{

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"id": "cc128f7d",

"metadata": {},

"outputs": [],

"source": [

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

"from tensorflow.keras.layers import Dense\n",

"from tensorflow.keras.layers import Convolution2D\n",

"from tensorflow.keras.layers import MaxPooling2D\n",

"from tensorflow.keras.layers import Flatten"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "9a20251b",

"metadata": {},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "729c5700",

"metadata": {},

"outputs": [],

"source": [

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

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

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"id": "44629271",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 3453 images belonging to 5 classes.\n",

"Found 864 images belonging to 5 classes.\n"

]

}

],

"source": [

"x\_train=train\_data.flow\_from\_directory(r\"E:\\assignment3\\dataset\\Training\",target\_size=(64,64),batch\_size=32,class\_mode=\"categorical\")\n",

"x\_test=test\_data.flow\_from\_directory(r\"E:\\assignment3\\dataset\\Testing\",target\_size=(64,64),batch\_size=32,class\_mode=\"categorical\")"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "f30c5e01",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 38,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train.class\_indices"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"id": "1870d327",

"metadata": {},

"outputs": [],

"source": [

"model=Sequential()"

]

},

{

"cell\_type": "code",

"execution\_count": 41,

"id": "27dcaf14",

"metadata": {},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 42,

"id": "48bc4d2d",

"metadata": {},

"outputs": [],

"source": [

"model.add(MaxPooling2D(pool\_size=(3,3)))"

]

},

{

"cell\_type": "code",

"execution\_count": 43,

"id": "81076674",

"metadata": {},

"outputs": [],

"source": [

"model.add(Flatten())"

]

},

{

"cell\_type": "code",

"execution\_count": 49,

"id": "013fff43",

"metadata": {},

"outputs": [],

"source": [

"model.add(Dense(units=5,kernel\_initializer=\"random\_uniform\",activation=\"softmax\"))"

]

},

{

"cell\_type": "code",

"execution\_count": 50,

"id": "cbafc2f0",

"metadata": {},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 51,

"id": "0d263994",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\Smile\\AppData\\Local\\Temp\\ipykernel\_19908\\1932184935.py:1: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.\n",

" model.fit\_generator(x\_train,steps\_per\_epoch=108,epochs=30,validation\_data=x\_test,validation\_steps=27)\n"

]

},

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Epoch 1/30\n",

"108/108 [==============================] - 88s 798ms/step - loss: 1.6050 - accuracy: 0.2299 - val\_loss: 1.6017 - val\_accuracy: 0.2431\n",

"Epoch 2/30\n",

"108/108 [==============================] - 70s 646ms/step - loss: 1.6005 - accuracy: 0.2438 - val\_loss: 1.5994 - val\_accuracy: 0.2431\n",

"Epoch 3/30\n",

"108/108 [==============================] - 71s 654ms/step - loss: 1.5992 - accuracy: 0.2438 - val\_loss: 1.5989 - val\_accuracy: 0.2431\n",

"Epoch 4/30\n",

"108/108 [==============================] - 70s 648ms/step - loss: 1.5988 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 5/30\n",

"108/108 [==============================] - 74s 685ms/step - loss: 1.5988 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 6/30\n",

"108/108 [==============================] - 77s 712ms/step - loss: 1.5986 - accuracy: 0.2438 - val\_loss: 1.5985 - val\_accuracy: 0.2431\n",

"Epoch 7/30\n",

"108/108 [==============================] - 80s 739ms/step - loss: 1.5985 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 8/30\n",

"108/108 [==============================] - 76s 702ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 9/30\n",

"108/108 [==============================] - 75s 692ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 10/30\n",

"108/108 [==============================] - 78s 721ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 11/30\n",

"108/108 [==============================] - 75s 693ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 12/30\n",

"108/108 [==============================] - 75s 695ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 13/30\n",

"108/108 [==============================] - 81s 745ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 14/30\n",

"108/108 [==============================] - 76s 700ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 15/30\n",

"108/108 [==============================] - 80s 742ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 16/30\n",

"108/108 [==============================] - 75s 693ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 17/30\n",

"108/108 [==============================] - 75s 692ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 18/30\n",

"108/108 [==============================] - 75s 696ms/step - loss: 1.5986 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 19/30\n",

"108/108 [==============================] - 75s 694ms/step - loss: 1.5986 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 20/30\n",

"108/108 [==============================] - 75s 694ms/step - loss: 1.5988 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 21/30\n",

"108/108 [==============================] - 75s 690ms/step - loss: 1.5986 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 22/30\n",

"108/108 [==============================] - 76s 702ms/step - loss: 1.5986 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 23/30\n",

"108/108 [==============================] - 76s 705ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 24/30\n",

"108/108 [==============================] - 75s 692ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 25/30\n",

"108/108 [==============================] - 75s 690ms/step - loss: 1.5988 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 26/30\n",

"108/108 [==============================] - 75s 695ms/step - loss: 1.5986 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 27/30\n",

"108/108 [==============================] - 76s 699ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 28/30\n",

"108/108 [==============================] - 75s 692ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 29/30\n",

"108/108 [==============================] - 75s 690ms/step - loss: 1.5987 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n",

"Epoch 30/30\n",

"108/108 [==============================] - 75s 696ms/step - loss: 1.5988 - accuracy: 0.2438 - val\_loss: 1.5987 - val\_accuracy: 0.2431\n"

]

},

{

"data": {

"text/plain": [

"<keras.callbacks.History at 0x23bcb0e0d60>"

]

},

"execution\_count": 51,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model.fit\_generator(x\_train,steps\_per\_epoch=108,epochs=30,validation\_data=x\_test,validation\_steps=27)"

]

},

{

"cell\_type": "code",

"execution\_count": 52,

"id": "71f43241",

"metadata": {},

"outputs": [],

"source": [

"model.save(\"flower.h5\")"

]

},

{

"cell\_type": "code",

"execution\_count": 53,

"id": "d20cecb3",

"metadata": {},

"outputs": [],

"source": [

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

"from tensorflow.keras.preprocessing import image"

]

},

{

"cell\_type": "code",

"execution\_count": 54,

"id": "8a0d46f3",

"metadata": {},

"outputs": [],

"source": [

"import numpy as np"

]

},

{

"cell\_type": "code",

"execution\_count": 55,

"id": "a0763401",

"metadata": {},

"outputs": [],

"source": [

"model=load\_model(\"flower.h5\")"

]

},

{

"cell\_type": "code",

"execution\_count": 102,

"id": "59a894ed",

"metadata": {},

"outputs": [],

"source": [

"img=image.load\_img(\"dandelion.jpg\",target\_size=(64,64))"

]

},

{

"cell\_type": "code",

"execution\_count": 103,

"id": "a9f9836b",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<PIL.Image.Image image mode=RGB size=64x64 at 0x23BC68938E0>"

]

},

"execution\_count": 103,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"img"

]

},

{

"cell\_type": "code",

"execution\_count": 89,

"id": "e7644c30",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"PIL.Image.Image"

]

},

"execution\_count": 89,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"type(img)"

]

},

{

"cell\_type": "code",

"execution\_count": 104,

"id": "047456eb",

"metadata": {},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 105,

"id": "b75e7bc9",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[[ 4., 4., 4.],\n",

" [ 11., 11., 11.],\n",

" [ 9., 9., 9.],\n",

" ...,\n",

" [ 11., 11., 11.],\n",

" [ 9., 9., 9.],\n",

" [ 10., 10., 10.]],\n",

"\n",

" [[ 12., 12., 12.],\n",

" [ 21., 21., 21.],\n",

" [ 19., 19., 19.],\n",

" ...,\n",

" [ 15., 15., 15.],\n",

" [ 23., 23., 23.],\n",

" [ 2., 2., 2.]],\n",

"\n",

" [[ 23., 23., 23.],\n",

" [ 52., 52., 52.],\n",

" [ 41., 41., 41.],\n",

" ...,\n",

" [ 49., 49., 49.],\n",

" [ 34., 34., 34.],\n",

" [ 42., 42., 42.]],\n",

"\n",

" ...,\n",

"\n",

" [[ 56., 56., 56.],\n",

" [121., 121., 121.],\n",

" [ 44., 44., 44.],\n",

" ...,\n",

" [113., 113., 113.],\n",

" [ 24., 24., 24.],\n",

" [ 12., 12., 12.]],\n",

"\n",

" [[ 54., 54., 54.],\n",

" [ 30., 30., 30.],\n",

" [ 36., 36., 36.],\n",

" ...,\n",

" [102., 102., 102.],\n",

" [ 42., 42., 42.],\n",

" [ 10., 10., 10.]],\n",

"\n",

" [[ 23., 23., 23.],\n",

" [ 20., 20., 20.],\n",

" [ 11., 11., 11.],\n",

" ...,\n",

" [ 77., 77., 77.],\n",

" [ 3., 3., 3.],\n",

" [ 10., 10., 10.]]], dtype=float32)"

]

},

"execution\_count": 105,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x"

]

},

{

"cell\_type": "code",

"execution\_count": 92,

"id": "d0db48cc",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(64, 64, 3)"

]

},

"execution\_count": 92,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 94,

"id": "0188420e",

"metadata": {},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 95,

"id": "b6aa4ed7",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"1/1 [==============================] - 0s 63ms/step\n"

]

}

],

"source": [

"pred\_prob=model.predict(x)"

]

},

{

"cell\_type": "code",

"execution\_count": 96,

"id": "6fd88c2f",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[0.17660806, 0.24450503, 0.18085377, 0.17052399, 0.22750916]],\n",

" dtype=float32)"

]

},

"execution\_count": 96,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pred\_prob"

]

},

{

"cell\_type": "code",

"execution\_count": 99,

"id": "4f4f8252",

"metadata": {},

"outputs": [],

"source": [

"class\_name=[\"Daisy\",\"Dandelion\",\"Rose\",\"Sunflower\",\"Tulip\"]\n",

"pred\_id=pred\_prob.argmax(axis=1)[0]"

]

},

{

"cell\_type": "code",

"execution\_count": 100,

"id": "591a760d",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"1"

]

},

"execution\_count": 100,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pred\_id"

]

},

{

"cell\_type": "code",

"execution\_count": 101,

"id": "48c4eba3",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Predicted animalis Dandelion\n"

]

}

],

"source": [

"print(\"Predicted animalis \",str(class\_name[pred\_id]))"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"id": "8811b770",

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.12"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}