COMP 448/548: Medical Image Analysis

Medical image segmentation: Part 2

Çiğdem Gündüz Demir
cgunduz@ku.edu.tr

1

Image segmentation

- Partition an image into *meaningful* parts
- Each part is a connected set of similar pixels that *go together*
- Many, many different image segmentation algorithms
  - Histogram-based
  - Clustering-based
  - Region growing (watersheds)
  - Split-and-merge
  - Graph-based
  - Active contour models

→ **What is meaningful?**

→ **What do we want to be similar in each part?**

2

Seed-controlled region growing

- Iterative algorithm that starts with a set of initial seeds (markers) and grows them onto the other pixels with respect to a growing function
- Each seed can be
  - A single pixel provided externally, e.g., by a human user
  - A single pixel or a connected component of pixels identified by another algorithm
- In each iteration, a pixel adjacent to one of these seeds is selected and it is merged with the corresponding seed
  - That pixel is the "*best*" with respect to the growing function

> This usually requires selecting a pixel with the minimum or the maximum value/similarity. When a straightforward implementation is used, selecting/updating candidate pixels becomes computationally expensive. You should use a "*proper*" data structure for effective implementation.

3

Seed-controlled region growing

- Growing (marking) function can be
  - Absolute feature value of a pixel
    - For example, to select the darkest pixel from all candidates
  - (Dis)similarity between a pixel and the entire seed to which that pixel is adjacent
    - You have to effectively calculate/update similarities for the existing candidate pixels when a new pixel is added to a seed
  - (Dis)similarity between a pixel and the seed pixel to which that pixel of interest is adjacent
- Growing stops
  - When all image pixels are covered
  - When all pixels in a given mask are covered (e.g., a mask for foreground pixels)
  - When the best similarity drops below a predefined threshold

4

Growing the seeds based on the grayscale intensity similarity

77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

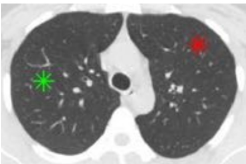
77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

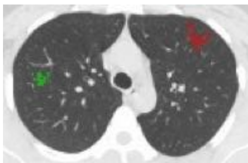
77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

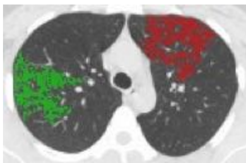
77	81	80	78	77	80	80	78	75	74
85	80	80	80	76	77	77	70	68	74
79	75	76	80	74	74	76	67	78	76
79	78	79	73	73	74	77	69	82	72
82	79	78	75	77	77	81	70	69	70

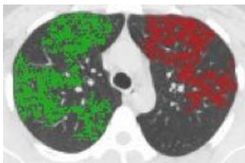
5

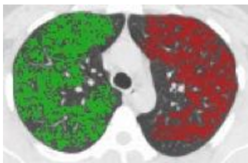
An example: Region growing on gray-level intensities for lung segmentation

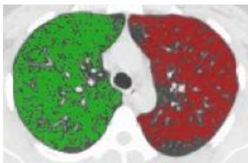

Initial seeds

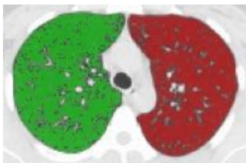

After 500 iterations


After 5000 iterations


After 10000 iterations


After 15000 iterations


After 20000 iterations

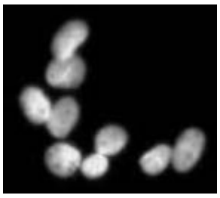

After 25000 iterations


At the stopping point
When the smallest gray-level of a candidate pixel is greater than a threshold

6

Another example: Region growing on distance transform for cell segmentation

RGB image

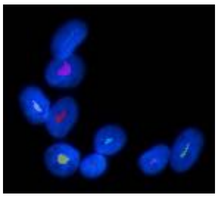


Blue channel

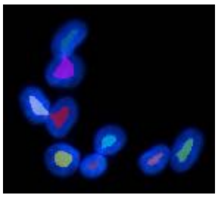
Binary image

Outer distance transform

Regional maxima on the suppressed distance map

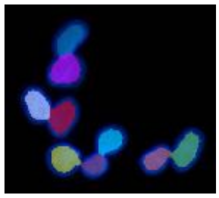


After 500 iterations



After 2500 iterations

After 5000 iterations

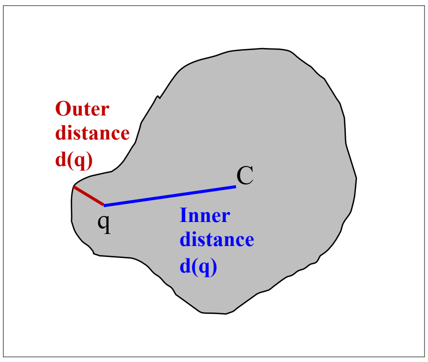


After 7500 iterations

At the stopping point When all pixels in the binary mask are covered

Distance transforms

- Measures how far each foreground pixel is from a given set of other pixels
  - Mostly, this set corresponds to the boundary pixels
  - But also possible to use other sets
- Signed distance transform is defined for both foreground and background pixels



Distance transforms

Let $A = \{a_i\}$ be a set of annotated cells,

$P(a_i) = \{p_{ik}\}$ be a set of pixels belonging to an annotated cell a_i,

$C(a_i)$ be the centroid pixel of a_i,

$B(a_i) = \{b_{ik}\}$ be a set of boundary pixels of a_i, and

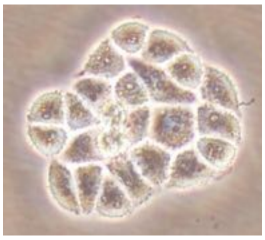
$B = \bigcup_{a_i \in A} B(a_i)$ be the union of all boundary pixels.

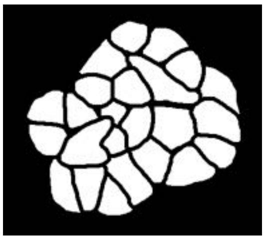
(Outer) distance transform

$$d(q) = \begin{cases} \min_{b_{ik} \in B} \|q - b_{ik}\|^2 & \text{if } q \in \text{foreground} \\ 0 & \text{if } q \in \text{background} \end{cases}$$

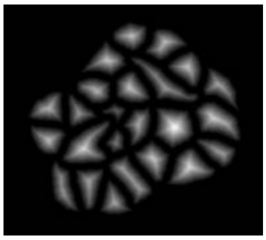
Normalized (outer) distance transform

$$d(q) = \begin{cases} \dfrac{\min_{b_{ik} \in B(a_i)} \|q - b_{ik}\|^2}{\max_{r \in P(a_i)} \min_{b_{ik} \in B(a_i)} \|r - b_{ik}\|^2} & \text{if } q \in P(a_i) \\ 0 & \text{if } q \in \text{background} \end{cases}$$

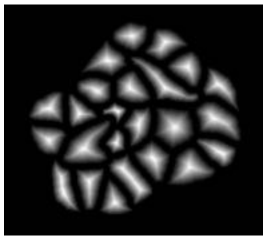




Foreground pixels



(Outer) distance transform



Normalized (outer) distance transform

9

Distance transforms

Let $A = \{a_i\}$ be a set of annotated cells,

$P(a_i) = \{p_{ik}\}$ be a set of pixels belonging to an annotated cell a_i,

$C(a_i)$ be the centroid pixel of a_i,

$B(a_i) = \{b_{ik}\}$ be a set of boundary pixels of a_i, and

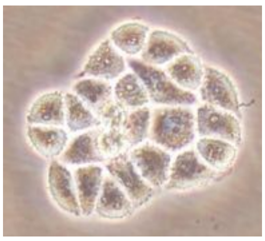
$B = \bigcup_{a_i \in A} B(a_i)$ be the union of all boundary pixels.

(Outer) distance transform

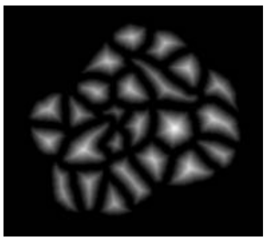
$$d(q) = \begin{cases} \min_{b_{ik} \in B} \|q - b_{ik}\|^2 & \text{if } q \in \text{foreground} \\ 0 & \text{if } q \in \text{background} \end{cases}$$

Inner distance transform

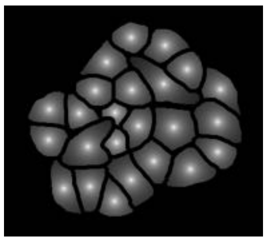
$$d(q) = \begin{cases} \dfrac{1}{1 + \alpha \, \|q - C(a_i)\|^2} & \text{if } q \in P(a_i) \\ 0 & \text{if } q \in \text{background} \end{cases}$$



Foreground pixels



(Outer) distance transform



Inner distance transform

10

Distance transforms

Let $A = \{a_i\}$ be a set of annotated cells,

$P(a_i) = \{p_{ik}\}$ be a set of pixels belonging to an annotated cell a_i,

$C(a_i)$ be the centroid pixel of a_i,

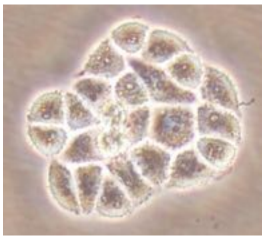
$B(a_i) = \{b_{ik}\}$ be a set of boundary pixels of a_i, and

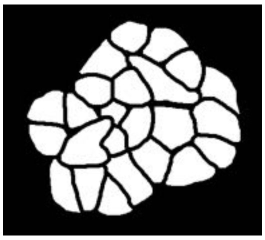
$B = \bigcup_{a_i \in A} B(a_i)$ be the union of all boundary pixels.

Signed distance transform is defined for both foreground and background pixels

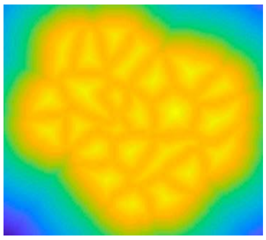
Signed distance transform

$$d(q) = \begin{cases} \min\limits_{b_{ik} \in B} \|q - b_{ik}\|^2 & \text{if } q \in \text{foreground and } q \notin B \\ 0 & \text{if } q \in B \\ -\min\limits_{b_{ik} \in B} \|q - b_{ik}\|^2 & \text{if } q \in \text{background} \end{cases}$$

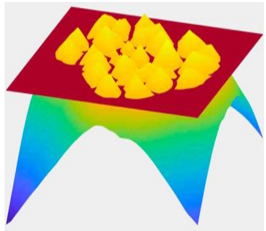




Foreground pixels



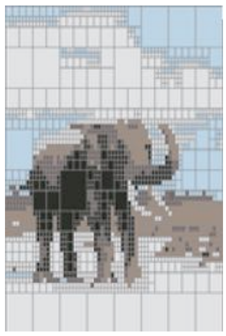
Signed distance transform

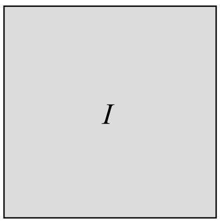


11

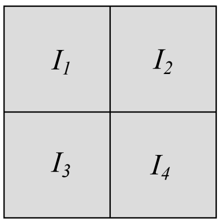
Split-and-merge

1. Start with an entire image
2. If the variance is high, split into quadrants
3. Merge any adjacent subregions that are similar enough
4. Repeat steps 2 and 3 iteratively, until no split or merge

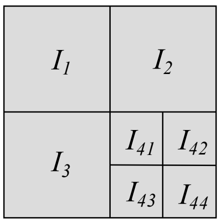




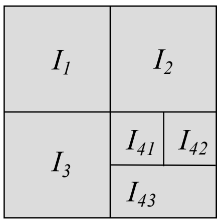
Entire image



First split



Second split

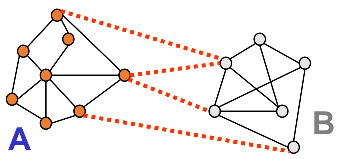


Merge

12

Graph-based segmentation

- Represent an image by a weighted graph
  - Each node is a pixel or a connected component of pixels (e.g., superpixels)
  - Edges are defined between every adjacent nodes
  - Edge weight is the (dis)similarity of the corresponding adjacent nodes
- Partition the nodes into two disjoint sets A and B by removing "*some*" of the edges
  - Corresponds to dividing an image into two segments



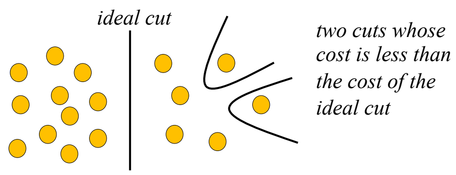
cut for A and B is a set of the removed edges that disconnects A and B

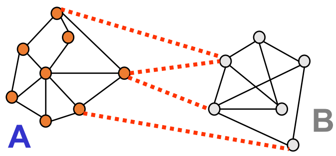
$cut(A, B)$ is the cost of this cut $\quad cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$

13

Graph-based segmentation

- *Minimum cut:* Select a cut for which the cost is the smallest





$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$

- *Normalized cut:* Normalize the costs according to the segment size and then select a cut for which the normalized cost is the smallest
  - Exact solution: NP-hard
  - Approximate solution: generalized eigenvalue problem
  - But still high computational time

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

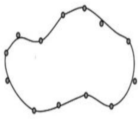
$$assoc(A,V) = \text{sum of weights of all edges that touch } A$$

Shi and Malik, "Normalized cuts and image segmentation," IEEE PAMI, 2000, https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf

14

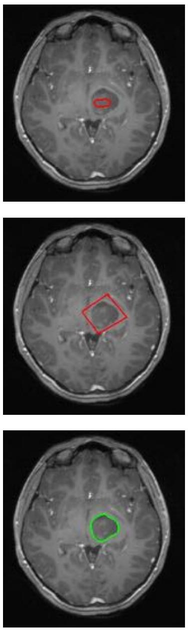
Active contour models (snakes)

- Starts with initial boundary points (active contour) and iteratively moves them to minimize an energy function
  - Boundary points are represented as a parametric curve
  - Energy function is associated with this parametric curve



$$E_{snake} = E_{internal} + E_{external} + E_{constraint}$$

- *Internal energy* is to control the continuity and smoothness of the contour
- *External energy* is to control how well the contour fits on the image data (e.g., object contours)
- *Constraint energy* is external energy due to other factors (e.g., factors introduced by a user) to guide the contour move



Kass et al., "Snakes: Active contour models,"Int J of Computer Vision, 1988, https://link.springer.com/article/10.1007/BF00133570

15

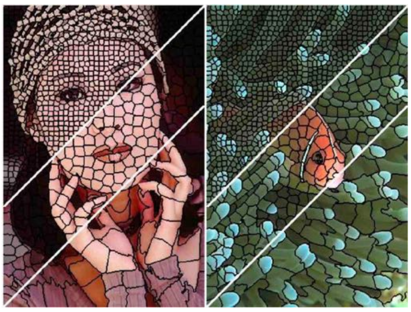
So far…

- We mostly used intensity features to quantify pixels
  - Also possible to use texture features that provide information in the spatial arrangement of intensities/colors **→ NEXT LECTURE**
- We talked about segmentation algorithms that run on pixels
  - Also possible to use other types of primitives (such as super-pixels and Voronoi polygons)

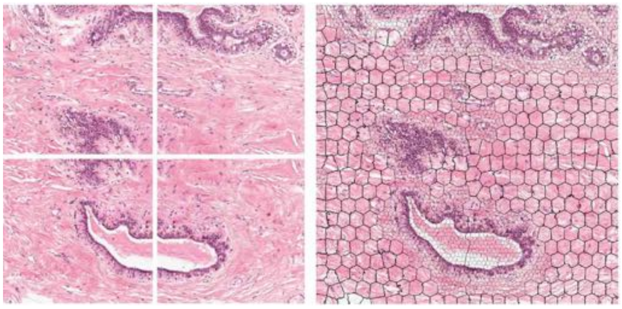
16

Superpixels

- Divide an image into a group of regions, called *superpixels*
- Superpixel is a group of pixels that share common characteristics
- Simple linear iterative clustering (SLIC) algorithm is a popular method
  Achanta et al., “SLIC superpixels compared to state-of-the-art superpixel methods,” IEEE PAMI, 2012
  https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6205760



Achanta et al., IEEE PAMI, 2012

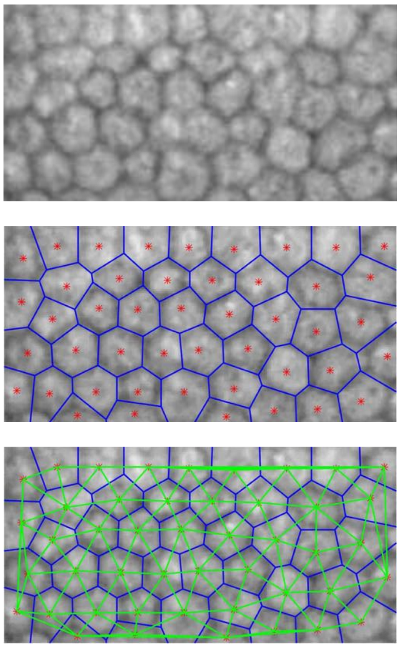


Bejnordi et al., “A multi-scale superpixel classification approach to the detection of regions of interest in whole slide histopathology images” SPIE Medical Imaging, 2015

17

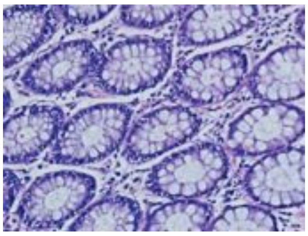
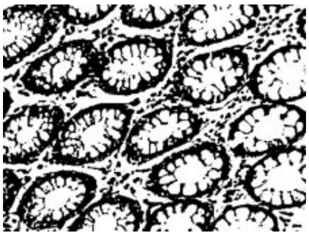
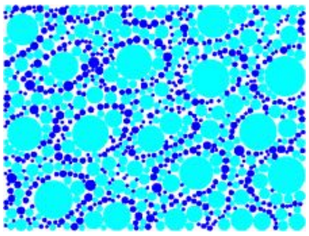
Voronoi diagram

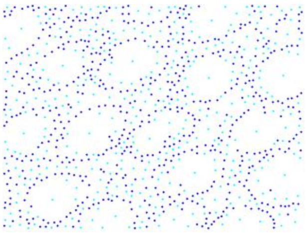
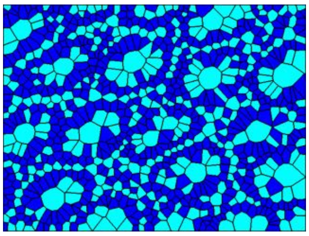
- Partitioning of a plane into convex polygons with respect to points P
  - Each polygon contains exactly one of these points
  - Every point in a given polygon is closer to its generating point than to any other points in P
- The dual graph for a Voronoi diagram corresponds to the Delaunay triangulation for the same points P



18

Example: Voronoi diagram to represent a tissue image

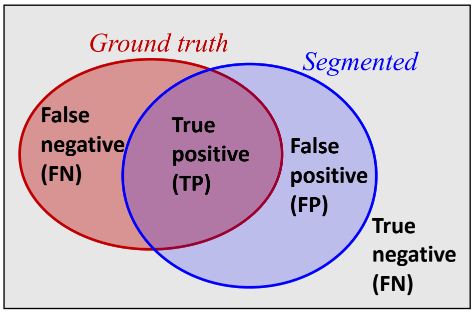
Locate two sets of circles, one set on the nucleus pixels, and the other set on the remaining ones

Construct a Voronoi diagram on their centroids

Use Voronoi polygons in your segmentation algorithm instead of using pixels

How to evaluate segmentation results?

- Pixel-level evaluation



$$\text{Sensitivity} = \frac{TP}{TP + FN} \qquad \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \qquad \text{Dice coeff} = \frac{2\,TP}{2\,TP + FP + FN}$$

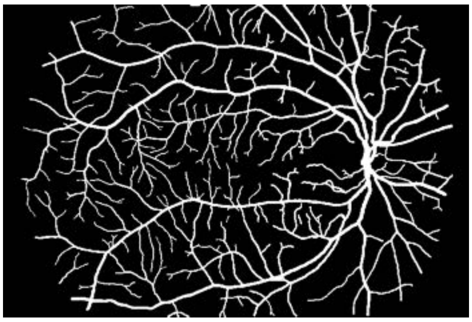
$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{F-score} = 2\,\frac{\text{Precision . Recall}}{\text{Precision + Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

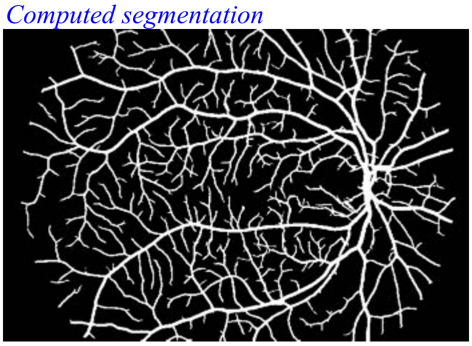
How to evaluate segmentation results?

- Sometimes, pixel-level evaluation may be misleading

Ground truth



Computed segmentation



Sensitivity = 0.9071
Specificity = 0.9474
Accuracy = 0.9410
Dice coeff = 0.8291

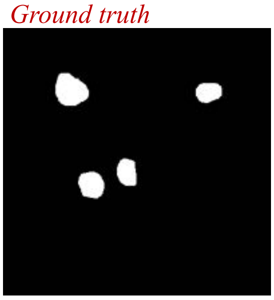
Recall = 0.9071
Precision = 0.7635
F-score = 0.8291

21

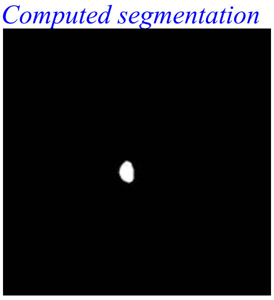
How to evaluate segmentation results?

- Sometimes, pixel-level evaluation may be misleading

Ground truth



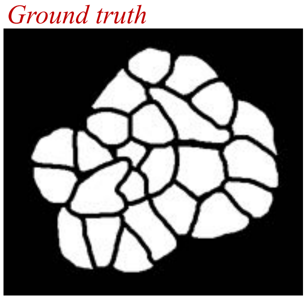
Computed segmentation



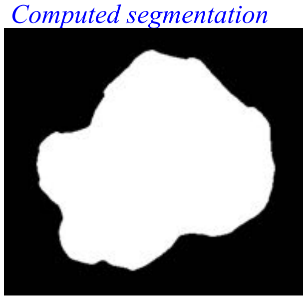
Sensitivity = 0.1071
Specificity = 1.0000
Accuracy = 0.9719
Dice coeff = 0.1935

Recall = 0.1071
Precision = 1.0000
F-score = 0.1935

Ground truth



Computed segmentation



Sensitivity = 0.9418
Specificity = 0.8564
Accuracy = 0.8883
Dice coeff = 0.8633

Recall = 0.9418
Precision = 0.7969
F-score = 0.8633

22

How to evaluate segmentation results?

- Object-level evaluation
  - Necessary for instance segmentation tasks
  - Need to match segmented instances with the ground truth objects

***Object-level F-score* is to assess what percentage of instances are correctly detected

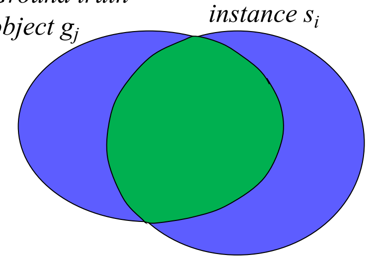
***Object-level Dice index* is to assess how accurately the pixels of the segmented instances overlap with those of their matching (maximally overlapping) ground truth objects

***Intersection over union (IoU)* is also to assess how accurately the pixels of the segmented instances and their matching ground truth objects overlap

***Object-level Hausdorff distance* is to assess the shape similarity between the segmented instances and their matching ground truth objects

23

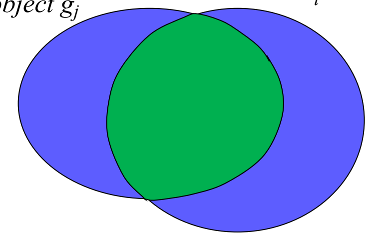
Object-level F-score



- First find TPs, but this time on objects
  - A segmented instance s_i is considered as TP if it intersects with at least 50 percent of a ground truth g_j and also g_j intersects with at least 50 percent of s_i
- You have already had the number of the segmented instances (TP + FP) and the number of the ground truth objects (TP + FN)
- Calculate precision, recall, and F-score using these values

24

Intersection-over-union (IoU)



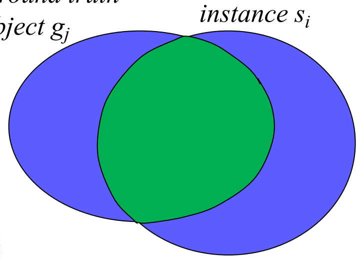
- Similar to the object-level F-score calculation
- Calculate IoU between any overlapping segmented instance and ground truth object

  $IoU\ (g_j,\ s_i)$ = ***Area of overlap*** / ***Area of union***

  - Segmented instance s_i is considered as TP if $IoU\ (g_j,\ s_i) > threshold$
  - You have already had the number of the segmented instances (TP + FP) and the number of the ground truth objects (TP + FN)
  - Calculate precision, recall, and F-score on these values → this is *F-score(threshold)*
  - Common to calculate IoUs for different thresholds usually from 0.5 to 0.95 and report them individually as well as the average of all F-scores as the final metric

25

Object-level Dice index



Let $S = \{s_i\}$ be a set of segmented instances

$G = \{g_j\}$ be a set of ground truth objects

$\gamma(s_i) \in G$ be the ground truth object that s_i maximally overlaps

$\sigma(g_j) \in S$ be the segmented instance that g_j maximally overlaps

$$\omega(s_i) = |s_i| \ / \sum_{s_m \in S} s_m$$

$$\omega(g_j) = |g_j| \ / \sum_{g_m \in G} g_m$$

$$DI(x,y) = 2 \cdot |x \cap y| / (|x| + |y|)$$

$$Dice(S,G) = \frac{1}{2}\left(\begin{array}{c} \sum_{s_i \in S} \omega(s_i) \cdot DI(s_i, \gamma(s_i)) \\ + \\ \sum_{g_j \in G} \omega(g_j) \cdot DI(g_j, \sigma(g_j)) \end{array}\right)$$

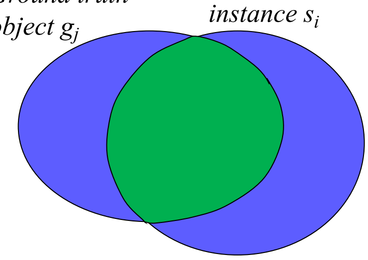
26

Object-level Hausdorff distance

$$HD(x,y) = \max\{\sup_{p_x \in x} \inf_{p_y \in y} ||p_x - p_y||, \sup_{p_y \in y} \inf_{p_x \in x} ||p_x - p_y||\}$$

$\sup_{p_x \in x} \inf_{p_y \in y} ||p_x - p_y||$ gives the maximum of the minimum distances calculated from every pixel p_x of object x to any pixel p_y of object y

$$Hausdorff(S,G) = \frac{1}{2}\left(\begin{array}{c} \sum_{s_i \in S} \omega(s_i) \cdot HD(s_i, \gamma(s_i)) \\ + \\ \sum_{g_j \in G} \omega(g_j) \cdot HD(g_j, \sigma(g_j)) \end{array} \right)$$



If there is no overlap for a segmented instance, use the minimum Hausdorff distance from this instance to any ground truth object

27

Thank you!

Next time:

Texture analysis

28