```
{"nbformat":4,"nbformat_minor":0,"metadata":{"colab":{"provenance":[{"file_id":"1WXVD1Sm2D55o-
C3LIJYj7fr8rAlRRIZh","timestamp":1665067761902}],"collapsed_sections":[],"authorship_tag":"ABX9Ty
M2+tNGc4yfNb5Qs/vCvgAm"},"kernelspec":{"name":"python3","display_name":"Python
3"},"language_info":{"name":"python"}},"cells":[{"cell_type":"code","execution_count":null,"metadata":
{"id":"HqdWxnkbxR1Q"},"outputs":[],"source":["{\n","  \"nbformat\": 4,\n","  \"nbformat_minor\":
0,\n","  \"metadata\": {\n","    \"colab\": {\n","      \"provenance\": [],\n","      \"collapsed_sections\":
[]\n","    },\n","    \"kernelspec\": {\n","      \"name\": \"python3\",\n","      \"display_name\": \"Python
3\"\n","    },\n","    \"language_info\": {\n","      \"name\": \"python\"\n","    }\n","  },\n","  \"cells\":
[\n","    {\n","      \"cell_type\": \"markdown\",\n","      \"source\": [\n","        \"# **Assignment
3**\"\n","      ],\n","      \"metadata\": {\n","        \"id\": \"j8bF22iNxMDx\"\n","      }\n","    },\n","    {\n","
\"cell_type\": \"markdown\",\n","      \"source\": [\n","        \"# **CNN MODEL FOR FLOWER
CLASSIFICATION**\\n\",\n","        \"# **Trained by Team ID : PNT2022TMID17050**\"\n","      ],\n","
\"metadata\": {\n","        \"id\": \"BdPpvl8Q0IlY\"\n","      }\n","    },\n","    {\n","      \"cell_type\":
\"markdown\",\n","      \"source\": [\n","        \"# **Pre-Requisites**\"\n","      ],\n","      \"metadata\":
{\n","        \"id\": \"d5mO5lxMq9BQ\"\n","      }\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","
\"execution_count\": 1,\n","      \"metadata\": {\n","        \"colab\": {\n","          \"base_uri\":
\"https://localhost:8080/\"\n","        },\n","        \"id\": \"h4I4dfJ6p3kx\",\n","        \"outputId\":
\"5605b9f7-ebc0-4587-88c6-268194f1335d\"\n","      },\n","      \"outputs\": [\n","        {\n","
\"output_type\": \"stream\",\n","          \"name\": \"stdout\",\n","          \"text\": [\n","            \"Drive
already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount(\\\"/content/drive\\\", force_remount=True).\\n\"\n","          ]\n","        }\n","      ],\n","
\"source\": [\n","        \"from google.colab import drive\\n\",\n","
\"drive.mount('/content/drive')\"\n","      ]\n","    },\n","    {\n","      \"cell_type\": \"markdown\",\n","
\"source\": [\n","        \"# **STEP 1 UNZIP FILES**\"\n","      ],\n","      \"metadata\": {\n","        \"id\":
\"B4Kss5zOrXUz\"\n","      }\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","
\"cd/content/drive/MyDrive/AI_IBM\"\n","      ],\n","      \"metadata\": {\n","        \"colab\": {\n","
\"base_uri\": \"https://localhost:8080/\"\n","        },\n","        \"id\": \"DRNhAxfvrWKc\",\n","
\"outputId\": \"f1a087a8-6f33-4d81-ff79-e50f4c1ff623\"\n","      },\n","      \"execution_count\": 2,\n","
\"outputs\": [\n","        {\n","          \"output_type\": \"stream\",\n","          \"name\": \"stdout\",\n","
\"text\": [\n","            \"/content/drive/MyDrive/AI_IBM\\n\"\n","          ]\n","        }\n","      ]\n","    },\n","
{\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","        \"!unzip Flowers-
Dataset.zip\"\n","      ],\n","      \"metadata\": {\n","        \"colab\": {\n","          \"base_uri\":
\"https://localhost:8080/\"\n","        },\n","        \"id\": \"B_hJ27NKrhKz\",\n","        \"outputId\":
\"9b319781-61d1-4a0e-ea93-f279d067bfc7\"\n","      },\n","      \"execution_count\": 3,\n","
\"outputs\": [\n","        {\n","          \"output_type\": \"stream\",\n","          \"name\": \"stdout\",\n","
\"text\": [\n","            \"Archive:  Flowers-Dataset.zip\\n\",\n","            \"replace
flowers/daisy/100080576_f52e8ee070_n.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
N\\n\"\n","          ]\n","        }\n","      ]\n","    },\n","    {\n","      \"cell_type\": \"markdown\",\n","
\"source\": [\n","        \"# **STEP 2 Image** **Augumentation**\"\n","      ],\n","      \"metadata\":
{\n","        \"id\": \"hvG8h70rrphq\"\n","      }\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","
\"source\": [\n","        \"from tensorflow.keras.preprocessing.image import
ImageDataGenerator\"\n","      ],\n","      \"metadata\": {\n","        \"id\": \"itQt2Ad8rtk8\"\n","      },\n","
\"execution_count\": 4,\n","      \"outputs\": []\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","
\"source\": [\n","        \"train_datagen=ImageDataGenerator(rescale=1./255,
zoom_range=0.2,horizontal_flip=True,vertical_flip=False)\"\n","      ],\n","      \"metadata\": {\n","
\"id\": \"9yZUiTxnr0UN\"\n","      },\n","      \"execution_count\": 5,\n","      \"outputs\": []\n","    },\n","
{\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","
\"test_datagen=ImageDataGenerator(rescale=1./255)\"\n","      ],\n","      \"metadata\": {\n","
\"id\": \"zD7ristVr3F3\"\n","      },\n","      \"execution_count\": 6,\n","      \"outputs\": []\n","    },\n","
{\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","
\"x_train=train_datagen.flow_from_directory(r\\\"/content/drive/MyDrive/AI_IBM/flowers\\\",target_s
ize=(64,64),class_mode='categorical',batch_size=24)\"\n","      ],\n","      \"metadata\": {\n","
\"colab\": {\n","          \"base_uri\": \"https://localhost:8080/\"\n","        },\n","        \"id\":
\"BjQo5zGHuHN4\",\n","        \"outputId\": \"d3d1e296-e74d-4e52-cce8-8d26459d10f1\"\n","      },\n","
\"execution_count\": 7,\n","      \"outputs\": [\n","        {\n","          \"output_type\": \"stream\",\n","
\"name\": \"stdout\",\n","          \"text\": [\n","            \"Found 4317 images belonging to 5
classes.\\n\"\n","          ]\n","        }\n","      ]\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","
\"source\": [\n","
\"x_test=test_datagen.flow_from_directory(r\\\"/content/drive/MyDrive/AI_IBM/flowers\\\",target_siz
e=(64,64),class_mode='categorical',batch_size=24)\"\n","      ],\n","      \"metadata\": {\n","
\"colab\": {\n","          \"base_uri\": \"https://localhost:8080/\"\n","        },\n","        \"id\":
\"e4YJwWrCukDq\",\n","        \"outputId\": \"e71a3e44-6642-4592-fa96-7af9c6edb08f\"\n","      },\n","
\"execution_count\": 8,\n","      \"outputs\": [\n","        {\n","          \"output_type\": \"stream\",\n","
\"name\": \"stdout\",\n","          \"text\": [\n","            \"Found 4317 images belonging to 5
classes.\\n\"\n","          ]\n","        }\n","      ]\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","
\"source\": [\n","        \"x_train.class_indices\"\n","      ],\n","      \"metadata\": {\n","        \"colab\":
{\n","          \"base_uri\": \"https://localhost:8080/\"\n","        },\n","        \"id\": \"EgBhHHYTuv4X\",\n","
\"outputId\": \"8a9f62e0-7d2b-4138-c5ce-4ca16b78fbd1\"\n","      },\n","      \"execution_count\":
9,\n","      \"outputs\": [\n","        {\n","          \"output_type\": \"execute_result\",\n","          \"data\":
{\n","            \"text/plain\": [\n","              \"{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip':
4}\"\n","            ]\n","          },\n","          \"metadata\": {},\n","          \"execution_count\":
9\n","        }\n","      ]\n","    },\n","    {\n","      \"cell_type\": \"markdown\",\n","      \"source\": [\n","
\"# **Step -3 Initializing CNN And Create Model**\"\n","      ],\n","      \"metadata\": {\n","        \"id\":
\"O5cz-9qOJM_s\"\n","      }\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","
\"from tensorflow.keras.models import Sequential\\n\",\n","        \"from tensorflow.keras.layers import
Dense,Convolution2D,MaxPooling2D,Flatten\"\n","      ],\n","      \"metadata\": {\n","        \"id\":
\"QAUHi2otRcoC\"\n","      },\n","      \"execution_count\": 10,\n","      \"outputs\": []\n","    },\n","
{\n","      \"cell_type\": \"markdown\",\n","      \"source\": [\n","        \"# **Step -4 Add
layers**\"\n","      ],\n","      \"metadata\": {\n","        \"id\": \"xew7skua3a0z\"\n","      }\n","    },\n","
{\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","        \"model=Sequential()\"\n","      ],\n","
\"metadata\": {\n","        \"id\": \"dack9NXYR2t6\"\n","      },\n","      \"execution_count\": 11,\n","
\"outputs\": []\n","    },\n","    {\n","      \"cell_type\": \"markdown\",\n","      \"source\": [\n","        \"#
**4.1 Input Layers (Convolution ,MaxPooling,Flatten)**\"\n","      ],\n","      \"metadata\": {\n","
\"id\": \"SzIvL8Q52DFR\"\n","      }\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","      \"source\":
[\n","        \"model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))\"\n","      ],\n","
\"metadata\": {\n","        \"id\": \"qPUbKxHGR7EX\"\n","      },\n","      \"execution_count\": 12,\n","
\"outputs\": []\n","    },\n","    {\n","      \"cell_type\": \"code\",\n","      \"source\": [\n","
\"model.add(MaxPooling2D(pool_size=(2,2)))\"\n","      ],\n","      \"metadata\": {\n","        \"id\":
```

\"IBGMZ7sSSAIB\"\n","   },\n","   \"execution_count\": 13,\n","   \"outputs\": []\n","   },\n","
{\n","   \"cell_type\": \"code\",\n","   \"source\": [\n","   \"model.add(Flatten())\"\n","   ],\n","
\"metadata\": {\n","   \"id\": \"c65fXm9KSErL\"\n","   },\n","   \"execution_count\": 14,\n","
\"outputs\": []\n","   },\n","   {\n","   \"cell_type\": \"code\",\n","   \"source\": [\n","
\"model.summary()\"\n","   ],\n","   \"metadata\": {\n","   \"colab\": {\n","   \"base_uri\":
\"https://localhost:8080/\"\n","   },\n","   \"id\": \"-go5E-VbSIau\",\n","   \"outputId\":
\"1f46f35d-1950-4456-bce9-16a06053d40f\"\n","   },\n","   \"execution_count\": 15,\n","
\"outputs\": [\n","   {\n","   \"output_type\": \"stream\",\n","   \"name\": \"stdout\",\n","
\"text\": [\n","   \"Model: \\\"sequential\\\"\\n\",\n","
\"____\\n\",\n","   \"
Layer (type)   Output Shape   Param #   \\n\",\n","
\"===\\n\",\n","   \"
conv2d (Conv2D)   (None, 62, 62, 32)   896   \\n\",\n","   \"
\\n\",\n","   \" max_pooling2d (MaxPooling2D (None, 31, 31, 32)   0   \\n\",\n","   \")
\\n\",\n","   \"   \\n\",\n","   \" flatten (Flatten)   (None,
30752)   0   \\n\",\n","   \"   \\n\",\n","
\"===\\n\",\n","
\"Total params: 896\\n\",\n","   \"Trainable params: 896\\n\",\n","   \"Non-trainable params:
0\\n\",\n","
\"____\\n\"\n","   ]\n","
}\n","   ]\n","   },\n","   {\n","   \"cell_type\": \"markdown\",\n","   \"source\": [\n","   \"# **4.2
Hidden Layers**\"\n","   ],\n","   \"metadata\": {\n","   \"id\":
\"f4ZSQPIFJaeb\"\n","   }\n","   },\n","   {\n","   \"cell_type\": \"code\",\n","   \"source\": [\n","
\"model.add(Dense(300,activation='relu'))\\n\",\n","
\"model.add(Dense(150,activation='relu'))\"\n","   ],\n","   \"metadata\": {\n","   \"id\":
\"x8MIUG1PSZ21\"\n","   },\n","   \"execution_count\": 16,\n","   \"outputs\": []\n","   },\n","
{\n","   \"cell_type\": \"markdown\",\n","   \"source\": [\n","   \"# **4.3 Output
Layers**\"\n","   ],\n","   \"metadata\": {\n","   \"id\": \"PNLk8KHHJf3K\"\n","   }\n","   },\n","
{\n","   \"cell_type\": \"code\",\n","   \"source\": [\n","
\"model.add(Dense(5,activation='softmax'))\"\n","   ],\n","   \"metadata\": {\n","   \"id\":
\"grI0IbuwSeqO\"\n","   },\n","   \"execution_count\": 17,\n","   \"outputs\": []\n","   },\n","   {\n","
\"cell_type\": \"code\",\n","   \"source\": [\n","
\"model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])\"\n","   ],\n","
\"metadata\": {\n","   \"id\": \"l44vMW4QShaw\"\n","   },\n","   \"execution_count\": 18,\n","
\"outputs\": []\n","   },\n","   {\n","   \"cell_type\": \"code\",\n","   \"source\": [\n","
\"len(x_train)\"\n","   ],\n","   \"metadata\": {\n","   \"colab\": {\n","   \"base_uri\":
\"https://localhost:8080/\"\n","   },\n","   \"id\": \"Beiar4NESkL4\",\n","   \"outputId\":
\"4b264b09-51b5-4786-b2a8-d60ac129229d\"\n","   },\n","   \"execution_count\": 19,\n","
\"outputs\": [\n","   {\n","   \"output_type\": \"execute_result\",\n","   \"data\": {\n","
\"text/plain\": [\n","   \"180\"\n","   ]\n","   },\n","   \"metadata\": {},\n","
\"execution_count\": 19\n","   }\n","   ]\n","   },\n","   {\n","   \"cell_type\": \"markdown\",\n","
\"source\": [\n","   \"# **Step -5 Train the Model**\"\n","   ],\n","   \"metadata\": {\n","
\"id\": \"Y9f3ElSv3Nc6\"\n","   }\n","   },\n","   {\n","   \"cell_type\": \"code\",\n","   \"source\":
[\n","   \"model.fit_generator(x_train,steps_per_epoch=len(x_train), validation_data=x_test,

validation_steps=len(x_test), epochs= 30)\"\n","   ],\n","   \"metadata\": {\n","   \"colab\": {\n","
\"base_uri\": \"https://localhost:8080/\"\n","   },\n","   \"id\": \"ATt0m5Cv6R-w\",\n","
\"outputId\": \"734d2b05-c864-450f-a46f-8ce129904306\"\n","   },\n","   \"execution_count\":
20,\n","   \"outputs\": [\n","   {\n","   \"output_type\": \"stream\",\n","   \"name\":
\"stderr\",\n","   \"text\": [\n","   \"/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be
removed in a future version. Please use `Model.fit`, which supports generators.\\n\",\n","   \"
\\\"\\\"\\\"Entry point for launching an IPython kernel.\\n\"\n","   ]\n","   },\n","   {\n","
\"output_type\": \"stream\",\n","   \"name\": \"stdout\",\n","   \"text\": [\n","   \"Epoch
1/30\\n\",\n","   \"180/180 [==============================] - 393s 2s/step - loss: 1.3213 -
accuracy: 0.4714 - val_loss: 1.1275 - val_accuracy: 0.5532\\n\",\n","   \"Epoch 2/30\\n\",\n","
\"180/180 [==============================] - 74s 409ms/step - loss: 1.0600 - accuracy: 0.5854 -
val_loss: 0.9406 - val_accuracy: 0.6301\\n\",\n","   \"Epoch 3/30\\n\",\n","   \"180/180
[==============================] - 73s 405ms/step - loss: 0.9678 - accuracy: 0.6247 - val_loss:
0.9603 - val_accuracy: 0.6203\\n\",\n","   \"Epoch 4/30\\n\",\n","   \"180/180
[==============================] - 77s 429ms/step - loss: 0.8884 - accuracy: 0.6546 - val_loss:
0.8187 - val_accuracy: 0.6938\\n\",\n","   \"Epoch 5/30\\n\",\n","   \"180/180
[==============================] - 76s 422ms/step - loss: 0.8358 - accuracy: 0.6787 - val_loss:
0.7393 - val_accuracy: 0.7225\\n\",\n","   \"Epoch 6/30\\n\",\n","   \"180/180
[==============================] - 75s 418ms/step - loss: 0.7924 - accuracy: 0.6965 - val_loss:
0.8389 - val_accuracy: 0.6928\\n\",\n","   \"Epoch 7/30\\n\",\n","   \"180/180
[==============================] - 73s 405ms/step - loss: 0.7521 - accuracy: 0.7158 - val_loss:
0.8503 - val_accuracy: 0.6789\\n\",\n","   \"Epoch 8/30\\n\",\n","   \"180/180
[==============================] - 74s 411ms/step - loss: 0.7048 - accuracy: 0.7313 - val_loss:
0.6492 - val_accuracy: 0.7521\\n\",\n","   \"Epoch 9/30\\n\",\n","   \"180/180
[==============================] - 72s 400ms/step - loss: 0.6502 - accuracy: 0.7521 - val_loss:
0.6458 - val_accuracy: 0.7438\\n\",\n","   \"Epoch 10/30\\n\",\n","   \"180/180
[==============================] - 74s 409ms/step - loss: 0.6182 - accuracy: 0.7684 - val_loss:
0.5721 - val_accuracy: 0.7818\\n\",\n","   \"Epoch 11/30\\n\",\n","   \"180/180
[==============================] - 72s 402ms/step - loss: 0.5662 - accuracy: 0.7931 - val_loss:
0.5968 - val_accuracy: 0.7725\\n\",\n","   \"Epoch 12/30\\n\",\n","   \"180/180
[==============================] - 72s 401ms/step - loss: 0.5600 - accuracy: 0.7908 - val_loss:
0.6907 - val_accuracy: 0.7612\\n\",\n","   \"Epoch 13/30\\n\",\n","   \"180/180
[==============================] - 72s 399ms/step - loss: 0.5064 - accuracy: 0.8138 - val_loss:
0.5185 - val_accuracy: 0.8117\\n\",\n","   \"Epoch 14/30\\n\",\n","   \"180/180
[==============================] - 71s 394ms/step - loss: 0.4830 - accuracy: 0.8249 - val_loss:
0.3613 - val_accuracy: 0.8673\\n\",\n","   \"Epoch 15/30\\n\",\n","   \"180/180
[==============================] - 71s 397ms/step - loss: 0.4650 - accuracy: 0.8196 - val_loss:
0.3396 - val_accuracy: 0.8768\\n\",\n","   \"Epoch 16/30\\n\",\n","   \"180/180
[==============================] - 71s 393ms/step - loss: 0.4117 - accuracy: 0.8559 - val_loss:
0.3472 - val_accuracy: 0.8738\\n\",\n","   \"Epoch 17/30\\n\",\n","   \"180/180
[==============================] - 71s 397ms/step - loss: 0.3892 - accuracy: 0.8631 - val_loss:
0.3314 - val_accuracy: 0.8826\\n\",\n","   \"Epoch 18/30\\n\",\n","   \"180/180
[==============================] - 70s 389ms/step - loss: 0.3441 - accuracy: 0.8726 - val_loss:

```
0.4008 - val_accuracy: 0.8589\\n\",\n","          \"Epoch 19/30\\n\",\n","          \"180/180
[==============================] - 73s 404ms/step - loss: 0.3467 - accuracy: 0.8719 - val_loss:
0.2484 - val_accuracy: 0.9060\\n\",\n","          \"Epoch 20/30\\n\",\n","          \"180/180
[==============================] - 72s 398ms/step - loss: 0.3327 - accuracy: 0.8758 - val_loss:
0.2234 - val_accuracy: 0.9210\\n\",\n","          \"Epoch 21/30\\n\",\n","          \"180/180
[==============================] - 73s 403ms/step - loss: 0.2807 - accuracy: 0.9009 - val_loss:
0.2830 - val_accuracy: 0.9036\\n\",\n","          \"Epoch 22/30\\n\",\n","          \"180/180
[==============================] - 70s 392ms/step - loss: 0.2751 - accuracy: 0.9013 - val_loss:
0.2392 - val_accuracy: 0.9141\\n\",\n","          \"Epoch 23/30\\n\",\n","          \"180/180
[==============================] - 73s 404ms/step - loss: 0.2549 - accuracy: 0.9097 - val_loss:
0.2221 - val_accuracy: 0.9189\\n\",\n","          \"Epoch 24/30\\n\",\n","          \"180/180
[==============================] - 72s 399ms/step - loss: 0.2412 - accuracy: 0.9243 - val_loss:
0.2029 - val_accuracy: 0.9291\\n\",\n","          \"Epoch 25/30\\n\",\n","          \"180/180
[==============================] - 72s 402ms/step - loss: 0.2360 - accuracy: 0.9199 - val_loss:
0.1965 - val_accuracy: 0.9307\\n\",\n","          \"Epoch 26/30\\n\",\n","          \"180/180
[==============================] - 72s 401ms/step - loss: 0.2199 - accuracy: 0.9201 - val_loss:
0.1919 - val_accuracy: 0.9331\\n\",\n","          \"Epoch 27/30\\n\",\n","          \"180/180
[==============================] - 72s 400ms/step - loss: 0.2008 - accuracy: 0.9363 - val_loss:
0.1218 - val_accuracy: 0.9560\\n\",\n","          \"Epoch 28/30\\n\",\n","          \"180/180
[==============================] - 73s 406ms/step - loss: 0.1889 - accuracy: 0.9310 - val_loss:
0.2838 - val_accuracy: 0.9108\\n\",\n","          \"Epoch 29/30\\n\",\n","          \"180/180
[==============================] - 70s 389ms/step - loss: 0.2046 - accuracy: 0.9275 - val_loss:
0.2116 - val_accuracy: 0.9307\\n\",\n","          \"Epoch 30/30\\n\",\n","          \"180/180
[==============================] - 70s 392ms/step - loss: 0.1886 - accuracy: 0.9372 - val_loss:
0.2091 - val_accuracy: 0.9280\\n\"\n","        ]\n","      },\n","      {\n","        \"output_type\":
\"execute_result\",\n","        \"data\": {\n","          \"text/plain\": [\n","
\"<keras.callbacks.History at 0x7f3e15438e50>\"\n","          ]\n","        },\n","        \"metadata\":
{},\n","        \"execution_count\": 20\n","      }\n","    ]\n","  },\n","  {\n","    \"cell_type\":
\"markdown\",\n","    \"source\": [\n","      \"# **Step -6 Save The model**\"\n","    ],\n","
\"metadata\": {\n","      \"id\": \"1uK880jw9Kru\"\n","    }\n","  },\n","  {\n","    \"cell_type\":
\"code\",\n","    \"source\": [\n","
\"model.save('Flowers_classification_model1.h5')\"\n","    ],\n","    \"metadata\": {\n","      \"id\":
\"scoaKurE9FZk\"\n","    },\n","    \"execution_count\": 21,\n","    \"outputs\": []\n","  },\n","  {\n","
\"cell_type\": \"markdown\",\n","    \"source\": [\n","      \"# **Step -7 Test The
model**\"\n","    ],\n","    \"metadata\": {\n","      \"id\": \"YAH2UVpi9RMV\"\n","    }\n","  },\n","
{\n","    \"cell_type\": \"code\",\n","    \"source\": [\n","      \"ls\"\n","    ],\n","    \"metadata\":
{\n","      \"colab\": {\n","        \"base_uri\": \"https://localhost:8080/\"\n","      },\n","      \"id\": \"Z-
co6hBAEmzg\",\n","      \"outputId\": \"bf8a661d-3210-4695-dcb7-48e6f365dfce\"\n","    },\n","
\"execution_count\": 22,\n","    \"outputs\": [\n","      {\n","        \"output_type\": \"stream\",\n","
\"name\": \"stdout\",\n","        \"text\": [\n","
\"\\u001b[0m\\u001b[01;34mflowers\\u001b[0m/  Flowers_classification_model1.h5  Flowers-
Dataset.zip  video.mp4\\n\"\n","        ]\n","      }\n","    ]\n","  },\n","  {\n","    \"cell_type\":
\"code\",\n","    \"source\": [\n","      \"import numpy as np\\n\",\n","      \"from
tensorflow.keras.models import load_model\\n\",\n","      \"from tensorflow.keras.preprocessing
```

import image\"\n","   ],\n","   \"metadata\": {\n","   \"id\": \"mJvRRo7VvkeO\"\n","   },\n","
\"execution_count\": 23,\n","   \"outputs\": []\n","   },\n","   {\n","   \"cell_type\": \"code\",\n","
\"source\": [\n","   \"# Load the model\\n\",\n","
\"model=load_model('Flowers_classification_model1.h5')\"\n","   ],\n","   \"metadata\": {\n","
\"id\": \"xo6F_4jw9KBZ\"\n","   },\n","   \"execution_count\": 24,\n","   \"outputs\": []\n","   },\n","
{\n","   \"cell_type\": \"code\",\n","   \"source\": [\n","
\"img=image.load_img(r\\\"/content/s3.jpg\\\",target_size=(64,64))\\n\",\n","
\"x=image.img_to_array(img)\\n\",\n","   \"x=np.expand_dims(x,axis=0)\\n\",\n","
\"y=np.argmax(model.predict(x),axis=1)\\n\",\n","   \"# x_train.class_indices\\n\",\n","
\"index=['daisy','dandelion','rose','sunflower','tulip']\\n\",\n","   \"index[y[0]]\"\n","   ],\n","
\"metadata\": {\n","   \"colab\": {\n","   \"base_uri\": \"https://localhost:8080/\",\n","
\"height\": 35\n","   },\n","   \"id\": \"2rnrfMAf-AB9\",\n","   \"outputId\": \"c6357a8b-5163-4884-c82e-05651a65571c\"\n","   },\n","   \"execution_count\": 38,\n","   \"outputs\": [\n","
{\n","   \"output_type\": \"execute_result\",\n","   \"data\": {\n","   \"text/plain\": [\n","
\"'sunflower'\"\n","   ],\n","   \"application/vnd.google.colaboratory.intrinsic+json\": {\n","
\"type\": \"string\"\n","   }\n","   },\n","   \"metadata\": {},\n","   \"execution_count\":
38\n","   }\n","   ]\n","   },\n","   {\n","   \"cell_type\": \"markdown\",\n","   \"source\": [\n","
\"# **We Achieved 93 percent of accuracy with this model** \\n\",\n","   \"# **Trained by Team ID :
PNT2022TMID17050**\"\n","   ],\n","   \"metadata\": {\n","   \"id\":
\"2f85wU8fL0Si\"\n","   }\n","   }\n","]\n","}"]},{"cell_type":"code","source":[],"metadata":{"id":"VYZTfv_uxTvF"},"execution_count":null,"outputs":[]}]}