ASSIGNMENT-03

BUILD CNN MODEL FOR CLASSIFICATION OF FLOWERS

Assignment Date	5 October 2022
Student Name	SAHANA J M
Student Roll Number	113219071033
Maximum Marks	2 Marks

QUESTION 1:

Download the Dataset

Dataset is downloaded and uploaded.

Dataset is unzipped using the Python command unzip.

```
[64] !unzip Flowers-Dataset.zip
```

QUESTION 2:

Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
rose_datagen=ImageDataGenerator(rescale=1.255,horizontal_flip=True,vert
ical_flip=True)
x_data= rose_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM A
ssignments/flowers",target_size=(64,64),class_mode="categorical",batch_
size=24)
x_data.class_indices
```

```
[5] from tensorflow.keras.preprocessing.image import ImageDataGenerator

[7] rose_datagen=ImageDataGenerator(rescale=1.255,horizontal_flip=True,vertical_flip=True)

[8] x_data= rose_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM Assignments/flowers",target_size=(64,64)

    Found 4317 images belonging to 5 classes.

    x_data.class_indices

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

QUESTION 3:

Create Model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Fl
atten
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

QUESTION 4:

Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers),Output

```
model.add(Convolution2D(34,(3,3),activation="relu",strides=(1,1),input_
shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

```
[188] from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

```
[189] model = Sequential()
```

```
[190] model.add(Convolution2D(34,(3,3),activation="relu",strides=(1,1),input_shape=(64,64,3)))
```

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

```
[192] model.add(Flatten())
```

```
model.summary()
```

```
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 62, 62, 34)        952

 max_pooling2d_12 (MaxPoolin  (None, 31, 31, 34)       0
 g2D)

 max_pooling2d_13 (MaxPoolin  (None, 15, 15, 34)       0
 g2D)

 flatten_6 (Flatten)         (None, 7650)              0

=================================================================
Total params: 952
Trainable params: 952
Non-trainable params: 0
_________________________________________________________________
```

```
model.add(Dense(400,activation="relu"))
model.add(Dense(400,activation="relu"))
model.add(Dense(400,activation="relu"))
model.add(Dense(5,activation="softmax"))
```

```
[194] model.add(Dense(400,activation="relu"))
      model.add(Dense(400,activation="relu"))
      model.add(Dense(400,activation="relu"))
```

```
[195] model.add(Dense(5,activation="softmax"))
```

QUESTION 5:

Compile the Model

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

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

QUESTION 6:

Fit the model

```
model.fit(x_data, epochs= 5, steps_per_epoch= len(x_data) ,validation_d
ata=0.0,validation_steps=0.0)
```

```
model.fit(x_data, epochs= 5, steps_per_epoch= len(x_data) ,validation_data=0.0,validation_steps=0.0)
```

```
Epoch 1/5
90/90 [==============================] - 29s 308ms/step - loss: 25.5278 - accuracy: 0.3484
Epoch 2/5
90/90 [==============================] - 29s 323ms/step - loss: 1.3084 - accuracy: 0.4966
Epoch 3/5
90/90 [==============================] - 28s 309ms/step - loss: 1.2148 - accuracy: 0.5434
Epoch 4/5
90/90 [==============================] - 27s 301ms/step - loss: 1.1023 - accuracy: 0.5891
Epoch 5/5
90/90 [==============================] - 27s 300ms/step - loss: 1.0337 - accuracy: 0.6254
<keras.callbacks.History at 0x7f9d4d2e81d0>
```

QUESTION 7:

Save the Model

```
model.save('flowers_test.h5')
```

```
[198] model.save('flowers_test.h5')
```

QUESTION 8:

Test the Model

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
[95] import numpy as np
     from tensorflow.keras.models import load_model
     from tensorflow.keras.preprocessing import image
```

#Loading the model

```
model = load_model('flowers_test.h5')
```

#Loading the image

```
flower_image = image.load_img(r"/content/drive/MyDrive/IBM Assignments/
flowers/tulip/10128546863_8de70c610d.jpg",target_size=(64,64))
flower_image
```

```
[202] flower_image = image.load_img(r"/content/drive/MyDrive/IBM Assignments/flowers/tulip/10128546863_8de70c
```

```
flower_image
```

#Converting to array

```
flower_image_array = image.img_to_array(flower_image)
flower_image_array
```

```
[206] flower_image_array = image.img_to_array(flower_image)
```

```
flower_image_array
```

```
array([[[133., 163., 197.],
        [134., 165., 196.],
        [134., 165., 196.],
        ...,
        [177., 195., 215.],
        [177., 195., 215.],
        [181., 198., 216.]],

       [[132., 162., 198.],
        [133., 163., 197.],
```

```
flower_image_array=np.expand_dims(x,axis=0)
prediction=model.predict(flower_image_array)
prediction
```

```
[209] flower_image_array=np.expand_dims(x,axis=0)
```

```
[210] prediction=model.predict(flower_image_array)
```

```
WARNING:tensorflow:5 out of the last 6 calls to <function Model.make_predict_function.<locals>.predict_f
1/1 [==============================] - 0s 69ms/step
```

```
[211] prediction
```

```
array([[0.00907495, 0.01952436, 0.03811397, 0.00240582, 0.9308809 ]],
      dtype=float32)
```

```
x_data.class_indices
```

```
[106] x_data.class_indices
```

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

```
index=['daisy','dandelion','rose','sunflower','tulip']
```

```
image.load_img(r"/content/drive/MyDrive/IBM Assignments/flowers/tulip/10128546863_8de70c610d.jpg")
```

```
flower_image
```

```
[119] index[np.argmax(pred)]
```

```
'tulip'
```

#predicting

```
index[np.argmax(pred)]
```

```
'tulip'
```