

Assignment –3

Build CNN Model for Classification of Flowers

Assignment Date	05 October 2022
Student Name	Harenee A.S
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/

[8] 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_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())
[31] 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(100, 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.h2')

[42] !ls
flowers/ flowers.h2
```

```

[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] path

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


[51] img = image.load_img('/content/drive/drive/CNN-ASS/flowers/daisy/45478673_7888455ab5.jpg')

[52] img



[54] img = image.load_img('/content/drive/drive/CNN-ASS/flowers/daisy/45478673_7888455ab5.jpg', target_size=(64,64))

[55] img



[56] x=image.img_to_array(img)

[57] x

array([[[[ 74.,  85.,  66.],
         [ 51.,  68.,  48.],
         [ 40.,  57.,  48.],
         ...,
         [164., 185., 204.],
         [ 64.,  75.,  61.],
         [ 51.,  57.,  21.]],

        [[ 60.,  81.,  64.],
         [ 58.,  73.,  47.],
         [ 43.,  61.,  37.],
         ...,
         [ 84.,  83.,  46.],
         [ 70.,  78.,  21.],
         [ 55.,  62.,  18.]],

        [[ 66.,  85.,  51.],
         [ 50.,  82.,  47.],
         [ 40.,  78.,  44.],
         ...,
         [ 91.,  87.,  11.],
         [ 77.,  63.,  19.],
         [ 70.,  88.,  19.]],

        ...,

        [[ 4.,  1.,  14.],
         [ 4.,  2.,  44.],
         [ 3.,  3.,  17.],
         ...,
         [ 22.,  38.,  27.],
         [ 22.,  88.,  27.],
         [ 18.,  48.,  24.]],

        [[ 3.,  1.,  15.],
         [ 3.,  1.,  15.],
         [ 3.,  1.,  15.],
         ...,
         [ 24.,  48.,  29.],
         [ 21.,  37.,  26.],
         [ 19.,  15.,  24.]],

        [[ 3.,  1.,  15.],
         [ 3.,  2.,  15.],
         [ 3.,  1.,  15.],
         ...,
         [ 24.,  48.,  29.],
         [ 20.,  88.,  29.],
         [ 18.,  34.,  28.]]], dtype=float32)]

[58] x.shape

(64, 64, 3)

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

[60] y = np.argmax(model.predict(x), axis=-1)

[61] y

array([1])

[62] x_train_class_indices

{}

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

[64] index[y[0]]

'sunflower'

[65] index[z]

'rose'

```

6. Test The model