New section

```
from google.colab
import drivedrive.
mount('/content/drive')
!unzip  drive/My\Drive/dataset.zip
```

```
            inflating:   dataset/test_set/Flood/1032.jpg
            inflating:   dataset/test_set/Flood/1033.jpg
            inflating:   dataset/test_set/Flood/1034.jpg
            inflating:   dataset/test_set/Flood/1035.jpg
            inflating:   dataset/test_set/Flood/1036.jpg
            inflating:   dataset/test_set/Flood/1037.jpg
            inflating:   dataset/test_set/Flood/1038.jpg
            inflating:   dataset/test_set/Flood/1039.jpg
            inflating:   dataset/test_set/Flood/1040.jpg
            inflating:   dataset/test_set/Flood/1041.jpg
            inflating:   dataset/test_set/Flood/1042.jpg
            inflating:   dataset/test_set/Flood/1043.jpg
            inflating:   dataset/test_set/Flood/1044.jpg
            inflating:   dataset/test_set/Flood/1045.jpg
            inflating:   dataset/test_set/Flood/1046.jpg
            inflating:   dataset/test_set/Flood/1047.jpg
            inflating:   dataset/test_set/Flood/1048.jpg
            inflating:   dataset/test_set/Flood/1049.jpg
            inflating:   dataset/test_set/Flood/1050.jpg
            inflating:   dataset/test_set/Flood/1051.jpg
            inflating:   dataset/test_set/Flood/1062.jpg
            inflating:  dataset/test_set/Flood/992.jpg

            inflating:   dataset/test_set/Flood/993.jpg
            inflating:   dataset/test_set/Flood/994.jpg
            inflating:   dataset/test_set/Flood/995.jpg
            inflating:   dataset/test_set/Flood/996.jpg
            inflating:   dataset/test_set/Flood/997.jpg
            inflating:   dataset/test_set/Flood/998.jpg
            inflating:
            dataset/test_set/Flood/999.jpg
              creating:  dataset/test_set/Wildfire/
            inflating:     dataset/test_set/Wildfire/1035.jpg
            inflating:     dataset/test_set/Wildfire/1036.jpg
            inflating:     dataset/test_set/Wildfire/1037.jpg
            inflating:     dataset/test_set/Wildfire/1038.jpg
            inflating:     dataset/test_set/Wildfire/1039.jpg
            inflating:     dataset/test_set/Wildfire/1040.jpg
            inflating:     dataset/test_set/Wildfire/1041.jpg
            inflating:     dataset/test_set/Wildfire/1042.jpg
```

```
inflating:     dataset/test_set/Wildfire/1043.jpg
inflating:     dataset/test_set/Wildfire/1044.jpg
inflating:     dataset/test_set/Wildfire/1045.jpg
inflating:     dataset/test_set/Wildfire/1046.jpg
inflating: dataset/test_set/Wildfire/1047.jpg

inflating:     dataset/test_set/Wildfire/1048.jpg
inflating:     dataset/test_set/Wildfire/1049.jpg
inflating:     dataset/test_set/Wildfire/1050.jpg
inflating:     dataset/test_set/Wildfire/1051.jpg
inflating:     dataset/test_set/Wildfire/1052.jpg
inflating:     dataset/test_set/Wildfire/1053.jpg
inflating:     dataset/test_set/Wildfire/1054.jpg
inflating:     dataset/test_set/Wildfire/1055.jpg
inflating:     dataset/test_set/Wildfire/1056.jpg
inflating:     dataset/test_set/Wildfire/1057.jpg
inflating:     dataset/test_set/Wildfire/1058.jpg
inflating:     dataset/test_set/Wildfire/1059.jpg
inflating:     dataset/test_set/Wildfire/1060.jpg
inflating:     dataset/test_set/Wildfire/1061.jpg
inflating: dataset/test_set/Wildfire/1062.jpg
```

data augmentation

```
#   import   necessarylib.
from  tensorflow.keras.preprocessing.image  import  ImageDataGenerator


#image Data  Agumentation

#setting  parameter  for  Image  Data  agumentation  to  the  traing  data
train_datagen  =  ImageDataGenerator  (rescale=1./255,  shear_range=0.2,zoom_range=0.2,  horizo #Image

Data  agumentation  to  the  testing  data

test_datagen=ImageDataGenerator(rescale=1./255)


#Loading our data and performing data
agumentation#performing  data  agumentation  to
train  data

x_train  =  train_datagen.flow_from_directory('/content/dataset/train_set',target_size=(64, #performing  data

agumentation  to  test  data

x_test  =  test_datagen.flow_from_directory('/content/dataset/test_set',target_size=(64,  64)
```

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

Train test and save model

```
#Importing Neccessary Libraries

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 computatio

#Dense layer is the regular deeply connected neural network Layer

from tensorflow.keras.layers import Dense, Flatten

#Faltten-used fot flattening the input or change the dimension

from tensorflow.keras.layers import Conv2D, MaxPooling2D #Convolutional Layer

#MaxPooling20-for downsampling the image

from keras.preprocessing.image import ImageDataGenerator
```

Initializing the model

```
classifier=Sequential()

# First convolution layer and pooling

classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution layer and pooling

classifier.add(Conv2D(32, (3, 3), activation='relu'))

# input_shape is going to be the pooled feature maps from the previous convolution I

classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the Layers

classifier.add(Flatten())
```

Adding dense layers cnn

```
# Adding a fully connected Layer
```

```
classifier.add(Dense (units=128, activation='relu'))
classifier.add(Dense (units=4, activation='softmax')) #
softmax for more than 2
classifier. summary()
```

```
Model: "sequential_8"

 Layer   (type)                      Output Shape              Param #
=====================================================================
 conv2d_10 (Conv2D)                  (None, 62, 62, 32)         896

 max_pooling2d_6 (MaxPooling          (None, 31, 31, 32)       0
 2D)

 conv2d_11 (Conv2D)                  (None, 29, 29, 32)         9248

 max_pooling2d_7 (MaxPooling          (None, 14, 14, 32)       0
 2D)

 flatten_3 (Flatten)                     (None, 6272)          0

 dense_4  (Dense)                        (None, 128)           802944

 dense_5  (Dense)                        (None, 4)             516

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

```
#Compili

ng the

model#

Compilin

g the

CNN

# categorical_crossentropy for more than 2

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


#fitting the model

classifier.fit_generator( generator=x_train, steps_per_epoch = len(x_train), epochs=20, va
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: `MThis is
    separate from the ipykernel package so we can avoid doing imports unt
Epoch 1/20
```

149/149 [===============================]	- 28s	179ms/step	- loss:	1.1876	- accu
Epoch 2/20 149/149 [===============================]	- 26s	176ms/step	- loss:	0.8671	- accu
Epoch 3/20 149/149 [===============================]	- 26s	178ms/step	- loss:	0.7304	- accu
Epoch 4/20 149/149 [===============================]	- 27s	179ms/step	- loss:	0.7039	- accu
Epoch 5/20					
149/149 [===============================]	- 28s	190ms/step	- loss:	0.5969	- accu
Epoch 6/20 149/149 [===============================]	- 26s	175ms/step	- loss:	0.5413	- accu
Epoch 7/20 149/149 [===============================]	- 26s	177ms/step	- loss:	0.5225	- accu
Epoch 8/20					
149/149 [===============================]	- 28s	190ms/step	- loss:	0.4258	- accu
Epoch 9/20 149/149 [===============================]	- 27s	179ms/step	- loss:	0.4013	- accu
Epoch 10/20 149/149 [===============================]	- 30s	201ms/step	- loss:	0.3676	- accu
Epoch 11/20 149/149 [===============================]	- 26s	177ms/step	- loss:	0.4074	- accu
Epoch 12/20					
149/149 [===============================]	- 27s	180ms/step	- loss:	0.3413	- accu
Epoch 13/20 149/149 [===============================]	- 26s	176ms/step	- loss:	0.3183	- accu
Epoch 14/20					
149/149 [===============================] Epoch 15/20	- 26s	177ms/step	- loss:	0.2462	- accu
149/149 [===============================]	- 29s	198ms/step	- loss:	0.3194	- accu
Epoch 16/20 149/149 [===============================]	- 26s	177ms/step	- loss:	0.2228	- accu
Epoch 17/20					

149/149 [==============================]	**- 26s**	**177ms/step**	**- loss:**	**0.1697**	**- accu**
Epoch 18/20 149/149 [==============================]	**- 26s**	**176ms/step**	**- loss:**	**0.1958**	**- accu**
Epoch 19/20					
149/149 [==============================]	**- 26s**	**176ms/step**	**- loss:**	**0.2352**	**- accu**
Epoch 20/20 149/149 [==============================]	**- 26s**	**176ms/step**	**- loss:**	**0.1357**	**- accu**

```
<keras.callbacks.History at 0x7fb853040910>
```

Save the model as. h5

```
# Save the model

classifier.save('disaster.h5')

model_json = classifier.to_json()

with open("model-bw.json", "w") as json_file:

 json_file. write(model_json)
```

```
WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet
```

Testing the model

```
from tensorflow.keras.models import
load_modelfrom keras.preprocessing import
image
model = load_model("disaster.h5") #Loading the mode
```

Taking the image as input and checking the result

By using the model we are predicting the output for the given input image. The predicted classindex name will be printed here.

```
from tensorflow.keras.preprocessing import image

import numpy as np

img = image.load_img('/content/dataset/test_set/Flood/1009.jpg', target_size=(64,64)) img
#Loading of the image
```

```
x= image.img_to_array(img)x
#image to array
```

array([[[115.,	137.,	135.],	
[107.,	141.,	142.],	
[121.,	151.,	153.],	
..., [106.,	120.,	84.],	
[86.,	101.,	80.],	
[71.,	86.,	63.]],	
[[124.,	142.,	142.],	
[102.,	130.,	133.],	
[109.,	139.,	139.],	
..., [103.,	115.,	101.],	
[120.,	115.,	93.],	
[93.,	101.,	77.]],	
[[139.,	146.,	154.],	
[99.,	114.,	119.],	
[106.,	130.,	130.],	
..., [157.,	156.,	138.],	
[180.,	172.,	159.],	
[114.,	125.,	91.]],	
...,			
[[63.,	77.,	44.],	
[81.,	96.,	57.],	
[106., ...,	115.,	60.],	
[76.,	71.,	51.],	
[62.,	66.,	43.],	
[60.,	57.,	38.]],	
[[17.,	35.,	21.],	
[9.,	28.,	9.],	
[12., ..., [131.,	27., 113.,	8.], 67.],	
[92.,	86.,	62.],	
[95.,	92.,	75.]],	
[[106.,	133.,	114.],	
[94.,	109.,	90.],	

[77., ..., [88.,	94., 66.,	75.], 16.],	
[157.,	134.,	67.],	
[89.,	82.,	56.]]],	dtype=float32)

```
x =
np.expand_dims(x,axis
= 0)x
#changing the shape
```

array([[[[[115.,	137.,	135.],
[107.,	141.,	142.],
[121.,	151.,	153.],
..., [106.,	 120.,	 84.],
[86.,	101.,	80.],
[71.,	86.,	63.]]],

[[[124.,	142.,	142.],
[102.,	130.,	133.],
[109.,	139.,	139.],
..., [103.,	 115.,	 101.],
[120.,	115.,	93.],
[93.,	101.,	77.]]],

[[[139.,	146.,	154.],
[99.,	114.,	119.],
[106., ...,	130.,	130.],
[157.,	156.,	138.],
[180.,	172.,	159.],
[114.,	125.,	91.]]],

...,

[[[63.,	77.,	44.],
[81.,	96.,	57.],
[106.,	115.,	60.],
..., [76.,	 71.,	 51.],
[62.,	66.,	43.],
[60.,	57.,	38.]]],

[[[	17.,	35.,	21.],
[	9.,	28.,	9.],

```
        [  12.,      27.,      8.],
        ...,
        [131.,      113.,    67.],
        [ 92.,       86.,    62.],
        [ 95.,       92.,    75.]]],


      [[[106.,    133.,     114.],
          [ 94.,    109.,      90.],
          [ 77.,     94.,      75.],


        ...,
        [ 88.,       66.,     16.],
        [157., 134.,          67.],
        [  89.,      82.,     56.]]]]], dtype=float32)
```

```
from tensorflow.keras.preprocessing import image

import numpy as np

img = image.load_img('/content/dataset/test_set/Flood/1009.jpg', target_size=(64,64))
x= image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred  =  np.argmax(model.predict(x))
Output=['earthquake','cyclone','flood','wildfire']
Output[pred]
#predicting the class
```

```
    1/1 [==============================] - 0s 20ms/step
    'flood'
```