Assignment -3

BUILD AN CNN MODEL FOR THE CLASSIFICATION OF FLOWERS

Assignment Date	10 October 2022
Student Name	Mr.E.R.GOWSHIGAN
Student Roll Number	913319104301
Maximum Marks	2 Marks

Question-1:

DOWNLOAD THE DATA SET

!unzip '/content/Flowers-Dataset.zip'

```
!unzip '/content/Flowers-Dataset.zip'
  inflating: flowers/dandelion/5996421299_b9bf488c1a_n.jpg
  inflating: flowers/dandelion/6012046444_fd80afb63a_n.jpg
  inflating: flowers/dandelion/6019234426_d25ea1230a_m.jpg
  inflating: flowers/dandelion/6035460327_4bbb708eab_n.jpg
  inflating: flowers/dandelion/6044710875_0459796d1b_m.jpg
  inflating: flowers/dandelion/6060576850_984176cf4f_n.jpg
  inflating: flowers/dandelion/6103898045_e066cdeedf_n.jpg
  inflating: flowers/dandelion/6104442744_ee2bcd32e7_n.jpg
  inflating: flowers/dandelion/61242541_a04395e6bc.jpg
  inflating: flowers/dandelion/6132275522_ce46b33c33_m.jpg
  inflating: flowers/dandelion/6146107825_45f708ecd7_n.jpg
  inflating: flowers/dandelion/6208857436_14a65fe4af_n.jpg
  inflating: flowers/dandelion/62293290_2c463891ff_m.jpg
  inflating: flowers/dandelion/6229634119_af5fec0a22.jpg
  inflating: flowers/dandelion/6250363717_17732e992e_n.jpg
  inflating: flowers/dandelion/6400843175_ef07053f8f_m.jpg
  inflating: flowers/dandelion/6412422565_ce61ca48a9_n.jpg
  inflating: flowers/dandelion/645330051_06b192b7e1.jpg
  inflating: flowers/dandelion/6495802659_98b57e0cca_m.jpg
  inflating: flowers/dandelion/674407101_57676c40fb.jpg
  inflating: flowers/dandelion/6888894675_524a6accab_n.jpg
  inflating: flowers/dandelion/6897671808_57230e04c5_n.jpg
  inflating: flowers/dandelion/6900157914_c3387c11d8.jpg
  inflating: flowers/dandelion/6901435398_b3192ff7f8_m.jpg
  inflating: flowers/dandelion/6918170172_3215766bf4_m.jpg
```

Question-2:

IMAGE AUGUMENTATION

Solution

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
xtrain
```

```
Found 4317 images belonging to 5 classes.
<keras.preprocessing.image.DirectoryIterator at 0x7f0656af6fd0>
```

Question-3:

CREATE MODEL

Solution

from tensorflow.keras.models import Sequential

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

Question-4:

ADD LAYERS (CONVOLUTION,MAX POOLING ,FLATTEN, DENSE(HIDDEN LAYERS) ,OUTPUT)

Solution

 # Flatten layer

model = Sequential()

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

model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling layer

model.add(Flatten())

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

```
<keras.engine.sequential.Sequential at 0x7f0657db3710>
```

 # Dense layers

model.add(Dense(300,activation='relu')) # Hidden layer

model.add(Dense(150,activation='relu')) # Hidden layer

model.add(Dense(5,activation='softmax')) # Output layer

```
[17] model.add(Dense(300,activation='relu')) # Hidden layer
     model.add(Dense(150,activation='relu')) # Hidden layer
     model.add(Dense(5,activation='softmax')) # Output layer
```

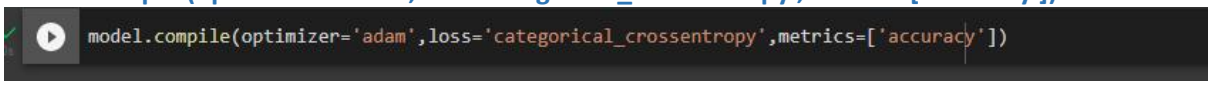
```
     <keras.engine.sequential.Sequential at 0x7f0657db3710>
```

Question-5:

COMPILE THE MODEL

Solution

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

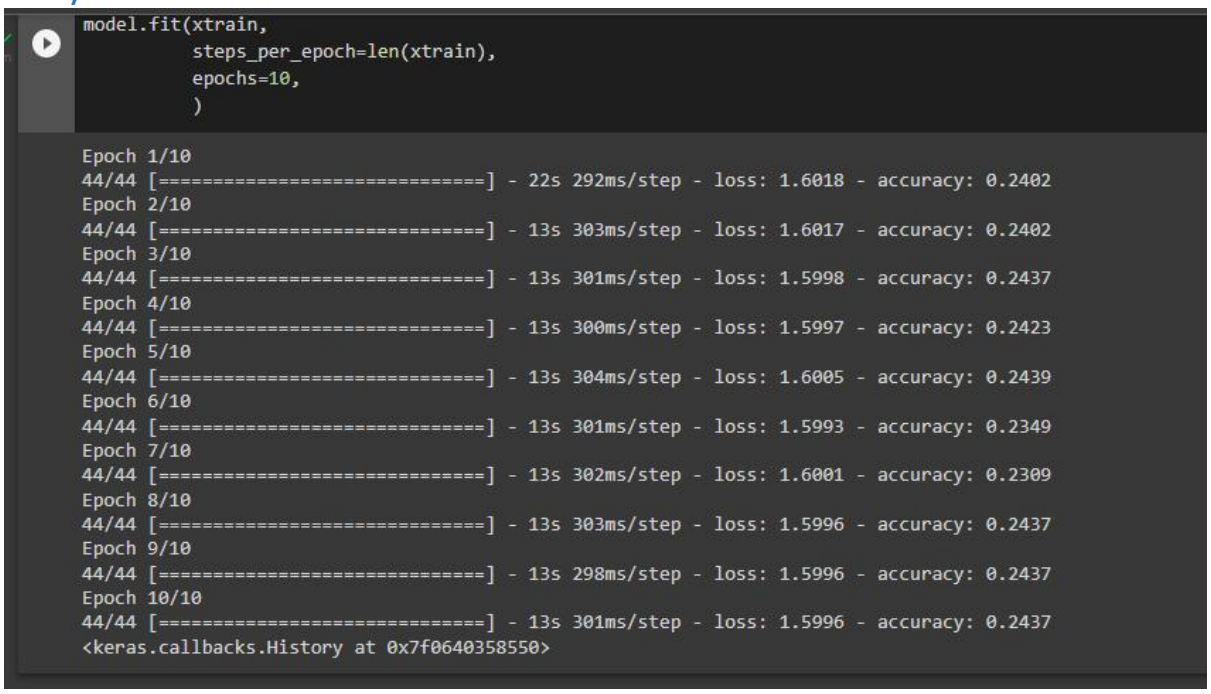


Question-6:

FIT THE MODEL

Solution

```
model.fit(xtrain,
      steps_per_epoch=len(xtrain),
      epochs=10,
      )
```

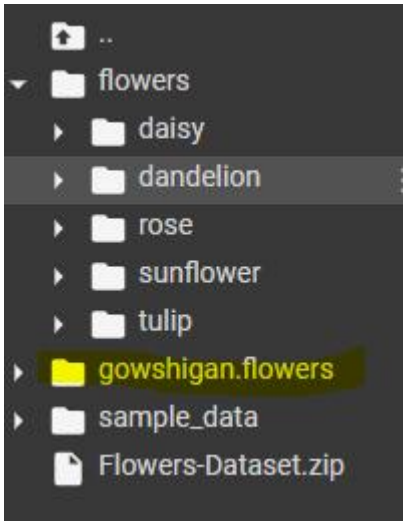


Question-7:

SAVE THE MODEL

Solution

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

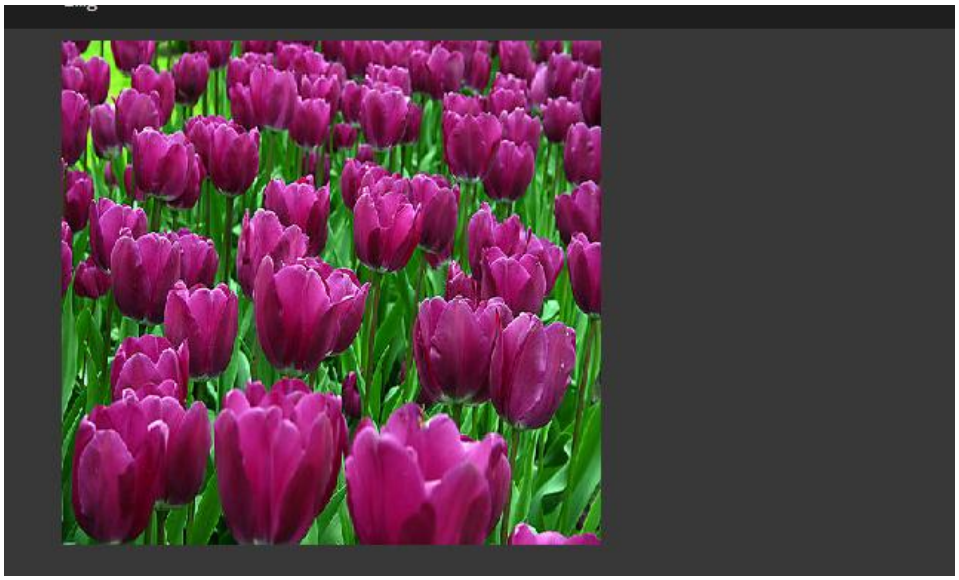


Question-8:

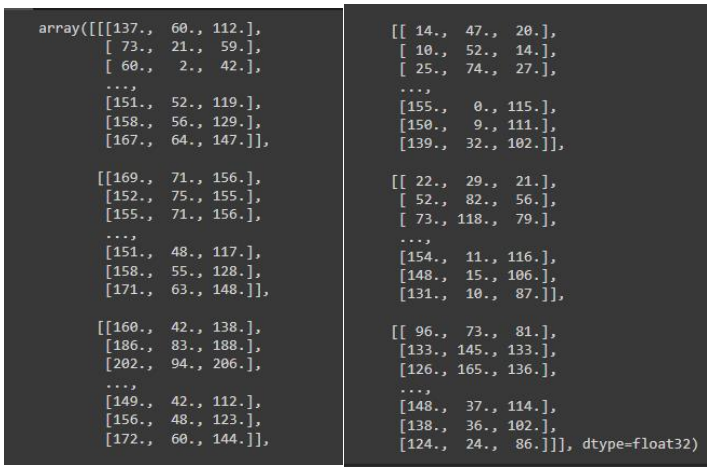
TEST THE MODEL

Solution

```
import numpy as np

from tensorflow.keras.preprocessing import image

img=image.load_img('/content/flowers/tulip/11746276_de3dec8201.jpg',target_size=(64,64)

Img
```



```
x = image.img_to_array(img)

X
```

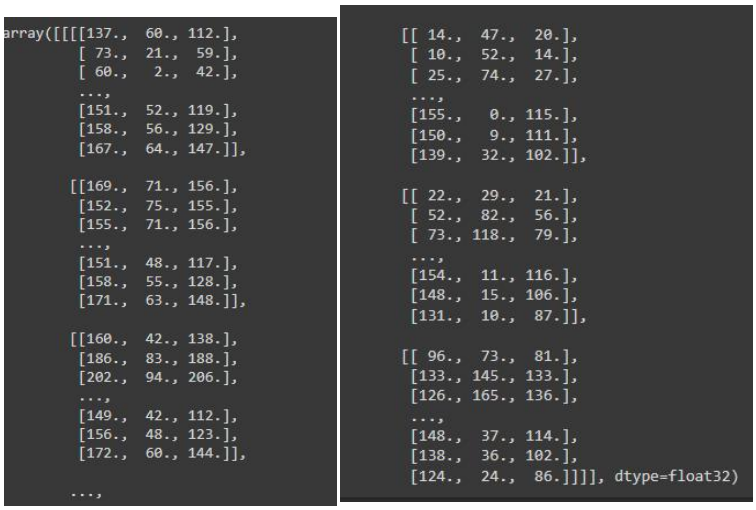


```
array([[[137.,  60., 112.],
        [ 73.,  21.,  59.],
        [ 60.,   2.,  42.],
        ...,
        [151.,  52., 119.],
        [158.,  56., 129.],
        [167.,  64., 147.]],

       [[169.,  71., 156.],
        [152.,  75., 155.],
        [155.,  71., 156.],
        ...,
        [151.,  48., 117.],
        [158.,  55., 128.],
        [171.,  63., 148.]],

       [[160.,  42., 138.],
        [186.,  83., 188.],
        [202.,  94., 206.],
        ...,
        [149.,  42., 112.],
        [156.,  48., 123.],
        [172.,  60., 144.]],
```

```
[[ 14.,  47.,  20.],
 [ 10.,  52.,  14.],
 [ 25.,  74.,  27.],
 ...,
 [155.,   0., 115.],
 [150.,   9., 111.],
 [139.,  32., 102.]],

[[ 22.,  29.,  21.],
 [ 52.,  82.,  56.],
 [ 73., 118.,  79.],
 ...,
 [154.,  11., 116.],
 [148.,  15., 106.],
 [131.,  10.,  87.]],

[[ 96.,  73.,  81.],
 [133., 145., 133.],
 [126., 165., 136.],
 ...,
 [148.,  37., 114.],
 [138.,  36., 102.],
 [124.,  24.,  86.]]], dtype=float32)
```

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

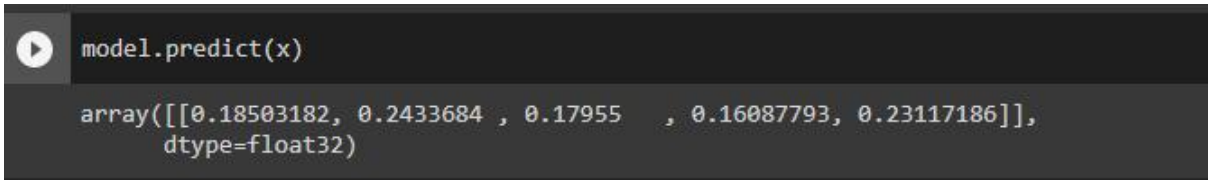
X



```
array([[[[137.,  60., 112.],
         [ 73.,  21.,  59.],
         [ 60.,   2.,  42.],
         ...,
         [151.,  52., 119.],
         [158.,  56., 129.],
         [167.,  64., 147.]],

        [[169.,  71., 156.],
         [152.,  75., 155.],
         [155.,  71., 156.],
         ...,
         [151.,  48., 117.],
         [158.,  55., 128.],
         [171.,  63., 148.]],

        [[160.,  42., 138.],
         [186.,  83., 188.],
         [202.,  94., 206.],
         ...,
         [149.,  42., 112.],
         [156.,  48., 123.],
         [172.,  60., 144.]],

        ...,
```

```
 [[ 14.,  47.,  20.],
  [ 10.,  52.,  14.],
  [ 25.,  74.,  27.],
  ...,
  [155.,   0., 115.],
  [150.,   9., 111.],
  [139.,  32., 102.]],

 [[ 22.,  29.,  21.],
  [ 52.,  82.,  56.],
  [ 73., 118.,  79.],
  ...,
  [154.,  11., 116.],
  [148.,  15., 106.],
  [131.,  10.,  87.]],

 [[ 96.,  73.,  81.],
  [133., 145., 133.],
  [126., 165., 136.],
  ...,
  [148.,  37., 114.],
  [138.,  36., 102.],
  [124.,  24.,  86.]]]], dtype=float32)
```

model.predict(x)



```
model.predict(x)

array([[0.18503182, 0.2433684 , 0.17955   , 0.16087793, 0.23117186]],
      dtype=float32)
```

xtrain.class_indices

```
xtrain.class_indices

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

op = ['daisy','dandelion','rose','sunflower','tulip']
pred = np.argmax(model.predict(x))
op[pred]

```
'dandelion'
```

img=image.load_img('/content/flowers/sunflower/1022552002_2b93faf9e7_n.jpg',target_size=(500,500)

Img