ASSIGNMENT DATE	8 OCTOBER 2022
STUDENT NAME	PRIYADHARSHAN S
STUDENT REGISTER NUMBER	2019504564
MAXIMUM MARKS	2 MARKS

ASSIGNMENT 3 :

IMPORTING THE NECESSARY LIBRARIES

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
import cv2
```

In [ ]:
```
from tensorflow.keras.preprocessing import image
```

In [ ]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [ ]:
```
datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, vertical_flip=False, validat
```

In [ ]:
```
x_train = datagen.flow_from_directory(r'C:\Users\spdpr\Downloads\flowers', target_size=(64,64),batch_size=32, class_mode='catego
```

```
Found 3457 images belonging to 5 classes.
```

In [ ]:
```
x_train.class_indices
```

Out[ ]:
```
{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

In [ ]:
```
classes = x_train.class_indices.keys()
```

In [ ]:
```
x_val = datagen.flow_from_directory(r'C:\Users\spdpr\Downloads\flowers', target_size=(64,64),batch_size=32, class_mode='categori
```

Found 860 images belonging to 5 classes.

PERFORMING CONVOLUTION

In [ ]:
```
from tensorflow.keras.models import Sequential
```

In [ ]:
```
from tensorflow.keras.layers import MaxPooling2D,Dense,Flatten,Convolution2D
```

In [ ]:
```
model = Sequential()
```

In [ ]:
```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
```

In [ ]:
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [ ]:
```
model.add(Convolution2D(64,(3,3),activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [ ]:
```
model.add(Convolution2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [ ]:
```
model.add(Flatten())
```

In [ ]:
```
model.add(Dense(units=300, kernel_initializer='random_uniform', activation='relu'))
```

In [ ]:
```
model.add(Dense(units=200, kernel_initializer='random_uniform', activation='relu'))
```

In [ ]:
```
model.add(Dense(units=5,kernel_initializer='random_uniform',activation='softmax'))
```

In [ ]:
```
model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["accuracy"])
```

In [ ]:
```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_3 (Dense)	(None, 300)	1382700
dense_4 (Dense)	(None, 200)	60200
dense_5 (Dense)	(None, 5)	1005

Total params: 1,537,153
Trainable params: 1,537,153
Non-trainable params: 0

TRAINING AND PREDICTION

In [ ]:
```
model.fit_generator(x_train,steps_per_epoch=40,epochs=25,validation_data=x_val,validation_steps=10)
```

```
c:\users\spdpr\appdata\local\programs\python\python37\lib\site-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.
  """Entry point for launching an IPython kernel.
```

```
Epoch 1/25
40/40 [==============================] - 17s 380ms/step - loss: 1.5273 - accuracy: 0.2778 - val_loss: 1.3131 - val_accuracy: 0.3
938
Epoch 2/25
40/40 [==============================] - 14s 344ms/step - loss: 1.2867 - accuracy: 0.4019 - val_loss: 1.2508 - val_accuracy: 0.3
938
Epoch 3/25
40/40 [==============================] - 14s 349ms/step - loss: 1.2319 - accuracy: 0.4516 - val_loss: 1.1553 - val_accuracy: 0.4
969
Epoch 4/25
40/40 [==============================] - 14s 342ms/step - loss: 1.1691 - accuracy: 0.5004 - val_loss: 1.1993 - val_accuracy: 0.4
688
Epoch 5/25
40/40 [==============================] - 14s 342ms/step - loss: 1.1010 - accuracy: 0.5547 - val_loss: 1.0948 - val_accuracy: 0.5
312
Epoch 6/25
40/40 [==============================] - 14s 339ms/step - loss: 1.0307 - accuracy: 0.5867 - val_loss: 1.0939 - val_accuracy: 0.5
813
Epoch 7/25
40/40 [==============================] - 12s 308ms/step - loss: 0.9989 - accuracy: 0.6031 - val_loss: 1.0756 - val_accuracy: 0.5
188
Epoch 8/25
40/40 [==============================] - 9s 220ms/step - loss: 0.9527 - accuracy: 0.6297 - val_loss: 1.0640 - val_accuracy: 0.57
19
Epoch 9/25
40/40 [==============================] - 9s 218ms/step - loss: 0.9599 - accuracy: 0.6273 - val_loss: 0.9842 - val_accuracy: 0.60
94
Epoch 10/25
40/40 [==============================] - 9s 218ms/step - loss: 0.9703 - accuracy: 0.6133 - val_loss: 1.0102 - val_accuracy: 0.60
94
Epoch 11/25
40/40 [==============================] - 9s 222ms/step - loss: 0.8912 - accuracy: 0.6703 - val_loss: 0.8741 - val_accuracy: 0.67
19
Epoch 12/25
40/40 [==============================] - 9s 221ms/step - loss: 0.8270 - accuracy: 0.6859 - val_loss: 0.9529 - val_accuracy: 0.61
25
Epoch 13/25
40/40 [==============================] - 9s 216ms/step - loss: 0.8351 - accuracy: 0.6717 - val_loss: 1.0848 - val_accuracy: 0.55
31
Epoch 14/25
40/40 [==============================] - 9s 217ms/step - loss: 0.9744 - accuracy: 0.6125 - val_loss: 0.9612 - val_accuracy: 0.58
75
```

```
Epoch 15/25
40/40 [==============================] - 10s 246ms/step - loss: 0.8755 - accuracy: 0.6677 - val_loss: 0.9291 - val_accuracy: 0.6
062
Epoch 16/25
40/40 [==============================] - 9s 229ms/step - loss: 0.8193 - accuracy: 0.6945 - val_loss: 0.9217 - val_accuracy: 0.63
75
Epoch 17/25
40/40 [==============================] - 10s 263ms/step - loss: 0.8017 - accuracy: 0.6984 - val_loss: 0.8865 - val_accuracy: 0.6
594
Epoch 18/25
40/40 [==============================] - 10s 249ms/step - loss: 0.8055 - accuracy: 0.6992 - val_loss: 0.8866 - val_accuracy: 0.6
687
Epoch 19/25
40/40 [==============================] - 10s 243ms/step - loss: 0.7708 - accuracy: 0.7094 - val_loss: 0.8153 - val_accuracy: 0.6
719
Epoch 20/25
40/40 [==============================] - 11s 281ms/step - loss: 0.7270 - accuracy: 0.7234 - val_loss: 0.9195 - val_accuracy: 0.6
219
Epoch 21/25
40/40 [==============================] - 11s 265ms/step - loss: 0.7572 - accuracy: 0.7156 - val_loss: 0.9728 - val_accuracy: 0.6
375
Epoch 22/25
40/40 [==============================] - 12s 290ms/step - loss: 0.7068 - accuracy: 0.7430 - val_loss: 0.9038 - val_accuracy: 0.6
375
Epoch 23/25
40/40 [==============================] - 11s 278ms/step - loss: 0.8054 - accuracy: 0.6982 - val_loss: 0.9059 - val_accuracy: 0.6
719
Epoch 24/25
40/40 [==============================] - 11s 268ms/step - loss: 0.7849 - accuracy: 0.6992 - val_loss: 0.8948 - val_accuracy: 0.6
750
Epoch 25/25
40/40 [==============================] - 10s 256ms/step - loss: 0.7892 - accuracy: 0.6950 - val_loss: 0.8992 - val_accuracy: 0.6
625
```

Out[ ]: <keras.callbacks.History at 0x1f6cd823848>

In [ ]:
```
test_img = image.load_img(r"C:\Users\spdpr\Downloads\red-rose-with-green-leaf_43623-944.jpg", target_size=(64,64,3))
```

In [ ]:
```
test_img
```

Out[ ]:

In [ ]:
```
test_img = image.img_to_array(test_img)
```

In [ ]:
```
test_img = np.expand_dims(test_img,axis=0)
```

In [ ]:
```
test_img.shape
```

Out[ ]:
```
(1, 64, 64, 3)
```

In [ ]:
```
pred = model.predict(test_img)[0]
```

```
1/1 [==============================] - 0s 19ms/step
```

In [ ]:
```
for i in range(len(pred)):
    if pred[i]==1:
        print('The type of flower is: ',classes[i])
```

```
The type of flower is:  rose
```

In [ ]:
```
classes = list(classes)
```

In [ ]:
```
model.save("flower_classifier.h5")
```

In [ ]:
```
from tensorflow import keras
```

In [ ]:
```
model = keras.models.load_model("flower_classifier.h5")
```

In [ ]:
```
model.predict(test_img)
```

```
1/1 [==============================] - 0s 66ms/step
```

Out[ ]:
```
array([[0., 0., 1., 0., 0.]], dtype=float32)
```

In [ ]: