{

"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": [

"# \*\*Assignment 3\*\*"

],

"metadata": {

"id": "j8bF22iNxMDx"

}

},

{

"cell\_type": "markdown",

"source": [

"# \*\*CNN MODEL FOR FLOWER CLASSIFICATION\*\*\n",

"# \*\*Trained by Team ID : PNT2022TMID17050\*\*"

],

"metadata": {

"id": "BdPpvl8Q0IlY"

}

},

{

"cell\_type": "markdown",

"source": [

"# \*\*Pre-Requisites\*\*"

],

"metadata": {

"id": "d5mO5lxMq9BQ"

}

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"colab": {

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

},

"id": "h4I4dfJ6p3kx",

"outputId": "5605b9f7-ebc0-4587-88c6-268194f1335d"

},

"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": "markdown",

"source": [

"# \*\*STEP 1 UNZIP FILES\*\*"

],

"metadata": {

"id": "B4Kss5zOrXUz"

}

},

{

"cell\_type": "code",

"source": [

"cd/content/drive/MyDrive/AI\_IBM"

],

"metadata": {

"colab": {

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

},

"id": "DRNhAxfvrWKc",

"outputId": "f1a087a8-6f33-4d81-ff79-e50f4c1ff623"

},

"execution\_count": 2,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"/content/drive/MyDrive/AI\_IBM\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"!unzip Flowers-Dataset.zip"

],

"metadata": {

"colab": {

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

},

"id": "B\_hJ27NKrhKz",

"outputId": "9b319781-61d1-4a0e-ea93-f279d067bfc7"

},

"execution\_count": 3,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Archive: Flowers-Dataset.zip\n",

"replace flowers/daisy/100080576\_f52e8ee070\_n.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: N\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*STEP 2 Image\*\* \*\*Augumentation\*\*"

],

"metadata": {

"id": "hvG8h70rrphq"

}

},

{

"cell\_type": "code",

"source": [

"from tensorflow.keras.preprocessing.image import ImageDataGenerator"

],

"metadata": {

"id": "itQt2Ad8rtk8"

},

"execution\_count": 4,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"train\_datagen=ImageDataGenerator(rescale=1./255, zoom\_range=0.2,horizontal\_flip=True,vertical\_flip=False)"

],

"metadata": {

"id": "9yZUiTxnr0UN"

},

"execution\_count": 5,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"test\_datagen=ImageDataGenerator(rescale=1./255)"

],

"metadata": {

"id": "zD7ristVr3F3"

},

"execution\_count": 6,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"x\_train=train\_datagen.flow\_from\_directory(r\"/content/drive/MyDrive/AI\_IBM/flowers\",target\_size=(64,64),class\_mode='categorical',batch\_size=24)"

],

"metadata": {

"colab": {

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

},

"id": "BjQo5zGHuHN4",

"outputId": "d3d1e296-e74d-4e52-cce8-8d26459d10f1"

},

"execution\_count": 7,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 4317 images belonging to 5 classes.\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"x\_test=test\_datagen.flow\_from\_directory(r\"/content/drive/MyDrive/AI\_IBM/flowers\",target\_size=(64,64),class\_mode='categorical',batch\_size=24)"

],

"metadata": {

"colab": {

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

},

"id": "e4YJwWrCukDq",

"outputId": "e71a3e44-6642-4592-fa96-7af9c6edb08f"

},

"execution\_count": 8,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 4317 images belonging to 5 classes.\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"x\_train.class\_indices"

],

"metadata": {

"colab": {

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

},

"id": "EgBhHHYTuv4X",

"outputId": "8a9f62e0-7d2b-4138-c5ce-4ca16b78fbd1"

},

"execution\_count": 9,

"outputs": [

{

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

"data": {

"text/plain": [

"{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}"

]

},

"metadata": {},

"execution\_count": 9

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*Step -3 Initializing CNN And Create Model\*\*"

],

"metadata": {

"id": "O5cz-9qOJM\_s"

}

},

{

"cell\_type": "code",

"source": [

"from tensorflow.keras.models import Sequential\n",

"from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten"

],

"metadata": {

"id": "QAUHi2otRcoC"

},

"execution\_count": 10,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"# \*\*Step -4 Add layers\*\*"

],

"metadata": {

"id": "xew7skua3a0z"

}

},

{

"cell\_type": "code",

"source": [

"model=Sequential()"

],

"metadata": {

"id": "dack9NXYR2t6"

},

"execution\_count": 11,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"# \*\*4.1 Input Layers (Convolution ,MaxPooling,Flatten)\*\*"

],

"metadata": {

"id": "SzIvL8Q52DFR"

}

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "qPUbKxHGR7EX"

},

"execution\_count": 12,

"outputs": []

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "IBGMZ7sSSAIB"

},

"execution\_count": 13,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.add(Flatten())"

],

"metadata": {

"id": "c65fXm9KSErL"

},

"execution\_count": 14,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.summary()"

],

"metadata": {

"colab": {

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

},

"id": "-go5E-VbSIau",

"outputId": "1f46f35d-1950-4456-bce9-16a06053d40f"

},

"execution\_count": 15,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Model: \"sequential\"\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

" Layer (type) Output Shape Param # \n",

"=================================================================\n",

" conv2d (Conv2D) (None, 62, 62, 32) 896 \n",

" \n",

" max\_pooling2d (MaxPooling2D (None, 31, 31, 32) 0 \n",

" ) \n",

" \n",

" flatten (Flatten) (None, 30752) 0 \n",

" \n",

"=================================================================\n",

"Total params: 896\n",

"Trainable params: 896\n",

"Non-trainable params: 0\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*4.2 Hidden Layers\*\*"

],

"metadata": {

"id": "f4ZSQPIFJaeb"

}

},

{

"cell\_type": "code",

"source": [

"model.add(Dense(300,activation='relu'))\n",

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

],

"metadata": {

"id": "x8MIUG1PSZ21"

},

"execution\_count": 16,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"# \*\*4.3 Output Layers\*\*"

],

"metadata": {

"id": "PNLk8KHHJf3K"

}

},

{

"cell\_type": "code",

"source": [

"model.add(Dense(5,activation='softmax'))"

],

"metadata": {

"id": "grI0IbuwSeqO"

},

"execution\_count": 17,

"outputs": []

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "l44vMW4QShaw"

},

"execution\_count": 18,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"len(x\_train)"

],

"metadata": {

"colab": {

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

},

"id": "Beiar4NESkL4",

"outputId": "4b264b09-51b5-4786-b2a8-d60ac129229d"

},

"execution\_count": 19,

"outputs": [

{

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

"data": {

"text/plain": [

"180"

]

},

"metadata": {},

"execution\_count": 19

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*Step -5 Train the Model\*\*"

],

"metadata": {

"id": "Y9f3ElSv3Nc6"

}

},

{

"cell\_type": "code",

"source": [

"model.fit\_generator(x\_train,steps\_per\_epoch=len(x\_train), validation\_data=x\_test, validation\_steps=len(x\_test), epochs= 30)"

],

"metadata": {

"colab": {

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

},

"id": "ATt0m5Cv6R-w",

"outputId": "734d2b05-c864-450f-a46f-8ce129904306"

},

"execution\_count": 20,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/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",

" \"\"\"Entry point for launching an IPython kernel.\n"

]

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/30\n",

"180/180 [==============================] - 393s 2s/step - loss: 1.3213 - accuracy: 0.4714 - val\_loss: 1.1275 - val\_accuracy: 0.5532\n",

"Epoch 2/30\n",

"180/180 [==============================] - 74s 409ms/step - loss: 1.0600 - accuracy: 0.5854 - val\_loss: 0.9406 - val\_accuracy: 0.6301\n",

"Epoch 3/30\n",

"180/180 [==============================] - 73s 405ms/step - loss: 0.9678 - accuracy: 0.6247 - val\_loss: 0.9603 - val\_accuracy: 0.6203\n",

"Epoch 4/30\n",

"180/180 [==============================] - 77s 429ms/step - loss: 0.8884 - accuracy: 0.6546 - val\_loss: 0.8187 - val\_accuracy: 0.6938\n",

"Epoch 5/30\n",

"180/180 [==============================] - 76s 422ms/step - loss: 0.8358 - accuracy: 0.6787 - val\_loss: 0.7393 - val\_accuracy: 0.7225\n",

"Epoch 6/30\n",

"180/180 [==============================] - 75s 418ms/step - loss: 0.7924 - accuracy: 0.6965 - val\_loss: 0.8389 - val\_accuracy: 0.6928\n",

"Epoch 7/30\n",

"180/180 [==============================] - 73s 405ms/step - loss: 0.7521 - accuracy: 0.7158 - val\_loss: 0.8503 - val\_accuracy: 0.6789\n",

"Epoch 8/30\n",

"180/180 [==============================] - 74s 411ms/step - loss: 0.7048 - accuracy: 0.7313 - val\_loss: 0.6492 - val\_accuracy: 0.7521\n",

"Epoch 9/30\n",

"180/180 [==============================] - 72s 400ms/step - loss: 0.6502 - accuracy: 0.7521 - val\_loss: 0.6458 - val\_accuracy: 0.7438\n",

"Epoch 10/30\n",

"180/180 [==============================] - 74s 409ms/step - loss: 0.6182 - accuracy: 0.7684 - val\_loss: 0.5721 - val\_accuracy: 0.7818\n",

"Epoch 11/30\n",

"180/180 [==============================] - 72s 402ms/step - loss: 0.5662 - accuracy: 0.7931 - val\_loss: 0.5968 - val\_accuracy: 0.7725\n",

"Epoch 12/30\n",

"180/180 [==============================] - 72s 401ms/step - loss: 0.5600 - accuracy: 0.7908 - val\_loss: 0.6907 - val\_accuracy: 0.7612\n",

"Epoch 13/30\n",

"180/180 [==============================] - 72s 399ms/step - loss: 0.5064 - accuracy: 0.8138 - val\_loss: 0.5185 - val\_accuracy: 0.8117\n",

"Epoch 14/30\n",

"180/180 [==============================] - 71s 394ms/step - loss: 0.4830 - accuracy: 0.8249 - val\_loss: 0.3613 - val\_accuracy: 0.8673\n",

"Epoch 15/30\n",

"180/180 [==============================] - 71s 397ms/step - loss: 0.4650 - accuracy: 0.8196 - val\_loss: 0.3396 - val\_accuracy: 0.8768\n",

"Epoch 16/30\n",

"180/180 [==============================] - 71s 393ms/step - loss: 0.4117 - accuracy: 0.8559 - val\_loss: 0.3472 - val\_accuracy: 0.8738\n",

"Epoch 17/30\n",

"180/180 [==============================] - 71s 397ms/step - loss: 0.3892 - accuracy: 0.8631 - val\_loss: 0.3314 - val\_accuracy: 0.8826\n",

"Epoch 18/30\n",

"180/180 [==============================] - 70s 389ms/step - loss: 0.3441 - accuracy: 0.8726 - val\_loss: 0.4008 - val\_accuracy: 0.8589\n",

"Epoch 19/30\n",

"180/180 [==============================] - 73s 404ms/step - loss: 0.3467 - accuracy: 0.8719 - val\_loss: 0.2484 - val\_accuracy: 0.9060\n",

"Epoch 20/30\n",

"180/180 [==============================] - 72s 398ms/step - loss: 0.3327 - accuracy: 0.8758 - val\_loss: 0.2234 - val\_accuracy: 0.9210\n",

"Epoch 21/30\n",

"180/180 [==============================] - 73s 403ms/step - loss: 0.2807 - accuracy: 0.9009 - val\_loss: 0.2830 - val\_accuracy: 0.9036\n",

"Epoch 22/30\n",

"180/180 [==============================] - 70s 392ms/step - loss: 0.2751 - accuracy: 0.9013 - val\_loss: 0.2392 - val\_accuracy: 0.9141\n",

"Epoch 23/30\n",

"180/180 [==============================] - 73s 404ms/step - loss: 0.2549 - accuracy: 0.9097 - val\_loss: 0.2221 - val\_accuracy: 0.9189\n",

"Epoch 24/30\n",

"180/180 [==============================] - 72s 399ms/step - loss: 0.2412 - accuracy: 0.9243 - val\_loss: 0.2029 - val\_accuracy: 0.9291\n",

"Epoch 25/30\n",

"180/180 [==============================] - 72s 402ms/step - loss: 0.2360 - accuracy: 0.9199 - val\_loss: 0.1965 - val\_accuracy: 0.9307\n",

"Epoch 26/30\n",

"180/180 [==============================] - 72s 401ms/step - loss: 0.2199 - accuracy: 0.9201 - val\_loss: 0.1919 - val\_accuracy: 0.9331\n",

"Epoch 27/30\n",

"180/180 [==============================] - 72s 400ms/step - loss: 0.2008 - accuracy: 0.9363 - val\_loss: 0.1218 - val\_accuracy: 0.9560\n",

"Epoch 28/30\n",

"180/180 [==============================] - 73s 406ms/step - loss: 0.1889 - accuracy: 0.9310 - val\_loss: 0.2838 - val\_accuracy: 0.9108\n",

"Epoch 29/30\n",

"180/180 [==============================] - 70s 389ms/step - loss: 0.2046 - accuracy: 0.9275 - val\_loss: 0.2116 - val\_accuracy: 0.9307\n",

"Epoch 30/30\n",

"180/180 [==============================] - 70s 392ms/step - loss: 0.1886 - accuracy: 0.9372 - val\_loss: 0.2091 - val\_accuracy: 0.9280\n"

]

},

{

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

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7f3e15438e50>"

]

},

"metadata": {},

"execution\_count": 20

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*Step -6 Save The model\*\*"

],

"metadata": {

"id": "1uK880jw9Kru"

}

},

{

"cell\_type": "code",

"source": [

"model.save('Flowers\_classification\_model1.h5')"

],

"metadata": {

"id": "scoaKurE9FZk"

},

"execution\_count": 21,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"# \*\*Step -7 Test The model\*\*"

],

"metadata": {

"id": "YAH2UVpi9RMV"

}

},

{

"cell\_type": "code",

"source": [

"ls"

],

"metadata": {

"colab": {

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

},

"id": "Z-co6hBAEmzg",

"outputId": "bf8a661d-3210-4695-dcb7-48e6f365dfce"

},

"execution\_count": 22,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\u001b[0m\u001b[01;34mflowers\u001b[0m/ Flowers\_classification\_model1.h5 Flowers-Dataset.zip video.mp4\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

"from tensorflow.keras.models import load\_model\n",

"from tensorflow.keras.preprocessing import image"

],

"metadata": {

"id": "mJvRRo7VvkeO"

},

"execution\_count": 23,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"# Load the model\n",

"model=load\_model('Flowers\_classification\_model1.h5')"

],

"metadata": {

"id": "xo6F\_4jw9KBZ"

},

"execution\_count": 24,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"img=image.load\_img(r\"/content/s3.jpg\",target\_size=(64,64))\n",

"x=image.img\_to\_array(img)\n",

"x=np.expand\_dims(x,axis=0)\n",

"y=np.argmax(model.predict(x),axis=1)\n",

"# x\_train.class\_indices\n",

"index=['daisy','dandelion','rose','sunflower','tulip']\n",

"index[y[0]]"

],

"metadata": {

"colab": {

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

"height": 35

},

"id": "2rnrfMAf-AB9",

"outputId": "c6357a8b-5163-4884-c82e-05651a65571c"

},

"execution\_count": 38,

"outputs": [

{

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

"data": {

"text/plain": [

"'sunflower'"

],

"application/vnd.google.colaboratory.intrinsic+json": {

"type": "string"

}

},

"metadata": {},

"execution\_count": 38

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*We Achieved 93 percent of accuracy with this model\*\* \n",

"# \*\*Trained by Team ID : PNT2022TMID17050\*\*"

],

"metadata": {

"id": "2f85wU8fL0Si"

}

}

]

}