

ใช้ Canny Edge Detection และ Contour Analysis

5 ขั้นตอนหลัก

ขั้นตอน	รายละเอียด
1. Load Image	โหลดภาพและแปลงเป็น grayscale เพื่อเตรียมสำหรับการหาขอบ
2. Canny Edge Detection	ใช้ Otsu threshold หา high/low threshold อัตราโน้มนำ > Canny edge > Dilation ปิดช่องว่าง
3. Find Contours	ใช้ cv2.findContours() แบบ RETR_CCOMP หา contours ทั้งหมดและจัดเป็น 2 ระดับ (outer + holes)
4. Shape Filtering	กรองด้วย: Circularity (>0.6), Aspect Ratio (0.6-1.4), Area (10-10000), Size tolerance (25% ของ max)
5. Concentric Detection	หา contour คู่ที่มี center ใกล้เคียง (<5 pixels) = ตัว O (outer + inner hole) + เพิ่มตัว O ที่หาย ด้วย manual

วิธีการทำงาน

1. cv2.Canny(): Edge detection หาขอบด้วยอัลกอริทึม Canny - ใช้ gradient และ non-maximum suppression
2. Otsu Threshold: หา threshold อัตราโน้มนำโดยแยก foreground/background ให้ได้ variance ต่ำสุด ใช้กับ Canny
3. cv2.dilate(): ขยายขอบด้วย morphological operation ปิดช่องว่างเล็กๆ ระหว่างขอบ kernel 3x3, iterations=1
4. cv2.findContours(): หา contours จาก binary image, RETR_CCOMP = 2-level hierarchy (เก็บ outer + holes)
5. วัดความกลม = $4\pi A/P^2$ โดย A=area, P=perimeter, ค่า=1 เป็นวงกลมสมบูรณ์
6. วัดสัดส่วน = width/height, ค่าใกล้ 1 = สี่เหลี่ยมจัตุรัส/วงกลม
7. cv2.contourArea(): คำนวณพื้นที่ contour ใช้ Green's theorem
8. cv2.arcLength(): คำนวณความยาวรอบ contour (perimeter)
9. หาคู่ contour ที่มี center ใกล้เคียง distance = $\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2} < 5$ pixels
10. $C = 4\pi A / P^2$ (ค่า 0-1, วงกลมสมบูรณ์ = 1)
11. $AR = W / H$ (ค่าใกล้ 1 = รูปทรงสมดุล)
12. $d = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$
13. Canny Thresholds: Low = $0.5 \times \text{Otsu_threshold}$, High = Otsu_threshold

พารามิเตอร์การกรอง

พารามิเตอร์	ค่า / เงื่อนไข
MIN_AREA	10 pixels ² (กรองจุดเสียงรบกวนเล็ก)
MAX_AREA	10,000 pixels ² (กรองวัตถุใหญ่เกินไป)
MIN_CIRCULARITY	0.6 (ยอมรับรูปค่อนข้างกลม ผ่อนปรนจาก 0.7)
ASPECT_RATIO	0.6 - 1.4 (ยอมรับรูปไม่สมดุลเล็กน้อย)
SIZE_TOLERANCE	0.25 (25% ของ max bbox area = ยอมรับ o เล็ก)

ผลลัพธ์

Detected O 36 ตัวอักษร