CAP 4453 Robot Vision

Dr. Gonzalo Vaca-Castaño

gonzalo.vacacastano@ucf.edu

Administrative details

- Allow grader to review your homework:
  - moazam4453@gmail.com
- Homework 1 review
- Any issues with hw3 ?

Credits

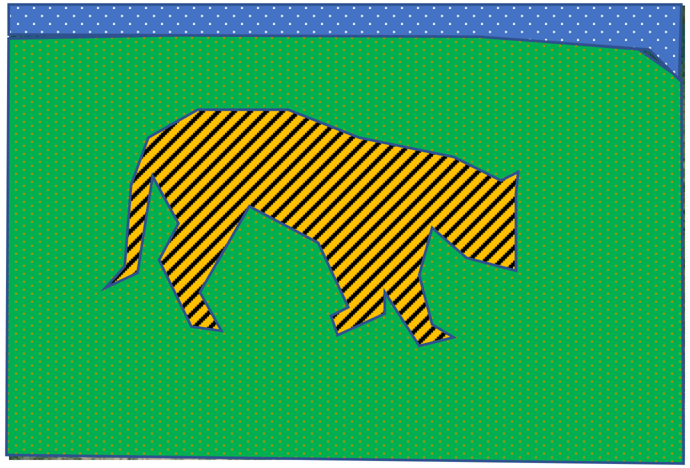
- Some slides comes directly from:
  - Yogesh S Rawat (UCF)
  - Noah Snavely (Cornell)
  - Ioannis (Yannis) Gkioulekas (CMU)
  - Mubarak Shah (UCF)
  - S. Seitz
  - James Tompkin
  - Ulas Bagci
  - L. Lazebnik

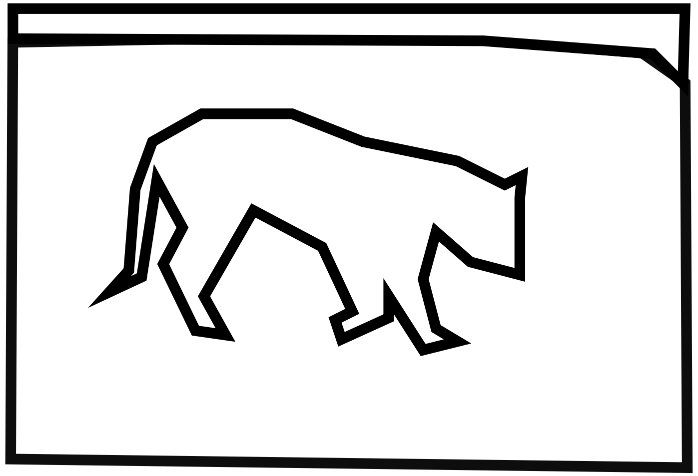
Short Review from last class

Last 2 classes

- Gradient operators
  - Prewit
  - Sobel
- Marr-Hildreth (Laplacian of Gaussian)
- Canny (Gradient of Gaussian)

Regions ⟷ Boundaries





Robot Vision

7. Segmentation I

Outline

- Image segmentation basics
- Thresholding based
  - Binarization
  - Otsu
- Region based
  - Merging
  - Splitting
- Clustering based
  - K-means (SLIC)

Outline

- **Image segmentation basics**
- Thresholding based
  - Binarization
  - Otsu
- Region based
  - Merging
  - Splitting
- Clustering based
  - K-means (SLIC)

Image segmentation

- Partition an image into a collection of set of pixels
  - Meaningful regions (coherent objects)
  - Linear structures (line, curve, …)
  - Shapes (circles, ellipses, …)

Image segmentation

- Content base image retrieval
- Machine vision
- Medical imaging applications
- Object detection (face detection, ..)
- 3D reconstruction
- Object/motion tracking
- …

Image segmentation

- In computer vision, image segmentation is one of the oldest and most widely studied problems
  - Early techniques -> region splitting or merging
  - Recent techniques -> Energy minimization, hybrid methods, and deep learning

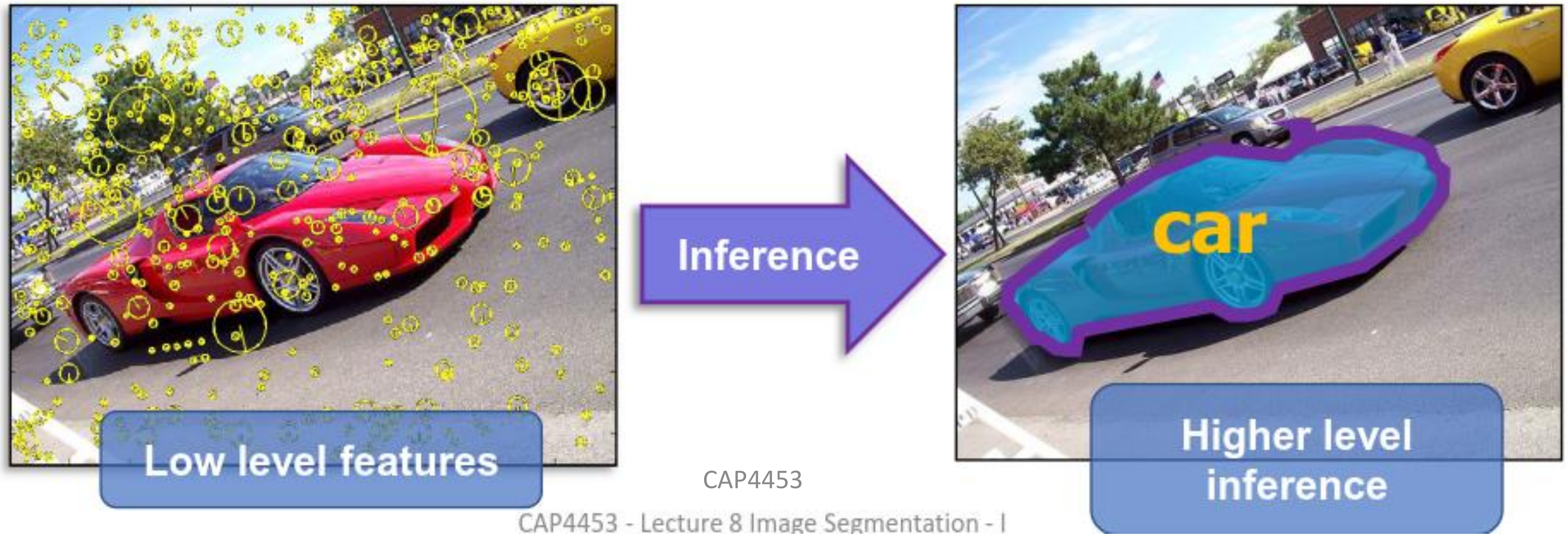


Image segmentation methods

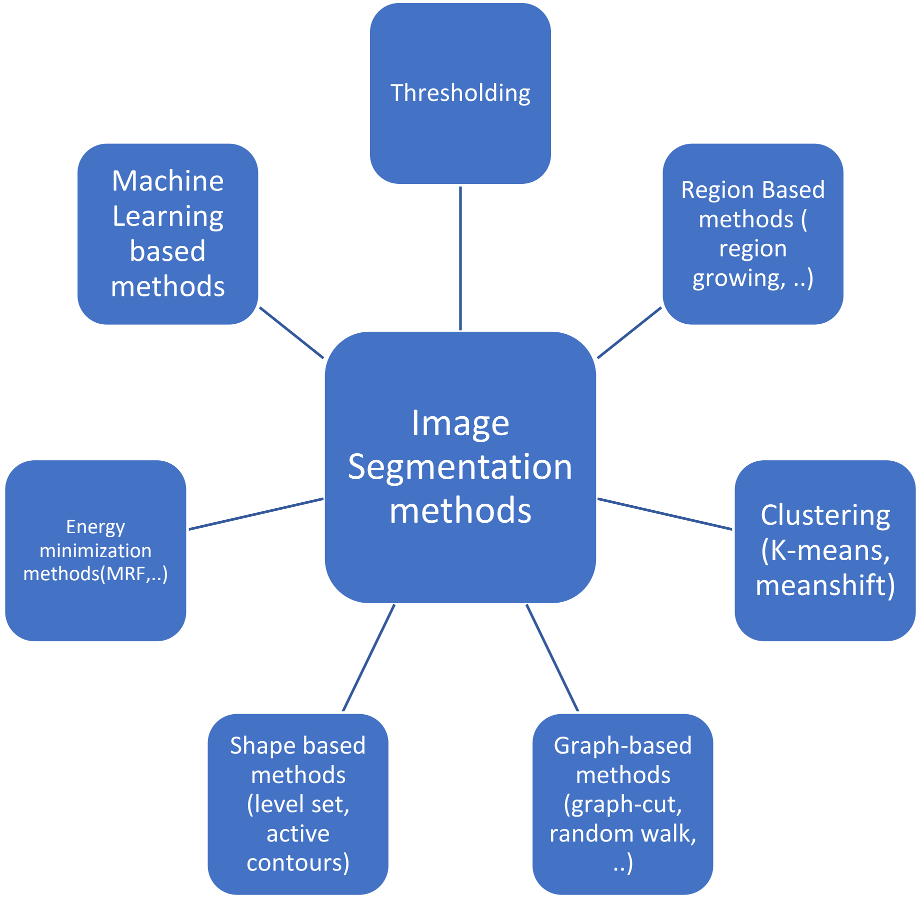


Image segmentation

- Image segmentation partitions an image into regions called segments.

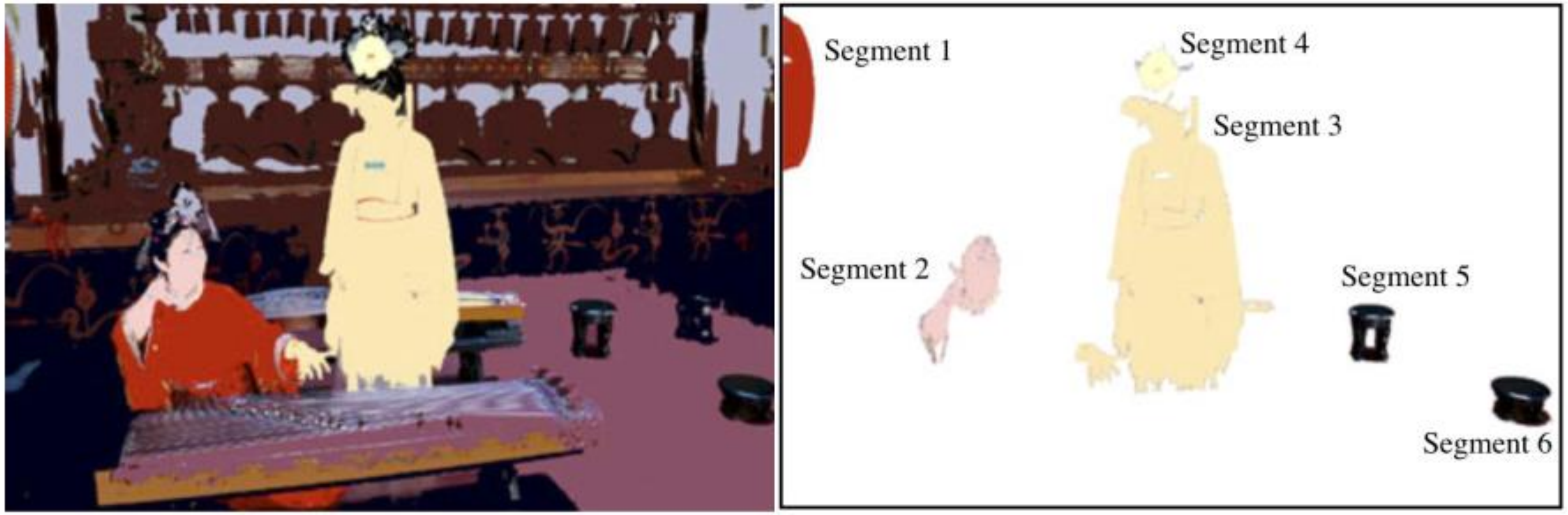
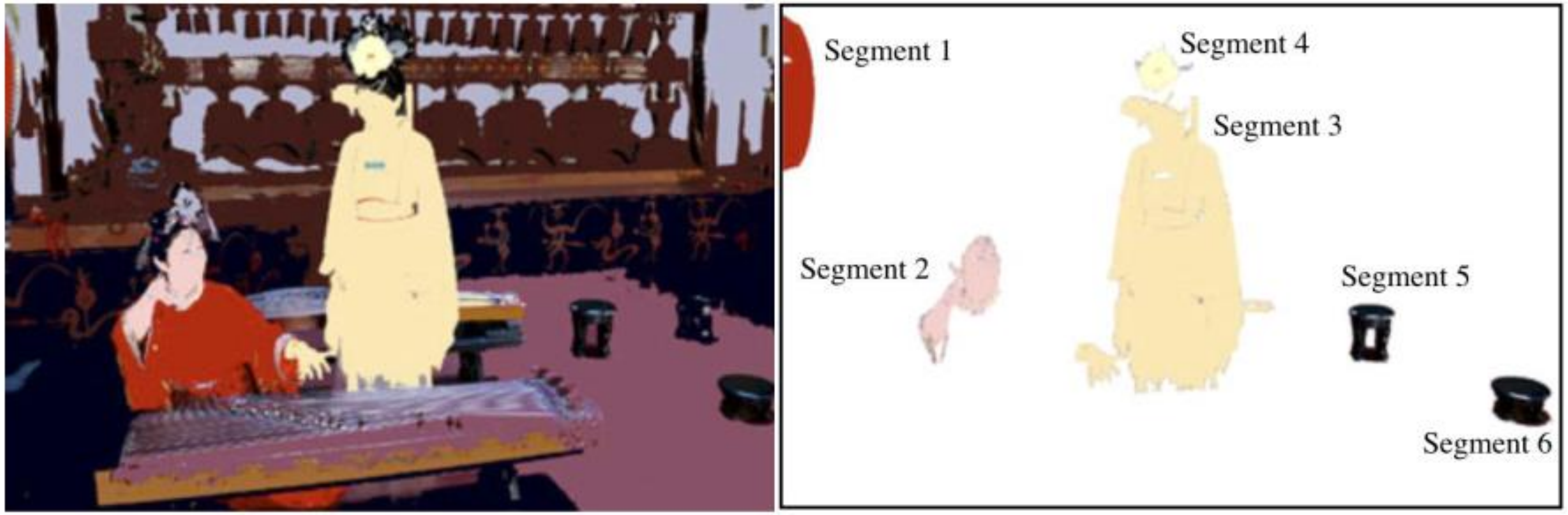


Image segmentation

- Image segmentation partitions an image into regions called segments.



- Image segmentation creates segments of connected pixels by analyzing some similarity criteria:
  - intensity, color, texture, histogram, features

Outline

- Image segmentation basics
- **Thresholding based**
  - **Binarization**
  - Otsu
- Region based
  - Merging
  - Splitting
- Clustering based
  - K-means (SLIC)

Image binarization

- Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image

Image binarization

- Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image

$$J(x,y) = \begin{cases} 0 & \text{if } I(x,y) < T \\ 1 & \text{otherwise.} \end{cases}$$

Image binarization

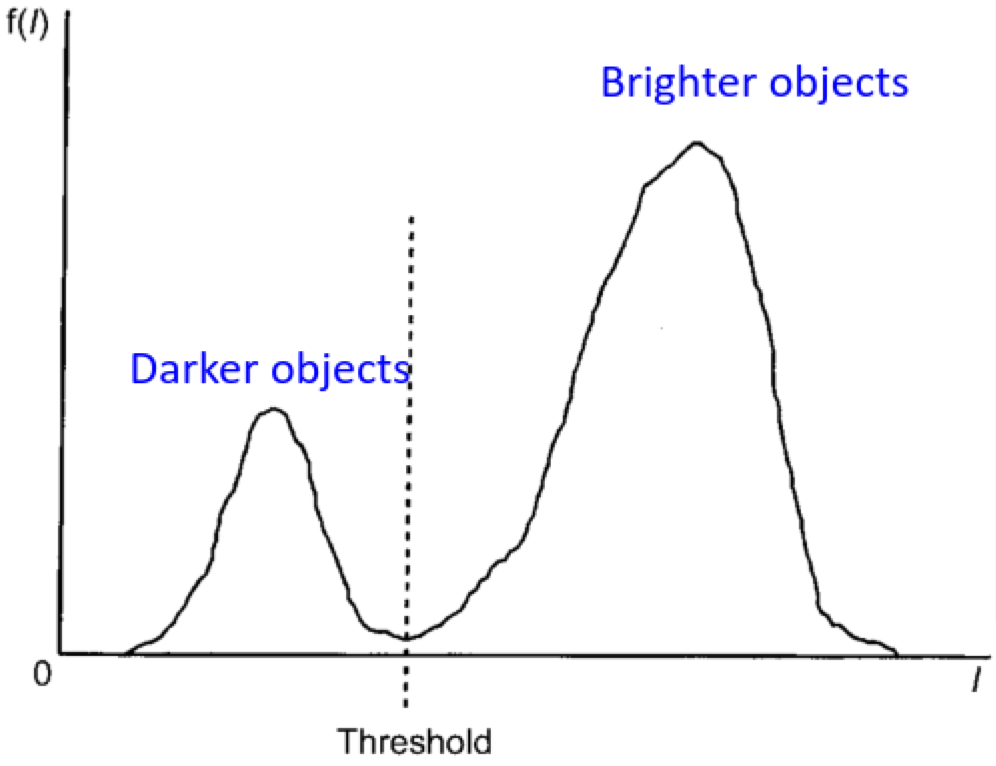
- Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image

$$J(x, y) = \begin{cases} 0 & \text{if } I(x, y) < T \\ 1 & \text{otherwise.} \end{cases}$$

- The global threshold can be identified by an optimization strategy aiming at creating “large” connected regions and at reducing the number of small-sized regions, called artifacts.

Image binarization

- Thresholding: Most frequently employed method for determining threshold is based on histogram analysis of intensity levels



Peak on the left of the histogram corresponds to dark objects

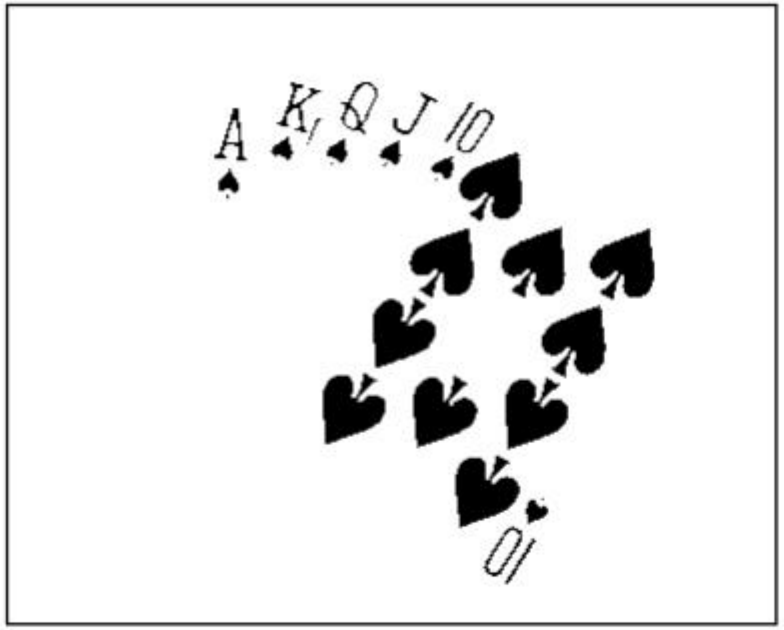
Peak on the right of the histogram corresponds to brigther objects

Difficulties

1. The valley may be so broad that it is difficult to locate a significant minimum
2. Number of minima due to type of details in the image
3. Noise
4. No visible valley
5. Histogram may be multi-modal

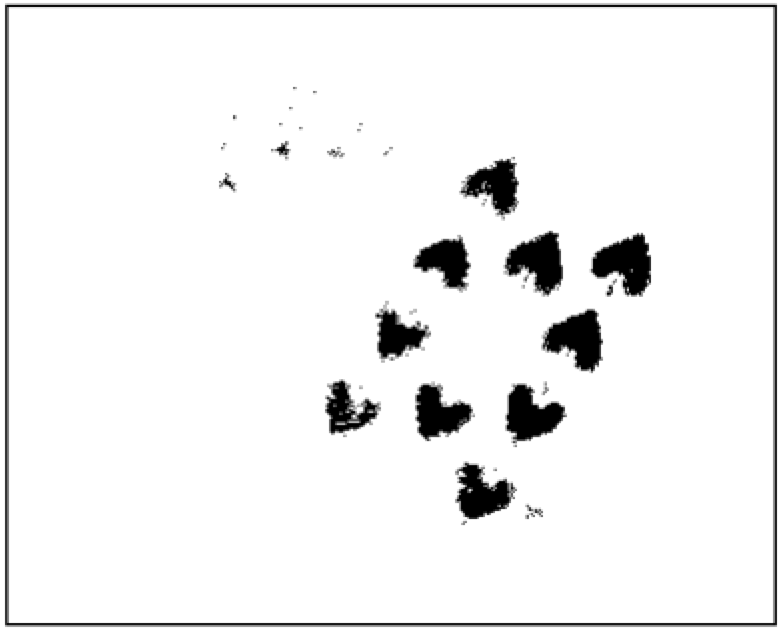
Thresholding examples

Original Image

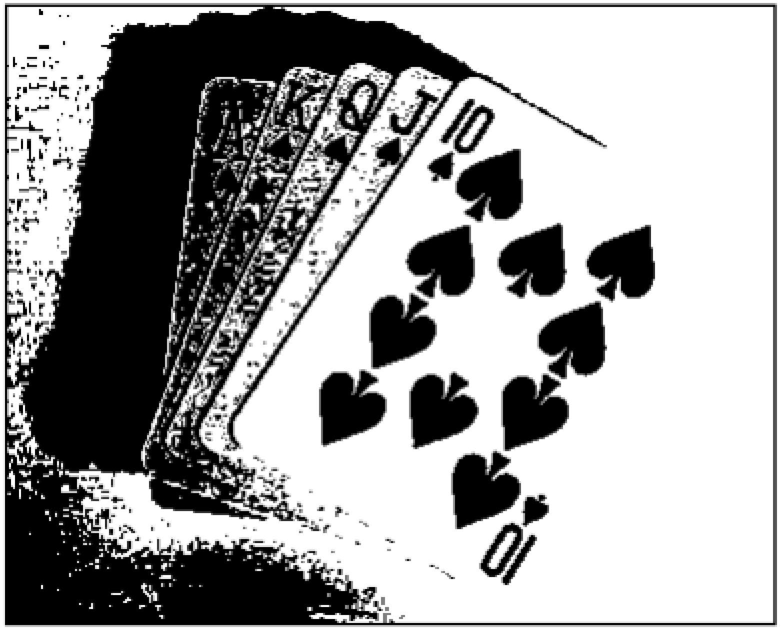


Thresholded Image

Thresholding examples

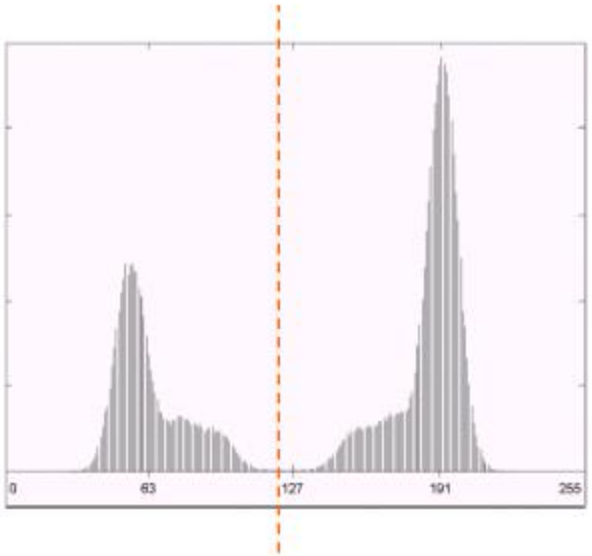


Threshold Too Low

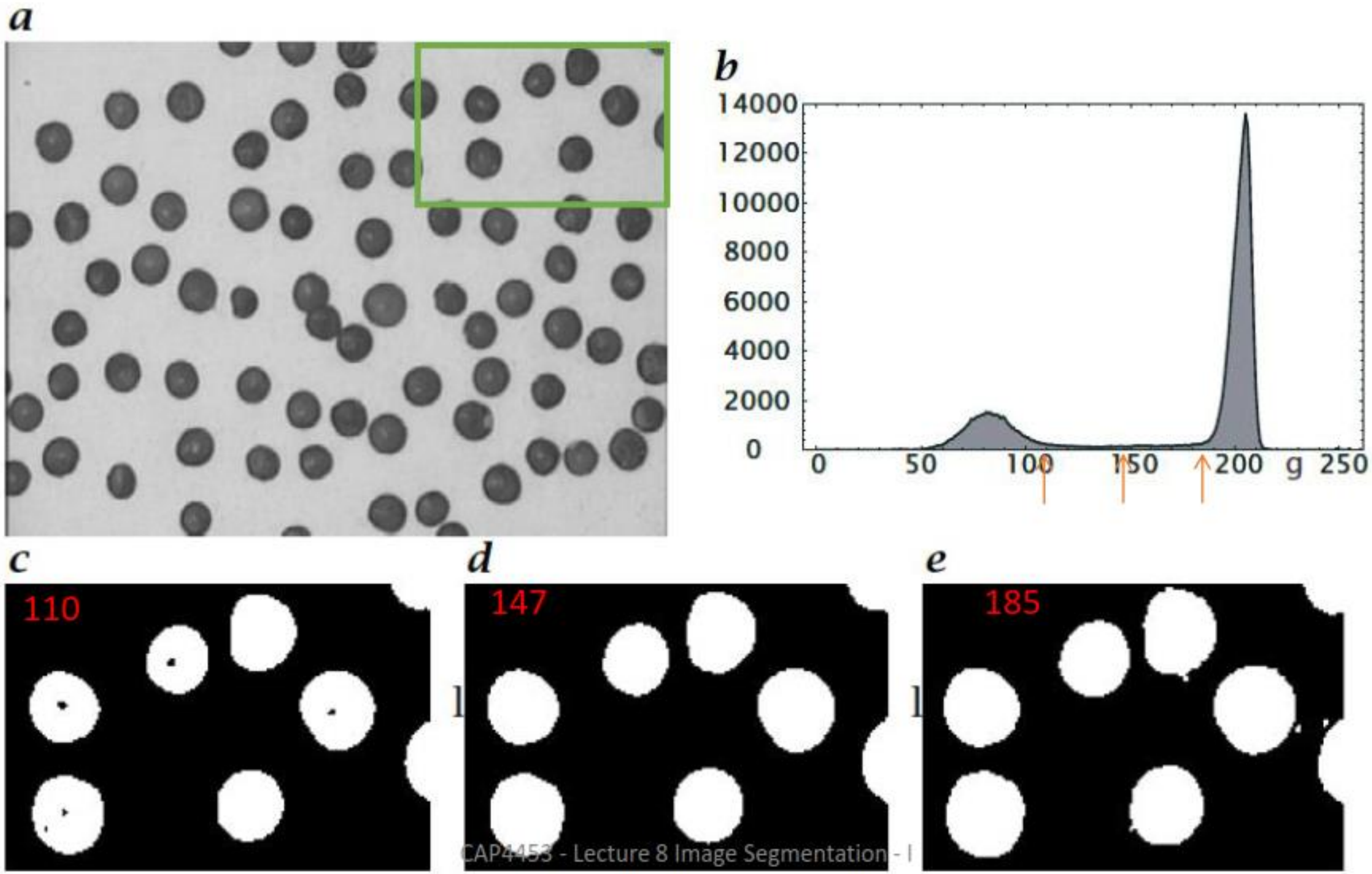


Threshold Too High

Thresholding examples



Thresholding examples

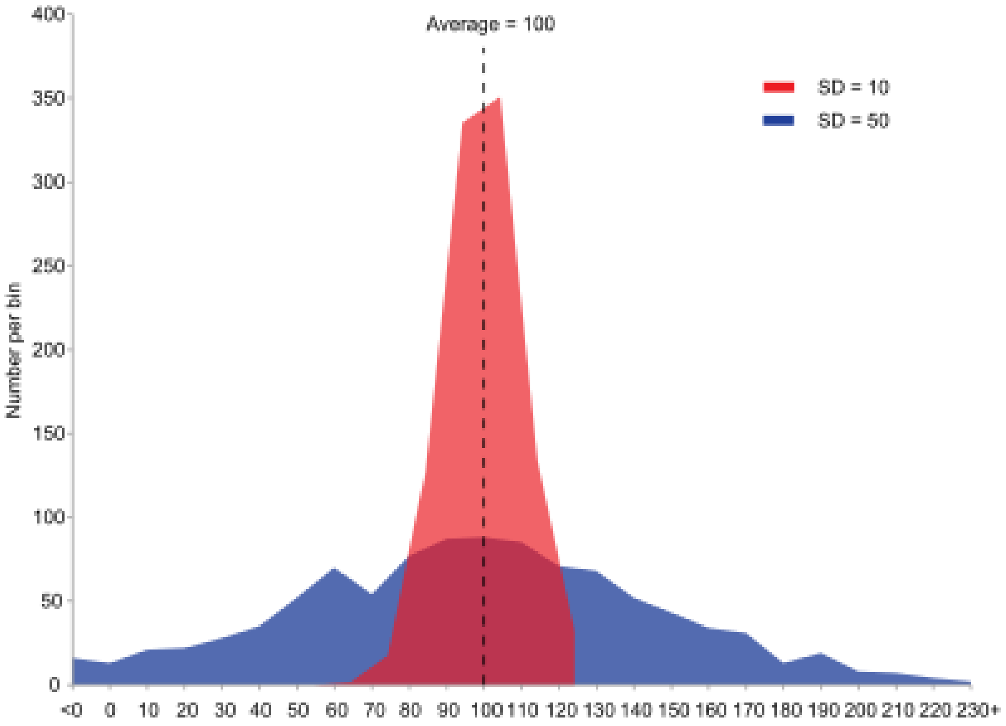


Outline

- Image segmentation basics
- Thresholding based
  - Binarization
  - **Otsu**
- Region based
  - Merging
  - Splitting
- Clustering based
  - K-means (SLIC)

Variance

$$\mathrm{Var}(X) = \mathrm{E}\left[(X-\mu)^2\right] = \sigma_X^2$$



where μ is the expected value. That is,

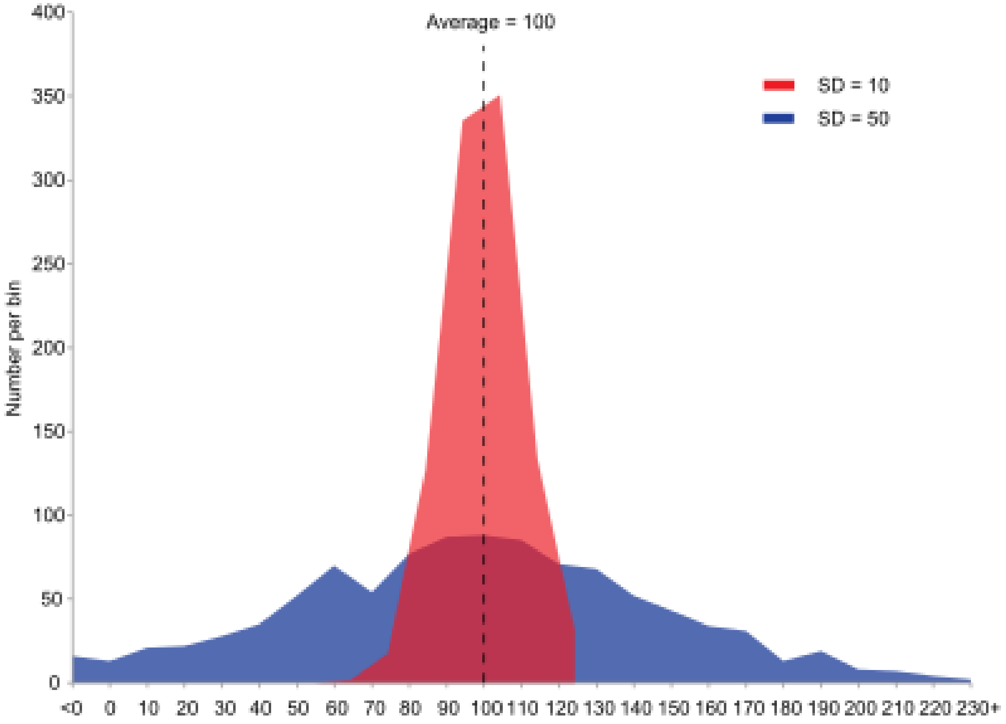
$$\mu = \sum_{i=1}^{n} p_i x_i .$$

$$p_i = \frac{\#values\ in\ i}{sum\ all\ values\ (area\ under\ curve)}$$

[Variance - Wikipedia]

Variance

$$\mathrm{Var}(X) = \mathrm{E}\left[(X-\mu)^2\right] = \sigma_X^2$$



Discrete random variable [edit]

If the generator of random variable X is discrete with probability mass function $x_1 \mapsto p_1, x_2 \mapsto p_2, \ldots, x_n \mapsto p_n$, then

$$\mathrm{Var}(X) = \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2,$$

where μ is the expected value. That is,

$$\mu = \sum_{i=1}^{n} p_i x_i .$$

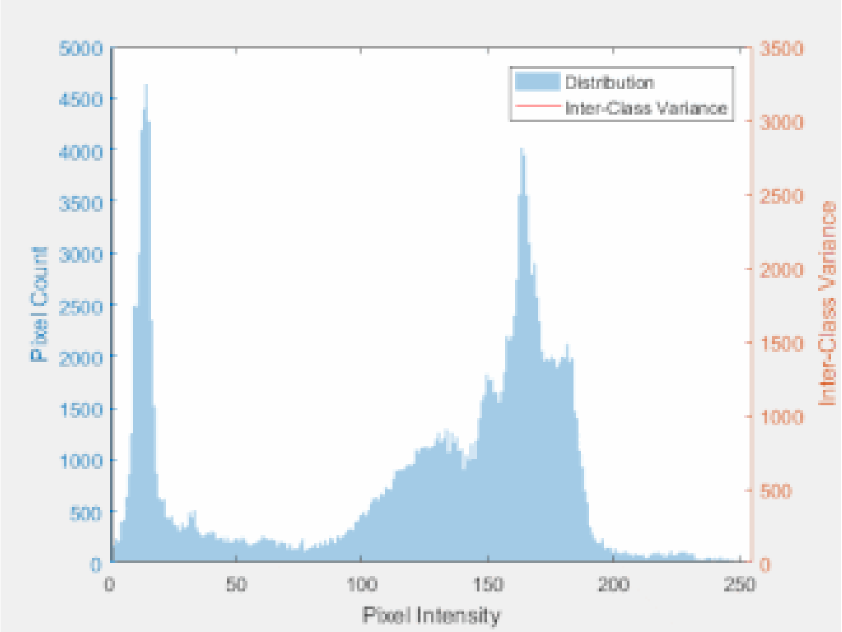
[Variance - Wikipedia]

Otsu thresholding

- Definition: The method uses grey-value histogram of the given image I as input and aims at providing the best threshold (foreground/background)
- Otsu's algorithm selects a threshold that maximizes the between-class variance σ_b^2 or minimize within-class variance σ_w^2

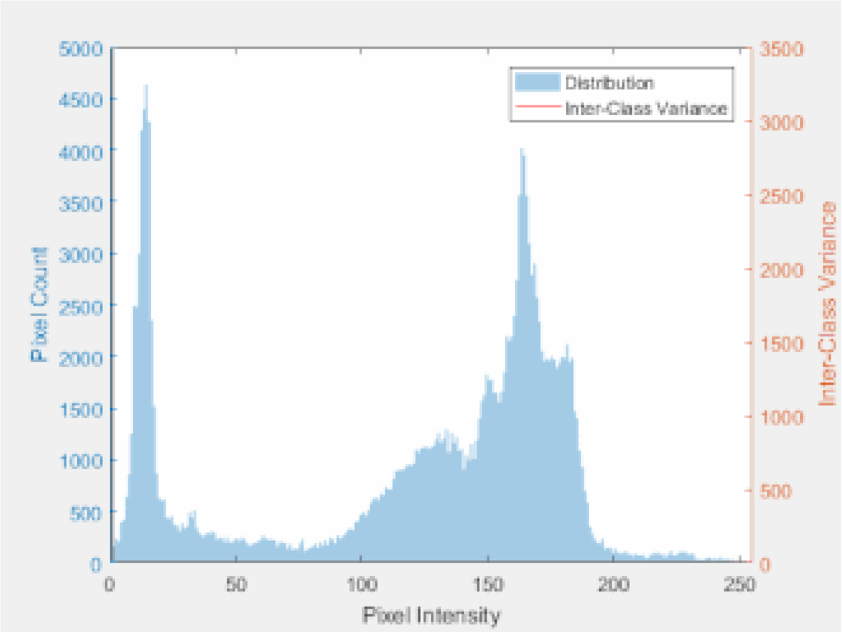
- For each threshold *t* in [0, 255], pixels can be separated into two classes, *C1* and *C2*; those pixels whose *Pi* < *t* are put into *C1*, otherwise into *C2*
- The possibilities of *C1* and *C2* separated by *t*, denoted as *W1* and *W2*, respectively. For example,

  W1 = (#pixels in *C1*) / (total pixels count).
- Given *H*, *W1*, and *W2*, for each *t*, compute the between-class variance σ_b^2 or within-class variance σ_w^2 (σ_b^2 →red curve)
- optimal cut *t** corresponds to *t* whose σ_b^2 is maximum or σ_w^2 is minimum.



Otsu thresholding

- Definition: The method uses grey-value histogram of the given image I as input and aims at providing the best threshold (foreground/background)
- Otsu's algorithm selects a threshold that maximizes the between-class variance σ_b^2.



Option 1: maximum of:

$$\sigma_b^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{w_1(t)} \qquad w_1(t) = \sum_{i=1}^{t} P(i)$$

$$\mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{w_2(t)} \qquad w_2(t) = \sum_{i=t+1}^{I} P(i)$$

Otsu thresholding

- Definition: The method uses grey-value histogram of the given image I as input and aims at providing the best threshold (foreground/background)
- Otsu's algorithm selects a threshold that maximizes the between-class variance or σ_w^2 or minimize within-class variance σ_w^2

Option 2: minimum of:

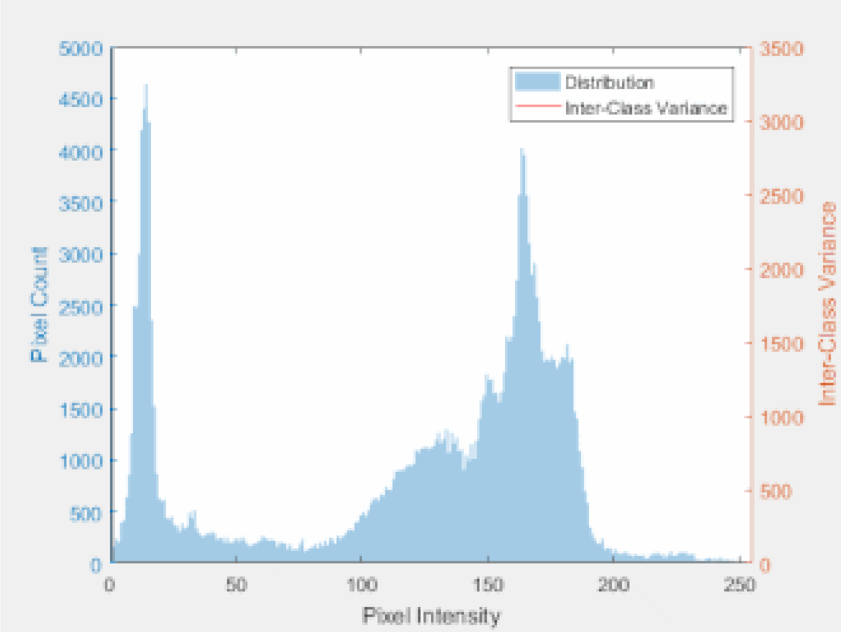
$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

$$w_1(t) = \sum_{i=1}^{t} P(i) \qquad P(i) = \frac{n_i}{n}$$

$$w_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{w_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{w_2(t)}$$



Step by step example

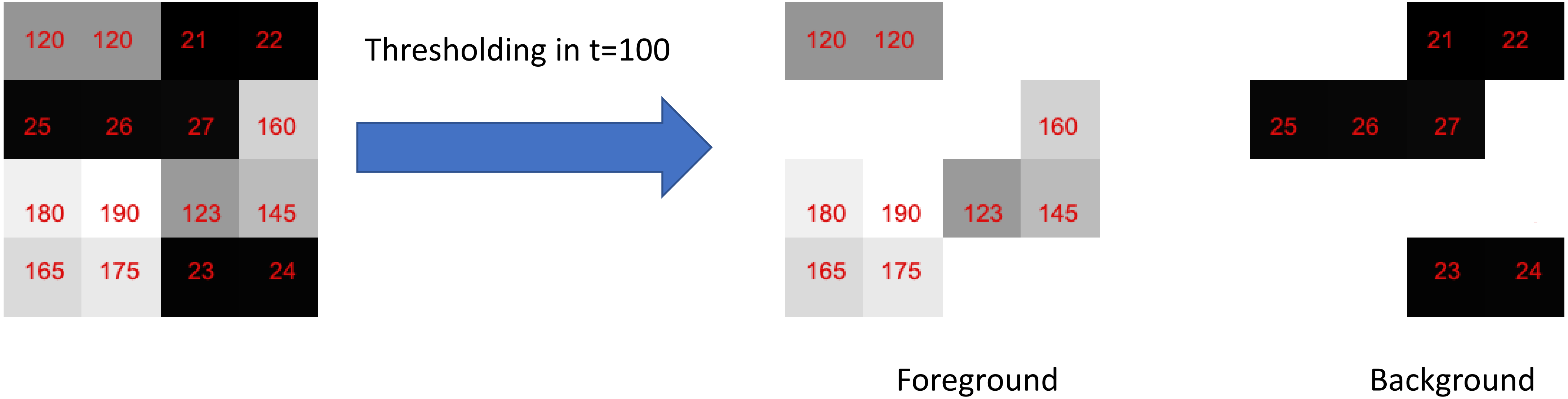
- Find Otsu threshold for this image

120	120	21	22
25	26	27	160
180	190	123	145
165	175	23	24

- By minimizing within-class variance

[Otsu's method for image thresholding explained and implemented – Muthukrishnan]

Step by step example



[Otsu's method for image thresholding explained and implemented – Muthukrishnan]()

Step by step example

120	120	21	22
25	26	27	160
180	190	123	145
165	175	23	24

$P_{all} = 16$

Thresholding in t=100

Foreground

120	120		
			160
180	190	123	145
165	175		

$P_{FG} = 9$

$$\omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}} = 9/16 = 0.56$$

Background

		21	22
25	26	27	
		23	24

$P_{BG} = 7$

$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}} = 7/16 = 0.44$$

[Otsu's method for image thresholding explained and implemented – Muthukrishnan]

Step by step example

120	120	21	22
25	26	27	160
180	190	123	145
165	175	23	24

$P_{all} = 16$

Thresholding in t=100

Foreground

120	120		
			160
180	190	123	145
165	175		

$P_{FG} = 9$

$$\omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}} = 9/16 = 0.56$$

$$\overline{x_{fg}} = \frac{120+120+160+180+190+123+145+165+175}{9} = 153.1$$

Background

		21	22
25	26	27	
		23	24

$P_{BG} = 7$

$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}} = 7/16 = 0.44$$

$$\overline{x_{bg}} = \frac{21+22+25+26+27+23+24}{7} = 24$$

Otsu's method for image thresholding explained and implemented – Muthukrishnan

Step by step example

120	120	21	22
25	26	27	160
180	190	123	145
165	175	23	24

$P_{all} = 16$

Thresholding in t=100

Foreground

120	120		
			160
180	190	123	145
165	175		

$P_{FG} = 9$

$$\omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}} = 9/16 = 0.56$$

$$\overline{x_{fg}} = \frac{120+120+160+180+190+123+145+165+175}{9} = 153.1$$

$$\sigma^2_{fg}(t=100) = \frac{(120-153.1)^2+(120-153.1)^2\ldots\ldots(175-153.1)^2}{9} = 657.43$$

Background

		21	22
25	26	27	
		23	24

$P_{BG} = 7$

$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}} = 7/16 = 0.44$$

$$\overline{x_{bg}} = \frac{21+22+25+26+27+23+24}{7} = 24$$

$$\sigma^2_{bg}(t=100) = \frac{(21-24)^2+(22-24)^2\ldots\ldots(24-24)^2}{7} = 4.0$$

[Otsu’s method for image thresholding explained and implemented – Muthukrishnan]

Step by step example

within-class variance

$$w_1(t)\sigma_1^2(t) \quad + \quad w_2(t)\sigma_2^2(t)$$

120	120	21	22
25	26	27	160
180	190	123	145
165	175	23	24

$P_{all} = 16$

Thresholding in t=100

Foreground

120	120		
			160
180	190	123	145
165	175		

$P_{FG} = 9$

$$\omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}} = 9/16 = 0.56$$

$$\overline{x_{fg}} = \frac{120+120+160+180+190+123+145+165+175}{9} = 153.1$$

$$\sigma_{fg}^2(t=100) = \frac{(120-153.1)^2+(120-153.1)^2 \ldots\ldots (175-153.1)^2}{9} = 657.43$$

Background

		21	22
25	26	27	
		23	24

$P_{BG} = 7$

$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}} = 7/16 = 0.44$$

$$\overline{x_{bg}} = \frac{21+22+25+26+27+23+24}{7} = 24$$

$$\sigma_{bg}^2(t=100) = \frac{(21-24)^2+(22-24)^2 \ldots\ldots (24-24)^2}{7} = 4.0$$

[Otsu's method for image thresholding explained and implemented – Muthukrishnan]

Step by step example

within-class variance

$$w_1(t)\sigma_1^2(t) \quad + \quad w_2(t)\sigma_2^2(t)$$

$$0.44 * 4.0 \quad + \quad 0.56 * 657.43 \quad = \quad 369.9208$$

Thresholding in t=100

120	120	21	22
25	26	27	160
180	190	123	145
165	175	23	24

$P_{all} = 16$

Foreground

120	120		
			160
180	190	123	145
165	175		

$P_{FG} = 9$

$$\omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}} = 9/16 = 0.56$$

$$\overline{x_{fg}} = \frac{120+120+160+180+190+123+145+165+175}{9} = 153.1$$

$$\sigma_{fg}^2(t=100) = \frac{(120-153.1)^2+(120-153.1)^2 \ldots\ldots (175-153.1)^2}{9} = 657.43$$

Background

		21	22
25	26	27	
		23	24

$P_{BG} = 7$

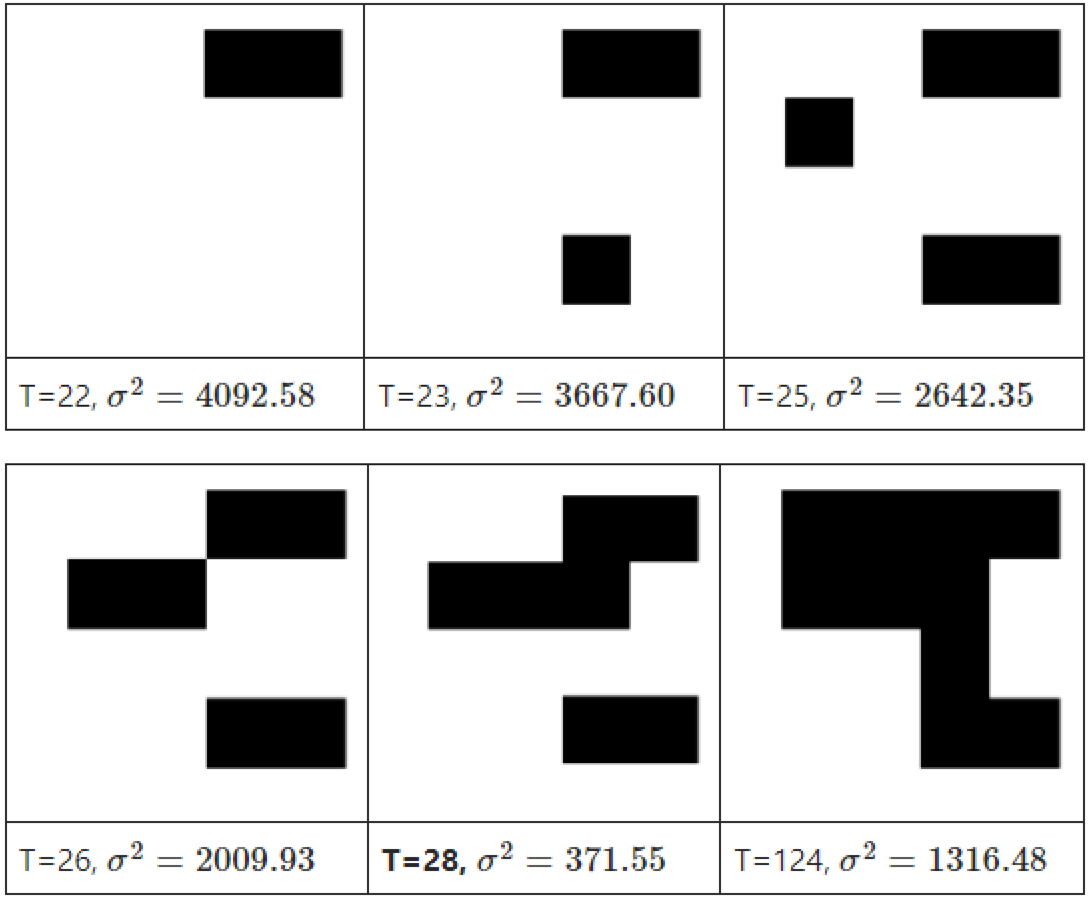
$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}} = 7/16 = 0.44$$

$$\overline{x_{bg}} = \frac{21+22+25+26+27+23+24}{7} = 24$$

$$\sigma_{bg}^2(t=100) = \frac{(21-24)^2+(22-24)^2 \ldots\ldots (24-24)^2}{7} =$$

[Otsu's method for image thresholding explained and implemented – Muthukrishnan]

Step by step (otsu thresholding)



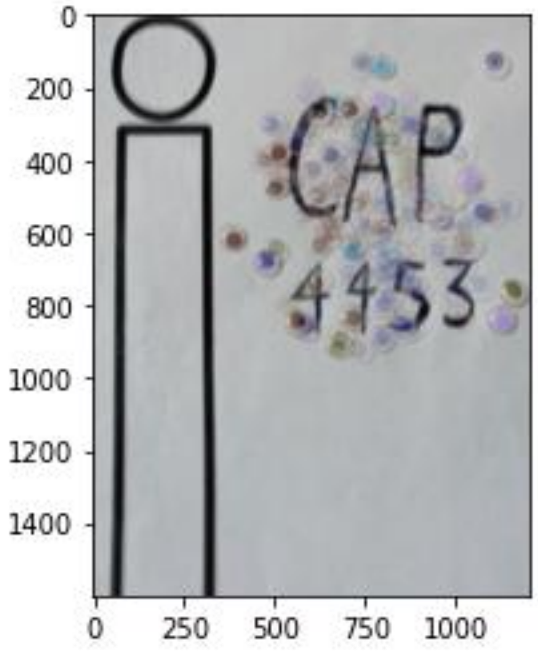
Minimize within-class variance

- The value of variance remains the same from 28 and 120.
- within-class variance is least at t=28 or more precisely between 28 to 120.
- Otsu threshold = 28.

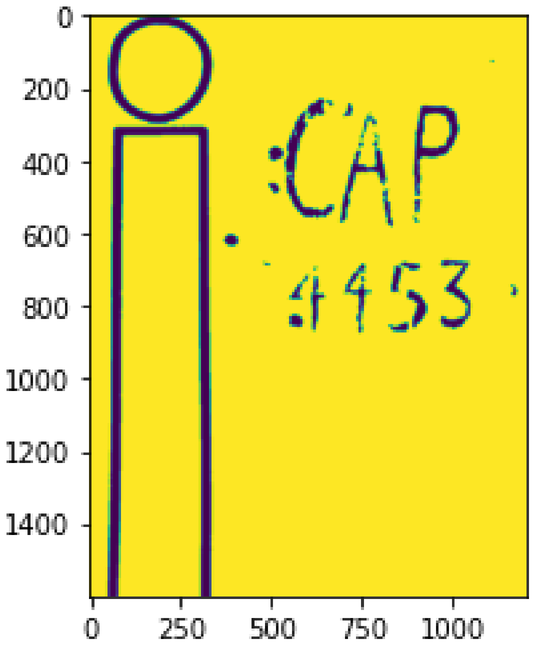
Otsu threshold implementation

```python
# Set total number of bins in the histogram
bins_num = 256

# Get the image histogram
hist, bin_edges = np.histogram(image, bins=bins_num)

# Get normalized histogram if it is required
if is_normalized:
    hist = np.divide(hist.ravel(), hist.max())

# Calculate centers of bins
bin_mids = (bin_edges[:-1] + bin_edges[1:]) / 2.

# Iterate over all thresholds (indices) and get the probabilities w1(t), w2(t)
weight1 = np.cumsum(hist)
weight2 = np.cumsum(hist[::-1])[::-1]

# Get the class means mu0(t)
mean1 = np.cumsum(hist * bin_mids) / weight1
# Get the class means mu1(t)
mean2 = (np.cumsum((hist * bin_mids)[::-1]) / weight2[::-1])[::-1]

inter_class_variance = weight1[:-1] * weight2[1:] * (mean1[:-1] - mean2[1:]) ** 2

# Maximize the inter_class_variance function val
index_of_max_val = np.argmax(inter_class_variance)

threshold = bin_mids[:-1][index_of_max_val]
print("Otsu's algorithm implementation thresholding result: ", threshold)
```

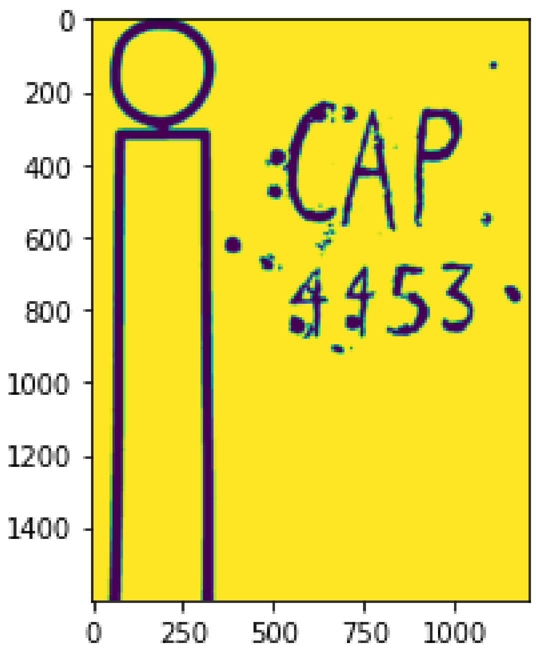
Otsu threshold implementation



Original image



Manually
Th =90



otsu
Th =127

[Otsu's Thresholding Technique | LearnOpenCV]

Otsu threshold implementation

```python
# Applying Otsu's method setting the flag value into cv.THRESH_OTSU.
# Use a bimodal image as an input.
# Optimal threshold value is determined automatically.
otsu_threshold, image_result = cv2.threshold(
    image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU,
)
print("Obtained threshold: ", otsu_threshold)
```

```
Obtained threshold:  132.0
```

[Otsu's Thresholding Technique | LearnOpenCV]

Otsu thresholding example

The math !

$$\mathrm{Var}(X) = \mathrm{E}\left[(X - \mathrm{E}[X])^2\right]$$
$$= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right]$$
$$= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2$$
$$= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2$$

$$\sigma_{total}^2 = E[(X - E[X])^2] = E[X_{total}^2] - \mu_{total}^2 \qquad (1)$$

$$P_i = \frac{n_i}{n_{total}}$$

When i<=t $\quad P_i^1 = \dfrac{n_i}{n_1} \quad P_i = \dfrac{n_1 P_i^1}{n_{total}} = w_1(t) P_i^1$

When t<i<T $\quad P_i^2 = \dfrac{n_i}{n_2} \quad P_i = \dfrac{n_2 P_i^2}{n_{total}} = w_2(t) P_i^2$

$$E\left[X_{total}^2\right] = \sum_{i=1}^{T} P_i\, x_i^2 = \sum_{i=1}^{t} P_i\, x_i^2 + \sum_{i=t+1}^{T} P_i\, x_i^2 = w_1(t) \sum_{i=1}^{t} P_i^1\, x_i^2 + w_2(t) \sum_{i=t+1}^{T} P_i^1\, x_i^2 = \mathrm{w_1(t)E[X_1^2]} + w_2(t) E[X_2^2] \qquad (2)$$

$$\mu_{total}^2 = (w_1(t)\mu_1 + w_2(t)\mu_2)^2 = w_1^2\mu_1^2 + 2w_1w_2\mu_1\mu_2 + w_2^2\mu_2^2 = w_1(1 - w_2)\mu_1^2 + 2w_1w_2\mu_1\mu_2 + w_2(1 - w_1)\mu_2^2 \qquad (3)$$

The math !

$$\mathrm{Var}(X) = \mathrm{E}\left[(X - \mathrm{E}[X])^2\right]$$
$$= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right]$$
$$= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2$$
$$= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2$$

$$\sigma_{total}^2 = E[(X - E[X])^2] = E[X_{total}^2] - \mu_{total}^2 \qquad (1)$$

$$E\left[X_{total}^2\right] = \mathrm{w}_1(t)\mathrm{E}[\mathrm{X}_1^2] + w_2(t)E[X_2^2] \qquad (2)$$

$$\mu_{total}^2 = w_1(1 - w_2)\mu_1^2 + 2w_1w_2\mu_1\mu_2 + w_2(1 - w_1)\mu_2^2 \qquad (3)$$

$$\sigma_{total}^2 = \mathrm{w}_1\mathrm{E}[\mathrm{X}_1^2] + w_2E[X_2^2] - [w_1\mu_1^2 + w_1w_2\mu_1^2 + 2w_1w_2\mu_1\mu_2 + w_2\mu_2^2 - w_1w_2\mu_2^2]$$

$$\sigma_{total}^2 = \mathrm{w}_1\mathrm{E}[\mathrm{X}_1^2] - w_1\mu_1^2 \quad + \quad w_2E[X_2^2] - w_2\mu_2^2 \quad + \quad [-w_1w_2\mu_1^2 - 2w_1w_2\mu_1\mu_2 + w_1w_2\mu_2^2]$$

$$\sigma_{total}^2 = \mathrm{w}_1(t)(\mathrm{E}[\mathrm{X}_1^2] - \mu_1^2) \quad + \quad w_2(t)(E[X_2^2] - \mu_2^2) \quad + \quad w_1(t)w_2(t)(\mu_2^2 - \mu_1^2) \qquad (4)$$

The math !

$$
\begin{aligned}
\mathrm{Var}(X) &= \mathrm{E}\left[(X - \mathrm{E}[X])^2\right] \\
&= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right] \\
&= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2 \\
&= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2
\end{aligned}
$$

$$\sigma_{total}^2 = E[(X - E[X])^2] = E[X_{total}^2] - \mu_{total}^2 \qquad (1)$$

$$\sigma_{total}^2 = \mathrm{w}_1(t)(\mathrm{E}[\mathrm{X}_1^2] - \mu_1^2) \quad + \quad w_2(t)(E[X_2^2] - \mu_2^2) \quad + \quad w_1(t)w_2(t)(\mu_2^2 - \mu_1^2)$$

$$\sigma_{total}^2 = \underbrace{\mathrm{w}_1(t)\sigma_1^2(t) \quad + \quad \mathrm{w}_2(t)\sigma_2^2(t)}_{\text{within-class variance}} \quad + \quad \underbrace{w_1(t)w_2(t)(\mu_2^2(t) - \mu_1^2(t))}_{\text{between-class variance}}$$

fixed

Minimize

Maximize

Questions?