assignment05

October 18, 2018

1 This is assignment05

1.1 Name : Jaehyun Lim

1.2 Student ID : 20145450

1.3 Denition of the convolution kernel for computing the derivative in x-direction

$$\frac{\partial f}{\partial x} = \lim_{h \to \infty} \frac{(x+h) + f(x)}{h} \Rightarrow kernel = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

1.4 Denition of the convolution kernel for computing the derivative in y-direction

$$\frac{\partial f}{\partial x} = \lim_{h \to \infty} \frac{(x+h) + f(x)}{h} \Rightarrow kernel = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

1.5 import packages

```
In [3]: import matplotlib.pyplot as plt
        import numpy as np
        from scipy import signal
        from skimage import io, color
        from skimage import exposure
        import pylab
```

1.6 Load the image

```
In [9]: im_color = io.imread('cau.jpg')
        plt.imshow(im_color)
        plt.axis('off')
        plt.show()
```

1.7 Convert the image to grayscale

```
In [29]: im_gray = color.rgb2gray(im_color)
         plt.imshow(im_gray, cmap='gray')
         plt.axis('off')
         plt.show()
```

1.8 Function : return convolution of image and kernel

```
In [16]: def convolution(image, kernel):

             kernel = np.flipud(np.fliplr(kernel))
             output = np.zeros_like(image)
             image_padded = np.zeros((image.shape[0] + 2, image.shape[1] + 2))
             image_padded[1:-1, 1:-1] = image
             for x in range(image.shape[1]):
                 for y in range(image.shape[0]):
                     output[y,x]=(kernel*image_padded[y:y+3,x:x+3]).sum()
             return output
```

1.9 Smoothing kernel

```
In [30]: kernel = np.array([[0,0,0],[0,1,0],[0,-0.5,0.3]])
         image = convolution(im_gray,kernel)
         plt.imshow(image, cmap='gray')
         plt.axis('off')
         plt.show()
```

1.10 My kernel

1.10.1 Color inversion image

```
In [31]: kernel = np.array([[-1,-1,-1],[0,1,0],[-1,-1,-1]])
         image = convolution(im_gray,kernel)
         plt.imshow(image, cmap='gray')
         plt.axis('off')
         plt.show()
```