PNT2022TMID52343


```
{
 "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": [
    "Team ID:PNT2022TMlD09657"
   ],
   "metadata": {
    "id": "A3S5qEZ82XHn"
```

```
  }
 },
 {
  "cell_type": "markdown",
  "source": [
   "**Importing The ImageDataGenerator Library**"
  ],
  "metadata": {
   "id": "Yq7MBIMzxa46"
  }
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "TwbVY9ldxT5E"
  },
  "outputs": [],
  "source": [
   "import keras \n",
   "from keras.preprocessing.image import ImageDataGenerator"
  ]
 },
 {
  "cell_type": "markdown",
  "source": [
   "**Define the parameters/arguments for ImageDataGenerator class**"
  ],
  "metadata": {
```

```
      "id": "449Gl96Uxb3j"
     }
    },
    {
     "cell_type": "code",
     "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)"
     ],
     "metadata": {
      "id": "7E0B6ZFCxcAC"
     },
     "execution_count": null,
     "outputs": []
    },
    {
     "cell_type": "markdown",
     "source": [
      "**Applying ImageDataGenerator functionality to trainset**"
     ],
     "metadata": {
      "id": "Ceh_nXLHxcId"
     }
    },
    {
     "cell_type": "code",
     "source": [
```

"x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/train_set',target_size=(128,128),batch_size=32, class_mode='binary')"

],

"metadata": {

"colab": {

"base_uri": "https://localhost:8080/"

},

"id": "C5FSZ_KnxcQW",

"outputId": "c28b17a4-def3-4de0-ca2d-711ede6b63a8"

},

"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 testset**"

],

"metadata": {

"id": "dRHL_n10yDdC"

```
   }
  },
  {
   "cell_type": "code",
   "source": [

"x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(128,128),batch_size=32, class_mode='binary')"

   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "D7_3jdIqxzhv",
    "outputId": "fe7c7493-17db-41aa-d1b0-0e918b80fb34"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Found 121 images belonging to 2 classes.\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
```

```
  "source": [
    "**Import model building libraries**"
  ],
  "metadata": {
    "id": "6fbxEKhryYhE"
  }
},
{
  "cell_type": "code",
  "source": [
    "#To define 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 Convolution 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": "6RuPf8_ZyYoc"
  },
  "execution_count": null,
  "outputs": []
},
```

```
{
 "cell_type": "markdown",
 "source": [
  "**Initializing the model**"
 ],
 "metadata": {
  "id": "-v8KAP49ysom"
 }
},
{
 "cell_type": "code",
 "source": [
  "model=Sequential()"
 ],
 "metadata": {
  "id": "DvWAyoLLysza"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "markdown",
 "source": [
  "**Add CNN Layer**"
 ],
 "metadata": {
  "id": "luqIqx4zyykT"
 }
},
```

```
{
 "cell_type": "code",
 "source": [
  "model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu')) \n",
  "#add maxpooling layer\n",
  "model.add(MaxPooling2D(pool_size=(2,2)))\n",
  "#add flatten layer\n",
  "model.add(Flatten())"
 ],
 "metadata": {
  "id": "XsWT_D1iyyVr"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "markdown",
 "source": [
  "**Add Hidden Layer**"
 ],
 "metadata": {
  "id": "wx8nTNr9zTgl"
 }
},
{
 "cell_type": "code",
 "source": [
  " #add hidden layer\n",
  "model.add(Dense(150,activation='relu')) \n",
```

```
    "#add output layer \n",
    "model.add(Dense(1,activation='sigmoid'))"
   ],
   "metadata": {
    "id": "o3PiIaCRzRbx"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Configure the learning process**"
   ],
   "metadata": {
    "id": "Q0hbp_tQzY5o"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "model.compile(loss='binary_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"
   ],
   "metadata": {
    "id": "7QHmgxsyzZIn"
   },
   "execution_count": null,
   "outputs": []
  },
```

```
 {
  "cell_type": "markdown",
  "source": [
   "**Train the model**"
  ],
  "metadata": {
   "id": "0OPIh_g-zhpc"
  }
 },
 {
  "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/"
   },
   "id": "dbRjnLDAzkxm",
   "outputId": "32b8e981-9d3b-466a-95d3-4d8b9e924365"
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
```

```
"Epoch 1/10\n",

"14/14 [==============================] - 205s 15s/step - loss: 2.7344 - accuracy: 0.7454 - val_loss: 0.2016 - val_accuracy: 0.9256\n",

"Epoch 2/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.3267 - accuracy: 0.8945 - val_loss: 0.2290 - val_accuracy: 0.9339\n",

"Epoch 3/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.2991 - accuracy: 0.8922 - val_loss: 0.0524 - val_accuracy: 0.9835\n",

"Epoch 4/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.2418 - accuracy: 0.9174 - val_loss: 0.1570 - val_accuracy: 0.9421\n",

"Epoch 5/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1984 - accuracy: 0.9083 - val_loss: 0.0767 - val_accuracy: 0.9752\n",

"Epoch 6/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1643 - accuracy: 0.9335 - val_loss: 0.0749 - val_accuracy: 0.9752\n",

"Epoch 7/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1538 - accuracy: 0.9312 - val_loss: 0.1264 - val_accuracy: 0.9421\n",

"Epoch 8/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1732 - accuracy: 0.9266 - val_loss: 0.0652 - val_accuracy: 0.9835\n",

"Epoch 9/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1514 - accuracy: 0.9358 - val_loss: 0.0567 - val_accuracy: 0.9835\n",

"Epoch 10/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1445 - accuracy: 0.9404 - val_loss: 0.0448 - val_accuracy: 0.9917\n"

 ]

},
```

```
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "<keras.callbacks.History at 0x7f51fdf33610>"
     ]
    },
    "metadata": {},
    "execution_count": 10
   }
  ]
 },
 {
  "cell_type": "markdown",
  "source": [
   "**Save The Model**"
  ],
  "metadata": {
   "id": "ExX269AR2OJO"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "model.save(\"forest1.h5\")"
  ],
  "metadata": {
   "id": "r_HNZErV2NPz"
  },
```

```
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "markdown",
 "source": [
  "**Predictions**"
 ],
 "metadata": {
  "id": "_KL58pMT2TR2"
 }
},
{
 "cell_type": "code",
 "source": [
  " #import load_model from keras.model\n",
  "from keras.models import load_model\n",
  "#import image class from keras\n",
  "from tensorflow.keras.preprocessing import image #import numpy\n",
  "import numpy as np\n",
  "#import cv2\n",
  "import cv2"
 ],
 "metadata": {
  "id": "ojWbOAng4RRA"
 },
 "execution_count": null,
 "outputs": []
},
```

```
{
 "cell_type": "code",
 "source": [
  "#load the saved model\n",
  "model = load_model(\"forest1.h5\")"
 ],
 "metadata": {
  "id": "lTQgwl0x4coy"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [

"img=image.load_img(r'/content/drive/MyDrive/Dataset/test_set/forest/0.48007200_1530881924_final_forest.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)"
 ],
 "metadata": {
  "id": "iD7BChtG2Tad"
 },
 "execution_count": null,
 "outputs": []
},
```

```
{
 "cell_type": "code",
 "source": [
  "pred= model.predict(x)"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "YfvDp7BZ4n41",
  "outputId": "fb723c8a-7bd3-4d05-8980-91a3911c4830"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "1/1 [==============================] - 0s 94ms/step\n"
   ]
  }
 ]
},
{
 "cell_type": "code",
 "source": [
  "pred"
 ],
 "metadata": {
```

```
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "rcnCY5ao6hg-",
      "outputId": "d56708af-7c8b-4b7c-c624-4442dc2a1a26"
    },
    "execution_count": null,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "array([[0.]], dtype=float32)"
          ]
        },
        "metadata": {},
        "execution_count": 16
      }
    ]
  }
 ]
}
```