Thresholding-based Segmentation

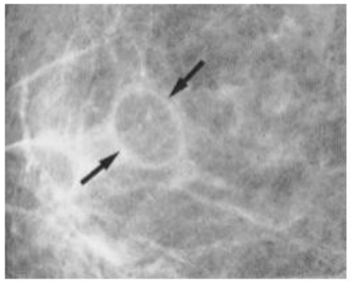
Dr. Tran Anh Tuan,

Faculty of Mathematics and Computer Science,

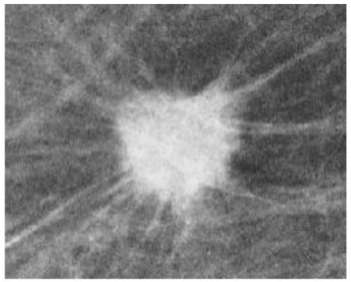
University of Science, HCMC

Context

- Segmentation decomposes the image into parts for further analysis
  - Example: background subtraction in human motion analysis
- Once the region of interest is segmented, the representation space can be changed (from imagespace to feature space)



Circumscribed (benign) lesions in digital mammography



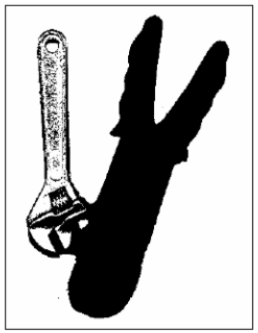
Spiculated (malignant) lesions in digital mammography

What is segmentation ?

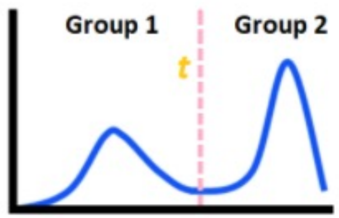
- Partitioning an image into regions corresponding to objects
- All pixels in a region share a common property
- Simplest property that pixels can share: intensity
- Thresholding=separation of light and dark regions

Original

Threshold = 50

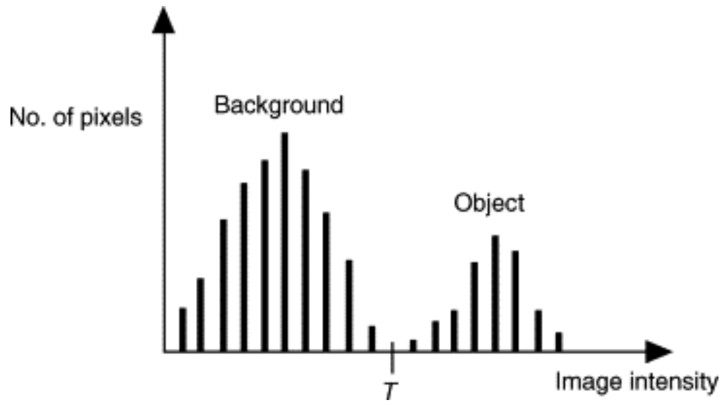


Threshold = 75



Global Thresholding

- Global thresholding is based on the assumption that the image has a bimodal histogram and, therefore, the object can be extracted from the background by a simple operation that compares image values with a threshold value T [32, 132]. Suppose that we have an image f(x,y) with the histogram



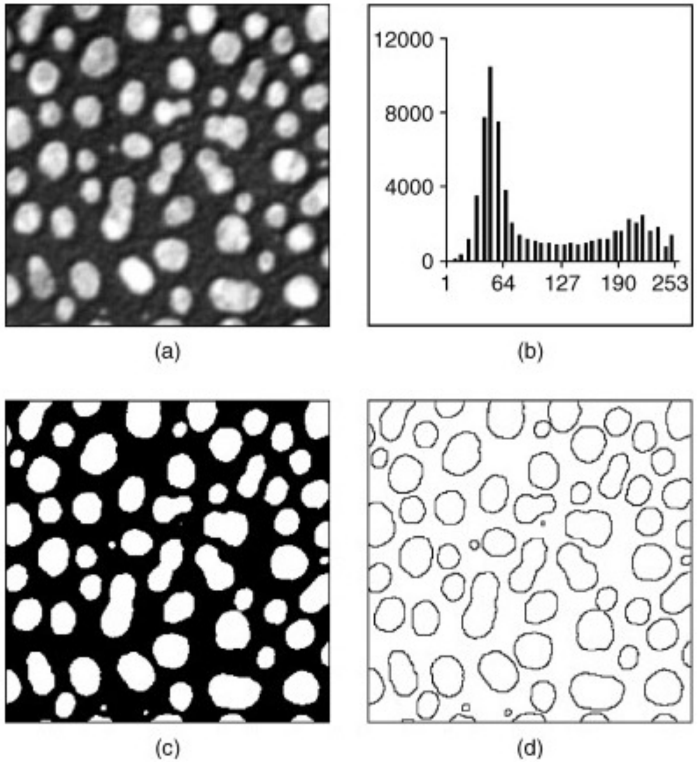
Global Thresholding

- The object and background pixels have gray levels grouped into two dominant modes. One obvious way to extract the object from the background is to select a threshold T that separates these modes.
- The thresholded image g(x,y) is defined as g(x, y)
- The result of thresholding is a binary image, where pixels with intensity value of 1 correspond to objects, whereas pixels with value 0 correspond to the background.

$$g\left(x,y\right)=\begin{cases}1 & \text{if } (x,y)>T \\ 0 & \text{if } (x,y)\leq T\end{cases}.$$

Global Thresholding

Figure shows the result of segmentation by thresholding. The original image Figure a contains white cells on a black background. Pixel intensities vary between 0 and 255. The threshold T = 127 was selected as the minimum between two modes on a histogram Figure b, and the result of segmentation is shown in Figure c, where pixels with intensity values higher than 127 are shown in white. In the last step Figure d the edges of the cells were obtained by a 3 × 3 Laplacian (second-order derivative), which was applied to the thresholded image on Figure c.



2nd order derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

$$\frac{\partial^2 f}{\partial y^2} = f(y+1) + f(y-1) - 2f(y)$$

Convolution Kernel

$$\begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -4 & 1 \\ \hline & 1 & \\ \hline \end{array}$$

Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Global Thresholding

- **P-tile method**

1. A priori information: object is brighter/darker than background and occupies a certain known percentile 1/p from the total image area (example: printed text sheet)
2. We set the threshold by finding the intensity level such that 1/p image pixels are below this value
3. We use the cumulative histogram
4. T verifies the equation c(T)=1/p (for a dark foreground)
5. c(T)=1-1/p (for a bright foreground)

$$c(g) = \sum_{k=0}^{g} h(k)$$

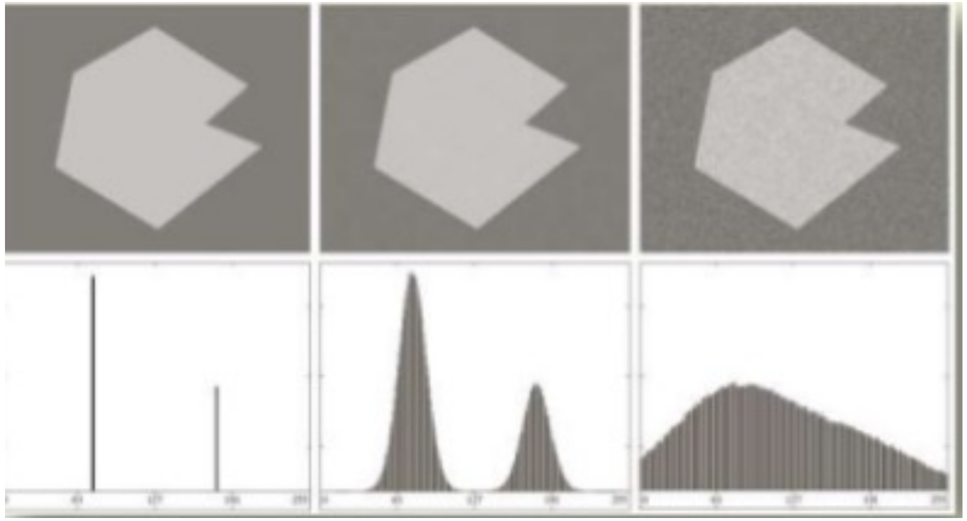
$$h(k) = \frac{n_k}{n}$$

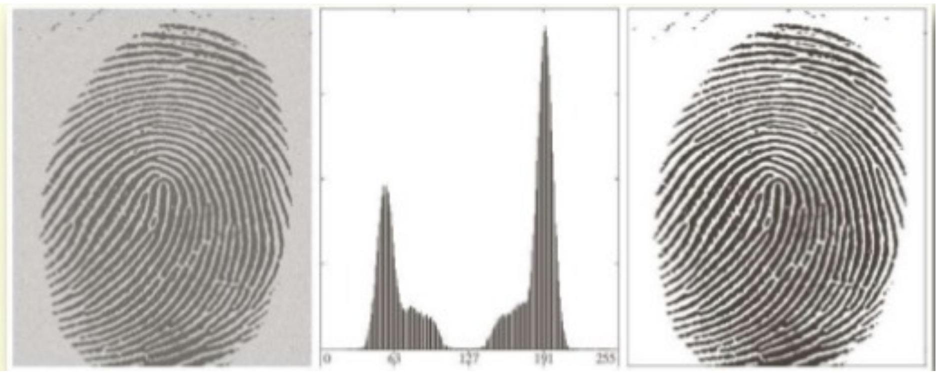
Global Thresholding

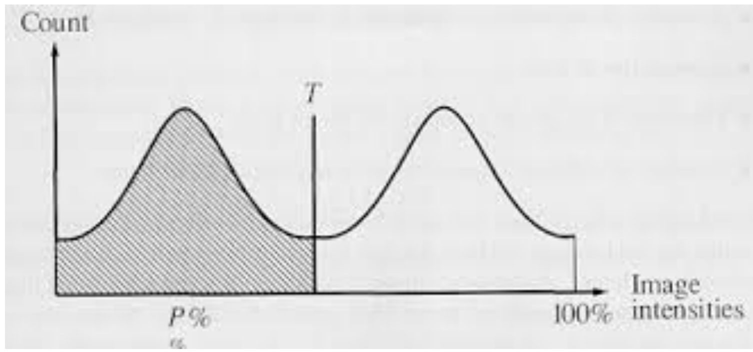
P-tile method

Original image

P-tile method







Global Thresholding

- **Otsu Method**
- Find the threshold that ***minimizes the weighted within-class variance***.
- Equivalent to: ***maximizing the between-class variance.***
- Operates directly on the gray level histogram
- It is fast (once the histogram is computed).

Global Thresholding

- **Otsu Method Assumptions**
- Histogram (and the image) are ***bimodal.***
- **No use of *spatial coherence*,** nor any other notion of object structure.
- Assumes **stationary statistics**, but can be modified to be locally adaptive
- Assumes **uniform illumination** (implicitly), so the bimodal brightness behavior arises from object appearance differences only.

Otsu's method: Formulation

The *weighted within-class variance* is:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where the class probabilities are estimated as:

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{L} P(i)$$

And the class means are given by:

$$\mu_1(t) = \frac{1}{q_1(t)} \sum_{i=1}^{t} iP(i) \qquad \mu_2(t) = \frac{1}{q_2(t)} \sum_{i=t+1}^{L} iP(i)$$

Otsu’s method: Formulation

Finally, the individual class variances are:

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

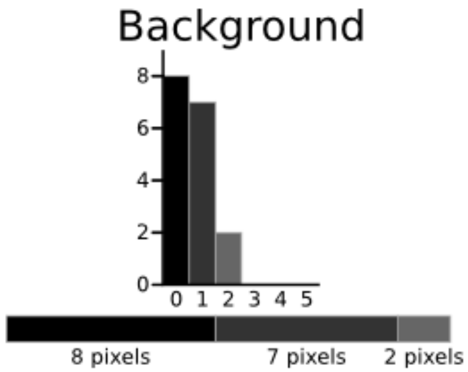
$$\sigma_2^2(t) = \sum_{i=t+1}^{L} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

$$\boxed{\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)}$$

Run through the full range of *t* values and pick the value that minimizes $\sigma_w^2(t)$

Is this algorithm first enough?

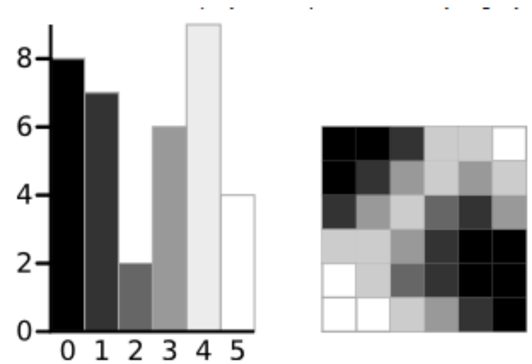
The calculations for finding the foreground and background variances (the measure of spread) for a single threshold are now shown. In this case the threshold value is 3.



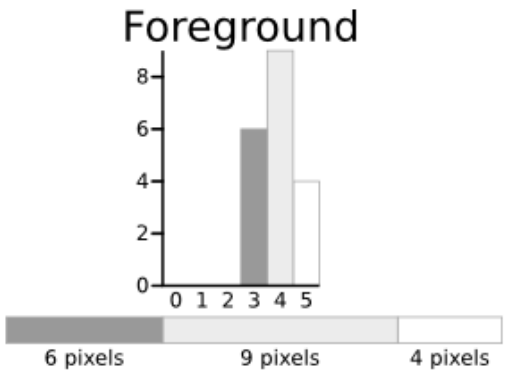
$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

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

$$\begin{aligned}\text{Variance} \quad \sigma_b^2 &= \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17}\\ &= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17}\\ &= 0.4637\end{aligned}$$



A 6-level greyscale image and its histogram



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

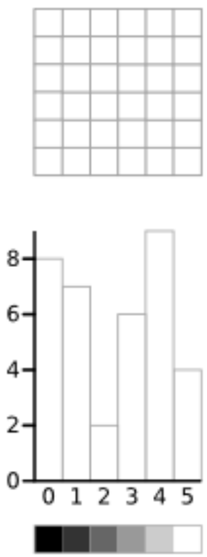
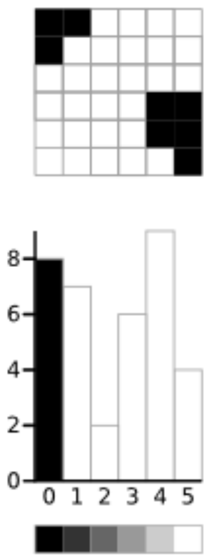
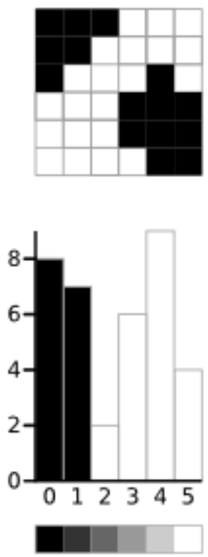
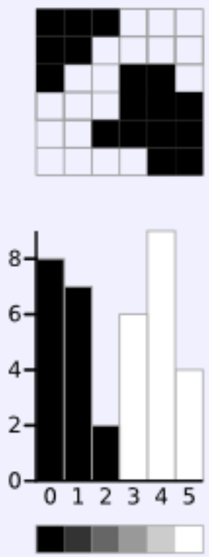
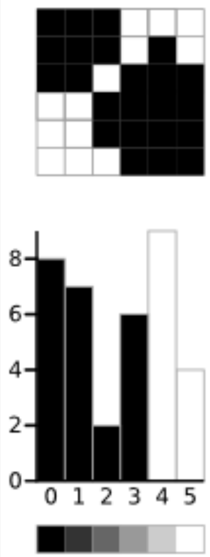
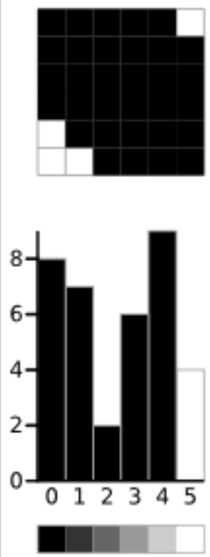
$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

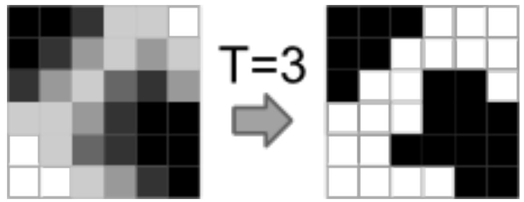
$$\begin{aligned}\text{Variance} \quad \sigma_f^2 &= \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}\\ &= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}\\ &= 0.5152\end{aligned}$$

The next step is to calculate the 'Within-Class Variance'. This is simply the sum of the two variances multiplied by their associated weights.

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

This final value is the 'sum of weighted variances' for the threshold value 3. This same calculation needs to be performed for all the possible threshold values 0 to 5. The table below shows the results for these calculations. The highlighted column shows the values for the threshold calculated above.

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	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	$\sigma^2_W = 0.4909$	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$



- ## A Faster Approach

By a bit of manipulation, you can calculate what is called the *between class* variance, which is far quicker to calculate. Luckily, the threshold with the maximum *between class* variance also has the minimum *within class* variance. So it can also be used for finding the best threshold and therefore due to being simpler is a much better approach to use.

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 \quad \text{(as seen above)}$$

$$\begin{aligned}\text{Between Class Variance} \quad \sigma_B^2 &= \sigma^2 - \sigma_W^2 \\ &= W_b(\mu_b - \mu)^2 + W_f(\mu_f - \mu)^2 \quad (\text{where } \mu = W_b\,\mu_b + W_f\,\mu_f) \\ &= W_b\,W_f\,(\mu_b - \mu_f)^2\end{aligned}$$

The table below shows the different variances for each threshold value.

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
Within Class Variance	σ^2_W = 3.1196	σ^2_W = 1.5268	σ^2_W = 0.5561	**σ^2_W = 0.4909**	σ^2_W = 0.9779	σ^2_W = 2.2491
Between Class Variance	σ^2_B = 0	σ^2_B = 1.5928	σ^2_B = 2.5635	**σ^2_B = 2.6287**	σ^2_B = 2.1417	σ^2_B = 0.8705

The total variance does not depend on threshold (obviously).
For any given threshold, the **total variance** is the weighted sum of the within-class variances
The ***between class variance***, which is the sum of weighted squared distances between the class means and the global mean.

Total variance

The total variance can be expressed as

$$\sigma^2 = \underbrace{\sigma_w^2(t)}_{\text{Within-class, from before}} + \underbrace{q_1(t)[1-q_1(t)][\mu_1(t)-\mu_2(t)]^2}_{\text{Between-class, } \sigma_B^2(t)}$$

Minimizing the within-class variance is the same as maximizing the between-class variance.

compute the quantities in $\sigma_B^2(t)$ *recursively* as we run through the range of *t* values.

Recursive algorithm

Initialization... $q_1(1) = P(1);\ \mu_1(0) = 0$

Recursion...

$$q_1(t+1) = q_1(t) + P(t+1)$$

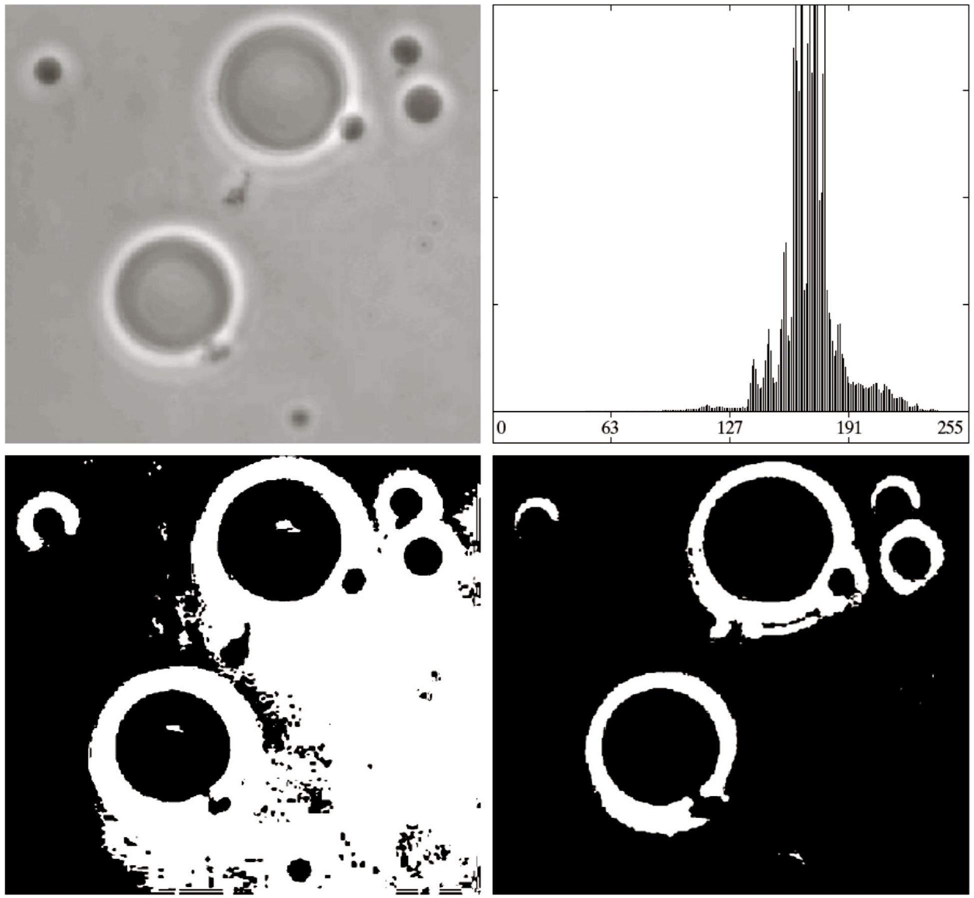
$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)}$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$

Example

Input image

Histogram

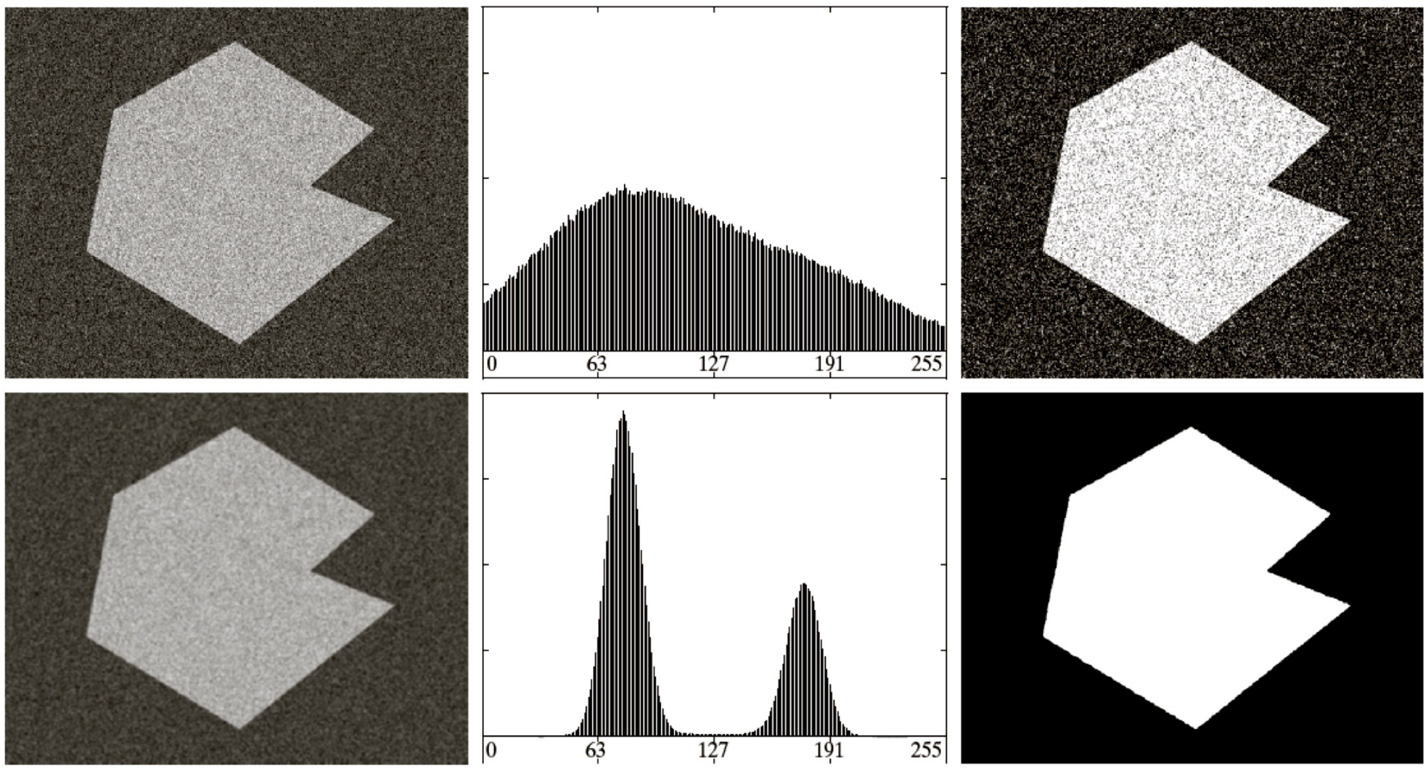


Global thresholding

Otsu's method

Example (in presence of noise)

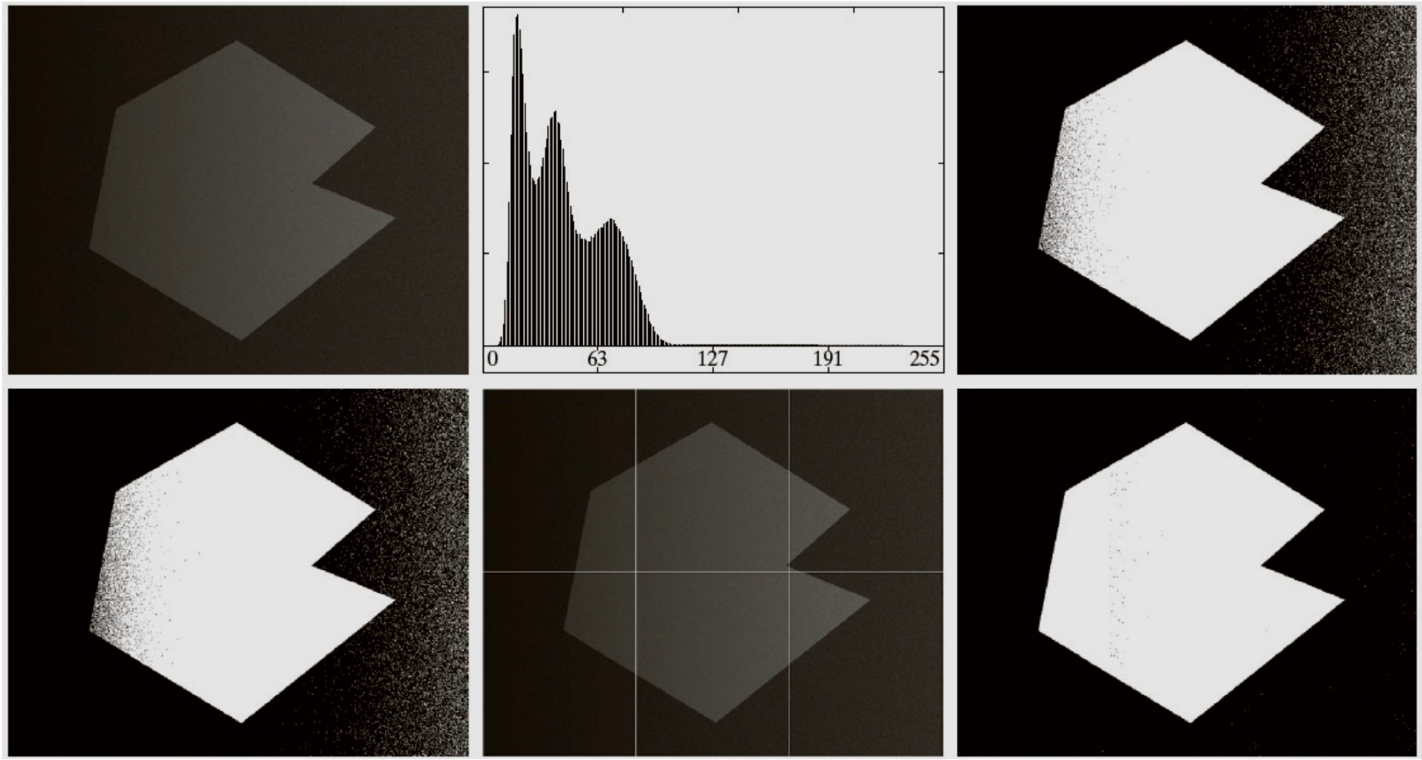
Input image **Histogram** **Global threshold**



Smoothened image **Histogram** **Otsu's method**

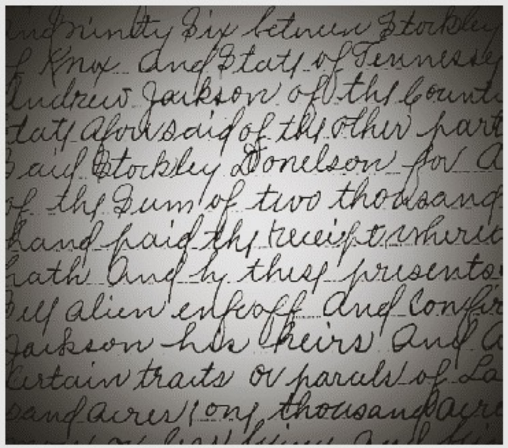
Image partitioning

Input image **Histogram** **Global threshold**

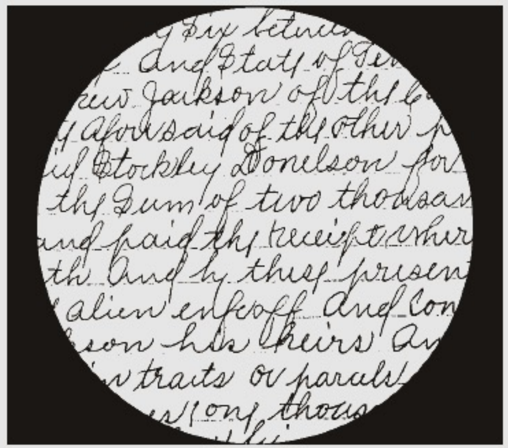


Global Otsu's Method **Image partitioning** **Local Otsu's method**

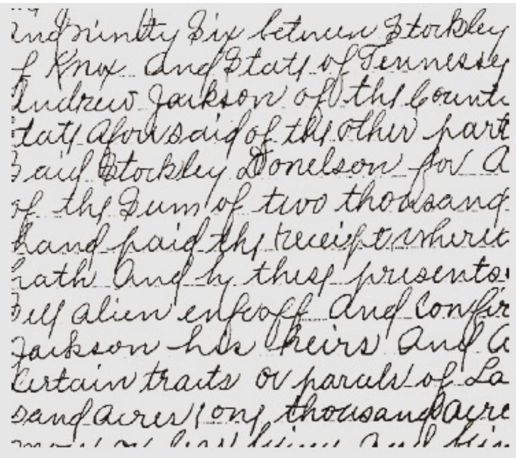
Thresholding (non-uniform background)



Input image

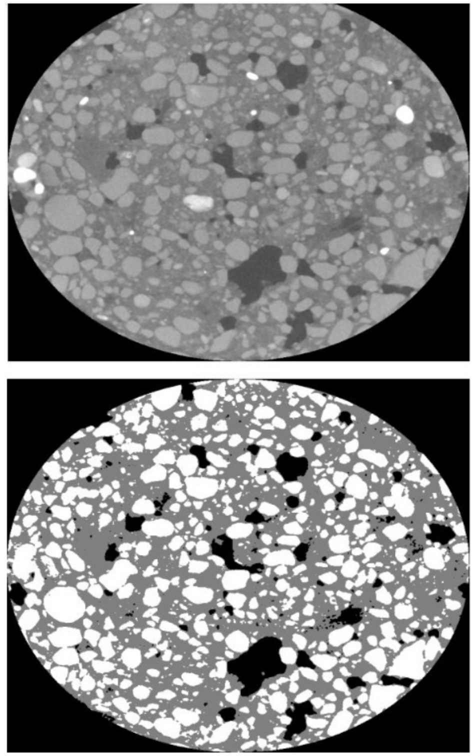


Global thresholding using Otsu's method



Local thresholding with moving average

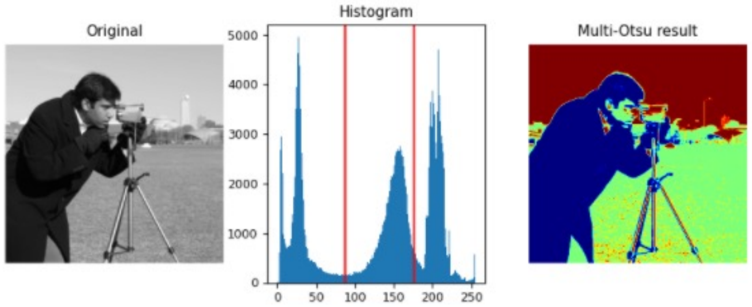
Multi Thresholding by Otsu



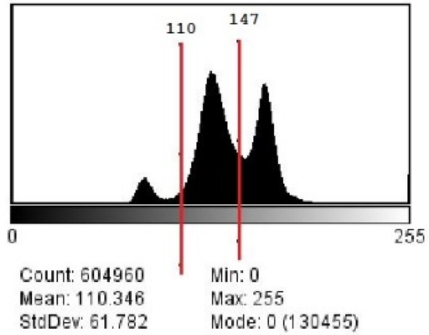
To extend Otsu's thresholding method to multi-level thresholding the between class variance equation becomes:

$$\sigma_B^2 = \sum_{k=0}^{K-1} \omega_k (\mu_k - \mu_T)^2$$

> Please check out Deng-Yuan Huang, Ta-Wei Lin, Wu-Chih Hu, Automatic Multilevel Thresholding Based on Two-Stage Otsu's Method with Cluster Determination by Valley Estimation, Int. Journal of Innovative Computing, 2011, 7:5631-5644 for more information.
>
> http://www.ijicic.org/ijicic-10-05033.pdf



$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_1 \le f(x,y) \le T_2 \\ c & \text{if } f(x,y) \le T_1 \end{cases}$$



Multi Thresholding by Otsu

Ref : https://ftp.iis.sinica.edu.tw/JISE/2001/200109_01.pdf

(a) bi-level (b) tri-level

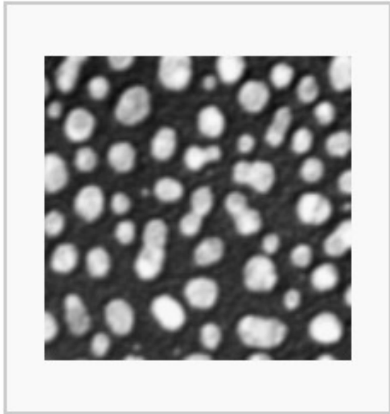
(c) four-level (d) five-level

(a) bi-level (b) tri-level

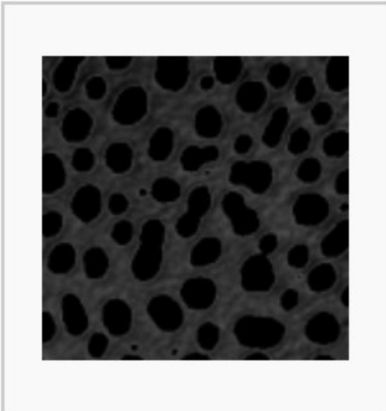
(c) four-level (d) five-level

Multi Thresholding by Otsu

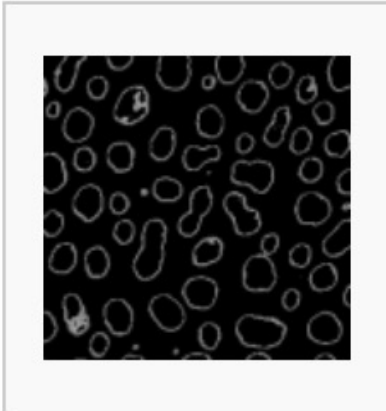
For example, by setting the desired number of classes to 3 (the algorithm then needs to find 2 thresholds), one can get background pixels, bright pixels and intermediate pixels. This might be of interest for images where there is such a pixel populations. In the example depicted below, based on the blob image, one could get the background, the blobs center and the blob edges out of it.



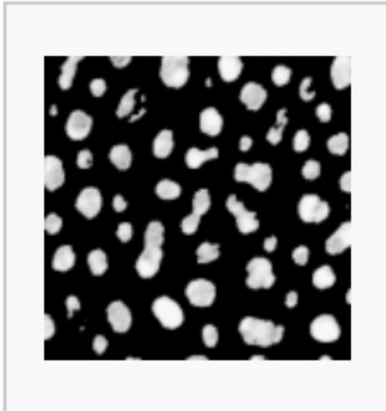
Original image: blobs on a dark background



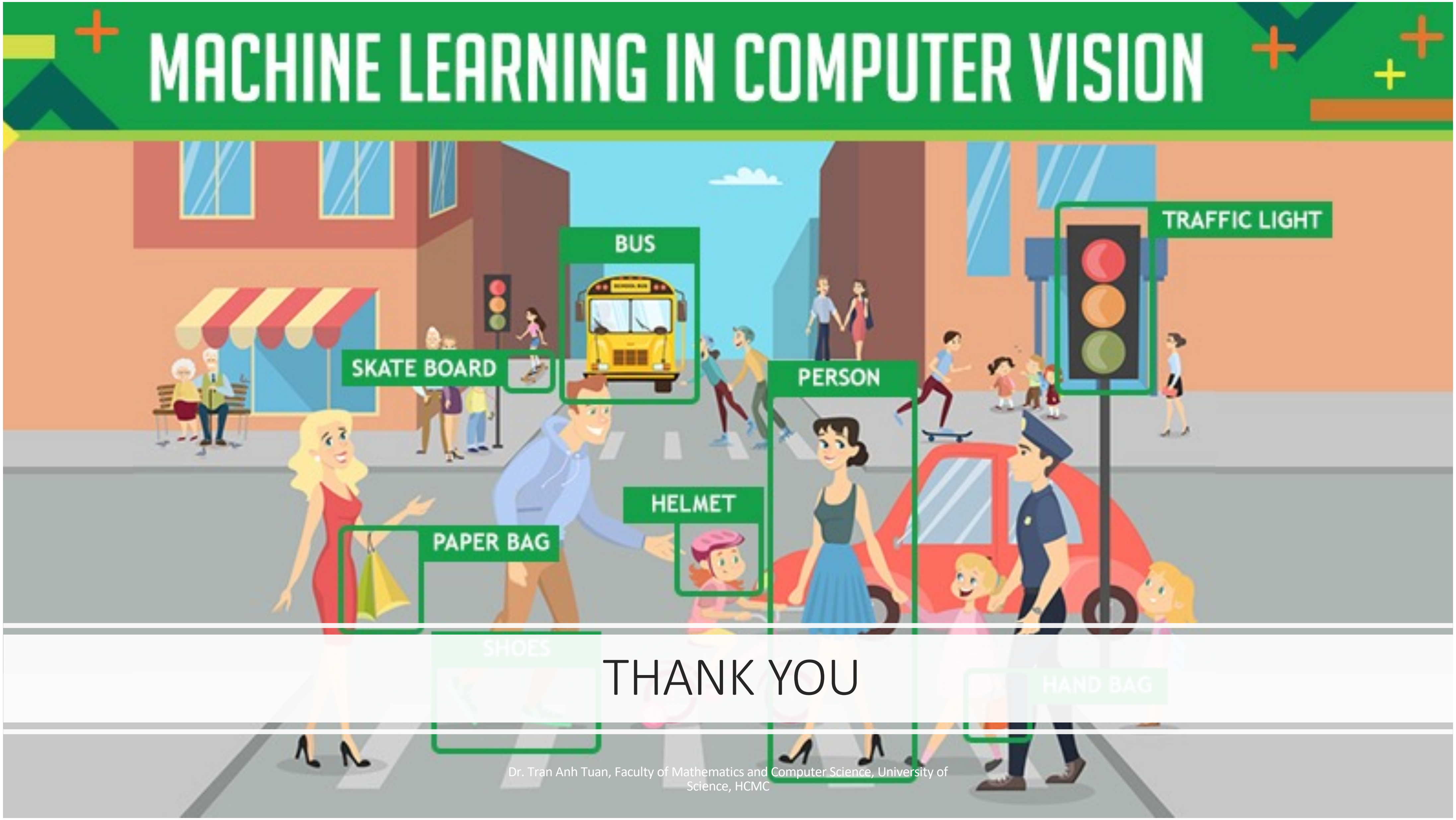
First class: the dark pixels as background



Second class: the intermediate pixels delineate the edges



Third class: the bright pixels form the blobs centers


MACHINE LEARNING IN COMPUTER VISION
BUS
TRAFFIC LIGHT
SKATE BOARD
PERSON
HELMET
PAPER BAG
SHOES
HAND BAG

THANK YOU