

**Risang Panggalih**

**24/550961/PPA/06961**

## **Laporan Tugas Mata Kuliah Computer Vision Lanjut**

### **“Object Detection”**

#### **I. Pipeline Umum**

##### **1. Load**

- o `img_rgb = Image.open("image.jpg").convert("RGB")`
- o `mask_img = Image.open("image_mask.jpg").convert("L")`

Keduanya diubah menjadi NumPy array.

##### **2. Grayscale**

- o `rgb_to_gray(img_rgb)` → mengubah citra menjadi single-channel dengan rentang 0–255 (float32). Fungsi ini menghasilkan matrix grayscale 2 dimensi.

##### **3. Threshold Otsu**

- o `t = otsu_threshold(gray)` menghitung threshold terbaik yang memisahkan “background” dengan “object” dengan cara memaksimalkan inter-class variance.
- o `Pred mask  $\approx$  (gray  $\geq$  t).astype(np.uint8)`

##### **4. Ground-truth tidy**

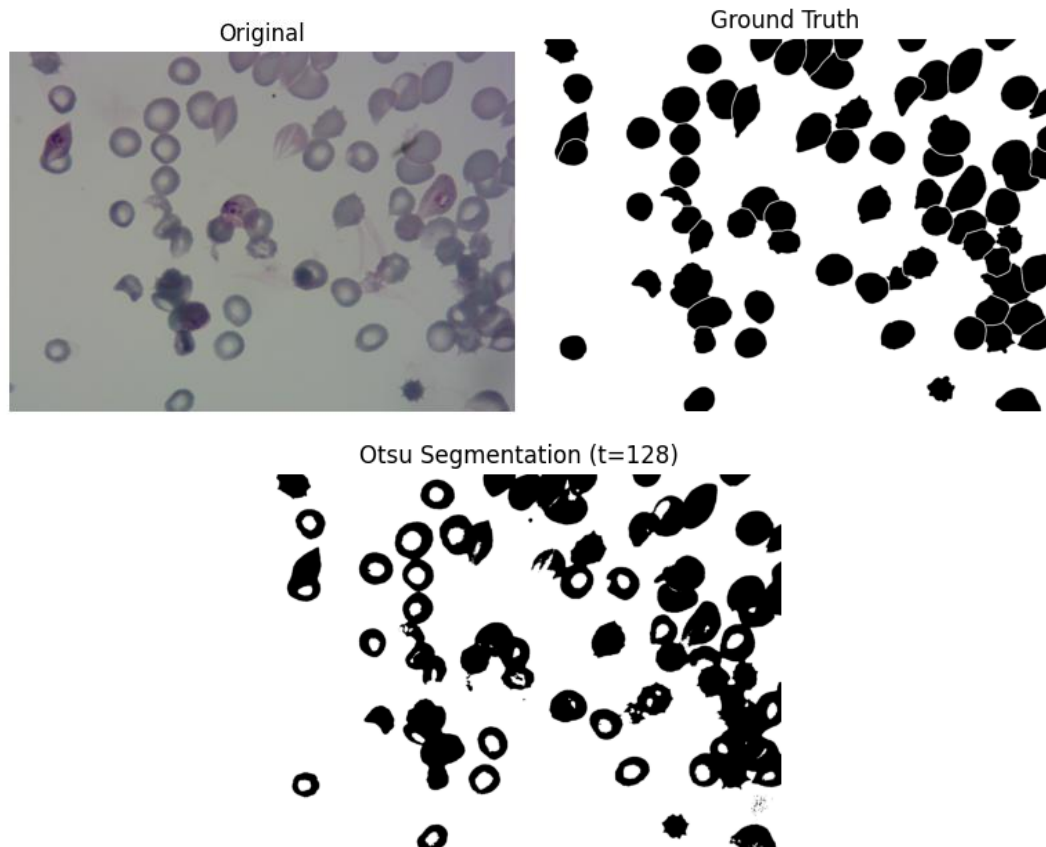
- o `ensure_binary_mask(mask_arr)` memastikan masking ground truth sesuai dengan citra.

##### **5. Metric**

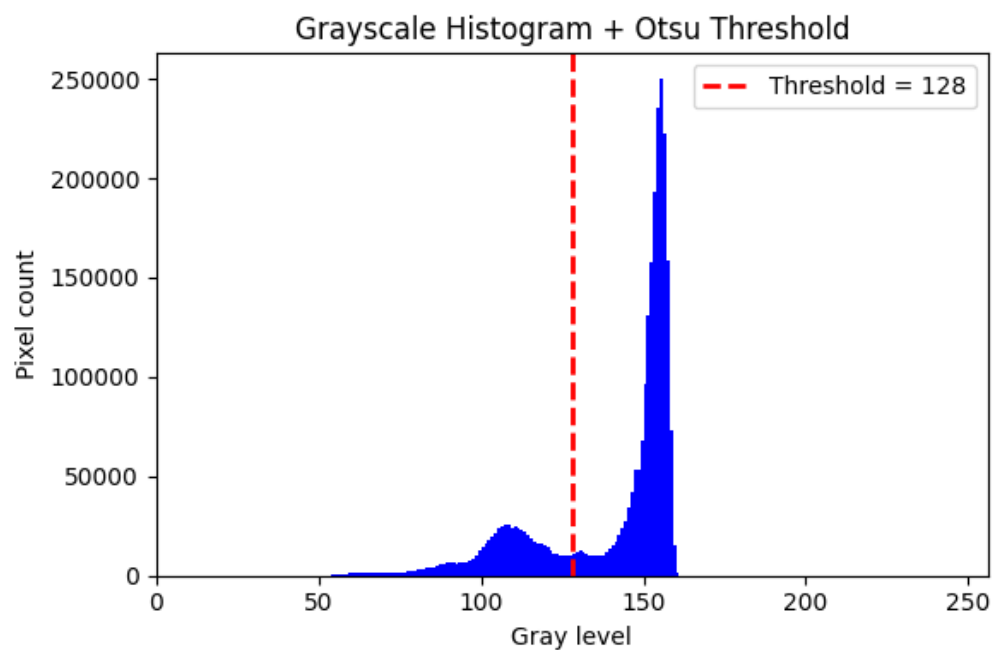
- o **Pixel Accuracy:** Tingkat akurasi piksel yang sama.
- o **IoU (Intersection over Union):** Mengukur seberapa tumpang tindih objek prediksi dengan ground truth?  
$$\text{inter} = \text{sum}(\text{pred}==1 \ \& \ \text{gt}==1); \text{union} = \text{sum}(\text{pred}==1 \ | \ \text{gt}==1); \text{iou} = \text{inter}/(\text{union}+1\text{e-}12).$$
  
1e-12 menghindari pembagian nol pada kasus degeneratif.

##### **6. Plot**

- o 3-panel: Original | Ground Truth | Predicted Mask.



- o **Histogram** grayscale value (0–255) dengan garis merah putus-putus menandakan threshold  $t$ .



## II. Cara Kerja Threshold Otsu

Diberikan gray levels 0...255:

- Membangun histogram  $\text{hist}[k]$ .
- Normalisasi ke probabilitas:  $p[k] = \text{hist}[k] / N$ .
- Probabilitas kelas kumulatif hingga  $t$ :  $\omega(t) = \sum_{k \leq t} p[k]$ .
- Mean/rata-rata kumulatif:  $\mu(t) = \sum_{k \leq t} k \cdot p[k]$ .
- Total mean:  $\mu_T = \sum_{k} k \cdot p[k]$ .

## III. Perhitungan Inter-class variance

$$\sigma_b^2(t) = (\mu_T \cdot \omega(t) - \mu(t))^2 / (\omega(t) \cdot (1 - \omega(t)))$$

Pick  $t^* = \arg\max_t \sigma_b^2(t) \rightarrow$  Otsu threshold.

## IV. Kesimpulan

- Otsu diimplementasikan dengan tahap: histogram  $\rightarrow$  probabilitas  $\rightarrow$  penghitungan kumulatif  $\rightarrow$  inter-class variance  $\rightarrow$  argmax. Hasil otsu thresholding dengan  $t$  optimal yaitu 128 adalah Pixel Accuracy = 0.9588 dan IoU = 0.9459.
- Otsu mengasumsikan histogram kurang lebih bimodal (satu gundukan untuk background, satu untuk foreground). Jika pencahayaan tidak merata, warnanya berantakan, atau intensitas objek/latar belakang tumpang tindih, Otsu akan kesulitan untuk memisahkan objek.
- Kode yang dibuat memberlakukan aturan " $\text{intensitas} \geq t$ " sebagai foreground. Jika objek yang ingin dideteksi ternyata lebih gelap daripada background maka logic perlu dibalik (seperti pada contoh penerapan penggunaan dengan citra sel darah dalam kode, posisi foreground dan background masih terbalik).