## Assignment 3

## Build CNN Model for Classification of Flowers

1)Download the Dataset and Unzip the file

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

```
  inflating: flowers/tulip/8712270243_8512c14fbd.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/8759594528_2534c0ec65_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
  inflating: flowers/tulip/8838975946_f54194894e_m.jpg
  inflating: flowers/tulip/8838983024_5c1a767878_n.jpg
  inflating: flowers/tulip/8892851067_79242a7362_n.jpg
  inflating: flowers/tulip/8904780994_8867d64155_n.jpg
  inflating: flowers/tulip/8908062479_449200a1b4.jpg
  inflating: flowers/tulip/8908097235_c3e746d36e_n.jpg
  inflating: flowers/tulip/9019694597_2d3bbedb17.jpg
  inflating: flowers/tulip/9020467406_05e93ff171_n.jpg
```

```
inflating: flowers/tulip/9030467406_05e93ff171_n.jpg
inflating: flowers/tulip/9048307967_40a164a459_m.jpg
inflating: flowers/tulip/924782410_94ed7913ca_m.jpg
inflating: flowers/tulip/9378657435_89fabf13c9_n.jpg
inflating: flowers/tulip/9444202147_405290415b_n.jpg
inflating: flowers/tulip/9446982168_06c4d71da3_n.jpg
inflating: flowers/tulip/9831362123_5aac525a99_n.jpg
inflating: flowers/tulip/9870557734_88eb3b9e3b_n.jpg
inflating: flowers/tulip/9947374414_fdf1d0861c_n.jpg
inflating: flowers/tulip/9947385346_3a8cacea02_n.jpg
inflating: flowers/tulip/9976515506_d496c5e72c.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 [00:00, 4827.78 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.h6')
```

### 8)Test The Model

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

```
model.save('flowers.h6')
```

```
img1 = image.load_img('/content/flowers/daisy/10555815624_dc211569b0.jpg') # Reading Image
img1 # Visualize the image
```

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

```
array([[[ 11.,  16.,   9.],
        [ 11.,  16.,   9.],
        [ 11.,  18.,  10.],
        ...,
        [ 55.,  40.,  17.],
        [ 55.,  40.,  17.],
        [ 55.,  40.,  17.]],

       [[ 12.,  17.,  10.],
        [ 11.,  16.,   9.],
        [ 12.,  17.,  10.],
        ...,
        [ 57.,  43.,  16.],
        [ 59.,  43.,  18.],
        [ 59.,  45.,  19.]],

       [[ 13.,  18.,  11.],
        [ 12.,  17.,  10.],
        [ 12.,  17.,  10.],
        ...,
        [ 60.,  47.,  15.],
        [ 61.,  46.,  15.],
        [ 62.,  47.,  18.]],

       ...,

       [[209., 196., 213.],
        [210., 197., 214.],
        [212., 199., 216.],
        ...,
        [ 22.,  34.,  14.],
        [ 18.,  31.,  11.],
        [ 16.,  29.,  11.]],

       [[204., 194., 205.],
        [205., 195., 206.],
        [206., 196., 207.],
        ...,
        [ 20.,  32.,  12.],
        [ 19.,  30.,  13.],
        [ 18.,  29.,  12.]],

       [[196., 189., 196.],
        [197., 190., 197.],
        [199., 192., 199.],
        ...,
```

```
               [ 20.,  32.,  12.],
               [ 18.,  30.,  10.],
               [ 19.,  28.,  11.]]], dtype=float32)
```

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

```
array([[[[ 11.,  16.,   9.],
         [ 11.,  16.,   9.],
         [ 11.,  18.,  10.],
         ...,
         [ 55.,  40.,  17.],
         [ 55.,  40.,  17.],
         [ 55.,  40.,  17.]],

        [[ 12.,  17.,  10.],
         [ 11.,  16.,   9.],
         [ 12.,  17.,  10.],
         ...,
         [ 57.,  43.,  16.],
         [ 59.,  43.,  18.],
         [ 59.,  45.,  19.]],

        [[ 13.,  18.,  11.],
         [ 12.,  17.,  10.],
         [ 12.,  17.,  10.],
         ...,
         [ 60.,  47.,  15.],
         [ 61.,  46.,  15.],
         [ 62.,  47.,  18.]],

        ...,

        [[209., 196., 213.],
         [210., 197., 214.],
         [212., 199., 216.],
         ...,
         [ 22.,  34.,  14.],
         [ 18.,  31.,  11.],
         [ 16.,  29.,  11.]],

        [[204., 194., 205.],
         [205., 195., 206.],
         [206., 196., 207.],
         ...,
         [ 20.,  32.,  12.],
         [ 19.,  30.,  13.],
         [ 18.,  29.,  12.]],

        [[196., 189., 196.],
         [197., 190., 197.],
         [199., 192., 199.],
         ...,
         [ 20.,  32.,  12.],
         [ 18.,  30.,  10.],
         [ 19.,  28.,  11.]]]], dtype=float32)
```

```
img=image.load_img("/content/flowers/daisy/10555815624_dc211569b0.jpg",target_size=(64,64)
```

```
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]]
```

```
'daisy'
```

Colab paid products - Cancel contracts here

✓ 0s completed at 8:21 PM