

# 2021 CV HW5 Report

## Results

a. dilation



b. erosion



c. opening



d. closing



### (a) Dilation

Step1. 首先定義 octagonal 3-5-5-5-3 kernel。

```
kernel = 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]])  
hh, ww = kernel.shape  
hh_half, ww_half = hh//2, ww//2
```

Step2. 每個 pixel 作為 kernel 中心點，找出 kernel 位置為 1 的 pixel value 中的**最大值**作為此 pixel 的值。

```
output = np.zeros((h, w), np.uint8)  
for i in range(h):
```

```

for j in range(w):
    value = []
    for ii in range(hh):
        for jj in range(ww):
            if kernel[ii][jj] == 1:
                if i+ii-hh_half < 0 or j+jj-ww_half < 0 or i+ii-
hh_half >= h or j+jj-ww_half >= w:
                    continue
                value.append(img[i+ii-hh_half][j+jj-ww_half])
            output[i][j] = max(value)

```

## (b) Erosion

每個 pixel 作為 kernel 中心點，找出 kernel 位置為 1 的 pixel value 中的最小值作為此 pixel 的值。

```

output = np.zeros((h, w), np.uint8)
for i in range(h):
    for j in range(w):
        value = []
        for ii in range(hh):
            for jj in range(ww):
                if kernel[ii][jj] == 1:
                    if i+ii-hh_half < 0 or j+jj-ww_half < 0 or i+ii-
hh_half >= h or j+jj-ww_half >= w:
                        continue
                    value.append(img[i+ii-hh_half][j+jj-ww_half])
                output[i][j] = min(value)

```

## (c) Opening

承 a、b 的程式，先做 erosion 再做 dilation。

```

erosion_img = erosion(img)
opening_img = dilation(erosion_img)

```

## (d) Closing

承 a、b 的程式，先做 dilation 再做 erosion。

```

dilation_img = dilation(img)
closing_img = erosion(dilation_img)

```