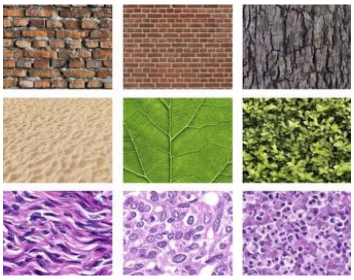
COMP 448/548: Medical Image Analysis

Texture analysis

Çiğdem Gündüz Demir
cgunduz@ku.edu.tr

1

Handcrafted feature extraction for medical images

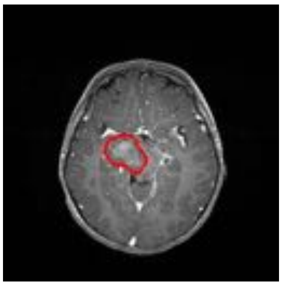


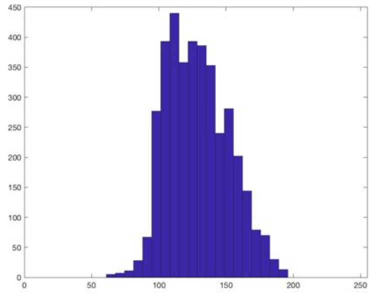
- Intensity features
  - Quantify color/grayscale distribution of pixels
  - First-order statistical texture
- Texture features
  - Similar structures repeated over and over again (repeated patterns)
  - Statistical approach is to quantify spatial arrangement of pixel intensities
  - Structural approach is to define texture on objects/primitives
- Morphological features to quantify the shape/size of a segmented object
- Structural features to quantify spatial distribution of objects/primitives

2

First-order statistical features

- Extracted based on histogram analysis
  - Mean
  - Standard deviation
  - Skewness
  - Kurtosis
  - Entropy
  - Min, max, quartiles
- More useful if you first identify regions of interest



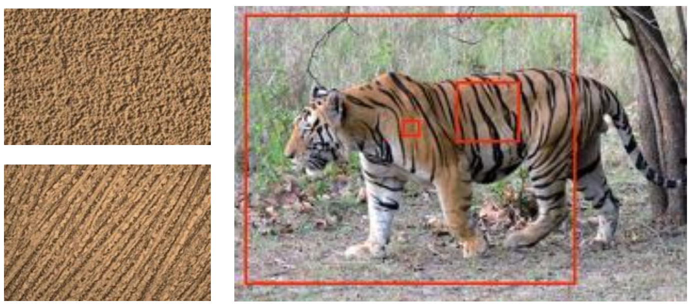


3

Texture

- Statistical approaches
  - Co-occurrence matrices
  - Run length matrices
  - Local binary patterns
  - Laws kernels
  - Gabor filters
  - …

Three images with the same intensity histogram, but different textures



Texture depends on the scale at which it is viewed

4

Gray-level co-occurrence matrix

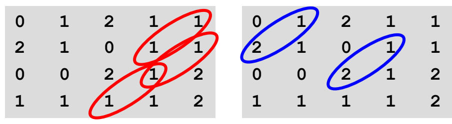
- Second-order statistical features
- A gray-level co-occurrence matrix P is an NxN array, where N is the number of gray levels in the image
- P(i, j) gives how many times gray-levels i and j co-occur at a given distance d = (di, dj)

0	1	2	1	1
2	1	0	1	1
0	0	2	1	2
1	1	1	1	2

Image

1	1	0
2	3	2
0	3	0

Co-occurrence matrix for d = (1, −1)



5

Gray-level co-occurrence matrix

- Common to use normalized co-occurrence matrix $N(i,j) = \frac{P(i,j)}{\sum \sum P(u,v)}$
- Sometimes useful to group gray-levels into bins
  - E.g., Four bins: [0, 63], [64, 127], [128, 191], [192, 255]
- Sometimes useful to accumulate co-occurrence matrices calculated for different distances
  - For example, for a rotation invariant image, it is common to calculate P(i, j) at d1 = (di, dj), d2 = (–di, dj), d3 = (di, –dj), and d4 = (–di, –dj) and take their summation
- Statistical features are computed from the co-occurrence matrix to represent the texture more compactly *(Haralick features)*

6

Haralick features

$$\text{Angular second moment} = \sum\sum N(i,j)^2$$

$$\text{Maximum probability} = \max N(i,j)$$

$$\text{Inverse difference moment} = \sum\sum \frac{N(i,j)}{1+(i-j)*(i-j)}$$

$$\text{Contrast} = \sum\sum (i-j)^2 N(i,j)$$

$$\text{Entropy} = -\sum\sum N(i,j) \log N(i,j)$$

$$\text{Correlation} = \frac{\sum\sum i\, j\, N(i,j) - \mu_i \mu_j}{\sigma_i \sigma_j}$$

μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of

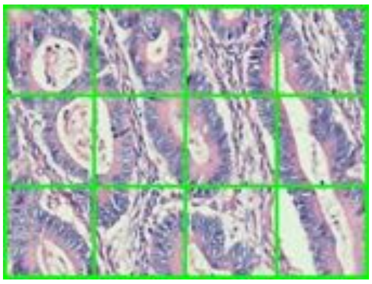
$N_i = \sum_j N(i,j)$ and $N_j = \sum_i N(i,j)$, respectively.

Haralick et al., "Textural features for image classification," IEEE Transactions on Systems, Man, and Cybernetics, 1973.

7

Image classification using Haralick features

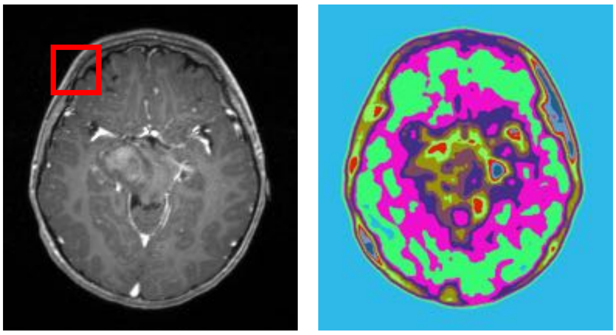
- Calculate a co-occurrence matrix on an entire image
- Calculate the Haralick features (or a subset of them) of this matrix
- Classify the image based on these features
- Grid-based approach
  - Divide an image into grids
  - Calculate a co-occurrence matrix on each grid
  - Calculate the Haralick features of each calculated matrix
  - Average the feature values
  - Classify the image based on these average features

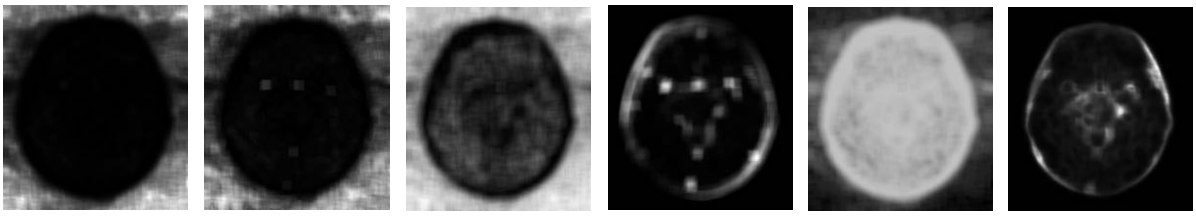


8

Image segmentation using Haralick features

- Use the sliding window approach and obtain maps of these features
- Use these maps in a segmentation algorithm (e.g., use clustering)





9

Run-length matrix

- *Run*: Consecutive, collinear pixels with the same value in a specified direction
- Each matrix entry R(i,l) keeps the number of runs with a value of i and a length of l

1	1	2	2	1
2	2	2	2	3
1	1	3	3	3
3	3	3	1	1

Example run

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

Run-length matrix for $\theta = 0°$ →

10

Run-length matrix features

$$\text{Short run emphasis} = \frac{1}{n_r}\sum_i\sum_l R(i,l) \,/\, l^2$$

$$\text{Long run emphasis} = \frac{1}{n_r}\sum_i\sum_l R(i,l) \,.\, l^2$$

$$\text{Gray level nonuniformity} = \frac{1}{n_r}\sum_i\left(\sum_l R(i,l)\right)^2$$

$$\text{Run length nonuniformity} = \frac{1}{n_r}\sum_l\left(\sum_i R(i,l)\right)^2$$

$$\text{Run percentage} = \frac{n_r}{n_p}$$

n_r and n_p are the total numbers of runs and pixels, respectively

Galloway, "Texture analysis using gray level run lengths," Computer Graphics and Image Processing, 1975.

11

Local binary patterns

- For each pixel p
  - Compare its value with the value of its 8-adjacent pixels
  - Create a binary string $b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8$
  - b_i = 0 if the pixel value of neighbor n_i is less than the value of p
  - b_i = 1 otherwise
- Represent an image with the histogram of the numbers represented by the binary strings of all image pixels

	n1	n2	n3	
	80	90	60	
n8	71	70	50	n4
	251	91	90	
	n7	n6	n5	

1 1 0 0 1 1 1 1

Binary string

Ojala et al., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, " IEEE PAMI, 2002.

12

Texture definition by filters

- Convolve an image with a set of filters and use filter responses to define texture features
  - Laws kernels
  - Gabor filters
  - Edge filters
  - …

13

Laws' kernels

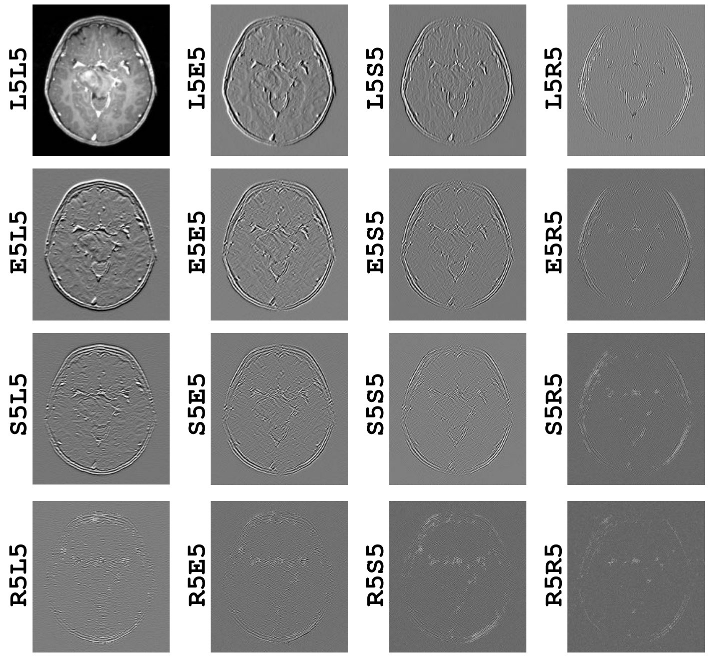
- Creates two-dimensional filters derived from the following vectors
  - To calculate a center-weighted average: **L5 (level) = [1 4 6 4 1]**
  - To detect edges: **E5 (edge) = [-1 -2 0 2 1]**
  - To detect spots: **S5 (spot) = [-1 0 2 0 -1]**
  - To detect ripples: **R5 (ripple) = [1 -4 6 -4 1]**
  - Example: E5L5 is computed as the product of E5 and L5

$$\begin{pmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \end{pmatrix} = \begin{pmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

14

Laws' kernels

- Image classification
  - For each kernel, accumulate responses over pixels
    - Mean, standard deviation, maximum, entropy, etc.
  - Possible to use a grid-based approach (see Slide 8)
- Image segmentation
  - Use response maps in a segmentation algorithm
    - Clustering (see Slide 9), region growing, etc.



Gabor filters

- Closely related to the function of primary visual cortex cells in primates
- Achieve simultaneous localization in both spatial and frequency domains

$$g(x,y) = s(x,y)\, w(x,y)$$

$s(x,y)$: complex sinusoid (**carrier**)

$w(x,y)$: 2D Gaussian-shaped function (**envelope**)

Gabor filters

$$g(x,y) = \exp\left(i\left(\frac{2\pi x'}{\lambda}+\psi\right)\right)\exp\left(-\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)\right)$$

$$g(x,y) = \underbrace{\cos\left(\frac{2\pi x'}{\lambda}+\psi\right)\exp\left(-\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)\right)}_{\text{Real part}} +$$

$$\underbrace{i\sin\left(\frac{2\pi x'}{\lambda}+\psi\right)\exp\left(-\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)\right)}_{\text{Imaginary part}}$$

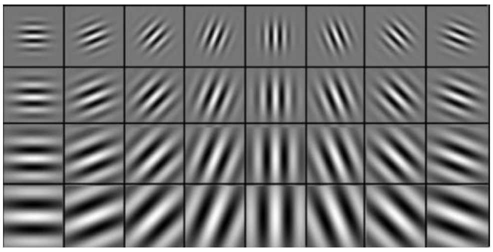
where $x' = x\cos\theta + y\sin\theta$ *and* $y' = -x\sin\theta + y\cos\theta$

x and y — spatial coordinates
x' and y' — rotated spatial coordinates
Θ — orientation of the Gabor function

σ_x and σ_y — standard deviations of the Gaussian envelope

λ — wavelength of the sinusoid

Ψ — phase offset of the sinusoid

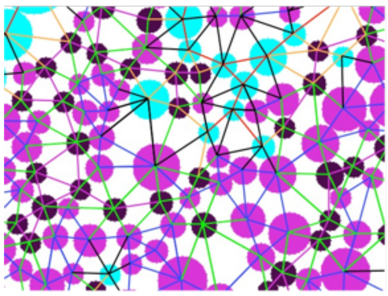


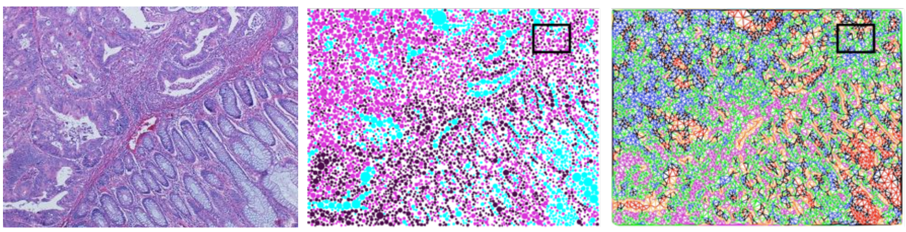
You can use the filter responses for image classification or image segmentation (see Slides 8, 9, and 15)

17

Example: Structural approach for tissue segmentation

- Represent an image with a graph of cytological tissue components
  - Approximate their locations with circles
  - Construct a Delaunay triangulation on the circles
  - Color each triangle edge according to the circle (component) type of its end-nodes



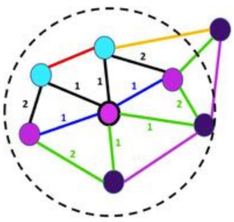


Tosun and Gunduz-Demir, "Graph run-length matrices for histopathological image segmentation," IEEE T Medical Imaging, 2011.

18

Example: Structural approach for tissue segmentation

- Define texture descriptors on this color graph
  - Gray-level run: Consecutive, collinear pixels with the same gray-level value in a specified direction
  - Graph-run: Path containing triangle edges of the same color
  - For each circle, locate a window and extract paths from this circle to every other circle within this window using *breadth first traversal*



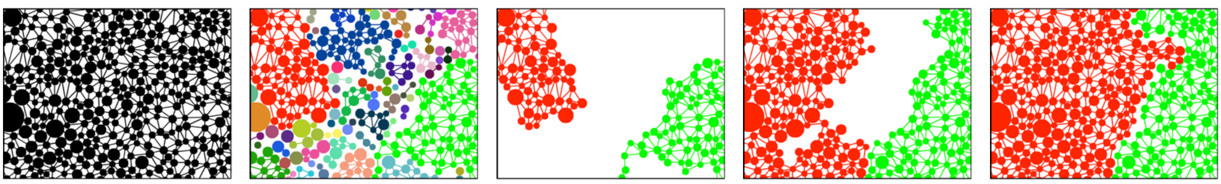
type\run	1	2	3
red	0	0	0
pink	0	0	0
black	0	2	0
blue	2	0	0
green	0	2	0
orange	0	0	0

For pixels	For objects
Short run emphasis	Short path emphasis
Long run emphasis	Long path emphasis
Gray-level nonuniformity	Edge type nonuniformity
Run length nonuniformity	Path length nonuniformity

Tosun and Gunduz-Demir, "Graph run-length matrices for histopathological image segmentation," IEEE T Medical Imaging, 2011.

Example: Structural approach for tissue segmentation

- Design a seed-controlled region growing algorithm on this graph
  - *Seed initialization*
    - ➢ Disconnect adjacent circles if the distance between their texture descriptors is greater than *a distance threshold*
    - ➢ Find connected components of the circles and eliminate those smaller than *a size threshold*
  - *Region growing*
    - ➢ Iteratively grow the seeds on the remaining circles with respect to the texture descriptors



Tosun and Gunduz-Demir, "Graph run-length matrices for histopathological image segmentation," IEEE T Medical Imaging, 2011.

Thank you!

Next time:

More on feature extraction