Assignment 3

Build CNN Model for Classification of Flowers

DATE : 10 OCTOBER 2022

Team Id : PNT2022TMID38674

PROJECT NAME AI based discourse for Banking Industry

Submitted : Thiyagarajan V

1.Download the Dataset

```
ls
```

```
Flowers-Dataset.zip
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```

```
cd /content/drive/MyDrive/
```

```
/content/drive/MyDrive
```

```
#!unzip Flowers-Dataset.zip
```

```
cd flowers/
```

```
/content/drive/MyDrive/dataset/flowers
```

2.Image Augmentation

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

```
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_
```

```
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/dataset/flowers/',target_s
```

```
Found 4317 images belonging to 5 classes.
```

```
x_test=train_datagen.flow_from_directory(r'/content/drive/MyDrive/dataset/flowers/',target_si
```

```
Found 4317 images belonging to 5 classes.
```

```
x_train.class_indices
```

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

```
x_test.class_indices
```

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

3. Create Model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model=Sequential()
```

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

```
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten

model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))

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

model.add(Flatten())

model.summary() #32*(3*3*3+1)
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_1 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 30752)             0

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

```
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))


model.add(Dense(4,activation='softmax'))
```

5. Compile The Model

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

6.Fit The Model

```
model.fit(x_train, epochs = 5, validation_data=x_test, steps_per_epoch=len(x_train), validati
```

Epoch 1/5 180/180 [==============================] - 711s 4s/step - loss: 1.6647 - accuracy: 0.2201 - val_loss: 1.6395 - val_accuracy: 0.2437

Epoch 2/5 180/180 [==============================] - 65s 362ms/step - loss: 1.6257 - accuracy: 0.2409 - val_loss: 1.6142 - val_accuracy: 0.2437

Epoch 3/5 180/180 [==============================] - 66s 366ms/step - loss: 1.6083 - accuracy: 0.2437 - val_loss: 1.6034 - val_accuracy: 0.2437

Epoch 4/5 180/180 [==============================] - 65s 361ms/step - loss: 1.6015 - accuracy: 0.2437 - val_loss: 1.5998 - val_accuracy: 0.2437

Epoch 5/5 180/180 [==============================] - 65s 360ms/step - loss: 1.5994 - accuracy: 0.2432 - val_loss: 1.5987 - val_accuracy: 0.2437

7.Save The Mode

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

```
ls #Flowers.h5 is trained model save
```

```
daisy/  dandelion/  Flowers.h5  rose/  sunflower/  tulip/
```

8.Test The Model

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

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

```
img=image.load_img(r"/content/drive/MyDrive/flowers/daisy/100080576_f52e8ee070_n.jpg")
```

```
img
```

```
img=image.load_img(r"/content/drive/MyDrive/flowers/daisy/100080576_f52e8ee070_n.jpg", target
img
```

```
x=image.img_to_array(img)
x
```

```
array([[[141., 141., 139.],
        [149., 149., 149.],
        [152., 152., 154.],
        ...,
        [162., 161., 166.],
        [154., 154., 152.],
        [153., 153., 153.]],

       [[136., 135., 131.],
        [146., 145., 143.],
        [169., 168., 174.],
        ...,
        [159., 158., 163.],
        [155., 155., 153.],
        [149., 149., 149.]],

       [[125., 125., 117.],
        [138., 140., 137.],
        [152., 152., 152.],
        ...,
        [156., 156., 156.],
        [157., 157., 155.],
        [143., 142., 140.]],

       ...,

       [[ 41.,  44.,  23.],
        [ 43.,  46.,  25.],
        [ 49.,  51.,  37.],
        ...,
        [128., 124., 121.],
        [125., 121., 118.],
        [125., 122., 117.]],

       [[ 43.,  46.,  25.],
        [ 43.,  46.,  25.],
        [ 54.,  55.,  37.],
        ...,
        [130., 126., 125.],
        [129., 125., 124.],
        [127., 123., 122.]],

       [[ 44.,  47.,  26.],
        [ 45.,  48.,  27.],
        [ 53.,  55.,  34.],
        ...,
        [137., 133., 132.],
```

```
               [133., 129., 128.],
               [130., 126., 125.]]], dtype=float32)
```

```
x=np.expand_dims(x,axis=0)
```

```
y=np.argmax(model.predict(x),axis=0)
y
```

```
     array([0, 0, 0, 0])
```

```
x_train.class_indices
```

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

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

```
index[y[0]]
```

```
     'daisy'
```

```
img=image.load_img(r"/content/drive/MyDrive/flowers/dandelion/10200780773_c6051a7d71_n.jpg",
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['daisy','dandelion','rose','sunflower']
index[y[0]]
```

```
     'daisy'
```

Colab paid products - Cancel contracts here

✓ 0s completed at 11:00 AM