Mid-Level Operations for Segmentation
Recall: Thresholding Example

original image                        pixels above threshold
original image kidney.jpg
Image Segmentation Methods from Dhawan (ch 10)

- Edge Detection
- Boundary Tracking
- Hough Transform
- Thresholding (we just covered)
- Clustering
- Region Growing (and Splitting)
- Estimation-Model Based
- Using Neural Networks (we do semantic segmentation this way)
We’ll look at

- Thresholding (we just covered)
- Edge Detection
- Hough Transform
- Clustering
- Using Neural Networks (we do semantic segmentation this way)
What’s an edge?

- Image is a function
- Edges are rapid changes in this function
Finding edges

- Could take derivative
- Edges = high response
To find edges, we use filters

• Use masks for filters
• Define a small mask
• Apply it to every pixel position in the image to produce a new output image
• In general, this is called filtering.
• We call linear filters CONVOLUTIONS (even though they are really correlations).
Averaging Filters
Smooth first, then derivative

\[
\frac{1}{2} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \ast \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \ast \text{grid}
\]
Smooth first, then derivative

\( \frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \ast \)
Smooth first, then derivative

\[
\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ 1 & 2 & 4 \\ 2 & 4 & 2 \end{pmatrix} \ast \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}
\]

\[
\frac{1}{2} \times \begin{bmatrix} 2 \end{bmatrix}
\]
Smooth first, then derivative

\[
\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}
\]
Smooth first, then derivative

\[
\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ -1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix} \ast \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -2 \end{pmatrix}
\]
Smooth first, then derivative

$\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \ast$ 

$\frac{1}{2} \times \begin{pmatrix} 2 & 0 & -2 \end{pmatrix}$
Smooth first, then derivative

$\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \ast \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$

$\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \ast \begin{pmatrix} 2 & 0 & -2 \\ 4 & 0 & 4 \\ 2 & 0 & 2 \end{pmatrix}$
Smooth first, then derivative

\[
\frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \ast \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -2 \\ 4 & 0 \end{pmatrix}
\]
Smooth first, then derivative
Smooth first, then derivative

\[
\frac{1}{\sqrt{2}} \times \left(\begin{array}{ccc}
-1 & 0 & 1 \\
1 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1
\end{array}\right) \ast \left(\begin{array}{ccc}
1 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1
\end{array}\right)
\]

\[
\frac{1}{\sqrt{2}} \times \left(\begin{array}{ccc}
2 & 0 & -2 \\
4 & 0 & -4 \\
2 & 0 & -2
\end{array}\right)
\]
Smooth first, then derivative

\[ \frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \]

\[ \rightarrow \frac{1}{2} \times \begin{pmatrix} 2 & 0 & -2 \\ 4 & 0 & -4 \\ 2 & 0 \end{pmatrix} \]
Smooth first, then derivative

\[ \frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \ast \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \]
Sobel filter! Smooth & derivative

\[ \frac{1}{2} \times \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \ast \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \]
2nd derivative!

- Crosses zero at extrema
Canny Edge Detection

- Your first image processing pipeline!
  - Old-school CV is all about pipelines

Algorithm:

- 1. Smooth image (only want “real” edges, not noise)
- 2. Calculate gradient direction and magnitude
- 3. Non-maximum suppression perpendicular to edge
- 4. Threshold into strong, weak, no edge
- 5. Connect together components

http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/
Canny Characteristics

• The Canny operator gives single-pixel-wide images with good continuation between adjacent pixels

• It is the most widely used edge operator today; no one has done better since it came out in the late 80s. Many implementations are available.

• It is very sensitive to its parameters, which need to be adjusted for different application domains.
Canny on Kidney
An edge is not a line...

How can we detect *lines*?
Finding lines in an image

• Option 1:
  – Search for the line at every possible position/orientation
  – What is the cost of this operation?

• Option 2:
  – Use a voting scheme: Hough transform
Finding lines in an image

- Connection between image (x,y) and Hough (m,b) spaces
  - A line in the image corresponds to a point in Hough space
  - To go from image space to Hough space:
    - given a set of points (x,y), find all (m,b) such that y = mx + b
Hough transform algorithm

• Typically use a different parameterization

\[ d = x \cos \theta + y \sin \theta \]

- \( d \) is the perpendicular distance from the line to the origin
- \( \theta \) is the angle of this perpendicular with the horizontal.
Hough transform algorithm

• Basic Hough transform algorithm

1. Initialize $H[d, \theta] = 0$
2. for each edge point $I[x, y]$ in the image
   compute gradient magnitude $m$ and angle $\theta$
   \[
   d = x\cos\theta + y\sin\theta
   \]
   \[
   H[d, \theta] += 1
   \]
3. Find the value(s) of $(d, \theta)$ where $H[d, \theta]$ is maximum
4. The detected line in the image is given by
   \[
   d = x\cos\theta + y\sin\theta
   \]

Complexity? How do you get the lines out of the matrix?
Line segments from Hough Transform

Fig. 7. Puppet scenes 211, 212, 214, 215 and the edges recovered by the algorithm.
Extensions

- Extension 1: Use the image gradient (we just did that)

- Extension 2
  - give more votes for stronger edges

- Extension 3
  - change the sampling of \((d, \theta)\) to give more/less resolution

- Extension 4
  - The same procedure can be used with circles, squares, or any other shape, How?

- Extension 5; the Burns procedure. Uses only angle, two different quantifications, and connected components with votes for larger one.
Finding lung nodules (Kimme & Ballard)

Fig. 4.7 Using the Hough technique for circular shapes. (a) Radiograph. (b) Window. (c) Accumulator array for $r = 3$. (d) Results of maxima detection.
K-Means Clustering

Form K-means clusters from a set of n-dimensional vectors

1. Set ic (iteration count) to 1

2. Choose randomly a set of K means \( m_1(1), \ldots, m_K(1) \).

3. For each vector \( x_i \) compute \( D(x_i, m_k(ic)) \), \( k=1,\ldots,K \) and assign \( x_i \) to the cluster \( C_j \) with nearest mean.

4. Increment ic by 1, update the means to get \( m_1(ic),\ldots,m_K(ic) \).

5. Repeat steps 3 and 4 until \( C_k(ic) = C_k(ic+1) \) for all \( k \).
Simple Example

INIT.

K=2
Arbitrarily choose K objects as initial cluster center

Assign each object to most similar center

Update the cluster means

reassign

Update the cluster means

reassign
Space for K-Means

- The example was in some arbitrary 2D space
- We don’t want to cluster in that space.
- We will be clustering in gray-scale space or color space.
- K-means can be used to cluster in any n-dimensional space.
K-Means Example 1
K-Means Example 2
K-Means Example 3

1. Select an image: imgs/2a170028.jpg
2. Select a processor: KMCluster
3. Click process>>

Options:
- Init Method: 0

Process done!
- Image size: 640*480
- (607,118): RGB(20,22,1)
- (228,36): RGB(255,170,0)
K-Means Example 4

Original 2-D MR image

After K-means clustering

Segmentation using traditional watershed algorithm

Final segmentation

2756 partitions

172 partitions

Published in 2006 IEEE Southwest Symposium on Image Analysis and Interpretation 2006
Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm
H. P. Ng, S. Ong, K. Foong, P. Goh, W. Nowinski
K-Means Example 5

<table>
<thead>
<tr>
<th>Original MR Image</th>
<th>After K-means clustering</th>
<th>Final Segmentation map</th>
<th>Segmentation map using conventional algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Published in 2006 IEEE Southwest Symposium on Image Analysis and Interpretation 2006 Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm H. P. Ng, S. Ong, K. Foong, P. Goh, W. Nowinski
K-Means Example 5

- Superpixel clustering in breast biopsy images

*benign*  *atypia*  *DCIS*
K-means Variants

- Different ways to initialize the means
- Different stopping criteria
- Dynamic methods for determining the right number of clusters (K) for a given image

- The EM Algorithm: a probabilistic formulation of K-means
Blobworld: Sample Results using color, texture, and EM
Semantic Segmentation

• Instead of grouping pixels based on color, texture or whatever properties
• Teach a classifier what important regions look like, so it can find them.
• This is usually done via deep learning, which we will discuss later in the course.
• But here’s a preview.
Meaning of Labels

- **Benign Epithelium**: epithelial cells from the benign and atypia categories
- **Malignant Epithelium**: epithelial cells from DCIS and invasive cancer
- **Normal Stroma**: normal connective tissue
- **Desmoplastic Stroma**: stroma associated with a tumor
- **Secretion**: benign substance filling the ducts
- **Necrosis**: dead cells at the center of the ducts in DCIS and invasive cases
- **Blood**: blood cells
- **Background**: empty areas inside ducts
Superpixel + SVM-based Segmentation

Ground Truth

no neighborhood

1 neighborhood

2 neighborhoods

color and texture histograms

- background
- benign epithelium
- normal stroma
- secretion
- necrosis

- malignant epithelium
- desmoplastic stroma
- blood
CNN-based Segmentation

Input Image
256 × 256

384 × 384

Encoder-Decoder

Encoder-Decoder

Ground Truth

Plain

Multi-Resolution

Segmentation
256 × 256

- background
- benign epithelium
- malignant epithelium
- normal stroma
- secretion
- necrosis
- desmoplastic stroma
- blood
Supervised Tissue Label Segmentation

Superpixel + SVM

• Each superpixel is assigned a class label.
• Context: Two circular neighborhoods
• Relatively simple model
• Faster to train (~3 hours)

CNN

• Each pixel is assigned a class label.
• Context: 256x256 and 384x384 pixel patches
• More complex model
• ~1 week to train on special hardware
Results

Mean $F_1$-score

<table>
<thead>
<tr>
<th></th>
<th>SP+SVM</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>precision</td>
<td>0.40</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Results

### Precision

![Precision Chart](chart.png)

### Recall

![Recall Chart](chart.png)
# Confusion Matrices

<table>
<thead>
<tr>
<th>Ground Truth</th>
<th>Background</th>
<th>Benign Epithelium</th>
<th>Malignant Epithelium</th>
<th>Normal Stroma</th>
<th>Desmoplastic Stroma</th>
<th>Secretion</th>
<th>Blood</th>
<th>Necrosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>.89</td>
<td>.01</td>
<td>.01</td>
<td>.00</td>
<td>.02</td>
<td>.07</td>
<td>.00</td>
<td>.01</td>
</tr>
<tr>
<td>Benign Epithelium</td>
<td>.02</td>
<td>.27</td>
<td>.31</td>
<td>.03</td>
<td>.27</td>
<td>.08</td>
<td>.00</td>
<td>.02</td>
</tr>
<tr>
<td>Malignant Epithelium</td>
<td>.02</td>
<td>.10</td>
<td>.47</td>
<td>.00</td>
<td>.24</td>
<td>.16</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>Normal Stroma</td>
<td>.05</td>
<td>.01</td>
<td>.06</td>
<td>.28</td>
<td>.35</td>
<td>.19</td>
<td>.04</td>
<td>.03</td>
</tr>
<tr>
<td>Desmoplastic Stroma</td>
<td>.04</td>
<td>.03</td>
<td>.17</td>
<td>.03</td>
<td>.61</td>
<td>.11</td>
<td>.00</td>
<td>.01</td>
</tr>
<tr>
<td>Secretion</td>
<td>.04</td>
<td>.02</td>
<td>.07</td>
<td>.27</td>
<td>.14</td>
<td>.20</td>
<td>.21</td>
<td>.05</td>
</tr>
<tr>
<td>Blood</td>
<td>.03</td>
<td>.01</td>
<td>.05</td>
<td>.13</td>
<td>.23</td>
<td>.05</td>
<td>.46</td>
<td>.04</td>
</tr>
<tr>
<td>Necrosis</td>
<td>.12</td>
<td>.04</td>
<td>.13</td>
<td>.00</td>
<td>.26</td>
<td>.20</td>
<td>.01</td>
<td>.24</td>
</tr>
</tbody>
</table>

**Superpixels + SVM**

<table>
<thead>
<tr>
<th>Ground Truth</th>
<th>Background</th>
<th>Benign Epithelium</th>
<th>Malignant Epithelium</th>
<th>Normal Stroma</th>
<th>Desmoplastic Stroma</th>
<th>Secretion</th>
<th>Blood</th>
<th>Necrosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>.93</td>
<td>.01</td>
<td>.01</td>
<td>.03</td>
<td>.01</td>
<td>.01</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>Benign Epithelium</td>
<td>.01</td>
<td>.72</td>
<td>.16</td>
<td>.09</td>
<td>.01</td>
<td>.01</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>Malignant Epithelium</td>
<td>.06</td>
<td>.09</td>
<td>.61</td>
<td>.13</td>
<td>.07</td>
<td>.02</td>
<td>.00</td>
<td>.03</td>
</tr>
<tr>
<td>Normal Stroma</td>
<td>.03</td>
<td>.02</td>
<td>.01</td>
<td>.88</td>
<td>.05</td>
<td>.00</td>
<td>.01</td>
<td>.00</td>
</tr>
<tr>
<td>Desmoplastic Stroma</td>
<td>.06</td>
<td>.02</td>
<td>.05</td>
<td>.66</td>
<td>.20</td>
<td>.00</td>
<td>.01</td>
<td>.00</td>
</tr>
<tr>
<td>Secretion</td>
<td>.15</td>
<td>.06</td>
<td>.05</td>
<td>.07</td>
<td>.02</td>
<td>.49</td>
<td>.07</td>
<td>.09</td>
</tr>
<tr>
<td>Blood</td>
<td>.01</td>
<td>.01</td>
<td>.01</td>
<td>.10</td>
<td>.04</td>
<td>.00</td>
<td>.83</td>
<td>.00</td>
</tr>
<tr>
<td>Necrosis</td>
<td>.10</td>
<td>.01</td>
<td>.11</td>
<td>.01</td>
<td>.01</td>
<td>.15</td>
<td>.01</td>
<td>.59</td>
</tr>
</tbody>
</table>

**CNN**
Segmentation Results

RGB | Ground Truth Labels | SVM Predictions | CNN Predictions
---|---------------------|-----------------|-----------------}

- background
- benign epi
- normal stroma
- secretion
- necrosis
- malignant epi
- desmoplastic stroma
- blood
Segmentation Summary

• Tissue-label segmentation is a useful abstraction.

• We developed a set of 8 tissue labels and collected pixel-label data from a pathologist on 58 ROIs.

• We trained two models: SVM and CNN

• CNNs performed significantly better than SVMs both quantitatively and qualitatively.