```json
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
   "colab": {
    "provenance": []
   },
   "kernelspec": {
    "name": "python3",
    "display_name": "Python 3"
   },
   "language_info": {
    "name": "python"
   }
  },
  "cells": [
   {
    "cell_type": "code",
    "execution_count": 1,
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "OIPBS3CsTuZs",
     "outputId": "ed1c6bf8-edc5-42dd-a0f7-3c64b96b57f8"
    },
    "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
```

```json
        "/usr/local/lib/python3.7/dist-packages/gdown/cli.py:131: FutureWarning: Option `--id` was deprecated in version 4.3.1 and will be removed in 5.0. You don't need to pass it anymore to use a file ID.\n",
        "  category=FutureWarning,\n",
        "Downloading...\n",
        "From: https://drive.google.com/uc?id=1npY_sDIDyQWjm2ZH4cCCuDhZA9liaNUm\n",
        "To: /content/dataset.zip\n",
        "100% 523M/523M [00:05<00:00, 98.0MB/s]\n"
      ]
    }
  ],
  "source": [
    "! gdown --id 1npY_sDIDyQWjm2ZH4cCCuDhZA9liaNUm"
  ]
},
{
  "cell_type": "code",
  "source": [
    "! unzip '/content/dataset.zip'"
  ],
  "metadata": {
    "id": "CZxNutu-UfPE"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "from tensorflow.keras.preprocessing.image import ImageDataGenerator"
  ],
  "metadata": {
```

```json
      "id": "-7AiozwqUnVK"
    },
    "execution_count": null,
    "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
     "#Data Augmentation"
   ],
   "metadata": {
     "id": "SSp_SLOGUsoP"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
     "train_datagen =
ImageDataGenerator(rescale=1./225,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)"
   ],
   "metadata": {
     "id": "jswWUyIPUwod"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
```

```
      "test_datagen = ImageDataGenerator(rescale=1./225)"
    ],
    "metadata": {
      "id": "rljamDTGU2lz"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "x_train = train_datagen.flow_from_directory('/content/dataset/train_set' ,
target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='categorical')"
    ],
    "metadata": {
      "id": "GUUbqoNJVFSh"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "markdown",
    "source": [
      "Found 742 images belonging to 4 classes.\n"
    ],
    "metadata": {
      "id": "usXQ_eiIVObh"
    }
  },
  {
    "cell_type": "code",
```

```
    "source": [

      "x_test = test_datagen.flow_from_directory('/content/dataset/test_set' ,
target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='categorical')\n"

    ],

    "metadata": {

      "id": "mEuVwe2dVFO_"

    },

    "execution_count": null,

    "outputs": []

  },

  {

    "cell_type": "markdown",

    "source": [

      "Found 198 images belonging to 4 classes."

    ],

    "metadata": {

      "id": "shAaI4w7VkLc"

    }

  },

  {

    "cell_type": "code",

    "source": [

      "#importing Required Packages"

    ],

    "metadata": {

      "id": "XetNtP4KVlWx"

    },

    "execution_count": null,

    "outputs": []

  },

  {
```

   "cell_type": "code",

   "source": [

    "from tensorflow.keras.models import Sequential\n",

    "from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense"

   ],

   "metadata": {

    "id": "CoNzTEu3VrrR"

   },

   "execution_count": null,

   "outputs": []

  },

  {

   "cell_type": "code",

   "source": [

    "#Building CNN block"

   ],

   "metadata": {

    "id": "MgELD1IDV2QY"

   },

   "execution_count": null,

   "outputs": []

  },

  {

   "cell_type": "code",

   "source": [

    "model = Sequential()\n",

    "model.add(Convolution2D(32,(3,3), input_shape=(64,64,3),activation='relu'))\n",

    "model.add(MaxPooling2D(pool_size=(2,2)))\n",

    "model.add(Convolution2D(32 ,(3,3) , activation='relu'))\n",

    "model.add(MaxPooling2D(pool_size=(2,2)))\n",

    "model.add(Flatten())\n",

```
    "model.add(Dense(units =128 , activation='relu'))\n",
    "model.add(Dense(units =4 , activation='softmax'))"
  ],
  "metadata": {
   "id": "SEhl4DA0V3Zp"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "#Compiling the model"
  ],
  "metadata": {
   "id": "ida-RJQ3WDZE"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "model.compile(optimizer='adam',loss='categorical_crossentropy' , metrics=['accuracy'])"
  ],
  "metadata": {
   "id": "v05lIC-oWHQj"
  },
  "execution_count": null,
  "outputs": []
 },
```

```
  {
    "cell_type": "code",

    "source": [

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

    ],

    "metadata": {

      "id": "1oVrLmduXTyC"

    },

    "execution_count": null,

    "outputs": []

  },

  {

    "cell_type": "markdown",

    "source": [

      "Epoch 1/20\n",

      "149/149 [==============================] - 37s 188ms/step - loss: 1.2177 - accuracy:
0.4367 - val_loss: 1.5993 - val_accuracy: 0.4646\n",

      "Epoch 2/20\n",

      "149/149 [==============================] - 28s 187ms/step - loss: 0.9553 - accuracy:
0.5943 - val_loss: 0.9020 - val_accuracy: 0.6515\n",

      "Epoch 3/20\n",

      "149/149 [==============================] - 28s 186ms/step - loss: 0.7419 - accuracy:
0.7116 - val_loss: 0.7201 - val_accuracy: 0.7424\n",

      "Epoch 4/20\n",

      "149/149 [==============================] - 28s 187ms/step - loss: 0.6761 - accuracy:
0.7345 - val_loss: 0.7405 - val_accuracy: 0.7576\n",

      "Epoch 5/20\n",

      "149/149 [==============================] - 28s 188ms/step - loss: 0.6356 - accuracy:
0.7480 - val_loss: 0.8016 - val_accuracy: 0.7424\n",

      "Epoch 6/20\n",

      "149/149 [==============================] - 30s 202ms/step - loss: 0.5759 - accuracy:
0.7749 - val_loss: 0.9617 - val_accuracy: 0.6869\n",
```

"Epoch 7/20\n",

"149/149 [==============================] - 28s 185ms/step - loss: 0.5246 - accuracy: 0.8181 - val_loss: 0.7854 - val_accuracy: 0.7071\n",

"Epoch 8/20\n",

"149/149 [==============================] - 28s 188ms/step - loss: 0.4662 - accuracy: 0.8248 - val_loss: 0.6588 - val_accuracy: 0.7273\n",

"Epoch 9/20\n",

"149/149 [==============================] - 28s 188ms/step - loss: 0.4304 - accuracy: 0.8302 - val_loss: 0.6534 - val_accuracy: 0.7727\n",

"Epoch 10/20\n",

"149/149 [==============================] - 28s 187ms/step - loss: 0.3771 - accuracy: 0.8544 - val_loss: 0.8804 - val_accuracy: 0.7222\n",

"Epoch 11/20\n",

"149/149 [==============================] - 28s 188ms/step - loss: 0.3379 - accuracy: 0.8733 - val_loss: 0.9850 - val_accuracy: 0.7222\n",

"Epoch 12/20\n",

"149/149 [==============================] - 28s 185ms/step - loss: 0.3635 - accuracy: 0.8464 - val_loss: 0.7546 - val_accuracy: 0.7727\n",

"Epoch 13/20\n",

"149/149 [==============================] - 28s 190ms/step - loss: 0.3426 - accuracy: 0.8733 - val_loss: 0.8590 - val_accuracy: 0.7222\n",

"Epoch 14/20\n",

"149/149 [==============================] - 28s 188ms/step - loss: 0.2759 - accuracy: 0.8949 - val_loss: 0.9976 - val_accuracy: 0.7374\n",

"Epoch 15/20\n",

"149/149 [==============================] - 28s 187ms/step - loss: 0.3028 - accuracy: 0.8854 - val_loss: 1.4439 - val_accuracy: 0.6313\n",

"Epoch 16/20\n",

"149/149 [==============================] - 28s 185ms/step - loss: 0.2939 - accuracy: 0.8949 - val_loss: 0.7897 - val_accuracy: 0.7576\n",

"Epoch 17/20\n",

"149/149 [==============================] - 29s 197ms/step - loss: 0.2254 - accuracy: 0.9191 - val_loss: 1.0229 - val_accuracy: 0.7677\n",

"Epoch 18/20\n",

      "149/149 [==============================] - 28s 187ms/step - loss: 0.2084 - accuracy: 0.9218 - val_loss: 1.0623 - val_accuracy: 0.7323\n",

      "Epoch 19/20\n",

      "149/149 [==============================] - 28s 186ms/step - loss: 0.1692 - accuracy: 0.9394 - val_loss: 1.0719 - val_accuracy: 0.7576\n",

      "Epoch 20/20\n",

      "149/149 [==============================] - 28s 186ms/step - loss: 0.1843 - accuracy: 0.9340 - val_loss: 0.9710 - val_accuracy: 0.7525\n"
    ],
   "metadata": {
    "id": "wG5IrrtcYX8Q"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "#Saving Model"
   ],
   "metadata": {
    "id": "PJdAmY1eYi5U"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "model.save('disaster.h5')"
   ],
   "metadata": {
    "id": "OaVRpWlsYrrQ"
   },

```json
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "#Testing the model"
    ],
    "metadata": {
     "id": "4tf2rx1JYvnq"
    },
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "import numpy as np\n",
     "from tensorflow.keras.preprocessing import image\n"
    ],
    "metadata": {
     "id": "MUHt3LhzYzZJ"
    },
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "img = image.load_img('/content/dataset/test_set/Flood/1015.jpg' , target_size=(64,64))\n",
     "x=image.img_to_array(img)\n",
```

```
    "x=np.expand_dims(x,axis=0)\n",

    "pred = np.argmax(model.predict(x))\n",

    "print(pred,model.predict(x))\n",

    "op=['Cyclone','Earthquake','Flood','Wildfire']\n",

    "print(op[pred])"
   ],
   "metadata": {

    "id": "56rxRxZtZDSc"
   },

   "execution_count": null,

   "outputs": []
  },
  {

   "cell_type": "markdown",

   "source": [

    "1/1 [==============================] - 0s 124ms/step\n",

    "1/1 [==============================] - 0s 16ms/step\n",

    "2 [[0. 0. 1. 0.]]\n",

    "Flood"
   ],

   "metadata": {

    "id": "n6UOHzhlZEad"
   }
  },
  {

   "cell_type": "code",

   "source": [

    "img = image.load_img('/content/dataset/test_set/Earthquake/1327.jpg' ,
target_size=(64,64))\n",

    "x=image.img_to_array(img)\n",

    "x=np.expand_dims(x,axis=0)\n",
```

```
    "pred = np.argmax(model.predict(x))\n",

    "print(pred,model.predict(x))\n",

    "op=['Cyclone','Earthquake','Flood','Wildfire']\n",

    "print(op[pred])"
  ],
  "metadata": {

    "id": "_BwmPT5zZNPH"

  },

  "execution_count": null,

  "outputs": []
},
{

  "cell_type": "markdown",

  "source": [

    "1/1 [==============================] - 0s 15ms/step\n",

    "1/1 [==============================] - 0s 14ms/step\n",

    "1 [[0. 1. 0. 0.]]\n",

    "Earthquake"
  ],
  "metadata": {

    "id": "aJiDmU3CZbHz"

  }
},
{

  "cell_type": "code",

  "source": [

    "img = image.load_img('/content/dataset/test_set/Wildfire/1050.jpg' , target_size=(64,64))\n",

    "x=image.img_to_array(img)\n",

    "x=np.expand_dims(x,axis=0)\n",

    "pred = np.argmax(model.predict(x))\n",

    "print(pred,model.predict(x))\n",
```

      "op=['Cyclone','Earthquake','Flood','Wildfire']\n",

      "print(op[pred])"

    ],

    "metadata": {

      "id": "n4Vw0lDNZiMh"

    },

    "execution_count": null,

    "outputs": []

  },

  {

    "cell_type": "markdown",

    "source": [

      "1/1 [==============================] - 0s 16ms/step\n",

      "1/1 [==============================] - 0s 16ms/step\n",

      "1 [[0. 1. 0. 0.]]\n",

      "Earthquake"

    ],

    "metadata": {

      "id": "a40uvvi8ZrAH"

    }

  },

  {

    "cell_type": "code",

    "source": [

      "img = image.load_img('/content/test.jpg' , target_size=(64,64))\n",

      "x=image.img_to_array(img)\n",

      "x=np.expand_dims(x,axis=0)\n",

      "pred = np.argmax(model.predict(x))\n",

      "print(pred,model.predict(x))\n",

      "op=['Cyclone','Earthquake','Flood','Wildfire']\n",

      "print(op[pred])"

```json
  ],
  "metadata": {
   "id": "qZh-2NCtZt2F"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "markdown",
  "source": [
   "1/1 [==============================] - 0s 16ms/step\n",
   "1/1 [==============================] - 0s 16ms/step\n",
   "0 [[1. 0. 0. 0.]]\n",
   "Cyclone"
  ],
  "metadata": {
   "id": "f7AXCKdWZ67O"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "img = image.load_img('/content/images.jpeg' , target_size=(64,64))\n",
   "x=image.img_to_array(img)\n",
   "x=np.expand_dims(x,axis=0)\n",
   "pred = np.argmax(model.predict(x))\n",
   "print(pred,model.predict(x))\n",
   "op=['Cyclone','Earthquake','Flood','Wildfire']\n",
   "print(op[pred])"
  ],
  "metadata": {
```

```json
      "id": "joBtoi2AaFnX"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "markdown",
    "source": [
      "1/1 [==============================] - 0s 125ms/step\n",
      "1/1 [==============================] - 0s 14ms/step\n",
      "1 [[0. 1. 0. 0.]]\n",
      "Earthquake"
    ],
    "metadata": {
      "id": "OWCrTO4NaI3v"
    }
  }
 ]
}
```