Assignment -3

Problem Statement:- Build CNN Model for Classification Of Flowers

Assignment Date	19 September 2022
Student Name	KHAMILA BANU K
Student Roll Number	2019504539
Maximum Marks	2 Marks

1. Download the Dataset

Importing all necesaary libraries

Solution:

```
!pip install split-folders

import splitfolders

import numpy as np

import tensorflow as tf
```

Load dataset

Solution:

```
import zipfile

from google.colab import drive
drive.mount('/content/drive')
!unzip /content/drive/My\ Drive/Flowers-Dataset.zip
```

Output:

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
Archive:  /content/drive/My Drive/Flowers-Dataset.zip
replace flowers/daisy/100080576_f52e8ee070_n.jpg? [y]es, [n]o, [A]ll, [N]one,
[r]ename:
```

Split dataset into training data, validation data, testing data

Solution:

```
splitfolders.ratio("/content/flowers", output="/content/flowers", seed=1337,
ratio=(.8, .1, .1), group_prefix=None)
```

Output:

```
Copying files: 4317 files [00:01, 3694.23 files/s]
```

2. Image Augmentation

Solution:

```
import keras
```

```
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)

gentrain=train_datagen.flow_from_directory("/content/flowers/train",target_si
ze=(64,64),class_mode="categorical",batch_size=100)

gentest=test_datagen.flow_from_directory("/content/flowers/val",target_size=(
64,64),class_mode="categorical",batch_size=100)

genval=test_datagen.flow_from_directory("/content/flowers/test",target_size=(
64,64),class_mode="categorical",batch_size=100)
```

3. Create Model

Solution:

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
model=Sequential()
```

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

Solution:

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

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

model.add(Flatten())

model.add(Dense(300,kernel_initializer="random_uniform",activation="relu"))

model.add(Dense(5,kernel_initializer="random_uniform",activation="softmax"))
```

5. Compile The Model

Solution:

```
model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["accu
racy"])
```

6. Fit The Model

Solution:

```
model.fit_generator(gentrain,steps_per_epoch=15,epochs=30,validation_data=gen
test,validation_steps=10)
```

Output:

```
Epoch 1/30
```

```
15/15 [==============================] - ETA: 0s - loss: 1.0158 - accuracy: 0
.6027
15/15 [==============================] - 13s 867ms/step - loss: 1.0158 - accu
racy: 0.6027 - val_loss: 1.0514 - val_accuracy: 0.6093
Epoch 2/30
15/15 [==============================] - 10s 665ms/step - loss: 0.9545 - accu
racy: 0.6220
Epoch 3/30
15/15 [==============================] - 10s 656ms/step - loss: 0.9348 - accu
racy: 0.6347
Epoch 4/30
15/15 [==============================] - 10s 675ms/step - loss: 0.9333 - accu
racy: 0.6253
Epoch 5/30
15/15 [==============================] - 11s 721ms/step - loss: 0.9238 - accu
racy: 0.6460
Epoch 6/30
15/15 [==============================] - 10s 644ms/step - loss: 0.8809 - accu
racy: 0.6563
Epoch 7/30
15/15 [==============================] - 10s 666ms/step - loss: 0.8613 - accu
racy: 0.6715
Epoch 8/30
15/15 [==============================] - 12s 739ms/step - loss: 0.8220 - accu
racy: 0.6707
Epoch 9/30
15/15 [==============================] - 10s 657ms/step - loss: 0.8181 - accu
racy: 0.6933
Epoch 10/30
15/15 [==============================] - 10s 664ms/step - loss: 0.8075 - accu
racy: 0.6860
Epoch 11/30
15/15 [==============================] - 10s 628ms/step - loss: 0.7976 - accu
racy: 0.6935
Epoch 12/30
15/15 [==============================] - 10s 640ms/step - loss: 0.7626 - accu
racy: 0.7073
Epoch 13/30
15/15 [==============================] - 10s 641ms/step - loss: 0.7525 - accu
racy: 0.7039
Epoch 14/30
15/15 [==============================] - 10s 650ms/step - loss: 0.7359 - accu
racy: 0.7153
Epoch 15/30
15/15 [==============================] - 10s 659ms/step - loss: 0.7489 - accu
racy: 0.7127
Epoch 16/30
15/15 [==============================] - 10s 633ms/step - loss: 0.7419 - accu
racy: 0.7190
Epoch 17/30
15/15 [==============================] - 10s 648ms/step - loss: 0.7459 - accu
racy: 0.7140
Epoch 18/30
15/15 [==============================] - 10s 631ms/step - loss: 0.7001 - accu
racy: 0.7410
Epoch 19/30
15/15 [==============================] - 10s 632ms/step - loss: 0.6900 - accu
racy: 0.7376
Epoch 20/30
15/15 [==============================] - 10s 659ms/step - loss: 0.6840 - accu
racy: 0.7487
Epoch 21/30
```

```
15/15 [==============================] - 10s 668ms/step - loss: 0.6905 - accu
racy: 0.7367
Epoch 22/30
15/15 [==============================] - 10s 635ms/step - loss: 0.6613 - accu
racy: 0.7493
Epoch 23/30
15/15 [==============================] - 10s 659ms/step - loss: 0.6479 - accu
racy: 0.7453
Epoch 24/30
15/15 [==============================] - 10s 621ms/step - loss: 0.6366 - accu
racy: 0.7672
Epoch 25/30
15/15 [==============================] - 10s 644ms/step - loss: 0.6610 - accu
racy: 0.7541
Epoch 26/30
15/15 [==============================] - 10s 656ms/step - loss: 0.6416 - accu
racy: 0.7627
Epoch 27/30
15/15 [==============================] - 12s 772ms/step - loss: 0.5998 - accu
racy: 0.7803
Epoch 28/30
15/15 [==============================] - 10s 658ms/step - loss: 0.5547 - accu
racy: 0.7887
Epoch 29/30
15/15 [==============================] - 10s 679ms/step - loss: 0.5501 - accu
racy: 0.7840
Epoch 30/30
15/15 [==============================] - 10s 672ms/step - loss: 0.5859 - accu
racy: 0.7833
```

7. Save The Model

Solution:

```
model.save("./flower.h5")
```

8. Test The Model

Solution:

```
model.evaluate(genval)
```

Output:

```
5/5 [==============================] - 2s 279ms/step - loss: 0.9311 - accurac
y: 0.6759
[0.9310972094535828, 0.6758620738983154]
```