PS5

Erode/dilate filters (closing)

Iteration of 1x1 kernel

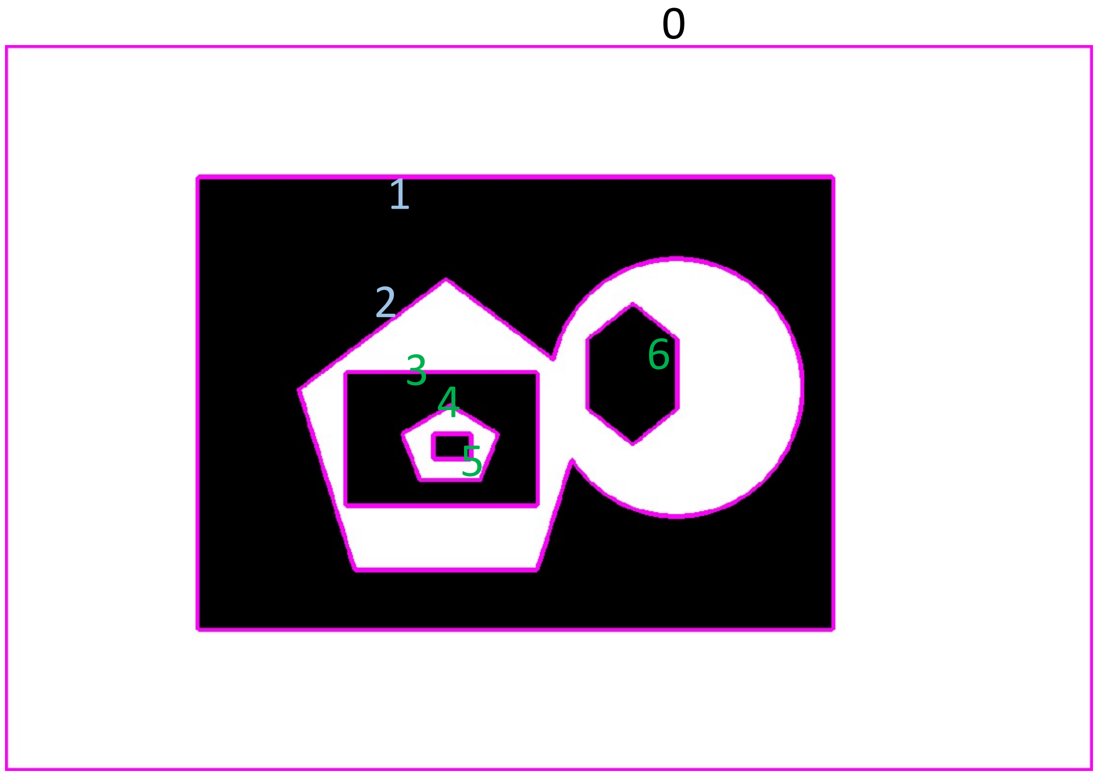
```
img1 = img0.copy()
for i in range(cnt):
   img1 = img0.erode(img1, None)
img2 = img1.copy()
for i in range(cnt):
   img2 = img2.dilate(img2, None)
```

Kernel mask type

```
k_e = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))
# other kernel shape
# MORPH_ELLIPSE, MORPH_RECT
img1 = cv2.erode(img0, k_e)
img2 = cv2.dirlate(img1, k_e)
```

OpenCV Contour and its features

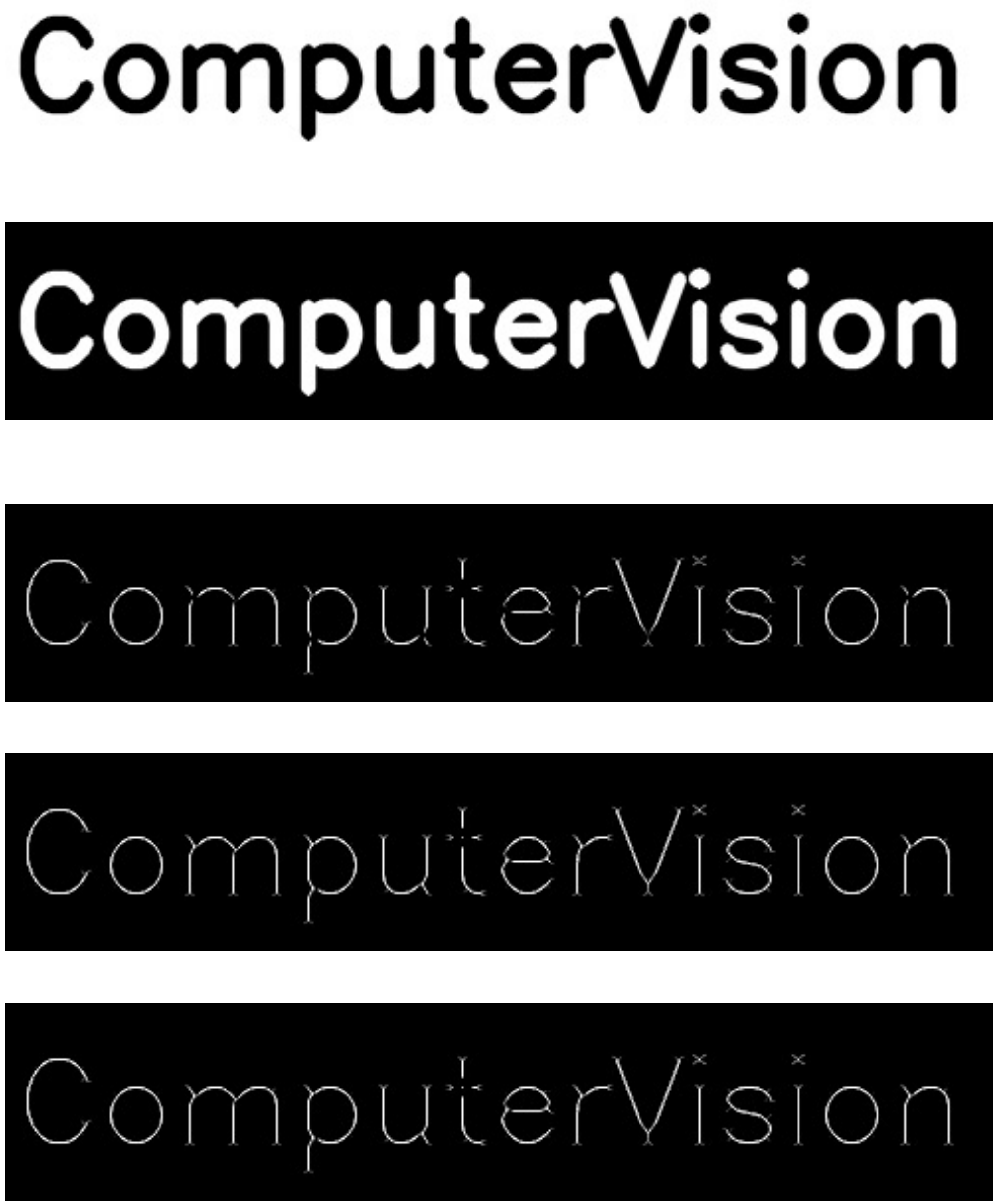
```
cont, hier = cv2.findContours(thr, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(img, cont, -1, (255,0,255),3)
print(hier)
print('Moment:  ', cv2.moments(cont[2]))
print('Area:    ',  cv2.contourArea(cont[2]))
print('Perimeter', cv2.arcLength(cont[2], True)) # closed contour

# hierarchy output (refer right side)
# data: next, previous, child, parent
[[[-1 -1  1 -1]                    # 0
  [-1 -1  2  0]                    # 1
  [-1 -1  3  1]                    # 2
  [ 6 -1  4  2]                    # 3
  [-1 -1  5  3]                    # 4
  [-1 -1 -1  4]                    # 5
  [-1  3 -1  2]]]                  # 6
Moment:   {'m00': 131093.5, 'm10': 79058336.66666666, 'm01': 53512399.0, …}
Area:     131093.5
Perimeter 1738.0844626426697
```



https://docs.opencv.org/4.5.3/dd/d49/tutorial_py_contour_features.html

Thinning

```
# Important: Reverse image (black background)
img1 = cv2.bitwise_not(img0)
# Kernel: 4 neighbor
k_e = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))
# Target image
thin = np.zeros(img1.shape, dtype=np.uint8)
# repeat until no white area
while cv2.countNonZero(img1) != 0:
  er = cv2.erode(img1, k_e)
  # OPEN: erosion then dilation (remove noise)
  op = cv2.morphologyEx(erode, cv2.MORPH_OPEN,k_e)
  subset = er – op
  thin = cv2.bitwise_or(subset, thin)
  img1 = er.copy()
```



Advanced (ex. local averaging)

```
blr = cv2.blur(gray, (5,5))
ave = cv2.addWeighted(gray, 4, blr, -4, 128)
```