

## Task 5 Morphological Image Processing

Identify 'O' in this picture

ponents or broken connection paths. There is no point past the level of detail required to identify those components. Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the effectiveness of computerized analysis procedures. For this reason, considerable effort must be taken to improve the probability of rugged segmentation. In applications such as industrial inspection applications, at least some degree of automation in the environment is possible at times. The experienced image processing designer invariably pays considerable attention to such details.

อันดับแรกลบตัวที่หางยาวทิ้งไปก่อน เพราะตัวหางยาวมีโอกาสที่จะเป็นตัว O (กลม) สูงเพิ่มที่จะตัดปัญหาก่อนที่จะเจออันถัดไป

```
ori_image = cv2.imread('image/text_frombook.png',0)

# จากคริปอาจารย์
def imreconstruct(img,marker):
    mask = img
    se = cv2.getStructuringElement(cv2.MORPH_RECT , (3,3))
    recon1 = marker
    recon1_old = np.zeros(recon1.shape , np.int8)
    while np.sum(np.sum(recon1 - recon1_old)) != 0 :
        recon1_old = recon1
        recon1 = cv2.dilate(recon1 , se)
        recon1 = recon1 & mask
    return recon1

se = cv2.getStructuringElement(cv2.MORPH_RECT , (1,51))
mark_img1 = cv2.erode(ori_image,se)

lt_img = imreconstruct(ori_image,mark_img1)

nolt_image = ori_image - lt_img
```

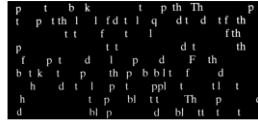
Original Image text\_frombook

ponents or broken connection paths. There is no point in going past the level of detail required to identify those segments. Segmentation of nontrivial images is one of the most difficult tasks in computer processing. Segmentation accuracy determines the effectiveness of computerized analysis procedures. For this reason, considerable attention must be taken to improve the probability of rugged segmentation. Such as industrial inspection applications, at least some of the time, the environment is possible at times. The experienced designer invariably pays considerable attention to such

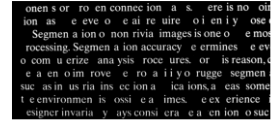
Mark only long tail



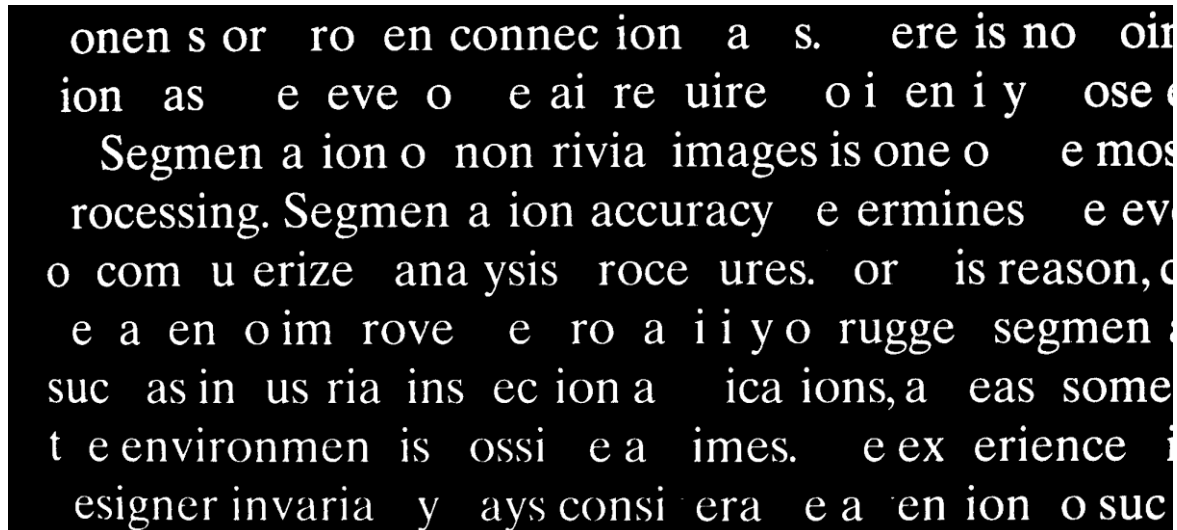
Long tail



No long tail



เราจะได้ภาพที่ไม่มีตัวหางยาวมาแล้ว



ถัดมาเราจะทำ Bitwise และเลือกเฉพาะตัวที่ กลมๆ ในการทำ Flood Fill และมา Merge รวมกัน

```
bw_image = cv2.bitwise_not(nolt_image)

h , w = bw_image.shape[:2]
mask = np.zeros((h+2,w+2) , np.uint8)

image_mark2 = bw_image.copy()
cv2.floodFill(image_mark2,mask,(0,0),0)

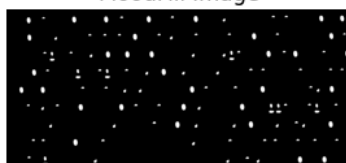
marge_image = nolt_image | image_mark2
```

จะได้รูปนี้

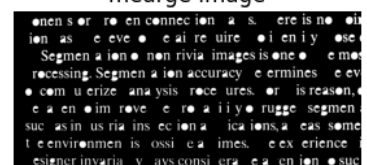
Bitwise Image

onen s or ro en connec ion a s. ere is no oit  
ion as e eve o e ai re uire oi en i y ose  
Segmen a ion o non rivia images is one o e mos  
rocessing. Segmen a ion accuracy e ermines e ev  
o com u erize ana ysis roce ures. or is reason,c  
e a en oim rove e ro a i i y o rugge segmen  
suc as in us ria ins ec ion a ica ions,a eas some  
t e environmen is ossi e a imes. e ex erience i  
esigner invaria y ays consi era e a en ion o suc

FloodFill Image



mearge Image



onen s or ro en connec ion a s. ere is no oir  
 ion as e eve o e ai re uire o i en i y ose  
 Segmen a ion o non rivia images is one o e mos  
 rocessing. Segmen a ion accuracy e ermines e eve  
 o com u erize ana ysis roce ures. or is reason,  
 e a en o im rove e ro a i i y o rugge segmen a  
 suc as in us ria ins ec ion a ica ions, a eas some  
 t e environmen is ossi e a imes. e ex erience i  
 esigner invaria y ays consi era e a en ion o suc

ถัดมาเราจะทำการ Erode มันทิ้งไป เพราะว่าเรา Fill ตัวกลมแล้ว เมื่อ Erode ไปตัวที่เล็กกว่าตัว  
 กลม ๆ จะหายไปเกือบหมด และ Dilate ที่เหลือกลับมาจะเหลือเฉพาะ ตัวกลม ๆ ที่เป็นตัว 'o'

```

se = cv2.getStructuringElement(cv2.MORPH_RECT , (27,27))
find_point = cv2.erode(marge_image,se)

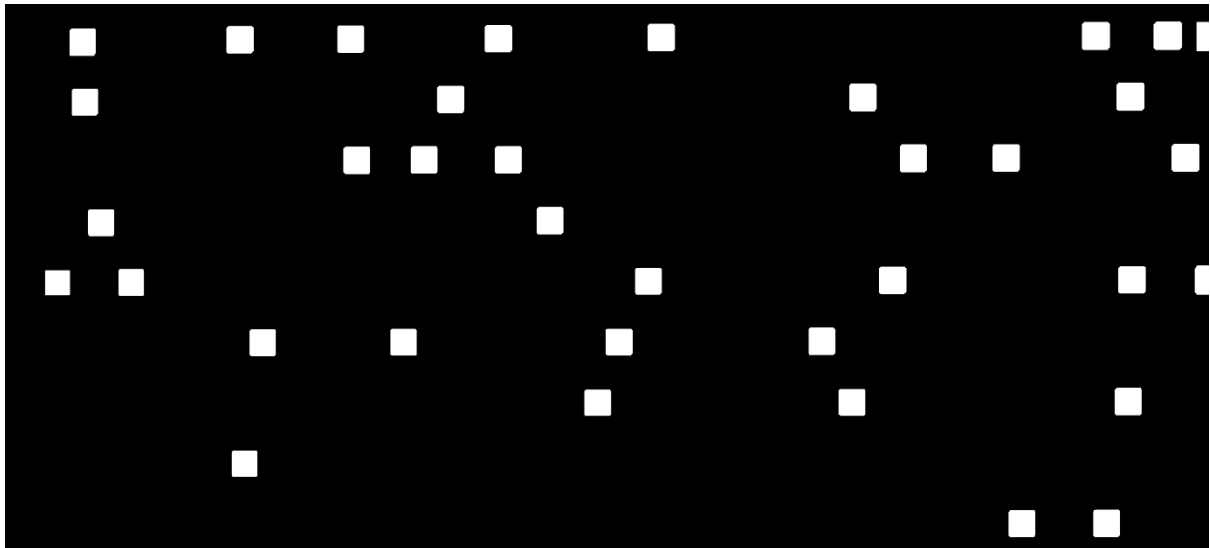
se = cv2.getStructuringElement(cv2.MORPH_RECT , (40,40))
fill_point = cv2.dilate(find_point,se)
  
```

Erode for find cicle



Dillate fill cicle





แต่พบว่ามีส่วนเกินที่เป็นตัวอักษรครึ่งตัวจากภาพต้นฉบับเราเลยจะทำการเลือกลบเฉพาะตรงนั้นออกไปและเอาส่วนที่เหลือมา And รูปต้นฉบับ เพื่อเอาแค่ตัว 'o' ที่เหลืออยู่มาก็จะได้แล้ว

```
h , w = ori_image.shape[:2]
mask = np.zeros((h,w) , np.uint8)

mask[:,1998:2018] = fill_point[:,1998:2018]

delete_excess = fill_point - mask

identified_o = ori_image & delete_excess
```

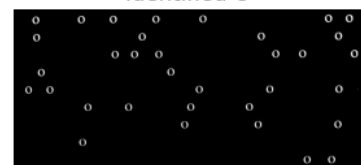
Mask Execess

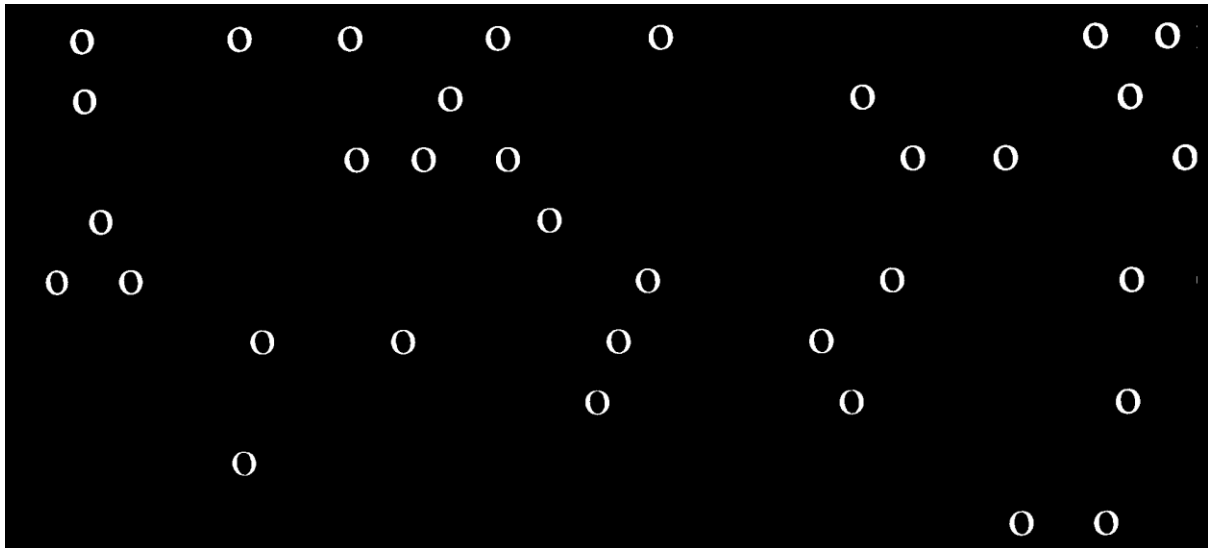


Delete Execess



Identified O





GitHub HW5 : [https://github.com/ThepokKung/FRA321-Image-processing/tree/main/LEB2\\_Hw/HW5](https://github.com/ThepokKung/FRA321-Image-processing/tree/main/LEB2_Hw/HW5)