Computer Vision HW4

R04921040 李中原

Language: Python3
Description: I use OpenCV to do image I/O, and load raw pixel data (height , length , channels) from lena.bmp.

Run: python hw4.py

Kernel:

```
octogonal_kernel = [
    [0 , 1 , 1 , 1 , 0],
    [1 , 1 , 1 , 1 , 1],
    [1 , 1 , 1 , 1 , 1],
    [1 , 1 , 1 , 1 , 1],
    [0 , 1 , 1 , 1 , 0]
]
```

Dilation:

```
def Dilation(cover):
    for i in range(cover.shape[0]):
        for j in range(cover.shape[1]):
            if octogonal_kernel[i][j] * cover[i][j] == 255:
                return 255

    return 0
```

每個點的值是將周遭 5*5 的範圍的點傳入 Dilation 做判斷，對應 kernel 形狀中只要有任意一點的值為 255，則回傳 255，若全部都沒有則回傳 0

Erosion:

```
def Erosion(cover):
    for i in range(cover.shape[0]):
        for j in range(cover.shape[1]):
            if (octogonal_kernel[i][j] != 0) and (cover[i][j] == 0):
                return 0

    return 255
```

跟 Dilation 概念類似，不過改為在 kernel 覆蓋範圍內只要有任意一點值為 0 則回傳 0

Opening、Closing:

```
for i in range(2 , height-2):
    for j in range(2 , width-2):
        cover = lena_dilation[i-2:i+3,j-2:j+3]
        lena_closing[i][j] = Erosion(cover)

        cover = lena_erosion[i-2:i+3,j-2:j+3]
        lena_opening[i][j] = Dilation(cover)
```

Opening 是先做 Erosion 再做 Dilation，Closing 則是先 Dilation 再做 Erosion

Opening:

Closing:

Hit-and-Miss:

同時對 kernel_j、kernel_k 的範圍做判斷:

以該點為中心，如果在 kernel_j 的範圍內都是 1 並且在 kernel_k 的範圍都是 0，則回傳 255，以本題為例可找出右上角的點

```
kernel_j = [
    [0 , 0 , 0],
    [1 , 1 , 0],
    [0 , 1 , 0]
]
```

```
kernel_k = [
    [0 , 1 , 1],
    [0 , 0 , 1],
    [0 , 0 , 0]
]
```

```
def HitandMiss(image):
    output = np.zeros(image.shape)

    for i in range(1 , image.shape[0] - 1):
        for j in range(1 , image.shape[1] - 1):
            if image[i][j - 1] == 255:
                if image[i][j] == 255:
                    if image[i + 1][j] == 255:
                        if image[i - 1][j] == 0:
                            if image[i - 1][j + 1] == 0:
                                if image[i][j + 1] == 0:
                                    output[i][j] = 255

    cv2.imwrite('HitandMiss.bmp' , output)
```