

Assignment – 3

Build CNN Model for Classification of Flowers

| | |
|---------------------|----------------------|
| Assignment Date | 05 October 2022 |
| Student Name | Navaneethakrishnan S |
| Student Roll Number | 195002076 |
| Maximum Marks | 2 Marks |

1. Download the Dataset

```
[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/

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

2. Creating Model

```
[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
{}

[22] x_train.class_indices
{}
```

3. Adding Layers

```
[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, 32, 32, 32) 896
max_pooling2d (MaxPooling2D) (None, 16, 16, 32) 0
flatten (Flatten) (None, 16384) 0
flatten_1 (Flatten) (None, 16384) 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'))
```

4. Compiling the model

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

5. Saving the model

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

```
[43] %s
```

flowers flowers.h5

6. Test The model

```
[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] print
      /content/drive/MyDrive/CNN-455
[48] img = image.load_img('/content/drive/MyDrive/CNN-455/flowers/daisy/044/044.jpg')
[49] img
[50] image.load_img('/content/drive/MyDrive/CNN-455/flowers/daisy/044/044.jpg', target_size=(64,64))
[51] img
[52] x=image.img_to_array(img)
[53] x
      array([[[[ 74., 83., 66.],
                [ 51., 66., 60.],
                [ 46., 57., 58.],
                ...,
                [166., 182., 201.],
                [ 66., 75., 61.],
                [ 51., 67., 58.]],
                [[ 66., 83., 64.],
                [ 50., 73., 67.],
                [ 45., 65., 57.],
                ...,
                [ 88., 83., 66.],
                [ 70., 70., 55.],
                [ 55., 65., 58.]],
                [[ 66., 85., 53.],
                [ 66., 82., 67.],
                [ 60., 76., 68.],
                ...,
                [ 93., 87., 81.],
                [ 77., 83., 81.],
                [ 70., 69., 59.]],
                ...,
                [[ 3., 3., 15.],
                [ 4., 2., 16.],
                [ 5., 7., 17.],
                ...,
                [ 22., 36., 27.],
                [ 22., 48., 27.],
                [ 18., 34., 22.]],
                [[ 3., 3., 15.],
                [ 3., 3., 15.],
                [ 3., 3., 15.],
                ...,
                [ 24., 40., 20.],
                [ 21., 37., 26.],
                [ 19., 35., 24.]],
                [[ 3., 3., 15.],
                [ 3., 3., 15.],
                [ 3., 3., 15.],
                ...,
                [ 24., 40., 20.],
                [ 20., 40., 20.],
                [ 18., 34., 24.]]], dtype=float32)])
[54] x.shape
      (64, 64, 3)
[55] x = np.expand_dims(x, axis=0)
[56] y = np.argmax(model.predict(x), axis=-1)
[57] y
      array([0])
[58] x_train_class_indices
      {}
[59] index = ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
[60] index[y[0]]
      'sunflower'
[61] index[y]
      'rose'
```