Digital Image Processing

COSC 6380/4393

Lecture – 7

Feb 7th, 2023

Pranav Mantini

Slides from Dr. Shishir K Shah and Frank (Qingzhong) Liu

Review: **DIGITAL IMAGE REPRESENTATION**

- Once an image is **digitized** (A/D) and stored it is an array of **voltage or magnetic potentials**
- Not easy to work with from an algorithmic point of view
- The **representation** that is easiest to work with from an **algorithmic perspective** is that of a **matrix of integers**

Matrix Image Representation

- Denote a (square) image matrix **I** = [I(i, j); 0 < i, j < N-1] where
- (i, j) = (row, column)
- I(i, j) = image value at coordinate or pixel (i, j)

Review: **DIGITAL IMAGE REPRESENTATION (contd.)**

- **Example** - Matrix notation

$$\mathbf{I} = \begin{bmatrix} I(0,0) & I(0,1) & \cdots & I(0,N-1) \\ I(1,0) & I(1,1) & \cdots & I(1,N-1) \\ \vdots & \vdots & & \vdots \\ I(N-1,0) & I(N-1,1) & \cdots & I(N-1,N-1) \end{bmatrix}$$

- **Example** - Pixel notation - an N x N image

What's the minimum number of bits/pixel allocated?

columns

rows	0	1	2	3	4	5	6	7	8		N-1
0	193	191	189	194	196	200	225	227	224	• • •	57
1	189	185	187	190	193	198	223	229	222	• • •	62
2	186	188	185	192	194	193	219	228	223	• • •	59
3	180	176	179	178	193	193	199	231	221	• • •	54
	•	•	•	•	•	•	•	•	•		•
N-1	0	0	1	11	13	11	12	10	15	• • •	189

Image **I**

Review: **DIGITAL IMAGE REPRESENTATION (contd.)**

- **Example** - Binary Image (2-valued, usually BLACK and WHITE)

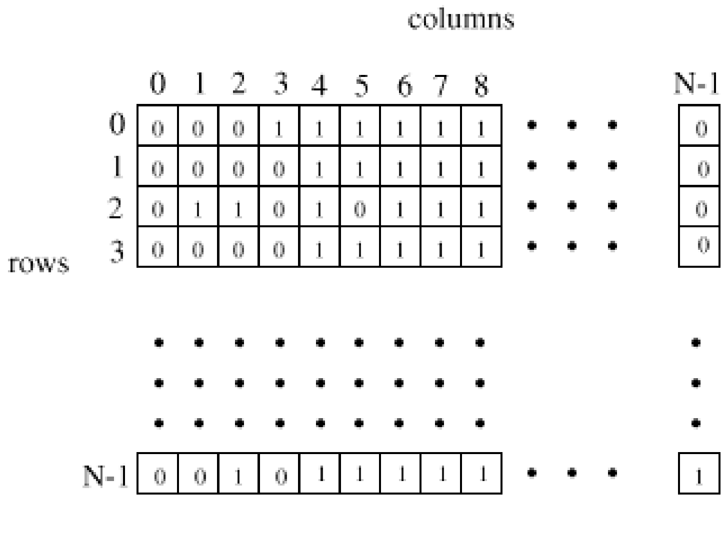


Image **I**

- Another way of depicting the image:

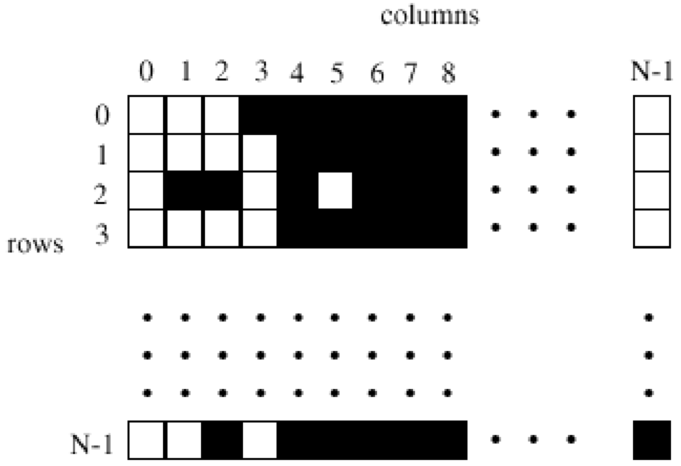


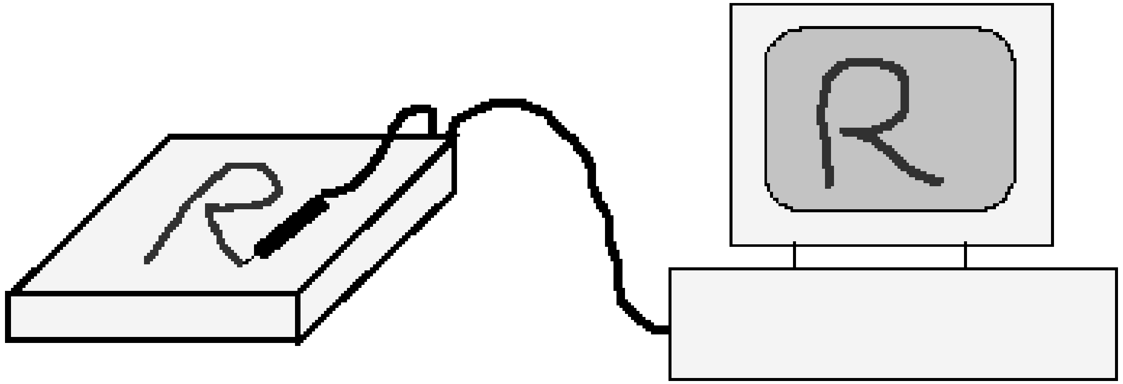
Image **I**

BINARY IMAGES

- Since **binary = bi-valued,** the (logical) values ’0’ or ’1’ usually indicate the **absence** or **presence** of an **image** property in an associated gray-level image:
  - Points of high or low intensity (brightness)
  - Points where an object is present or absent
  - More abstract properties, such as smooth vs. nonsmooth, etc.
- **Convention** - We will make the associations
- ’1’ = BLACK
- ’0’ = WHITE

BINARY IMAGE GENERATION

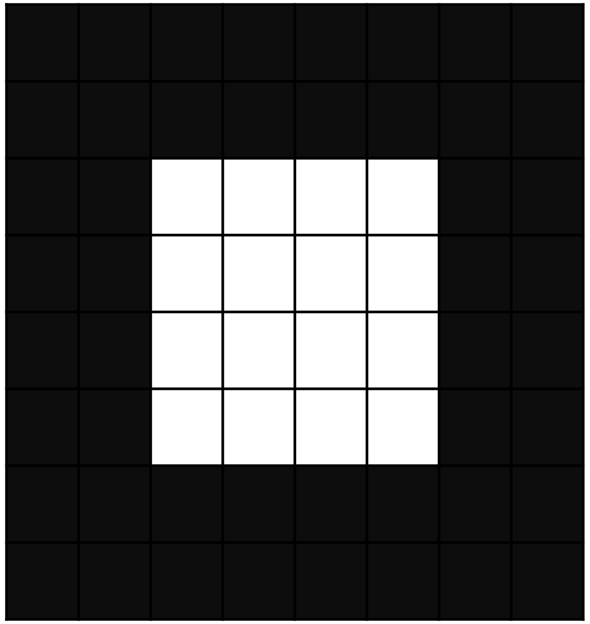
- **Tablet-Based Input:**
- Binary images can derive from **simple sensors** with binary output
- Simplest example: **tablet, resistive pad,** or **light pen**
- All pixels **initially** assigned value '0':
  I = [I(i, j)], I(i, j) = '0' for all (i, j) = (row column)
- When pressure or light is applied at (i_0 , j_0), the image is assigned the value '1': I(i_0 , j_0) = '1'
- This continues until the user completes the drawing



BINARY IMAGE

- Usually a binary image is obtained from a gray-level image
- Advantages:
  - B-fold reduction in required storage
  - Simple abstraction of information
  - Fast processing - **logical operators**
  - Can be further compressed

Grey Level → Binary Image



8X8 image → Black box on white background

Grey Level → Binary Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Grey scale Pixels values

→

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Binary image

GRAY-LEVEL THRESHOLDING

Simple Thresholding

- The simplest of image processing operations
- An extreme form of **gray-level quantization**
- Define an integer **threshold** T (in the gray-scale range)
- Compare each pixel intensity to T

THRESHOLDING

- Suppose **gray-level** image **I** has K gray-levels: 0, 1, 2, ..., K-1
- Select threshold T $\in$ { 0, 1, 2, ..., K-1}
- Compare **every** gray-level in **I** to T
- Define a new **binary image J** as follows:
- J(i, j) = '0' if I(i, j) <= T
- J(i, j) = '1' if I(i, j) > T
- A new binary image **J** is created from a gray-level image **I**

Grey Level → Binary Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Grey scale Pixels values

$Threshold$(T) →

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Binary image

Grey Level → Binary Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Grey scale Pixels values

$Threshold$(T) →

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Binary image

What is good value of T?

Grey Level → Binary Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8X8 image → grey box on black background

What is good value of T?

Grey Level → Binary Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8X8 image → grey box on black background

What is good value of T?

240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240

8X8 image → white box on dark white background

What is good value of T?

THRESHOLD SELECTION

- The quality of the **binary image J** obtained by thresholding **I** depends very heavily on the **threshold T**
- Indeed it is instructive to observe the result of thresholding an image at many different levels in sequence
- Different thresholds can produce different valuable abstractions of the image
- Some images do not produce any interesting results when thresholded by any T
- So: How does one decide if thresholding is possible ?
- How does one decide on a threshold T ?

Grey Level → Binary Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8X8 image → black box on grey background

240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240

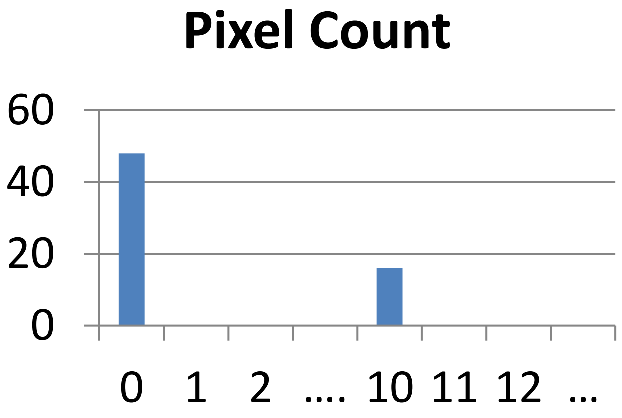
8X8 image → light white box on white background

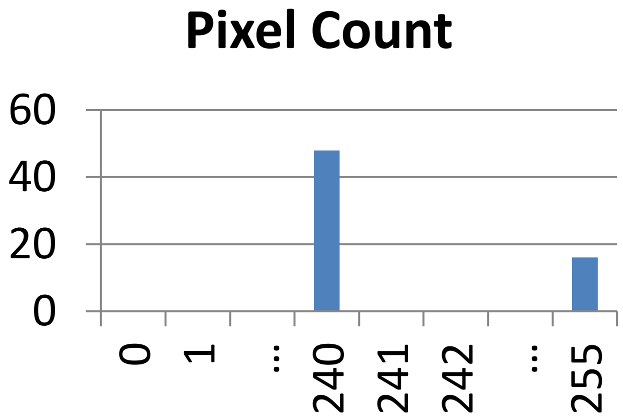
How do we determine T?

Determine modes

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

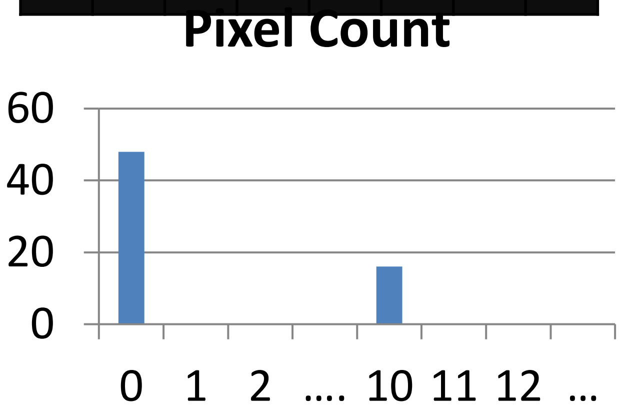
240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240





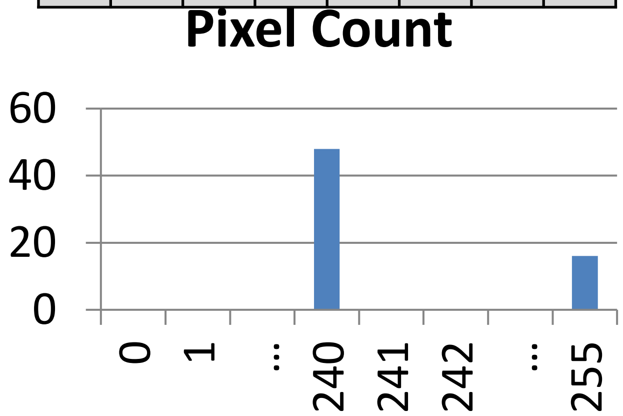
Determine modes

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	10	10	10	10	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



$$mode_1 = 0; mode_2 = 10$$
$$T = avg(mode) = \mathbf{5}$$

240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	255	255	255	255	240	240
240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240



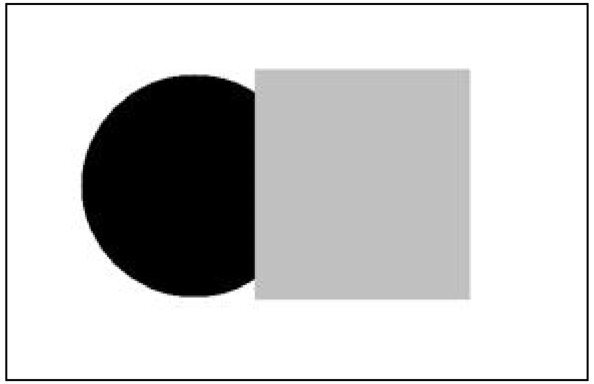
$$mode_1 = 240; mode_2 = 255$$
$$T = avg(mode) = \mathbf{247.5}$$

GRAY-LEVEL IMAGE HISTOGRAM

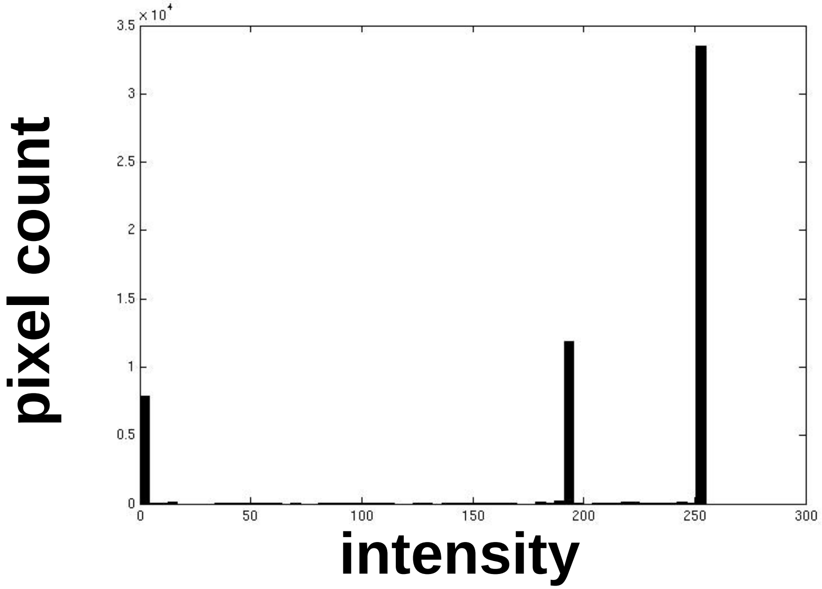
- The **histogram H_I** of image **I** is a **plot** or **graph** of the **frequency of occurrence** of each gray level in **I**
- H_I is a one-dimensional function with domain 0, ... , K-1
- $H_I(x) = n$ if **I** contains **exactly** n occurrences of gray level x, for each x = 0, ... K-1

Histogram Example

Black = 0
Gray = 190
White = 254

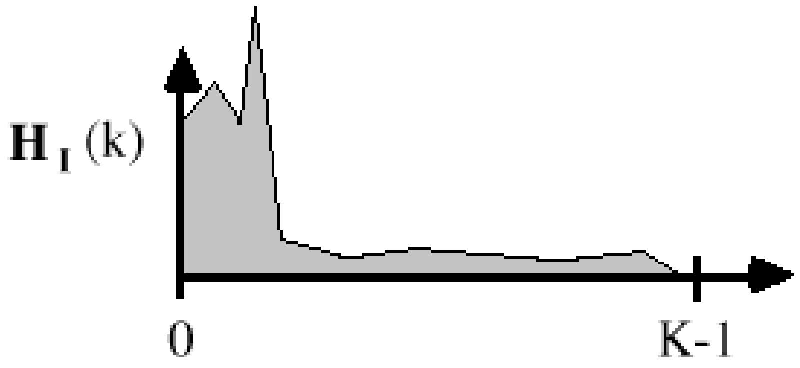


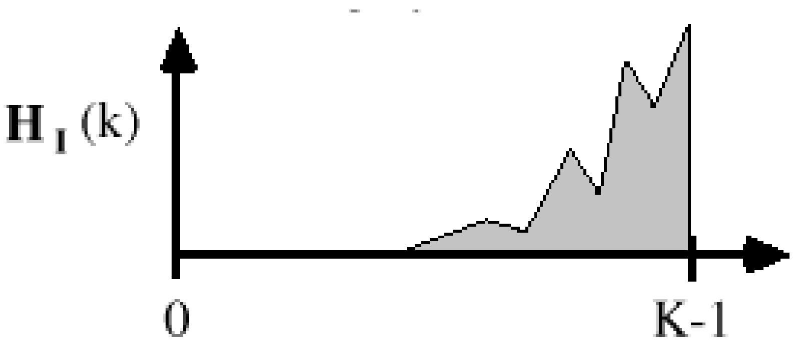
input image



HISTOGRAM APPEARANCE

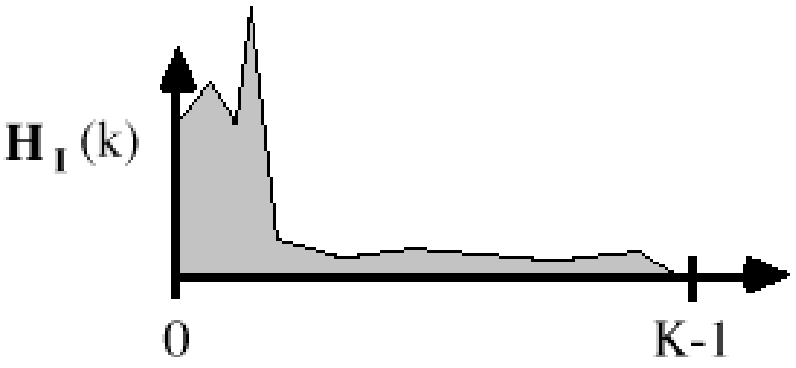
- The **appearance** of a histogram suggests much about the image



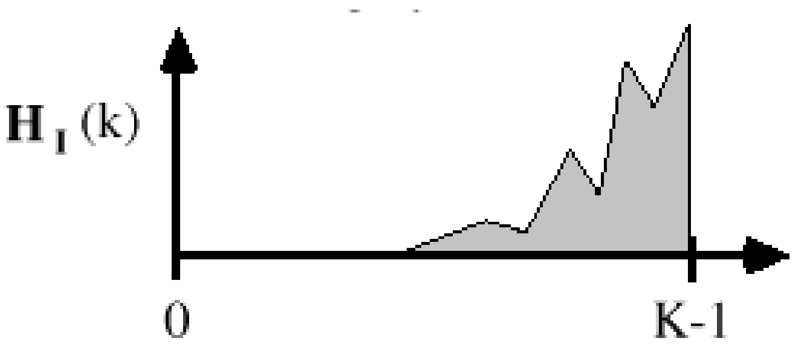


HISTOGRAM APPEARANCE

- The **appearance** of a histogram suggests much about the image



predominantly
dark image

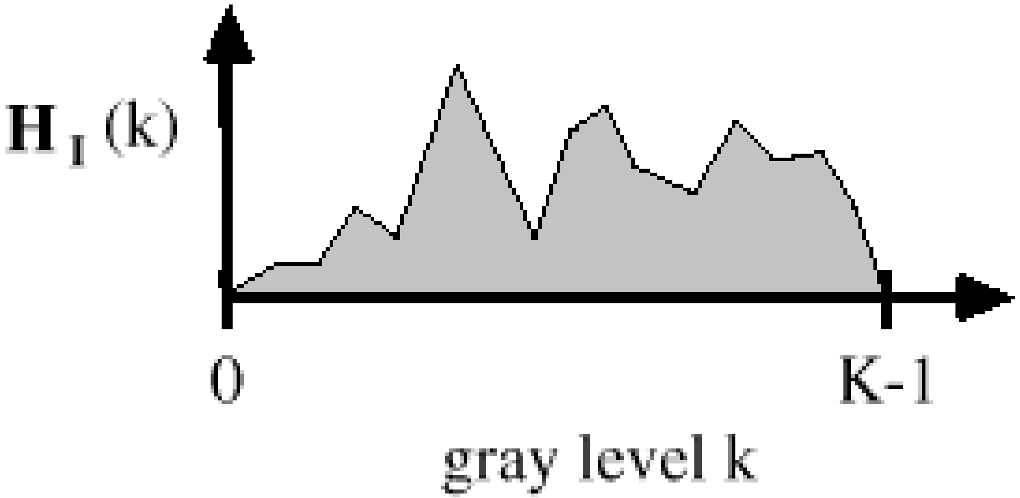


predominantly
light image

- These could be histograms of **underexposed** and **overexposed** images, respectively

HISTOGRAM APPEARANCE

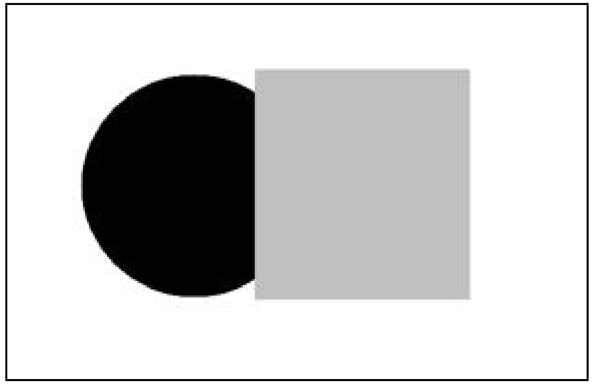
- This histogram may show better use of the gray-scale range



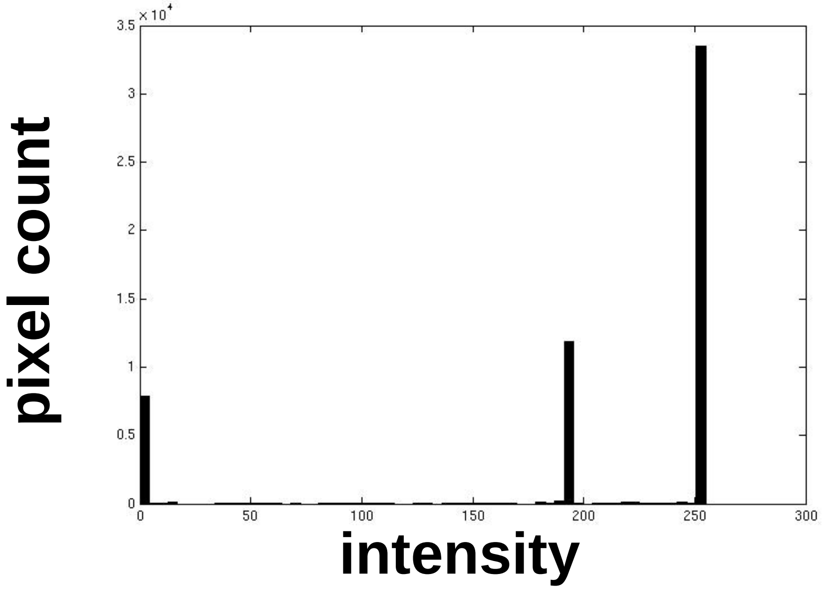
- Well-distributed histogram

Histogram Example

Black = 0
Gray = 190
White = 254

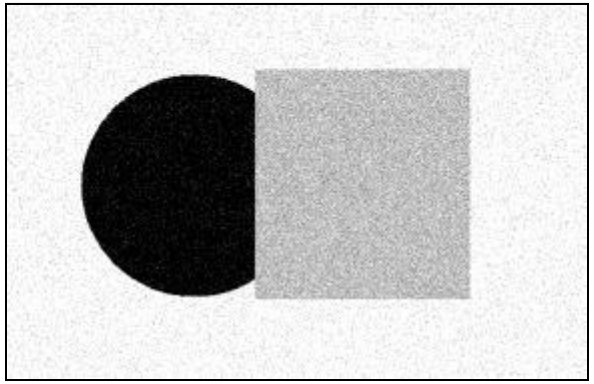


input image

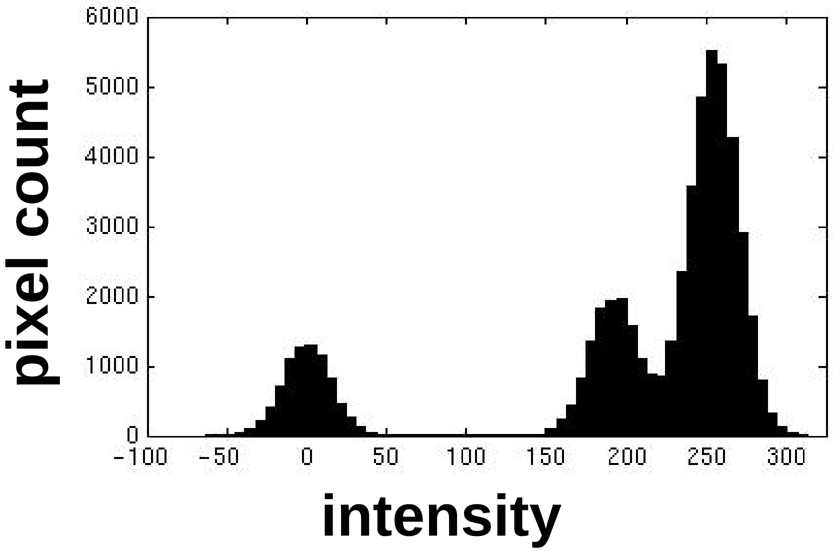


Histogram Example

- **Reality**



input image

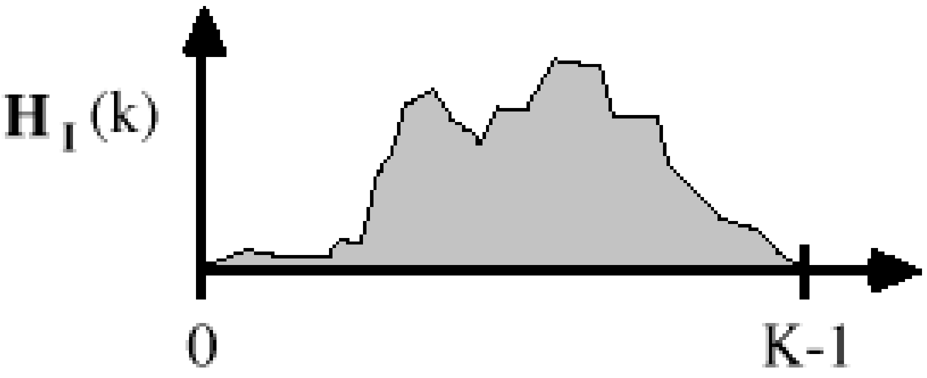


BIMODAL HISTOGRAM

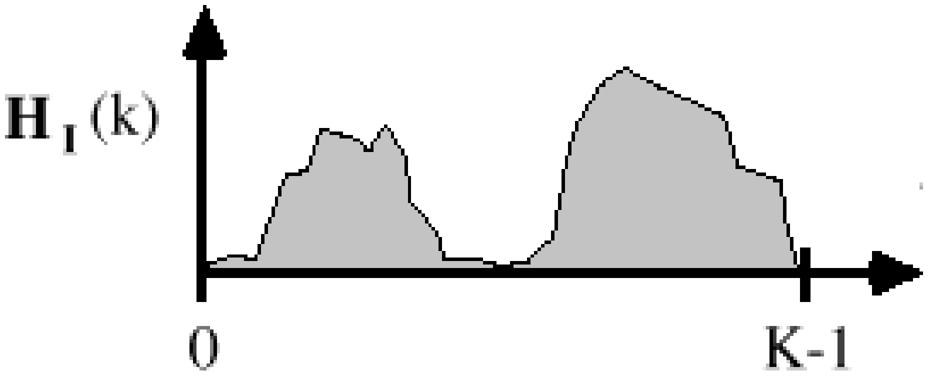
- Thresholding usually works best when there are **dark objects** on a **light background**
- Or when there are **light objects** on a **dark background**
- Images of this type tend to have histograms with **multiple distinct peaks or modes in them**

BIMODAL HISTOGRAM

- If the peaks are well-separated, threshold selection can be easy



bimodal histogram
poorly separated

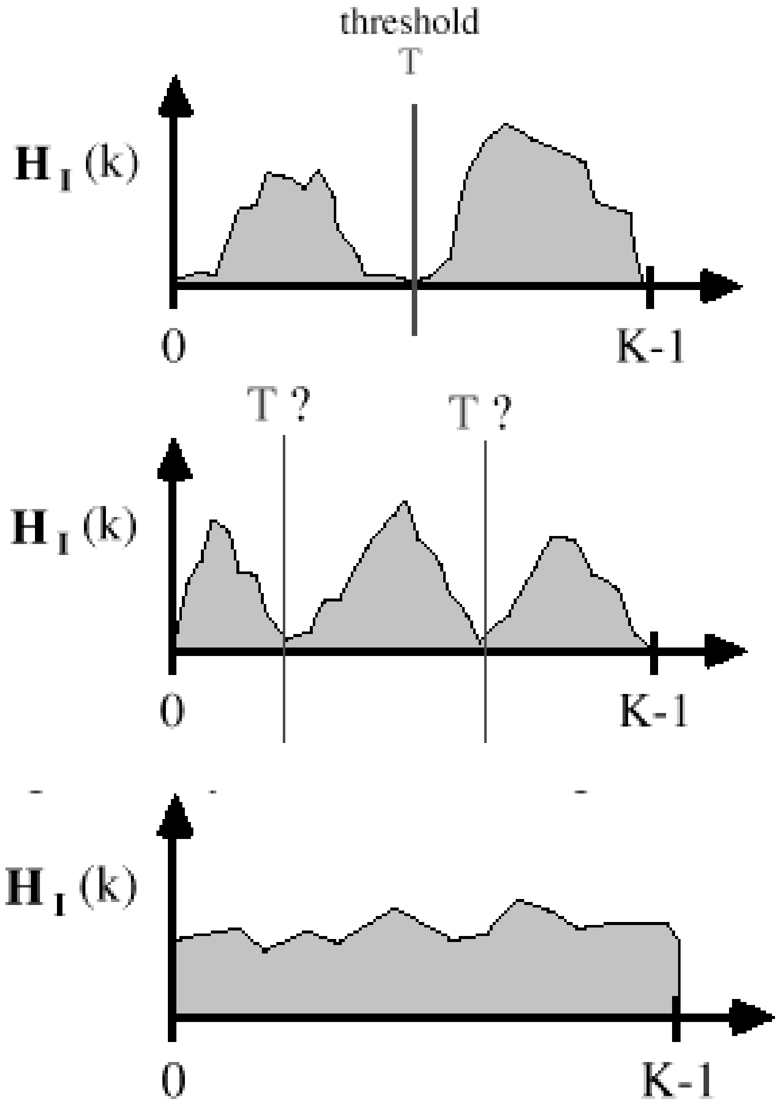


bimodal histogram
well separated peaks

- Set the threshold T somewhere between the peaks
- It may be an interactive trial-and-error process

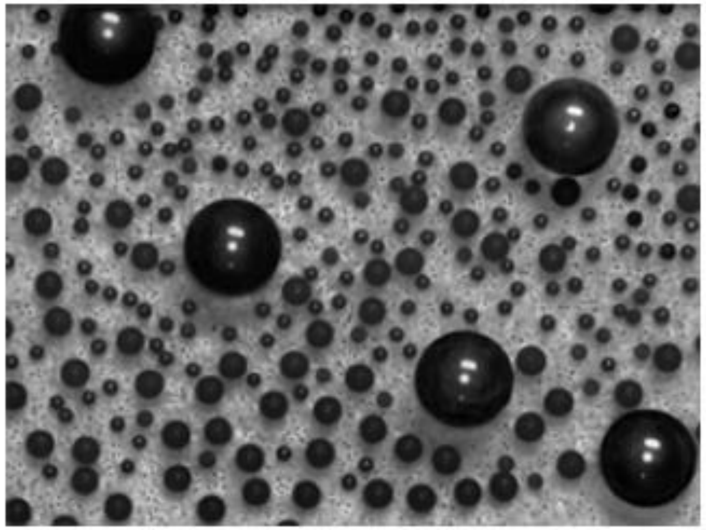
THRESHOLD SELECTION FROM HISTOGRAM

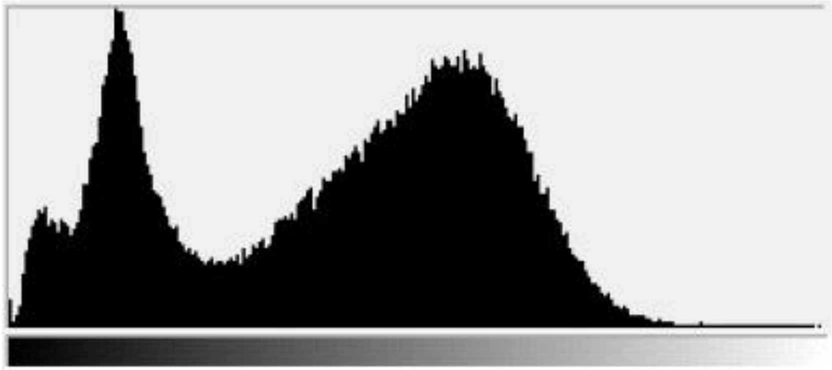
- Placing threshold T between modes may yield acceptable results
- Exactly **where in between** can be difficult to determine
- An image histogram may contain multiple modes. Placing the threshold in different places will produce very different results
- Histogram may be "flat," making threshold selection **difficult**



Example: How to find T

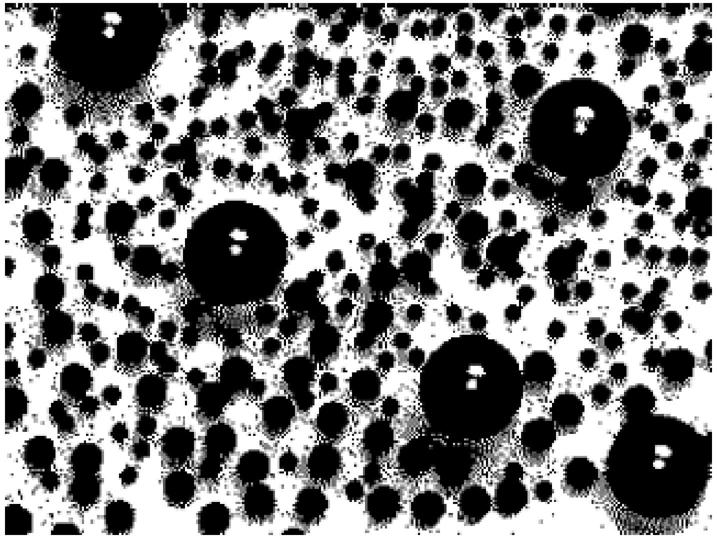
- Microscopic image
- Grey level → binary image
- Binary: 1-cell present, 0-cell absent

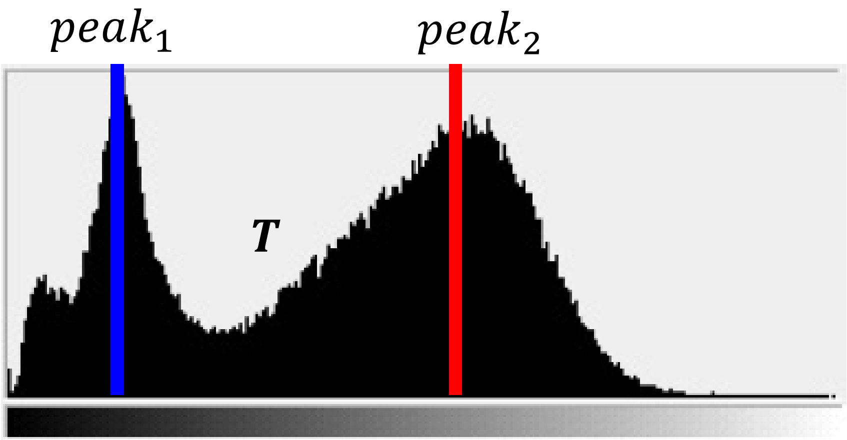




Example: How to find T

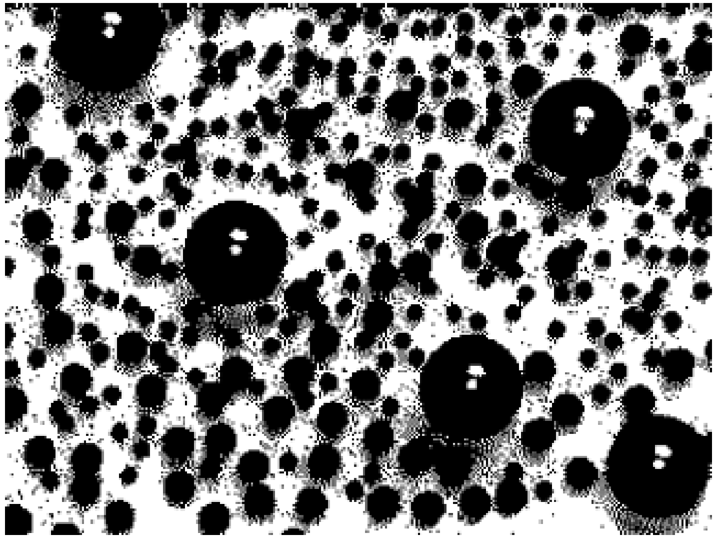
- If the peaks are known
- We can choose T between peaks (say average)

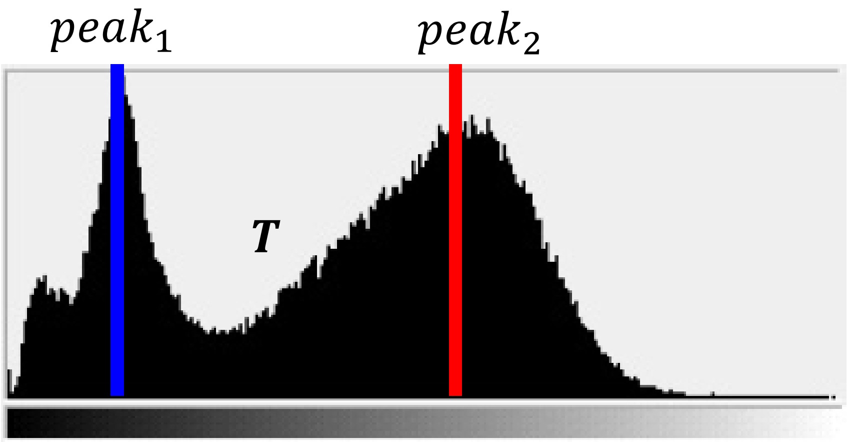




Example: How to find T

- If the peaks are known
- We can choose T between peaks (say average)

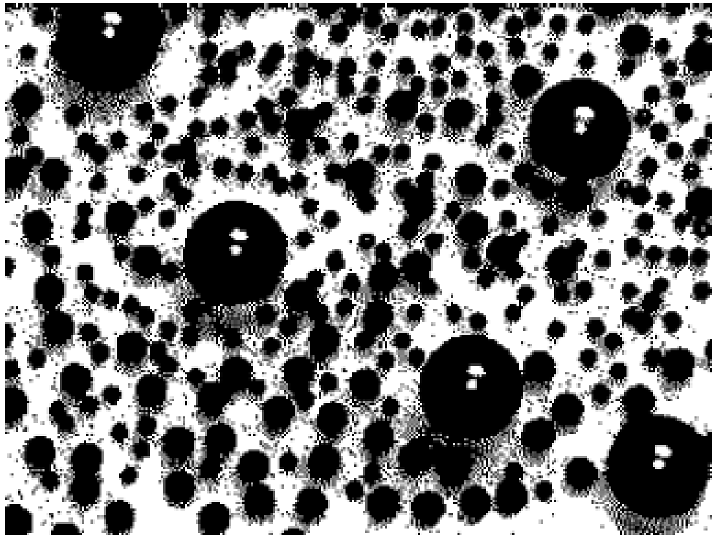


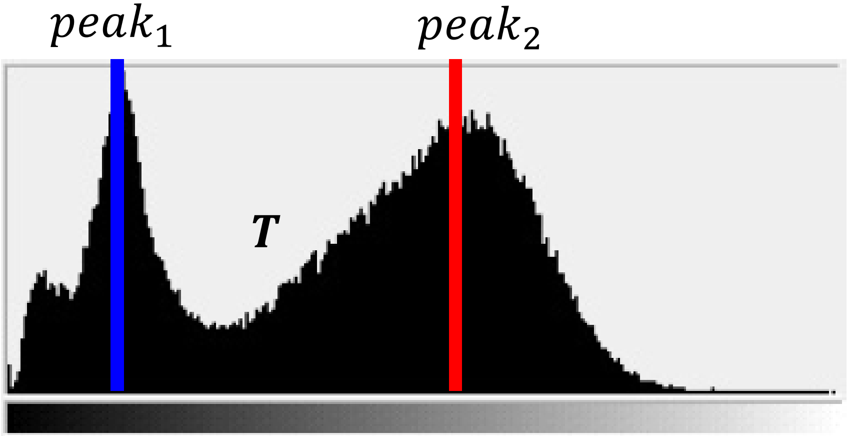


We do not know the peaks!!

Example: How to find T

- If the Threshold (T) is known
- Can we determine the peaks?





Recap: Probability

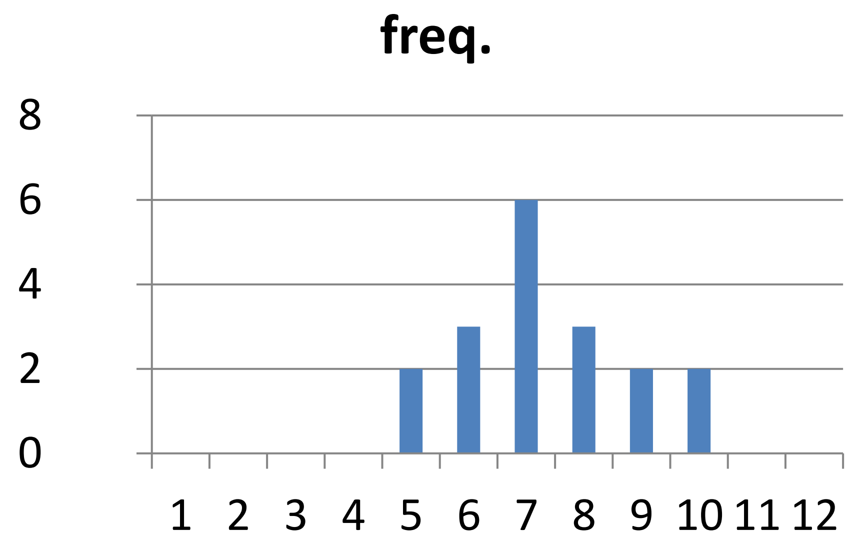
- Data: {5,5,6,6,6,7,7,7,7,7,7,8,8,8,9,9,10,10}
- X: random variable
- P: X → [0,1] probability function
- $P(X = 7) = ?$

Recap: Probability

- Data: {5,5,6,6,6,7,7,7,7,7,7,8,8,8,9,9,10,10}
- X: random variable
- P: probability function
- $P(X = 7) = 0.33$

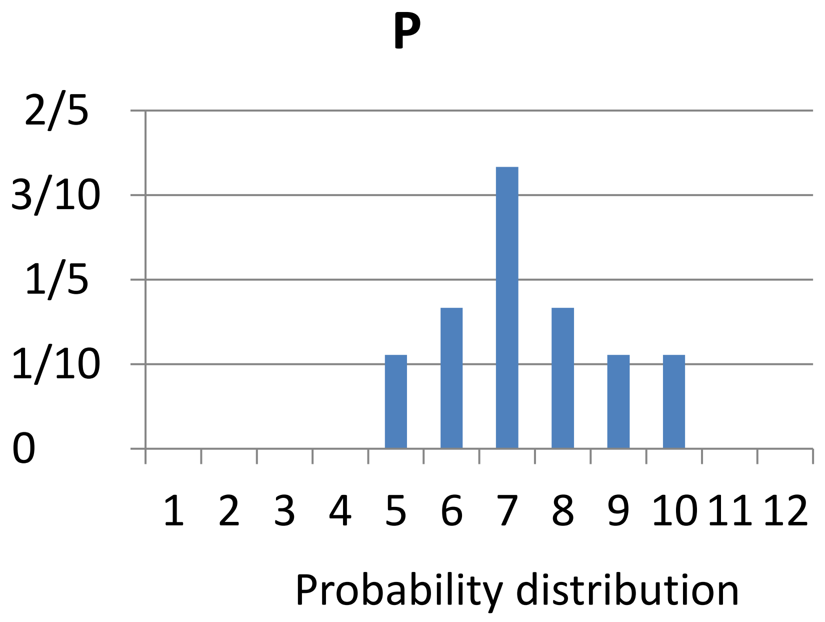
Recap: Histogram

- Data: {5,5,6,6,6,7,7,7,7,7,7,8,8,8,9,9,10,10}
- X: random variable
- p: probability function
- $p(X = 7) = 0.33$
- Histogram



Recap: Probability Distribution

- Data: {5,5,6,6,6,7,7,7,7,7,7,8,8,8,9,9,10,10}
- X: random variable
- p: probability function
- $\mathrm{p}(X = 7) = 0.33$
- p→Normalize(Histogram)



- Where is the peak for this case?

Expectation

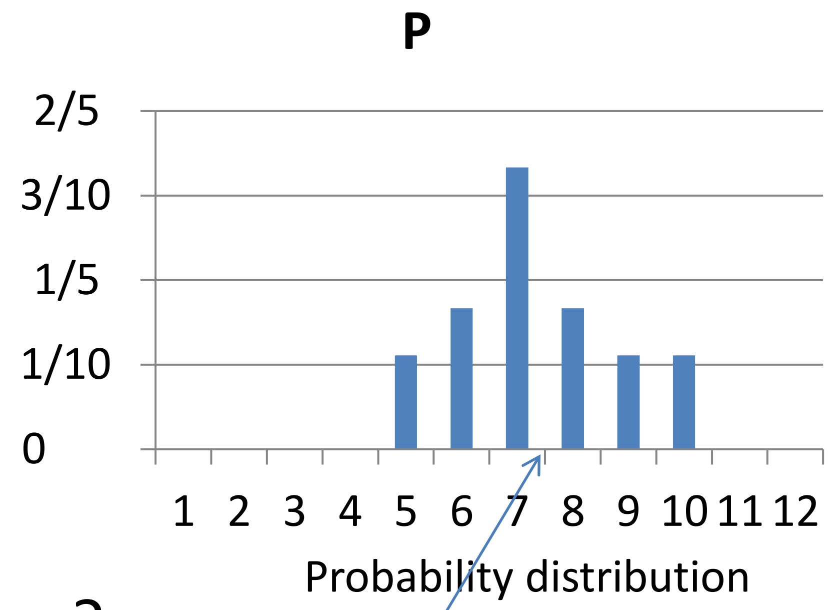
- $E(X)$→Expected value of random variable X
- ~Average of all the expected values of random variable X
- $E(X) =?$

Expectation

- $E(X)$→Expected value of random variable X
- ~Average of all the values
- $E(X) = \sum X\, p(X)$

Recap: Probability Distribution

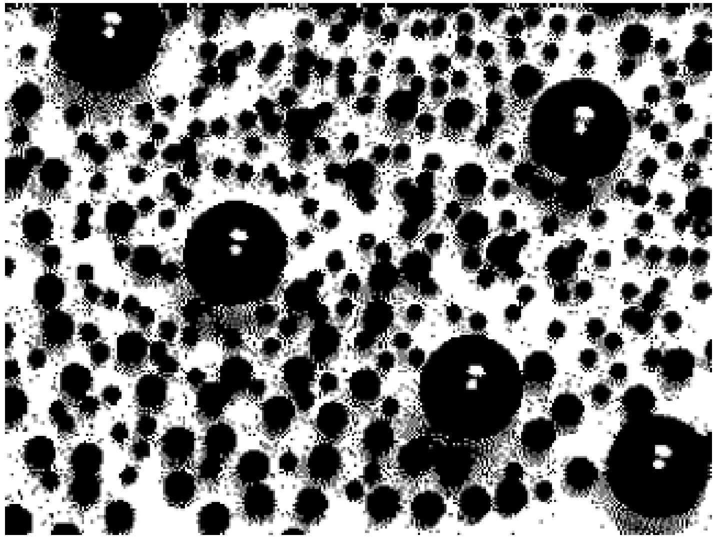
- Data: {5,5,6,6,6,7,7,7,7,7,7,8,8,8,9,9,10,10}
- X: random variable
- p: probability function
- $p(X = 7) = 0.33$
- p→Normalize(Histogram)

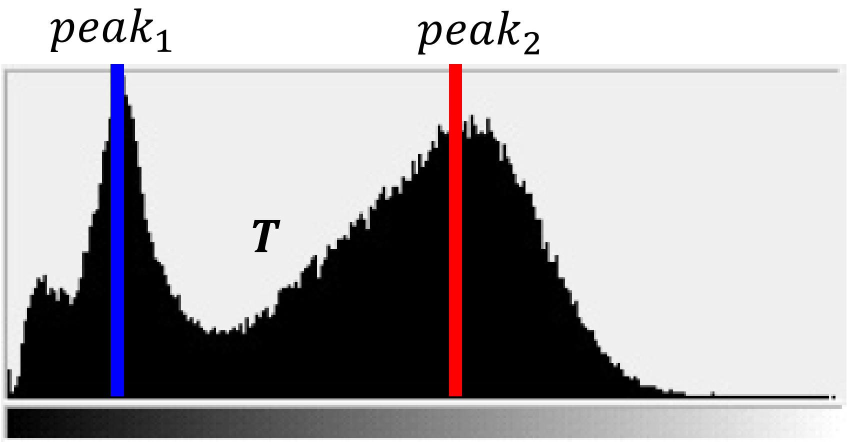


- Where is the peak for this case?
- E(X) = 5 * 0.11 + 6*0.17 + 7*0.33 +… = **7.33**

Example: How to find T

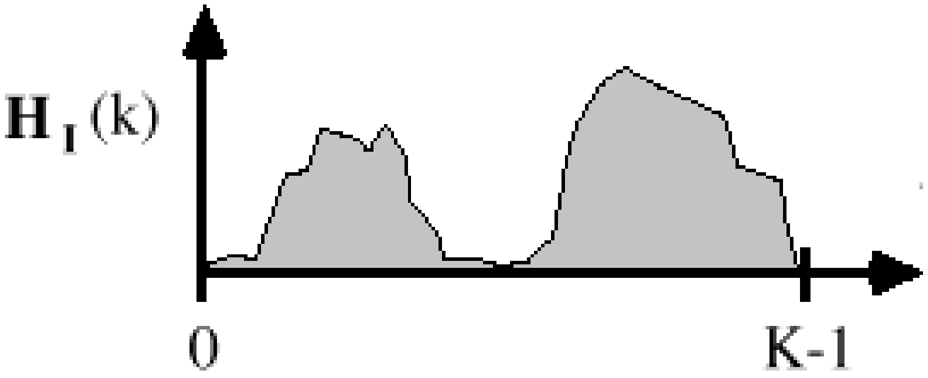
- If the Threshold (T) is known
- Can we determine the peaks?
- Yes, Compute Expectation on either side of the threshold.





Algorithm

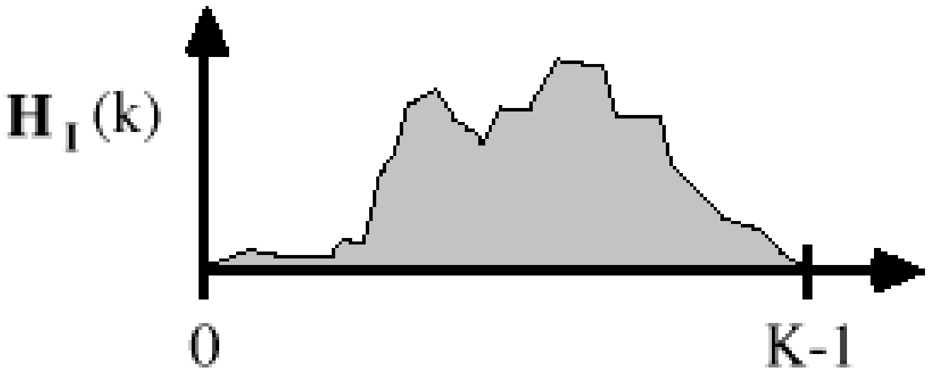
$$Initialize\ T = K/2$$
$$Do$$
$$\quad Compute\ \mu_1 = E(X) \forall X < T$$
$$\quad Compute\ \mu_2 = E(X) \forall X \geq T$$
$$\quad Set\ T = \frac{\mu_1 + \mu_2}{2}$$
$$While\ \Delta\mu_1 != 0\ \&\Delta\mu_2 != 0$$



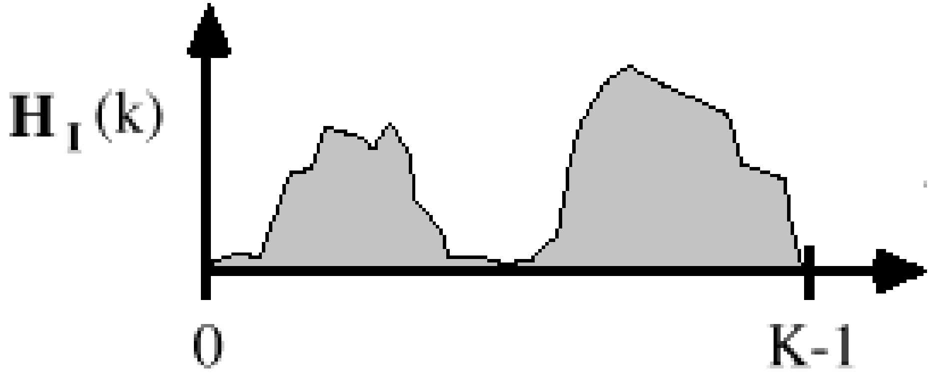
bimodal histogram
well separated peaks

AKA: Expectation Maximization (simple version)

BIMODAL HISTOGRAM



bimodal histogram
poorly separated



bimodal histogram
well separated peaks