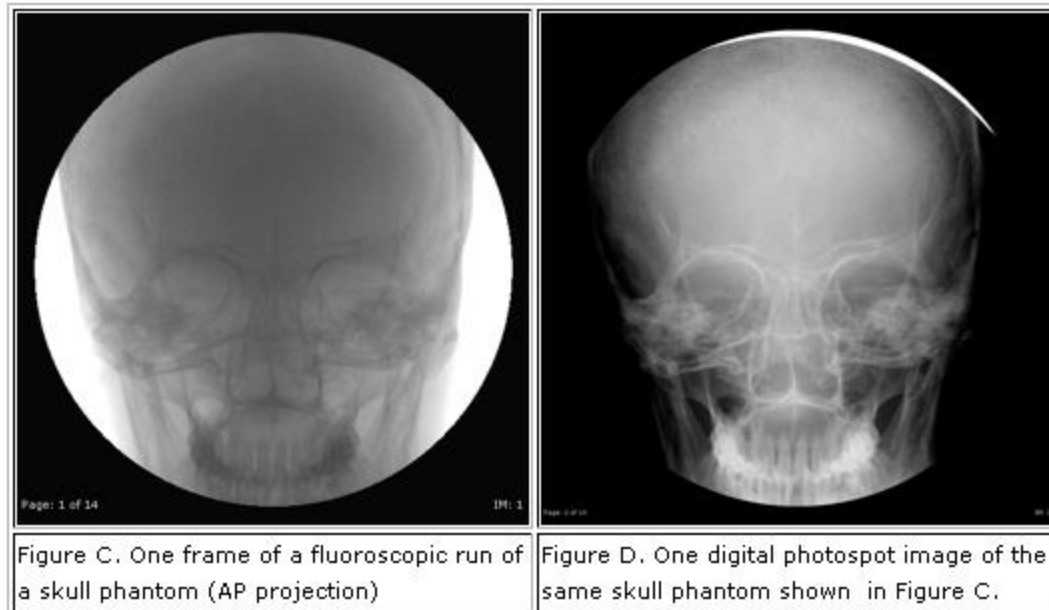


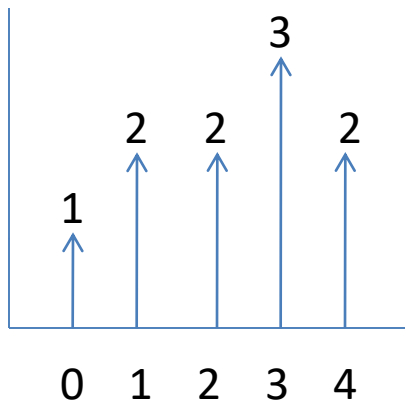
Common Low-level Operations for Processing & Enhancement



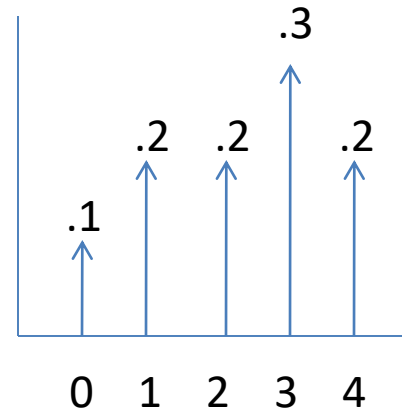
Histogram Equalization

- A **histogram** of image I is a data structure h in which $h(i)$ is the number of pixels in I that have value i . Usually $0 \leq i \leq 255$.
- A **normalized** histogram is a histogram in which each value is divided by the total number of pixels in the image. These normalized values are often thought of as **probabilities** $\text{Pr}(i)$ of the different pixel values.

Example of Normalization



original histogram
5 gray tones, 10 pixels



normalized histogram
probability of each gray tone

What's the probability of a pixel having value 3 in the image?

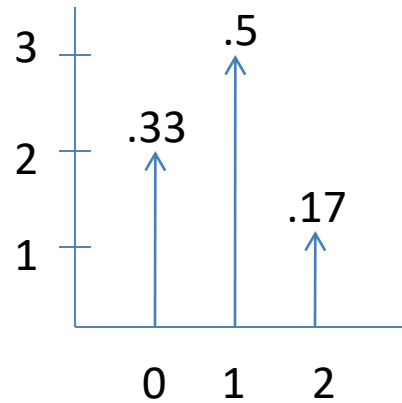
Histogram Equalization Operator

- Goal: transform the gray tones of the image so that they are all approximately equally likely
- Method: use a transformation function $T(r)$ to transform each gray tone r of the L gray tones.

$$T(r) = (L - 1) \sum_{w=0}^r \text{Pr}(w) \quad \text{discrete form we use}$$

$$T(r) = (L - 1) \int_0^r \text{Pr}(w) dw \quad \text{continuous form (CDF)}$$

Small Example



L = 3 graytones

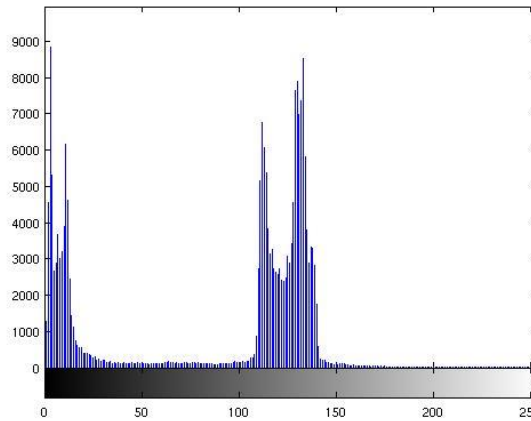
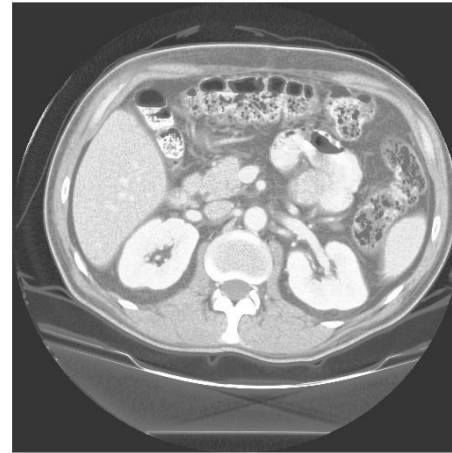
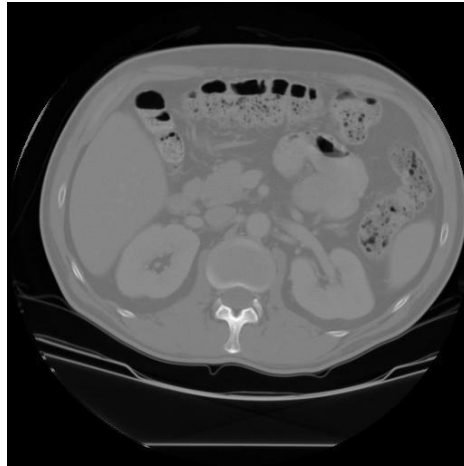
$$T(0) = 2(.33) = .66 \quad = 1$$

$$T(1) = 2(.33 + .5) = 1.66 \quad = 2$$

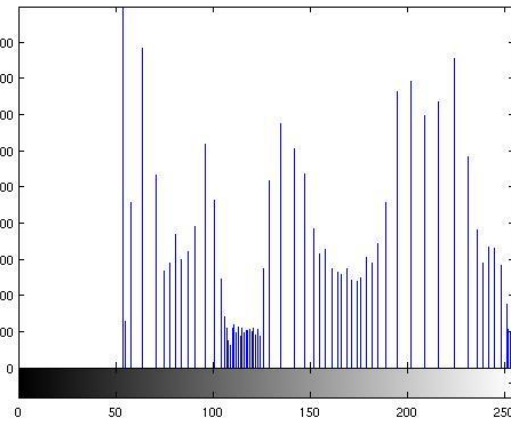
$$T(2) = 2(.33 + .5 + .17) = 2.0 \quad = 2$$

Why is the new histogram not a perfect uniform distribution?

CT Abdomen Image Example



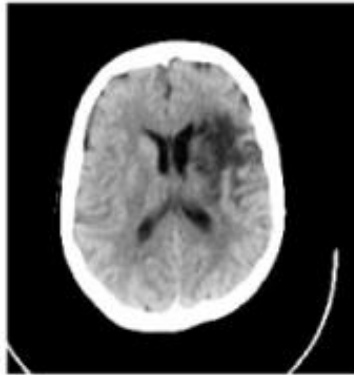
original image
and histogram



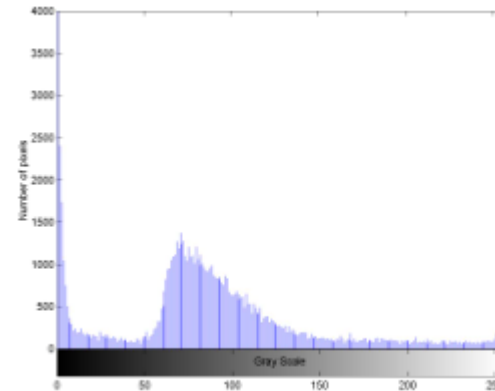
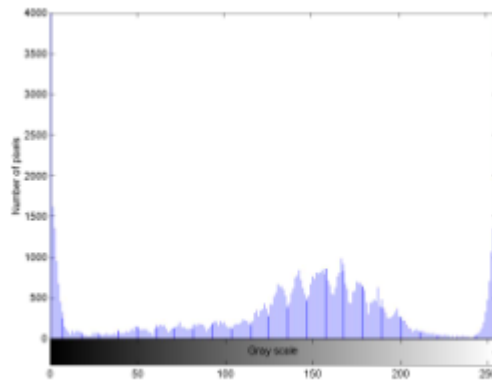
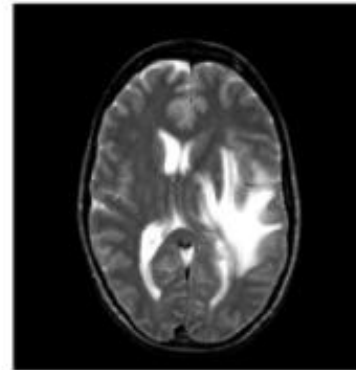
equalized image
and histogram

Images and Histograms*

CT



MRI



*from Medical Image Analysis by Dhawan

Histogram Equalization

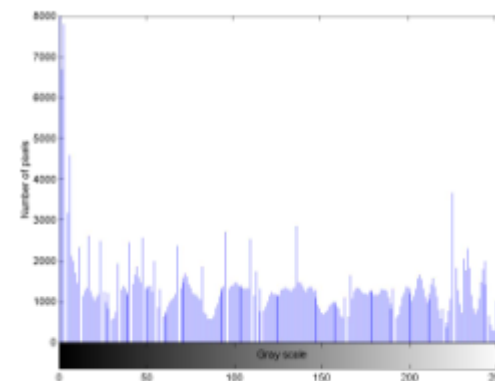
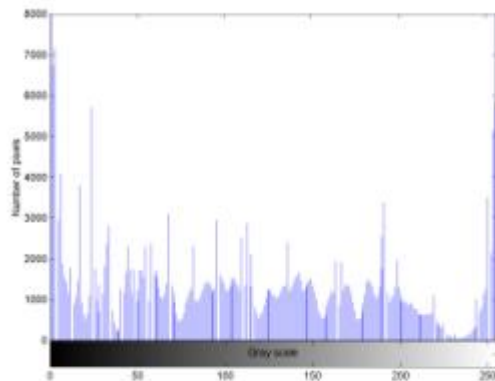
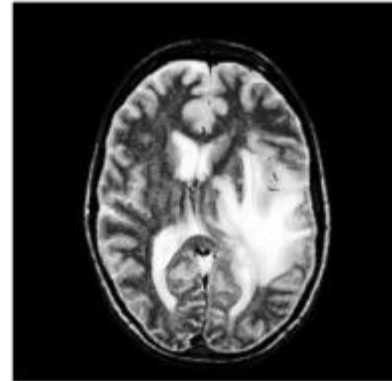
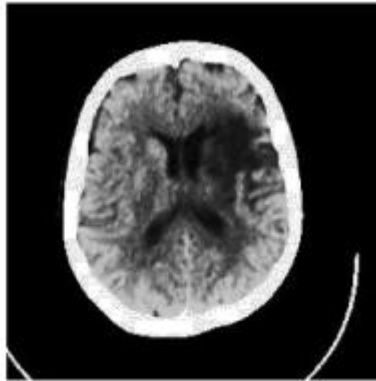


Image Averaging and Subtraction

- If images are noisy, multiple images may be taken and averaged to produce a new image that is smoothed.
- Image subtraction to show differences over time or with and without a dye or tracer.

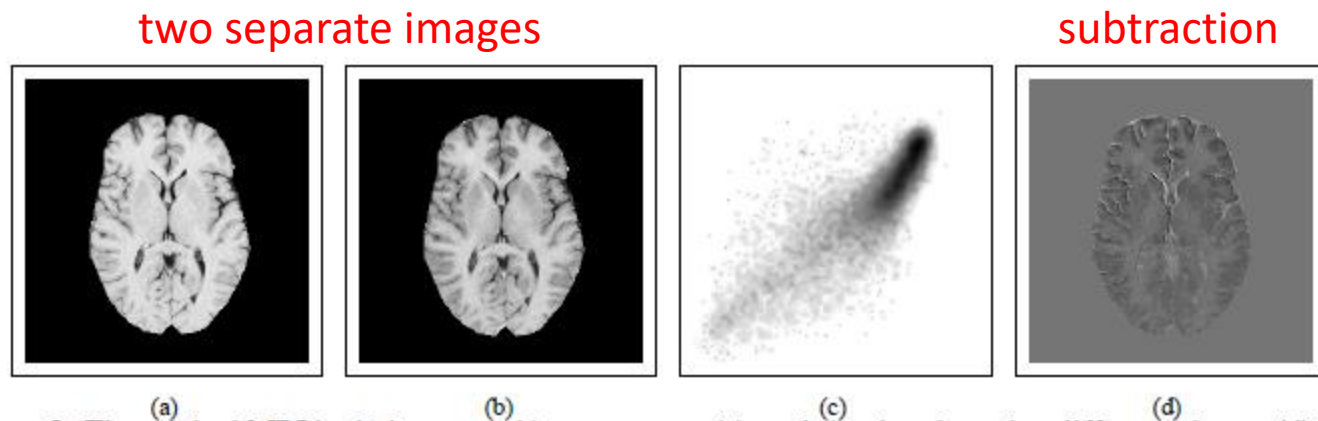


Figure 1. The original MRI brain images (a,b), scattergram (c), and simple subtraction difference image (d), with a 2σ offset added to a small region of image (b).

Image Averaging Masks

For each pixel of the input, the output is the mean of the pixels in its 4-neighborhood or 8-neighborhood.

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$$

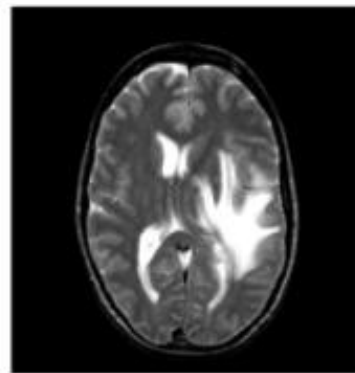
	$f_{(-1,0)}$	
$f_{(0,-1)}$	$f_{(0,0)}$	$f_{(0,1)}$
	$f_{(1,0)}$	

$f_{(-1,-1)}$	$f_{(-1,0)}$	$f_{(-1,1)}$
$f_{(0,-1)}$	$f_{(0,0)}$	$f_{(0,1)}$
$f_{(1,-1)}$	$f_{(1,0)}$	$f_{(1,1)}$

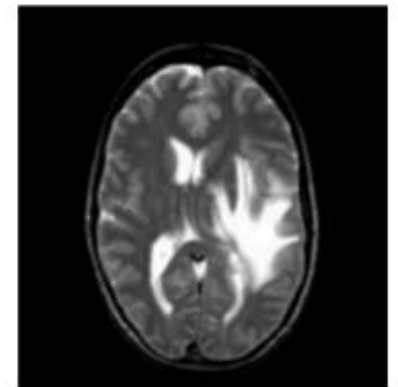
Weighted Image Averaging

$$g(x, y) = \frac{1}{\sum_{x'=-p}^p \sum_{y'=-p}^p w(x', y')} \sum_{x'=-p}^p \sum_{y'=-p}^p w(x', y') f(x + x', y + y')$$

1	2	1
2	4	2
1	2	1



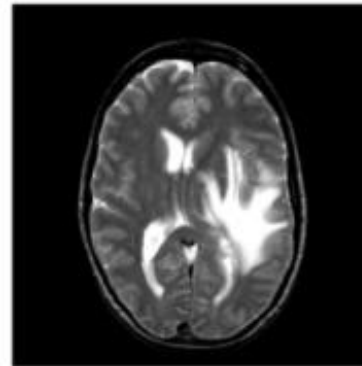
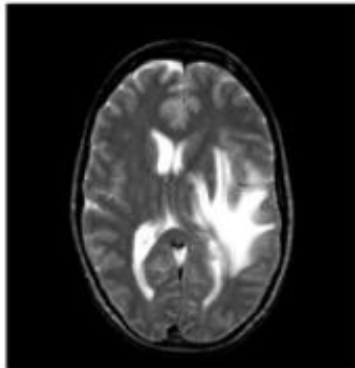
original image f



output image g

Median Filter

$$\hat{f}(x, y) = \underset{(i, j) \in N}{\text{median}} \{g(i, j)\}$$



Laplacian: Second Order Gradient for Edge Detection

$$\nabla^2 f(x, y) = \frac{\delta^2 f(x, y)}{\delta x^2} + \frac{\delta^2 f(x, y)}{\delta y^2}$$
$$= [4f(x, y) - f(x+1, y) - f(x-1, y) - f(x, y+1) - f(x, y-1)]$$

	-1	
-1	4	-1
	-1	

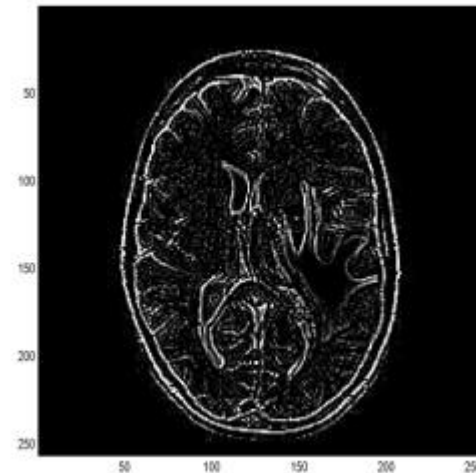
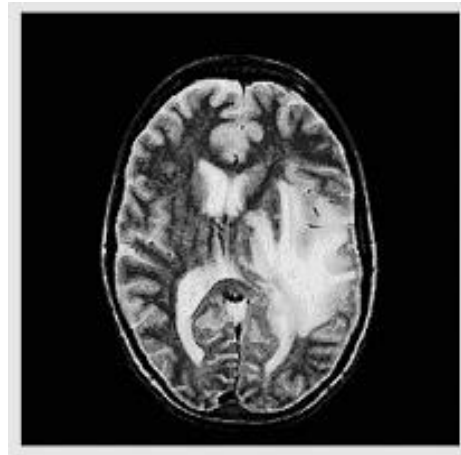
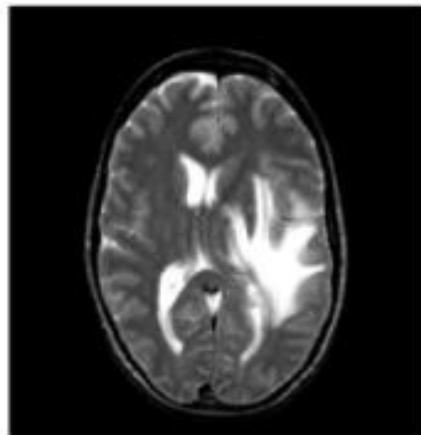


Image Sharpening with Laplacian

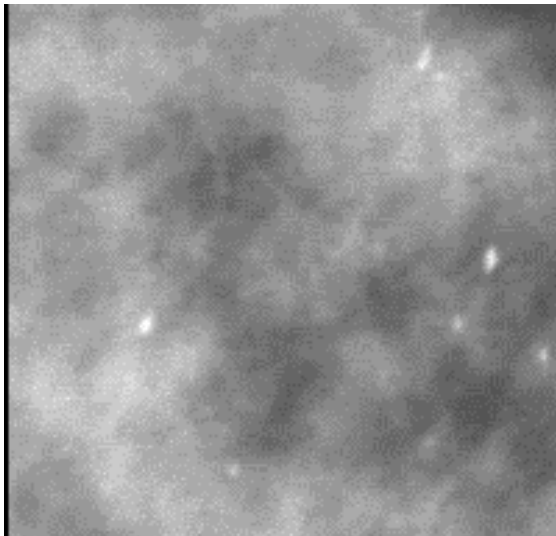
-1	-1	-1
-1	9	-1
-1	-1	-1

Apply a Laplacian and add it to the original image to enhance the edges.

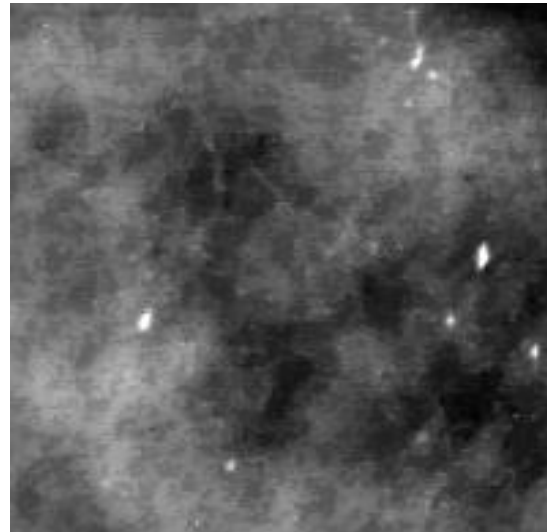


Micro-calcification Enhancement*

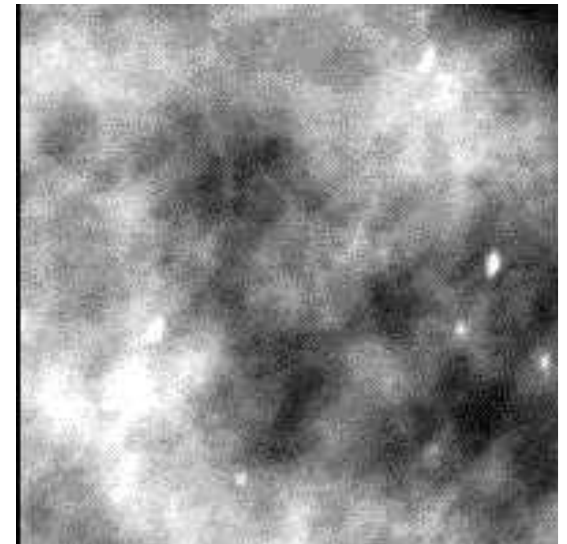
original mammogram



enhancement



histogram equalization



* Using adaptive neighborhood processing, from Dhawan book.

Frequency-Domain Methods

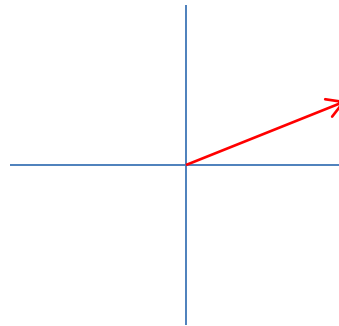
- Convert the image from the spatial domain to the frequency domain via a Fourier Transform
- Apply filters in the frequency domain
- Convert back to the spatial domain
- Used to emphasize or de-emphasize specified frequency components

Fourier Transform Basics

- complex numbers
- complex exponentials
- Euler's formula
- n^{th} roots of unity
- orthogonal basis
- 1D discrete Fourier transform
- 2D discrete Fourier transform
- examples

Complex Numbers (Review)

- Some equations have no real roots: $x^2 = -1$
- Mathematicians handle this by defining an imaginary number i which is the square root of -1
- Complex numbers are of the form $a + bi$, where a and b are real
- Complex numbers may be added and multiplied
- The complex number $a + bi$ can be viewed as a 2D vector $[a,b]$
- Complex numbers are used heavily in physics and EE.



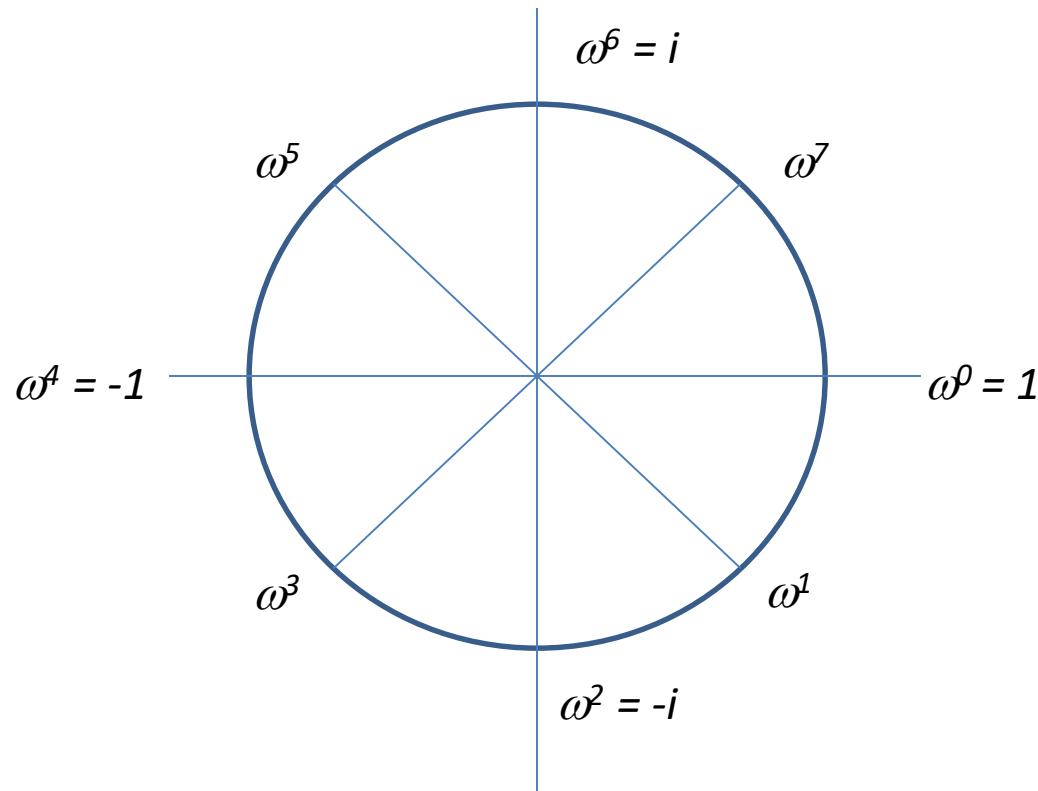
Complex Exponentials

- Euler's Formula: $e^{i\theta} = \cos \theta + i \sin \theta$
- Thus raising e to a power corresponds to taking the sine and cosine of an angle
- $e^{i\theta}$ is a periodic function; when $\theta = 0$ or multiples of 2π , it has value 1.
- $\omega = e^{-2\pi i/n}$ is an n -th root of one for any integer n

Example with $n = 8$

- $\omega^0 = (e^{-2\pi i/8})^0 = 1$
- $\omega^1 = (e^{-2\pi i/8})^1 = e^{(-\pi/4)i} = \cos(-\pi/4) + i\sin(-\pi/4) = \sqrt{2}/2 - \sqrt{2}/2i$
- $\omega^2 = (e^{-2\pi i/8})^2 = e^{(-\pi/2)i} = \cos(-\pi/2) + i\sin(-\pi/2) = -i$
- *etc.*

n^{th} roots of unity for $n = 8$



Generated from the principal root $\omega = e^{-2\pi i/n}$

Orthogonal Basis

- Start with the first root (1) and take its first n powers to get the n -dimensional vector $[1, 1, 1, \dots, 1]$
- Repeat with each of the other roots for a given n . The second root ω gives the vector $[1, \omega, \omega^2, \dots, \omega^{n-1}]$.
- For the example $n=8$, there are 8 orthogonal vectors:
 - $[1, 1, 1, 1, 1, 1, 1, 1]$
 - $[1, \omega, \omega^2, \dots, \omega^7]$
 - $[1, \omega^2, \omega^4, \dots, \omega^{14}]$
 - $[1, \omega^3, \omega^6, \dots, \omega^{21}]$
 - ...
 - $[1, \omega^7, \omega^{14}, \dots, \omega^{49}]$

1D Discrete Fourier Transform

Fourier
coefficients

basis as a matrix

original
vector

$$\begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega & \dots & \omega^{n-1} \\ \vdots & & & \\ 1 & \omega^{n-1} & \dots & \omega^{n^2-2n+1} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{bmatrix}$$

This corresponds to the formula commonly used to define the Discrete Fourier Transform:

$$F_k = \sum_{j=0}^{n-1} e^{-2\pi jki/n} v_j$$

2D Discrete Fourier Transform

- Detect the frequency components of the image in both the horizontal and vertical directions
- First apply the 1D discrete Fourier transform to each row of the image, producing an intermediate image or row transforms
- Then apply the 1D discrete Fourier transform to each column of the intermediate image
- The result is the 2D discrete Fourier transform of the image, which is an image in the frequency domain.

Filtering with the Fourier Transform

1. Compute the 2D discrete Fourier transform of the image
2. Apply an image operator to the transformed image
3. Computer the inverse 2D discrete Fourier transform of the result of the image operator

Two most common filters:

1. low-pass filtering zeroes out the frequency components above a threshold
2. high-pass filtering zeroes out the frequency components below a threshold

Example from Steve Tanimoto's Book Low-Pass Filter: Smooths

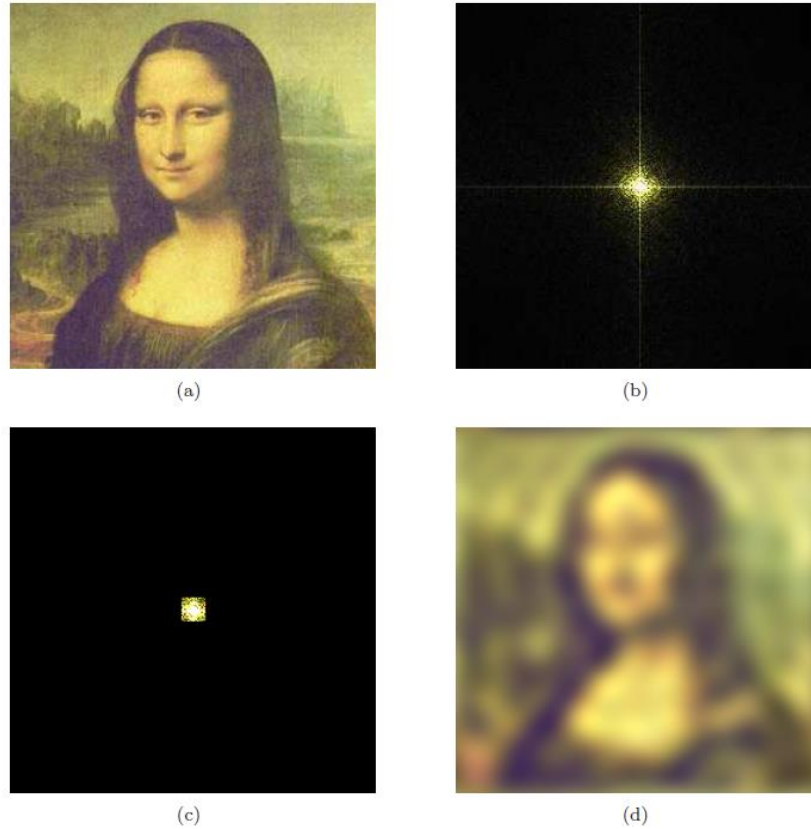
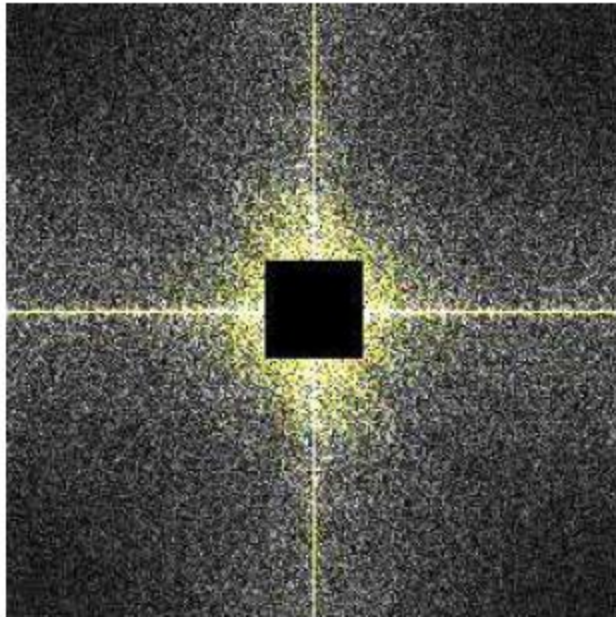


Figure 10.7: Low-pass filtering with the Mona Lisa image: (a) original, (b) 2D Discrete Fourier Transform, (c) edited transform, and (d) inverse of the edited transform.

High-Pass Filter: Finds Edges



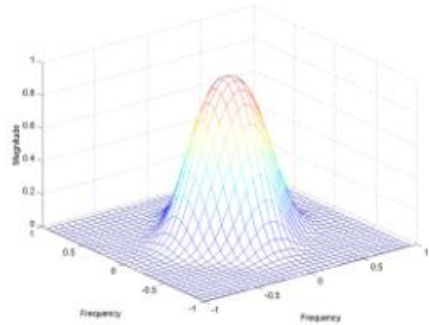
(a)



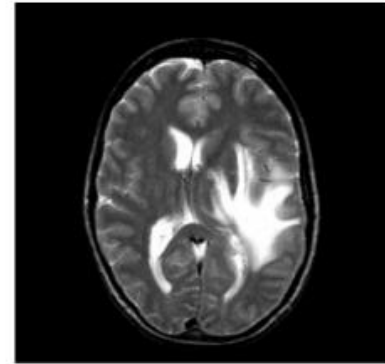
(b)

Figure 10.8: High-pass filtering with the Mona Lisa image: (a) edited transform, and (b) inverse of the edited transform.

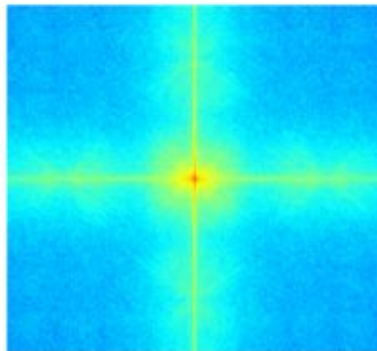
Low-Pass Filtering



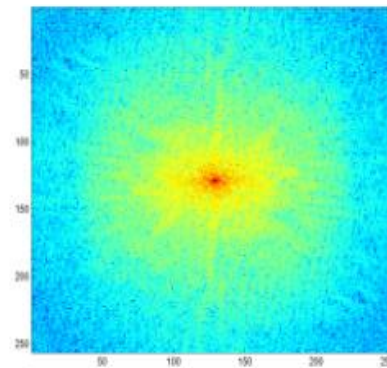
low-pass filter $H(u,v)$ in the frequency domain



filtered MRI brain image

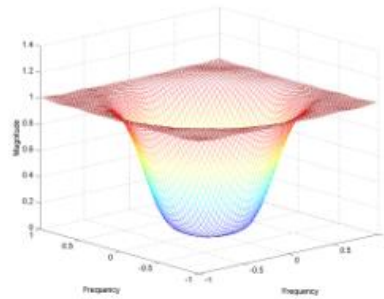


Fourier transform of original image

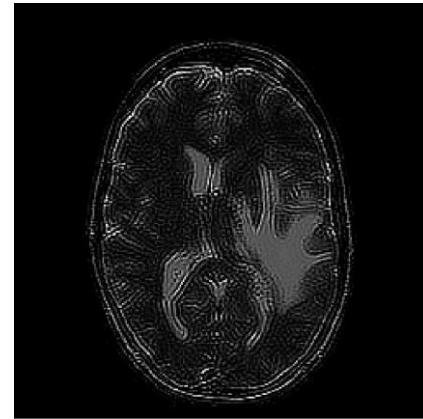


Fourier transform of filtered image

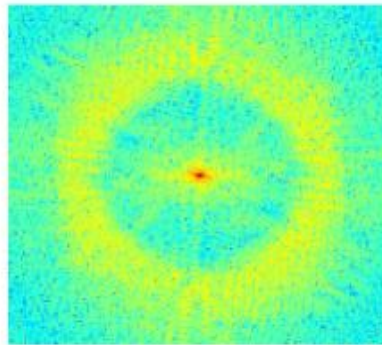
High-Pass Filtering



high-pass filter



filtered image



Fourier transform of filtered image

Wavelet Transform

- Fourier Transform only provides frequency information.
- Wavelet Transform is a method for complete time-frequency localization for signal analysis and characterization.
- Wavelet Transform : works like a microscope focusing on **finer time resolution as the scale becomes small** to see how the impulse gets better localized at higher frequency permitting a local characterization

Basics of Wavelets*

- Wavelets are a mathematical tool for **hierarchically decomposing functions**.
- They allow a function to be described in terms of a coarse overall shape, plus details that range from narrow to broad.
- Wavelets represent the signal or image as a linear combination of basis functions.
- The simplest basis is the **Haar wavelet basis**.

* Material from Stollnitz et al., Wavelets for Computer Graphics: A Primer Part 1.

How the Haar Wavelet Transform Works in 1D

- Image Pyramid

level 2: original signal with 4 pixels: [9 7 3 5]

level 1: averaged image with 2 pixels: [8 4]

level 0: averaged image with 1 pixel: [6]

- To recover the original four pixels in level 2 from the two pixels at level 1, use two detail coefficients +1 and -1.

$$9 = 8 + (+1)$$

$$7 = 8 - (+1)$$

$$3 = 4 + (-1)$$

$$5 = 4 - (-1)$$

level 2: [9 7 3 5]

level 1: [8 4]

level 0: [6]

- To recover the two pixels in level 1 from the one pixels at level 0, use one **detail coefficient of 2**.

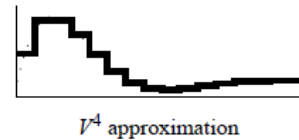
$$8 = 6 + 2$$

$$4 = 6 - 2$$

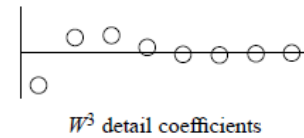
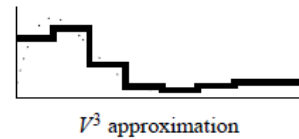
- The wavelet transform of the original 4-pixel image is
[6 2 1 -1]
- It has the same number of coefficients as the original image, but it lets us reconstruct the image at any resolution.

Idea of the Wavelet Hierarchy

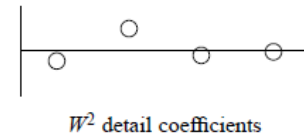
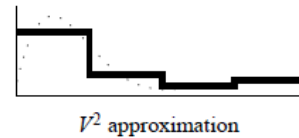
level 4



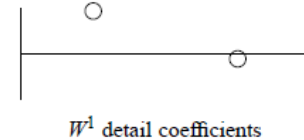
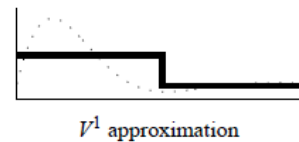
level 3



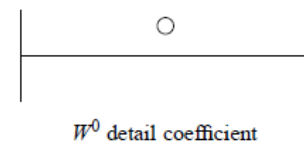
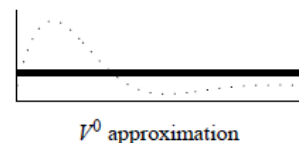
level 2



level 1



level 0



Haar Wavelet Decomposition of [9 7 3 5]

[9 7 3 5]

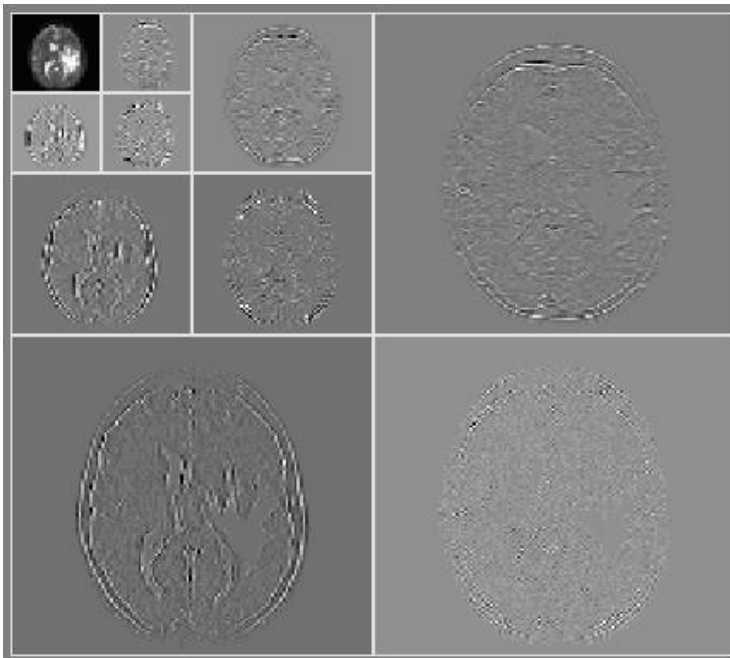
$$\begin{aligned} &= 6 \times \text{[rectangle from 0 to 4]} \\ &+ 2 \times \text{[rectangle from 0 to 2]} \\ &\quad - \text{[rectangle from 2 to 4]} \\ &+ 1 \times \text{[rectangle from 0 to 1]} \\ &\quad - \text{[rectangle from 1 to 2]} \\ &\quad + \text{[rectangle from 2 to 3]} \\ &\quad - \text{[rectangle from 3 to 4]} \end{aligned}$$

Two-Dimensional Haar Wavelet Transform: Standard Decomposition

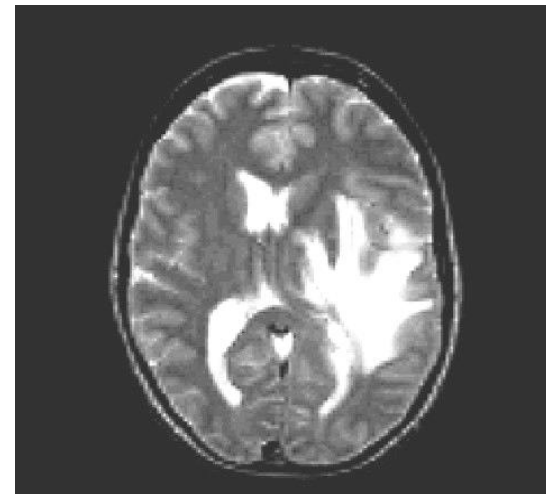
- First apply the 1D wavelet transform to each row of pixel values
- This gives an average value along with detail coefficients for each row
- Next treat these transformed rows as an image and apply the 1D transform to each column
- The resulting values are all detail coefficients except for the single overall average coefficient

Wavelet Decomposition of MRI Brain Image

3 –level decomposition

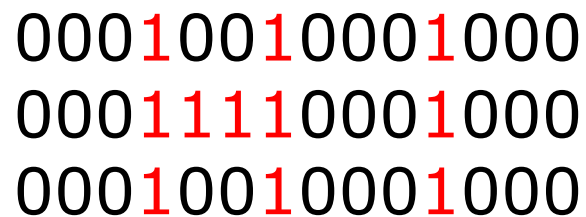


the original MRI image



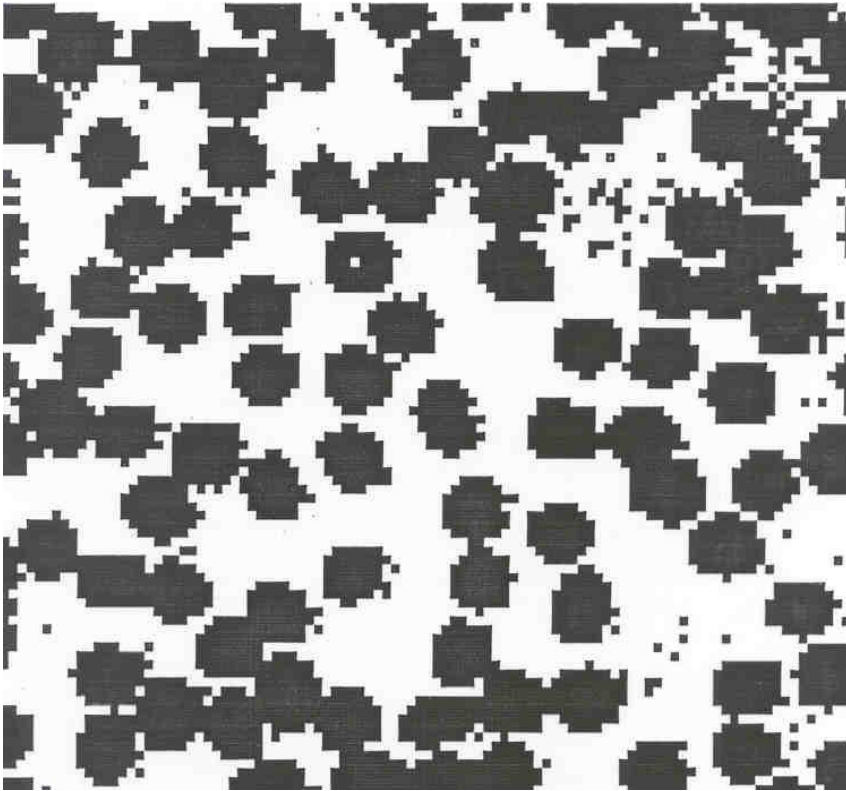
Binary Image Analysis

- used in a variety of applications:
 - part inspection
 - riveting
 - fish counting
 - document processing
- consists of a set of image analysis operations that are used to produce or process binary images, usually images of 0's and 1's.



```
00010010001000
00011110001000
00010010001000
```

Example: red blood cell image



- Many blood cells are separate objects
- Many touch – bad!
- Salt and pepper noise from thresholding
- What operations are needed to clean it up?

Results of analysis

- 63 separate objects detected
- Single cells have area about 50
- Noise spots
- Gobs of cells

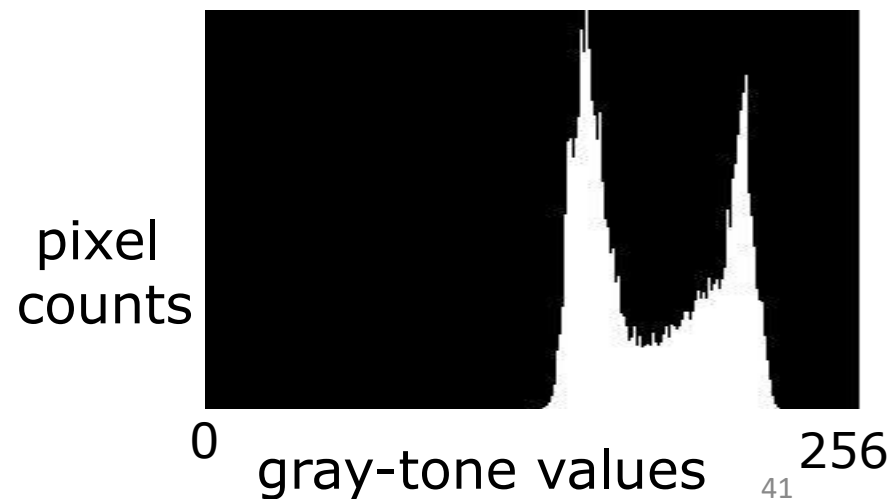
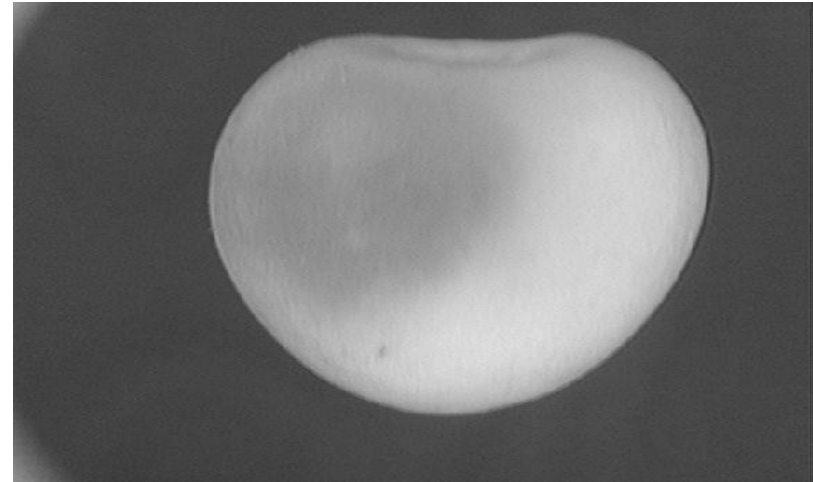
Object	Area	Centroid	Bounding Box	
=====				
1	383	(8.8 , 20)	[1 22 1 39]	
2	83	(5.8 , 50)	[1 11 42 55]	
3	11	(1.5 , 57)	[1 2 55 60]	
4	1	(1 , 62)	[1 1 62 62]	
5	1048	(19 , 75)	[1 40 35 100]	gobs
32	45	(43 , 32)	[40 46 28 35]	cell
33	11	(44 , 1e+02)	[41 47 98 100]	
34	52	(45 , 87)	[42 48 83 91]	cell
35	54	(48 , 53)	[44 52 49 57]	cell
60	44	(88 , 78)	[85 90 74 82]	
61	1	(85 , 94)	[85 85 94 94]	
62	8	(90 , 2.5)	[89 90 1 4]	
63	1	(90 , 6)	[90 90 6 6]	

Useful Operations

- 1. Thresholding a gray-tone image**
- 2. Determining good thresholds**
- 3. Filtering with mathematical morphology**
- 4. Connected components analysis**
- 5. Numeric feature extraction**
 - location features**
 - gray-tone features**
 - shape features ...**

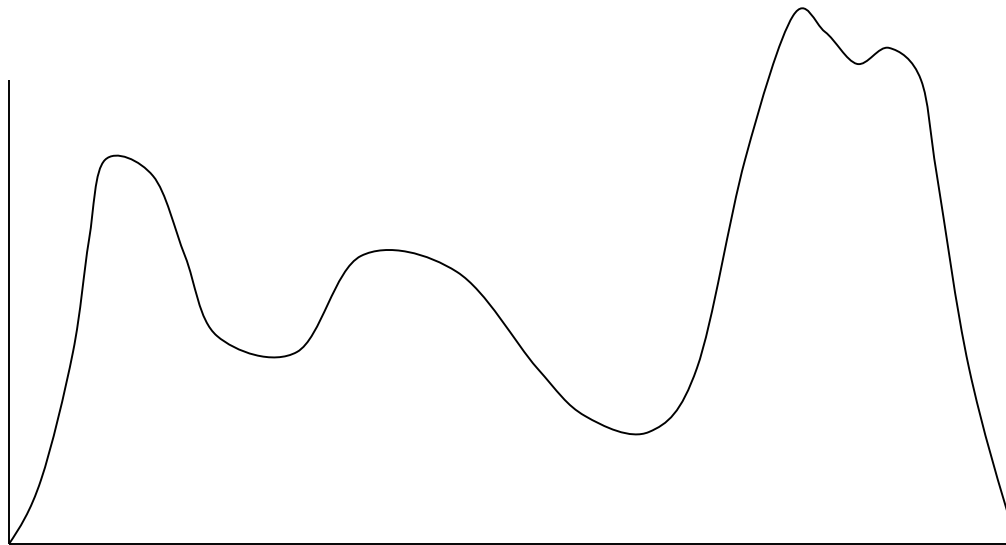
Thresholding

- Background is black
- Healthy cherry is bright
- Bruise is medium dark
- Histogram shows two cherry regions (black background has been removed)



Histogram-Directed Thresholding

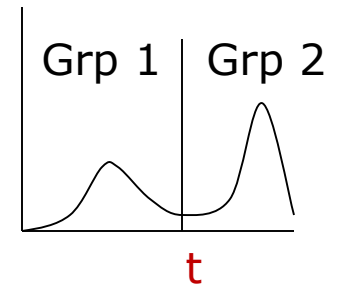
How can we use a histogram to separate an image into 2 (or several) different regions?



Is there a single clear threshold? 2? 3?

Automatic Thresholding: Otsu's Method

Assumption: the histogram is bimodal



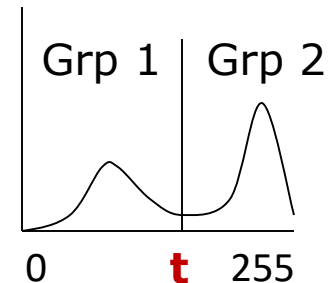
Method: find the threshold t that minimizes the **weighted sum of within-group variances** for the two groups that result from separating the gray tones at value t .

Computing Within-Group Variance

- For each possible threshold t
- For each group i (1 and 2)
 - compute the **variance** of its gray tones $\sigma_i^2(t)$
 - compute its **weight** $q_i(t) = \sum_{\tau \text{ in Grp } i} P(\tau)$

where $P(\tau)$ is the normalized histogram value for gray tone τ , normalized by sum of all gray tones in the image and called the **probability of τ** .

- Within-group variance = $q_1(t) \sigma_1^2(t) + q_2(t) \sigma_2^2(t)$



See S&S text (Section 3.8) for the efficient recurrence relations; in practice, this operator works very well for true bimodal distributions and not too badly for others, but not the CTs.

Thresholding Example



original image



pixels above threshold

Mathematical Morphology

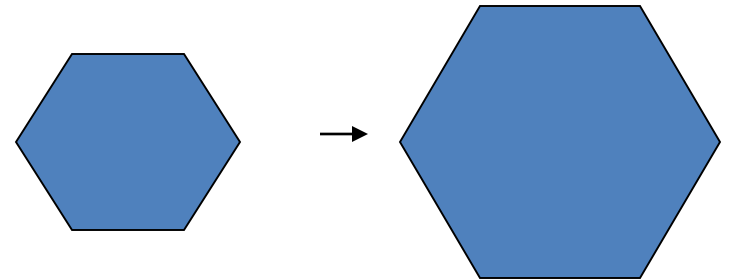
(Dilation, Erosion, Closing, Opening)

- **Dilation**

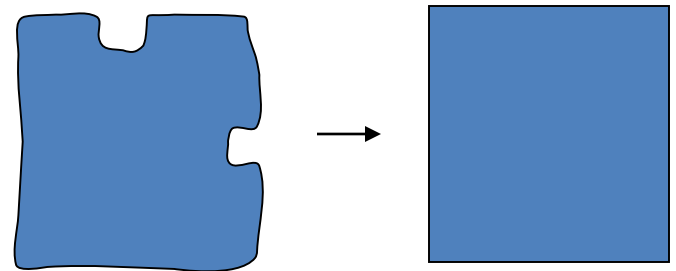
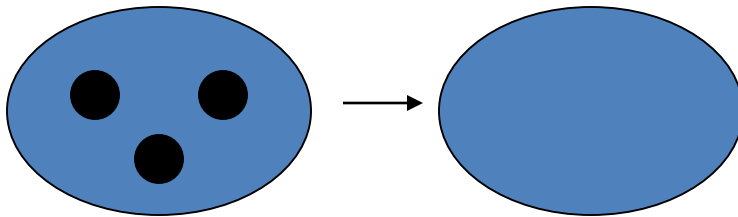
Dilation **expands** the connected sets of 1s of a binary image.

It can be used for

1. growing features



2. filling holes and gaps

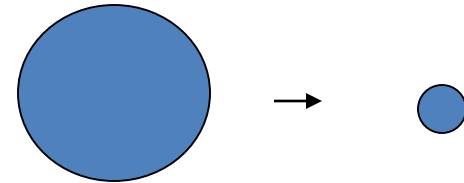
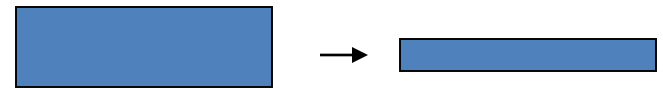


• Erosion

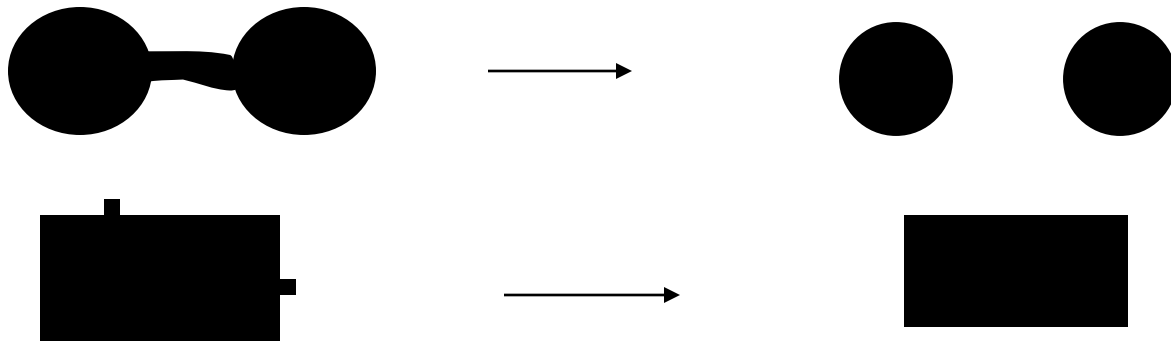
Erosion **shrinks** the connected sets of 1s of a binary image.

It can be used for

1. shrinking features



2. Removing bridges, branches and small protrusions



Structuring Elements

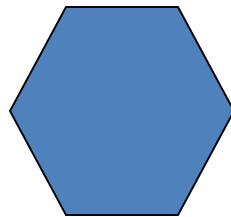
A **structuring element** is a shape mask used in the basic morphological operations.

They can be any shape and size that is digitally representable, and each has an **origin**.

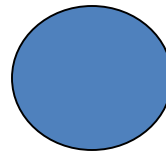


box

`box(length,width)`

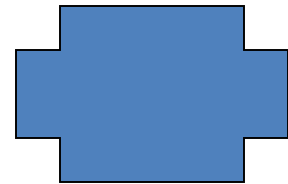


hexagon



disk

`disk(diameter)`



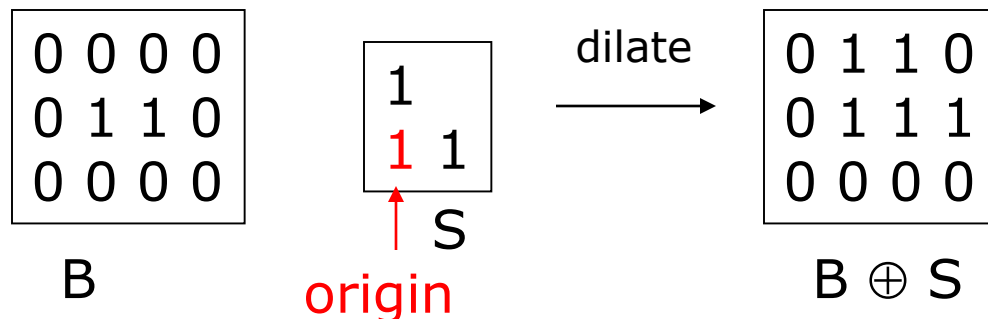
something

Dilation with Structuring Elements

The arguments to dilation and erosion are

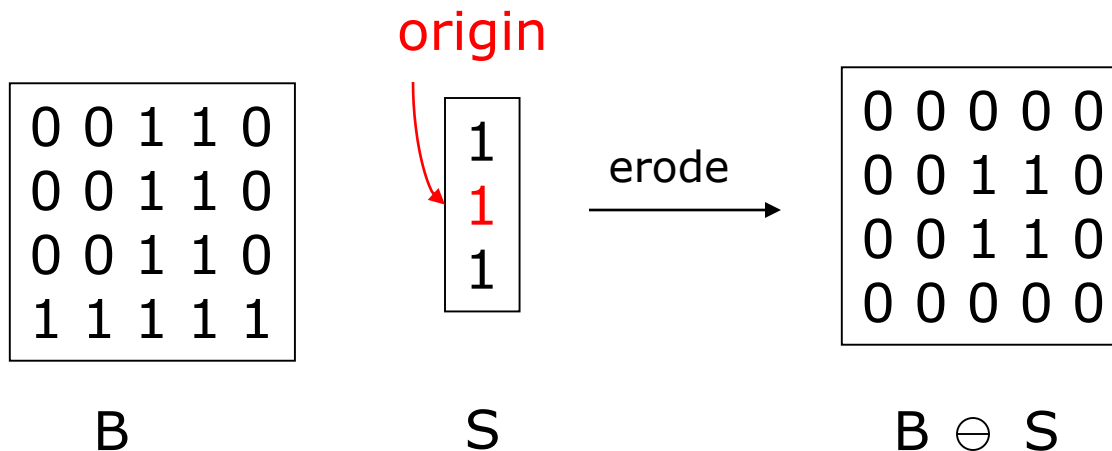
- 1. a binary image B**
- 2. a structuring element S**

`dilate(B,S)` takes binary image B, places the origin of structuring element S over each 1-pixel, and ORs the structuring element S into the output image at the corresponding position.



Erosion with Structuring Elements

erode(B,S) takes a binary image B, places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image only if every position of S (with a 1) covers a 1 in B.



Opening and Closing

- **Closing** is the compound operation of dilation followed by erosion (with the same structuring element)
- **Opening** is the compound operation of erosion followed by dilation (with the same structuring element)

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

a) Binary image B

1	1	1
1	1	1
1	1	1

b) Structuring Element S

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

c) Dilation $B \oplus S$

				1	1		
				1	1		
				1	1		

d) Erosion $B \ominus S$

	1	1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1				

e) Closing $B \bullet S$

			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	

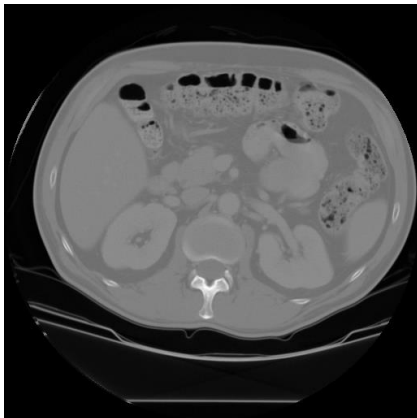
f) Opening $B \circ S$

Connected Components Labeling

Once you have a binary image, you can identify and then analyze each **connected set of pixels**.

The connected components operation takes in a binary image and produces a **labeled image** in which each pixel has the integer label of either the background (0) or a component.

original



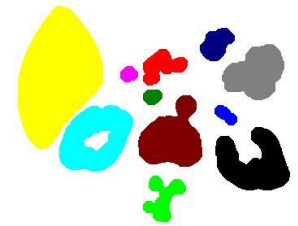
thresholded



opening+closing



components



Methods for CC Analysis

1. Recursive Tracking (almost never used)
2. Parallel Growing (needs parallel hardware)
3. Row-by-Row (most common)
 - a. propagate labels down to the bottom, recording equivalences
 - b. Compute equivalence classes
 - c. Replace each labeled pixel with the label of its equivalence class.


Equivalent Labels

Original Binary Image

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

Equivalent Labels

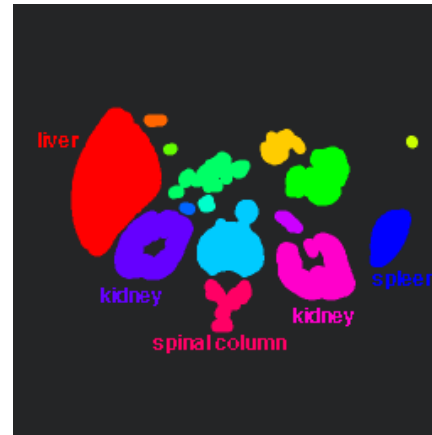
The Labeling Process:
Left to Right, Top to Bottom



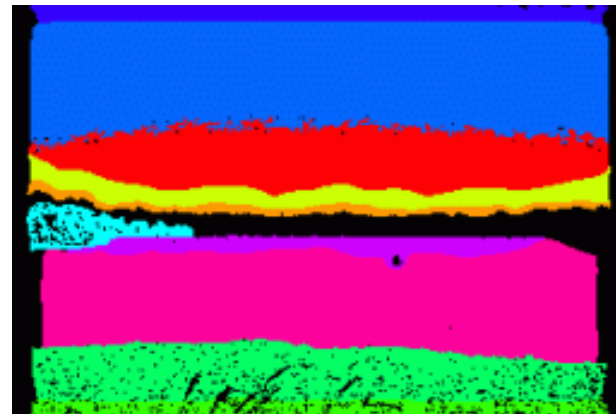
0	0	0	1	1	1	0	0	0	0	2	2	2	2	0	0	0	0	3
0	0	0	1	1	1	1	0	0	0	2	2	2	2	0	0	0	3	3
0	0	0	1	1	1	1	1	0	0	2	2	2	2	0	0	3	3	3
0	0	0	1	1	1	1	1	1	0	2	2	2	2	0	0	3	3	3
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	3	3	3
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	3	3	3
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1

$$\begin{array}{l} 1 \equiv 2 \\ 1 \equiv 3 \end{array}$$

Labeling shown as Pseudo-Color

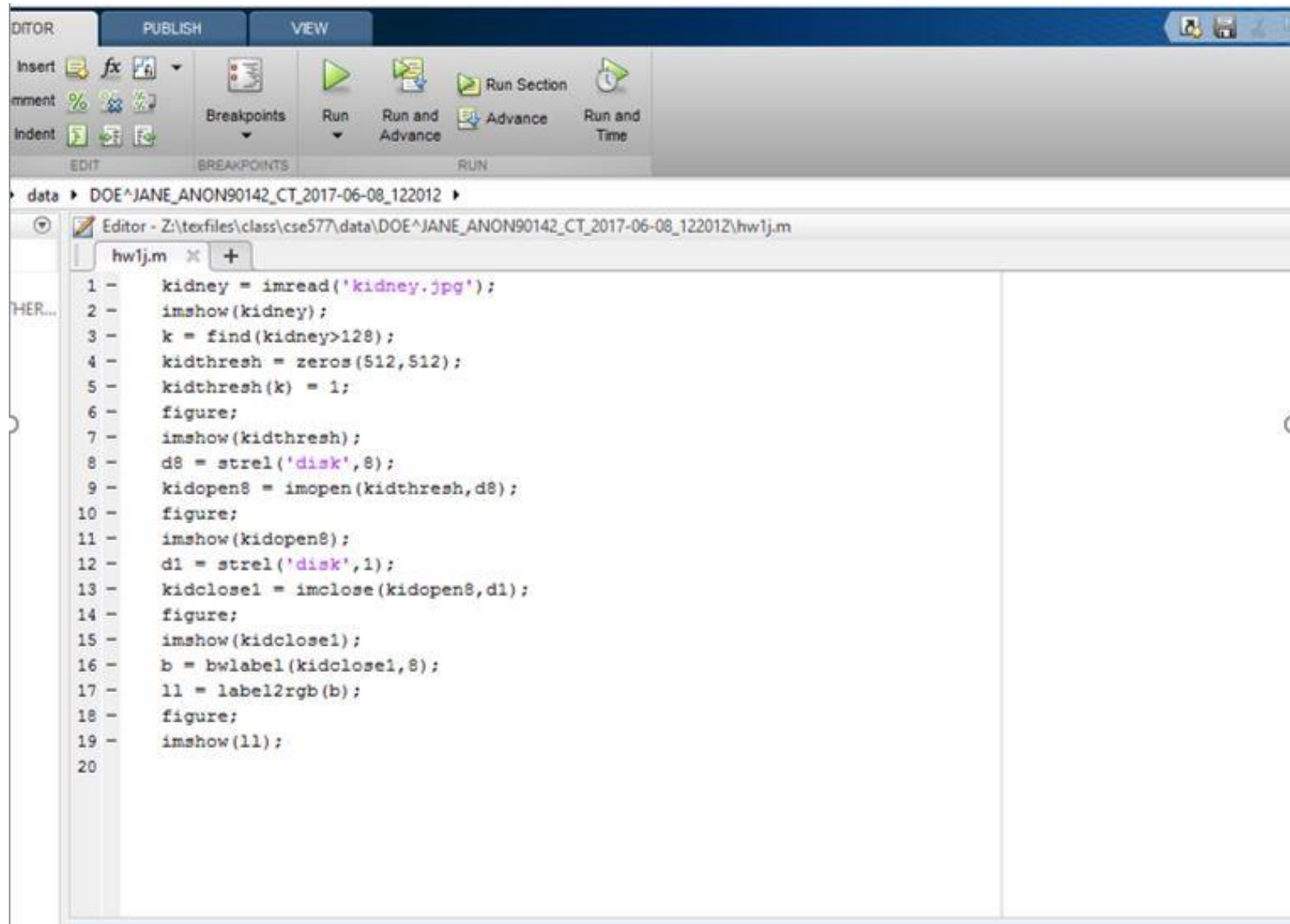


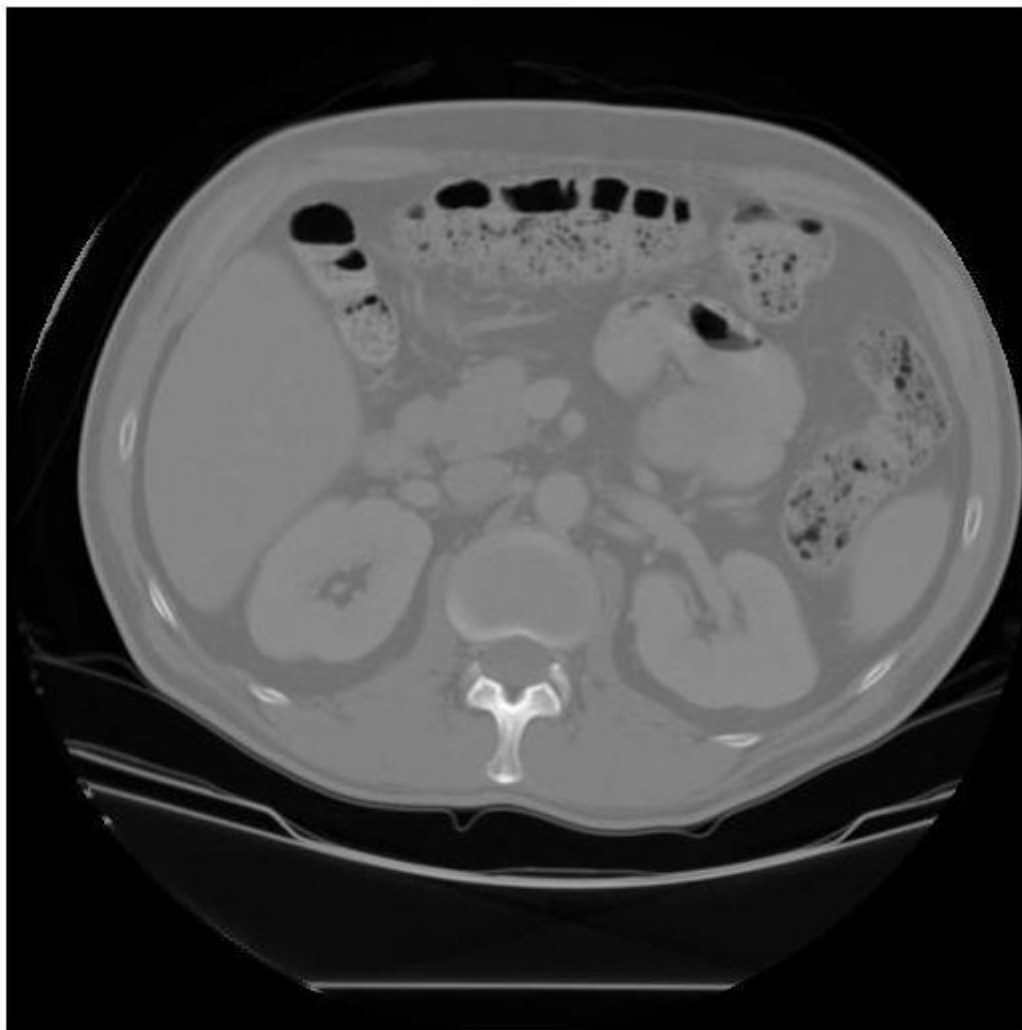
connected
components
of 1's from
thresholded
image



connected
components
of cluster
labels

Matlab Exercise

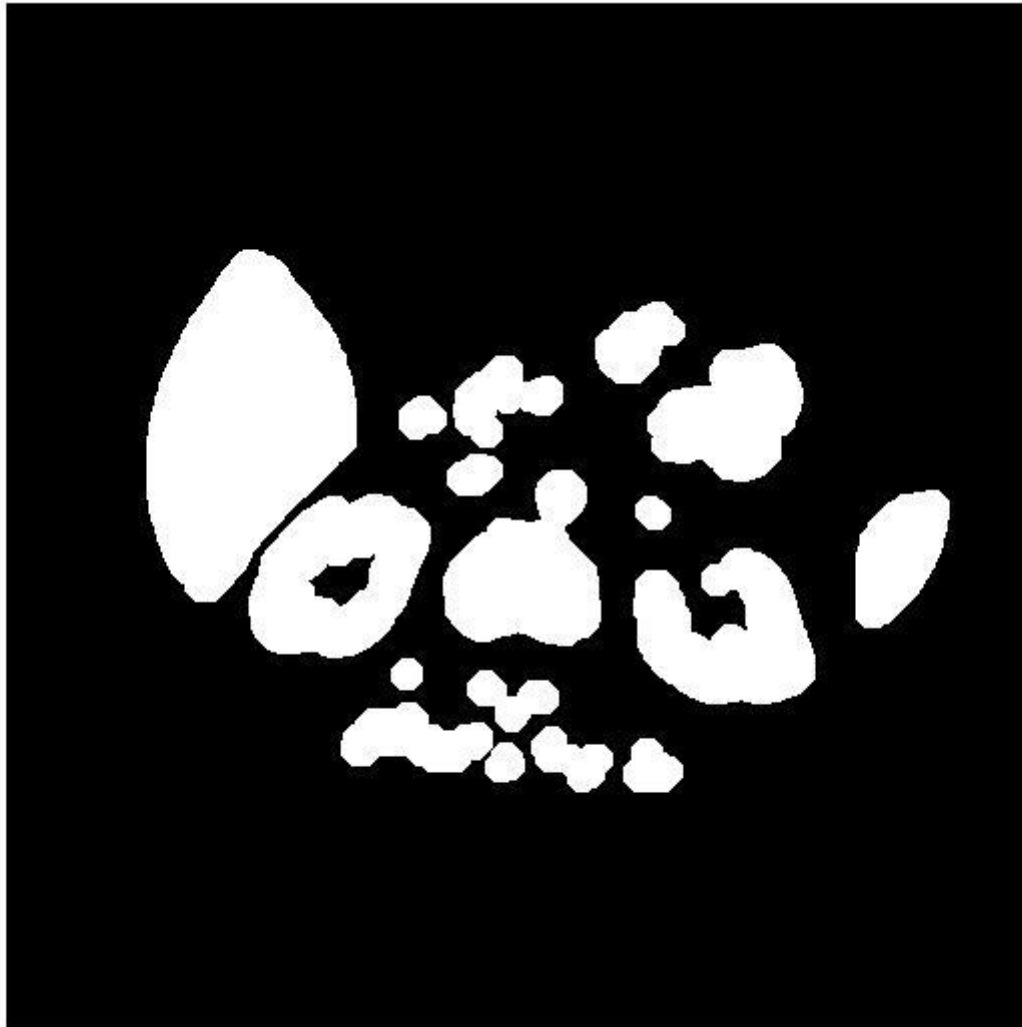




original image kidney.jpg



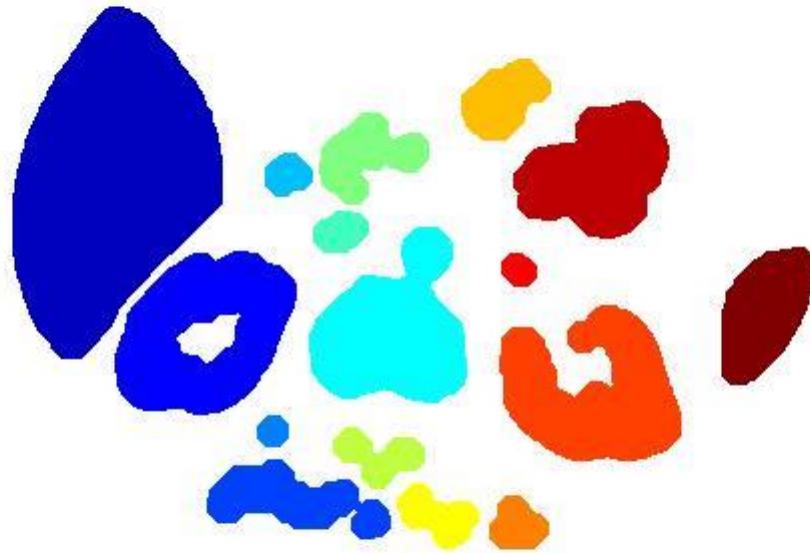
result of thresholding pixels > 128



result of opening with a disk of radius 8



and closing with a disk of radius 1



result of connected components labeling
and coloring the labels.

Warmup Exercise:

You do NOT turn anything in, but we can discuss results

- Use Matlab to run this script and get the same outputs.
- Try playing around with parameters to see what you get.
- Try the other two images that were part of the 2011 first assignment.
- The 2011 assignment is here:
- <https://courses.cs.washington.edu/courses/cs577/17au/assignments/warmup.html>