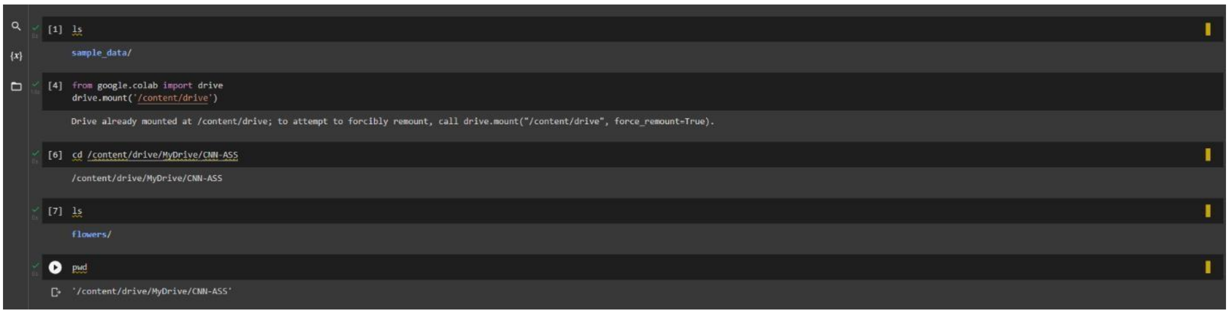
Assignment –3

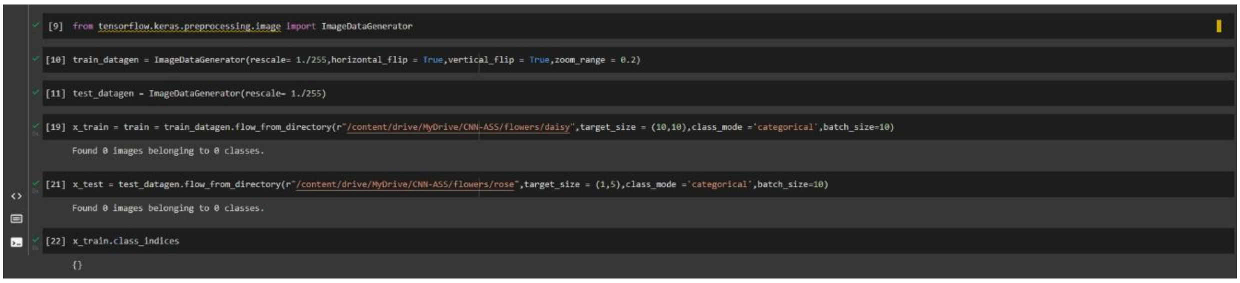
Build CNN Model for Classification of Flowers

Assignment Date	05 October 2022
Student Name	Janani priya R
Student Roll Number	2019PITIT301
Maximum Marks	2 Marks



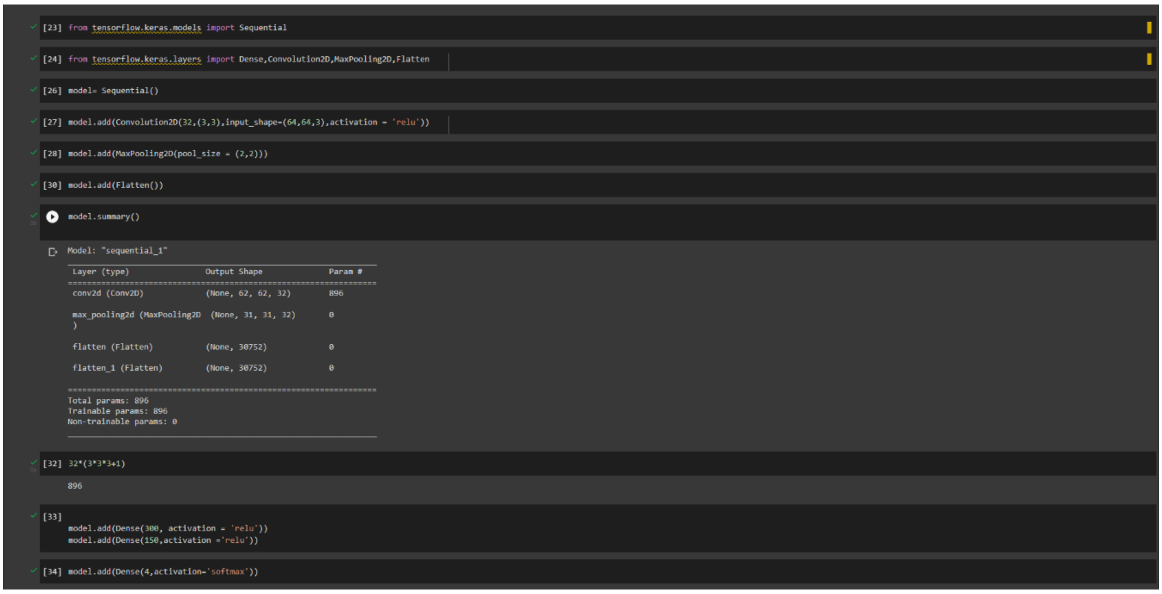
```
[1] ls
    sample_data/

[4] from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[6] cd /content/drive/MyDrive/CNN-ASS
    /content/drive/MyDrive/CNN-ASS

[7] ls
    flowers/

    pwd
    '/content/drive/MyDrive/CNN-ASS'
```

1. Download the Dataset



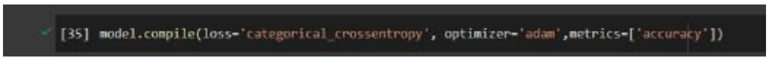
```
[9] from tensorflow.keras.preprocessing.image import ImageDataGenerator

[10] train_datagen = ImageDataGenerator(rescale= 1./255,horizontal_flip = True,vertical_flip = True,zoom_range = 0.2)

[11] test_datagen = ImageDataGenerator(rescale= 1./255)

[19] x_train = train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/CNN-ASS/flowers/daisy",target_size = (10,10),class_mode ='categorical',batch_size=10)
     Found 0 images belonging to 0 classes.

[21] x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/CNN-ASS/flowers/rose",target_size = (1,5),class_mode ='categorical',batch_size=10)
     Found 0 images belonging to 0 classes.

[22] x_train.class_indices
     {}
```

2. Creating Model



```
[23] from tensorflow.keras.models import Sequential

[24] from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten

[26] model= Sequential()

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

[28] model.add(MaxPooling2D(pool_size = (2,2)))

[30] 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

      flatten_1 (Flatten)         (None, 30752)             0

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

[32] 32*(3*3*3+1)

     896

[33]
     model.add(Dense(300, activation = 'relu'))
     model.add(Dense(150,activation ='relu'))

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

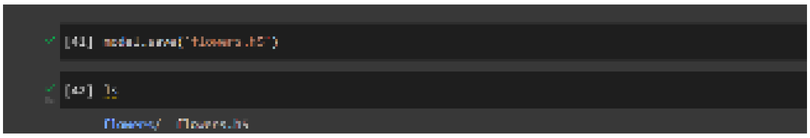
3. Adding Layers



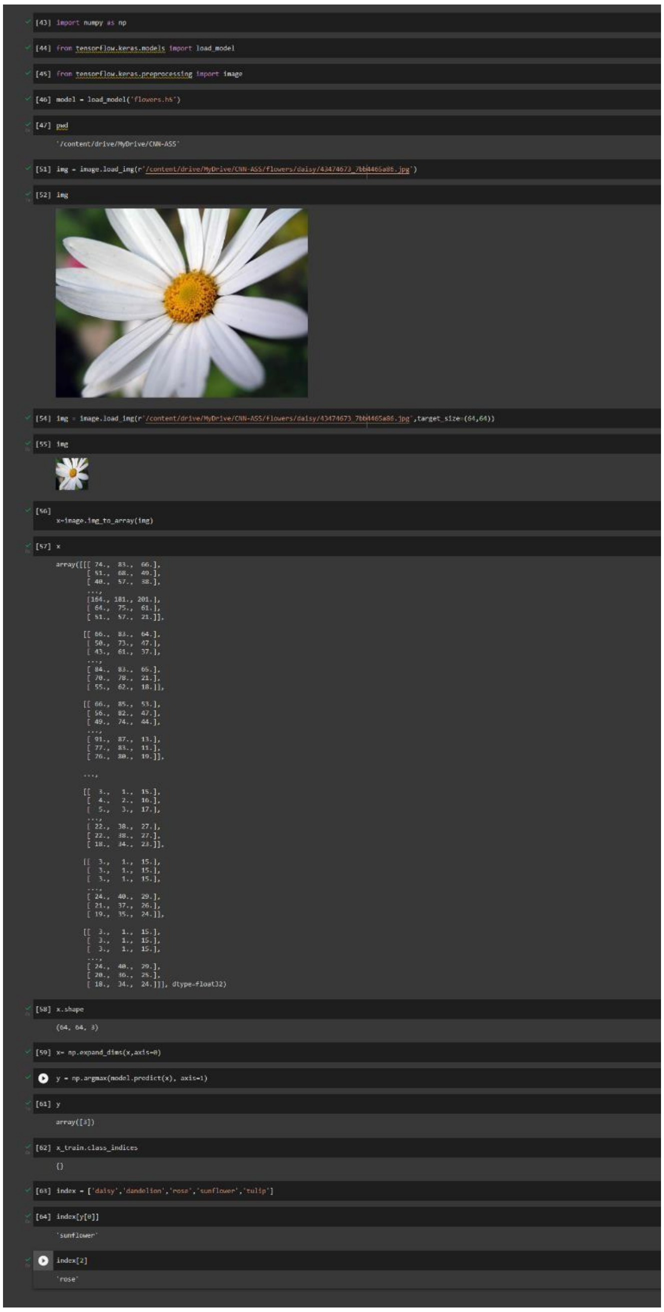
```
[35] model.compile(loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
```

4. Compiling the model

5. Saving the model



```
[41] model.save('flowers.h5')

[42] ls

     flowers/  flowers.h5
```



```
[43] import numpy as np

[44] from tensorflow.keras.models import load_model

[45] from tensorflow.keras.preprocessing import image

[46] model = load_model('flowers.h5')

[47] pwd

     '/content/drive/MyDrive/CNN-ASS'

[51] img = image.load_img(r'/content/drive/MyDrive/CNN-ASS/flowers/daisy/43474673_7bb4465a86.jpg')

[52] img
```

```
[54] img = image.load_img(r'/content/drive/MyDrive/CNN-ASS/flowers/daisy/43474673_7bb4465a86.jpg',target_size=(64,64))

[55] img
```

```
[56]
     x=image.img_to_array(img)

[57] x

     array([[[ 74.,  83.,  66.],
             [ 51.,  68.,  49.],
             [ 40.,  57.,  38.],
             ...,
             [164., 181., 201.],
             [ 64.,  75.,  61.],
             [ 51.,  57.,  21.]],

            [[ 66.,  83.,  64.],
             [ 50.,  73.,  47.],
             [ 43.,  61.,  37.],
             ...,
             [ 84.,  83.,  65.],
             [ 70.,  78.,  21.],
             [ 55.,  62.,  18.]],

            [[ 66.,  85.,  53.],
             [ 56.,  82.,  47.],
             [ 49.,  74.,  44.],
             ...,
             [ 91.,  87.,  13.],
             [ 77.,  83.,  11.],
             [ 76.,  80.,  19.]],

            ...,

            [[  3.,   1.,  15.],
             [  4.,   2.,  16.],
             [  5.,   3.,  17.],
             ...,
             [ 22.,  38.,  27.],
             [ 22.,  38.,  27.],
             [ 18.,  34.,  23.]],

            [[  3.,   1.,  15.],
             [  3.,   1.,  15.],
             [  3.,   1.,  15.],
             ...,
             [ 24.,  40.,  29.],
             [ 21.,  37.,  26.],
             [ 19.,  35.,  24.]],

            [[  3.,   1.,  15.],
             [  3.,   1.,  15.],
             [  3.,   1.,  15.],
             ...,
             [ 24.,  40.,  29.],
             [ 20.,  36.,  25.],
             [ 18.,  34.,  24.]]], dtype=float32)

[58] x.shape

     (64, 64, 3)

[59] x= np.expand_dims(x,axis=0)

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

[61] y

     array([3])

[62] x_train.class_indices

     {}

[63] index = ['daisy','dandelion','rose','sunflower','tulip']

[64] index[y[0]]

     'sunflower'

     index[2]

     'rose'
```

6. Test The model