Computer Vision Homework 4

Name: 黃新予
Student ID: f08922136

I. Environment Setup

Language: Python 3 (on VS code)
Library: numpy, PIL

II. Q1: Dilation

將 kernel 中心點對齊原始圖的每一點，如果原圖的 pixel 值為 255，則把周圍的點與 kernel 重疊且 kernel 為 1 的部分全部改成 255

```python
def Dilation(image, kernel, kernelCenter):
    '''
    image: Image (from PIL)
    kernel: np.array (from numpy)
    kernelCenter: tuple(x, y)
    '''
    width, height = image.size
    dilationImage = Image.new('L', image.size)
    # compare image with kernel
    for c in range(width):
        for r in range(height):
            # dilate for white elements
            if image.getpixel((c,r)) == 255:
                for i in range(kernel.shape[0]):
                    for j in range(kernel.shape[1]):
                        if 0 <= c+i-kernelCenter[0] < width and 0 <= r+j-kernelCenter[1] < height and kernel[i, j] == 1:
                            dilationImage.putpixel((c+i-kernelCenter[0], r+j-kernelCenter[1]),255)
    return dilationImage
```

III. Q2: Erosion

將 kernel 中心放進白色 255 的區域，針對 kernel 中為 1 的值進行比對。如果比對結果完全一致，則設為 255，若有一格不同，則設為 0

```python
def Erosion(image, kernel, kernelCenter):
    '''
    image: Image (from PIL)
    kernel: np.array (from numpy)
    '''
    width, height = image.size
    erosionImage = Image.new('1', image.size)
    # compare image with kernel
    for c in range(width):
        for r in range(height):
            # erosion for white elements
            # if all match => put 1, not match => 0
            isMatch = True
            for i in range(kernel.shape[0]):
                for j in range(kernel.shape[1]):
                    # out of range case
                    if kernel[i, j] == 1:
                        if 0 <= c+i - kernelCenter[0] < width and 0 <= r+j - kernelCenter[1] < height:
                            if image.getpixel((c+i - kernelCenter[0], r+j - kernelCenter[1])) == 0:
                                isMatch = False
                                break
                        else:
                            isMatch = False
                            break
                if not isMatch:
                    break
            if isMatch:
                erosionImage.putpixel((c,r), 255)
    return erosionImage
```

IV. Q3:opening

先做 erosion 再做 dilation

```python
def Opening(image, kernel,kernelCenter):
    return Dilation(Erosion(image,kernel, kernelCenter),kernel,kernelCenter)
```

V. Q4: Closing

與 opening 相反，先做 dilation, 再做 erosion

```python
def Closing(image, kernel,kernelCenter):
    return Erosion(Dilation(image, kernel,kernelCenter), kernel,kernelCenter)
```

VI. Q5: Hit and Miss

先定義兩個 kernel J, K，再根據講義順序依序 call functions

intersection: 比較兩張圖，如果兩圖皆為 255，則新圖設為 255，其餘則設為 0

Complement: 將原圖形 255 和 0 對調

$$A \otimes (J,K) = (A \ominus J) \cap (A^c \ominus K)$$

```
def Hit_and_miss_transform(image, kernelJ, cneterJ, kernelK, centerK):
    return Intersection(Erosion(image, kernelJ, centerJ), Erosion(Complement(image), kernelK, centerK))
```