Assignment 3

Build CNN Model for Classification of Flowers

1)Download the Dataset and Unzip the file

```
!unzip "/content/drive/MyDrive/Colab Notebooks/Flowers-Dataset.zip"
```

```
  inflating: flowers/tulip/8695372372_302135aeb2.jpg
  inflating: flowers/tulip/8697784345_e75913d220.jpg
  inflating: flowers/tulip/8702982836_75222725d7.jpg
  inflating: flowers/tulip/8706523526_a0f161b72b.jpg
  inflating: flowers/tulip/8708209606_d3aede4801.jpg
  inflating: flowers/tulip/8708856019_f3be2353a4_n.jpg
  inflating: flowers/tulip/8710148289_6fc196a0f8_n.jpg
  inflating: flowers/tulip/8711277462_b43df5454b_m.jpg
  inflating: flowers/tulip/8712230357_1298b8513b.jpg
  inflating: flowers/tulip/8712243901_54d686319e_m.jpg
  inflating: flowers/tulip/8712244311_da8e90bf8e_n.jpg
  inflating: flowers/tulip/8712260079_c0ff42e0e2_n.jpg
  inflating: flowers/tulip/8712263493_3db76c5f82.jpg
  inflating: flowers/tulip/8712266605_3787e346cd_n.jpg
  inflating: flowers/tulip/8712267391_c756f18ee7_n.jpg
  inflating: flowers/tulip/8712267813_f7a9be2ec5.jpg
  inflating: flowers/tulip/8712268519_f4c2c39a06_n.jpg
  inflating: flowers/tulip/8712269349_2b933da2b8_n.jpg
  inflating: flowers/tulip/8712270243_8512cf4fbd.jpg
  inflating: flowers/tulip/8712270665_57b5bda0a2_n.jpg
  inflating: flowers/tulip/8712282563_3819afb7bc.jpg
  inflating: flowers/tulip/8713357842_9964a93473_n.jpg
  inflating: flowers/tulip/8713387500_6a9138b41b_n.jpg
  inflating: flowers/tulip/8713388322_e5ae26263b_n.jpg
  inflating: flowers/tulip/8713389178_66bceb71a8_n.jpg
  inflating: flowers/tulip/8713390684_041148dd3e_n.jpg
  inflating: flowers/tulip/8713391394_4b679ea1e3_n.jpg
  inflating: flowers/tulip/8713392604_90631fb809_n.jpg
  inflating: flowers/tulip/8713394070_b24561b0a9.jpg
  inflating: flowers/tulip/8713396140_5af8136136.jpg
  inflating: flowers/tulip/8713397358_0505cc0176_n.jpg
  inflating: flowers/tulip/8713397694_bcbcbba2c2_n.jpg
  inflating: flowers/tulip/8713398114_bc96f1b624_n.jpg
  inflating: flowers/tulip/8713398614_88202e452e_n.jpg
  inflating: flowers/tulip/8713398906_28e59a225a_n.jpg
  inflating: flowers/tulip/8713407768_f880df361f.jpg
  inflating: flowers/tulip/8717900362_2aa508e9e5.jpg
  inflating: flowers/tulip/8722514702_7ecc68691c.jpg
  inflating: flowers/tulip/8723767533_9145dec4bd_n.jpg
  inflating: flowers/tulip/8729501081_b993185542_m.jpg
  inflating: flowers/tulip/8733586143_3139db6e9e_n.jpg
  inflating: flowers/tulip/8748266132_5298a91dcf_n.jpg
  inflating: flowers/tulip/8750288831_5e49a9f29b.jpg
  inflating: flowers/tulip/8757486380_90952c5377.jpg
  inflating: flowers/tulip/8758464923_75a5ffe320_n.jpg
  inflating: flowers/tulip/8758519201_16e8d2d781_n.jpg
```

```
inflating: flowers/tulip/8759597778_7fca5d434b_n.jpg
inflating: flowers/tulip/8759601388_36e2a50d98_n.jpg
inflating: flowers/tulip/8759606166_8e475013fa_n.jpg
inflating: flowers/tulip/8759618746_f5e39fdbf8_n.jpg
inflating: flowers/tulip/8762189906_8223cef62f.jpg
inflating: flowers/tulip/8762193202_0fbf2f6a81.jpg
inflating: flowers/tulip/8768645961_8f1e097170_n.jpg
inflating: flowers/tulip/8817622133_a42bb90e38_n.jpg
inflating: flowers/tulip/8838347159_746d14e6c1_m.jpg
inflating: flowers/tulip/8838354855_c474fc66a3_m.jpg
inflating: flowers/tulip/8838914676_8ef4db7f50_n.jpg
```

2)Image Augmentation

```
# Import required lib

from tensorflow.keras.preprocessing.image import ImageDataGenerator


# Creating augmentation on training variable
train_datagen = ImageDataGenerator(rescale=1./255 , zoom_range = 0.2 , horizontal_flip=Tru


test_datagen = ImageDataGenerator(rescale=1./255)


pip install split-folders
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
```

```
import splitfolders
input_folder = "/content/flowers"


splitfolders.ratio(input_folder,output='/content/flowers',
                   ratio=(.8,0,.2),
                   group_prefix=None)
```

```
Copying files: 4317 files [00:01, 3687.48 files/s]
```

```
x_train=train_datagen.flow_from_directory("/content/flowers/test",
                                          target_size=(64,64),
                                          class_mode='categorical',
                                          batch_size=19)
```

```
Found 865 images belonging to 5 classes.
```

```
x_test=test_datagen.flow_from_directory("/content/flowers/train",
                                        target_size=(64,64),
```

```
                                        class_mode='categorical',
                                        batch_size=19)
```

```
Found 3452 images belonging to 5 classes.
```

```
x_train.class_indices
```

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

3)Create Model

```
# Importing required lib

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


model=Sequential()
```

4)Add Layers (Convolution,MaxPooling,Flatten,Dense-(HiddenLayers),Output)

```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3))) # Convolution l
model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling layer
model.add(Flatten()) # Flatten layer


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
_________________________________________________________________
```

```
model.add(Dense(300,activation='relu')) # Hidden layer 1
model.add(Dense(150,activation='relu')) # Hidden layer 2
model.add(Dense(4,activation='softmax')) # Output layer
```

5)Compile The Model

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

```
len(x_train)
len(x_test)
```

```
182
```

```
1238/24
```

```
51.583333333333336
```

```
326/24
```

```
13.583333333333334
```

6)Fit The Model

```
model.fit_generator(x_train,steps_per_epoch=len(x_train),
                    validation_data=x_test,
                    validation_steps=len(x_test),
                    epochs=20)
```

7)Save The Model

```
model.save('Flowers.h7')
```

8)Test The Model

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

```
model.save('flowers.h7')
```

```
img1 = image.load_img('/content/flowers/dandelion/1241011700_261ae180ca.jpg') # Reading In
img1 # Visualize the image
```

```
x=image.img_to_array(img1)
x  # Converting image to array
```

```
array([[[ 21.,  46.,  17.],
        [ 19.,  44.,  15.],
        [ 18.,  43.,  14.],
        ...,
        [ 99., 122.,  70.],
        [ 97., 120.,  68.],
        [ 97., 120.,  68.]],

       [[ 20.,  45.,  16.],
        [ 19.,  44.,  15.],
        [ 18.,  43.,  14.],
        ...,
        [100., 123.,  71.],
        [100., 123.,  71.],
        [ 95., 118.,  66.]],

       [[ 20.,  45.,  16.],
        [ 20.,  45.,  16.],
        [ 19.,  44.,  15.],
        ...,
        [ 99., 122.,  70.],
        [ 99., 122.,  70.],
        [ 97., 120.,  66.]],

       ...,

       [[ 46.,  95.,  29.],
        [ 44.,  93.,  27.],
        [ 46.,  95.,  29.],
        ...,
        [ 51., 102.,  27.],
        [ 54., 105.,  30.],
        [ 55., 106.,  31.]],

       [[ 45.,  94.,  28.],
        [ 45.,  94.,  28.],
        [ 46.,  95.,  29.],
        ...,
        [ 52., 103.,  28.],
        [ 52., 103.,  26.],
        [ 56., 107.,  30.]],

       [[ 44.,  93.,  27.],
        [ 46.,  95.,  29.],
```

```
                [ 45.,  94.,  28.],
                ...,
                [ 50., 101.,  26.],
                [ 52., 103.,  28.],
                [ 54., 105.,  28.]]], dtype=float32)
```

```
x = np.expand_dims(x,axis=0)
x # Expanding dimensions
```

```
     array([[[[ 21.,  46.,  17.],
              [ 19.,  44.,  15.],
              [ 18.,  43.,  14.],
              ...,
              [ 99., 122.,  70.],
              [ 97., 120.,  68.],
              [ 97., 120.,  68.]],

             [[ 20.,  45.,  16.],
              [ 19.,  44.,  15.],
              [ 18.,  43.,  14.],
              ...,
              [100., 123.,  71.],
              [100., 123.,  71.],
              [ 95., 118.,  66.]],

             [[ 20.,  45.,  16.],
              [ 20.,  45.,  16.],
              [ 19.,  44.,  15.],
              ...,
              [ 99., 122.,  70.],
              [ 99., 122.,  70.],
              [ 97., 120.,  66.]],

             ...,

             [[ 46.,  95.,  29.],
              [ 44.,  93.,  27.],
              [ 46.,  95.,  29.],
              ...,
              [ 51., 102.,  27.],
              [ 54., 105.,  30.],
              [ 55., 106.,  31.]],

             [[ 45.,  94.,  28.],
              [ 45.,  94.,  28.],
              [ 46.,  95.,  29.],
              ...,
              [ 52., 103.,  28.],
              [ 52., 103.,  26.],
              [ 56., 107.,  30.]],

             [[ 44.,  93.,  27.],
              [ 46.,  95.,  29.],
              [ 45.,  94.,  28.],
              ...,
              [ 50., 101.,  26.],
              [ 52., 103.,  28.],
              [ 54., 105.,  28.]]]], dtype=float32)
```

```
img=image.load_img("/content/flowers/sunflower/10386525695_2c38fea555_n.jpg",target_size=(
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
x_train.class_indices
index=['daisy','dandellion','rose','sunflower','tulip']
index[y[0]]
```

```
'sunflower'
```