```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": [],
   "collapsed_sections": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "cells": [
  {
   "cell_type": "markdown",
   "source": [
    "#Importing Keras libraries"
   ],
   "metadata": {
    "id": "5UKKDaCfUNCG"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "import keras"
   ],
   "metadata": {
    "id": "ipCWLeFzUXGa"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "#Importing ImageDataGenerator from Keras"
   ],
   "metadata": {
```

```
      "id": "RNeZ0vrEUc55"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "from matplotlib import pyplot as plt\n",
      "from keras.preprocessing.image import ImageDataGenerator"
     ],
     "metadata": {
      "id": "Mx2JxqSDUnPk"
     },
     "execution_count": null,
     "outputs": []
    },
    {
     "cell_type": "markdown",
     "source": [
      "#Defining the Parameters"
     ],
     "metadata": {
      "id": "p7bB9Cd1b8kG"
     }
    },
    {
     "cell_type": "code",
     "source": [

"train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoo
m_range=0.2,horizontal_flip=True)\n",
      "test_datagen=ImageDataGenerator(rescale=1./255)\n"
     ],
     "metadata": {
      "id": "A1iR_1R-cKY4"
     },
     "execution_count": null,
     "outputs": []
    },
    {
     "cell_type": "markdown",
     "source": [],
     "metadata": {
      "id": "6mmovZihrOW6"
```

```
    }
   },
   {
    "cell_type": "markdown",
    "source": [
     "#Applying ImageDataGenerator functionality to train dataset\n",
     "\n"
    ],
    "metadata": {
     "id": "pbYsUSl9rrd9"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "from google.colab import drive\n",
     "drive.mount('/content/drive')"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "jkgacY2Su-U1",
     "outputId": "abdf4da7-ed6a-4495-c482-0ea4ff2fa122"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
       "Mounted at /content/drive\n"
      ]
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM
PROJECT/dataset/DATA
SET/archive/Dataset/Dataset/train_set',target_size=(128,128),batch_size=32,class_mode='bina
ry')\n"
```

```
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "3ou6I0e9sR7o",
     "outputId": "620d27ff-bc9a-442b-9218-616594b7f228"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
       "Found 436 images belonging to 2 classes.\n"
      ]
     }
    ]
   },
   {
    "cell_type": "markdown",
    "source": [
     "#Applying ImageDataGenerator functionality to test dataset"
    ],
    "metadata": {
     "id": "943_QP5BtsLF"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IBM
PROJECT/dataset/DATA
SET/archive/Dataset/Dataset/test_set',target_size=(128,128),batch_size=32,class_mode='binar
y')\n"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "vIgeaeLht0Sn",
     "outputId": "a49ef9f5-07ae-4583-9af1-d5bb5502a627"
    },
```

```
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "Found 121 images belonging to 2 classes.\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "#Importing Model Building Libraries"
  ],
  "metadata": {
    "id": "ARyKyJLrVbK5"
  }
},
{
  "cell_type": "code",
  "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')"
  ],
  "metadata": {
    "id": "0sMPVWxFVg5P"
  },
  "execution_count": 11,
  "outputs": []
},
{
```

```
  "cell_type": "markdown",
  "source": [
    "#Initializing the model"
  ],
  "metadata": {
    "id": "_LVXcOT9eLM3"
  }
},
{
  "cell_type": "code",
  "source": [
    "model=Sequential()"
  ],
  "metadata": {
    "id": "TquY2jOfeRQF"
  },
  "execution_count": 12,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "#Adding CNN Layers"
  ],
  "metadata": {
    "id": "CqOjIFO2eaKf"
  }
},
{
  "cell_type": "code",
  "source": [
    "model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))\n",
    "#add maxpooling layers\n",
    "model.add(MaxPooling2D(pool_size=(2,2)))\n",
    "#add faltten layer\n",
    "model.add(Flatten())"
  ],
  "metadata": {
    "id": "myoMJiBxeh3q"
  },
  "execution_count": 13,
  "outputs": []
},
```

```
{
  "cell_type": "markdown",
  "source": [
    "#Add Dense layers"
  ],
  "metadata": {
    "id": "x67h-tMPfpHG"
  }
},
{
  "cell_type": "code",
  "source": [
    "#add hidden layers\n",
    "model.add(Dense(150,activation='relu'))\n",
    "#add output layer\n",
    "model.add(Dense(1,activation='sigmoid'))"
  ],
  "metadata": {
    "id": "JodC_f4Vfsn3"
  },
  "execution_count": 19,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "#configuring the learning process"
  ],
  "metadata": {
    "id": "ELm-4mBdgfOv"
  }
},
{
  "cell_type": "code",
  "source": [
    "model.compile(loss='binary_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"
  ],
  "metadata": {
    "id": "evmGQQWXgo1t"
  },
  "execution_count": 20,
  "outputs": []
},
```

```
   {
    "cell_type": "markdown",
    "source": [
     "#Training the model"
    ],
    "metadata": {
     "id": "80s2C38ThYaj"
    }
   },
   {
    "cell_type": "code",
    "source": [

"model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_
steps=4)"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 0
     },
     "id": "3Cy0ktrJhcfT",
     "outputId": "771e8548-7ee5-447c-fc6a-dafc669c30fd"
    },
    "execution_count": 23,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
       "Epoch 1/10\n",
       "14/14 [==============================] - 322s 19s/step - loss: 1.5998 -
accuracy: 0.7133 - val_loss: 0.1776 - val_accuracy: 0.9421\n",
       "Epoch 2/10\n",
       "14/14 [==============================] - 26s 2s/step - loss: 0.3427 - accuracy:
0.8624 - val_loss: 0.2325 - val_accuracy: 0.8926\n",
       "Epoch 3/10\n",
       "14/14 [==============================] - 32s 2s/step - loss: 0.2979 - accuracy:
0.8853 - val_loss: 0.0970 - val_accuracy: 0.9752\n",
       "Epoch 4/10\n",
       "14/14 [==============================] - 29s 2s/step - loss: 0.2585 - accuracy:
0.8922 - val_loss: 0.0621 - val_accuracy: 0.9752\n",
       "Epoch 5/10\n",
```

```
          "14/14 [==============================] - 29s 2s/step - loss: 0.1926 - accuracy:
0.9243 - val_loss: 0.0688 - val_accuracy: 0.9835\n",
          "Epoch 6/10\n",
          "14/14 [==============================] - 30s 2s/step - loss: 0.1971 - accuracy:
0.9266 - val_loss: 0.0815 - val_accuracy: 0.9752\n",
          "Epoch 7/10\n",
          "14/14 [==============================] - 32s 2s/step - loss: 0.1781 - accuracy:
0.9289 - val_loss: 0.0571 - val_accuracy: 0.9835\n",
          "Epoch 8/10\n",
          "14/14 [==============================] - 30s 2s/step - loss: 0.1796 - accuracy:
0.9243 - val_loss: 0.0561 - val_accuracy: 0.9835\n",
          "Epoch 9/10\n",
          "14/14 [==============================] - 31s 2s/step - loss: 0.2306 - accuracy:
0.8968 - val_loss: 0.0704 - val_accuracy: 0.9835\n",
          "Epoch 10/10\n",
          "14/14 [==============================] - 27s 2s/step - loss: 0.2593 - accuracy:
0.8899 - val_loss: 0.1232 - val_accuracy: 0.9504\n"
        ]
       },
       {
        "output_type": "execute_result",
        "data": {
         "text/plain": [
          "<keras.callbacks.History at 0x7fd537101390>"
         ]
        },
        "metadata": {},
        "execution_count": 23
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "#Save the model"
      ],
      "metadata": {
       "id": "CSmW5ypkizjF"
      }
     },
     {
      "cell_type": "code",
      "source": [
```

```
      "model.save(\"forest.h5\")"
     ],
     "metadata": {
      "id": "Q2DVxGYgi36I"
     },
     "execution_count": 24,
     "outputs": []
    },
    {
     "cell_type": "markdown",
     "source": [
      "#Predictions"
     ],
     "metadata": {
      "id": "G8KMppTojF__"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "#import load model from keras.model\n",
      "from keras.models import load_model\n",
      "#import image from keras\n",
      "from tensorflow.keras.preprocessing import image\n",
      "import numpy as np\n",
      "#import cv2\n",
      "import cv2\n",
      "#load the saved model\n",
      "model=load_model('forest.h5')\n",
      "img=image.load_img('/content/drive/MyDrive/IBM PROJECT/dataset/DATA
SET/archive/Dataset/Dataset/test_set/forest/0.64133000_1519374442_forest_deep.jpg')\n",
      "x=image.img_to_array(img)\n",
      "res=cv2.resize(x,dsize=(128,128),interpolation=cv2.INTER_CUBIC)\n",
      "#expand the image shape\n",
      "x=np.expand_dims(res,axis=0)\n",
      "\n",
      "\n"
     ],
     "metadata": {
      "id": "aZxNjrmJjImS"
     },
     "execution_count": 25,
     "outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "pred=model.predict(x)"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 0
  },
  "id": "A3CYH-vNmkX0",
  "outputId": "533cfc63-e8f6-47df-a0e2-fcb694bb70a6"
 },
 "execution_count": 26,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "1/1 [==============================] - 0s 118ms/step\n"
   ]
  }
 ]
},
{
 "cell_type": "code",
 "source": [
  "pred"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 0
  },
  "id": "eQkolLKImlSq",
  "outputId": "545fed67-663e-40d6-fca7-fb1ed840768e"
 },
 "execution_count": 27,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
```

```
          "text/plain": [
            "array([[0.]], dtype=float32)"
          ]
        },
        "metadata": {},
        "execution_count": 27
      }
    ]
  }
 ]
}
```