

Thresholding

Thresholding

เป็นการกำหนดเกณฑ์ เพื่อแปลงภาพต้นฉบับเพื่อทำให้สามารถนำภาพไปประมวลผลต่อได้ เช่น การทำ contour การทำ edge-detection เพื่อตรวจจับวัตถุต่างๆ



```
1 import cv2 as cv
2 import matplotlib.pyplot as plt
3 import numpy as np
4 img = cv.imread("image/0d71318976feb4c9c85be3a20a84ea02.jpg")
5 img  = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
6
7 x,y=img.shape[:2]
8 imgth = np.zeros([x,y],np.uint8)
9
10 for i in range(x) :
11     for j in range(y):
12         if img[i,j] < 128:
13             imgth[i,j] = 0
14         elif img[i,j] >= 128:
15             imgth[i,j] = 1
16 plt.imshow(imgth,cmap="gray")
17 plt.show()
```

Thresholding

จะเห็นได้ว่า จะเป็นการแปลง grayscale ให้เป็นภาพ Binary ซึ่งค่าสีอยู่เพียง 2 ค่า คือ 0 และ 1 โดยประโยชน์ของ Thresholding จะเป็น ระบุตำแหน่งของวัตถุที่ต้องการในการการแพทย์ อุตสาหกรรมแยกสินค้า เป็นต้น

Thresholding

ໃນ open cv ການກຳ thresholding ຈະມີຄໍາສັ່ງດື່ວ

`cv.threshold(img, threshold_value, maxval, type_threshold)`

`Img` ດື່ວ ຮູບ grayscale

`threshold_value` ດື່ວ ຜ່າເກນທີ່ໃຊ້ໃນການແປ່ງໜ່ວງ

`Maxval` ດື່ວ ຜ່າສູງສຸດ

`type_threshold` ດື່ວ ຮູບແບບຂອງການກຳ threshold

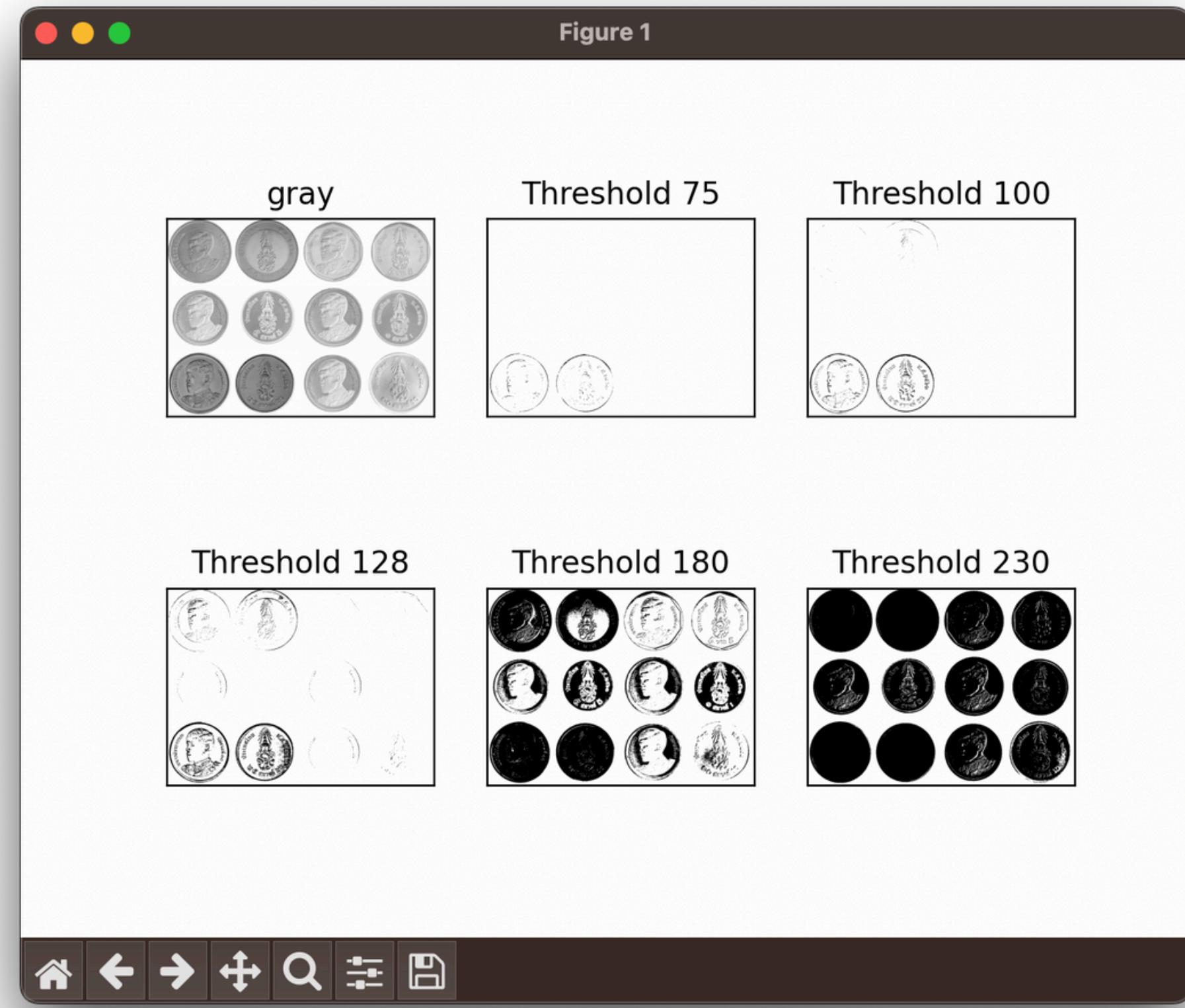


```
1 thresh_value = [75,100,128,180,230]
```



```
1 for i in range(len(thresh_value)):
2     thresh,result = cv.threshold(gray_img,thresh_value[i],255,cv.THRESH_BINARY)
3     plt.subplot(232+i,title="Threshold %d"%(thresh_value[i]),xticks=[],yticks[])
4     plt.imshow(result,cmap="gray")
```

Figure 1



ឧបແບບការកំ Thresholding

รูปแบบการทำ Thresholding

- การทำ thresholding นั้นมีหลายรูปแบบเนื่องจากรูปภาพแต่ละภาพที่นำเข้ามา มีความแตกต่างกันโดยจะมีทั้งหมด 6 รูปแบบดื้อ

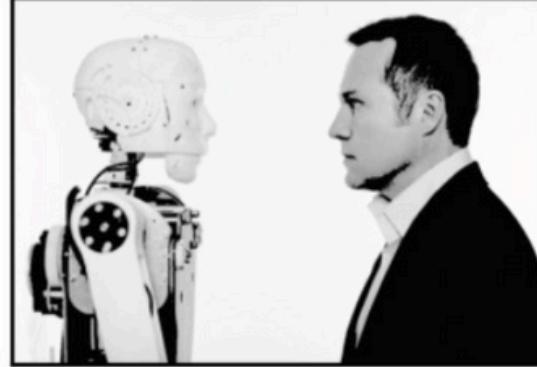
รูปแบบการทำ Thresholding

Type_threshold	ค่า	ความหมาย	Type_threshold
cv2.THRESH_BINARY	0	โดยค่าที่สูงกว่าเกณฑ์ threshold Value ที่กำหนดให้เปลี่ยน 255 สว่างสูงสุดคือสีขาว ต่ำกว่าเกณฑ์เป็น 0 คือ สีดำ	threshold binary 
cv2.THRESH_BINARY_INV	1	โดยค่าที่สูงกว่าเกณฑ์ threshold Value ที่กำหนดให้เปลี่ยน 0 สีดำ และต่ำกว่าเกณฑ์เป็น 255 คือ สีขาว	threshold binary_inv 

รูปแบบการทำ Thresholding

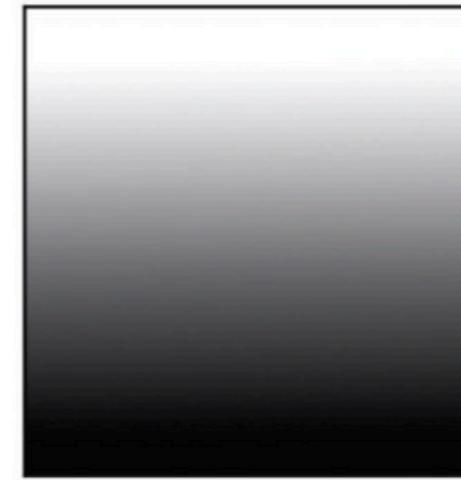
Type_threshold	ค่า	ความหมาย	Type_threshold
cv2.THRESH_TRUNC		โดยค่าที่สูงกว่าเกณฑ์ จะถูกเปลี่ยน และค่าที่ต่ำกว่ามีค่าคงเดิม หมายความว่า ค่าความสว่าง 255 ถึงเกณฑ์ให้เท่ากับเกณฑ์ ค่าต่ำกว่าเกณฑ์ให้มีค่าเท่าเดิม	threshold trunc 

รูปแบบการทำ Thresholding

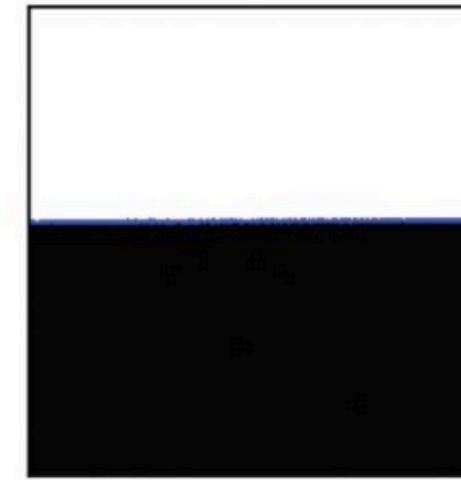
Type_threshold	ค่า	ความหมาย	Type_threshold
cv2.THRESH_TOZERO	1	ถ้าต่ำกว่าค่าจุดเปลี่ยนจะเป็น 0 หรือสีดำ ค่าที่มากกว่าจุดเปลี่ยนเหลือคงเดิม หมายความว่า ถ้าน้อยกว่า 255 ถึงก็จะให้ คงค่าเดิมค่าที่ต่ำกว่าให้เป็นสีดำ	threshold tozero 
cv2.THRESH_TOZERO_INV		ถ้าสูงกว่าค่าจุดเปลี่ยนจะเป็น 0 สีดำ ที่เหลือคงเดิม	threshold tozero inv 

សម្រាប់ការទាំ Thresholding

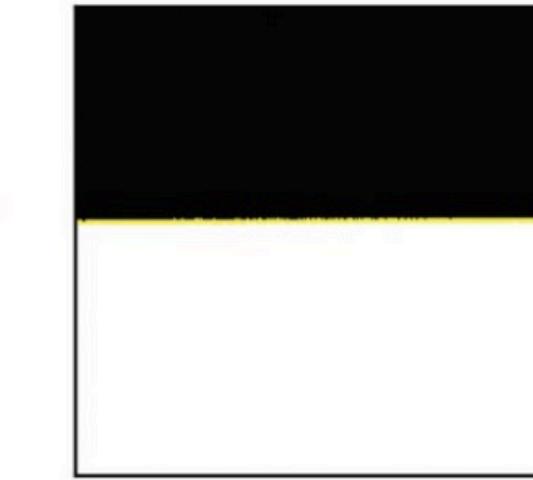
origin



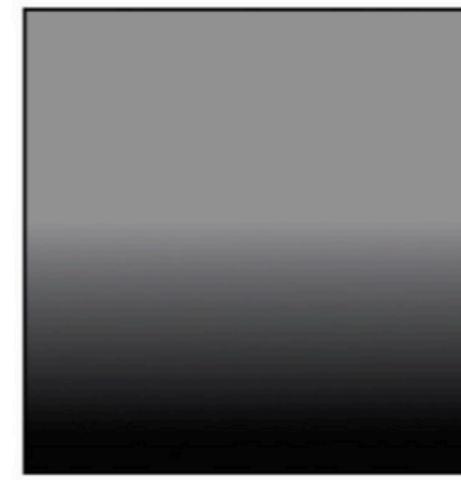
threshold binary



threshold binary_inv



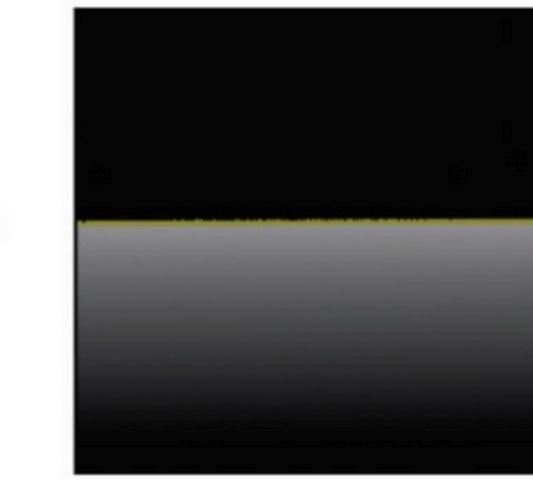
threshold trunc



threshold tozero



threshold tozero_inv



รูปแบบการทำ Thresholding

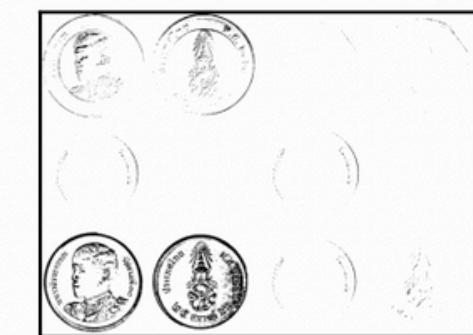
Type_threshold	ค่า	ความหมาย	Type_threshold
cv2.THRESH_OTSU	1	ทำการคำนวณค่าจุดเปลี่ยนอัตโนมัติ ไม่ใช้ค่า thresh ที่ใส่เข้าไป	threshold OTSU 

Figure 1

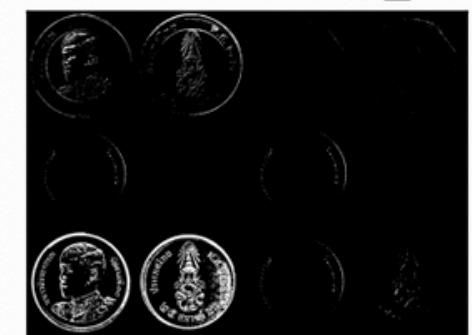
original



Thresholdbinary



thresholdbinary_inv



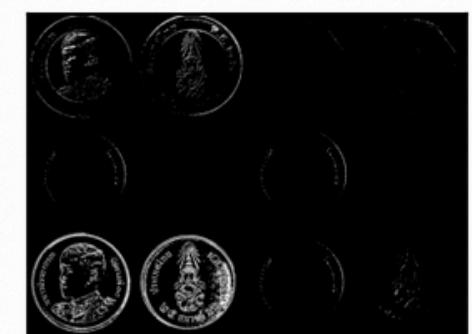
thresholdtrunc



thresholdtozero



thresholdtozeroinv



Adaptive Thresholding

AdaptiveThresholding

- AdaptiveThresholding จะทำมาเพื่อแก้ปัญหาภาพที่มีความสว่างไม่เท่ากันทุกจุด วิธีมีผังชั้น AdaptiveThreshold เพื่อมาแก้ปัญหา โดยจะใช้คำสั่ง `cv.adaptiveThreshold()`

Adaptive Thresholding

cv.adaptivethreshold(img,maxvalue, adaptiveMethod, thresholdtype,blocksize,c)

Img គឺ រូបភាព grayscale

Maxvalue គឺ តម្លៃកំណត់សុខសុទ្ធសាស្ត្រ

adaptiveMethod គឺ រូបរាងនៃ adaptivethreshold

Thresholdtype គឺ រូបរាងនៃការធានា threshold

blocksize គឺ ឈាននៃបន្ទីកដែលបានប្រើប្រាស់ដើម្បីធានាបានពីរតាមចំណាំ

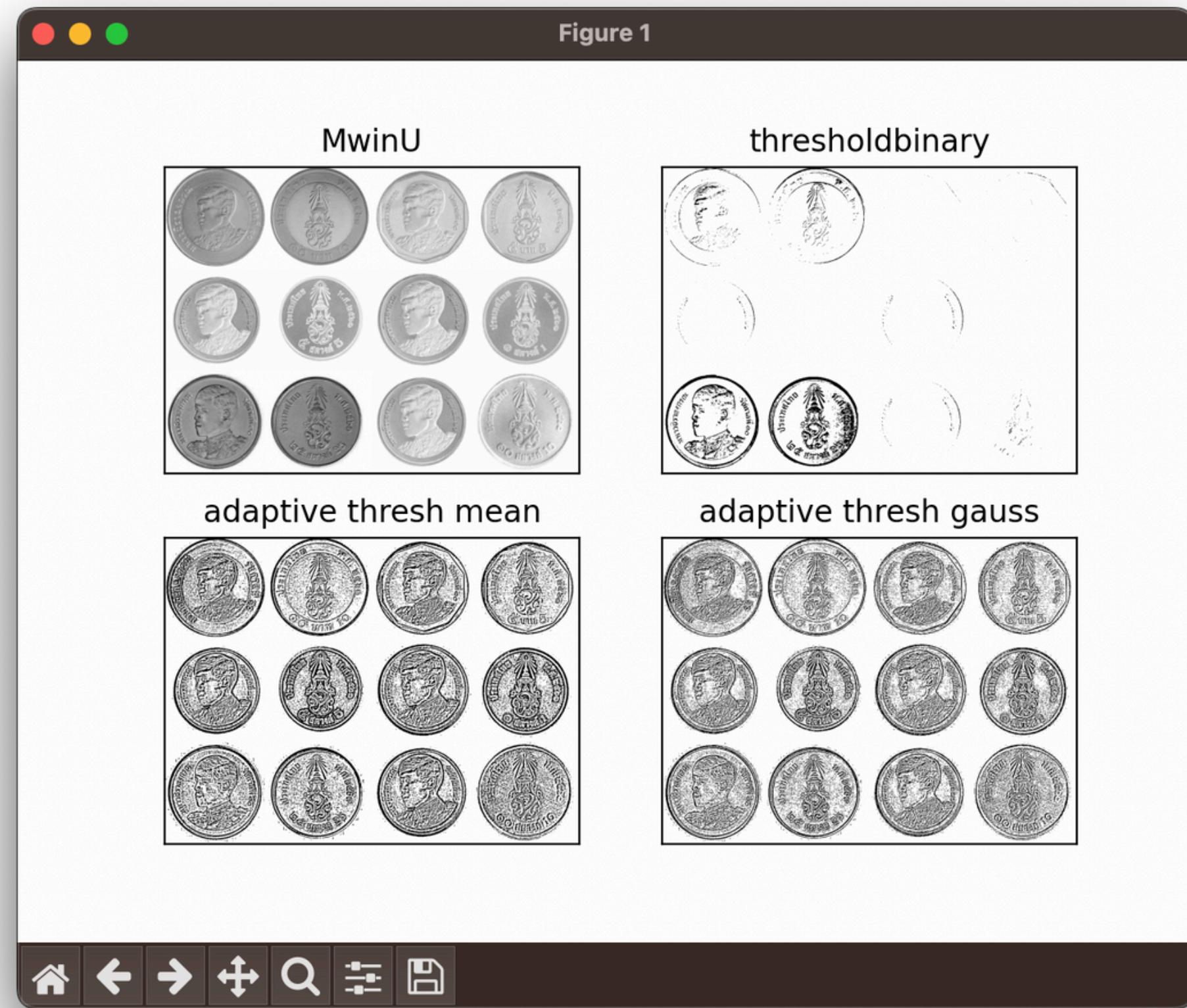
c ជា c

Adaptive Thresholding

Adaptive Thresholding	ค่า	ความหมาย	ตัวอย่างภาพ
<code>cv2.ADAPTIVE_THRESH_MEAN_C</code>	0	คำนวณค่าเฉลี่ยรอบข้างขนาด <code>blockSize * blockSize</code> แล้วหักจากค่า C	 thresh mean
<code>cv2.ADAPTIVE_THRESH_GAUSSIAN_C</code>	1	คำนวณค่าเฉลี่ยรอบข้างขนาด <code>blockSize * blockSize</code> โดยถ่วงน้ำหนักด้วยฟังก์ชัน <code>gaussian</code> แล้วหักจากค่า C	 thresh gauss

```
● ● ●
1 import cv2 as cv
2 import matplotlib.pyplot as plt
3
4 img = cv.imread("image/0d71318976feb4c9c85be3a20a84ea02.jpg")
5 gray_img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
6
7 plt.subplot(221,xticks=[],yticks[])
8 plt.title('MwinU',family='PSL-omyim')
9 plt.imshow(gray_img,cmap='gray')
10
11 thresh,result0=cv.threshold(gray_img,128,255, cv.THRESH_BINARY)
12 plt.subplot(222,title='thresholdbinary',xticks=[],yticks[])
13 plt.imshow(result0,cmap='gray')
14
15 result1=cv.adaptiveThreshold(gray_img,255,cv.ADAPTIVE_THRESH_MEAN_C, cv.THRESH_BINARY,9,3)
16 plt.subplot(223, title='adaptive thresh mean',xticks=[],yticks[])
17 plt.imshow(result1,cmap='gray')
18
19 result2= cv.adaptiveThreshold(gray_img,255, cv.ADAPTIVE_THRESH_GAUSSIAN_C, cv.THRESH_BINARY,9,3)
20 plt.subplot(224, title='adaptive thresh gauss',xticks=[],yticks[])
21 plt.imshow(result2,cmap='gray')
22
23 plt.show()
```

Figure 1



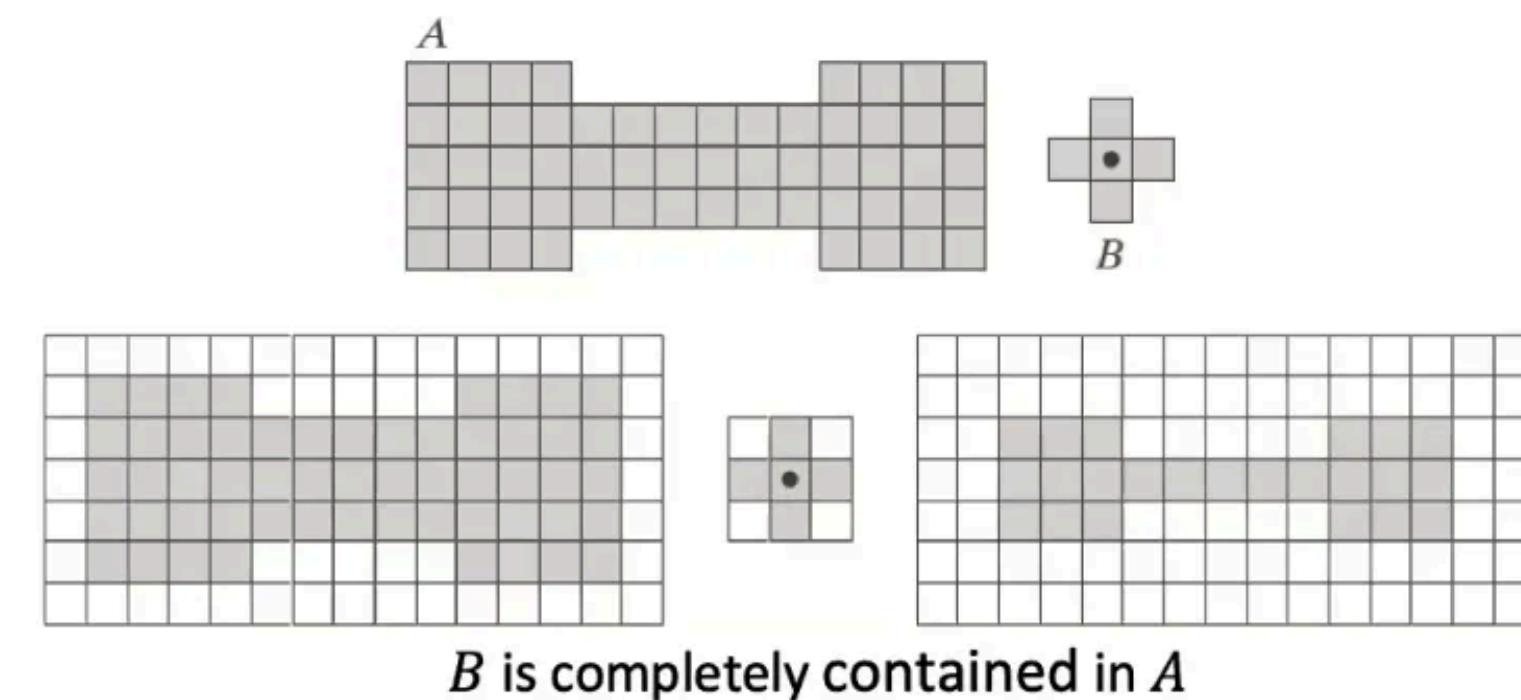
MORPHOLOGICAL

การปรับปรุงรูปร่างหรือโครงสร้างด้วยการใช้ MORPHOLOGICAL

Morphological Image Processing ถือเป็นหนึ่งในรูปแบบการประมวลผลรูปภาพดิจิตอล ซึ่งมีเป้าหมายในการดึงองค์ประกอบและคุณลักษณะต่างๆ ของรูปภาพที่เราต้องการ

การปรับปรุงรูปร่างหรือโครงสร้างด้วยการใช้ MORPHOLOGICAL

Morphology แบบหนึ่งเราเรียกว่า erosion คือการนำ Structuring Elements ข้ามมาวางเทียบกับ Binary Image ในรูปแบบของ Sliding window คือการนำ SE วิ่งจากซ้ายไปขวาทีละ pixel



ซึ่งฟังก์ชันในการ morphological มีทั้งหมด 7 รูปแบบ แต่จะเรียนรู้กันเพียง 5 รูปแบบดังนี้

รูปแบบ morphological และความหมาย	คำสั่งและรายละเอียด	ภาพ input	ภาพ output
<p>1. การกร่อนภาพ (Erosion) คือ การกัดหรือลดขนาดของรูปร่าง ของภาพ โดยถ้าภาพ input - ถ้าเป็น gray จะลดภาพและ ลดความสว่าง - ถ้าเป็น binary จะลดขนาด</p>	<p>erosion = cv2.erode (img, kernel, iterations=1)</p> <p>1. img คือ ภาพ array 3 มิติ 2. kernel คือ ชนิดรูปร่าง 2 มิติ ที่ใช้กรองภาพ เช่น kernel=np.ones((5,5),np.uint8) คือ ขนาด 5*5 3. iteration คือ จำนวนรอบที่ให้ทำการกรอง ภาพ</p>		

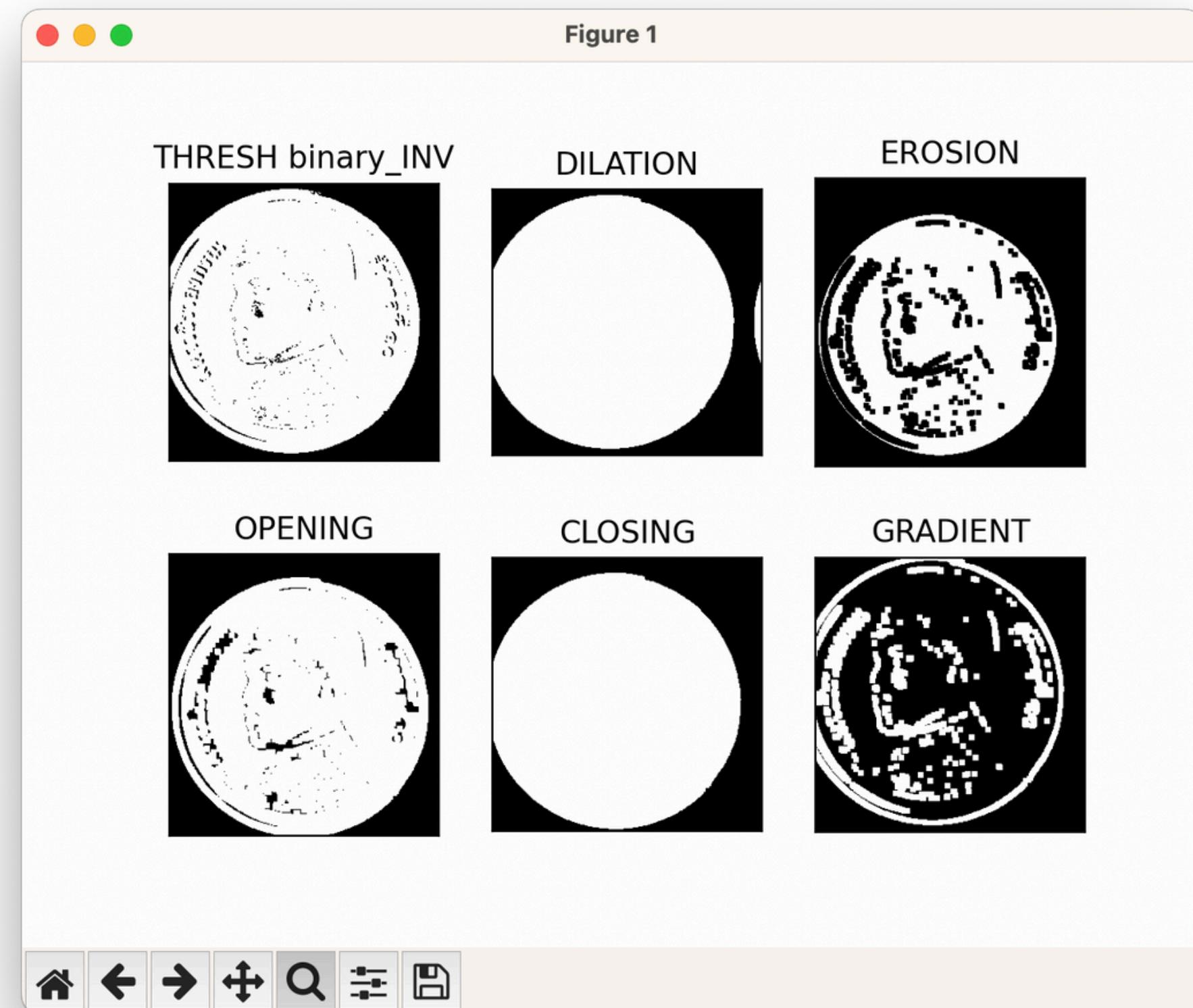
รูปแบบ morphological และความหมาย	คำสั่งและรายละเอียด	ภาพ input	ภาพ output
<p>2.การขยายภาพ (dilation) คือ การขยายขนาดของรูปร่างของภาพ โดยถ้าภาพ input - ถ้าเป็น gray จะเพิ่มภาพและเพิ่มความสว่าง - ถ้าเป็น binary จะเพิ่มขนาด</p>	<p>dilation = cv2.dilate (img, kernel, iterations=1)</p> <ol style="list-style-type: none"> img คือ ภาพ array 3 มิติ kernel คือ ชนิดรูปร่าง 2 มิติ ที่ใช้กรองภาพ เช่น kernel=np.ones((5,5),np.uint8) คือ ขนาด 5*5 iteration คือ จำนวนรอบที่ให้ทำการกรองภาพ 		
<p>3.การเปิดภาพ (opening) คือ การลดสิ่งที่เป็นการรบกวนภาพออกจากบริเวณที่เป็นสีดำหรือเป็นการเชื่อมส่วนสีดำ</p>	<p>opening = cv2.morphologyEx (img, cv2.MORPH_OPEN, kernel, iterations)</p> <ol style="list-style-type: none"> img คือ ภาพ array 3 มิติ kernel คือ ชนิดรูปร่าง 2 มิติ ที่ใช้กรองภาพ เช่น kernel=np.ones((5,5),np.uint8) คือ ขนาด 5*5 iteration คือ จำนวนรอบที่ให้ทำการกรองภาพ 		

รูปแบบ morphological และความหมาย	คำสั่งและรายละเอียด	ภาพ input	ภาพ output
<p>4.การปิดภาพ (Closing) คือ การทำตรงกันข้ามกับเปิดภาพ คือการเชื่อมส่วนสีขาว</p>	<pre>closing = cv2.morphologyEx (img, cv2.MORPH_CLOSE, kernel, iterations) 1.img คือ ภาพ array 3 มิติ 2.kernel คือ ชนิดรูปร่าง 2 มิติ ที่ใช้กรองภาพ เช่น kernel=np.ones((5,5),np.uint8) คือ ขนาด 5*5 3.iteration คือ จำนวนรอบที่ให้ทำการกรองภาพ</pre>		
<p>5.ส่วนการขยายตัวลบส่วนที่ร่อน (Gradient) คือ การทำให้การขยาย dilate และ นำมาลบด้วยการทำ erode จะได้ ภาพคล้ายโครงของวัตถุ</p>	<pre>gradient = cv2.morphologyEx (img, cv2.MORPH_GRADIENT, kernel, iterations) 1.img คือ ภาพ array 3 มิติ 2.kernel คือ ชนิดรูปร่าง 2 มิติ ที่ใช้กรองภาพ เช่น kernel=np.ones((5,5),np.uint8) คือ ขนาด 5*5 3.iteration คือ จำนวนรอบที่ให้ทำการกรองภาพ</pre>		

การใช้งานคำสั่ง

```
● ● ●  
1 thresh,result = cv.threshold(img,180,255,cv.THRESH_BINARY_INV)  
2 kernel = np.ones((2,2),np.uint8)  
3  
4 dilation=cv.dilate(result,kernel,iterations=5)  
5 erosion=cv.erode(result,kernel,iterations=5)  
6 opening=cv.morphologyEx(result,cv.MORPH_OPEN,kernel,iterations=5)  
7 closing=cv.morphologyEx(result,cv.MORPH_CLOSE,kernel,iterations=5)  
8 gradient=cv.morphologyEx(result,cv.MORPH_GRADIENT,kernel,iterations=5)
```

ผลลัพธ์

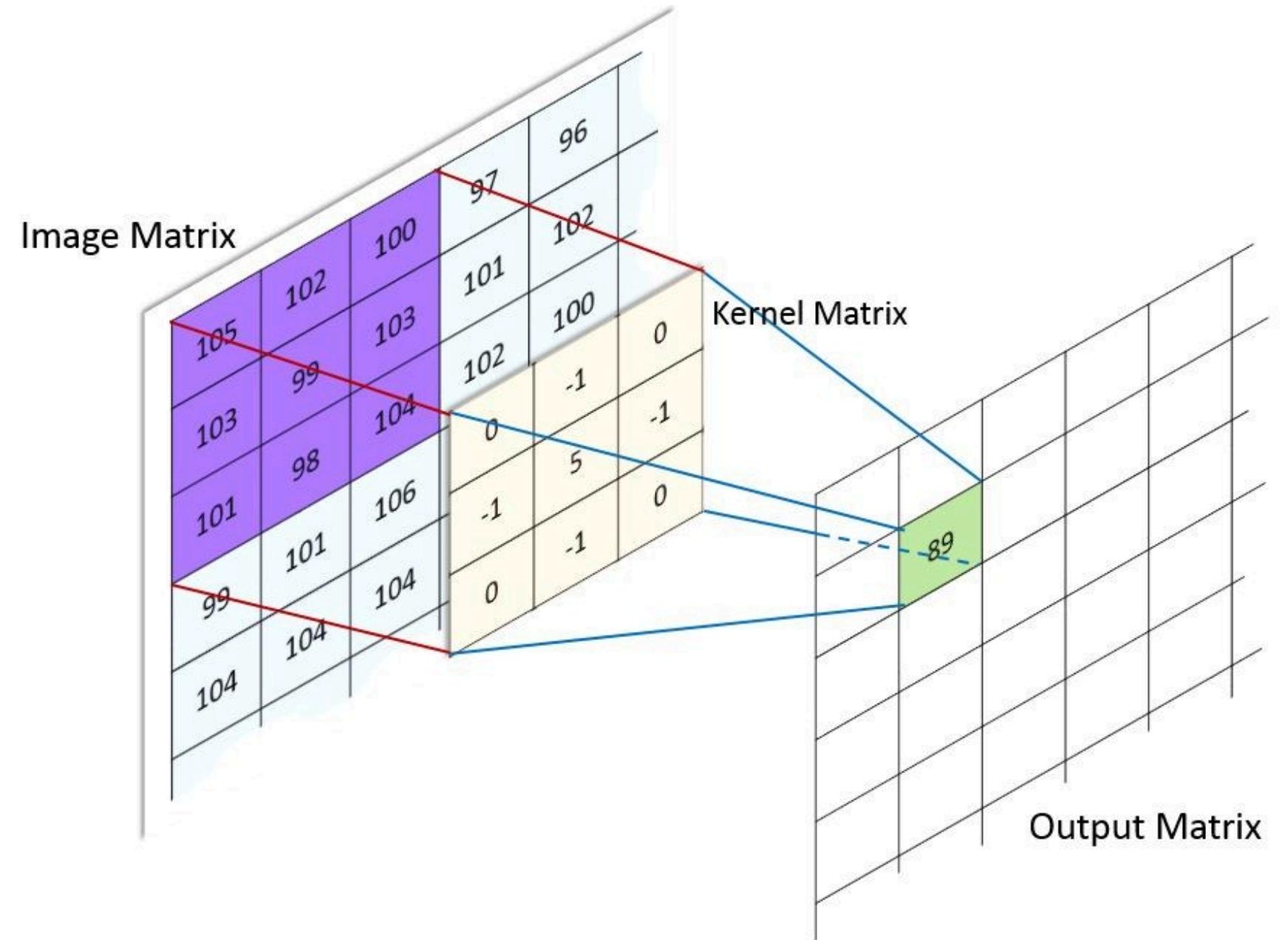


CONVOLUTION

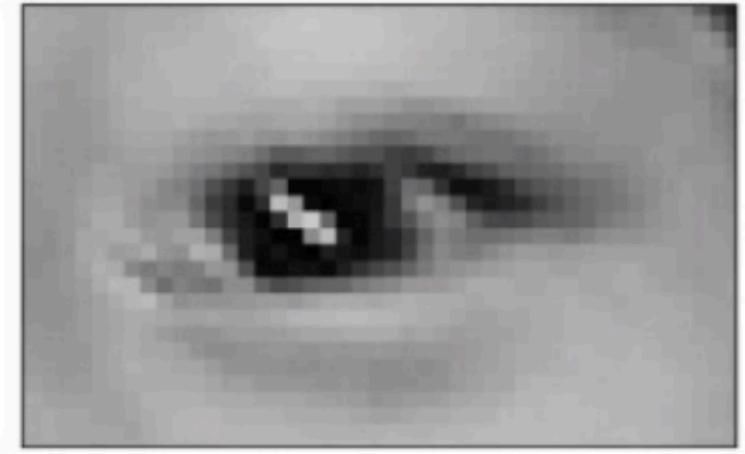
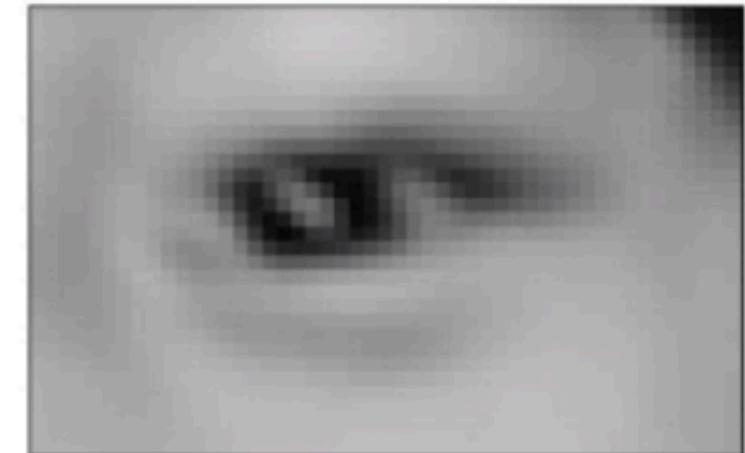
CONVOLUTION

การทำ Convolution ใน OpenCV คือการนำเอา "ตัวกรอง" ซึ่งอาจเรียกว่า เดอร์เบา(kernel) 2 มิติมาวิ่งผ่านภาพและทำการ ประมวลผลแต่ละจุดpixelของภาพ Input เพื่อทำให้ภาพนั้นมี ดุลภาพที่ดีขึ้นหรือทำภาพให้ เนียนขึ้น (Smoothing Images) เพื่อจะสามารถนำไปใช้ประโยชน์ในการทำ image processing ได้ดีขึ้น

CONVOLUTION



โดยพังก์ชันในการ Convolution มีทั้งหมด 5 รูปแบบ แต่จะเรียนรู้กันเพียง 4 รูปแบบดังนี้

รูปแบบ Convolution และความหมาย	คำสั่งและรายละเอียด	ภาพ input และ ภาพหลังทำ Convolution
<p>1.filter2D คือ การกรองภาพ 2 มิติ ไปริ่งผ่านภาพ</p> $K = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ <p>ตัวกรองแบบ 3x3</p>	<p>filter2D = cv2.filter2D(img, ddepth, kernel)</p> <p>1.img คือ ภาพ array 3 มิติ grayscale 2.ddepth คือ ชนิดตัวแปรในอาร์เรย์ของผลลัพธ์ที่จะได้ออกมาจากตัวกรอง โดยทั่วไปจะใส่เป็น -1 ซึ่งหมายถึง ให้ออกมาเป็นชนิดเดียวกับ array รูปภาพ input 3.kernel คือ ชนิดรูปร่าง 2 มิติ ที่ใช้กรองภาพ เช่น kernel=numpy.ones((3,3),np.float32)/9 kernel=numpy.ones((3,3),np.float32)/25</p>	<p>ORIGINAL</p>  <p>CONVOLUTION 3*3</p> 

รูปแบบ Convolution และความหมาย

2. ตัวกรองค่าเฉลี่ยแบบ เบลอ (Mean Filtering)

$$K = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

ตัวกรองแบบ 5x5

คำสั่งและรายละเอียด

mean = cv2.blur(img, ksize)

1. **img** คือ ภาพ array 3 มิติ grayscale
2. **ddepth** คือ ขนาด kernel ซึ่งสามารถระบุได้
ดังนี้ (3,3) หรือ (5,5)

หมายเหตุ หากใช้ค่าตัวกรองมากภาพจะเบลอมาก

ภาพ input และ ภาพหลังทำ Convolution



รูปแบบ Convolution และความหมาย

3. ตัวกรองค่ามัธยฐาน
(Median Filter)
คือ การกรองที่ใช้ค่า
มัธยฐานหรือค่าที่อยู่
กึ่งกลางเป็นเกณฑ์ในการ
กรองค่า ซึ่งจะสามารถลด
ความเบลอของขอบภาพ
แต่ไม่ได้เพิ่มความเบลอ
ของภาพ

คำสั่งและรายละเอียด

median = cv2.medianblur(img, ksize)

1. **img** คือ ภาพ array 3 มิติ grayscale
2. **ddepth** คือ ขนาด kernel ซึ่งสามารถระบุได้
ดังนี้ (3,3) หรือ (5,5)

หมายเหตุ จากภาพจะเห็นได้ว่า ภาพที่ได้จาก
ฟังก์ชัน median blur ไม่ได้เบลอเหมือนที่ใช้
mean blur

ภาพ input และ ภาพหลังทำ Convolution



รูปแบบ Convolution และความหมาย

4.ตัวกรองแบบเกาส์เซียน (Gaussian Filter)

เป็นฟังก์ชันที่จะมีค่าเป็นขนาดใหญ่ที่สุดตรงกลาง และเล็กลงเรื่อยๆ เมื่อห่างจากใจกลางไป

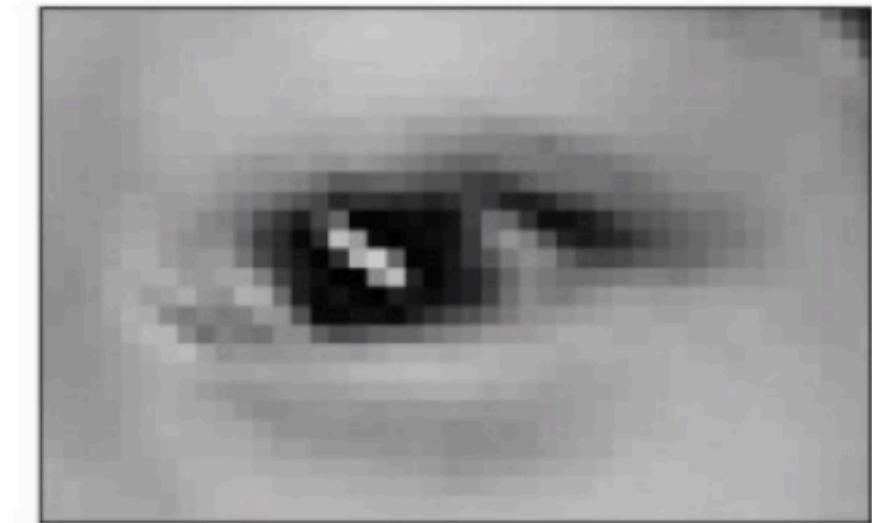
คำสั่งและรายละเอียด

gaussian = cv2.GaussianBlur(img, ksize, sigma)

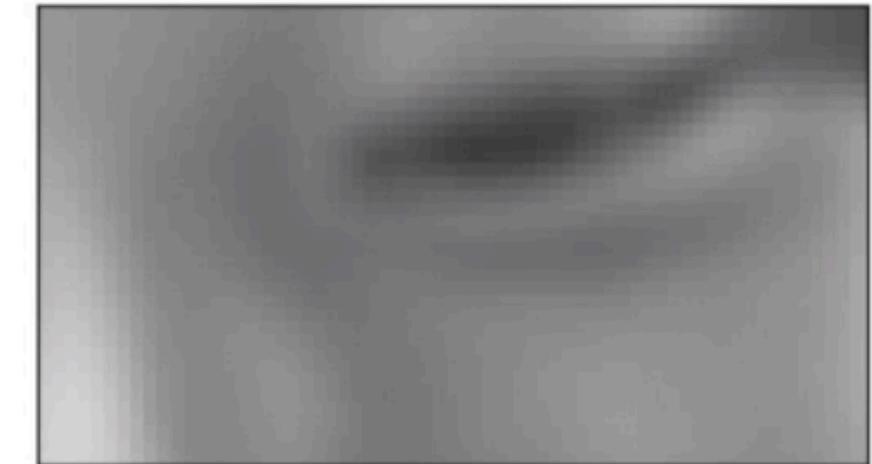
- 1.**img** คือ ภาพ array 3 มิติ grayscale
- 2.**ddepth** คือ ขนาด kernel ซึ่งสามารถระบุได้ดังนี้ (3,3) หรือ (5,5)
- 3.**sigma** คือ ยิ่งมีค่าเยอะ ยิ่งใกล้เคียงค่าใน blur() ซึ่งปกติจะตั้งไว้ที่ 0

ภาพ input และภาพหลังทำ Convolution

ORIGINAL



GAUSSION BLUR



GAUSSIAN FILTER

1	2	1
2	4	2
1	2	1

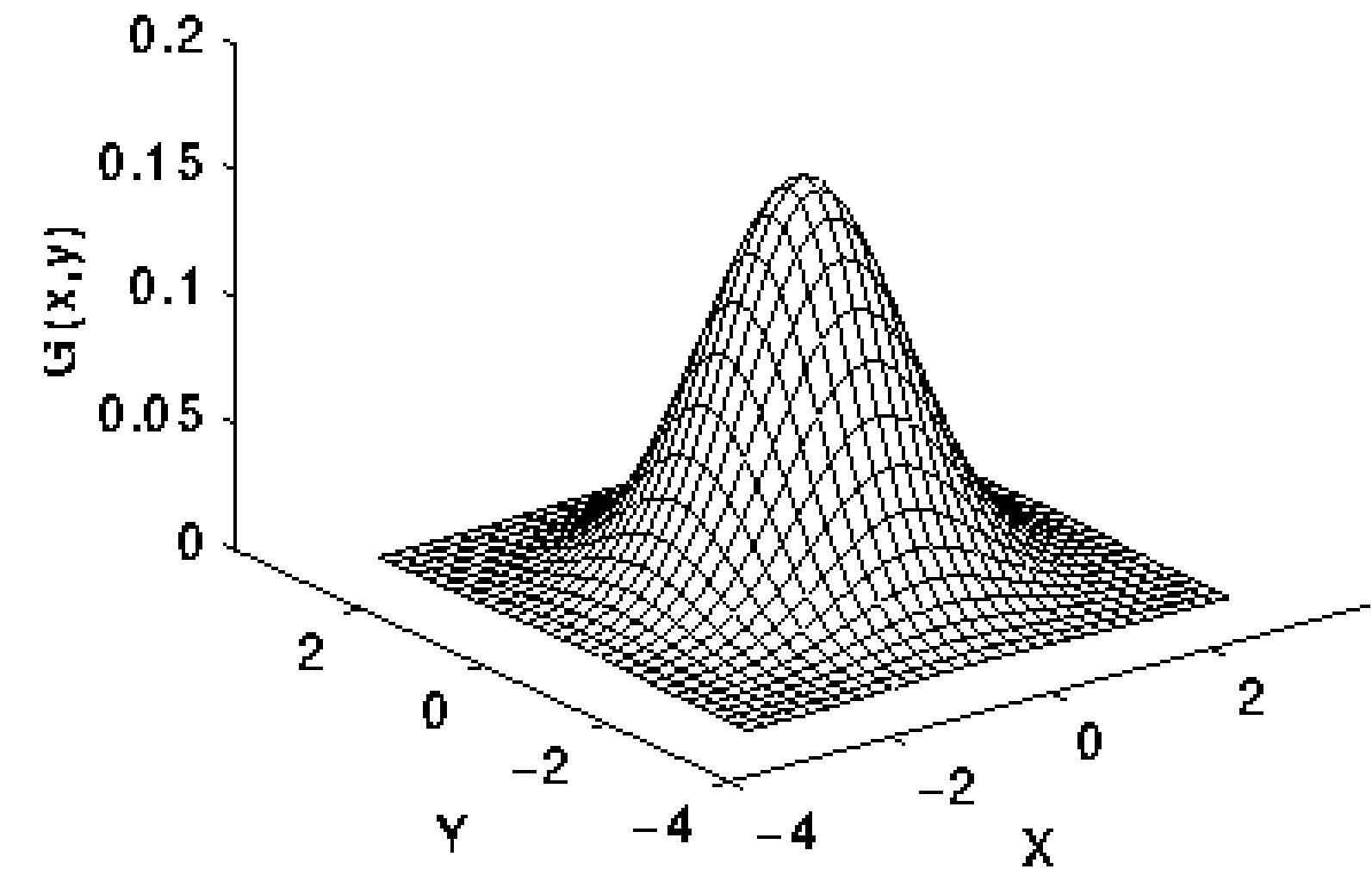
1/16

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

1/273

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0

1/1003



คำสั่ง

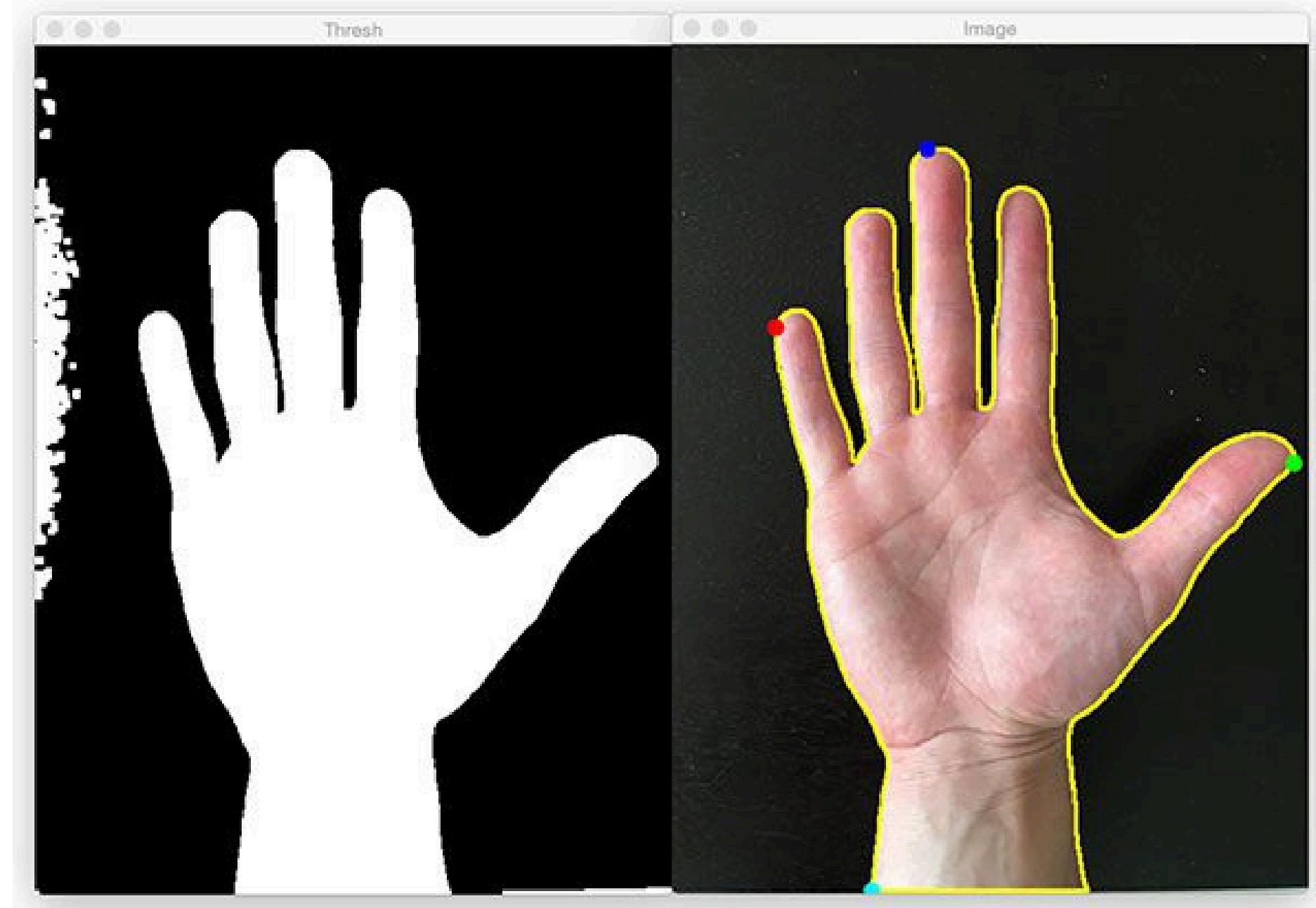


```
1 kernel=np.ones((3,3),np.float32)/9
2 kernel2=np.ones((5,5),np.float32)/25
3
4 filter2D=cv.filter2D(img,-1,kernel)
5 mean=cv.blur(img,(5,5))
6 median_blur=cv.medianBlur(img,5)
7 gaussian_blur=cv.GaussianBlur(img,(5,5),10)
```

CONTOUR

CONTOUR

เส้นเด้าโครง(contour) คือ
เส้นที่อยู่บริเวณภาพ binary ที่
เป็นสีดำ และสีขาว การสร้างเส้น
นี้ขึ้น มา มีความสำคัญ เพราะ
สามารถนำไปกำหนดจุดคำนวณ
พื้นที่และอะไรได้อีกมากมาย



วิธีการสร้างเส้น Contour

วิธีการสร้างเส้น Contour มีหลักการทำงานด้วยกัน 2 ขั้นตอน ดังนี้
คำสั่งการหาเส้นเด้าໂຄรงของรูปภาพ cv2.findContours() โดยตัวแปรที่กำหนดมี 2 ตัวແປรด້ອ
ຕัวແປรที่ 1 (contour) គື່ອຂ້ອມລຸ ເດັ່ນcontour
ຊື່ສາມາດຕຽວຈັບວ່າມີເສັນໄດ້ກັ້ງຮມດກີ່ເສັນໂດຍໃຊ້คำສັ່ງ print(len(contour))
ຕัวແປรທີ່ 2 (hierarchy) ຄື່ອຂ້ອມນູລລຳດັບຫຸ້ນ

วิธีการสร้างเส้น Contour

contour, hierarchy = cv2.findContours(img, mode, method)

1. **img** คือ ตัวแปรภาพข้อมูล array 3 มิติ (ใช้ภาพที่เป็น grayscale)
2. **mode** คือ รูปแบบการพิจารณาลำดับชั้นของเส้น contour flag (int)
3. **method** คือ รูปแบบในการหาเส้นเด้าໂຄ戎 flag(int)

โดยรูปแบบ mode ในการพิจารณาเส้น contour มีด้วยกัน 4 รูปแบบ

mode ในการหา contour	ค่า	ความหมาย
cv2.RETR_EXTERNAL	0	พิจารณาเฉพาะวงนอกสุด
cv2.RETR_LIST	2	พิจารณาทุกวงรอบโดยไม่มีการสร้างเป็นลำดับชั้น
CV2.RETR_CCOMP	3	พิจารณาทุกวงโดยมีการสร้างเป็นลำดับชั้นแต่ไม่ลึกลงไปกว่าสองชั้น
CV2.RETR_TREE*	4	เอาทุกวงโดยพิจารณาลำดับชั้นให้ตัวข้างในเป็นเค้าโครงลูกของตัวนอกໄลไปเรื่อยๆ

รูปแบบ method ในการพิจารณาเส้น contour มีด้วยกัน 2 รูปแบบดังนี้

method ในการหา contour	ความหมาย
<code>cv2.CHAIN_APPROX_NONE*</code>	แบบละเอียด
<code>cv2.CHAIN_APPROX_SIMPLE</code>	แบบไม่ละเอียด

วิธีการสร้างเส้น Contour

การวาดเส้นเด้าโครงลับภาพ cv2.DrawContours()

drawcontour = cv2.DrawContours(**img**, **contours**, **hierarchy**, **color**, **thickness**)

1. **img** คือ ตัวแปรภาพข้อมูล **array 3 มิติ** (ใช้ภาพที่เป็น grayscale)
2. **contours** คือ เส้น contour ที่จะวาดลงไปในภาพ ตัวแปรที่ได้จาก cv2.findContours
3. **hierarchy** คือ ลำดับชั้นของเส้น contour ที่จะวาด โดย -1 หมายถึงวดกั้งหมด (**int**)
4. **color** คือ สี (B,G,R) 0-255 (**tuple int**) 5. **thickness** คือ ความหนาของเส้น (**int**)

CODE

```
● ● ●  
1 import cv2 as cv  
2  
3 img=cv.imread('image/0d71318976feb4c9c85be3a20a84ea02.jpg')  
4  
5 gray_img=cv.cvtColor(img,cv.COLOR_BGR2GRAY)  
6  
7 thresh,result=cv.threshold(gray_img,168,255,cv.THRESH_BINARY)  
8 contours,hierarchy=cv.findContours(result,cv.RETR_TREE,cv.CHAIN_APPROX_NONE)  
9  
10 draw_contour=cv.drawContours(img,contours,-1,(0,255,0),2)  
11  
12 print(contours)  
13 print(len (contours))  
14  
15 cv.imshow("output",img)  
16 cv.waitKey(0)  
17 cv.destroyAllWindows()
```

ขนาดของ contour

- การหาขนาดของ contour

`Area = cv2.contourArea(contour)`

1. `Area` คือ พื้นที่ของ เส้น `contour`
2. `contour` คือ เส้น `contour` ที่จะวัดลงมาในภาพ ตัวแปรที่ได้จาก `cv2.findContours`

การประยุกต์ใช้ขนาดของ contour

การสร้างกรอบโดยใช้ contour

```
● ● ●  
1 for contour in contours:  
2     if cv.contourArea(contour) > 2000:  
3         x, y, w, h = cv.boundingRect(contour)  
4         cv.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 3)
```

การใช้ contour กับ color detect

นอกจาก contour จะสามารถใช้ได้กับ ภาพ binary image และ ก็ยังสามารถใช้งานได้กับ mask ของ color detect ได้อีกด้วย



```
1 mask_red = cv2.inRange(img, lower_red, upper_red)
2 mask_red_contours, hierarchy = cv2.findContours(mask_red, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```