Assignment 3

Assignment Date	6 October 2022
Student Name	V. Swathi
Student Roll no.	510919106017
Maximum Marks	2 Marks

1. Download the Dataset:

```
In [ ]: from google.colab import drive
        drive.mount('/content/drive')

Mounted at /content/drive

In [ ]: from google.colab import files
        uploaded = files.upload()

In [ ]: ls

drive/  sample_data/

In [ ]: cd /content/drive/MyDrive/New folder

/content/drive/MyDrive/New folder

In [ ]: ls

Flowers-Dataset.zip

In [ ]: !unzip Flowers-Dataset.zip

Archive:  Flowers-Dataset.zip
  inflating: flowers/daisy/100080576_f52e8ee070_n.jpg
  inflating: flowers/daisy/10140303196_b88d3d6cec.jpg
  inflating: flowers/daisy/10172379554_b296050f82_n.jpg
  inflating: flowers/daisy/10172567486_2748826a8b.jpg
  inflating: flowers/daisy/10172636503_21bededa75_n.jpg
  inflating: flowers/daisy/102841525_bd6628ae3c.jpg
  inflating: flowers/daisy/10300722094_28fa978807_n.jpg
  inflating: flowers/daisy/1031799732_e7f4008c03.jpg
  inflating: flowers/daisy/10391248763_1d16681106_n.jpg
  inflating: flowers/daisy/10437754174_22ec990b77_m.jpg
  inflating: flowers/daisy/10437770546_8bb6f7bdd3_m.jpg
  inflating: flowers/daisy/10437929963_bc13eebe0c.jpg
  inflating: flowers/daisy/10466290366_cc72e33532.jpg
  inflating: flowers/daisy/10466558316_a7198b87e2.jpg
```

2. Image Augmentation

Data Augmentation

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

3. Create Model

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

In [ ]: train_datagen = ImageDataGenerator(rescale=1./255,zoom_range = 0.2,horizontal_flip =True,vertical_flip =False)

In [ ]: test_datagen = ImageDataGenerator(rescale=1./255)

In [ ]: x_train = train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/New folder/flowers/daisy",target_size = (10,10),class_mode ='categorical'
Found 0 images belonging to 0 classes.

In [ ]: x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/New folder/flowers/rose",target_size = (1,5),class_mode ='categorical',batch_size=1
Found 0 images belonging to 0 classes.

In [ ]: x_train.class_indices
Out[ ]: {}
```

4. Add Layers (Convolution, Max Pooling, Flatten, Dense- (Hidden Layers), Output)

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

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

In [ ]: model= Sequential()

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

In [ ]: model.add(MaxPooling2D(pool_size = (2,2)))  #Pooled Matrix

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

In [ ]: model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________

In [ ]: 32*(3*3*3+1)
Out[ ]: 896

In [ ]: model.add(Dense(300, activation = 'relu'))
        model.add(Dense(150,activation ='relu'))

In [ ]: model.add(Dense(4,activation='softmax'))
```

5. Compile The Model

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

6. Fit The Model

Fit the model.

```
[ ] model.fit_generator(xtrain,
                        steps_per_epoch=len(xtrain),
                        epochs=20,
                        validation_data=xtest,
                        validation_steps=len(xtest))
```

```
C:\Users\mm\AppData\Local\Temp\ipykernel_6696\312721451.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which su
  model.fit_generator(xtrain,
Epoch 1/20
44/44 [==============================] - 37s 835ms/step - loss: 1.9038 - accuracy: 0.3836 - val_loss: 1.1672 - val_accuracy: 0.5219
Epoch 2/20
44/44 [==============================] - 34s 779ms/step - loss: 1.0908 - accuracy: 0.5606 - val_loss: 1.0398 - val_accuracy: 0.5965
Epoch 3/20
44/44 [==============================] - 36s 815ms/step - loss: 1.0262 - accuracy: 0.5925 - val_loss: 1.0038 - val_accuracy: 0.6185
Epoch 4/20
44/44 [==============================] - 36s 823ms/step - loss: 0.9335 - accuracy: 0.6410 - val_loss: 0.8923 - val_accuracy: 0.6560
Epoch 5/20
44/44 [==============================] - 36s 809ms/step - loss: 0.8781 - accuracy: 0.6604 - val_loss: 0.8886 - val_accuracy: 0.6646
Epoch 6/20
44/44 [==============================] - 34s 764ms/step - loss: 0.8512 - accuracy: 0.6713 - val_loss: 0.8784 - val_accuracy: 0.6771
Epoch 7/20
44/44 [==============================] - 33s 758ms/step - loss: 0.7922 - accuracy: 0.6931 - val_loss: 0.7586 - val_accuracy: 0.7121
Epoch 8/20
44/44 [==============================] - 35s 811ms/step - loss: 0.7471 - accuracy: 0.7107 - val_loss: 0.6955 - val_accuracy: 0.7262
Epoch 9/20
44/44 [==============================] - 35s 795ms/step - loss: 0.7157 - accuracy: 0.7311 - val_loss: 0.6671 - val_accuracy: 0.7482
Epoch 10/20
44/44 [==============================] - 36s 817ms/step - loss: 0.6867 - accuracy: 0.7336 - val_loss: 0.6537 - val_accuracy: 0.7524
Epoch 11/20
44/44 [==============================] - 37s 851ms/step - loss: 0.6314 - accuracy: 0.7628 - val_loss: 0.6081 - val_accuracy: 0.7751
Epoch 12/20
44/44 [==============================] - 34s 773ms/step - loss: 0.6109 - accuracy: 0.7744 - val_loss: 0.6052 - val_accuracy: 0.7716
Epoch 13/20
44/44 [==============================] - 34s 777ms/step - loss: 0.5710 - accuracy: 0.7853 - val_loss: 0.5747 - val_accuracy: 0.7760
```

7. Save The Model

```
In [ ]: model.save('flowers.h5')
```

```
In [ ]: ls
```

```
flowers/  Flowers-Dataset.zip  flowers.h5
```

8. Test The Model

```
In [ ]: import numpy as np
```

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

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

```
In [ ]: model = load_model('flowers.h5')
```

```
In [ ]: pwd
```

```
Out[ ]: '/content/drive/MyDrive/New folder'
```

```
In [ ]: img = image.load_img(r'/content/drive/MyDrive/New folder/flowers/daisy/43474673_7bb4465a86.jpg')
```

```
In [ ]: img
```

```
Out[ ]:
```

In [ ]:
```
img = image.load_img(r'/content/drive/MyDrive/New folder/flowers/daisy/43474673_7bb4465a86.jpg',target_size=(64,64))
```

In [ ]:
```
img
```

Out[ ]:


In [ ]:
```
x=image.img_to_array(img)
```

In [ ]:
```
x
```

Out[ ]:
```
array([[[ 74.,  83.,  66.],
        [ 51.,  68.,  49.],
        [ 40.,  57.,  38.],
        ...,
        [164., 181., 201.],
        [ 64.,  75.,  61.],
        [ 51.,  57.,  21.]],

       [[ 66.,  83.,  64.],
        [ 50.,  73.,  47.],
        [ 43.,  61.,  37.],
        ...,
        [ 84.,  83.,  65.],
        [ 70.,  78.,  21.],
        [ 55.,  62.,  18.]],

       [[ 66.,  85.,  53.],
        [ 56.,  82.,  47.],
        [ 49.,  74.,  44.],
        ...,
        [ 91.,  87.,  13.],
        [ 77.,  83.,  11.],
        [ 76.,  80.,  19.]],

       ...,



        ...,
        [ 24.,  40.,  29.],
        [ 20.,  36.,  25.],
        [ 18.,  34.,  24.]]], dtype=float32)
```

In [ ]:
```
x.shape
```

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

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

In [ ]:
```
y = np.argmax(model.predict(x), axis=1)
```

In [ ]:
```
y
```

Out[ ]:
```
array([1])
```

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

Out[ ]:
```
{}
```

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

In [ ]:
```
index[y[0]]
```

Out[ ]:
```
'dandelion'
```

In [ ]:
```
index[3]
```

Out[ ]:
```
'sunflower'
```