{

"cells": [

{

"cell\_type": "markdown",

"metadata": {

"id": "5UKKDaCfUNCG"

},

"source": [

"#Importing Keras libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"id": "ipCWLeFzUXGa"

},

"outputs": [],

"source": [

"import keras"

]

},

{

"cell\_type": "code",

"source": [

"pip install twilio"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "AJOv7DJ5qXao",

"outputId": "96cfef0b-1f85-402d-8a64-258930e7a3d6"

},

"execution\_count": 2,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

"Collecting twilio\n",

" Downloading twilio-7.15.2-py2.py3-none-any.whl (1.4 MB)\n",

"\u001b[K |████████████████████████████████| 1.4 MB 14.6 MB/s \n",

"\u001b[?25hCollecting PyJWT<3.0.0,>=2.0.0\n",

" Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)\n",

"Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)\n",

"Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)\n",

"Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)\n",

"Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.24.3)\n",

"Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)\n",

"Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)\n",

"Installing collected packages: PyJWT, twilio\n",

"Successfully installed PyJWT-2.6.0 twilio-7.15.2\n"

]

}

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "RNeZ0vrEUc55"

},

"source": [

"#Importing ImageDataGenerator from Keras"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {

"id": "Mx2JxqSDUnPk"

},

"outputs": [],

"source": [

"from matplotlib import pyplot as plt\n",

"from keras.preprocessing.image import ImageDataGenerator"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "p7bB9Cd1b8kG"

},

"source": [

"#Defining the Parameters"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {

"id": "A1iR\_1R-cKY4"

},

"outputs": [],

"source": [

"train\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,rotation\_range=180,zoom\_range=0.2,horizontal\_flip=True)\n",

"test\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,rotation\_range=180,zoom\_range=0.2,horizontal\_flip=True)\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "pbYsUSI9rrd9"

},

"source": [

"#Applying ImageDataGenerator functionality to train dataset\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "jkgacY2Su-U1",

"outputId": "74b62fae-54ba-49b1-b68d-87b11004ac22"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount(\"/content/drive\", force\_remount=True).\n"

]

}

],

"source": [

"from google.colab import drive\n",

"drive.mount('/content/drive')"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3ou6I0e9sR7o",

"outputId": "95d24a2c-b80b-401c-9a9a-bed02c4da93f"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 436 images belonging to 2 classes.\n"

]

}

],

"source": [

"x\_train=train\_datagen.flow\_from\_directory('/content/drive/MyDrive/IBM PROJECT/dataset/DATA SET/archive/Dataset/Dataset/train\_set',target\_size=(64,64),batch\_size=32,class\_mode='binary')\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "943\_QP5BtsLF"

},

"source": [

"#Applying ImageDataGenerator functionality to test dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "vIgeaeLht0Sn",

"outputId": "58ca80a5-177b-4b6e-9c12-278eebf0e2aa"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 121 images belonging to 2 classes.\n"

]

}

],

"source": [

"x\_test=test\_datagen.flow\_from\_directory('/content/drive/MyDrive/IBM PROJECT/dataset/DATA SET/archive/Dataset/Dataset/test\_set',target\_size=(64,64),batch\_size=32,class\_mode='binary')\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ARyKyJLrVbK5"

},

"source": [

"#Importing Model Building Libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {

"id": "0sMPVWxFVg5P"

},

"outputs": [],

"source": [

"#to define the linear Initialisation import sequential\n",

"from keras.models import Sequential\n",

"#to add layers import Dense\n",

"from keras.layers import Dense\n",

"#to create Convolutional kernel import convolution2D\n",

"from keras.layers import Convolution2D\n",

"#import Maxpooling layer \n",

"from keras.layers import MaxPooling2D\n",

"#import flatten layer\n",

"from keras.layers import Flatten\n",

"import warnings\n",

"warnings.filterwarnings('ignore')"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "\_LVXcOT9eLM3"

},

"source": [

"#Initializing the model"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {

"id": "TquY2jOfeRQF"

},

"outputs": [],

"source": [

"model = Sequential()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "CqOjIFO2eaKf"

},

"source": [

"#Adding CNN Layers"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {

"id": "myoMJiBxeh3q"

},

"outputs": [],

"source": [

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

"#add maxpooling layers\n",

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

"#add faltten layer\n",

"model.add(Flatten())"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "x67h-tMPfpHG"

},

"source": [

"#Add Dense layers"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {

"id": "JodC\_f4Vfsn3"

},

"outputs": [],

"source": [

"#add hidden layers\n",

"model.add(Dense(150,activation='relu'))\n",

"#add output layer\n",

"model.add(Dense(1,activation='sigmoid'))"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ELm-4mBdgfOv"

},

"source": [

"#configuring the learning process"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {

"id": "evmGQQWXgo1t"

},

"outputs": [],

"source": [

"model.compile(loss='binary\_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "80s2C38ThYaj"

},

"source": [

"#Training the model"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3Cy0ktrJhcfT",

"outputId": "06fe42b8-5b70-4a6e-e1fd-f8724c61fda0"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/10\n",

"14/14 [==============================] - 137s 10s/step - loss: 0.9627 - accuracy: 0.7294 - val\_loss: 0.3491 - val\_accuracy: 0.8595\n",

"Epoch 2/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.3139 - accuracy: 0.8532 - val\_loss: 0.1041 - val\_accuracy: 0.9752\n",

"Epoch 3/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.2354 - accuracy: 0.9037 - val\_loss: 0.1252 - val\_accuracy: 0.9421\n",

"Epoch 4/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.2281 - accuracy: 0.9106 - val\_loss: 0.0831 - val\_accuracy: 0.9752\n",

"Epoch 5/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1916 - accuracy: 0.9289 - val\_loss: 0.0753 - val\_accuracy: 0.9752\n",

"Epoch 6/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.1879 - accuracy: 0.9128 - val\_loss: 0.0738 - val\_accuracy: 0.9669\n",

"Epoch 7/10\n",

"14/14 [==============================] - 20s 2s/step - loss: 0.1720 - accuracy: 0.9312 - val\_loss: 0.0745 - val\_accuracy: 0.9669\n",

"Epoch 8/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1707 - accuracy: 0.9266 - val\_loss: 0.1375 - val\_accuracy: 0.9339\n",

"Epoch 9/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.1539 - accuracy: 0.9358 - val\_loss: 0.0504 - val\_accuracy: 0.9917\n",

"Epoch 10/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1467 - accuracy: 0.9450 - val\_loss: 0.0694 - val\_accuracy: 0.9752\n"

]

},

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7f5149cce2d0>"

]

},

"metadata": {},

"execution\_count": 13

}

],

"source": [

"model.fit\_generator(x\_train,steps\_per\_epoch=14,epochs=10,validation\_data=x\_test,validation\_steps=4)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "CSmW5ypkizjF"

},

"source": [

"#Save the model"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {

"id": "Q2DVxGYgi36I"

},

"outputs": [],

"source": [

"model.save(\"forest.h5\")"

]

}

],

"metadata": {

"colab": {

"collapsed\_sections": [],

"provenance": []

},

"kernelspec": {

"display\_name": "Python 3",

"name": "python3"

},

"language\_info": {

"name": "python"

}

},

"nbformat": 4,

"nbformat\_minor": 0

}