

Lab 9 Erode / Dilate

ให้นักศึกษาใช้ Colab/Jupyter Notebook ทำตามคำสั่งด้านล่าง (คำสั่งพื้นฐานของ Morphological / ลบลายเส้นวงจร-ขยายขนาดลายเส้นวงจร) แล้ว Capture เป็นไฟล์ภาพหรือ PDF Upload ส่งงานมาใน LMS

1. ให้นักศึกษา ทดลองสร้าง Kernel (Structuring Element) ที่มีขนาด 3x3 3 รูปแบบ โดยสร้างไว้ใน Numpy Array เช่น Rectangle / Cross / Ellipse และ print ผลลัพธ์ตัว Kernel ที่ได้สร้างออกมาแสดง

1.1. รหัสต้นฉบับ

Generate my kernel

```
cross_3x3 = np.array([[0,1,0],[1,1,1],[0,1,0]],dtype=np.uint8)
print("Kernel(Structuring Element) Cross 3x3")
print(cross_3x3)
rect_3x3 = np.ones((3,3),dtype=np.uint8)
print("Kernel(Structuring Element) Rectangle 3x3")
print(rect_3x3)
eiei_3x3 = np.array([[1,0,1],[0,1,0],[1,0,1]],dtype=np.uint8)
print("Kernel(Structuring Element) My custom 3x3")
print(eiei_3x3)
```

```
Kernel(Structuring Element) Cross 3x3
[[0 1 0]
 [1 1 1]
 [0 1 0]]
Kernel(Structuring Element) Rectangle 3x3
[[1 1 1]
 [1 1 1]
 [1 1 1]]
Kernel(Structuring Element) My custom 3x3
[[1 0 1]
 [0 1 0]
 [1 0 1]]
```

2. ทดลองสร้าง Kernel (Structuring Element) ที่มีขนาด 3x3 และ 5x5 รูปแบบ Rectangle / Cross / Ellipse โดยสร้างจากคำสั่ง cv.getStructuringElement() และ print ผลลัพธ์ตัวKernelที่ได้สร้างออกมาแสดง

2.1. รหัสต้นฉบับ

```
rect3x3 = cv.getStructuringElement(cv.MORPH_RECT,(3,3))
cross3x3 = cv.getStructuringElement(cv.MORPH_CROSS,(3,3))
ellipse3x3 = cv.getStructuringElement(cv.MORPH_ELLIPSE,(3,3))
rect5x5 = cv.getStructuringElement(cv.MORPH_RECT,(5,5))
cross5x5 = cv.getStructuringElement(cv.MORPH_CROSS,(5,5))
ellipse5x5 = cv.getStructuringElement(cv.MORPH_ELLIPSE,(5,5))
print(rect3x3)
print(cross3x3)
print(ellipse3x3)
print(rect5x5)
print(cross5x5)
print(ellipse5x5)
```

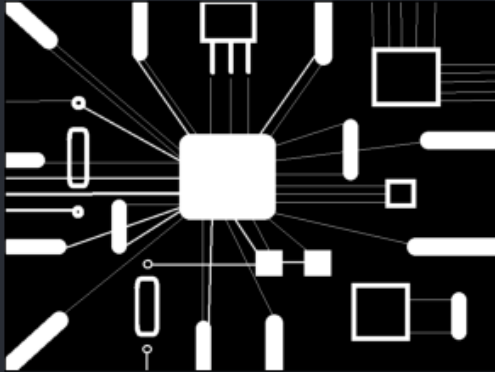
2.2. ผลลัพธ์

```
[[1 1 1]
 [1 1 1]
 [1 1 1]]
[[0 1 0]
 [1 1 1]
 [0 1 0]]
[[0 1 0]
 [1 1 1]
 [0 1 0]]
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]]
[[0 0 1 0 0]
 [0 0 1 0 0]
 [1 1 1 1 1]
 [0 0 1 0 0]
 [0 0 1 0 0]]
[[0 0 1 0 0]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [0 0 1 0 0]]
```

3. โหลดภาพจาก <https://assets.gezdev.com/dip/circuit.png> แปลงเป็น GrayScale ถัดจากนั้นทำให้ภาพเป็น Binary และ plot ภาพออกมา

3.1. รหัสต้นฉบับ และผลลัพธ์

```
src = cv.imread("./circuit.png")
gray_img = cv.cvtColor(src,cv.COLOR_BGR2GRAY)
_,bin_img = cv.threshold(gray_img,127,255,cv.THRESH_BINARY)
plt.imshow(bin_img,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
```



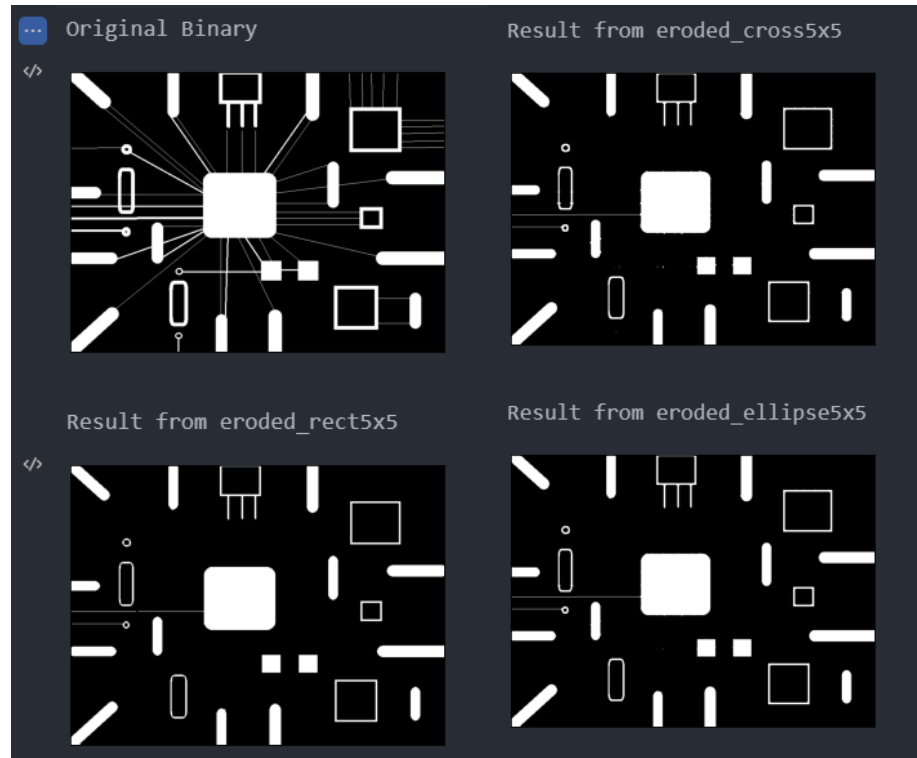
4. ทดสอบ Erode ภาพด้วย Kernel rect5x5, cross5x5, ellipse5x5 และแสดงผลลัพธ์

4.1. รหัสต้นฉบับ

```
show_erodirect5x5= cv.erode(bin_img,rect5x5,iterations=1)
show_erodecross3x3 = cv.erode(bin_img,cross5x5,iterations=1)
show_ellipse5x5 = cv.erode(bin_img,ellipse5x5,iterations=1)

print("Original Binary")
plt.imshow(bin_img,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
print("Result from eroded_rect5x5")
plt.imshow(show_erodirect5x5,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
print("Result from eroded_cross5x5")
plt.imshow(show_erodecross3x3,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
print("Result from eroded_ellipse5x5")
plt.imshow(show_ellipse5x5,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
```

4.2. ผลลัพธ์



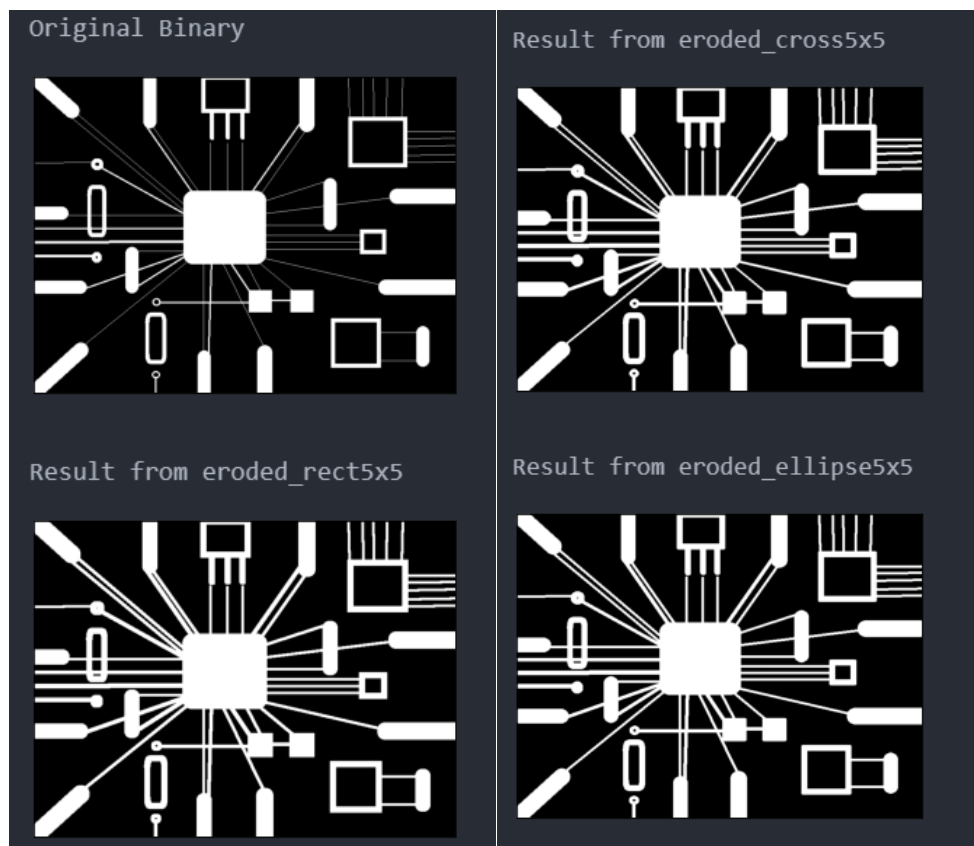
5. ทดสอบ Dilate ภาพด้วย Kernel rect5x5, cross5x5, ellipse5x5 และแสดงผลลัพธ์

5.1. รหัสต้นฉบับ

```
dilate_rect5x5= cv.dilate(bin_img,rect5x5,iterations=1)
dilate_cross3x3 = cv.dilate(bin_img,cross5x5,iterations=1)
dilate_ellipse5x5 = cv.dilate(bin_img,ellipse5x5,iterations=1)

print("Original Binary")
plt.imshow(bin_img,cmap='gray'),plt.xticks([],plt.yticks([]))
plt.show()
print("Result from eroded_rect5x5")
plt.imshow(dilate_rect5x5,cmap='gray'),plt.xticks([],plt.yticks([]))
plt.show()
print("Result from eroded_cross5x5")
plt.imshow(dilate_cross3x3,cmap='gray'),plt.xticks([],plt.yticks([]))
plt.show()
print("Result from eroded_ellipse5x5")
plt.imshow(dilate_ellipse5x5,cmap='gray'),plt.xticks([],plt.yticks([]))
plt.show()
```

5.2. ผลลัพธ์



6. ทำตามคำสั่ง 6.1 6.2 โดยใช้การ Dilate Erode Opening Closing โดยสามารถเขียนแบบcodeในข้อ 4 และ 5

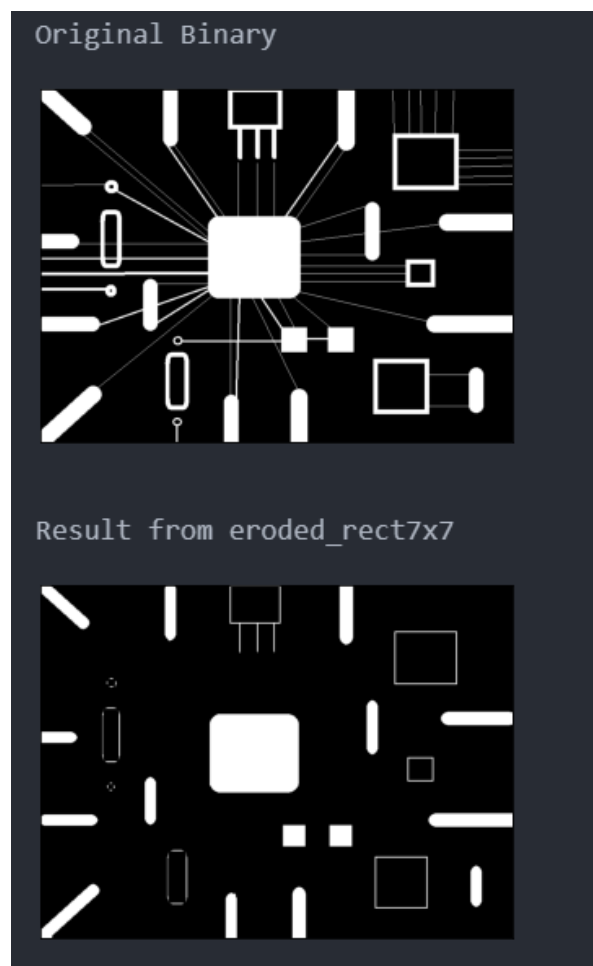
6.1. ให้นำเส้นสายสัญญาณทั้งหมดออกให้เหลือแค่ Component

6.1.1. รหัสต้นฉบับ

```
# 1. ให้นำเส้นสายสัญญาณทั้งหมดออกให้เหลือแค่ Component
rect7x7 = cv.getStructuringElement(cv.MORPH_RECT,(7,7))
show_eroderect7x7= cv.erode(bin_img,rect7x7,iterations=1)
print("Original Binary")
plt.imshow(bin_img,cmap='gray'),plt.xticks([],plt.yticks([]))
plt.show()
print("Result from eroded_rect7x7")
plt.imshow(show_eroderect7x7,cmap='gray'),plt.xticks([],plt.yticks([]))
plt.show()
```

✓ 0.2s

6.1.2. ผลลัพธ์



6.2. ให้เพิ่มขนาดสายสัญญาณทุกเส้น

6.2.1. รหัสต้นฉบับ

```
# 2. ให้เพิ่มขนาดสายสัญญาณ
print("Original Binary")
plt.imshow(bin_img,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
print("Result from eroded_rect5x5")
plt.imshow(dilate_rect5x5,cmap='gray'),plt.xticks([]),plt.yticks([])
plt.show()
```

✓ 0.1s

6.2.2. ผลลัพธ์

