Report Topic 01 - Team 03:

The implementation and evaluation of Otsu thresholding

Data Analysis project, B.Sc.Molekulare Biotechnologie

Supervisors: PD Dr. Karl Rohr, Dr. Leonid Kostrykin, Kerem Celikay, Janis Meyer

Team: *Franka Begall, Kira Effenhauser, Selina Geiger, Anna-Lena Schulz*

Tutor: *Hannah Winter*

19.07.2023

Abstract

Image segmentation plays a role in many fields of research such as medicine and biology. One of the most effective ways to perform image segmentation is thresholding, in particular Otsu thresholding. This report mainly focuses on the optimization of Otsu thresholding using different preprocessing methods such as the mean, median and Gaussian filter, as well as histogram stretching. Additionally, Otsu thresholding was compared to more sophisticated versions like two-level-Otsu thresholding and local adaptive thresholding. All algorithms were applied to three different datasets displaying images of cell nuclei (N2DH-GOWT1, N2DL-HeLa, NIH3T3) and evaluated by using the Dice Score. Each dataset challenged and therefore tested the algorithm in a different way. As a result this report presents that Otsu thresholding is a reliable and fast method of image segmentation.

Table of contents

1. Introduction

Image segmentation is a research hotspot in computer vision. It allows the subdivision of an image into segments with higher information content (Yu *et al*, 2023). In biomedical application each segment displays a different object or structure of interest which is essential for diagnosis and treatment planning (Xu *et al.*, 2022). Microscope images of cell nuclei are often processed with image segmentation to count cells, differentiate them from certain bright spots or determine their size.

An often used and important technique for image segmentation is thresholding. Otsu´s method is very common as it gives reliable results while still being fast. The automatically generated threshold separates the image into two groups resulting in a binary image where the objects are clearly separated from the background (Bangare *et al.*, 2015). For even better results Otsu´s method can be extended to two-level-Otsu thresholding or local adaptive thresholding. These techniques adapt more effectively to the images because they do not rely on only one threshold.

Images, especially in biomedical context, are often distorted by noise caused by transmission and procurement. This also applies for the images of the three datasets used in this project. Challenges such as bright spots, low contrast and noise need to be eliminated to differentiate cells from the background. Different filters were used to master these challenges and improve the quality and reorganisation of the images. This is important in biomedical application because it increases the diagnostic value of the image (Bharati and Podder, 2020).

To evaluate the performance of the different preprocessing methods as well as the thresholding techniques we used the Dice Score. By comparing the segmented image to the ground truth image the Dice Score evaluates how effective the preprocessing was.

2. Description of the Datasets

For this project the data consists of three different, publicly available datasets, each showing cell nuclei. All of the datasets consist of the original images, as well as the ground truth images, which can be used to evaluate the implemented segmentation algorithm.

2.1. N2DH-GOWT1

The first set of data contains six images of N2DH-GOWT1 mouse embryonic stem cells, which were stained by using a green flourescent protein targeted against the transcription facor Oct4. These images were captured using timelapse confocal microscopy and have a size of 1024x1024 pixels. The low contrast to the background and image noise are the main challlenges this datasets presents.

2.2. N2DL-HeLa

Dataset 2 contains four images of human N2DL-HeLa cells of cervical cancer with a size of 1100x700 pixels. The staining of these nuclei was also executed by using a green fluorescent protein, but targeted against H2b. The images were captured by an Olympus Ix81 microscope used for live imaging of fluorescently labeled chromosomes. The main challenge this dataset presents is the low contrast.

2.3. NIH3T3

The last dataset consists of 18 images of mouse embryonic fibroblast cells which were visualized via Hoechst staining and captured with a fluorescence microscope. The size of these images is 1344x1024 pixels. A challenge this dataset embodies is the varying brigntness of the cell nuclei, as well as reflections, appearing as bright spots in the images.

3. Methods

To achieve an optimal image for applying the otsu thresholding algorithm we tested different preprocessing methods and compared how they performed when applied to each datatset by using the Dice Score as an evaluation method.

3.1. Preprocessing

3.1.1. Mean filter

The mean filter is used to blur an image in order to remove noise which is achieved by reducing the intensity variation between a pixel and its neighbors. At first the mean of the pixel values within a n x n kernel is determined and then the pixel intensity of the center element is replaced by the mean. However, this filter is strongly affected by outliers (pixels with a very unrepresentative value). A single outlier can significantly affect the mean value of all the pixels in its neighborhood.

3.1.2. Median filter

Compared to the mean filter, the median filters evades the problem of outliers. It ranks the intensities of all pixels within a predefined n x n neighborhood for each pixel. The median of this series then replaces the intesity value of the central pixel. It is especially useful when trying to reduce impulsive noise (e.g. salt and pepper noise).

3.1.3. Gaussian filter

The Gaussian filter is used to blur images and remove noise and detail. The Gaussian kernel has the shape of the two-dimensional zero-mean Gaussian Function (bell-shape), which means pixels in the center affect the new value of the central pixel more, than pixels further out.

$$G_\sigma(x,y) = \frac{1}{\sigma^2 2\pi} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

The intensity of the blurring effect depends on the standard deviation σ, a higher σ will lead to a stronger blurring effect.

3.1.4. Histogram stretching

Histogram stretching describes a linear scaling method to enlarge the contrast of an image. To perform histogram stretching, the intesity values of the pixels are stretched out, ranging from a desired minimum to a desired maximum (e.g. 0 to 255), this defines the lower and the upper limit *a* and *b*. Values below *a* are set to *a* and values above *b* are set to *b*. *c* and *d* represent the current minimum and maximum intensity values in the image. P_{in} stands for the pixel intesity value before the stretching is performed.

$$P_{out} = (P_{in} - c)\frac{(b-a)}{(d-c)} + a$$

3.2. Thresholding

Thresholding is used for the segmentation of images. Starting from a grayscale image it creates a binary image by setting all pixels whose intenisty values are above a certain threshold to a foreground value and all those whose values are below the threshold to a background value.

3.2.1. Otsu thresholding

Defining a threshold can be very challenging if the image has no perfect bimodal distribution of pixel intensities. A bimodal distribution allows to select your threshold easily by choosing the lowest point between the two peaks of your histogram for the threshold. Otsu´s method selects a threshold automatically by either minimizing the within class variance or maximizing the between class variance of the pixel intensities of your grayscale image no matter how many peaks your histogram has. The pixels are assigned new intensities according to the threshold: pixels with higher intensities than the threshold or the same intensity turn white, pixels with a lower intensity turn black. By performing this method your objects will be separated from the background of your image. The algorithm used in this project computes the within-class variances by iterating over all possible thresholds. The within-class variance σ_w is calculated from the probability of class occurrence ω and the class variance σ_i (Otsu, 1979).

$$\sigma_w^2 = \omega_1 * \sigma_1^2 + \omega_2 * \sigma_2^2$$

To achieve a binary and segmented image the pixels $g_T(x, y)$ are assigned new intensities depending on whether the pixels had a higher, lower, or the same intensity as the threshold T. Pixels with a higher or the same intensity are set to 255 respectively 65535 for tif-files, which is equivalent to the objects. Pixels with lower intensities are set to 0, which is equivalent to the background.

$$g_T(x, y) = \begin{cases} 0 \ \text{ if } g(x, y) \leq T \\ 255 \ / \ 65535 \ \text{ if } g(x, y) > T \end{cases}$$

3.2.2. Two-level-Otsu thresholding

Otsu thresholding reaches its limits as soon as there is certain bright spots, reflections or illumination in your image because these pixels tend to get assigned incorrectly. Fortunately, Otsu thresholding can be easily extended to two-level thresholding. By adding another threshold, you can solve that problem and separate bright spots from your objects. You get three different classes of pixels: bright spots, objects, and background. Just like the global Otsu thresholding the threshold is selected by minimizing the within-class variance.

$$\sigma_w^2 = \omega_1 * \sigma_1^2 + \omega_2 * \sigma_2^2 + \omega_3 * \sigma_3^2$$

The two-level Ostu thresholding performed in this task works in two different ways: The first algorithm sets the intensities of the pixels whose intensity is higher than or equal to the first threshold and intensities that are between the two thresholds to 255 or 65535 for tif files. The remaining pixels with intensities less than or equal to the second threshold are set to 0. The second alogithm works similar but in this case the intensities of the pixels with a higher intensity than the first threshold are set to 0 intstead of 255 or 65535.

3.2.3. Local adaptive thresholding

While Otsu thresholding and two-level-Otsu provide fast results they reach their limitations if the image contains different lighting conditions throughout the image resulting from a strong illumination gradient or shadows. If a global threshold is used for such an image, this can lead to a loss of information and detail. Local adaptive thresholding is a more sophisticated version of tresholding which evades this problem by setting the threshold dynamically throughout the image. The algorithm iterates over every pixel and assigns each pixel to either a foreground value or a background value. The pixels are compared to the values of other pixels within a n x n predefined neighborhood. This means the threshold is set individually for each pixel depending on the values of its neighborhood.

3.3. Evaluation

To evaluate the perfomance of the implemented algorithm the segmented images were compared to the provided ground truth images.

3.3.1. Dice Score

The Dice Score (DSC) is a measure of similarity between two sets of data. In this case it can be used to evaluate the similarity between the predicted segmentation mask and the ground truth segmentation mask. It ranges from 0 to 1, 0 indicting the thresholding algorithm assigned all pixels to the wrong values and 1 indicating a perfect segmentation.

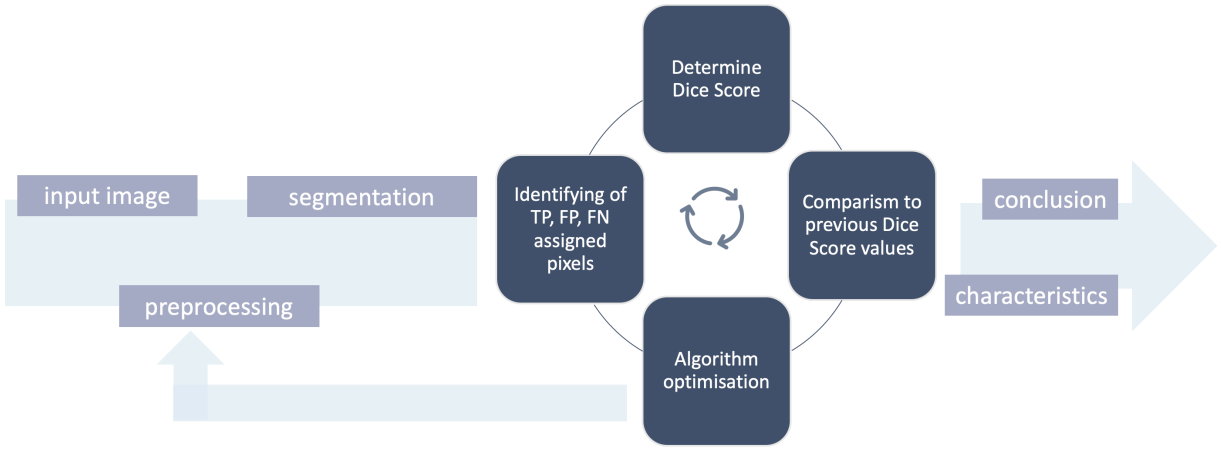
$$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

To calculate the DSC the overlay between the prediction and the ground truth is identified (*TP*), as well as pixels which were falsely assigned as negative (*FN*) and pixels falsely set as positive (*FP*).

Figure 1 illustrates the general steps of image segmentation. The DSC is applied to the aglgorithm after the segmentation is performed. If it reaches sufficient values, the segmentation algorithm works well and further steps, like analyzing the characteristics of the segmented image, can be executed. If the DSC values are not sufficient, the segmentation algorithm needs further optimisation.

```
In [ ]: # Figure 1: Visualisation of the general process of image segmentation to
        from IPython.display import Image
        Image('../Bilder Report/DiceScoreBild.png',  width=700, height=400)
```

Out[ ]:



4. Results

The following boxplots visualize our results regarding the comparison of the efficiency of different preprocessing methods, the combination of these preprocessing methods and also different variations of thresholding like Otsu thresholding, two-level-Otsu thresholding and local adaptive thresholding by using the DSC as the evaluation method.

4.1. Otsu thresholding combined with different preprocessing methods

This part of the results compares different preprocessing methods and the combination of different filters with histogram stretching. In addition, each filter was tested and compared in different variations regarding its size or its standard deviation.

N2DH-GOWT1

To evaluate the most effective preprocessing method for the N2DH-GOWT1 dataset different preprocessing methods and Otsu thresholding were applied to the images. The distribution of the such achieved DSC values is illustrated in figure 2. The mean DSC value of the images without any image processing is 0.591961. The application of only Otsu and the application of Otsu after applying different preprocessing methods all led to DSC values within a small range from 0.908740 (Otsu + Gauss, σ = 3) to 0.910361 (Otsu + Histo).

Figure 3 visualizes the distribution of the DSC values of the combined application of different filters together with histogram stretching and Otsu thresholding. As mentioned above, the mean DSC value of the images without any image processing is 0.591961. The mean DSC values of only Otsu thresholding and Otsu together with a filter and histogram stretching are all located in a small range, starting from 0.909265 (Histo + Median filter, 3x3) and ranging to 0.910804 (Histo + Gauss, σ = 1) as the highest obtained value.

In [ ]:

```
# Figure 2+3: Visualisation of the DSC values of different preprocessing
from IPython.display import Image
Image('../Plots/BilderPlots/N2DH-GOWT1/plots_N2DH-GOWT1.png',  width=900,
```

Out[ ]:

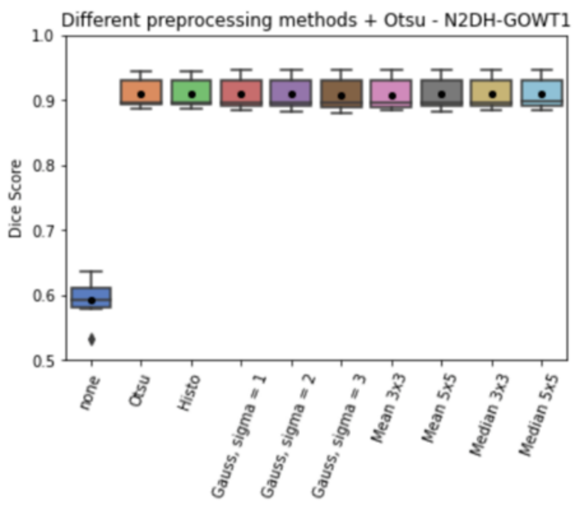


Figure 2

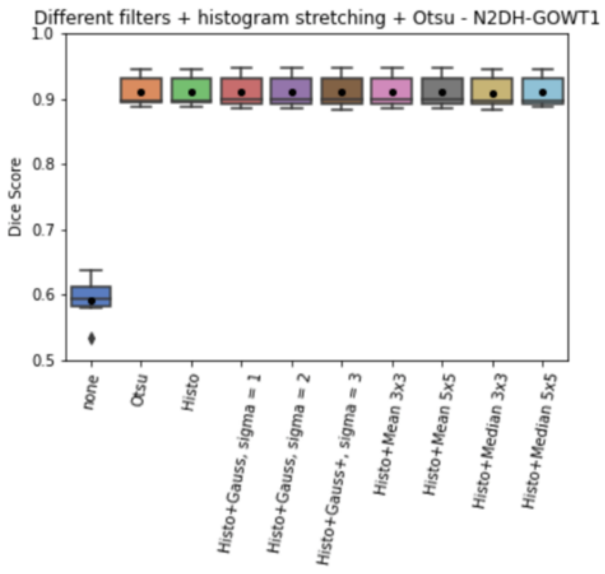


Figure 3

Generally, all of the tested methods led to high DSC values with a mean above 0.9 which is notably higher than the DSC of the unsegmented images.

N2DL-HeLa

Figure 4 was obtained through the application of different preprocessing methods and Otsu thresholding and visualizes the distribution of the DSC values. The mean DSC value of the images without any image processing is 0.153967. Histogram stretching and the application of a Median filter with a kernel size of 5x5 protrude with very high values. Histogram stretching reaches a value of 0.995984 and the Median filter 5x5 led to the highest possible DSC value of 1. All of the other DSC values are within a small range between 0.933156 (Gauss, $\sigma = 3$) and 0.935381 (Gauss, $\sigma = 1$).

The values shown in figure 5 represent the DSCs after implementing different filters combined with histogram stretching and Otsu thresholding. The values in this plot are similar to the DSC values in figure 4 except that no value reaches a perfect DSC. As already mentioned, the mean DSC value of the images without any image processing is 0.153967. Again histogram stretching with the value 0.995984 and histogram stretching in combination with the Median filter 5x5 with a value of 0.999842 lead to the best DSC values in this data set. The small range containing the remaining values reaches from 0.932934 (Histo + Gauss, $\sigma = 3$) to 0.936363 (only Otsu)

In [ ]:

```
# Figure 4+5: Visualisation of the DSC values of different preprocessing
from IPython.display import Image
Image('../Plots/BilderPlots/N2DL-HeLa/plots_N2DL-HeLa.png',  width=900, h
Out[ ]:
```

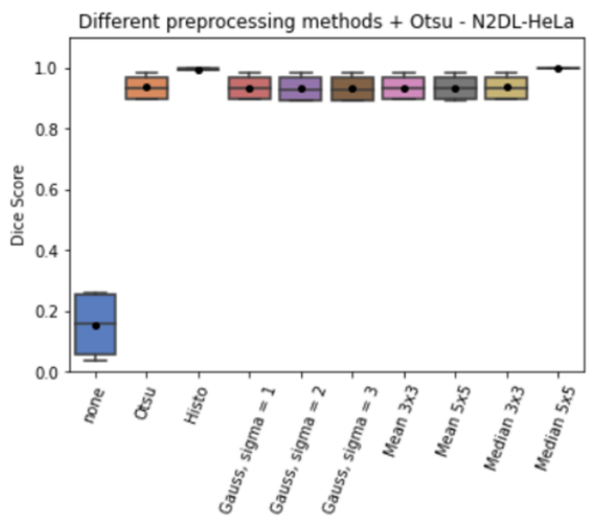


Figure 4

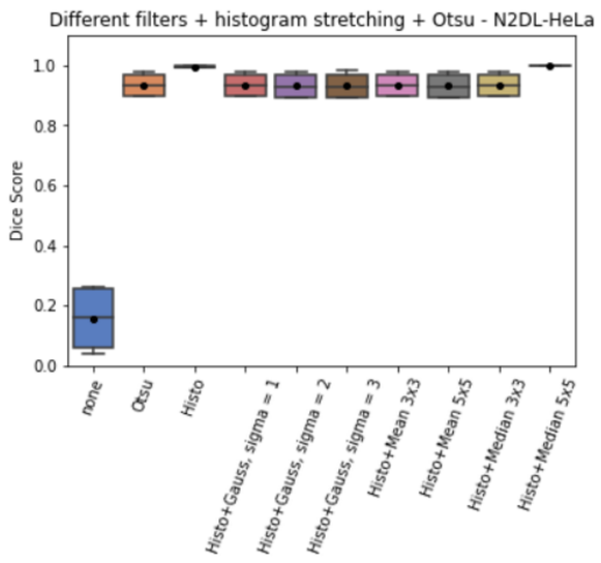


Figure 5

All processed images regardless of the method used have DSC values above 0.9 which is a significant difference as these values are way higher than the DSC mean value of the unprocessed images.

NIH3T3

The application of different preprocessing methods and Otsu thresholding to the NIH3T3 dataset led to the DSC values visualized in figure 6. The mean DSC value of the images without any image processing is 0.198225. The application of just the thresholding algorithm and the application of Otsu thresholding after applying different preprocessing methods all led to DSC values within a small range from 0.702818 (Otsu + Gauss σ = 3) to 0.711658 (Otsu + Histo).

Figure 7 visualizes the distribution of the DSC values of the combined application of different filters together with histogram stretching and Otsu thresholding. The lowest mean DSC was achieved through the combination of Histogram stretching, a Gauss filter with σ = 3 and Otsu thresholding with a value of 0.623451. The combined application of Histogram stretching, a Gauss filter with σ = 2 and Otsu thresholding also led to a worse DSC (0.680817) when compared to the other obtained values which all range from 0.705433 (only Otsu) to 0.713342 (Histo + Median filter, 3x3).

```
In [ ]: # Figure 6+7: Visualisation of the DSC values of different preprocessing
        from IPython.display import Image
        Image('../Plots/BilderPlots/NIH3T3/plots_NIH3T3.png',  width=900, height=
```

Out[ ]:

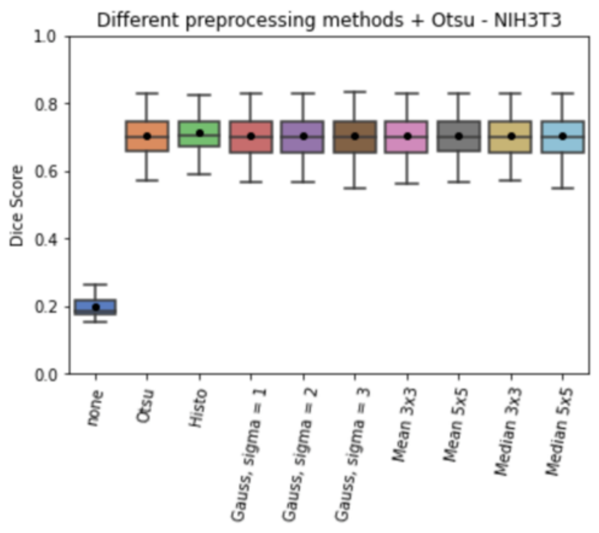


Figure 6

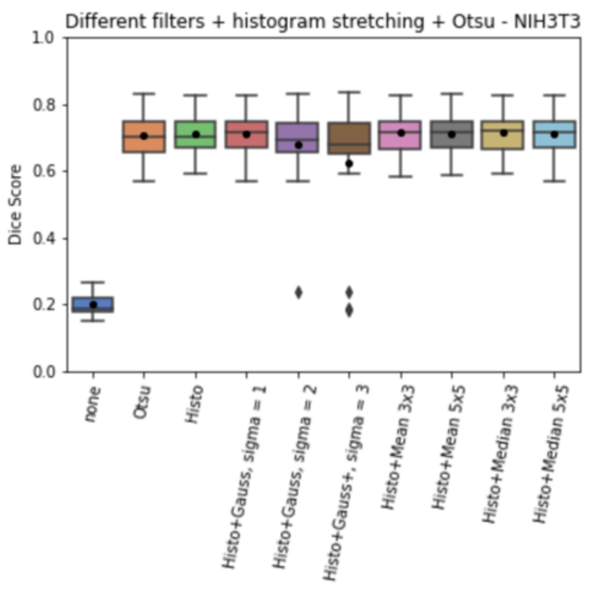


Figure 7

Overall the best DSC was achieved via Histogram stretching and a Median filter with 3x3 kernel, together with Otsu. However, the DSC values of only one preprocessing method + Otsu and a filter, histogramm stretching and Otsu are all very similiar and the combined application of a filter and histogram stretching did not lead to notably better DSC values.

4.2. Comparison of Otsu thresholding, two-level-Otsu thresholding and local adaptive thresholding

To appraise which of the different thresholding techniques produces the best results, Otsu thresholding, two-level-Otsu thresholding and local adaptive thresholding were applied to all three sets of data. For evaluation the DSC was used to compare the different techniques.

N2DH-GOWT1

Figure 8 visualizes the DSC values obtained by applying the three different thresholding algorithms to the N2DH-GOWT1 dataset. Local adaptive thresholding led to the worst results with a mean DSC of 0.802870. Two-level-otsu thresholding led to a mean DSC of 0.896784. The highest mean DSC (0.91036) was achieved via Otsu thresholding.

N2DL-HeLa

Figure 9 represents the DSC values of the mentioned thresholding algorithms of the N2DL-HeLa cells. Two-level Otsu reached the lowest DSC value of 0.929118. Otsu thresholding with a DSC value of 0.936363 is similar to the value of local thresholding which is 0.912420. Otsu thresholding shows a wider spread of data.

NIH3T3

Figure 10 represents the different DCS values of the thresholding algorithms applied to the NIH3T3 set. The DSC values show the least differences in this dataset. Otsu thresholding again led to the best value (0.705433) followed by local thresholding (0.659381) and two-level-Otsu thresholding (0.645896)

```
In [ ]: # Figure 8+9+10: Visualisation of the results achieved through different
        from IPython.display import Image
        Image('../Plots/BilderPlots/thresholding_plots.png',  width=1350, height=
```

Out[ ]:

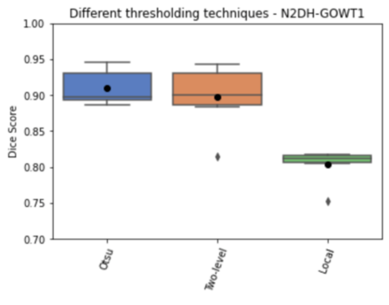


Figure 8

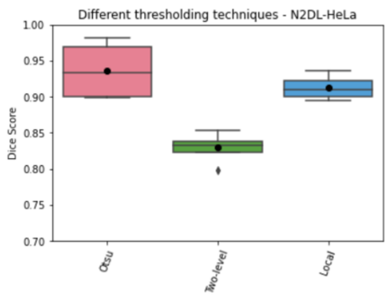


Figure 9

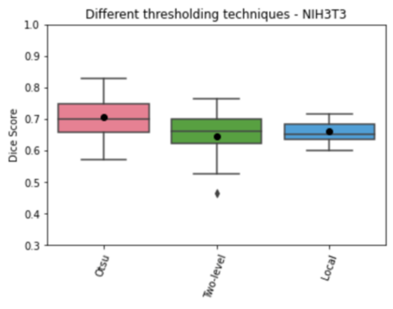


Figure 10

5. Discussion

The main focus of this project was to evaluate the ideal preprocessing method in combination with a self-implemented Otsu thresholding algorithm for each dataset and also the comparison of Otsu thresholding with more advanced thresholding algorithms like two-level-Otsu thresholding and local adaptive thresholding. All results were evaluated via the DSC.

Generally, the high DSC values indicate that Otsu thresholding worked well on all of the datasets. The application of preprocessing methods did not noticeably improve the segmentation results and did not differ much between the different preprocessing methods. This suggests that the Otsu thresholding algorithm itself already achieved results that did not need any more optimization and therefore preprocessing did not affect the DSC. Nevertheless, there still is a small trend, which suggests histogram stretching as the most valuable preprocessing method. Within the small range of DSC values, it provided the best results in all three datasets. The DSC values for the NIH3T3 dataset are overall lower compared to N2DH-GOWT1 and N2DL-HeLa. One of the main challenges this dataset presented were certain bright spots which made it hard to differentiate between cells and background. This could be the reason for the worse performance of Otsu thresholding. If other thresholding techniques were able to solve this problem will be discussed later on.

The comparison of Otsu thresholding with two-level-Otsu thresholding and local adaptive thresholding revealed that Otsu thresholding led to the best results in all three sets of data. Two-level-Otsu thresholding tends to assort cells overlapping with bright spots to the background since the values are above the second threshold. Therefore, certain cells are not recognized as objects leading to a loss of information and a low DSC value. In addition, some artifacts are misinterpreted as objects since the first threshold is lower than in Global Otsu thresholding, as the algorithm uses within class variance.
Local adaptive thresholding reaches its limits when the objects are not evenly spread throughout the image. In this case in areas without objects artifacts are also recognized as objects, similar to two-level-Otsu thresholding. This explains why local adaptive thresholding worked the best on the N2DL-HeLa dataset, since the cell nuclei are equally spread.
The data in figure 10 are relatively wide spread compared to the other two plots, because this dataset consists of the most images which explains the high variance.

6. Conclusion

The just presented results show that Otsu thresholding is the state of the art for a reason. Not only does it provide fast results, but it also best overcomes the challenges of the data sets. Since Otsu thresholding alone has worked so well, it is difficult to predict how well the individual preprocessing methods work. As our segmentation performed very well, we could continue to work with the images and apply a cell counting algorithm or an algorithm that measures the cell size.

7. Bibliography

Yu, Y.; Wang, C.; Fu, Q.; Kou, R.; Huang, F.; Yang, B.; Yang, T.; Gao, M. Techniques and Challenges of Image Segmentation: A Review. Electronics 2023, 12 (5).

Xu, Y.; He, X.; Xu, G.; Qi, G.; Yu, K.; Yin, L.; Yang, P.; Yin, Y.; Chen, H. A Medical Image Segmentation Method Based on Multi-Dimensional Statistical Features. Frontiers in Neuroscience 2022, 16.

Bangare, S. L.; Dubal, A.; Bangare, P. S.; Patil, S. T. Reviewing Otsu's Method For Image Thresholding. In International Journal of Applied Engineering Research; 2015; Vol. 10, pp 21777–21783.

Bharati, S.; Podder, P. Performance Analysis of Gaussian, Median, Mean and Weiner Filters on Biomedical Image De-Noising; 2020.

Otsu, N. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics 1979, 9 (1), 62–66.

8. Appendix

Python packages

The following Python packages were used:

- NumPy
- OS module
- Scikit-image
- Pandas
- Seaborn
- Matplotlib