

Computer Vision 2019 Fall

Homework #5

B06902059 資工三 謝宜儒

Description

This homework focuses on grayscale morphology on a image.

Results

(a) Dilation



(b) Erosion



(c) Opening



(d) Closing



Source Code (fragment)

```
octagon = np.array([[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]])

class Kernel:
    def __init__(self, origin, pattern):
        offsets = []
        for i in range(pattern.shape[0]):
            for j in range(pattern.shape[1]):
                if(pattern[i][j] == 1):
                    offsets.append([i - origin[0], j - origin[1]])

        self._offsets = offsets

    def offsets(self):
        return self._offsets

def plot_img(img, filename, title):
    cv2.imwrite(filename, img)
    cv2.imshow(title, img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

def dilation(img, kernel):
    dilated_img = np.zeros((length, width), dtype = np.uint8)
    for i in range(length):
```

```

    for j in range(width):
        max_color = 0
        for offset in kernel.offsets():
            if(i + offset[0] >= 0 and i + offset[0] < length
               and j + offset[1] >= 0 and j + offset[1] < width):
                max_color = max(max_color, img[i + offset[0]][j + offset[1]])
            dilated_img[i][j] = int(max_color)

    return dilated_img

def erosion(img, kernel):
    eroded_img = np.zeros((length, width), dtype = np.uint8)
    for i in range(length):
        for j in range(width):
            min_color = 255
            for offset in kernel.offsets():
                if(i + offset[0] >= 0 and i + offset[0] < length
                   and j + offset[1] >= 0 and j + offset[1] < width):
                    if(img[j + offset[0]][j + offset[1]] == 0):
                        break
                    else:
                        min_color = min(min_color, img[i + offset[0]][j + offset[1]])
            else:
                break
        else:
            eroded_img[i][j] = int(min_color)

    return eroded_img

def opening(img, kernel):
    return dilation(erosion(img, kernel), kernel)

def closing(img, kernel):
    return erosion(dilation(img, kernel), kernel)

if __name__ == "__main__":
    # Read the input image in grayscale mode
    img = cv2.imread(sys.argv[1], cv2.IMREAD_GRAYSCALE)
    length, width = img.shape[0], img.shape[1]

    oct_kernel = Kernel((2, 2), octagon)

    if(sys.argv[2] == "1"):
        plot_img(dilation(img, oct_kernel), '01_dilated.bmp', 'Dilated Image')
    elif(sys.argv[2] == "2"):
        plot_img(erosion(img, oct_kernel), '02_eroded.bmp', 'Eroded Image')
    elif(sys.argv[2] == "3"):
        plot_img(opening(img, oct_kernel), '03_opening.bmp', 'Opening Image')
    elif(sys.argv[2] == "4"):
        plot_img(closing(img, oct_kernel), '04_closing.bmp', 'Closing Image')
    else:

```

```
printf("Invalid argument.\nUsage: python3 hw5.py [input image] [problem  
number]\n")
```

To run the source code, type the following line in a terminal:

```
python3 hw5.py [input image] [problem number]
```

where in this homework, the input image is **lena.bmp** and the problem numbers are **1 ~ 4** .