuracy: 0.5455
           Epoch 2/20
           149/149 [==============================] - 41s 276ms/step - loss: 0.9080 - accuracy: 0.6280 - val_loss: 0.7980 - val_accuracy: 0.6667
           Epoch 3/20
           149/149 [==============================] - 43s 292ms/step - loss: 0.7576 - accuracy: 0.6927 - val_loss: 0.9666 - val_accuracy: 0.5909
           Epoch 4/20
           149/149 [==============================] - 43s 287ms/step - loss: 0.7092 - accuracy: 0.7318 - val_loss: 0.8579 - val_accuracy: 0.6768
           Epoch 5/20
           149/149 [==============================] - 46s 310ms/step - loss: 0.5862 - accuracy: 0.7749 - val_loss: 0.8467 - val_accuracy: 0.7121
           Epoch 6/20
           149/149 [==============================] - 41s 267ms/step - loss: 0.5294 - accuracy: 0.8113 - val_loss: 0.8981 - val_accuracy: 0.6919
           Epoch 7/20
           149/149 [==============================] - 46s 310ms/step - loss: 0.4706 - accuracy: 0.8100 - val_loss: 0.9177 - val_accuracy: 0.6869
           Epoch 8/20
           149/149 [==============================] - 43s 287ms/step - loss: 0.4569 - accuracy: 0.8315 - val_loss: 0.7505 - val_accuracy: 0.7475
           Epoch 9/20
           149/149 [==============================] - 42s 284ms/step - loss: 0.4118 - accuracy: 0.8437 - val_loss: 0.6471 - val_accuracy: 0.7778
           Epoch 10/20
           149/149 [==============================] - 43s 290ms/step - loss: 0.3950 - accuracy: 0.8558 - val_loss: 0.6048 - val_accuracy: 0.8131
           Epoch 11/20
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code

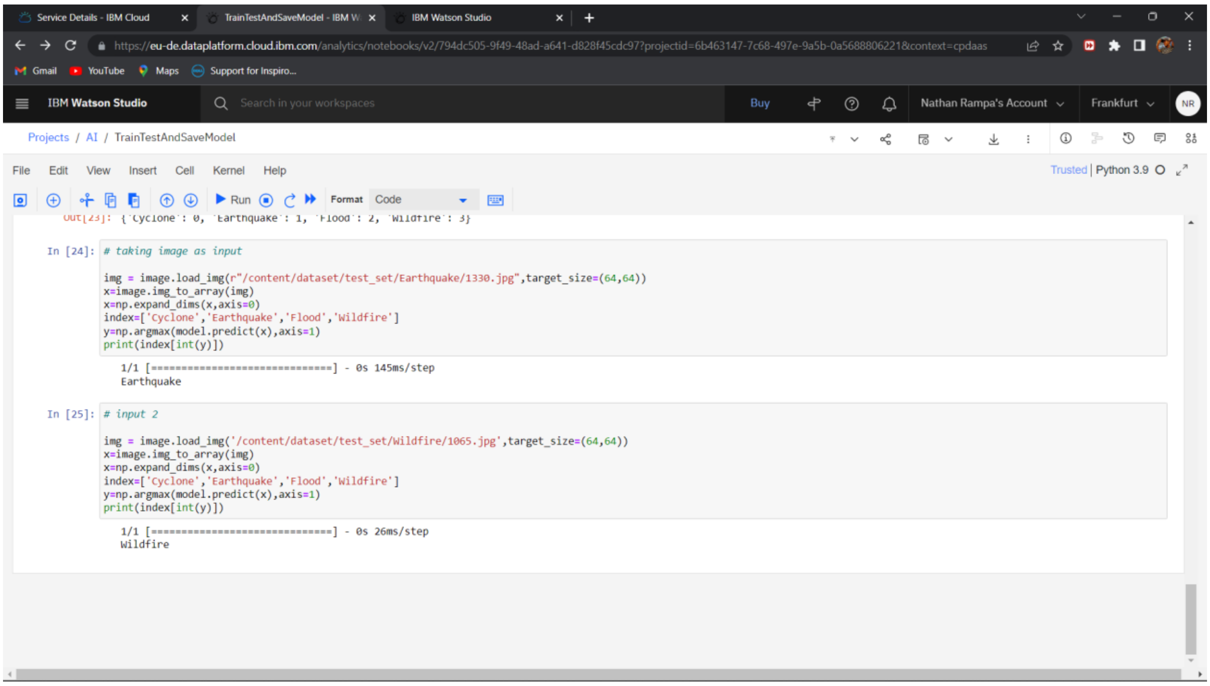
```
Epoch 5/20
149/149 [==============================] - 46s 310ms/step - loss: 0.5862 - accuracy: 0.7749 - val_loss: 0.8467 - val_accuracy: 0.7121
Epoch 6/20
149/149 [==============================] - 41s 267ms/step - loss: 0.5294 - accuracy: 0.8113 - val_loss: 0.8981 - val_accuracy: 0.6919
Epoch 7/20
149/149 [==============================] - 46s 310ms/step - loss: 0.4706 - accuracy: 0.8100 - val_loss: 0.9177 - val_accuracy: 0.6869
Epoch 8/20
149/149 [==============================] - 43s 287ms/step - loss: 0.4569 - accuracy: 0.8315 - val_loss: 0.7505 - val_accuracy: 0.7475
Epoch 9/20
149/149 [==============================] - 42s 284ms/step - loss: 0.4118 - accuracy: 0.8437 - val_loss: 0.6471 - val_accuracy: 0.7778
Epoch 10/20
149/149 [==============================] - 43s 290ms/step - loss: 0.3950 - accuracy: 0.8558 - val_loss: 0.6048 - val_accuracy: 0.8131
Epoch 11/20
149/149 [==============================] - 44s 297ms/step - loss: 0.3459 - accuracy: 0.8625 - val_loss: 0.7258 - val_accuracy: 0.7727
Epoch 12/20
149/149 [==============================] - 42s 284ms/step - loss: 0.2768 - accuracy: 0.9030 - val_loss: 1.2781 - val_accuracy: 0.6919
Epoch 13/20
149/149 [==============================] - 44s 297ms/step - loss: 0.3136 - accuracy: 0.8868 - val_loss: 1.0907 - val_accuracy: 0.7323
Epoch 14/20
149/149 [==============================] - 42s 285ms/step - loss: 0.3000 - accuracy: 0.8895 - val_loss: 0.7949 - val_accuracy: 0.8182
Epoch 15/20
149/149 [==============================] - 42s 285ms/step - loss: 0.2860 - accuracy: 0.8976 - val_loss: 0.8344 - val_accuracy: 0.7879
Epoch 16/20
149/149 [==============================] - 40s 271ms/step - loss: 0.2309 - accuracy: 0.9191 - val_loss: 0.7629 - val_accuracy: 0.8182
Epoch 17/20
149/149 [==============================] - 43s 285ms/step - loss: 0.2567 - accuracy: 0.9097 - val_loss: 0.8296 - val_accuracy: 0.7626
Epoch 18/20
149/149 [==============================] - 40s 272ms/step - loss: 0.2716 - accuracy: 0.9084 - val_loss: 0.8616 - val_accuracy: 0.7929
Epoch 19/20
149/149 [==============================] - 42s 282ms/step - loss: 0.1847 - accuracy: 0.9394 - val_loss: 0.9607 - val_accuracy: 0.8081
Epoch 20/20
149/149 [==============================] - 43s 286ms/step - loss: 0.1647 - accuracy: 0.9380 - val_loss: 1.2730 - val_accuracy: 0.7273

Out[20]: <keras.callbacks.History at 0x7f89d7b5a890>
```


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code

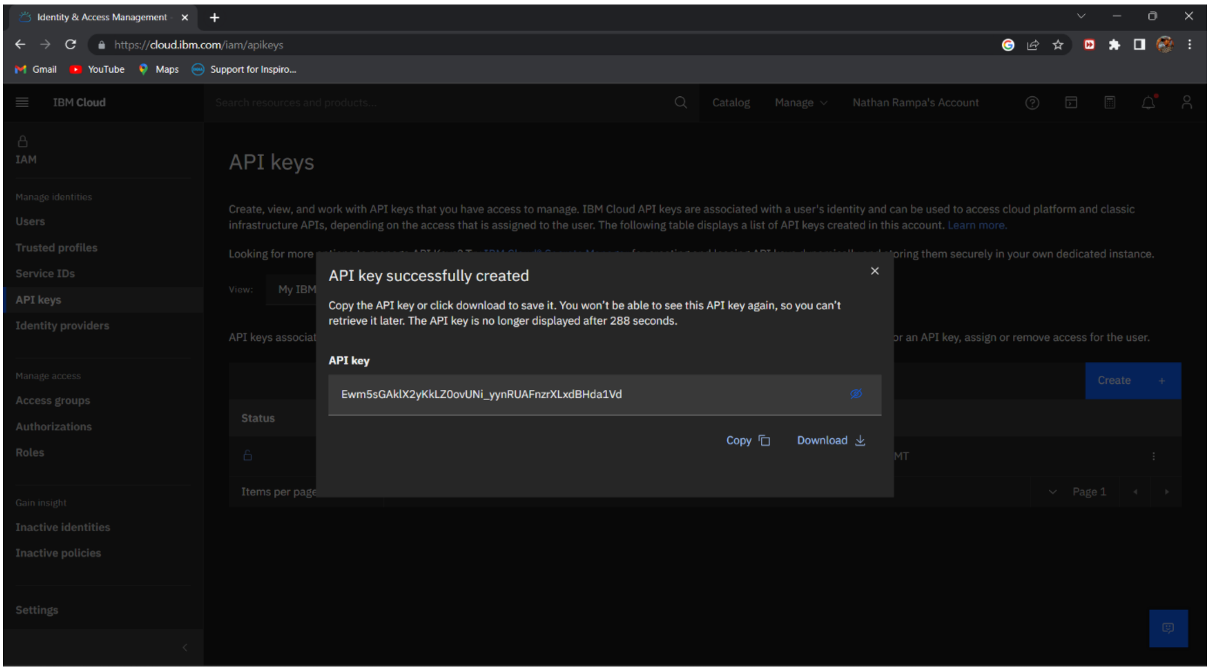
```
In [21]: # Save the model

         model.save('disaster.h5')
         model_json = model.to_json()
         with open("model-bw.json", "w") as json_file:
           json_file.write(model_json)

In [22]: # Load the saved model

         from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image
         model = load_model('disaster.h5')

In [23]: x_train.class_indices

Out[23]: {'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}

In [24]: # taking image as input

         img = image.load_img(r"/content/dataset/test_set/Earthquake/1330.jpg",target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         index=['Cyclone','Earthquake','Flood','Wildfire']
         y=np.argmax(model.predict(x),axis=1)
         print(index[int(y)])

           1/1 [==============================] - 0s 145ms/step
           Earthquake

In [25]: # input 2

         img = image.load_img('/content/dataset/test_set/Wildfire/1065.jpg',target_size=(64,64))
```



Name	Last modified
data_set.zip application/x-zip-compressed	19 hours ago Modified by you


Service Details - IBM Cloud
TrainTestAndSaveModel - IBM W
IBM Watson Studio
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/794dc505-9f49-48ad-a641-d828f45cdc97?projectid=6b463147-7c68-497e-9a5b-0a5688806221&context=cpdaas
Gmail
YouTube
Maps
Support for Inspiro...
IBM Watson Studio
Search in your workspaces
Buy
Nathan Rampa's Account
Frankfurt
NR
Projects / AI / TrainTestAndSaveModel
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3.9
Run
Format
Code

```
Out[23]: {'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}

In [24]: # taking image as input

         img = image.load_img(r"/content/dataset/test_set/Earthquake/1330.jpg",target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         index=['Cyclone','Earthquake','Flood','Wildfire']
         y=np.argmax(model.predict(x),axis=1)
         print(index[int(y)])

           1/1 [==============================] - 0s 145ms/step
           Earthquake

In [25]: # input 2

         img = image.load_img('/content/dataset/test_set/Wildfire/1065.jpg',target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         index=['Cyclone','Earthquake','Flood','Wildfire']
         y=np.argmax(model.predict(x),axis=1)
         print(index[int(y)])

           1/1 [==============================] - 0s 26ms/step
           Wildfire
```



API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

Looking for more … oring them securely in your own dedicated instance.

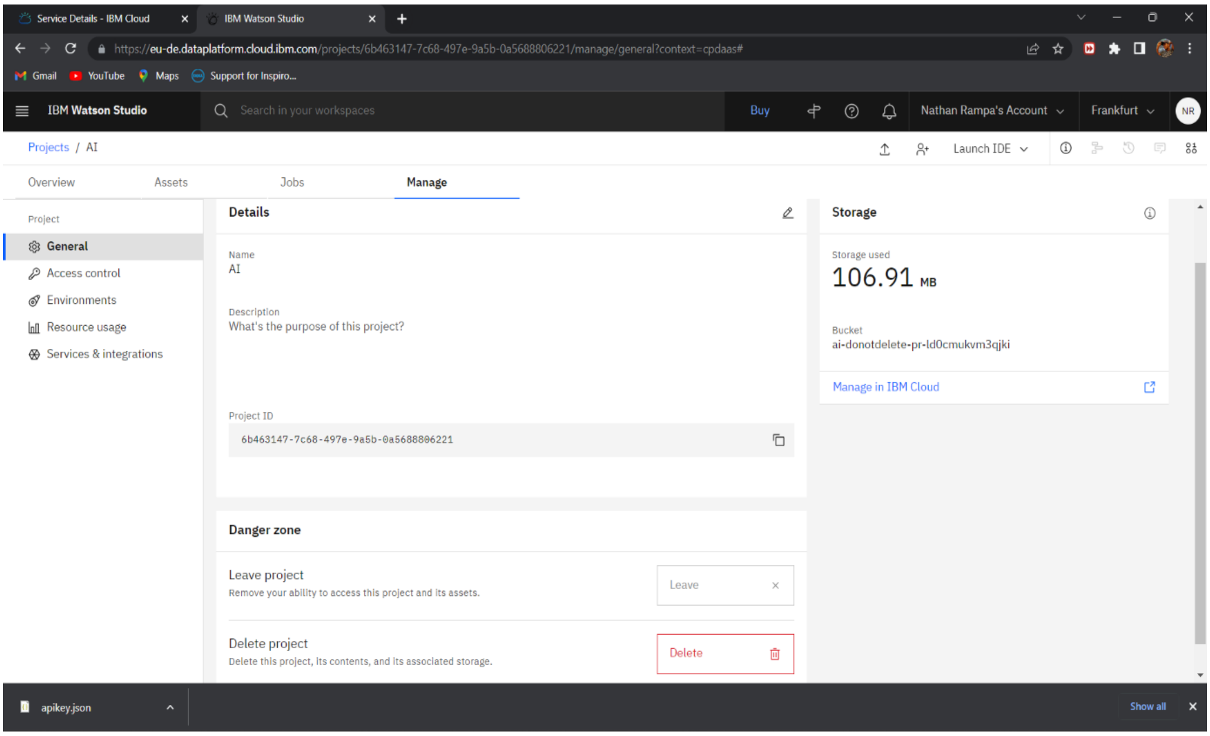
API keys associat… or an API key, assign or remove access for the user.

API key successfully created

Copy the API key or click download to save it. You won't be able to see this API key again, so you can't retrieve it later. The API key is no longer displayed after 288 seconds.

API key

Ewm5sGAklX2yKkLZ0ovUNi_yynRUAFnzrXLxdBHda1Vd



Details

Name
AI

Description
What's the purpose of this project?

Project ID
6b463147-7c68-497e-9a5b-0a5688806221

Storage

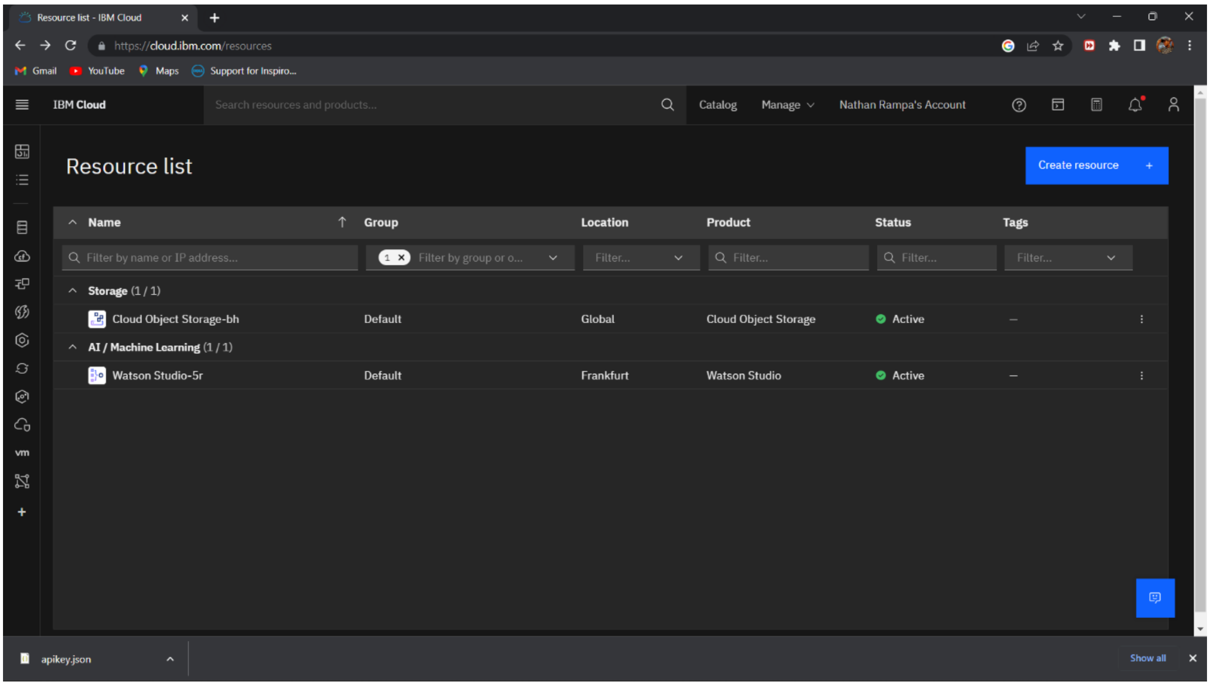
Storage used
106.91 MB

Bucket
ai-donotdelete-pr-ld0cmukvm3qjki

Danger zone

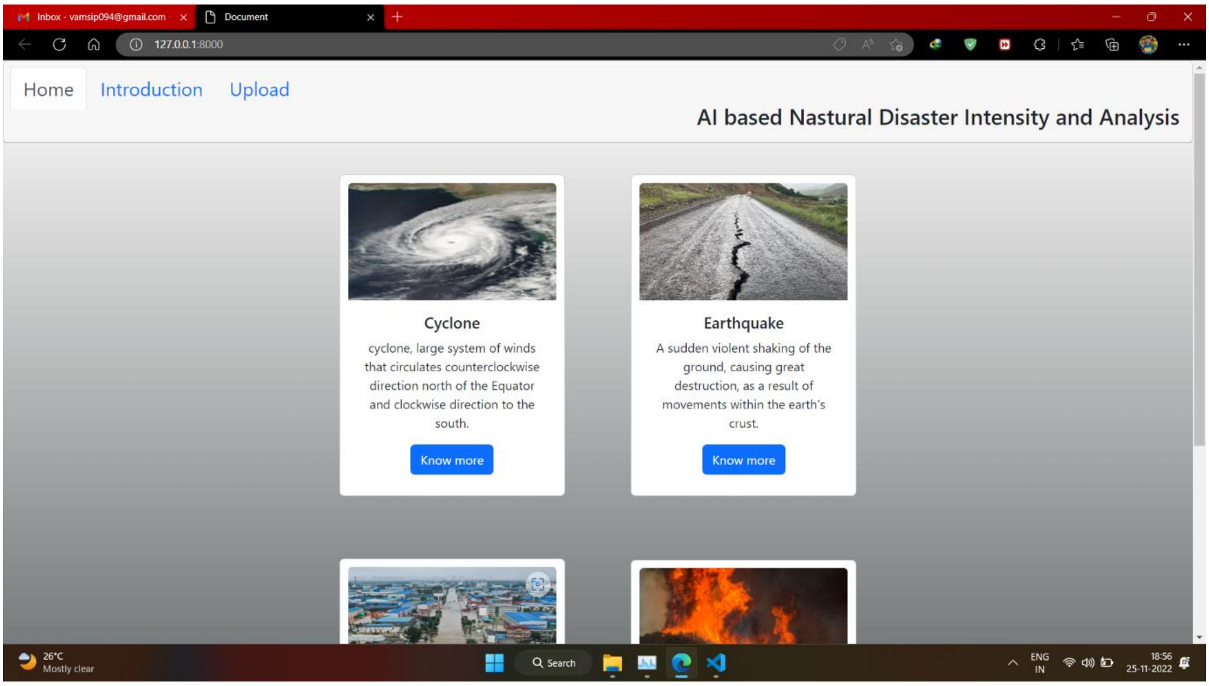
Leave project
Remove your ability to access this project and its assets.

Delete project
Delete this project, its contents, and its associated storage.



Resource list

Name	Group	Location	Product	Status	Tags
Storage (1 / 1)					
Cloud Object Storage-bh	Default	Global	Cloud Object Storage	Active	–
AI / Machine Learning (1 / 1)					
Watson Studio-5r	Default	Frankfurt	Watson Studio	Active	–



AI based Nastural Disaster Intensity and Analysis

Cyclone

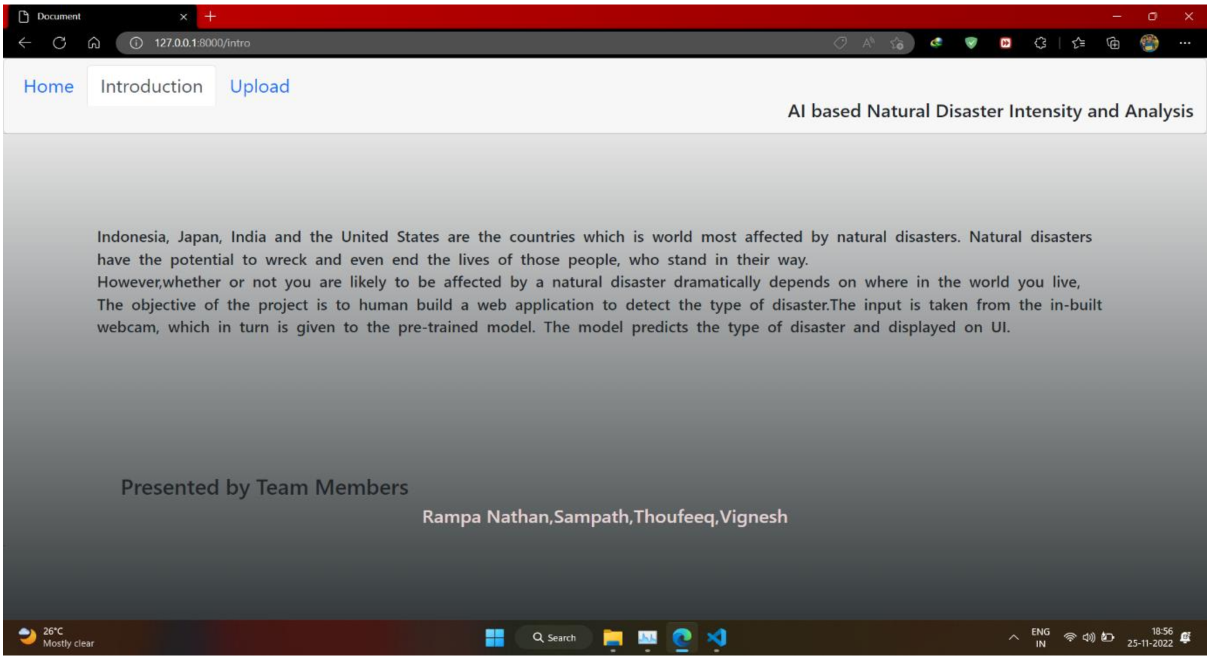
cyclone, large system of winds that circulates counterclockwise direction north of the Equator and clockwise direction to the south.

Know more

Earthquake

A sudden violent shaking of the ground, causing great destruction, as a result of movements within the earth's crust.

Know more

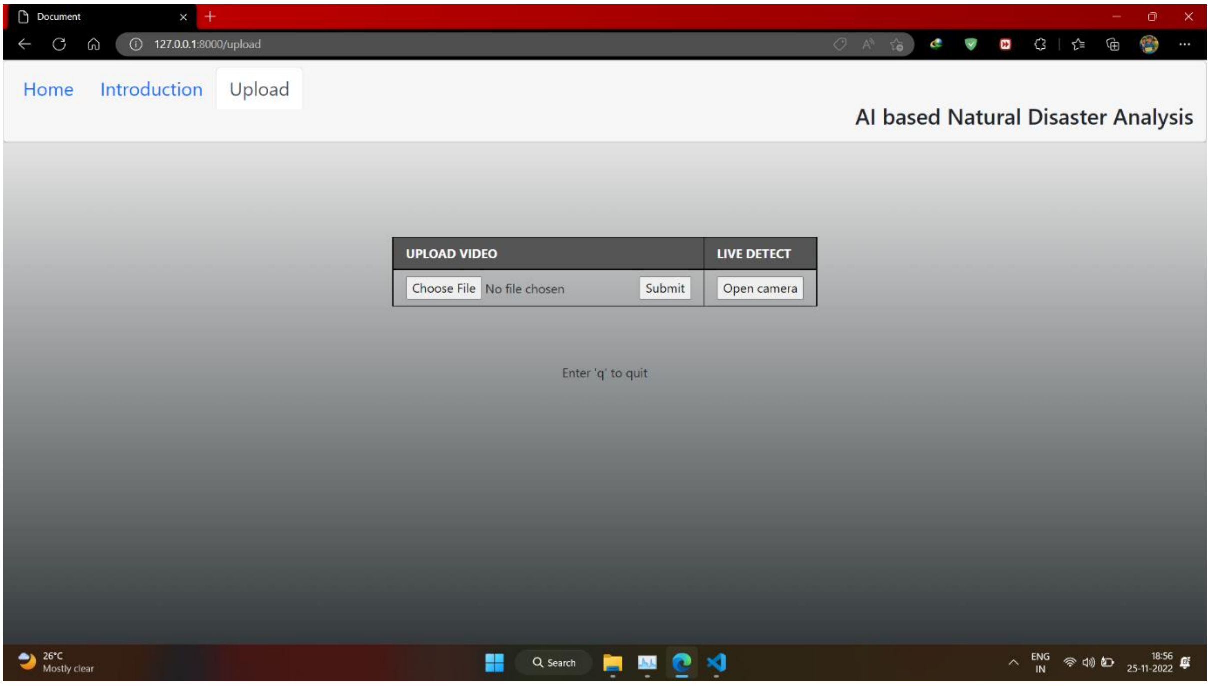


AI based Natural Disaster Intensity and Analysis

Indonesia, Japan, India and the United States are the countries which is world most affected by natural disasters. Natural disasters have the potential to wreck and even end the lives of those people, who stand in their way.
However,whether or not you are likely to be affected by a natural disaster dramatically depends on where in the world you live,
The objective of the project is to human build a web application to detect the type of disaster.The input is taken from the in-built webcam, which in turn is given to the pre-trained model. The model predicts the type of disaster and displayed on UI.

Presented by Team Members

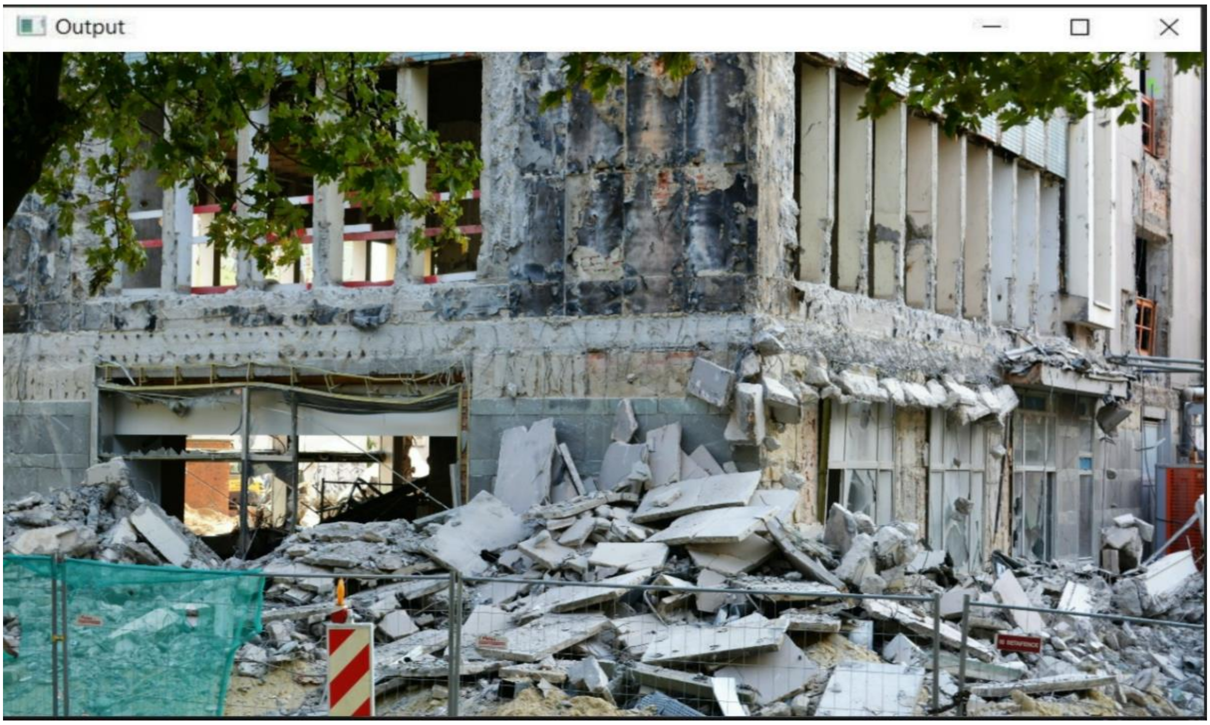
Rampa Nathan,Sampath,Thoufeeq,Vignesh



AI based Natural Disaster Analysis

UPLOAD VIDEO	LIVE DETECT
Choose File No file chosen Submit	Open camera

Enter 'q' to quit


Output