KOÇ ÜNİVERSİTESİ
COLLEGE OF ENGINEERING

COMP 448/548
Medical Image Analysis

Spring 23
Homework #1

Yiğit Can Ateş 75617 - Melis Oktayoğlu 64388

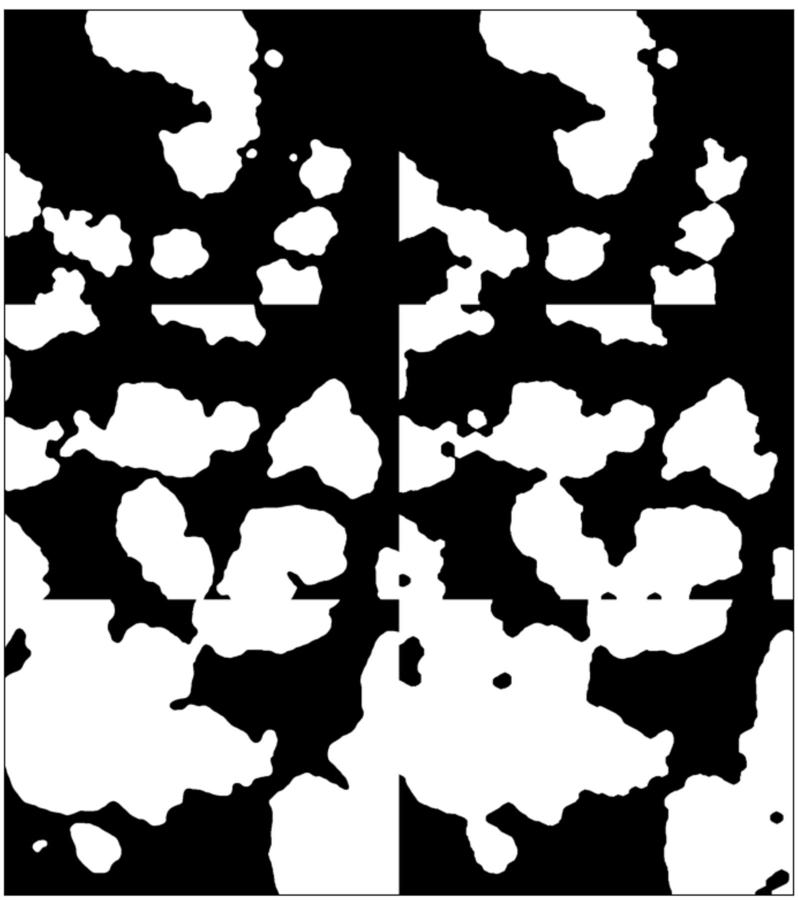
Part 1: Obtaining Foreground Images

Pseudocode:

1. Otsu threshold
2. Invert image to get foreground to be white
3. Close with 5x5 kernel 8 times
4. Close with 3x3 kernel 1 times
5. Close with 7x7 kernel 5 times
6. Opening with 5x5 kernel 6 times

Discussion:

We first applied Otsu thresholding to reduce the background noise and help differentiate the foreground cells. Then we inverted the image to have the cells in white and background in black. Then we first applied closing since cells are separated both between one another and also from their inside and edges. Then we applied opening to reduce the noisy closings from the background and have a clearer separation between the cell colonies.



```
Performance for img no 1
Precision: 0.92
Recall: 0.96
F-score: 0.94

Performance for img no 2
Precision: 0.95
Recall: 0.97
F-score: 0.96

Performance for img no 3
Precision: 0.98
Recall: 0.97
F-score: 0.97
```

Figure 1. Groundtruth (on the left) vs. our results (middle) for 3 images. On the right, their performance results.

Part 2: Find Cell Locations

Pseudocode:

1. Enhance contrast
2. Gaussian Blurring
3. Thresholding (pixel>180 →255, pixel<180 →0)
4. Close with 5x5 kernel 1 times
5. Calculate distance transform using inverted
6. Normalize
7. Find local maximum peaks
8. Filter out non-foreground pixels

Discussion:

We first enhanced the contrast and applied gaussian blurring to have a smoother image. Then we applied thresholding to capture boundaries in the image. We decided on the threshold value by trial and error. Then we applied closing to fill the holes and close the small gaps. We calculated the distance transform of the inverted image. After normalization, we calculated local maximum values which should correspond to cell centers. Finally, we filtered out any peak outside of the foreground mask.

```
Performance for img no 1
Precision: 0.89
Recall: 0.86
F-score: 0.87

Performance for img no 2
Precision: 0.91
Recall: 0.90
F-score: 0.90

Performance for img no 3
Precision: 0.88
Recall: 0.84
F-score: 0.86
```

Figure 2. Performance results for part 2.

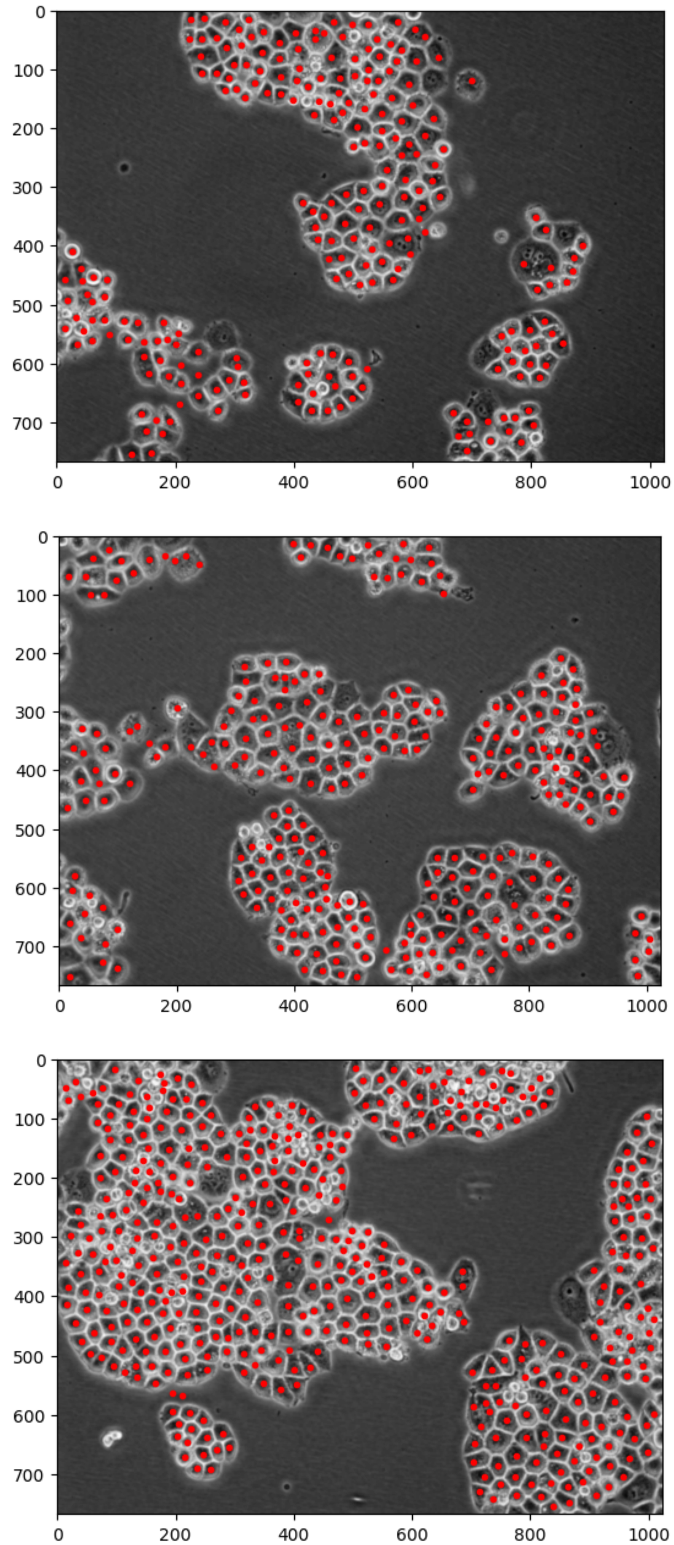


Figure 3. Estimated cell locations plotted on original cell images.

Part 3: Finding Cell Boundaries

Pseudocode:

1- Preprocess image to get the white boundaries of the cells clearer.
2- Do Region Growing:
  2.1 Add all the coordinates found in part 2 to seeds list.
  2.2 - While seed list is not empty:
    2.2.a - Pop the seed
    2.2.b - Add to the segmentation map its ID
    For all neighbors (x4):
      2.2.c - Calculate distance to neighbor
      2.2.d - Stop growing when distance (L2) exceeds 20 pxs
      2.2.e - Stop growing if neighbor exceeds image boundaries
      2.2.f - Skip if the neighbor is already visited
      2.2.g - If marking function below threshold, and neighbor not marked, grow to the region
        2.2.g.1 - Add this new point as a new seed

Discussion

We first preprocessed the image, first converting to HSV and getting the Value of the image which describes the brightness of the image. Then we selected the marking function to be above a certain pixel value threshold to decide on marking the current pixel. This way we can exploit the fact that the cell boundaries are deliberately white in the original image and use this as a stopping condition in our region growing algorithm. We also added minimum and maximum size constraints in the region growing algorithm to prevent noise caused early termination or late termination of the region growing algorithm.

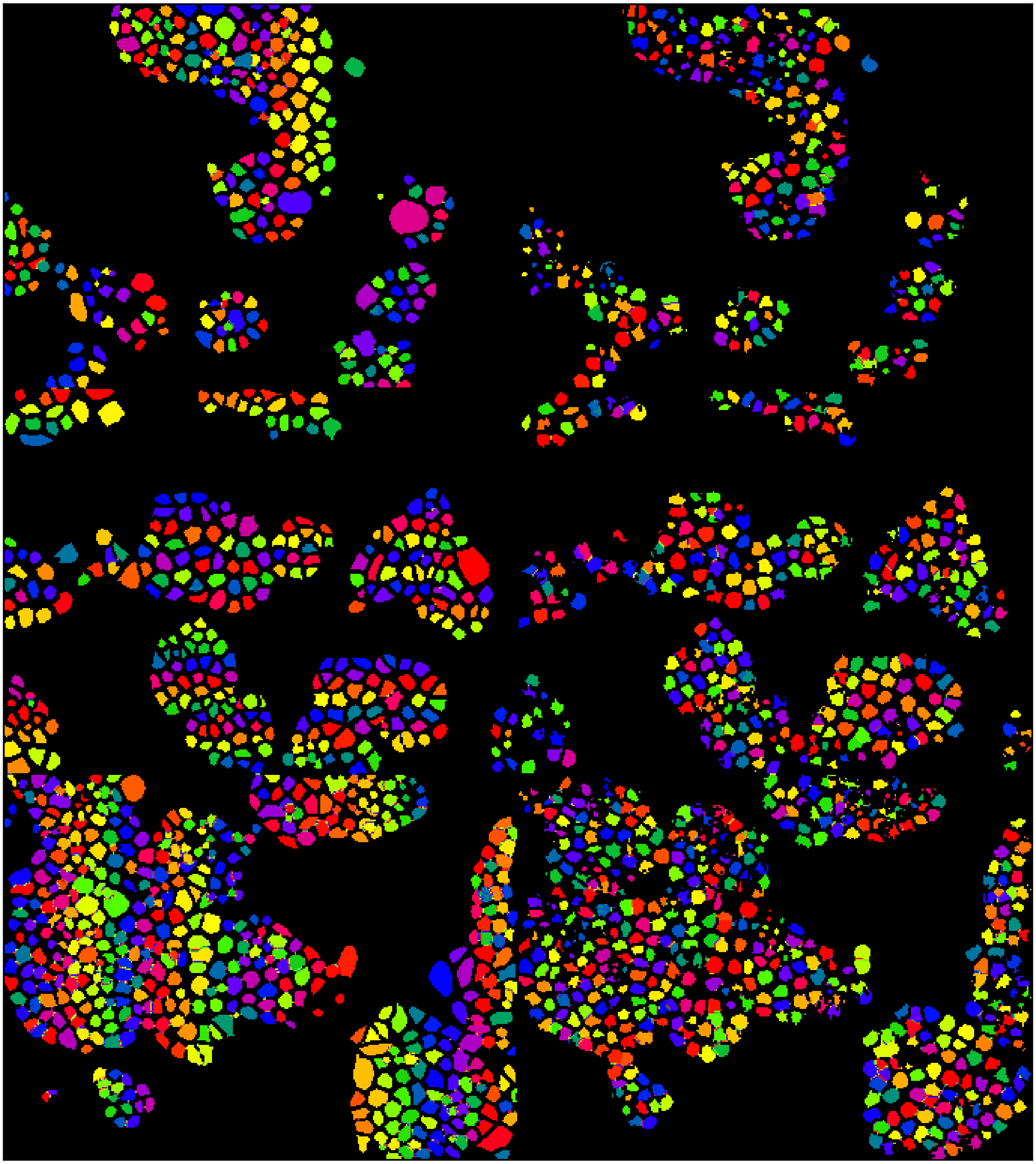


Figure 4. Ground truth cell boundaries (on the left) and our results (on the right) for 3 images.

```
IoU for threshold  = 0.50

Performance for img no 1
Precision: 0.73
Recall: 0.71
F-score: 0.72

Performance for img no 2
Precision: 0.82
Recall: 0.80
F-score: 0.81

Performance for img no 3
Precision: 0.77
Recall: 0.73
F-score: 0.75
```

```
IoU for threshold  = 0.75

Performance for img no 1
Precision: 0.32
Recall: 0.31
F-score: 0.31

Performance for img no 2
Precision: 0.36
Recall: 0.36
F-score: 0.36

Performance for img no 3
Precision: 0.34
Recall: 0.32
F-score: 0.33
```

```
IoU for threshold  = 0.90

Performance for img no 1
Precision: 0.02
Recall: 0.02
F-score: 0.02

Performance for img no 2
Precision: 0.01
Recall: 0.01
F-score: 0.01

Performance for img no 3
Precision: 0.01
Recall: 0.01
F-score: 0.01
```

Figure 5. IoU performance metrics for 3 thresholds for each image.

```
Dice index for img no 1
Dice: 0.15

Dice index for img no 2
Dice: 0.23

Dice index for img no 3
Dice: 0.35
```

Figure 6. Dice Index for each image.

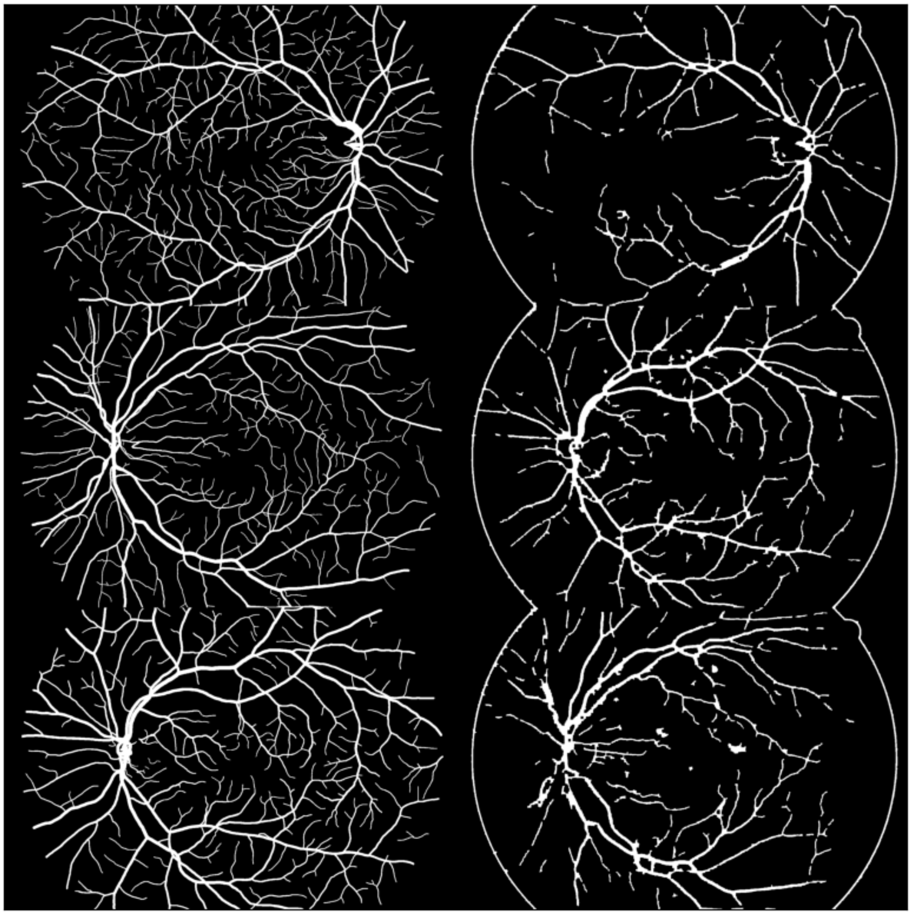
Part 4: Segmentation of blood vessels in fundus photography images

Pseudocode:

1. Preprocess the image:
   1.a - Turn images to gray scale
   1.b - Dilate 3x3 once
   1.c - Gaussian blur 3x3 twice
2. Apply log filter
3. Binary threshold (pixel>180 →255, pixel<180 →0)
4. Open with 2x2 kernel once
5. Close with 3x3 kernel 3 times
6. Open with 3x3 kernel once
7. Sum with output from step 2
8. Denoise
9. Threshold
10. Invert
11. Apply area opening with removing pixels with area < 5 , and connectivity = 0
12. Dilate with 2x2 kernel in 2 iterations
13. Apply area opening with removing pixels with area < 10 , and connectivity = 1
14. Close with 2x2 kernel 4 times
15. Open with 3x3 kernel once
16. Apply area opening with removing pixels with area < 120 , and connectivity = 2
17. Open with 3x3 kernel once
18. Close with 3x3 kernel 4 times

Discussion:

First we applied some preprocessing, by dilation we wanted to make the vessels more visible and blurred the image to prevent background noise. Then we applied LoG filter. Then we denoised and applied open and closings with different iterations and kernels we set by trying out different other sizes to balance between preventing the noise but meanwhile preserving the details of the vessels.



```
Performance for img no 1
Precision: 0.41
Recall: 0.29
F-score: 0.34

Performance for img no 2
Precision: 0.12
Recall: 0.10
F-score: 0.11

Performance for img no 3
Precision: 0.13
Recall: 0.10
F-score: 0.11
```

Figure 7. Groundtruth (on the left) vs. our results (middle) for 3 images. On the right, their performance results.