Assignment -3
Build CNN Model for Classification Of Flowers

Assignment submission	10 October 2022
Student Name	Srivarthini J
Student Roll Number	951919CS100
Maximum Marks	2 Marks

1. Download the dataset: Dataset

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

```
Mounted at /content/drive
```

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

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

```
>!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
```

2. Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_fli
p=True,vertical_flip=False)

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

3. Create Model

```
>X_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Flowers-
Dataset/flowers',target_size=(64,64),class_mode='categorical',batch_size=24)
```

```
Found 30 images belonging to 5 classes.
```

```
>X_test=train_datagen.flow_from_directory('/content/drive/MyDrive/Flowers-
Dataset/flowers',target_size=(64,64),class_mode='categorical',batch_size=24)
```

```
Found 40 images belonging to 5 classes.
```

```
>X_train.class_indices

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

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

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
model=Sequential()
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()
Model: "sequential_1"
_________________________________________________________________
 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'))
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_generator(X_train,steps_per_epoch=len(X_train),validation_data=X_te
st,validation_steps=len(X_test),epochs=10)
```

7. Save The Model

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

8. Test The Model

```
import numpy as np
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

model=load_model('/content/drive/MyDrive/flowersss')

img=image.load_img("/content/drive/MyDrive/flowers/daisy/153210866_03cc9f2f36
.jpg" )
img
```

```
>img=image.load_img("/content/drive/MyDrive/flowers/daisy/153210866_03cc9f2f3
6.jpg",target_size=(64,64) )
img
```

```
>X=image.img_to_array(img)
>X
array([[[13., 20., 13.], [14., 23., 18.], [20., 27., 20.], ..., [50., 41.,
32.], [46., 37., 28.], [17., 19., 14.]], [[18., 20., 15.], [25., 31., 29.],
[29., 31., 28.], ..., [46., 48., 34.], [50., 41., 32.], [ 3., 5., 4.]],
[[14., 20., 16.], [17., 22., 16.], [18., 20., 17.], ..., [52., 50., 38.],
[50., 47., 38.], [21., 23., 20.]], ..., [[21., 26., 20.], [40., 40., 32.],
[34., 35., 30.], ..., [21., 28., 21.], [11., 15., 14.], [22., 21., 17.]],
[[26., 31., 27.], [53., 53., 43.], [32., 37., 31.], ..., [28., 34., 24.],
[21., 31., 22.], [50., 50., 38.]], [[34., 36., 31.], [43., 46., 35.], [24.,
```

```
26.,  21.],  ...,  [71.,  65.,  49.],  [69.,  63.,  47.],  [83.,  76.,  60.]]],
dtype=float32)

>y=np.argmax(model.predict(X),axis=1)
>y

array([0])

>X_train.class_indices

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

>index=['daisy', 'dandelion','rose', 'sunflower','tulip']
>index[y[0]]

'daisy'
```