```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "## Assignment 3"
      ],
      "metadata": {
        "id": "ZhTkO_xJNZgz"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "Load the Dataset"
      ],
      "metadata": {
```

```
    "id": "lXNu-ZihNmxJ"
   }
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "t1uZKj5bEwwQ",
    "outputId": "a59ef5bc-8a7b-4fef-c4b2-12cdb1c73481"
   },
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "\u001b[0m\u001b[01;34msample_data\u001b[0m/\n"
     ]
    }
   ],
   "source": [
    "ls\n"
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "cd /content/drive/MyDrive/Colab Notebooks/CNN1"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "LHanloDoGrzY",
    "outputId": "d4b2fb1b-23bd-4850-ee47-ddca3e51e44a"
   },
   "execution_count": 2,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[Errno 2] No such file or directory: '/content/drive/MyDrive/Colab Notebooks/CNN1'\n",
      "/content\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "ls"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "lyDGkn-kG6oA",
    "outputId": "7203bca4-5df6-400e-9c77-6c2633051a2e"
   },
   "execution_count": 3,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "\u001b[0m\u001b[01;34msample_data\u001b[0m/\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "pwd"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 36
    },
    "id": "YOyE9_5BG8S7",
    "outputId": "9c96e0fc-21c1-411c-8c20-a18b6fd81b59"
   },
   "execution_count": 4,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "'/content'"
      ],
      "application/vnd.google.colaboratory.intrinsic+json": {
       "type": "string"
      }
     },
     "metadata": {},
     "execution_count": 4
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "!unzip Flowers-Dataset.zip"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "l4P1mFeUHFUN",
    "outputId": "ce354296-539c-4c9c-d8a8-6834a89ade91"
   },
   "execution_count": 5,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "unzip:  cannot find or open Flowers-Dataset.zip, Flowers-Dataset.zip.zip or Flowers-Dataset.zip.ZIP.\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "Image Augmentation"
   ],
   "metadata": {
    "id": "pF_AjEokNrd6"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "from tensorflow.keras.preprocessing.image import ImageDataGenerator"
   ],
   "metadata": {
    "id": "M-iicHQ7HbUM"
   },
   "execution_count": 6,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "train_data = ImageDataGenerator(rescale= 1./255,horizontal_flip = True,vertical_flip = True,zoom_range = 0.2)"
   ],
   "metadata": {
    "id": "LrpG4TOMI5V5"
   },
   "execution_count": 7,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "test_data = ImageDataGenerator(rescale= 1./255)"
   ],
   "metadata": {
    "id": "UA_tym2xI-e0"
   },
   "execution_count": 8,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "x_train = train_data.flow_from_directory(r\"/content/drive/MyDrive/Colab Notebooks/CNN1/flowers\",target_size = (64,64),\n",
    "                                         class_mode = \"categorical\",batch_size = 40)"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "YlvIAQUVJEkc",
    "outputId": "367a06b3-6605-4ceb-eb0c-38b626dc8f51"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Found 4317 images belonging to 5 classes.\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "x_test = test_data.flow_from_directory(r\"/content/drive/MyDrive/Colab
Notebooks/CNN1/flowers\",target_size = (64,64),\n",
    "                                       class_mode = \"categorical\",batch_size = 40)"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "0-NIu-P_M5nH",
    "outputId": "ce335c2c-363a-436c-d61c-7968eecaf008"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Found 4317 images belonging to 5 classes.\n"
     ]
  }
 ]
},
{
 "cell_type": "markdown",
 "source": [
  "Creating CNN Model"
 ],
 "metadata": {
  "id": "2nmCuO7LN3X6"
 }
},
{
 "cell_type": "code",
 "source": [
  "from tensorflow.keras.models import Sequential\n",
  "from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense"
 ],
 "metadata": {
  "id": "2q6Z1dVINFvU"
 },
 "execution_count": 9,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "model = Sequential()"
 ],
 "metadata": {
  "id": "3m9wrrSGN-ql"
```

```
   },
   "execution_count": 10,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "Adding Layers"
   ],
   "metadata": {
    "id": "0-ZbChbCODbR"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "model.add(Convolution2D(32,(3,3),activation = \"relu\",input_shape = (64,64,3)))"
   ],
   "metadata": {
    "id": "HyiAfTMnOCFl"
   },
   "execution_count": 11,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "model.add(MaxPooling2D(pool_size = (2,2)))"
   ],
   "metadata": {
    "id": "aNK9ADG7Psmh"
  },
  "execution_count": 12,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "model.add(Flatten())"
  ],
  "metadata": {
   "id": "-OZXxyf7P9zV"
  },
  "execution_count": 13,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "model.add(Dense(300, activation='relu'))"
  ],
  "metadata": {
   "id": "fn8xaafaQCmx"
  },
  "execution_count": 14,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "model.add(Dense(300, activation='relu'))"
  ],
   "metadata": {
    "id": "Tv3NMFETQYwo"
   },
   "execution_count": 15,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "model.add(Dense(5, activation=\"softmax\"))"
   ],
   "metadata": {
    "id": "1aqyKtr0Qak4"
   },
   "execution_count": 16,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "Compile the Model"
   ],
   "metadata": {
    "id": "-E8pQVHwdjBS"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "model.compile(loss=\"categorical_crossentropy\", metrics=[\"accuracy\"], optimizer='adam')"
   ],
   "metadata": {
    "id": "4YtIphm3Q95a"
   },
   "execution_count": 17,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "len(x_train)"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "qImUDvL1RV0y",
    "outputId": "53a7cad5-5288-4e5b-8e80-75a2fd5c0f5e"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "108"
      ]
     },
     "metadata": {},
     "execution_count": 24
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "Fit the Model"
   ],
   "metadata": {
    "id": "eytEyopbdm97"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "model.fit(x_train, epochs = 5, validation_data=x_test, steps_per_epoch=len(x_train), validation_steps=len(x_test))"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "fmRZdzd4RbP7",
    "outputId": "ddc35e62-a02e-4fc5-cfc4-82ee857705d1"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Epoch 1/5\n",
      "108/108 [==============================] - 62s 578ms/step - loss: 1.0442 - accuracy: 0.5789 - val_loss: 1.0239 - val_accuracy: 0.5967\n",
      "Epoch 2/5\n",
      "108/108 [==============================] - 60s 560ms/step - loss: 0.9752 - accuracy: 0.6101 - val_loss: 0.9477 - val_accuracy: 0.6319\n",
      "Epoch 3/5\n",
      "108/108 [==============================] - 61s 569ms/step - loss: 0.9167 - accuracy: 0.6456 - val_loss: 0.8871 - val_accuracy: 0.6613\n",
      "Epoch 4/5\n",
      "108/108 [==============================] - 61s 565ms/step - loss: 0.8663 - accuracy: 0.6704 - val_loss: 0.8931 - val_accuracy: 0.6595\n",
      "Epoch 5/5\n",
      "108/108 [==============================] - 62s 577ms/step - loss: 0.8402 - accuracy: 0.6706 - val_loss: 0.9205 - val_accuracy: 0.6335\n"
     ]
    },
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "<keras.callbacks.History at 0x7f0ecc3d9890>"
      ]
     },
     "metadata": {},
     "execution_count": 25
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "Save the Model"
   ],
   "metadata": {
    "id": "Qmv_oD0LdrCo"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "model.save(\"flower.h5\")"
   ],
   "metadata": {
    "id": "J1QhEn5iSMXx"
   },
   "execution_count": 18,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "Test the Model"
   ],
   "metadata": {
    "id": "MagSVFKjdt_W"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "from tensorflow.keras.models import load_model\n",
    "from tensorflow.keras.preprocessing import image\n",
    "import numpy as np"
   ],
   "metadata": {
    "id": "8c1ET9KqUO8i"
   },
   "execution_count": 19,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    " model = load_model(\"/content/drive/MyDrive/Colab Notebooks/CNN1/flower.h5\")"
   ],
   "metadata": {
    "id": "et9uzSWlUnVP"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "img = image.load_img(\"/content/drive/MyDrive/Colab Notebooks/CNN1/flowers/rose/10090824183_d02c613f10_m.jpg\",target_size=(64,64))"
   ],
   "metadata": {
    "id": "EkgNz_RWVuDz"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
     "img"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 81
     },
     "id": "mmucha8AWT5z",
     "outputId": "e78dbade-56f4-43fb-e3a5-0b63795a8d57"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "<PIL.Image.Image image mode=RGB size=64x64 at 0x7F0ECC0DFC90>"
       ],
       "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAEAAAABACAIAAAAlC+aJAAAKMWlDQ1BJQ0MgUHJvZmlsZQAAeJy
dlndUU9kWh8+9N71QkhCKlNBraFICSA29SJEuKjEJEErAkAAiNkRUcERRkaYIMijggKNDkbEiioUBUbHrBB
lE1HFwFBuWSWStGd+8ee/Nm98f935rn73P3Wfvfda6AJD8gwXCTFgJgAyhWBTh58WIjYtnYAcBDPAAA
2wA4HCzs0IW+EYCmQJ82IxsmRP4F726DiD5+yrTP4zBAP+flLlZIjEAUJiM5/L42VwZF8k4PVecJbdPyZi2N
E3OMErOIlmCMlaTc/IsW3z2mWUPOfMyhDwZy3PO4mXw5Nwn4405Er6MkWAZF+cI+LkyviZjg3RJhk
DGb+SxGXxONgAoktwu5nNTZGwtY5IoMoIt43kA4EjJX/DSL1jMzxPLD8XOzFouEiSniBkmXFOGjZMTi+H
Pz03ni8XMMA43jSPiMdiZGVkc4XIAZs/8WRR5bRmyIjvYODk4MG0tbb4o1H9d/JuS93aWXoR/7hlEH/jD
9ld+mQ0AsKZltdn6h21pFQBd6wFQu/2HzWAvAIqyvnUOfXEeunxeUsTiLGcrq9zcXEsBn2spL+jv+p8Of0
NffM9Svt3v5WF485M4knQxQ143bmZ6pkTEyM7icPkM5p+H+B8H/nUeFhH8JL6IL5RFRMumTCBMlrV
byBOIBZlChkD4n5r4D8P+pNm5lona+BHQllgCpSEaQH4eACgqESAJe2Qr0O99C8ZHA/nNi9GZmJ37z4L+
fVe4TP7IFiR/jmNHRDK4ElHO7Jr8WgI0IABFQAPqQBvoAxPABLbAEbgAD+ADAkEoiARxYDHgghSQAUQg
FxSAtaAYlIKtYCeoBnWgETSDNnAYdlFj4DQ4By6By2AE3AFSMA6egCnwCsxAEISFyBAVUod0IEPIHLKFW
JAb5AMFQxFQHJQIJUNCSAIVQOugUqgcqobqoWboW+godBq6AA1Dt6BRaBL6FXoHIzAJpsFasBFsBbN
gTzgIjoQXwcnwMjgfLoK3wJVwA3wQ7oRPw5fgEVgKP4GnEYAQETqiizARFsJGQpF4JAkRIauQEqQCaUD
akB6kH7mKSJGnyFsUBkVFMVBMlAvKHxWF4qKWoVahNqOqUQdQnag+1FXUKGoK9RFNRmuizdHO6
AB0LDoZnYsuRlegm9Ad6LPoEfQ4+hUGg6FjjDDGGOGH9MHCYVswKzGbMb0445hRnGjGGmsVisOtYc64o
NxXKwYmwxtgp7EHsSewU7jn2DI+J0cLY4X1w8TogrxFXgWnAncFdwE7gZvBLeEO+MD8Xz8Mvvx
9+CD+OnyEoE4wJroRIQiphLaGS0EY4S7hLeEEkEvWITsRwooC4hlhJPEQ8TxwlviVRSGYkNimBJCFtIe0nn
SLdIr0gk8lGZA9yPFlM3kJuJp8h3ye/UaAqWCoEKPAVViRUKHQqXFF4RsFTDCleCxSXFKFsXNFT0VFysmK9YoXhEcUjxR
```

JeyUiJrcRRWqVUo3RU6YbStDJV2UY5VDlDebNyi/IF5UcULMWI4kPhUYoo+yhnKGNUhKpPZVO51HXUR
upZ6jgNQzOmBdBSaaW0b2iDtCkVioqdSrRKnkqNynEVKR2hG9ED6On0Mvph+nX6O1UtVU9Vvuom1T
bVK6qv1eaoeajx1UrU2tVG1N6pM9R91NPUt6l3qd/TQGmYaYRr5Grs0Tir8XQObY7LHO6ckjmH59zWh
DXNNCM0V2ju0xzQnNbS1vLTytKq0jqj9VSbru2hnaq9Q/uE9qQOVcdNR6CzQ+ekzmOGCsOTkc6oZPQx
pnQ1df11Jbr1uoO6M3rGelF6hXrtevf0Cfos/ST9Hfq9+lMGOgYhBgUGrQa3DfGGLMMUw12G/YavjYyN
Yow2GHUZPTJWMw4wzjduNb5rQjZxN1lm0mByzRRjyjJNM91tetkMNrM3SzGrMRsyh80dzAXmu82HL
dAWThZCiwaLG0wS05OZw2xljlrSLYMtCy27LJ9ZGVjFW22z6rf6aG1vnW7daH3HhmITaFNo02Pzq62ZLd
e2xvbaXPJc37mr53bPfW5nbse322N3055qH2K/wb7X/oODo4PIoc1h0tHAMdGx1vEGi8YKY21mnXdCO
3k5rXY65vTW2cFZ7HzY+RcXpkuaS4vLo3nG8/jzGueNueq5clzrXaVuDLdEt71uUnddd457g/sDD30Pnke
Tx4SnqWeq50HPZ17WXiKvDq/XbGf2SvYpb8Tbz7vEe9CH4hPlU+1z31fPN9m31XfKz95vhd8pf7R/kP82
/xsBWgHcgOaAqUDHwJWBfUGkoAVB1UEPgs2CRcE9IXBIYMj2kLvzDecL53eFgtCA0O2h98KMw5aFfR+
OCQ8Lrwl/GGETURDRv4C6YMmClgWvIr0iyyLvRJlESaJ6oxWjE6Kbo1/HeMeUx0hjrWJXxl6K04gTxHXHY
+Oj45vipxf6LNy5cDzBPqE44foi40V5iy4s1licvvj4EsUlnCVHEtGJMYktie85oZwGzvTSgKW1S6e4bO4u7h
OeB28Hb5Lvyi/nTyS5JpUnPUp2Td6ePJninlKR8lTAFlQLnqf6p9alvk4LTduf9ik9Jr09A5eRmHFUSBGmCfs
ytTPzMoezzLOKs6TLnJftXDYlChI1ZUPZi7K7xTTZz9SAxESyXjKa45ZTk/MmNzr3SJ5ynjBvYLnZ8k3LJ/J987
9egVrBXdFboFuwtmB0pefK+lXQqqWrelfrry5aPb7Gb82BtYS1aWt/KLQuLC98uS5mXU+RVtGaorH1fut
bixWKRcU3NrhsqNuI2ijYOLhp7qaqTR9LeCUXS61LK0rfb+ZuvviVzVeVX33akrRlsMyhbM9WzFbh1uvb3
LcdKFcuzy8f2x6yvXMHY0fJjpc7l+y8UGFXUbeLsEuyS1oZXNldZVC1tep9dUr1SI1XTXutZu2m2te7ebuv7
PHY01anVVda926vYO/Ner/6zgajhop9mH05+x42Rjf2f836urlJo6m06cN+4X7pgYgDfc2Ozc0tmi1lrXCrp
HXyYMLBy994f9Pdxmyrb6e3lx4ChySHHn+b+O31w0GHe4+wjrR9Z/hdbQe1o6QT6lzeOdWV0iXtjuseP
hp4tLfHpafje8vv9x/TPVZzXOV42QnCiaITn07mn5w+lXXq6enk02O9S3rvnIk9c60vvG/wbNDZ8+d8z53p
9+w/ed71/LELzheOXmRd7LrkcKlzwH6g4wf7HzoGHQY7hxyHui87Xe4Znjd84or7ldNXva+euxZw7dLI/JH
h61HXb95IuCG9ybv56Fb6ree3c27P3FlzF3235J7SvYr7mvcbfjT9sV3qID0+6j068GDBgztj3LEnP2X/9H68
6CH5YcWEzkTzI9tHxyZ9Jy8/Xvh4/EnWk5mnxT8r/1z7zOTZd794/DIwFTs1/lz0/NOvm1+ov9j/0u5l73TY
9P1XGa9mXpe8UX9z4C3rbf+7mHcTM7nvse8rP5h+6PkY9PHup4xPn34D94Tz+6TMXDkAACLzSURBVHi
cTbppuK5pVR641nqmd/jGPZ29zz6nqDqnThVVDFVFDASMSAkaxBijQsfI4NCXVwK2GYxJOoNgoibRS41
Nd6uBTtsMJhJkuEgQNEEoAUGpAQqsgZrOuIezh29/3/eOz7BWfuzK1fn3/nre51nTfa91LxxtuASSUgIglGi
tVhE9iO4RKc0WiQQBRQBEQCnFnECgGNiYOHBkBkTSmgSlHNt67lNiYNy+cJYc9m2jlAIAYRARIgrRI5CIM
PPF229lZgJcLE8W88aHxnd9URQiKQnMTxqtsrPnV3b3j4au2L96DAiQABG2bt+wGTGyQsTh2oCRAZkToI
IYIzQxL3We2X7ZiOiTPjprFGJTeUBQCgBBIokIaUzCJjMbW+tdWk7Wp8985SoyDKdZ1AkVaa1FBAC01mf
Pnu26DgCYWWullGraRVmWi1k9m5147wEAFTGzQkKFHCG0LCDD8Yixt86REgrm+PD4BS88p4iZoYle20
IDYuIgjCKolCuGBpn2d24OR9ZanuRZNe+QNFkEAEaQgEgsAPmodAPHkDpps9LWi04p0IWpU2fQZc5pQ
wAiCAB8PJspDZosInjfnRzP27YFOICESRAUnL5WKZWipMASZXN7tWpqjpJE2uizXDfzOvVYd8vMmkRML
DoGNs4qArImpYCIncQss0WpFktPANZGl6Em6pqoSQFAypNBk0CKtdx7T0RIYp1Z9l1KYJ1aH6/MlkddW
2eS1XWdlwUpJTbGSG1/Mp8vQZRwIkIAECUSILUgxCKCiAI8Xpk0TbOoO0RtrUqgU0TlzPT8YCXxbGexv5i
bXK+dm+Bwe0QESqkIAlGApe97aP2KLY5CD4DeR0OoNJAVQB0EyIRUKzOkYjQkxUbnATujVGjZ2iyyMEj
mbKzbo4OjIh+wxLwgYTWbLSQBACAiKmBmpVRqURKIREEUgqzIo++V0QopASIFDoHQxeSVVWVZKqsV
4nK2qOdeMyCWZIxBlNhHYEGklJK1NqQowojEwtoaSCJ9JA35UKGxfdXq0uXjYZblANJUdT4cdtE7TSwx9
h5EA1Ns+xijdfrkeA4AwkgKTrNCBBCIGSAxKCIiBcjIzyc9AgCIpNDL3fedeeabs+3t7ZSS95333lrd1HWMofc
BySGiQhQGARZETIkBAAgINaeECohIKRLNinVf9aLg/EgdE62d3RREStIlrxTGGEejSVPViIgmj31XzQ8h6Wr
RkmIAICJmFgEAIDQSGViAANTzbgFCETHGMDMAg8hkRdcVdU20OZR50Uq0gMySFVlkbJYLJAecABCfPy
KxCAACKjr9JoOoUDRorcthAQowgpiUx3B43A2nI8psWQ5D6JkTkSoGZVVVoDD5vp4vmiooRCJKKYGcFl
BBBEQNKaEAKvqfDY+IIiIiwAJAd98zfvyROSvIiF1mo5bQiXUahGLyAlorjcwAp25NggIICIgSBQVBhFmMUa
KRtBJBgyZqb2Aw52p1PTnjLl85SNvRWGWMITRN3YGQ733usmCCV4ACRMDp1MGAyKfFlOA0TgQJT1E
CEJ6/PUDy/NJXnHn0wQPkKAiSrFB0hc6MXlZdWeiNzXJzyqiGlDxDD0AIIkarJCwiBEgawZDKtDL69E9KKa
LTDwQAq13rl1uTfF53rUczHBTWnkagUqoNPsbIDIV18+Oj4D2C4iinNmZ+PmyIgJkRkbThmJBFBGKKL3/
NxmNfmzfHPRCiVhASgWQ5EQELe4YkWOYasVQKgPuESEmYFIoIMkoiWxrKES1AYqNs0/WImGWWDBI
pALImczrNq/Se3/zw7rOXP/jx9y6Xi6vXni0H9rte+8bXv+b7xB91HorJ+O0//RYDajGfxwbIIBASqOfLEWJK

SUQkJmUNKgqL/s6XbT/58JGyHbMhjAyCEXIH1kGTKAoPJuV4OLClQRookYQRJQoAEEFC0EaxJH7+fGRm
ZYiIlFLGWWOMtfbXf+UDt91yR5br/RtXlstljHE8Hucq3Ni5IuCtW08gibksB0qpZcU//L98V55nbduhpr/yh
u9/x9v/zmAw6Pv+//n3v/WR330/WFRapS66QZYP5WSnJ0OnbpckGhVadEUeOaUQjeULd97ShxADoR4
Z5iheJAIiEAErUBpjEMTTaqQZksqeN5jNHKHeXL/lP/2HT0L0bUjHhzdHo7Julhvr2113dPNg5+DgoCgKkTg
ebWRZ1nVelHrvv/u/fv9Tn3jlq1772//uAybPEyRJLDEhaB86EXnFq1+csDLk+q5G1sHz/8hmgAjowOYZKI
ohnL9lzWaOERAF7YrGSL7zEBBIEJFRtFUAEH1ChcJISsgpbRUADcpRn6qPfvCBzfUzbdv2fV/VM02GOVprr
VN7O1c59SLRGFMWE4H4zLOP11W/srrxqm97vSZMUe3sXj46vI7KFeVEq8xqg8bVzeKvvenVcQGUMad
kNGRO69zMDzsIwEoAlc2tsfSC29a0ybvgFQNihoigkGLLggIAoBBAAAEEgICIRLicjFan53/+537hO77tdSEF
JTHLiqqtZrNZ3SzathaRlERr8l2lSJQW59zs8GpdN8oOsnxwx6V7nC6qptnZvdK088XJzGWEqFMUa1fqGO
550Z1/8qeP/tTffxsprTX5LipkRcZLoESgxFhQhoqiOHfuFtLqeH4YYqdtaWMbQpuUphQFEJBFTm/PAAQg
RIaY+Vvuufe13/4dFpFEjmcLgTTIHI1HmSYYDy5fedrYTNgXBfV9r5Xq64Ok29WJ84T33n3XfNHsHe1436
VUpdRrQ8YqTZS01PXBxtrgyuWbd9/1EmOIUAuL0qxA+RgRCRRy4hBAGbW5tdr6BdfotOMAmmPSWifx
AATAICAAIIgCpw4hBYyACj71wIcfefDHLl68OBkPykEWY6xSPLi5J+yTxO2z6zdv3tTaKhKtGFLtdE9c5uPxd
Nx77vKMqnoBKLnL+r7XhLmzTVX5Pg6ybm24vXfSXn7usbBgO0qnsE1AxigRSSmRUhC5Wfjr13ereS8RG
KEca4qL5JcBETkmBAQBAAAUAEEkYIw+TidrXWgMZPfe+7K8SE8/+WWSWBSFUsoYpQwRekmL9bWBc
85mxlpbZkiqKVdoUqZLt33rdLLKoVmdyMXzW3fcfnFQZivrk8hiskE5Gr7wrleev+1lq9PJZGhAAJUWJKsFF
CRhQQBCQSBjSGFbpdgBodYWy3KMeFpr4Hl7g8DzbyAAAkBQmbbWtSf11ct7V557rhwOVqbrMfnBoF
wsFsE3J4vdsiwtLwGam0c3Mzd0WccMSpmMFsIL8doNVppKsumE6FxKqRxkj3z960U2HQwGZzc3O88x
5ZzwpOoe+uqD//ifv0MkMQQAyGzZ9+GUZgMLM3vvARASl1M3HA8R3GmaCgAAi8TT5yBqgQRCAIAA
Yq353fd/dDJZuXjx1vl84X0/GmdIdnGy3y33rGuLfKxtyFS6und5dTLN0apiFud/TjTp++cGQIF3z1z6uWXcP
jzanwyERXVJTSaTvku+l2Xtj44qV6x991tf6bSJrfdV41mSgNUGEQFIRGKMRGQsx+jLsiTrCBE5MSJKOr39K
V2RU+IOAMqgLZ3vww/84Pevra3s7OxUVWWs6vvehwopCDYSFoBLBgGZbW0Mx/myaz9DaVYM0GW
PToonPD/UmTuPll882N8bjcfz+eNIlaTjZrnX1Yfed4Ram2LRHYwHo8wNh5MpWo2IKcSUQkoppRBCD8C
IkiIoZWKMKQSNUcAp1IQiEgSRAE+bDgCNgIAuGxfTz/3xH3WL42axv6z7LB+ezE5SpOmkEFTO6kyBCoeC
yzxfv2X7luOrX5e48eDnHr7/O49d4hvL23a7ldkjz5x56cpoE+v+2tZWrpU+ONjFwaZWrvd+saz+0c/9k+sH
1zUwKO1jX65O4XjRL4Mdud7XmbitjbOXb+4GAUoRQUg7jIynsY4aldWxD9ADKhABUIgKUKtbts995IMfl
zYNpsPULx9/7Ct9OtraujMfrCL0gqzRk55tn/sBt9I0R8313ebg4Q+uDz9xcFS9+FY/Gh8/d/X2zRdOBtP/db
/RWTLjM5+T9PovPvilM5t3D8otBFf36ROf/G8BURn3xBNP/tcH/shmKvhWKX28P+cAwsFktO1sctnNtkM
vXdMPVx0qwlNuiAio8HncBhQRzFAIlCHnXDNr3vCG1//qO3/NWutTjL3/2jc+3feXy9GZzTNnHSqg/clwoj
X38wcP+2/d3rh/r38ofP1vlmSL3K4O563f/vTn/Jt/9nsH6lVNtXzo639u8kuD6eDKc/t7Nxdr6y/cOHu+zNe
UNgCUkuwc3Pgbb3sjOsHW1F0PIuIjMaQAbmLX1tzRwdJkTjmDYBBZhOF56JXnWwpQoBwJIRohUGWZ
/8nHv2TzjJlPmeO1na9cufoFInbWbm7kpT42wKuTawA/3k/vf+7GQ7Pf+4fJPTPO8q1zfvfqpr5U3POGD4
STEdP1nYNPP/rQn1l3/sX33rcyvcu4DWVLEJsYU5LgZdlVN/Z2f+jH3miS8tHHGCWxRJYEpKEYK4uqY3a5R
TfNJICvOyQQAQIQJEmMCsiQoBRjJYzKmU+//zPT1ZU8z5k5So9ofFfdvP7ZK1e+ql2HXhQevPCiofBIN9s/
Wv/s2dtvZg++9dkD/7If+jPfDDN7BrWnoJ+89uVzWy+6evkPCgWD1YGkNW0nNp+AOAETvDRdLyLVSf/r
7/ulT3z09ziKiEROSEol9k2gATiFpii6pkdQBMwICEYkAZ42rvg8FJSTwlor2NSVR7Kf/d1P59k4YhqWI1vmf
btM3CgIqdu7vvuFoU63XnI7u7cc+XPT6XRYRh1vmOEFCklnd9lxpKTqtEhHH3Vqm6kKUCo1JJkIlQCZoI2J
eh+7po+os6z89te9fFkvBXpFJkYGTJDICtohh6hiwNgnBAIAUBmh1nHp4TSrtYopjdcG5EC7um/R6OFsd4
FDzEH1bXjPr/6a0pCXQ42qrW6g6r77DW8+XvaO6t3DucOciiDY+aM/SPldpdks1r5FqseeePTfGBx7tTIan
BuvntVmTKpAsKiGSDnDUMSmJL5ffuB9v/GVx5798sOfU05rQylEQ4Y5cs3KsFWaCBg0EWlQgAqZWSqP
BCjAAJISICyqZqBs32ulVIw9OoFAycmlOy8yw+xkJgdHzoJztsjgcAHDoiQ9Op+t7u1cPbjye0W4HuCycmY5
XGezv3j6l4t82tTkrPa+mR/toOuVGTgakBaVlyjIElPiJqhX3f/63/rwW5VVCJwCi4hPUQFinkTM0VGQiKAii
OhT6FJKhRiFAQROkwEVYKLliTfOJBKBMJwW9bIFVo6sK9yYssVir5rtVxGKixfKbISohFjpcn27HKy8M6Ud
P39uefxUU//+/PiV+eRvVUefNPmmF4od91FB1ypimxlbnKNlZDwLMINWtlp2IQStdYoJkBFRKTCamyMb
FkkIAQTi6fQkI7QYT0c0AgopMSOdciQBAO6FDHIQIAALW7esS129/U1vqPyCiJi9Vdq5ZLPb3vz2f+j06JRc
xRhD8DH0iBJ9zQxt2zb1PIQUkvfeh75LEICddiOTTa0rjHEhBE7gLP3kz/zEc9efZeYYowRfTsp6UXMURZpA
dbOeUDNHREQcKhSAAIiShEFROSyVUov53KD67fe87957XsbcaAyAObKqqpN8t8BGW1/cPHnr4C18Myc
+Xs2G2UreHL3zR677z+398NBp576P3LDGlEGNEiXUzy4sVFtM21XxRxcR9186Xxzs7ey6bjCfrSilnbIg+pfC

TP/N2ysR777seE/su6ex0tkIxMgopA+yZW0AChOK0bxcKSErcygCAiqI42rv5uvtf/+5f/T/7flFVC2eItO3aFF
NniZpu2Syvbm4MPvXxD42HxbUbex//6Oc/9pk/Xlbd5tkXEEGKqK3d39+frIycge7kAI0DGtVd19aLqgmB/
fHNwx9+69tect/dl+6824D+3Bcf2N/fLccDZRUAOK264AFEQX9yEEkbZiZmVqIMZU6388BBENwpZSOyO
ChKZgZUVttXfctf+sd//+8lkb7tnHOCoIFFMMsyoZAbKwjT8fjK7rNf+i8f5tnOPa/7vjvuemnVtCJycLTXLtu2
6utm1ix7YKahMbpd2bjwsvvur3pu6vbw8NC37Tv+3o/MopeO+ibqzE6mU5+SCCstzmkgil2zrKLENlNQL
0EUqYyNBavt8jBIEnQjF/oeAhin+y6iAXJGAZI1H/yN3ywGY6UzpRShGKO01igMAAK90iWLFZEQ2hj6YdZ
MVoZtE3yoqkXb1k1TLWKbIgMrTFCT0kf7V7TQH37uiTf96PfpOPo/3v1Pp+fHh+3ym1+rOt+CaBFkjmVZ
GqRsK646G3bp6fq4cE4ptdhvdE6i0BisbsZTtQbRAhBgRE5icvuy+17+d/+3n7q5v2+L/L6XvHQxPyLURIQK
CusEolYSUyKKCLpq+Xh2AEJI4mwxLOPqauhamc+OfRv7et53HMGjEu/jM089cuHChcl0qzm8+cCfP/Hx//
rAG+/7C83V+f5yfrWIT1/tJTWT6WB7bXBLdvx3/upYIVnvW+X+0ceyhx+7jILcChiYruTzZSs9oZLTETfaMfk
6gcAr/+Krf+od/wAwAoBzbjqdIiRmD8hKKURGSTG0KYk1XpvY+2xZNSlCVmZWZVr7UdkpOLp5sOyrpu3
7FJZRRKTvfAPQ+lRaN1UmZnZ87sK39Z7f8hM/eH5j6+knbiIpGvntjc0F7m1Nz33HC9zO/CDQ2le/ctNlg4
e+sWdWcPNivqLzxx9dZEYWJ1FlJMBa58q3kaxiTm/90R9h6WKIAMAc89wqVFluAZjFxz603SLPcyIhNXry
8T8+e8t91mCAHpMxDpTKEkcAG7r9ZbUQfyxaC0cfGQAG+bhjyIpJlg2UGj795NPXrt+cH8DJ0S71mFIaab
u3O8sdXA03dtX0x14va4Odq7frN793AQipz659vd3RDQN4D+iAFJRljnqktNYMKCL/9l/93/OTvTKbuMxo
rQfl2BijtVYaYmqOjw+qo6fWN++qqs4NRoWtrJ5M13U9P1aolRoKVmWRrJLL3/i9II0ASNIqX2n69tbzL7e
ukFRUnpUtqzpICnUbCM27f+N9jz72RGa7QrsT34M4U8pwIj6EvDBHM5GUn443u3kDFSWJYFBlSERJIqq
h5hBFABFefNe3/PiPvG00HMYYq2oxna5aq8fjcQj97ORwd/c5TIe+a9fO3M0IhGaSNy+4eJtIub/7tNaGU
7M6Hpf5tavPfIWhFcxMOY3LZjTOi9EZiDnribBNQl0Q73WIEiNrW5ws4z95579sjhbFRgGkkjptCImZUUH
wLQCklBIzLEiiYKlTiORBRDBfLdpFoxSlxBDoP/zHD1249UJKqa6rg8Nd7zuXmZTiycnx4uR6SgvuK+PsML+
NTTMcjc6sbLtSmkWVUoVQ5fK1kZNnrj8zyM/6sO1GG3Z4qyvXKKjYPxQxKWVEnEAeoxHWfYgMRd91B
8f1+z/44YPmcFktkkdSnCQosjGysnQK7THGeMiEJFYEBRJgAoQcjAZmSB7Wzo7a1v/sP/j5qmq2zmxqQy
n1IJKkb5uFhjlRi+iR276pjQ6clgRpMhp11c5w4CcFEcxVfmZeHeT6bBUY1ZixyIcXlbv7cHZ4biuL/jKhY5og5
cwaxIQgSVS1rJvOrm+fe8vb34RgFCsQMTZjCELIgZuqlToBIFsBIhQQFvSCbgygdN/Es5eGfZ8W19hOCHr4
lX/9a199+KGtra08s6Shb70z/aBgiJ2mxveNcRFpcWYy2lx9wZMPfpgj1LN94PzatXTfX+4G0y0xmbVOeAh
u6AbrdRyNVr53OFqbHfxhXQUBZcxYWKeoSLuUcFEdn8zjz7zzXZjHEPqBnh4fzCAgsIABsGC8CSmoUqcYF
YFEYADdV1CuAxlofVWOxnR7oJj9xq/+vyUO7rrz7hh9TO3R0cGffvkL45H1PTjDVpkyy4ZFTcjaLrh/8i/85V
d+5D3/5WAGqGYk2Te+vLa+vZdW7zADVv3e+VuyiKMoGMKsmcOtt93/jUf/UJusC54UkXLLuhGwiwbqp
gvSZ2QAqI/LfD3jGvq5J0FJHDkQKVScuaw77vF0B0DlIBxXzuaiVF/D2vTc4498s695MOKu9cYiiBqPV+6//z
tTXLTVvjH1uICu32ur2kr2W//qP//w2y++9+27/+xd9qGvqdDFYhD+9Evzetnf8YpDS5Nej24cRYY/L8stNl/l
0atnyzSerDmdEiqR/OAw1W0ffHs8b4zgWpE1IqJ0TMA+2tKaAKpKcwYEYE5KafbhVKxFQFy7FaojyIem7
oLqtAdWJnv/v/+dbllPp1NjFRFrQyCRVE+w5HCTpPPd9Xp5HMJsktkPvfdT3/GdZ9qb63bwnE+1CDhdJP
He33v+vpiWG3MapjhgnYG4rDx37yu++5mv/tHaWqy92dnXVX0MYo22yubWZhJ7tIMf+6m3KaVPBQ6L
5tUx++TsCCwYRgLxishzCoAK9NaF7XDuZDnvMsh2r/QKBBQfnNw8M5i0XS2gWYLyGqSzJqVUKxRMKso
wcb02Pj8qxj/9LvMHH3n8wl1P3fj6Wifp/J3doFixefuxD9D5F49hML19vBLU+HDhgcZRwle+9IXxUD95Zff
kJAPm2tdKlcYNBqoksQL66GhhnOXIp7oypfD54O+KYNlkCLXwU5R8QgQBRnry4cPFsa7nWB+LM5oSKC
X/+z/76btfct9gOHZ2GBP1XaiadnbSLZdpuZR5s0gh2zhzp8rPffPpazdu4Iu/7a4X3PuRx57qX/99b9y/sart
pRde+oHX/uCkiW48WqNyKx9u3Xr+petnbnfutjL3ypQuv3W6upINykGxqigTpsRUN7FNTMpK4ucVWM
TeEI/KDJRFRAUOYYgkgkIgIJpMOtzrEQBNAuR8xXZL/JVffvfxyUKZTJJHZZCgcEVXHyZuGPtMD4ydBvExHo
w310Uhpt57+NGf/tuorr/yNa9av+0e7oaT9d2q09oUSg0iZlqPfKcy14wGG20ElxmW1iQijKAgJmAElqggC
6H5/+VuAAHood+nRJDORt0h94ZMTIEAE+issMywmMXCaXRQ5LqreTI9e3TzgIzt27qPXeJ+Miqnqy84nl
11NM20UhoAYkiZ0qt9XyGo5fKBRXP39poer6zP2rOFI9ATVxiyjnEAKU+KJ6NScOwDS9dok/pIk2IwO6kV
+5TgZF5leSm+et/vvCelhEinDAKEIKV+azRt9d7x/LoSxQlLoAYYAEfns5Skq70rVHsSJ5mDoUWe/NIv/jrHk
Gc6Rq+UGpbZ9RvPXrj1ou+OtYo21xwWWnFdHxtDRifFlJdNPrxjZ/doZbJq9Ml8doUI83wVzLbTq6IsmRG
zZtEpyXI5a0OzczAnwKpuE+PJsirywS/+m38hMGcOp2snSilIzKTWZv0oQMM4JSWOnVJlOT6oWsw3TLsI
pJUCEojMsro5QkuXLrziTd/7Q0bhZDRGlN29HUjx4m2XUHUQF0idNgKpD7FRSpVl2VfH2sn5W+5JHLzvf
F+19dyZDHWZlbewENJA6TwEipH7EH0M3/09rzl/8dZXvPzVK6OVB7/6Z1/+k88P18qUAqIighggczZAYAjA
4l+33lqDTcK8U4OOWGZ6EZkmMdmjFR2RB0oESWWOHYpwOTUxN+c53/QIKJd97XxXZ9J6XvCjFFtHH

uG8MiwhLr7Uui0l1sgPY3XHHX2Lm3ne+X4Jo51ZB68iEqIiKKBoC1n3fMaTAOztX3/GOt0w2aLkMiVEIjT
HMz+8fhJCYGRITgQarQs8gbZ9eIQPdB8XKGn0jVogWjCXuOTHoXLNymQ56nOUhHB20NMz8vBMEY+lj
H3qAJQAwct/3B85EbbFpOpdNRqNRaPazcnBm65KISOg637VeoxTW2q6vQDnfY4zew/OLdF3TB05v/u
G/6nJ2OqqFmdlIRpMAKIrRozAFZOZRF7Ywe+XmbV+cX9ZkLpf9Yj9FH8cTWw60NoXhPrnc9RBZONPp/F
FKx3Vj1aVBeTm0KKA0MHNZFsvlMnFKCQUHITUaLSgCtERDOzAxJuYMkCJroFwp6D10Te+9MDdZXgAa
BEhJmFkrUzf9vS+678pTXwka9heVLg0yWGuBGVlFH4PG1y6Lk745b9w3n3lqaHjVDc4s7IE0LcBt3WAw
VxqQWbNHIS8QmUopyvxMNGez0Yyb/a5PGfZdAgZSNmKaN01GxKxQFzqIiGt7ZX0wehQFBJTvY4wQ2
QAgEQtoHzqXl/NlozWxaABgwL7t2r7Zm93srPStFBqaZcA6ROU37xjN9rqNoG+dJZ/B1I3n/VFYu+1S1R2
2TabUHTjY46rpAoJQaFOuXWiCyYzOiFkOM7VZrvm2O5gvU4jWZCDgRnnVt/OTypGJKTGcKvMggqRA2
GitAdK1G/tJVOc5xFQ1bdv5LkRts2pxwswiynvftm2MsfYppfTcE0+bzAkpHGitAQWEebl3YiidaHpqIF/r5q
2pbmg80/Ez/qRNYQnRF0VkjqlvudPZwLZ9n4+KlHp0GL0chva38fJdiBtMmqCe1VYr8XF+dGht0cdWBDk
mC6ENpMlYpZX2iE5rrZQ9mVe+jwkkRvZ9IK2cc4rEOnOy7FIKIsI+dG3/vv/vN3VB3MW+7gdTF2rBmAB
ws9eLOjSbUbSd1/ERhL92bv2R69dfgsMD1Tex7dv+lnz9bLF6pdmhPvhiYFD3UbhrAUhi8i/2UG+uPQDd
TMM9zn17Pr0UaXlwooQTAZHO8zz0MbRHpRvqshTJg6imC41PdQgPPvLw0fE8xNR07fHxyU+89Y0xw
W/+1rtj9BFwUS2X85PE/WNPf3n93FrdB6l5ca0NnAICrbirOhyPrMaBPQj3nslB0X88OnxW4hOuPUQfCO
ciD8adP1g8vksRx2eLrvPCSjFI5Ne1+Ix4rfSZxE/fu3b20YPbeL0xgUGlkL3m3b8w2TxzvHt9kBfatmvrK598
7+9+z9/6213nfR+QbNu288XRZz/zyW8++N82zt/zipf+xc/+yeft4w/VqB7PmWKCBGYwfu1f+Z4//NTHrSl
StzRF6mtOyLZXY6ZEYV4CMA3J3r+gO5T7gKv2c8JFIK1jH6KItir2yeUqAGG2aohNlASEqe/+ep+7iBeGa3u
2eqztHoLmjkZt6Lzh2FKYRHyO1D9/34dC2/ynn//Z1b1ZG9o3feJj0sfg03K5jH3c33/uA7/zb49uLorR8Nxx
70DNKd6koEY5a7KuVAjVydyVRfIp+Ma3cXWzCCe8zRaacMjdmaJAA43Th0fzFUeXNY1iyifZQOx9y+yP0
myWonM2oohPOJ7aZRcBhJ0Cz2/t3BjLjWL1qWr3G9Tup1AqOmMKE2Wp4+2xWIkKB4PPxxv3xdF2Pq
0lfrM6nLzlb77qtd917erTv/xL78oy1TetNYpzBT2GtgcBtAqRR5NhkpiS9K0viwytgyjL43mWcVPBy2m4kI4
RvjXlEzu8itW1FJ8Z+WoRVsej79XTtaUOqXvUzz/LFa6Uvu2ctjhYy+uq00SYBCdZ2YafrMs2+RzUcySPQ+9
ElcbZBA+q+T3eDKxZEHQJLm6d295d/H666bV9VgVvMMtzkxnvvYTIzEQUYyQtxLo77vVIjSbDfDxeLpeL6
8cEAJmyhCrTaw1kq3B0vdNFTsvuPjO+XZUNz49jfE7i7fnKphk+3OzergbXuP0MLTWSCKhMSSu4ub2y9C
F1fR8CsSSGtZVsG4qnqC5mXpPa8jIQBQCdkqFo6xCSuJRfGG42vnqkPzjScjB2SbiZL7XWZLTEZApbL2rtiK
MoVopjIh1jFABrKXYsApKACC6M8smS82l+s1rudqlf14Ni+NevpyXB9VBHUhtqqEke4cWRpFAikOXoY8+
oEAD+O4B7M7QnIeWwAAAAElFTkSuQmCC\n"

        },

        "metadata": {},

        "execution_count": 32

      }

    ]

  },

  {

   "cell_type": "code",

   "source": [

    "x = image.img_to_array(img)"

   ],

   "metadata": {

    "id": "d3AL9SaGWU2N"

```
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "x"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "yihZyOxDYyte",
    "outputId": "14949ad9-1a03-4d1b-86ff-3f6715d253db"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array([[[14., 22.,  7.],\n",
       "        [11., 22.,  6.],\n",
       "        [ 8., 19.,  3.],\n",
       "        ...,\n",
       "        [32., 47., 24.],\n",
       "        [30., 48., 22.],\n",
       "        [33., 49., 23.]],\n",
       "\n",
       "       [[13., 20., 12.],\n",
"        [11., 21., 10.],\n",
"        [11., 22.,  8.],\n",
"        ...,\n",
"        [37., 51., 26.],\n",
"        [35., 49., 26.],\n",
"        [25., 45., 20.]],\n",
"\n",
"       [[19., 30., 16.],\n",
"        [19., 31., 17.],\n",
"        [16., 29., 12.],\n",
"        ...,\n",
"        [31., 47., 20.],\n",
"        [28., 49., 18.],\n",
"        [27., 43., 17.]],\n",
"\n",
"       ...,\n",
"\n",
"       [[15., 17.,  6.],\n",
"        [ 2.,  9.,  2.],\n",
"        [ 2.,  9.,  1.],\n",
"        ...,\n",
"        [ 8., 21., 11.],\n",
"        [ 2., 12.,  3.],\n",
"        [ 9., 16.,  9.]],\n",
"\n",
"       [[12., 20.,  9.],\n",
"        [ 1.,  8.,  1.],\n",
"        [ 5., 10.,  3.],\n",
"        ...,\n",
"        [ 3.,  8.,  2.],\n",
"        [ 6., 16.,  5.],\n",
       "        [ 5.,  7.,  4.]],\n",
       "\n",
       "       [[24., 27., 18.],\n",
       "        [11., 21., 13.],\n",
       "        [ 8., 13.,  6.],\n",
       "        ...,\n",
       "        [ 1.,  6.,  0.],\n",
       "        [ 2.,  9.,  1.],\n",
       "        [ 2.,  9.,  1.]]], dtype=float32)"
      ]
     },
     "metadata": {},
     "execution_count": 49
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "x.ndim"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "p4pRB7AMZoX4",
    "outputId": "74bfef14-8fb3-4cd6-89dc-cfe963f3626d"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "3"
      ]
     },
     "metadata": {},
     "execution_count": 50
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "x = np.expand_dims(x,axis = 0)"
   ],
   "metadata": {
    "id": "mAUk7tH7aFq4"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "x.ndim"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
  "id": "D-MSkf0taJmw",
  "outputId": "1c27fced-2208-4995-c574-91d117b9e372"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "4"
    ]
   },
   "metadata": {},
   "execution_count": 52
  }
 ]
},
{
 "cell_type": "code",
 "source": [
  "pred = model.predict(x)"
 ],
 "metadata": {
  "id": "t5DQqIIVaK1x"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
    "pred"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "aP7rI4BXaNYL",
    "outputId": "e1046fae-9f8d-43fe-b0fd-5912eb8f2bf6"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array([[1., 0., 0., 0., 0.]], dtype=float32)"
      ]
     },
     "metadata": {},
     "execution_count": 54
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "labels=[\"Daisy\",\"Dandalion\",\"Rose\",\"Sunflower\",\"Tulip\"]"
   ],
   "metadata": {
    "id": "LWIqmHldaQDs"
   },
 "execution_count": 20,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "np.argmax(pred)"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "THFchlaLbEhE",
  "outputId": "65839c90-60a9-4c71-d250-9ac5c3a154b3"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "0"
    ]
   },
   "metadata": {},
   "execution_count": 56
  }
 ]
},
{
 "cell_type": "code",
  "source": [
    "labels[np.argmax(pred)]"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
    },
    "id": "i56HhEIgbG0o",
    "outputId": "cfec579d-d110-44a0-f5a8-cee6980ab465"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "'Daisy'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
          "type": "string"
        }
      },
      "metadata": {},
      "execution_count": 57
    }
  ]
},
{
  "cell_type": "code",
  "source": [],
   "metadata": {
    "id": "SyoEw6NmbKHt"
   },
   "execution_count": 20,
   "outputs": []
  }
 ]
}
```