University of Rajshahi

Assignments of DIP Lab

Course Name: Digital Image Processing Lab
Course Code: CSE-4182
Date: September 9, 2022

Prepared by
Name: Md. Meem Mursalin Chowdhury
ID: 1810276103
Year: 4th, Semester: Odd
Session: 2017-18

Submitted to
Md. Khademul Islam Molla, Professor
MD. Rokanujjaman, Professor
Sangeeta Biswas, Associate Professor

Dept. of Computer Science and Engineering,
University of Rajshahi

Abstract

Digital Image Processing or DIP is a very well-known term to most of the people. It refers to the processing of different images. This processing is done images on basis of the need and demand. In this lab all the images are processed using python3 and the grayscale images or 2D images are considered.

Contents

1 Assignment-12 **1**
1.1 Introduction 1
1.2 Coding 1
1.3 Result Discussion 2

1 Assignment-12

1.1 Introduction

Problem: Write a report on what different convolution layers of a CNN based image classifier see. You may use the attached code. You need to install Tensorflow in your computer. You can run it in Google Colab for GPU support if you do not have GPU or enough memory.

Solution: Here we have provide a report on what happens on different convolution layers of a CNN based image classifier. As the CNN model uses different layers, we can see different things happens to the images in different layers. The result and code of it given as follows.

1.2 Coding

```python
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
import cv2
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.models import Model

DIR = '/media/mursalin/New Volume/Image-Processing/code_13/'
# DIR = './'

def main():
  # Load a pre-trained model.
  baseModel = VGG16()
  baseModel.summary()

  # Prepare a new model having the desired layer as the output layer.
  for i in range(10):
    layer_number = i +1# *** Change layer number to see different convolution layer's output.
    print(layer_number)
    inputs = baseModel.input
    outputs = baseModel.layers[layer_number].output
    model = Model(inputs, outputs)

    # Prepare data.
    img = prepare_data()

    # Predict output of a specific layer.
    outputs = model.predict(img)

    # Display what different channls see.
    display_channels(outputs, layer_number)

def display_channels(chSet, layer_no):
  plt.figure(figsize = (20, 20))
  plt.suptitle('Layer-' + str(layer_no))
  for i in range(9):
    plt.subplot(3, 3, i + 1)
    plt.title('Channel-' + str(i))
    plt.imshow(chSet[0, :, :, i], cmap = 'gray')
    plt.axis('off')

  plt.savefig(DIR+'fig-'+str(layer_no)+'.jpg')
  plt.show()
  plt.close()

def prepare_data():
  # Load an image
```

```python
    imgPath = DIR + 'rose.jpg'  #'Elephant.jpg' #'Baby.jpeg' #'Rose.jpeg' #'Boat.
jpeg' #
    bgrImg = cv2.imread(imgPath)
    print(bgrImg.shape)

    # Convert the image from BGR into RGB format
    rgbImg = cv2.cvtColor(bgrImg, cv2.COLOR_BGR2RGB)

    # Reshape the image so that it can fit into the model.
    #display_img(rgbImg)
    rgbImg = cv2.resize(rgbImg, (224, 224))
    display_img(rgbImg)

    # Expand dimension since the model accepts 4D data.
    print(rgbImg.shape)
    rgbImg = np.expand_dims(rgbImg, axis = 0)
    print(rgbImg.shape)

    # Preprocess image
    rgbImg = preprocess_input(rgbImg)

    return rgbImg

def display_img(img):
    plt.imshow(img)
    plt.show()
    plt.close()

if __name__ == '__main__':
    main()


```

Listing 1: Using pre trained model(VGG16)

1.3 Result Discussion

As we are using the VGG16 model as CNN model, there are 16 different layers are produced here. As different convolution layer are present there, so if we give input of an image, we will get different kinds of output for each kind of layers. For this experiment, we have taken output of 10 picture for each layer and found out the difference. The output picture of 10 different layers is given below. From the 1st layer, we can see that the input picture can be easily recognised. Here the edges of the object can be easily understood and here sobel filter is perhaps used. From layer-2 output picture, the picture continues to become blur and at the end of layer 10, the picture can be no more understood. After the finished layer output, the given picture can be no more recognised.

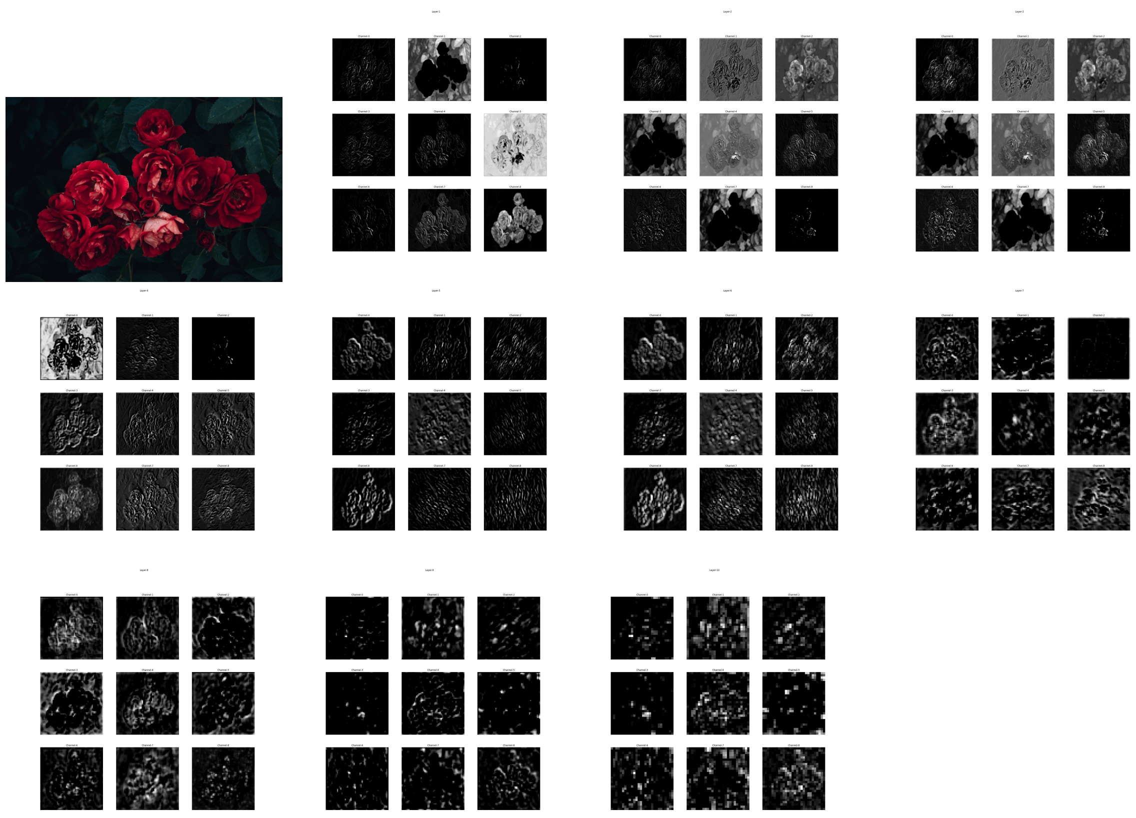

Figure 1: Input image, Output of Different Layers of VGG16 model from layer 1-10