

LAB : Unsupervised Object Counting & Classification (OpenCV + KMeans/DBSCAN)

การนับและจัดกลุ่มวัตถุแบบไม่มีผู้สอน (Unsupervised) ด้วย OpenCV และอัลกอริทึม Clustering (K-Means, DBSCAN)

1) วัตถุประสงค์

1. สร้าง **mask** ของวัตถุจากภาพจริงด้วยการแปลงค่าสีเทา, threshold และ morphology
2. แยกวัตถุเป็นชิ้น (connected components) และสกัด **shape features** ที่ใช้อธิบายรูปทรง
3. จัดกลุ่มวัตถุอัตโนมัติด้วย **K-Means** (กำหนด **k**) และ **DBSCAN** (ไม่ต้องกำหนดจำนวนกลุ่มล่วงหน้า)
4. แสดงผลการจัดกลุ่มด้วยการระบายมาสก์/เส้นขอบ พร้อม **Class ID** และ **จำนวนต่อคลาส**

2) พื้นฐานทฤษฎี

2.1 การสร้าง Mask และการเตรียมภาพ (Preprocessing)

- **Grayscale**: เปลี่ยนจาก BGR \rightarrow เทา: $I = 0.299R + 0.587G + 0.114B$
$$0.299R + 0.587G + 0.114B = 0.299R + 0.587G + 0.114B$$
- **Blur (Gaussian)**: ลดสัญญาณรบกวนก่อน threshold
- **Threshold**:
 - Fixed Threshold: $I(x) > T \Rightarrow 1$ (วัตถุ), ไม่งั้น 0 (ฉากหลัง)
 - Otsu/Adaptive: กำหนดเกณฑ์อัตโนมัติตามฮิสโตแกรม/บริบทพื้นที่
- **Morphology**:
 - Opening = Erosion \rightarrow Dilation (ลบจุดเล็ก ๆ)
 - Closing = Dilation \rightarrow Erosion (ปิดช่องว่าง/รอยขาด)
- **Connected Components (CC)**: รวมพิกเซลที่เชื่อมถึงกันให้เป็นออบเจกต์เดียว

2.2 ฟีเจอร์รูปร่าง (Shape Features)

ให้คอนทัวร์ของวัตถุเป็น C , พื้นที่ A , เส้นรอบรูป P

- Aspect ratio: $AR = \frac{w}{h}$
- Circularity: $Circ = \frac{4\pi A}{P^2}$ (ใกล้ 1 = กลม)
- Solidity: $Sol = \frac{A}{A_{hull}}$
- Equivalent diameter: $\sqrt{\frac{4A}{\pi}}$
- Hu Moments (7 ค่า): อินเวเรียนต์ต่อการเลื่อน/หมุน/สเกล (ในโค้ดแปลงเป็น log เพื่อสเกลค่านี้)

2.3 K-Means Clustering

- วัตถุประสงค์: ลด Sum of Squared Errors (SSE)

$$\min_{\{S_j, \mu_j\}} \sum_{j=1}^k \sum_{x \in S_j} \|x - \mu_j\|^2$$

- ขั้นตอน: (1) สุ่มเซนทรอยด์ k จุด \rightarrow (2) จัดตัวอย่างเข้าคลัสเตอร์ที่ใกล้สุด \rightarrow (3) อัปเดตเซนทรอยด์ \rightarrow ทำซ้ำจนคงที่
- พารามิเตอร์สำคัญ: k (จำนวนกลุ่ม), การสเกลข้อมูล (ใช้ **StandardScaler** ในโค้ด)

2.4 DBSCAN

- นิยามจุด **core** เมื่อมีเพื่อนบ้านภายในรัศมี **eps** ไม่น้อยกว่า **min_samples**
- จุดที่เข้าถึงได้จาก core จะเป็นสมาชิกคลัสเตอร์เดียวกัน
- จุดที่ไม่เข้าถึงได้จาก core ใด ๆ จะเป็น **noise** (label = -1)
- ข้อดี: ไม่ต้องบอกจำนวนคลัสเตอร์ล่วงหน้า, จัดการ outlier ได้ดี

3) ขั้นตอนการทดลอง

1. โหลดภาพจาก [Link](#) นี้
2. ใช้คำสั่งนี้เพื่อติดตั้ง libraries ที่จำเป็น

```
pip -q install opencv-python-headless scikit-learn matplotlib numpy
```
3. นำเข้า jupyter notebook จาก [Link](#) นี้
4. ทดลองโดยใช้ทั้ง 2 รูป โดยลองใช้ทั้ง **K-Means Clustering** และ **DBSCAN** โดยปรับค่า parameters ต่าง ๆ ตามความเหมาะสม