CENG 466 Fundamentals of Image Processing HW4

Andaç Berkay Seval
2235521

***Abstract*—This report is prepared for the fourth homework of CENG 466.**

I. Introduction

This report includes methodology and rationale behind the algorithms that are implemented in the homework, difficulties and errors encountered in the design, implementation, experimentation stages and their solutions, analysis and comments on the results and the requirements of the code.

II. Requirements

This homework is done with the help of some libraires. Therefore, there are 6 requirements for the code:

- scikit-image (skimage)
- opencv (cv2)
- numpy
- scikit-learn (sklearn)
- networkx
- matplotlib

The details and the purposes of these libraries will be explained in the next section.

III. Implementation

A. Object Counting

Object counting function is implemented with the help of skimage, opencv and numpy libraries. It takes 1 argument for the image number. For image 1: Firstly, image is binarized with a threshold value. This threshold value is found by threshold otsu method. However, it is adjusted a little bit for the image. Then, a square structuring element of size 5 is created and opening is applied to the image with that structure. After that, partial thinning is applied to the opened image to thin the image objects. Then, erosion is applied with that square structure to the last image. After that, opening is applied to the last image again with the square structure. Then, partial thinning is applied to the last image again. I created a new disk shaped structuring element of size 3 to apply erosion to the last image. Finally, to remove noise from the output image, some unnecessary connected components are deleted from the image based on their areas. The algorithm works fine.

For image 2: Firstly, image is binarized just like the same method with image 1. Then, a square structuring element of size 5 is created. After that, opening is applied to the binary image with that structuring element. Then, a new square shaped structuring element of size 20 is crated and closing applied to the last image with the new structure. Finally, to remove noise from the output image, some unnecessary connected components are deleted from the image based on their areas. The algorithm works fine.

For image 3: Firstly, image is binarized just like the same method with image 1. Then, a square shaped structuring element is created and opening is applied to the image with that structure. After that, to remove noise from the output image, some unnecessary connected components are deleted from the image based on their areas. Then, a new disk shaped structuring element of size 40 is created and erosion is applied to the image with this new structure. Finally, to remove noise from the output image, some unnecessary connected components are deleted from the image based on their areas again. The algorithm works fine.

B. Segmentation

Segmentation function is implemented with a name ”clustering.” since segmentation name is conflicting with skimage.segmentation library. Thus, clustering function is implemented with the help of skimage, sklearn, networkx and matplotlib libraries. It takes 2 argument as image number and segmentation function name as ”meanshift” and ””ncut. For image 1: In mean shift segmentation, firstly, in order to finish the algorithm in a logical amount of time, image is rescaled with a factor of 0.25 with bilinear interpolation (Since it has a size of 4032 x 3024). After that, image is flattened for the estimate bandwith method of the sklearn library. With that method, 3 different parameter sets are experimented on the image. In first parameter set, quantile is chosen to 0.1 and number of samples is chosen to 75. Then, mean shift segmentation is applied to the image with that parameter set. However, image is segmented in just 2 regions. In second parameter set, quantile is chosen to 0.1 and number of samples is chosen to 50. Then, mean shift segmentation is applied to the image with that parameter set. However, image is segmented in just 2 regions again. In third parameter set, quantile is chosen to 0.1 and number of samples is chosen to 25. Then, mean shift segmentation is applied to the image with that parameter set. Image is segmented in 14 regions. Thus, let’s look at n-cut segmentation. In n-cut segmentation, image is used as its original size since the algorithm finished in a normal time. Firstly, skimage.segmentation.slic method is used on the image to apply k-means clustering before normalized cut segmentation. Therefore, 3 different parameter

sets are experimented on the image with slic method. Do not forget that slic method returns image labels based on the parameter. In first parameter set, compactness is chosen to 10 and number of segments is chosen to 300. Then, n-cut segmentation is applied to the image with that parameter set. Image is segmented in 3 regions. In second parameter set, compactness is chosen to 10 and number of samples is chosen to 500. Image is segmented in 7 regions. In third parameter set, compactness is chosen to 30 and number of samples is chosen to 500. Image is segmented in 36 regions!

For image 2: In mean shift segmentation, firstly, in order to finish the algorithm in a logical amount of time, image is rescaled with a factor of 0.25 with bilinear interpolation (Since it has a size of 4032 x 3024). After that, image is flattened for the estimate bandwith method of the sklearn library. With that method, 3 different parameter sets are experimented on the image. In first parameter set, quantile is chosen to 0.1 and number of samples is chosen to 300. Image is segmented in 5 regions. In second parameter set, quantile is chosen to 0.2 and number of samples is chosen to 700. Image is segmented in 3 regions. In third parameter set, quantile is chosen to 0.3 and number of samples is chosen to 100. Image is segmented in 2 regions. Thus, let's look at n-cut segmentation. In n-cut segmentation, image is used as its original size since the algorithm finishes in a normal time. Firstly, skimage.segmentation.slic method is used on the image to apply k-means clustering before normalized cut segmentation. Therefore, 3 different parameter sets are experimented on the image with slic method. Do not forget that slic method returns image labels based on the parameter. In first parameter set, compactness is chosen to 1 and number of segments is chosen to 300. Then, n-cut segmentation is applied to the image with that parameter set. Image is segmented in 9 regions. In second parameter set, compactness is chosen to 10 and number of samples is chosen to 300. Image is segmented in 5 regions. In third parameter set, compactness is chosen to 10 and number of samples is chosen to 500. Image is segmented in 8 regions.

For image 3: In mean shift segmentation, original image is used since it has a size of 443 x 666. Thus, it is small compared to other images. Therefore, image is flattened for the estimate bandwith method of the sklearn library. With that method, 3 different parameter sets are experimented on the image. In first parameter set, quantile is chosen to 0.1 and number of samples is chosen to 300. Image is segmented in 7 regions. In second parameter set, quantile is chosen to 0.1 and number of samples is chosen to 700. Image is segmented in 8 regions. In third parameter set, quantile is chosen to 0.2 and number of samples is chosen to 100. Image is segmented in 5 regions. Thus, let's look at n-cut segmentation. Firstly, skimage.segmentation.slic method is used on the image to apply k-means clustering before normalized cut segmentation. Therefore, 3 different parameter sets are experimented on the image with slic method. Do not forget that slic method returns image labels based on the parameter. In first parameter set, compactness is chosen to 1 and number of segments is chosen to 300. Then, n-cut segmentation is applied to the image with that parameter set. Image is segmented in 8 regions. In second parameter set, compactness is chosen to 1 and number of samples is chosen to 200. Image is segmented in 6 regions. In third parameter set, compactness is chosen to 10 and number of samples is chosen to 500. Image is segmented in 16 regions.

For image 4: In mean shift segmentation, firstly, in order to finish the algorithm in a logical amount of time, image is rescaled with a factor of 0.25 with bilinear interpolation (Since it has a size of 3024 x 4032). After that, image is flattened for the estimate bandwith method of the sklearn library. With that method, 3 different parameter sets are experimented on the image. In first parameter set, quantile is chosen to 0.1 and number of samples is chosen to 300. Image is segmented in 5 regions. In second parameter set, quantile is chosen to 0.2 and number of samples is chosen to 700. Image is segmented in 3 regions. In third parameter set, quantile is chosen to 0.1 and number of samples is chosen to 700. Image is segmented in 5 regions. Thus, let's look at n-cut segmentation. In n-cut segmentation, image is used as its original size since the algorithm finishes in a normal time. Firstly, skimage.segmentation.slic method is used on the image to apply k-means clustering before normalized cut segmentation. Therefore, 3 different parameter sets are experimented on the image with slic method. Do not forget that slic method returns image labels based on the parameter. In first parameter set, compactness is chosen to 20 and number of segments is chosen to 300. Then, n-cut segmentation is applied to the image with that parameter set. Image is segmented in 9 regions. In second parameter set, compactness is chosen to 10 and number of samples is chosen to 300. Image is segmented in 13 regions. In third parameter set, compactness is chosen to 10 and number of samples is chosen to 500. Image is segmented in 13 regions.

For comparing mean shift segmentation and n-cut segmentation, they both give pretty good result for clustering images. Of course, segmented regions in the images depending on the parameter sets that given by us, however, for my choices, n-cut segmentation algorithm segments image into more regions generally. However, outputs of the n-cut segmentation in segmentation map is like pastel painting. Furthermore, outputs of the mean shift segmentation in segmentation map is a little bit more realistic than n-cut segmentation. Thus, the choice of using them is definitely application dependent. Thus, it is a design issue to prefer one over the other. Nevertheless, in the end, they can both segment the image in a good way.

Note: Firstly, tree relationship structure is implemented in the code. However, in every computer I experiment, it freezes the program and kill the process eventually. Therefore, unfortunately, I discarded tree relationship structure from the code.

The program gives output for original image, segmentation map, boundary overlay and region adjacency graph, but not for tree structure. Secondly, I could not solve how subfigure or subplots work correctly on matplotlib. I can save all the images and figures in the same file, but their sizes are inconsistent and region adjacency graph is appearing half. Thus, I decided to give separate outputs for each of them. For original image, corresponding output is "BX algorithm Y parameterset Z original.png". For segmentation map, corresponding output is "BX algorithm Y parameterset Z segmentationmap.png". For boundary overlay, corresponding output is "BX algorithm Y parameterset Z boundaryoverlay.png". For region adjacency graph, corresponding output is "BX algorithm Y parameterset Z rag.png".

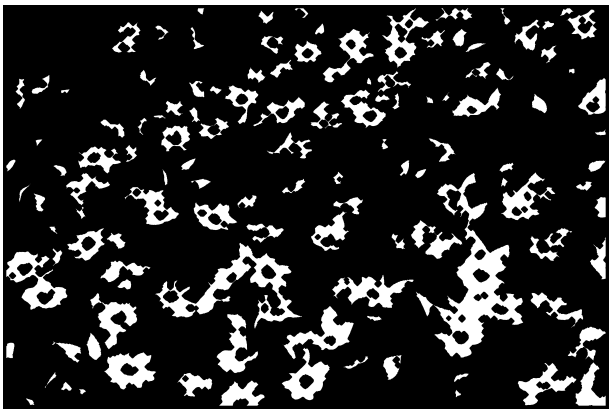

Fig. 1. A1 image after morphological operations are applied

Fig. 2. A2 image after morphological operations are applied

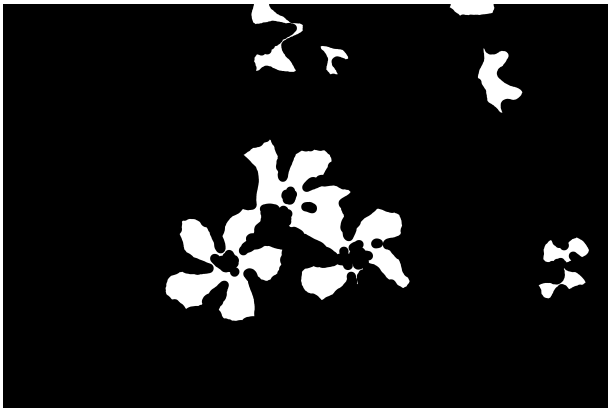

Fig. 3. A3 image after morphological operations are applied

REFERENCES

[1] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) https://doi.org/10.7717/peerj.453.

[2] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[3] Harris, Charles R. and Millman, K. Jarrod. "Array programming with NumPy". Nature. 2020. doi: 10.1038/s41586-020-2649-2

[4] Bradski, G. Dr. Dobb's Journal of Software Tools. "The OpenCV Library". 2000.

[5] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX", in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008

[6] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.