Image Segmentation

Image Segmentation

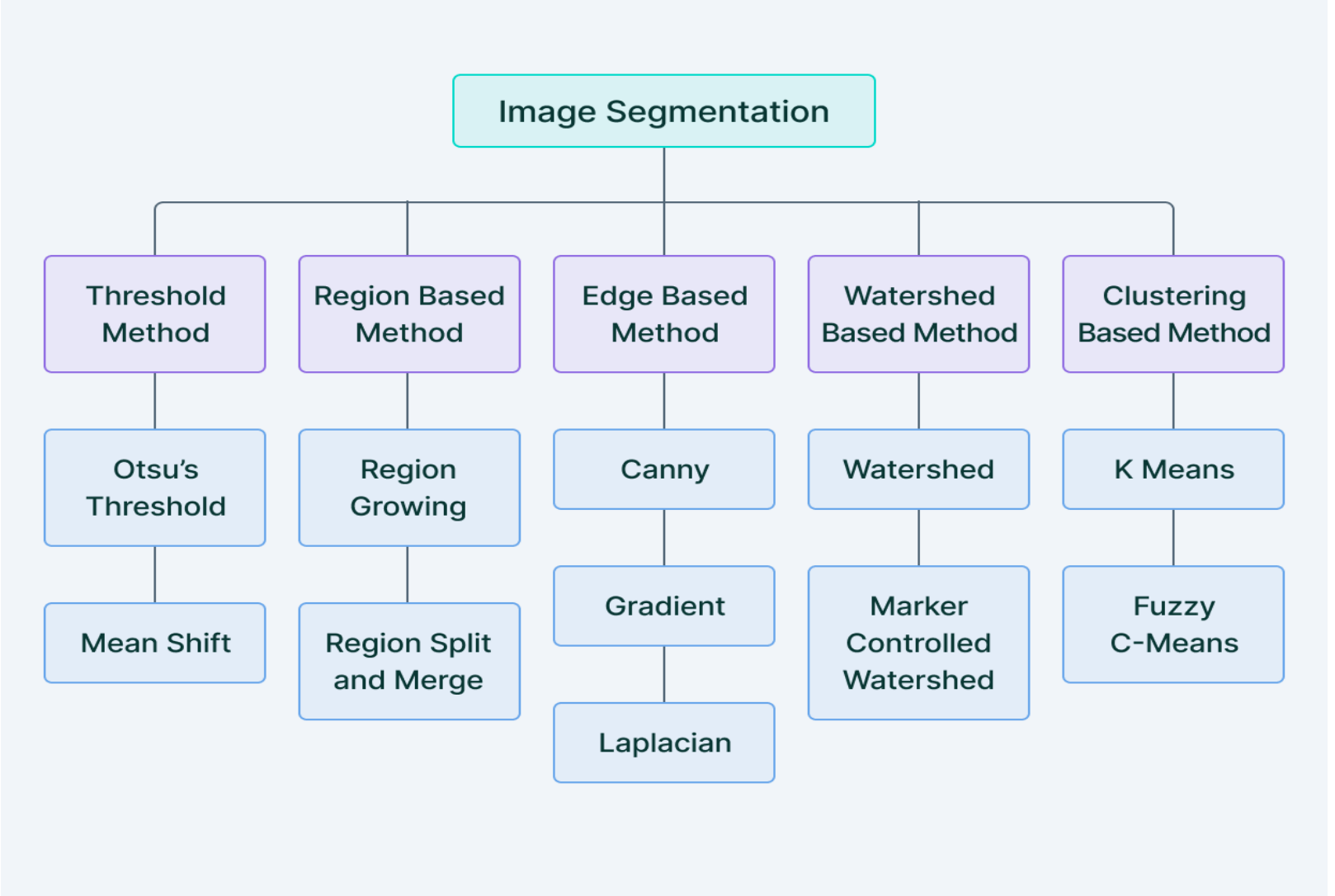


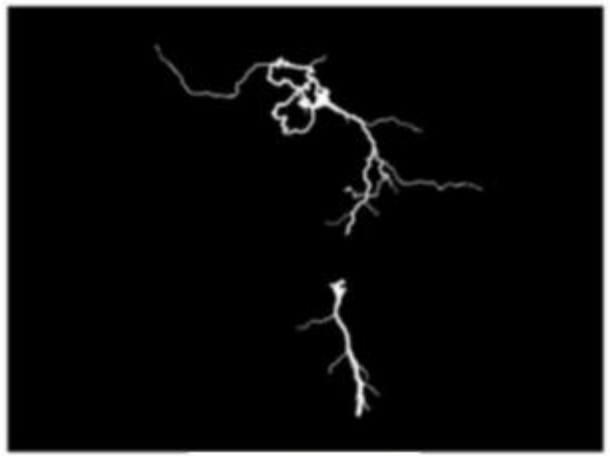
Image Segmentation

- 1. Edge-based Segmentation
- 2. Threshold-based Segmentation
- 3. Region-based Segmentation
- 4. Clustering-based Segmentation
- 5. Watershed Segmentation
- 6. **Neural Network-based segmentation**

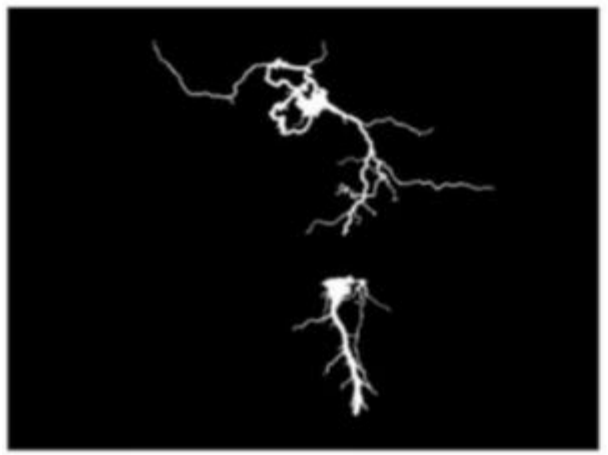
Region Growing

- Start from seed point(s) to grow a region. It can be a random seed.
- Given input image $f(x, y)$, seed points image $S(x, y)$ and threshold T (typically intensity/distance)
  - If $|f(x_i, y_j) - S(x, y)| \leq T, f(x_i, y_j)$ belongs to $S_i(x, y)$
  - Else $f(x_i, y_j)$doesn't belong to $S_i(x, y)$

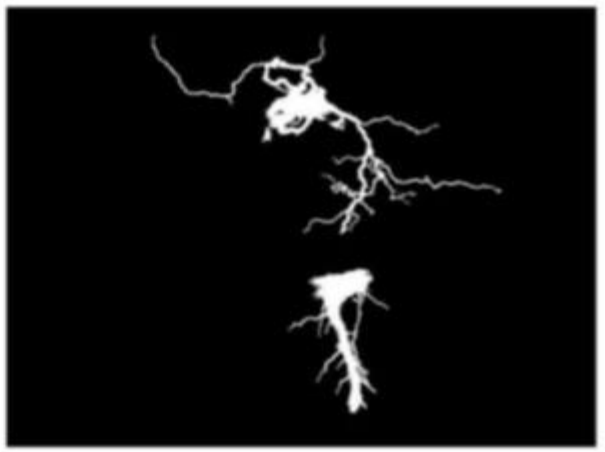
- Repeat:
  - Aggregate neighboring pixels that are similar to the cluster model
  - Update cluster model with newly incorporated pixels
- This is a generalized flood fill
- When the cluster stops growing, begin with a new seed pixel and continue
- One danger: Since multiple regions are not grown simultaneously, the threshold must be appropriate, or else early regions will dominate



T: 190-255



T: 155-255

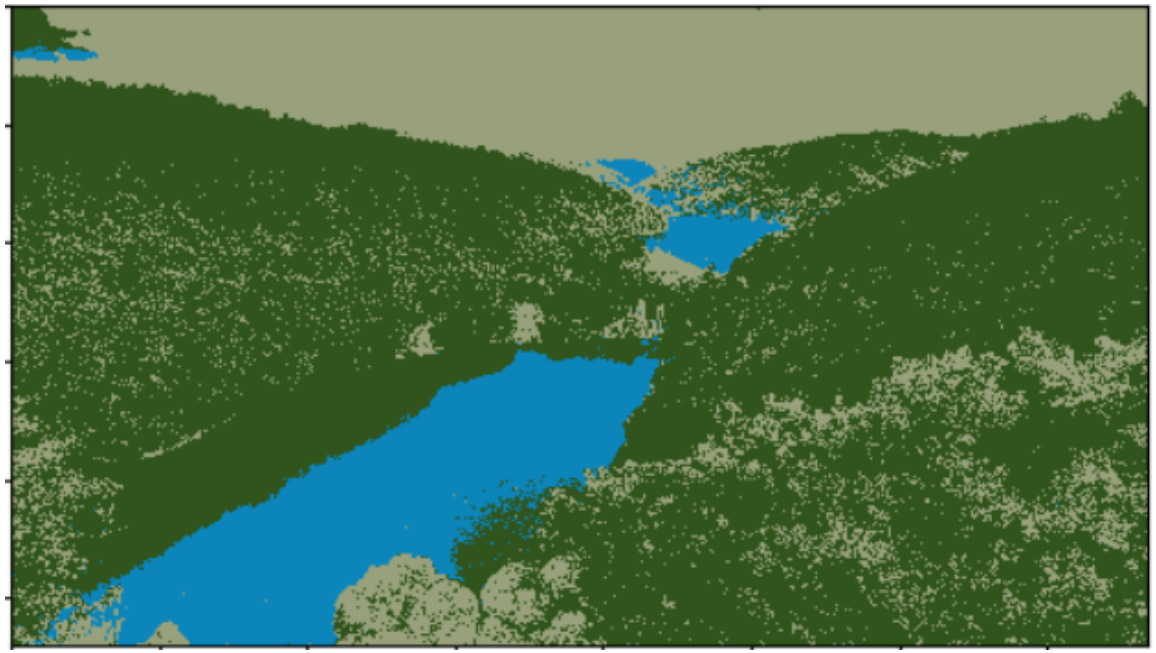


T: 125-255

K-mean Clustering

- Given input image $f(x, y)$
  - 1. Specify an initial set of cluster center, m_j for all *j* =1,….k (within the image intensity range)
  - 2. For each pixel in image $f(x_i, y_i)$, assign it to the closest cluster based on the Euclidian norm. i.e., $f(x_i, y_i)$ belongs to cluster *j* obtained $min_j ||f(x_i, y_i) - m_j||$, j=1,2,…,*k*
  - 3. Update the cluster means
  - 4. Repeat Step 2 to 3 until convergence criteria (either number of iterations or m_j stops changing)

K-mean Clustering

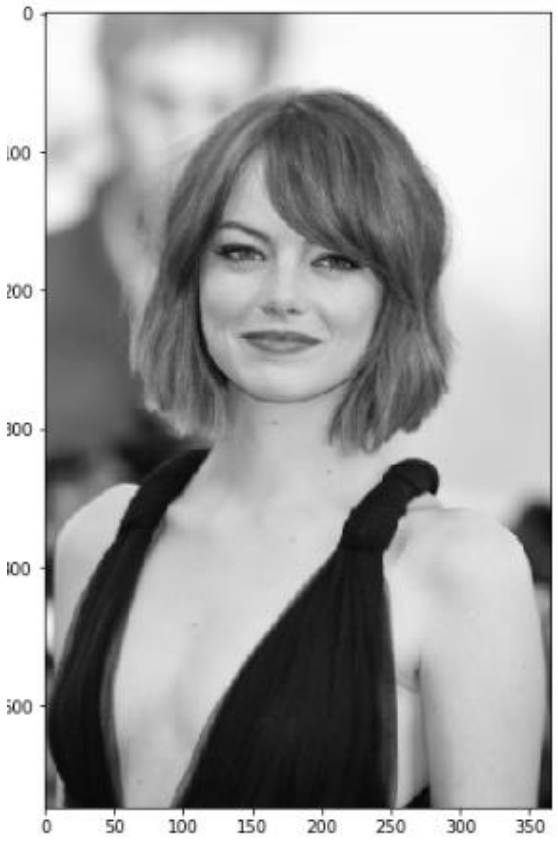


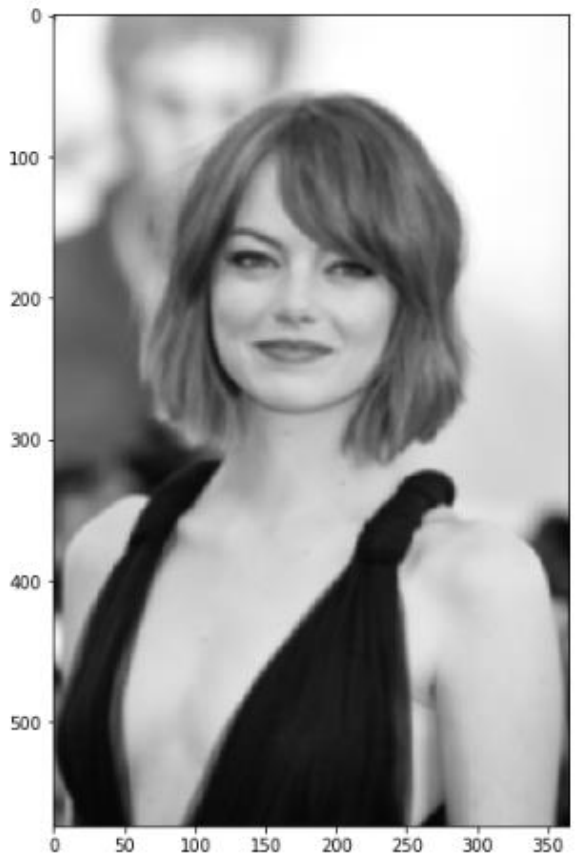
Canny Edge Detection

- The **Canny edge detector** is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.
- Composed of 5 steps:
    1. Noise reduction (using Gaussian blur);
    2. Gradient calculation;
    3. Non-maximum suppression;
    4. Double threshold;
    5. Edge Tracking by Hysteresis.

Canny Edge Detection

- **GAUSSIAN BLUR**
  - The blur removes some of the noise before further processing the image.
  - A sigma of 1.4 is used in this example and was determined through trial and error.
  - Kernel size 5x5





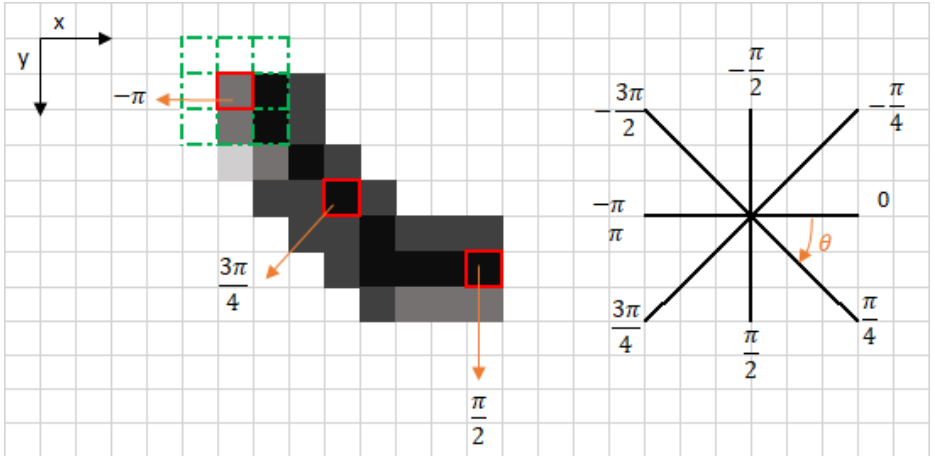
Canny Edge Detection

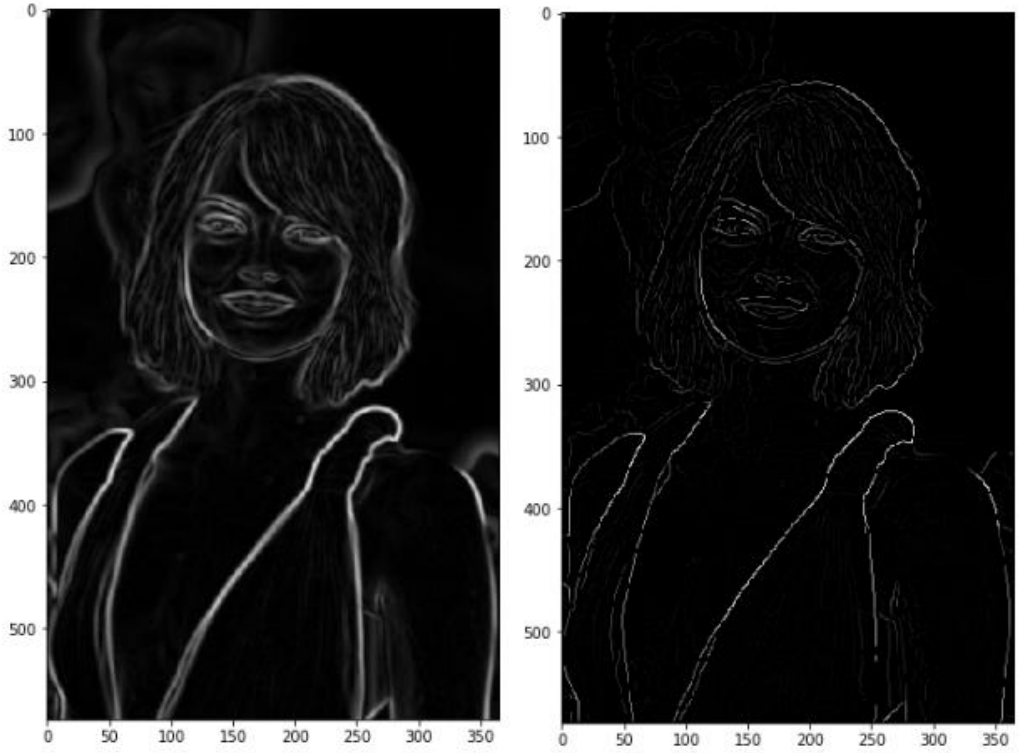
- **GRADIENT CALCULATION**

- The gradients can be determined by using a **Sobel filter**.
- An edge occurs when the color of an image changes, hence the intensity of the pixel changes as well.

Canny Edge Detection

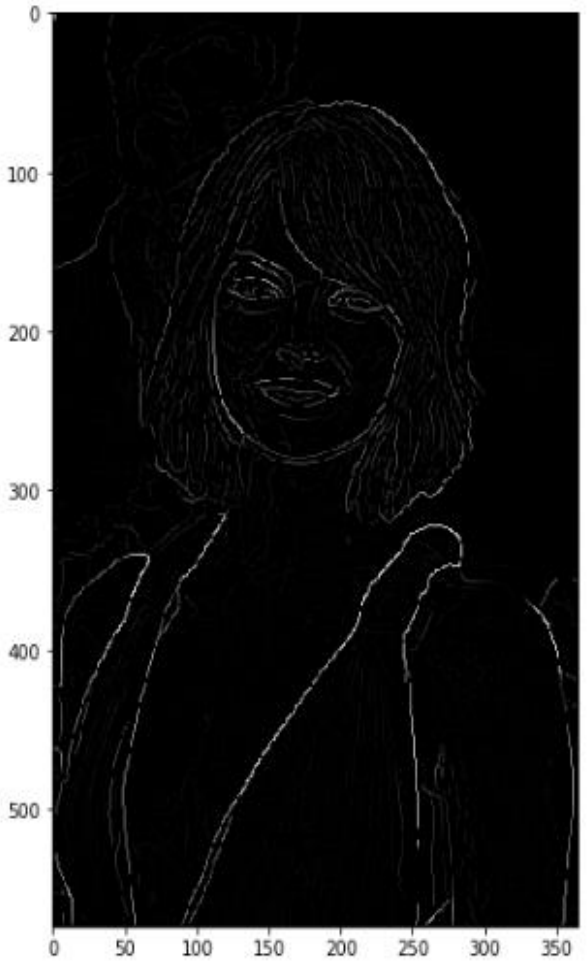
- **NON MAXIMUM SUPPRESSION**
  - Ideally, the final image should have thin edges. Thus, we must perform non-maximum suppression to thin out the edges.
  - The principle is simple: the algorithm goes through all the points on the gradient intensity matrix and finds the pixels with the maximum value in the edge directions.

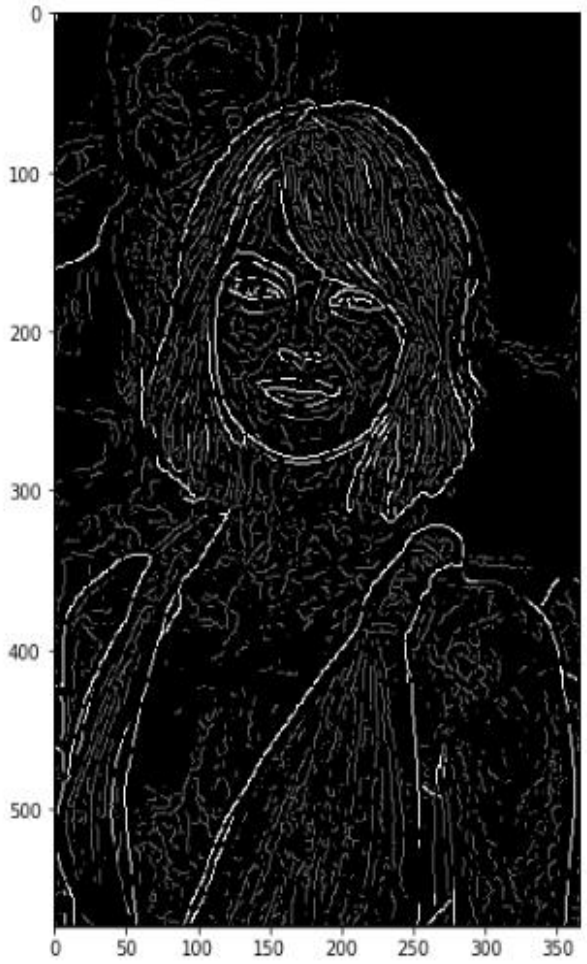




Canny Edge Detection

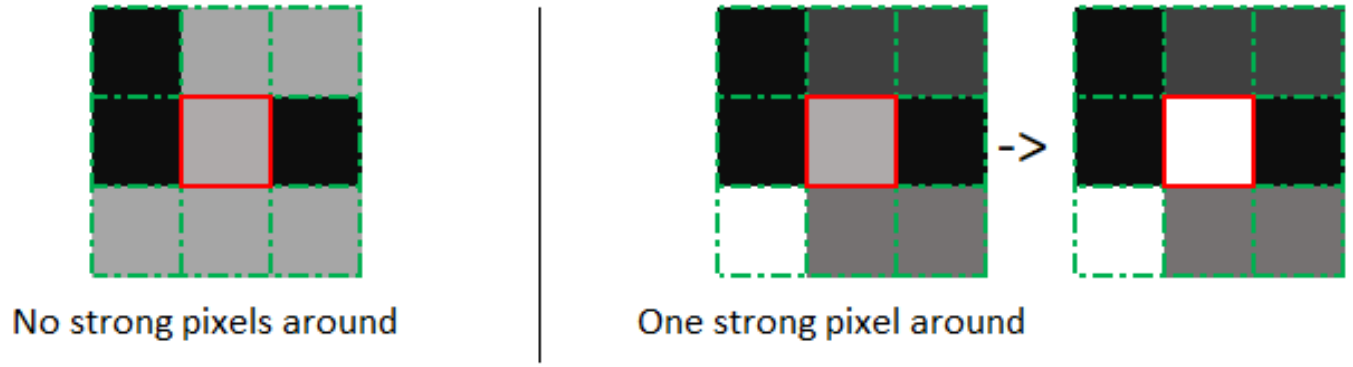
- **DOUBLE THRESHOLDING**
- The double threshold step aims at identifying 3 kinds of pixels: strong, weak, and non-relevant:
  - Strong pixels are pixels that have an intensity so high that we are sure they contribute to the final edge.
  - Weak pixels are pixels that have an intensity value that is not enough to be considered strong ones, but yet not small enough to be considered non-relevant for edge detection.
  - Other pixels are considered non-relevant for an edge.
- Select pixels above the High Threshold, below the Low Threshold, and pixels in between the thresholds.

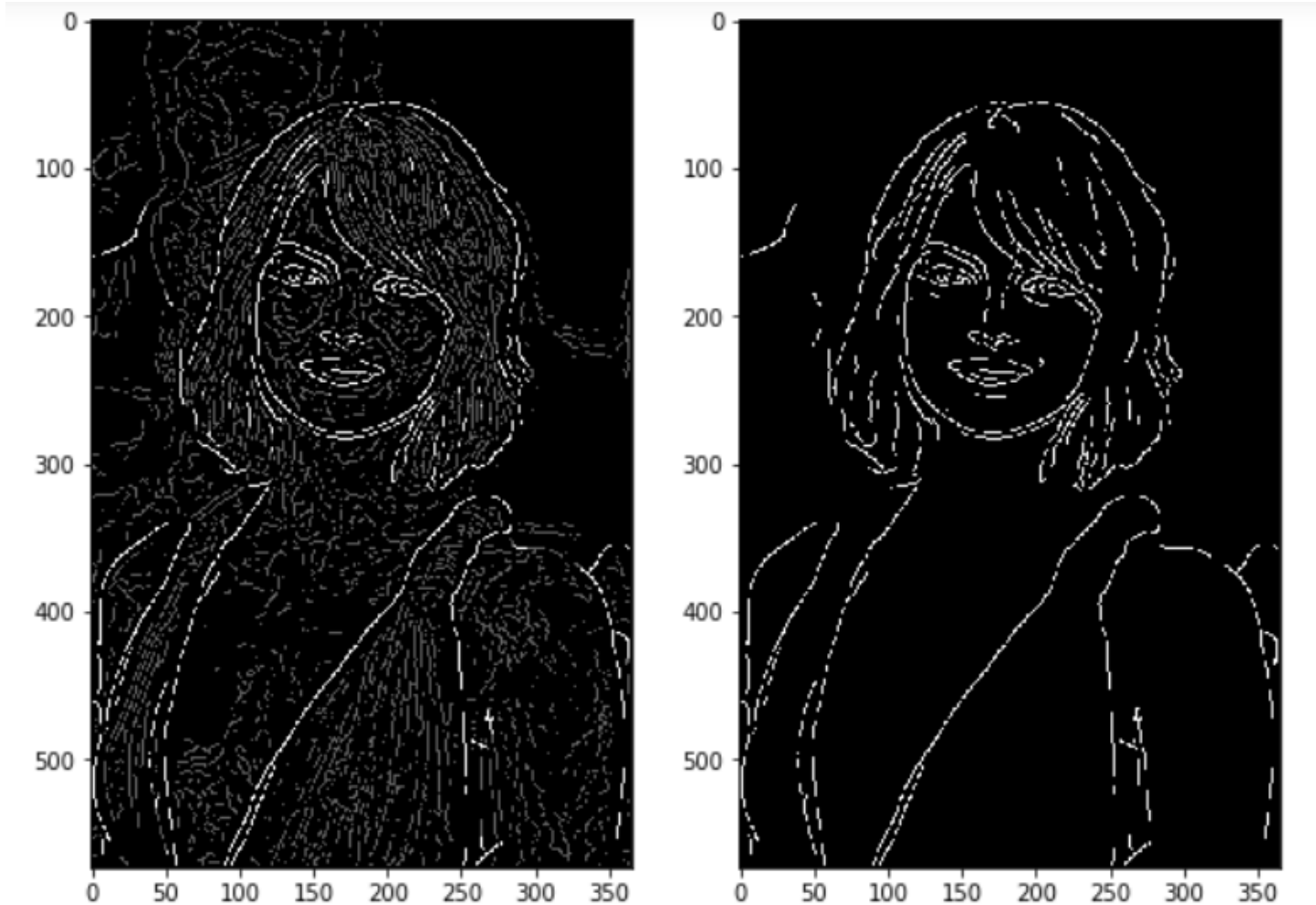




Canny Edge Detection

- **EDGE TRACKING BY HYSTERESIS**
  - Based on the threshold results, the hysteresis consists of transforming weak pixels into strong ones, if and only if at least one of the pixels around the one being processed is a strong one, as described below:
  -




0
100
200
300
400
500
0
50
100
150
200
250
300
350
0
100
200
300
400
500
0
50
100
150
200
250
300
350

Otsu’s Thresholding Concept

- Automatic global thresholding algorithms usually have the following steps.
    1. Process the input image
    2. Obtain image histogram (distribution of pixels)
    3. Compute the threshold value *T*
    4. Replace image pixels with white in those regions, where saturation is greater than *T* and into the black in the opposite cases.
    - Usually, different algorithms differ in step 3.

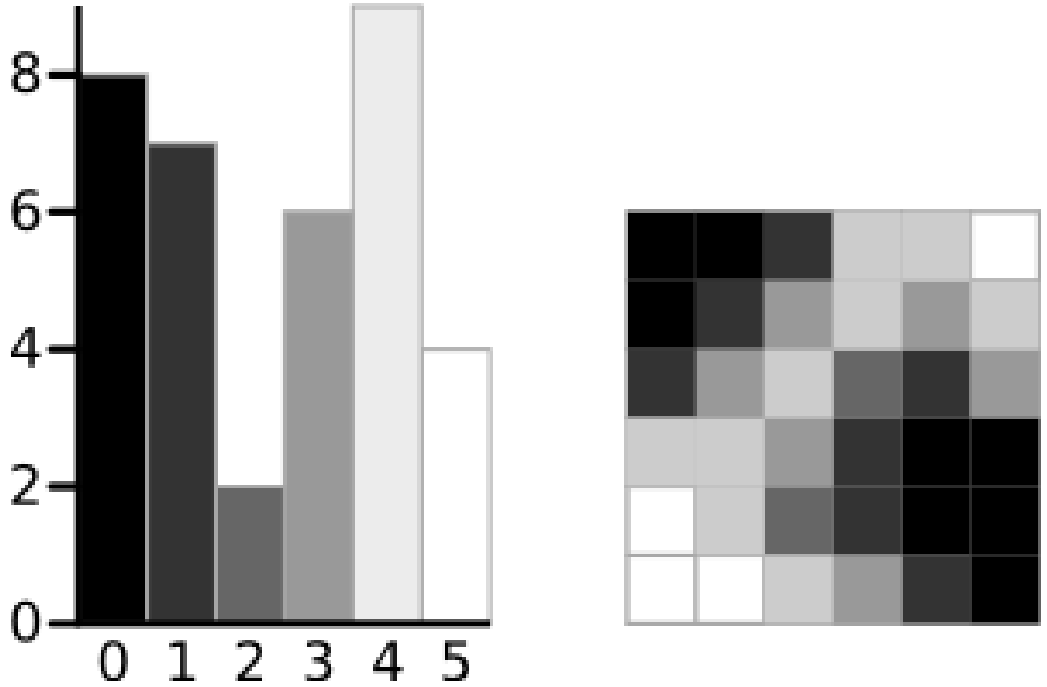
Otsu's Thresholding Concept

- The core idea is to separate the image histogram into two clusters with a threshold defined as a result of minimization of the weighted variance of these classes denoted by σ_w^2

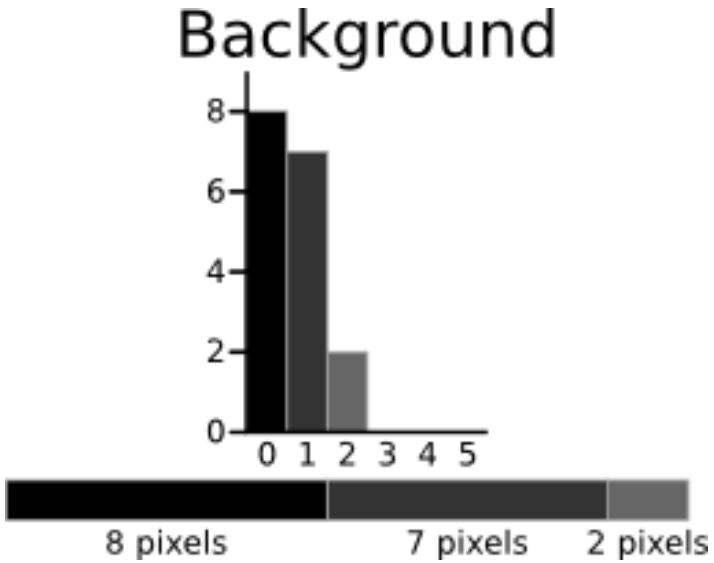
$$\sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2$$

Where , W_b, W_f are the probabilities of the two classes divided by a threshold t, Which value is within the range from 0 to 255 inclusively.

Otsu’s Thresholding Concept



Otsu’s Thresholding Concept



$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

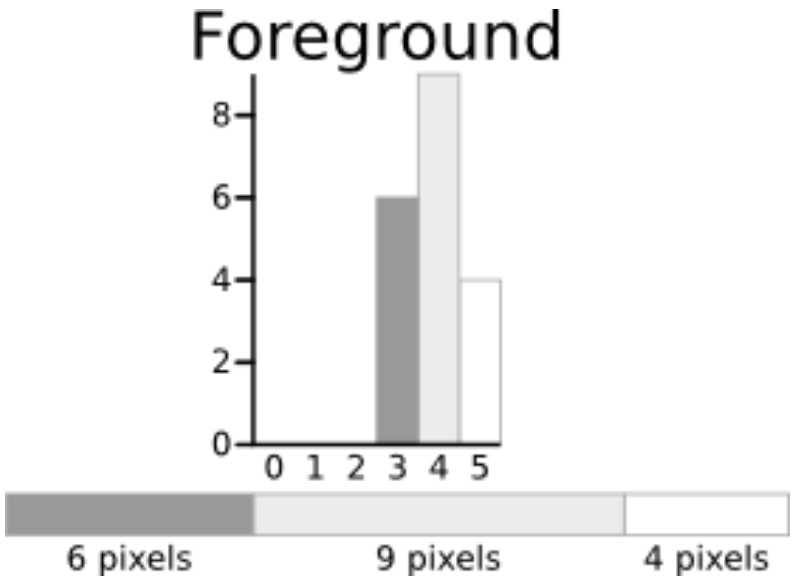
$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

$$\text{Variance} \quad \sigma_b^2 = \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17}$$

$$= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17}$$

$$= 0.4637$$

Otsu’s Thresholding Concept



$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}$$

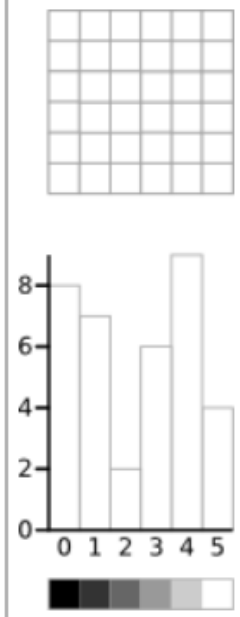
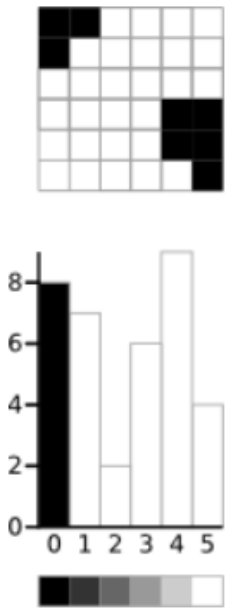
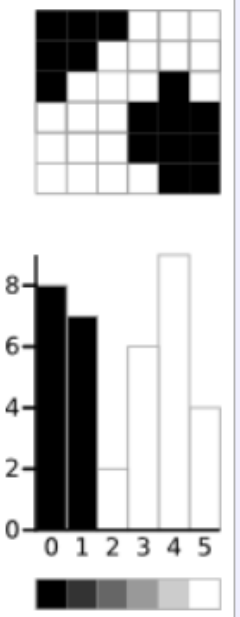
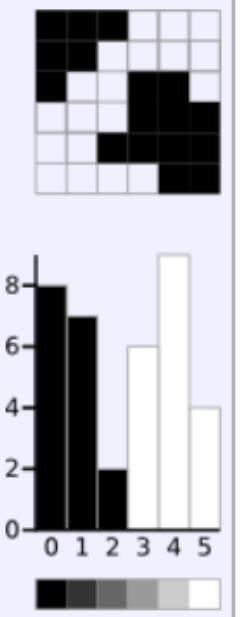
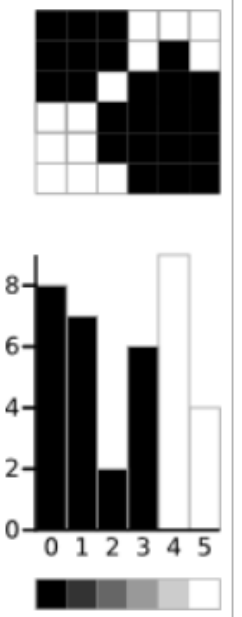
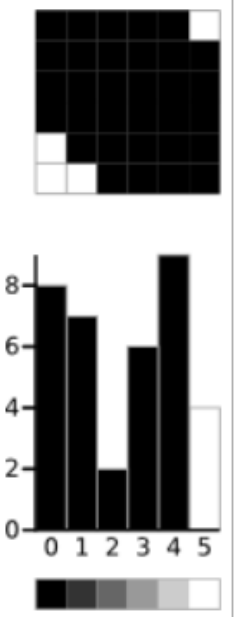
$$= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}$$

$$= 0.5152$$

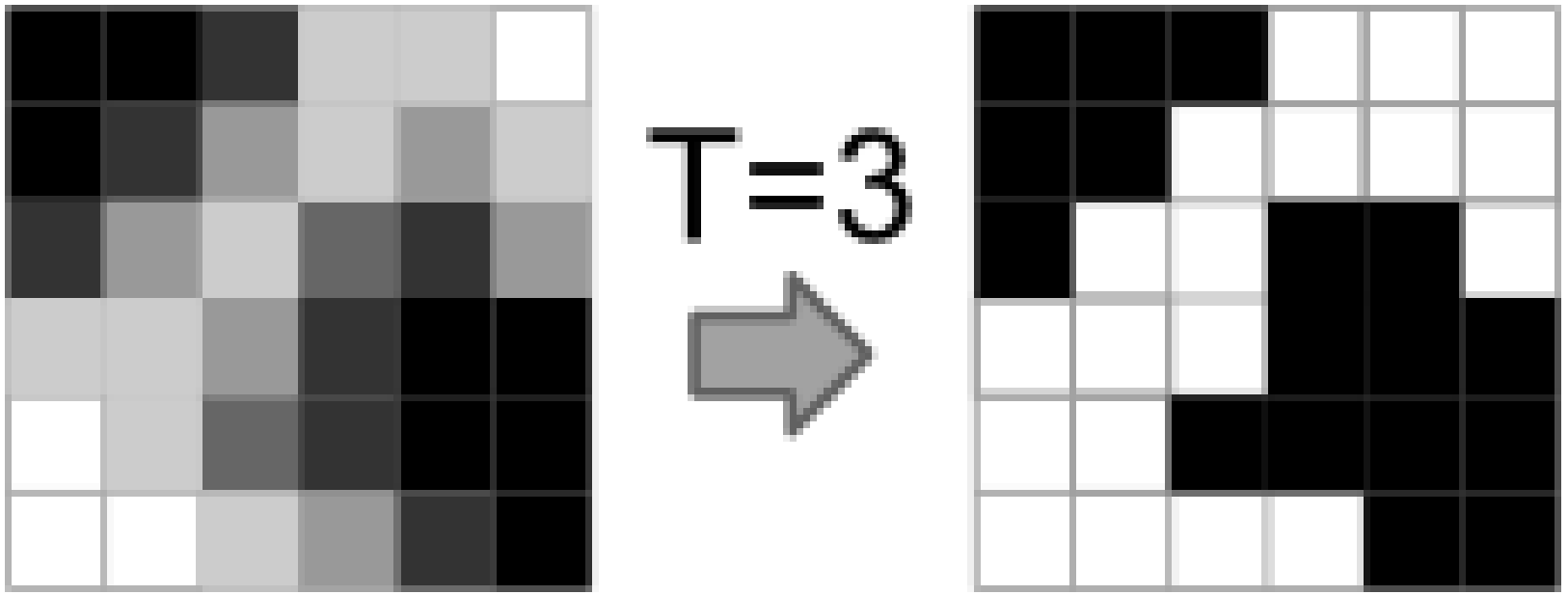
Otsu’s Thresholding Concept

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152$$
$$= 0.4909$$

Otsu's Thresholding Concept

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2_b = 0$	$\sigma^2_b = 0$	$\sigma^2_b = 0.2489$	$\sigma^2_b = 0.4637$	$\sigma^2_b = 1.4102$	$\sigma^2_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2_f = 3.1196$	$\sigma^2_f = 1.9639$	$\sigma^2_f = 0.7755$	$\sigma^2_f = 0.5152$	$\sigma^2_f = 0.2130$	$\sigma^2_f = 0$
Within Class Variance	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	$\mathbf{\sigma^2_W = 0.4909}$	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$

Otsu’s Thresholding Result



Result of Otsu’s method