**Computer Vision HW10**

R08922a27 資工系 人工智慧碩士班 李吉昌

I use python 3.7 to implement all image processing requirements. Reading .bmp file by **PIL**, and then processing through **NumPy** array.

* **1. Results**

****

* **2. Code fragment**

def binarize(img, thr, Islower=True):

img\_bin = np.zeros(img.shape)

if Islower:

img\_bin[img < thr] = 255

else:

img\_bin[img >= thr] = 255

return img\_bin

def laplace\_mask1(img):

k = np.array([

[0, 1, 0],

[1, -4, 1],

[0, 1, 0]

])

return signal.convolve2d(img, k)

def laplace\_mask2(img):

k = np.array([

[1., 1, 1],

[1, -8, 1],

[1, 1, 1]

]) / 3

return signal.convolve2d(img, k)

def minimum\_variance\_laplacian(img):

k = np.array([

[2., -1, 2],

[-1, -4, -1],

[2, -1, 2]

]) / 3

return signal.convolve2d(img, k)

def laplace\_of\_gaussian(img):

k = np.array([

[0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0],

[0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0],

[0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0],

[-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1],

[-1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1],

[-2, -9, -23, -1, 103, 178, 103, -1, -23, -9, -2],

[-1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1],

[-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1],

[0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0],

[0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0],

[0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0]

])

return signal.convolve2d(img, k)

def difference\_of\_gaussian(img):

k = np.array([

[-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1],

[-3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],

[-4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],

[-6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6],

[-7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7],

[-8, -13, -17, 15, 160, 283, 160, 15, -17, -13, -8],

[-7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7],

[-6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6],

[-4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],

[-3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],

[-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1],

])

return signal.convolve2d(img, k)

* **3. Brief Description**

All the assigned operators' implementation details follow the course's lecture slides. The binarize function are shared for all the assigned operators and the 2d convolution operation is called from the **scipy.signal** package.