university of groningen

faculty of science and engineering

Live Cell Image Segmentation for STED Microscopy

Niek Löke

university of groningen / faculty of science and engineering

University of Groningen

Live Cell Image Segmentation for STED Microscopy

Niek Löke (s3226832)

September 3, 2024

Master's Thesis

To fulfill the requirements for the degree of Master of Science
in Computer Science at University of Groningen
under the supervision of:
Prof. K. Bunte (Intelligent Systems, University of Groningen)
and
Dr. M.H.F. Wilkinson (Intelligent Systems, University of Groningen)

Contents

Page

Acknowledgements **5**

Abstract **6**

Symbols **7**

1 Introduction **8**

2 Methods **11**
2.1 MTObjects . . . 11
2.2 Data set . . . 13
2.3 Background Noise . . . 13
2.4 Blurring . . . 15
2.5 Image Reconstruction . . . 16
2.6 Height Plots, Isolines and Inner Segmentation . . . 18
2.7 Object Detection . . . 18

3 Experiments **21**
3.1 Experimental Setup . . . 21
3.2 Metrics . . . 22
3.3 Optimization . . . 23

4 Discussion & Results **25**
4.1 Image Segmentation for Low Dwell Time Images . . . 25
4.2 LR1 Image Reconstruction . . . 31
4.3 Sensitivity Analysis . . . 32
4.4 Mapping . . . 34
4.5 Secondary Experiments . . . 36

5 Conclusion & Future work **39**
5.1 Conclusion . . . 39
5.2 Future Work . . . 40

Bibliography **41**

Appendices **43**
A Additional Segmentation Maps 43
B Additional Image Reconstruction Comparisons 44

Acknowledgments

I would like to express my gratitude to my supervisors, Kerstin Bunte and Michael Wilkinson, for their invaluable support and guidance throughout the development of this thesis. Their insightful feedback and advice were instrumental in overcoming key challenges and improving the quality of this work. I also wish to extend my thanks to Frank Mol for kindly providing the STED image data sets that were essential to this project.

Abstract

Stimulated Emission Depletion (STED) microscopy is an advanced fluorescence microscopy technique that manages to break the diffraction limit of light, allowing for super-resolution imaging of cellular structures. This is achieved by the use of fluorophores, fluorescent molecules that can be selectively excited and quenched by two high-intensity lasers. However this intense light can detrimentally affect both the fluorophores' ability to emit light, and the viability of the observed cells. The most reliable way to reduce this damage is to limit the exposure to the STED lasers by reducing the dwell time. This thesis proposes a method to get the most vital information from low dwell time images using segmentation maps. To accomplish this a segmentation tool for Astronomy, MTObjects, is used as a starting point and adapted for the use on live-cell images. These adaptations include improved noise-detection, adaptive blurring, image reconstruction, and a novel approach to inner-segmentation. Consequently, a clear picture is created of how much information is retained for each lower dwell time selection, demonstrating that most vital information is preserved for all but the lowest dwell times.

Symbols

λ	move factor
f(x)	intensity of pixel x
P	a node in our Max Tree
$Panc$	closest significant ancestor
σ^2	variance
σ^2_{bg}	background variance
g	gain
α	significance level
d	distance
μ	mean
$P(x)$	probability distribution for x
k	number of events
ν	average number of occurrences
$P(k;\nu)$	probability of observing k events given mean rate ν
κ	edge sensitivity
γ	diffusion rate

Chapter 1

Introduction

Stimulated Emission Depletion (STED) microscopy is a relatively new and exciting technique that manages to surpass the diffraction limit of conventional light microscopy, enabling visualization of structures only visible at the nanoscale. First described by Stefan Hell and Jan Wichmann in 1994 [1], STED microscopy operates by utilizing fluorophores, fluorescent chemical compounds attached to molecules, to visualize, track, and quantify these molecules in specific regions of interest. These fluorophores are selectively deactivated in the excitation area, while only allowing those present in a very reduced center volume to emit light. This is achieved through a dual-beam setup: a Gaussian excitation beam and a doughnut-shaped depletion beam, which work together to confine the fluorescence emission to a subdiffraction region. This method effectively reduces the size of the fluorescent spot, thereby enhancing the resolution. [2]
To achieve high-resolution images necessitates the use of a high intensity STED beam in order to actively quench fluorescence via stimulated emission. However, this requirement introduces a significant trade-off: the high amount of illumination needed for high-resolution imaging can lead to photobleaching and phototoxicity [3]. Photobleaching occurs when the fluorophores undergo irreversible chemical reactions due to prolonged exposure to high-intensity light, leaving them non-fluorescent. The high-intensity light also causes phototoxicity, which involves damage to biological samples, which can cause irreparable harm, especially in live-cell imaging [4]. These negative effects can compromise the integrity and longevity of the imaging and therefore a solution has to be found that ensures the minimization of these effects while retaining as much information as possible.[5]

In (Vicidomini et al., 2018) [5] several STED-microscopy methods are discussed that would allow for clear imaging while reducing the effect of photobleaching and phototoxicity. Reducing unnecessary illumination should reduce these adverse effects and can be achieved by methods such as time-gated detection [6], where the fluorescence signal is collected after a delay, but also the development of adaptive optics and beam alignment systems can help in this regard. They also discuss newly designed fluorosphores that are increasingly stable due to enhanced photostability and reduced absorption at the STED beam's wavelength, therefore reducing the amount of photonbleaching [7]. Moreover, advanced strategies such as RESCue-STEd [8] and MINFIELD-STED [9] ensure more selective application of the STED beam, reducing overall exposure to the cell. Finally, using different wavelengths, most notably around infrared light, is

shown to reduce phototoxicity [10], making them more suitable for live-cell imaging. Although all these advancements enhance the practicality and effectiveness of STED microscopy, they do not completely eliminate the problems caused by photobleaching and phototoxicity.

In their research, Ebrahimi et al. (2023) [11] employ deep learning to significantly reduce the pixel dwell time in STED microscopy, addressing the challenges of photobleaching and phototoxicity. By integrating a two-step prediction architecture that combines a U-Net and a Residual Channel Attention Network (RCAN), they developed a method to restore high-quality images from low Signal-to-Noise (SNR) inputs obtained with shortened dwell times. This approach allows for efficient and robust restoration of noisy 2D and 3D STED images, even with significantly reduced exposure times. The reduction in dwell time minimizes the exposure of fluorophores to high-intensity light, thereby decreasing the likelihood of photobleaching and phototoxicity. Consequently, this technique enables long-term imaging of dynamic cellular processes, with minimal negative effects on the samples. However, the use of deep learning may introduce artifacts or elements that are not present in the original image, which is a problem found frequently in uses of deep learning for image reconstruction. These deviations mean that the restored images might not completely and accurately represent the actual cellular structures.

A great way to filter the most essential information out of an image is the use of segmentation maps. As shown in figure 1.1, these maps allocate specific pixels of an image to a particular source, aiding in the identification and separation of distinct objects. Segmentation maps are commonly used in Astronomy in order to detect structures such as galaxies, star clusters and nebulae. A paper [12] comparing four of these segmentation tools, SExtractor[13], ProFound [14], NoiseChisel[15], and MTObjects [16], shows the usefulness of these segmentation maps to find distinct objects. The paper showed that across all evaluation methods, MTObjects performed the best, citing "MTObjects produced the highest values for both F1-score and Area score".
Segmentation maps can be an invaluable tool for extracting critical information from cell imagery. By tracking the cell area through identifying regions of increased fluorescence activity, we can use this map to precisely delineate cell boundaries [17]. Furthermore, in STED microscopy, regions with varying fluorophore concentrations can be identified, where higher photon count areas indicate regions with a higher number of fluorophores, which are strongly correlated with high-density parts of the cell. This enables us to create accurate inner segmentation, revealing more active, dense and structurally significant regions of a cell. Therefore, segmentation maps can help us extract the most important information needed, both in terms of cell area as well as cellular components and inner detail.

In this paper we propose a way to get the most essential information out of low dwell time STED microscopy cell images using segmentation maps. To accomplish this we will use the MTObjects segmentation tool as a starting point. As MTObjects was originally created for the segmentation of astronomical objects, we propose several adaptations in order to optimize it for cell imaging such as noise adaptation, image reconstruction and improved inner-segmentation. Due to our use of low dwell time images this approach should significantly reduce the problem of phototoxicity and photobleaching in order to increase the lifetime of both the fluorophores and cells, allowing for live cell imaging. Meanwhile this would eliminate the risk of artifacts associated with the use of Deep Learning. We will present a detailed analysis of the MTObjects

tool, including all adaptations in Section 2. Following this, Section 3 will cover our experimental setup, metrics and optimization procedure. In Section 4 we will present and discuss our findings and results in detail. Finally, in Section 5, we will conclude our research and look at potential future works.

Figure 1.1: An example of a segmentation map, showing how a traffic image can be segmented into distinct objects. [18]

Chapter 2

Methods

In this section, we will present the methods utilized and developed for this project, beginning with our starting point, the MTObjects segmentation tool. MTObjects was chosen due to its great performance in detecting astronomical structures, both in terms of speed and accuracy [12]. The main dataset will be discussed before detailing our modifications in order to provide insight and context into our approach and the specific challenges we addressed. Our adaptations start with the background detection method, where we modified it to account for Poisson noise as this is much more accurate when dealing with low photon count images. Additionally we incorporated an additional blurring method, Perona-Malik (anisotropic) smoothing, to investigate if this can increase performance. To address challenges with low photon count images, we implemented image reconstruction techniques to further enhance segmentation accuracy. We also reconstructed the inner segmentation of found objects using a combination of height-plots and isolines to get a more detailed and accurate representation of our cells. Finally, we will describe a method of object detection for one of our secondary data sets. In the following section we will discuss each of these components in detail.

2.1 MTObjects

The MTObjects algorithm is a sophisticated tool designed for the detection of faint astronomical sources in images, utilizing a Max-Tree structure [19]. The Max-Tree structure organizes an image into connected components at varying intensity levels, where each node represents a component above a specific threshold. In the construction of the Max-Tree, each node can be associated by various attributes, such as size and intensity. These attributes are then saved for each node, resulting in a tree where the root node represents the entire image, while leaf nodes denote local peaks. An example how this tree structure is organized can be found in 2.1.

In order to find all significant nodes present in an image several steps have to be taken. After the creation of the Max-Tree structure, containing each connected component, we then start by estimating the background. A flat square region devoid of objects is found using a Gaussian significance test to make our estimation. We estimate the background noise to distinguish between true signals and noise, enabling more accurate detection and analysis of faint astronomical sources. Afterwards we start the process of finding so called significant nodes. First, two similar power attributes are defined, `power(P)`(2.1) and `powerAlt(P)`(2.2). In these formulas we calculate the power attribute by summing the squared difference between

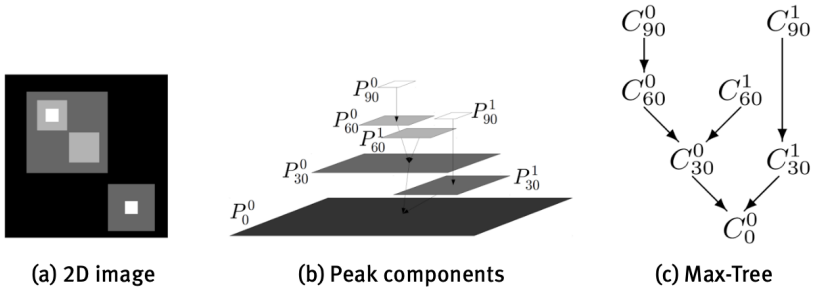


Figure 2.1: a) a grey-scale image with intensities ranging from 0 to 90, b) its corresponding peak components and c) the corresponding Max-Tree structure.[19].

the intensity of each pixel f(x) in a node P and either the intensity of the parent node (2.1) or the closest significant ancestor (2.2).

$$\text{power}(P) = \sum_{x \in P} (f(x) - f(\text{parent}(P)))^2 \tag{2.1}$$

$$\text{powerAlt}(P) = \sum_{x \in P} (f(x) - f(P_{anc}))^2 \tag{2.2}$$

Here the difference between a parent node and the closest significant ancestor can be summarized as follows: A parent node is always the immediate predecessor of a given node, while the closest significant ancestor may be several levels higher, representing the closest node deemed significant. As we process nodes from the bottom to the top of the Max-Tree (from the full image to the peaks), we can dynamically update and use the closest significant ancestor to determine the significance of higher nodes. These power attributes are then used as the core for a statistical test to finally determine if a node is significant or not. While the paper mentions four different proposed significance tests, the final statistical test is used in the MTObjects code and is defined as follows: first a Gaussian smoothing filter (2.3) is applied to reduce noise. Afterwards we want to find the local noise variance using Eq. (2.4), where σ^2_{bg} denotes the background variance and $g \cdot f(P_{anc})$ is the combination of the intensity of the closest significant ancestor, multiplied with some gain variable. Finally, we can determine if a node is significant using formula (2.5). In this step, we divide the power attribute `PowerAlt(P)` by the local noise variance and check whether it exceeds the threshold given by the `InversePowerAltCDF` function. This function takes as input a significance level α as well as parameters for area and distance d, and returns the threshold that must be surpassed to consider a node significant, based on these parameters.

$$H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.3}$$

$$\sigma^2 = \sigma^2_{\text{bg}} + g^{-1} \cdot f(P_{anc}) \tag{2.4}$$

$$\frac{\text{powerAlt}(P)}{\sigma^2} > \text{inversePowerAltCDF}(\alpha, \text{area}, d) \tag{2.5}$$

After the identification of significant nodes within the Max-Tree structure, the next step involves accurately finding and segmenting objects from each other. The process starts by an initial object marking: significant nodes without significant ancestors are assumed to be objects while those that do will be compared to their closest significant ancestor based on some significance test, smoothing and connectivity. Once our list of objects is complete our final step involves finding the correct area for each object. Here the variable move factor (λ) plays a crucial role in finding the true extend of structures, using Eq. (2.6)

$$h = f(P_{anc}) + \lambda\sqrt{\sigma_{\text{bg}}^2 + g^{-1} \cdot f(P_{anc})} \tag{2.6}$$

Here, h represents the threshold value to determine if an object is significant or not. An increase in this threshold means that the object marker will move further up the tree (to above the threshold). Move factor therefore influences how far the object marker will be placed along the tree, which directly influences the overall size of the object found. Increasing λ can reduce the inclusion of noise in the objects area, but might result in fainter objects being missed by the algorithm. This variable also has a strong correlation with the SNR, as we can keep move factor high when the ratio is high to make sure we don't include noise in our objects, while if it is low we can keep it low to make sure we find faint structures. After the relevant objects are found, a segmentation map can be constructed, distinguishing each object from background noise and nearby other structures.

2.2 Data set

For our initial testing and optimization we use a single data set consisting of artificial STED microscopy images. These images resemble real world scenarios and are generated using PySTED [20], where for each High-Resolution (HR) real-life STED image, 5 different low resolution [lr1 - lr5] images are generated. An example of a full set of these images can be found in figure 2.2, while the corresponding dwell times for these low resolution images can be found in table 2.3. By creating a varying set of exposure time images we can study how much information we can retain when using different dwell times, while also creating diverse scenarios, including high and low photon count images. Besides this main data set, two secondary data sets are defined: a data set consisting of high resolution STED images and another consisting of pairs of images taken by both STED and confocal microscopes.

2.3 Background Noise

In astronomical imaging, modeling noise accurately is extremely important for the detection of structures. Originally, MTObjects assumes a Gaussian noise approximation as sufficient to estimate background noise levels. Gaussian noise is characterized by a normal distribution with a mean μ and a standard deviation σ and it's probability density function is given by Eq. (2.7).

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.7}$$

This model assumes that noise is both symmetric and continuous and that the variance is constant. While this is a fair estimate when the photon count is high and the central limit theorem applies, it becomes inaccurate when dealing with low photon count scenarios.

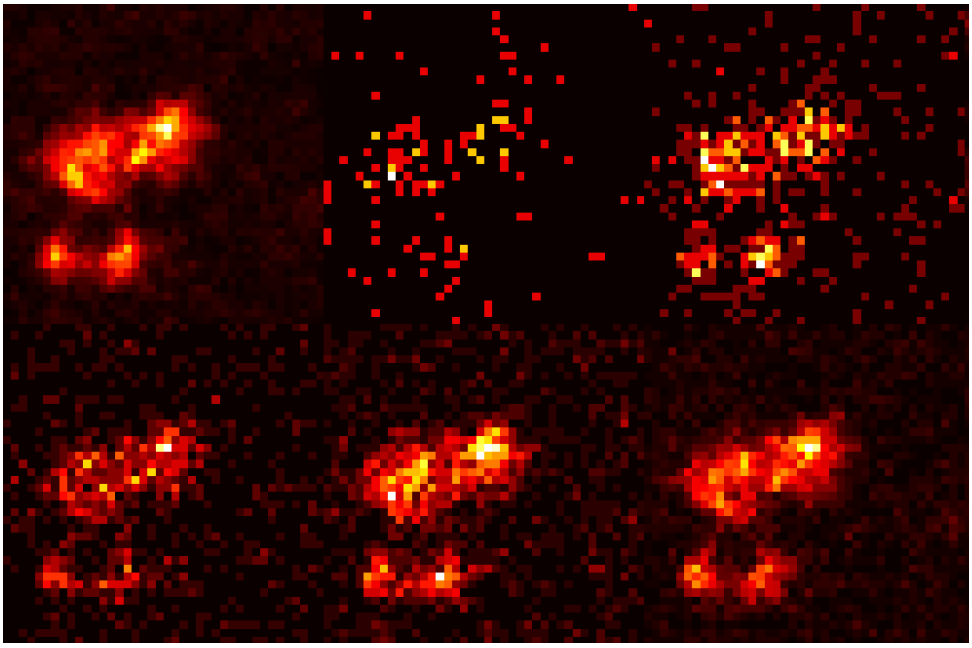


Figure 2.2: Our STED image data set, from top left to bottom right: the original High-Resolution (HR) STED microscopy image and the generated Low-Resolution images (LR1 - LR5)

Figure 2.3: Dwell times for different sets

Set	**Dwell Time**
hr	200 µs
lr1	1 µs
lr2	5 µs
lr3	10 µs
lr4	20 µs
lr5	50 µs

$$P(k;\nu) = \frac{\nu^k e^{-\nu}}{k!} \tag{2.8}$$

To contrast this, Poisson noise, which follows a Poisson distribution, is directly linked to the photon-counting process, given by Eq. (2.8). Here the function P gives the probability of k occurrences, given the mean rate ν. The Poisson distribution has a probability mass function (PMF) which uses discrete variables to construct its distribution, opposed to the Gaussian's probability density function (PDF), which uses continuous variables. This is much closer to the real-life low photon-count scenario where we are also dealing with a discrete number of photons. The distinction between these types of noise and distributions can be further visualized in figure 2.4, where we compare the Poisson PMF with the Gaussian PDF. Not only does this show the discrete nature of Poisson noise, it also illustrates a distinct skew to the left in the distribution when dealing with low mean values. Using Poisson noise in these scenarios will then more closely mimic real world scenarios. Due to these considerations, we adapted the MTObjects Gaussian noise estimation with one using Poisson noise. This adaptation will result

in better background noise estimation, especially in low photon count scenarios, and improve the reliability of source detection.

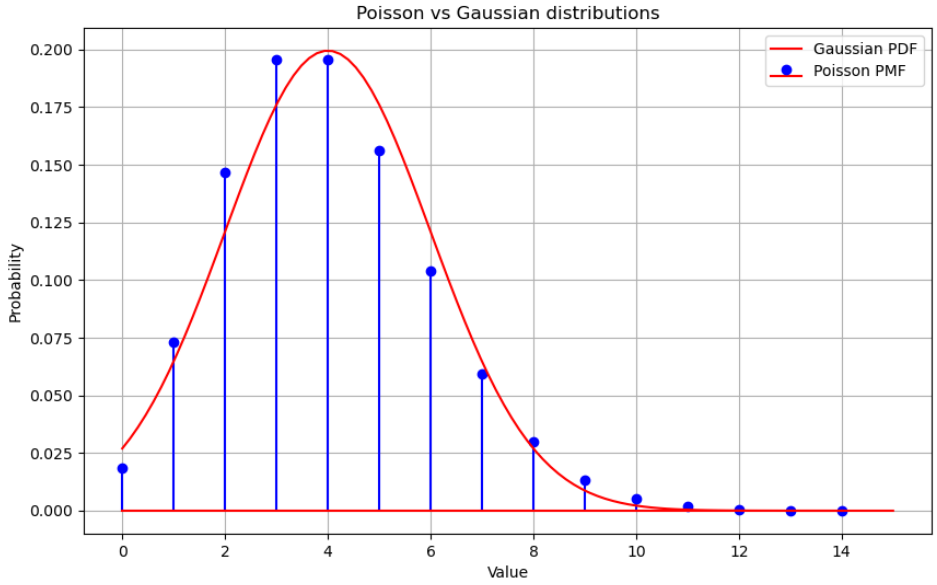


Figure 2.4: The probability density function (PDF) of a gaussian distribution and the probability mass function (PMF) of a poisson distribution for a mean of $\nu = 4$. It shows how for low mean values the distributions can differ significantly.

2.4 Blurring

As mentioned in the previous section, MTObjects uses a Gaussian blurring filter on its input images in order to reduce noise and smooth out the image. Smoothing the image helps by reducing the impact of noise while keeping larger structures intact, as illustrated by figure 2.5. The filter works by convolving the image with a Gaussian matrix (2.3), which spreads out the intensity of each pixel over a wider area. In our pipeline we adapt this by using a variable blur, where the amount of blurring depends on the overall pixel intensity found within the image (see also: Experiments - Optimization). As we are dealing with a large spectrum of image intensities, by making the amount of blurring adaptable we can make sure we strike the perfect balance between reducing noise and maintaining structures.

A further adaptation to the blurring structure is the use of an alternative blurring technology: Perona-Malik (anisotropic diffusion) blurring [21], defined by formula (2.4).

$$I_{t+1}(x,y) = I_t(x,y) + \gamma \cdot \nabla \cdot (c\,(\|\nabla I_t(x,y)\|) \cdot \nabla I_t(x,y))$$

In this formula, $I_t(x, y)$ represents the image intensity at position (x, y) at iteration t. The parameter γ controls the diffusion rate, influencing the extent of smoothing per iteration. The term $\|\nabla I_t(x, y)\|$ is the gradient magnitude, indicating edge strength at each pixel. The function $c(\|\nabla I_t(x, y)\|)$ is the diffusion coefficient, which reduces the amount of smoothing near strong edges, influenced by the κ parameter. This smoothing technique further enhances the idea of reducing noise while maintaining interesting structures. By encouraging smoothing within homogeneous regions while inhibiting it across edges, anisotropic diffusion aims to maintain sharp region boundaries which are essential for finding cell regions. This contrasts Gaussian blurring, where the amount of smoothing is uniform across the entirety of the image. By implementing both blurring methods, while also making both methods dynamic based on pixel intensity, we make sure that we explore all viable options to find the optimal segmentation result.

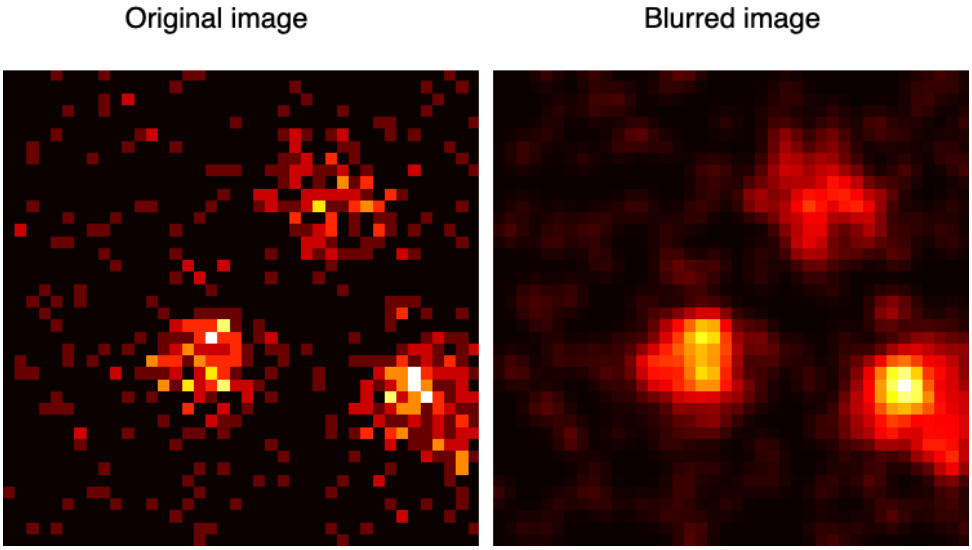


Figure 2.5: An image containing three cells showcasing how blurring the image can result in lower noise and clearer object areas.

2.5 Image Reconstruction

The method described in [22] introduces a technique for reconstructing images using data from randomly distributed sensors. Here the method first starts with an initial estimate of the image based on random sensor data. This estimate is then refined iteratively: during each step, the current estimate is adjusted by comparing the sensor data and updating it using a specific formula. This formula combines a band-limited operator, which focuses on the most important parts of the signal, and a sampling operator, which deals with the randomly scattered sensor data. Each iteration reduces the error, bringing the estimate closer to the true image and converging eventually to an estimate close to the true signal. A reconstructed image using this general idea can be found in figure 2.6.

The method described in the paper can be effectively applied to low dwell time images, which suffer from high noise levels and limited photon counts. Our implementation adapts the method described in this paper by focusing on selectively updating only the pixels that fall

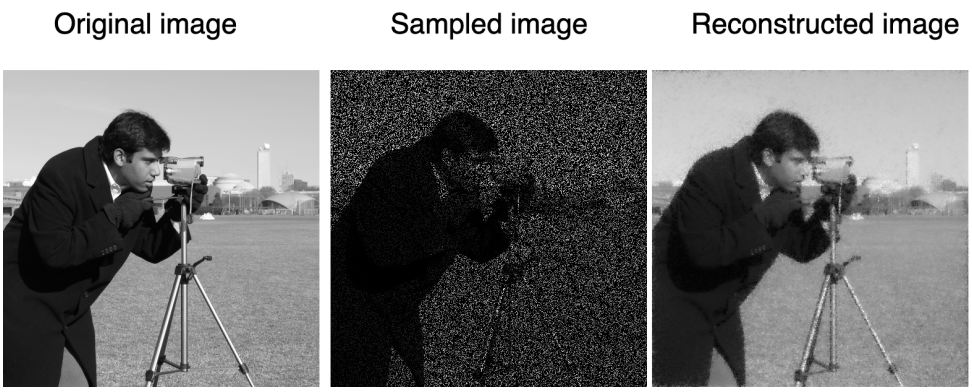


Figure 2.6: An example image that illustrates how the image reconstruction works. From the original image (left) we draw random samples (0.25 of total) to get the sampled image (middle). We then reconstruct the image using an iterative reconstruction technique.

below a background mean threshold. This will allow us to only update parts of the image that did not receive any photons while maintaining already created structures. By applying the algorithm this way we aim to close any gaps present as a consequence of the low dwell time, significantly enhancing the process of finding objects along with their true area in these low photon count situations. After applying our mask to identify below background mean pixels, we follow the same basic approach: iteratively refining the initial image estimate using a low pass filter. The filter's radius starts small and increases gradually following an exponential formula, allowing for finer reconstruction over time. Key steps include applying the low pass filter, calculating the correction by comparing the filtered and current estimates, and updating the image using this correction while ensuring only low-value pixels are modified. The application of the algorithm on a low dwell time cell image can be found in 2.7. By iterating this process, the method effectively reconstructs the image, making it particularly suitable for enhancing low dwell time images where data is sparse and unevenly distributed.

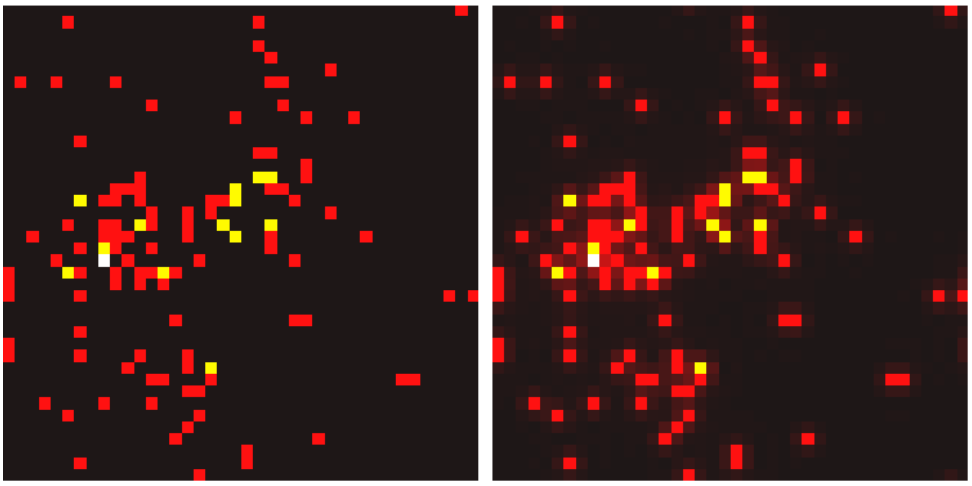


Figure 2.7: The iterative reconstruction technique applied to one of our low resolution (lr1) cell images. The original image (left) is reconstructed (right) by subtlety filling in some of the gaps, allowing our algorithm to more accurately find the objects area.

2.6 Height Plots, Isolines and Inner Segmentation

When analyzing a cell image, both the object's area and location are important factors, but additional information is also essential. Just as critical is how the inner parts of the cell are concentrated, especially when dealing with clumps of cells merged into one object, or during the process of cell reproduction. Inner segmentation is crucial as it reveals where the fluorophores are located within the cell, highlighting regions of high density that are essential for understanding cellular structures and functions. To get the most information from microscopy cell imaging, we need inner segmentation of objects found within the image. While MTObjects is a great tool for detecting distinct objects within an image, it falls short in accurately identifying and segmenting internal features. This is due to the inherent nature of the algorithm and its Max-Tree structure: by using thresholds to denote different objects, it solely focuses on high intensity differences while ignoring finer details. Furthermore, it will only show segmentation with intensity increases if it believes it belongs to a completely different object (for example, when dealing with overlapping galaxies). This is in contrast to our need for segmenting a single object into multiple part and find low and high-intensity regions. The ability to segment and identify these high-density areas is pivotal for mapping the distribution of fluorophores, which are an indication of various biological processes and molecular concentrations within the cell. Therefore, we need a separate method for inner segmentation to obtain the information required for detailed cell segmentation mapping.

To accomplish this, we will use a combination of height plots and isolines. Height plots are a great tool to show values or intensity at a point by depicting it as height or elevation. Using height plots in our cell images shows us the denser, more intense regions. By combining this with isolines we can create distinct regions where intensities are higher or lower than some threshold, effectively segmenting the inner parts of an object. To accomplish this we first need to find the intensity range within an image and based on this create bins which will define our isoline borders. By using bins we can also adjust the level of segmentation; if we want more segmentation we can increase the number of bins and vice versa. After drawing our isolines based on the found values we can find the areas enclosed by them to finally denote the higher intensity centers of these objects. By combining the area object segmentation of MTObjects and the inner segmentation provided by this method we end up with a very informative and detailed segmentation map. The general approach is visualized in figure 2.8 and the inner segmentation implementation is defined as pseudo code in Algorithm 1, figure 2.6, finalizing our general pipeline.

2.7 Object Detection

One of our secondary dataset consists of pairs of images, with one image taken by a STED microscope and the other by a confocal microscope, both capturing the same scene. To properly evaluate the segmentation maps created from these images, our goal is to determine how many objects identified in the higher-resolution STED segmentation maps are also present in the corresponding confocal images. This is accomplished by leveraging several advanced image processing techniques, such as object markers, the watershed algorithm [23] and intersection

Algorithm 1 Inner Segmentation

Input: Image I, Number of bins N
Create height plot from I
Get contours and contour levels from height plot based on N intensity bins
for *each contour level* L **do**
    for *each contour path* P *in* L **do**
        Find path of contour P
        Mark each point within path P
        Update each pixel within path P to the level of L
return segmented image

over union (IoU). Using these techniques we aim to assess the consistency and accuracy of object detection across these two microscopy technologies.

Our goal is to enhance object detection in our dataset, by applying the watershed algorithm, leveraging the high-resolution information from the STED images to guide the segmentation process in the confocal images. First, markers are placed at the centers of identified objects in the STED images, serving as starting points for the segmentation process. We then applied a distance transform to the inverted confocal images and used this map to guide the watershed process, starting from the STED-derived markers. This method effectively separates corresponding objects in the lower resolution confocal images by simulating flooding from these markers, ensuring accurate object boundaries. Consequently, the watershed algorithm enhances the alignment and matching of objects across our technologies, ensuring comparative analysis between STED and confocal microscopy data.

After segmentation, we matched objects by calculating the Intersection over Union (IoU) between objects in the STED and confocal images. This measurement shows how much the objects overlap, helping us find matching pairs based on some IoU threshold. The algorithm also identifies merged hits, where several STED objects are within one confocal object, which highlights the resolution differences between the techniques. Our results include a visualization of matched objects along with metrics concluding information about (merged-)matched objects. This approach helps us understand the effectiveness and accuracy of the segmentation maps created using these two microscopy methods.

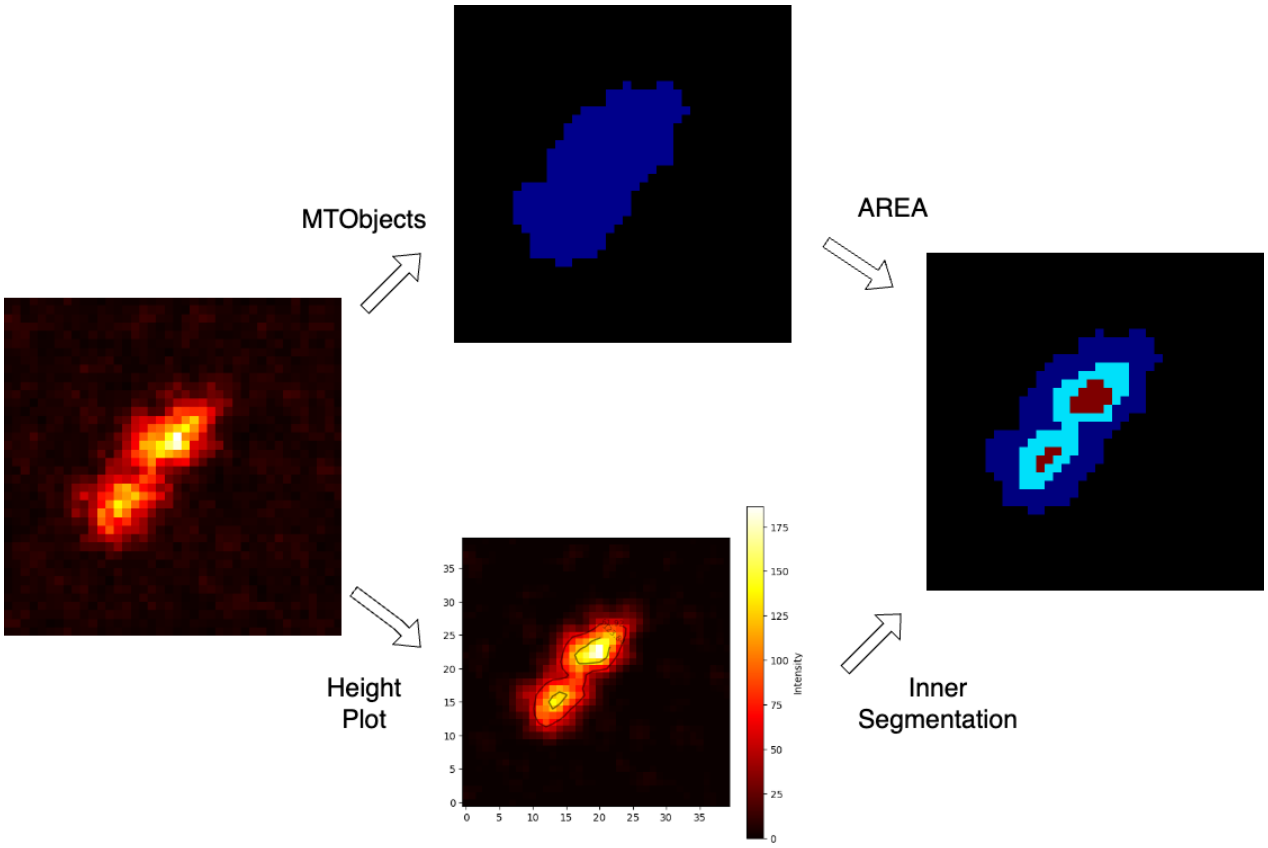


Figure 2.8: An overview of the complete pipeline. It highlights how the cell area and inner segmentation are split up and accomplished using different methods. Afterwards they are combined to create the full segmentation map.

Chapter 3

Experiments

3.1 Experimental Setup

In this section, we will present the experiments designed to evaluate the performance of our adapted MTObjects segmentation tool. Our primary goal in this study was the generation of accurate segmentation maps from low dwell time images and compare them to our original high dwell time maps, which serve as a ground truth. To fully compare these images we will use several different metrics, including Precision, Recall, F1-Score, Area Score and Normalized Mean Absolute Error (NMAE). To optimize our input parameters we will use a combination of bayesian optimization, parameter matching and regression methods. This will be done for varying training set sizes, namely 3, 10, 80 and 100 (full). Here we both want to find out how many training samples are needed to produce a nice fitting regression line, while also making sure a suitable test set size is maintained. For the final set we will both train and test on the full data set, to find the ideal settings for our generalization testing on the secondary datasets. Lastly we also want to do the same metric evaluation and optimization with and without the image reconstruction for lr1 images to compare how this feature impacts performance.

To learn more about both our data set and our pipeline we also developed other experiments including a sensitivity analysis on the move factor variable and a deep dive into pixel mapping between the high and low dwell time images. The move factor variable was chosen for the sensitivity analysis because experimental testing revealed it to be by far the most influential parameter. The sensitivity analysis should show both how much the performance of the pipeline varies with different move factor values and how well our optimization performance manages to find the optimal settings. Meanwhile, the mapping should give us more information how well the dwell times relate to the ground truth, both in terms of intensity and noise variation.

Finally we also devised two further experiments using our other two datasets: a dataset containing higher resolution STED images and one comparing confocal and STED images of the same scene. The first data set will allow us to see how well our optimized pipeline generalizes to different, higher resolution STED images. As this data set does not contain any ground truth images (and therefore no ground truth segmentation maps) this would mostly be a visual inspection to evaluate its correctness. Our second data set will give a detailed comparison between the segmentation maps received from STED and confocal images. To assess how much

information the lower resolution confocal image map maintains, we will determine how many objects contained in the STED image are also in the confocal image. Here we will make the distinction between one-to-one hits, where a single object in STED matches a single object in confocal and merged hits, where multiple STED objects matches a single object. By making this distinction we can not only tell how well the confocal segmentation map manages to find objects but also how well the confocal map delineates objects and how many are merged due to the decrease in resolution.

3.2 Metrics

To fully compare the segmentation maps of the lower and higher dwell time images as well as to evaluate the optimization, several metrics are defined. As we have split up our pipeline between cell area and inner-segmentation we can evaluate and optimize these separately. To evaluate the cell area we will use four different formulas, namely Precision , Recall, F1-Score and Area Score. Here it is important to note that for cell area evaluation we only have two values: 1 (positive) for a cell pixel and 0 (negative) for a background pixel. This is needed as these metrics can only be applied to binary classification problems. Precision (3.1) is defined by dividing the True Positives (TP) by True Positives plus the False Positives (FP), essentially measuring how many of our true predictions are in fact true. Meanwhile, recall (3.2) is calculated by dividing the True Positives by the True Positives plus the False Negatives (FN). Here we find how many of the true positives we have managed to find. As both of these metrics range from 0 to 1, it therefore means that if we only have true positives that precision is 1, while if we found all true positives than recall will be 1 (which means that if both are equal to 1 we have a perfect match). The F1-score (3.3) combines these metrics to achieve a balanced measure that takes both into account.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.2}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

Our final area metric, Area Score (3.4), is simply a score from 0 to 100 measuring how many pixels match between the lr segmentation map and the ground truth. It is defined by summing how many pixels from both images are either both cell or both background pixels and at the end dividing this by the total number of pixels present in the image.

$$\text{Area Score} = \frac{1}{n}\sum_{x=1}^{n}\delta(f_1(x), f_2(x))$$
$$\text{where}$$
$$\delta(f_1(x), f_2(x)) = \begin{cases} 1 & \text{if } (f_1(x) = 1 \text{ and } f_2(x) = 1) \\ & \text{or } (f_1(x) = 0 \text{ and } f_2(x) = 0) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

For the inner-segmentation we only have a single metric defined, namely the Normalized Mean Absolute Error (NMAE). NMAE is calculated by first normalizing the pixel values from both segmentation maps; this is to make sure that both intensities range from 0 to 1. Furthermore this means, as the intensity bins are created based on the intensity found within the image, that if both pixels are on the same relative intensity level the NMAE should be 0. Moreover, the NMAE should increase linearly based on the number of intensity levels separating both pixels. We therefore get a metric that only increases if the pixels are not on the same intensity level, which shows how well the inner-segmentation matches between maps. These 5 metrics should give us insight how well our final segmentation maps match, both in terms of cell area and inner-segmentation.

$$\text{NMAE} = \frac{1}{n}\sum_{x=1}^{n}|f_1(x) - f_2(x)| \tag{3.5}$$
Here the values $f_1(x)$ and $f_2(x)$ are normalized.

3.3 Optimization

Parameter optimization is an important step to ensure we get the optimal results for our pipeline. In our case we need to fine tune parameters for both blurring technique's: the blur parameter for Gaussian blurring as well as κ and γ for Perona-Malik anisotropic diffusion. For Gaussian blurring the blur parameter simply increases the strength of the blurring, while κ and γ influence the edge sensitivity and diffusion rate respectively. A third parameter for Perona-Malik, number of time steps t, is remained constant at 5. As mentioned in the previous section we can optimize these variables for both objectives separately to ensure the best results. Lastly, we also want to optimize the move factor λ to ensure the right balance between ignoring noise while retaining faint objects. To get the optimal values for each of these parameters we will use Bayesian optimization in combination with several regression techniques, using the average pixel intensity found within the cell image as independent variable.

Bayesian Optimization is a sequential model-based approach used for global optimization. It uses a probabilistic model in order to predict the performance of input parameters and iteratively updates these based on some objective function. In our case this objective function is two-fold: one for the cell area and one for the inner segmentation. For the cell area we will use the F1-Score ((3.3) as this is a well balanced metric that ensures we minimize both false positives and false negatives. For the inner segmentation we will use the NMAE (3.5). As bayesian optimization tries to maximize the objective score we reverse the absolute error in

order to turn it into a maximization problem.

By using this approach we can find the optimal settings for each STED image, giving us numerous data points matching average pixel intensity to their optimal parameter. In order to match each pixel intensity to the optimal parameters we will use varying regression techniques where the formula will be used to find the optimal settings. When determining the best fitting equation several things need to be considered: a predefined or experimental limit and over- and under-fitting. Taking this both into account we have opted for a linear regression for our blur parameter, a polynomial function for γ and κ, and an increasing exponential decay function has been used to fit the move factor variable.

Chapter 4

Discussion & Results

4.1 Image Segmentation for Low Dwell Time Images

Our main experiment focuses on image segmentation for low dwell time images, analyzing the extent of information retention when reducing the dwell time. We will begin by discussing our optimization for varying training sizes—specifically 3, 10, 80, and 100 (full). In this phase, our goal is to determine the minimum number of training samples needed to create a well-fitting regression line while ensuring a sufficiently large test set is maintained. Following this, we will evaluate how well our pipeline performs across all train/test sets for each metric outlined in the experiments section. Additionally, both during optimization and result analysis, we will examine the performance of each blurring type separately, comparing their effectiveness. Finally, we will identify the settings, parameters, and blur types that yield optimal results for our pipeline.

Gaussian Results

The optimization results for the Gaussian blurring method can be found in fig 4.1. Here we can see (in order from top to bottom) for the training sizes 3, 10, 80, and 100 the plots for both the blurring and move factor variables against the average pixel intensity, including a best fitting regression line. As we can see, for the blurring parameter this regression line seems to be very stable across all training sizes, only differing slightly in height or slope. The move factor varies more across the different plots; for the lowest training size λ's regression curve is significantly higher than the rest. However, as the training size increases, this variation stabilizes, with the curves for 80 and 100 being nearly identical.

A table showing the results for each training size can be found in fig 4.2. Here, it is evident that a training size of three yields significantly worse results across all metrics. When the training size is increased to ten or higher, the results become very similar: while the two largest training sizes slightly outperform the others on some metrics, on average, all three are almost identical. This closely aligns with the findings from our optimization; with three training samples, we observed both a significantly different curve and lower results, whereas the curves and results for the other three sizes are very similar. From this, we can conclude that three image sets are insufficient to fit the optimal regression curve, whereas ten seems to be adequate for achieving this.

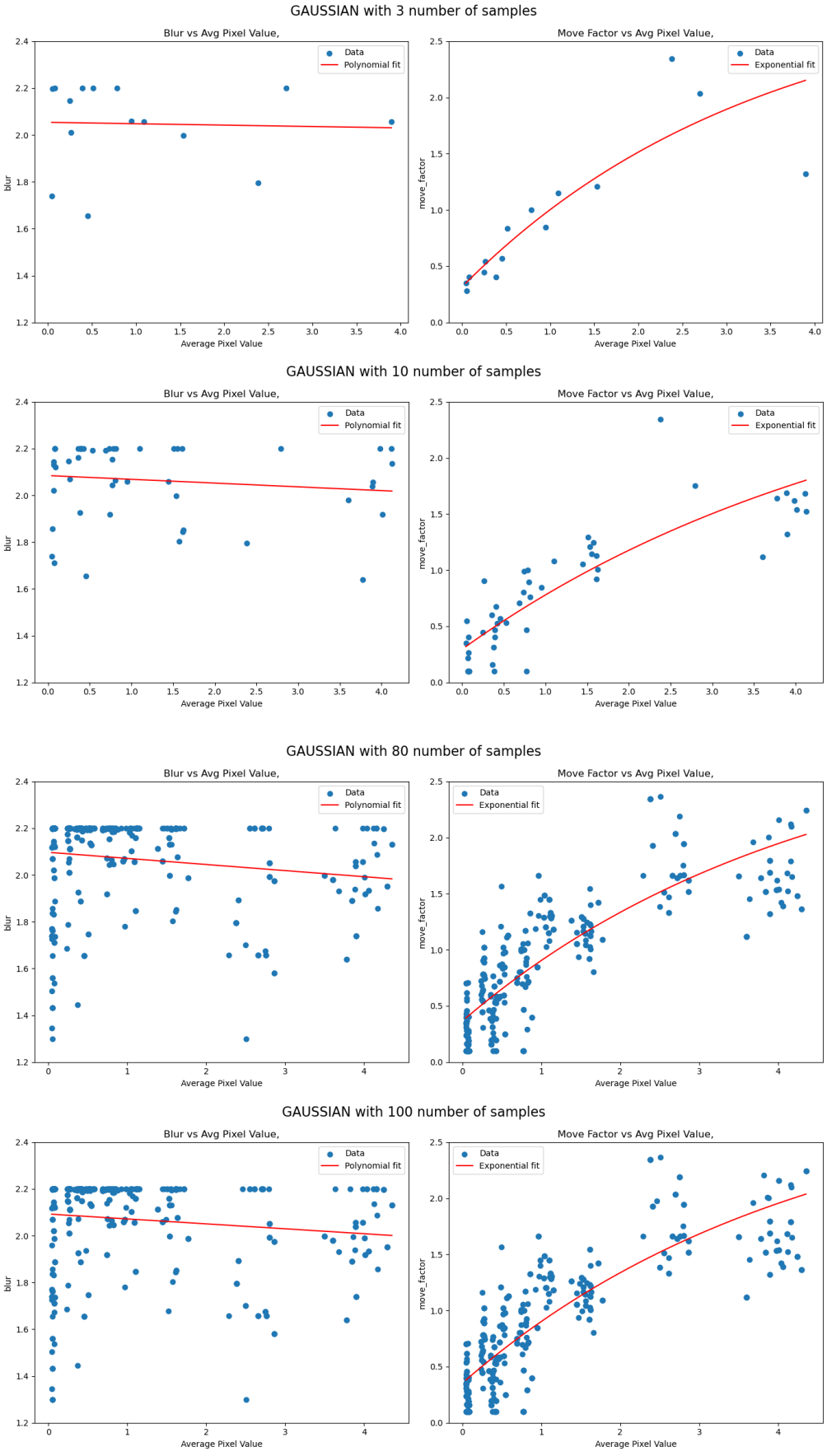


Figure 4.1: The optimization results for the Gaussian blurring method. For each training size (3,10,80,100) we see the blur (left) and move factor (right) matched against the average pixel intensity, with a best fitting regression line. This regression line shows what input parameters are used based on this average pixel intensity.

Figure 4.2: Tables showing the final results for Gaussian blurring, optimized with 3, 10, 80, and 100 training image sets. Gaussian 100 is optimized and tested on the full dataset.

Gaussian 3

LR	Score (%)	MAE	F1 Score
lr1	90.49	0.041	0.646
lr2	94.50	0.026	0.824
lr3	95.68	0.021	0.866
lr4	97.34	0.015	0.920
lr5	97.28	0.014	0.920

Gaussian 10

LR	Score (%)	MAE	F1 Score
lr1	90.72	0.041	0.668
lr2	94.58	0.026	0.829
lr3	95.70	0.020	0.870
lr4	97.41	0.014	0.923
lr5	97.63	0.013	0.930

Gaussian 80

LR	Score (%)	MAE	F1 Score
lr1	91.40	0.040	0.684
lr2	94.79	0.025	0.838
lr3	95.44	0.021	0.863
lr4	97.51	0.013	0.925
lr5	97.84	0.011	0.935

Gaussian 100

LR	Score (%)	MAE	F1 Score
lr1	91.22	0.040	0.690
lr2	94.78	0.025	0.837
lr3	95.73	0.020	0.871
lr4	97.37	0.014	0.921
lr5	97.65	0.013	0.930

Perona-Malik Results

We performed the same optimization for the Perona-Malik blurring, this time for three parameters, γ, κ and λ. The results of these optimizations can be found in fig 4.3. Here we see a similar pattern to the Gaussian blurring: For the lowest training samples a curve that deviates across all parameters, while the curves become extremely similar from ten training samples onward. However, the curves for both γ and κ seem to differ more than was the case for the Gaussian blurring, while λ is consistently slightly higher for both blurring methods.

Table 4.4 illustrates the effect of these divergent curves found for our lowest training size. Here we see a significantly worse results, especially for lr1, when only using this small sample size to fit our regression line. A difference of 2% and a decrease of 0.1 in F1-score are sizeable reductions in performance for the lowest dwell times, while the other dwell times also show a (reduced) performance degradation compared to the higher training sample curves. For the other training sizes we again see the same pattern as the Gaussian curve, where slight performance differences can be seen between these training sizes but not significant enough to deem them relevant. For both blurring methods we see that ten image sets should be sufficient to properly optimize the input parameters, while for Perona-Malik blurring we see a steeper drop in performance when using three image sets, most notably for the lr1 segmentation maps.

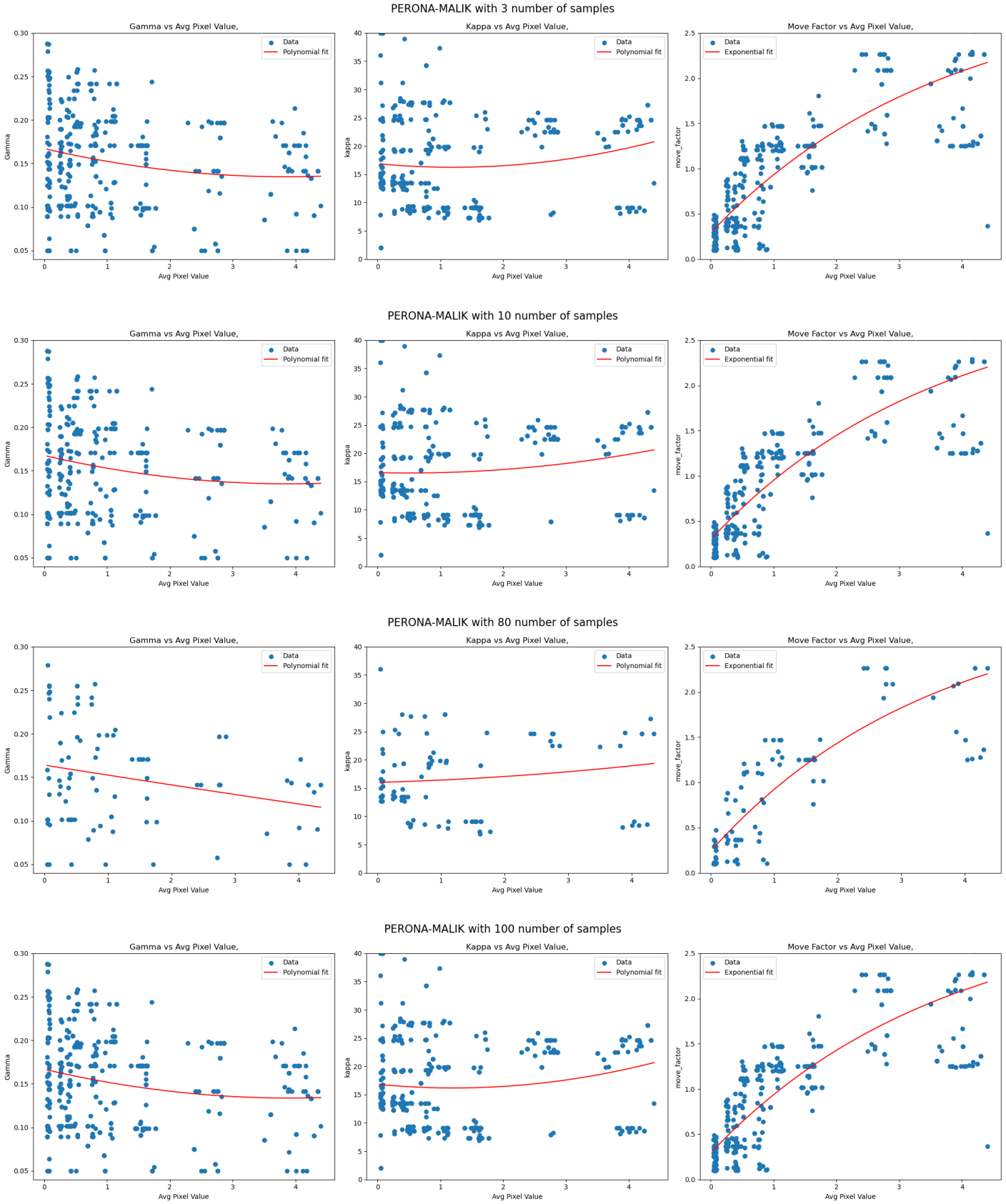


Figure 4.3: The optimization results for the Perona-Malik blurring method. For each training size (3,10,80,100) we see the γ (left), κ (middle) and move factor λ (right) matched against the average pixel intensity, with a best fitting regression line. This regression line shows what input parameters are used based on this average pixel intensity.

Figure 4.4: Tables showing the final results for Perona-Malik blurring, optimized with 3, 10, 80, and 100 training image sets. Perona-Malik 100 is optimized and tested on the full dataset.

Perona-Malik 3

LR	Score (%)	MAE	F1 Score
lr1	90.68	0.039	0.647
lr2	95.02	0.025	0.842
lr3	95.91	0.021	0.874
lr4	97.39	0.015	0.923
lr5	97.61	0.015	0.929

Perona-Malik 10

LR	Score (%)	MAE	F1 Score
lr1	92.48	0.036	0.746
lr2	95.25	0.024	0.855
lr3	96.07	0.020	0.884
lr4	97.40	0.016	0.924
lr5	97.54	0.015	0.927

Perona-Malik 80

LR	Score (%)	MAE	F1 Score
lr1	92.53	0.036	0.738
lr2	95.51	0.022	0.865
lr3	95.80	0.020	0.875
lr4	97.47	0.015	0.926
lr5	97.82	0.013	0.934

Perona-Malik 100

LR	Score (%)	MAE	F1 Score
lr1	92.34	0.036	0.738
lr2	95.29	0.024	0.857
lr3	96.05	0.020	0.883
lr4	97.33	0.015	0.921
lr5	97.68	0.014	0.930

Optimal results

To find our final optimal pipeline we want to fully compare both blurring methods. In table 4.5 we see a comparison between both blurring methods, for all specified metrics, using the full data set for both training and testing. The results provide a clear overview of the strengths and weaknesses of each method: Gaussian blurring shows slightly higher precision and a lower MAE for the higher dwell times, while Perona-Malik significantly outperforms Gaussian in Area Score, Recall, and F1-Score. The differences in Precision and Recall between these blurring methods highlight their distinct effects on image processing: Gaussian blurring leads to higher Precision by smoothing the image uniformly, reducing false positives and focusing on clearer, more prominent objects. In contrast, Perona-Malik filtering preserves edges and fine details, resulting in higher Recall by detecting more true positives, including smaller or less distinct objects, even if it increases false positives slightly. Since the increase in Precision for Gaussian blurring is much smaller than the increase in Recall for Perona-Malik, this results in higher Area and F1-Scores for the latter, making it the preferred method for our area pipeline.

However, the results for mean absolute error are much closer, with Perona-Malik outperforming Gaussian at lower dwell times, while Gaussian has the advantage at the three highest dwell times. This makes sense when considering the inherent application of anisotropic diffusion: it increases blurring in homogeneous areas, thereby smoothing out subtle intensity variations and reducing the visibility of inner-segmentation details. This characteristic is illustrated in Fig. 4.6 , where we compare the ground truth (left), Gaussian (middle), and Perona-Malik (right) side by side. It is evident that Perona-Malik struggles more with identifying peaks, missing one entirely and showing a much smaller area for the other. These inherent characteristics of anisotropic diffusion make it less suited for optimal inner-segmentation.

Therefore, to get the best possisble results, we will combine both blurring methods to finalize

our optimal pipeline. Here we use Perona-Malik's adaptive blurring to find precise cell areas, while Gaussian uniform blurring is used to find the best possible inner-segmentation. Our final results can be found in table 4.7, showcasing an identical Area and F1-score to the previous Perona-Malik scores, however now with equal or lower MAE for four of the five dwell times. An example of how these segmentation maps compare to each other and to the ground truth can be found in fig 4.8, while three more examples can be found in Appendix A.

Figure 4.5: Final results, showing all metrics for Gaussian (top) and Perona-Malik (bottom). Both of these are optimized and tested on the entire data set.

LR	Score (%)	MAE	Precision	Recall	F1 Score
lr1	91.22	0.040	0.879	0.584	0.690
lr2	94.78	0.025	0.914	0.789	0.837
lr3	95.73	0.020	0.894	0.863	0.871
lr4	97.37	0.014	0.914	0.932	0.921
lr5	97.65	0.013	0.949	0.917	0.930

LR	Score (%)	MAE	Precision	Recall	F1 Score
lr1	92.34	0.036	0.870	0.663	0.738
lr2	95.29	0.024	0.904	0.837	0.857
lr3	96.05	0.020	0.883	0.898	0.883
lr4	97.33	0.015	0.906	0.940	0.921
lr5	97.68	0.014	0.942	0.924	0.930

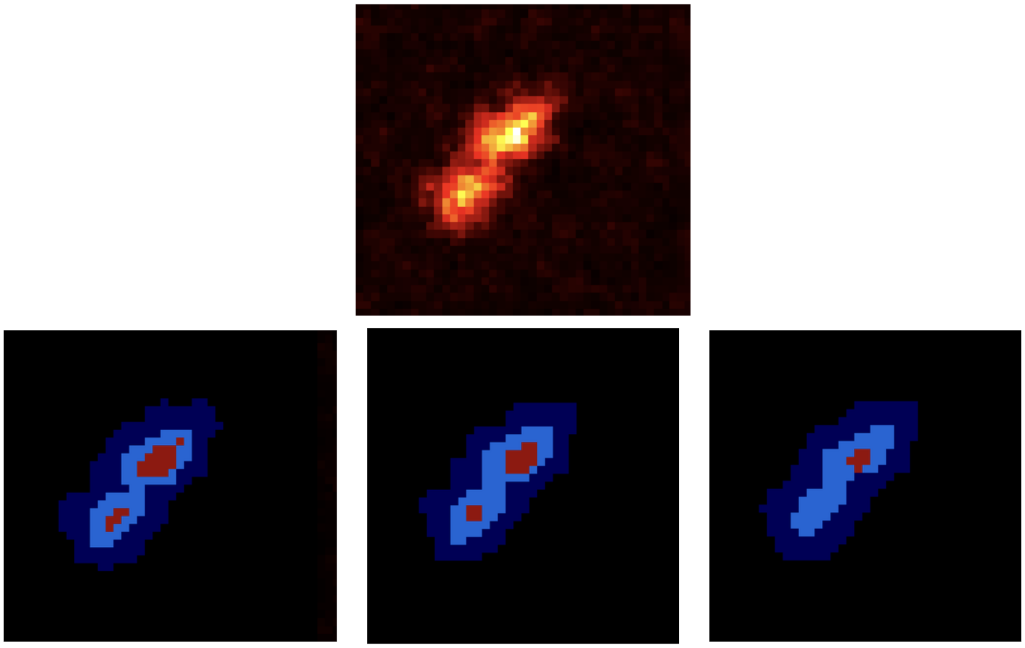


Figure 4.6: A cell image (top) with their corresponding ground truth (left), Gaussian (middle) and Perona-Malik (right) segmentation maps. This image showcases how Perona-Malik blurring has difficulties in showcasing intensity increases and therefore is not ideal for inner-segmentation.

Figure 4.7: Final results, combining Perona-Malik blurring for cell area and Gaussian blurring for inner-segmentation. These results are based on the optimization and testing on the entire data set.

LR	**Score (%)**	**MAE**	**Precision**	**Recall**	**F1 Score**
lr1	92.34	0.040	0.870	0.663	0.738
lr2	95.29	0.024	0.904	0.837	0.857
lr3	96.05	0.020	0.883	0.898	0.883
lr4	97.33	0.014	0.906	0.940	0.921
lr5	97.68	0.013	0.942	0.924	0.930

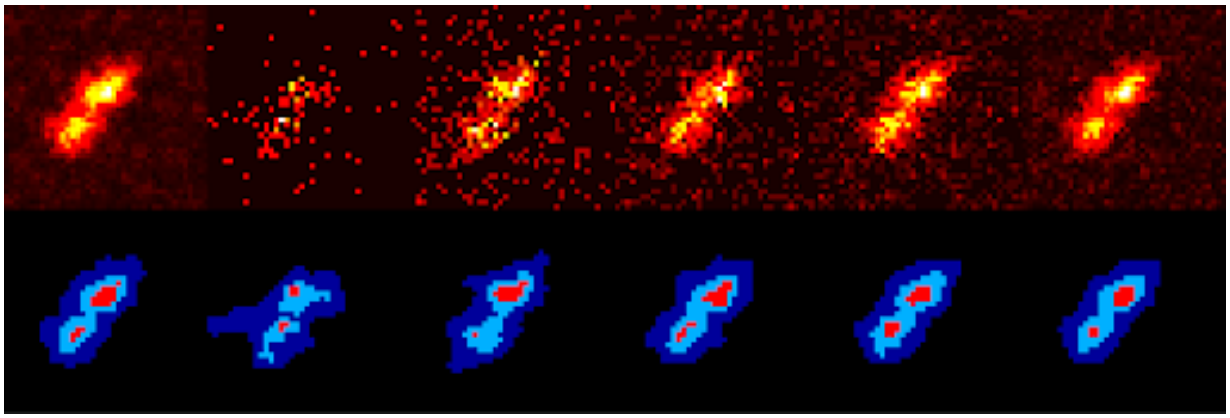


Figure 4.8: Cell images (top) ranging from the original image and lr1-lr5, with their corresponding segmentation maps below them. This image illustrates how the segmentation maps improve with dwell time, more closely mimicking the ground truth.

4.2 LR1 Image Reconstruction

Our second experiment focuses on the application of image reconstruction for the lowest dwell time images. The goal here is to fill in gaps to more easily and accurately determine object sizes. The results for running our pipeline with and without this image reconstruction can be found in table 4.9. For the precision and recall we see some expected results, where the precision is higher when using no image reconstruction, while the recall is significantly higher when we do use it. As we are effectively filling in gaps within the image, we might also increase some noise patches, resulting in a slightly lower precision. However, we do see that we find much more cell area pixels, as our Recall is increased by over 50% percent. This slight decrease in Precision and this steep increase in Recall results in a significantly increased Area and F1-Score, showcasing the efficacy of our image reconstruction implementation. These results can be verified when examining the produced segmentation maps for both methods, showing objects that are either drastically decreased in size or absent altogether when not using image reconstruction. An example for this can be found in fig 4.10, while several more can be found in appendix B.

Figure 4.9: Comparison of metrics for lr1-NORC and lr1-RC

LR	Score (%)	MAE	Precision	Recall	F1 Score
lr1-NORC	89.72	0.041	0.949	0.431	0.578
lr1-RC	92.34	0.040	0.870	0.663	0.738

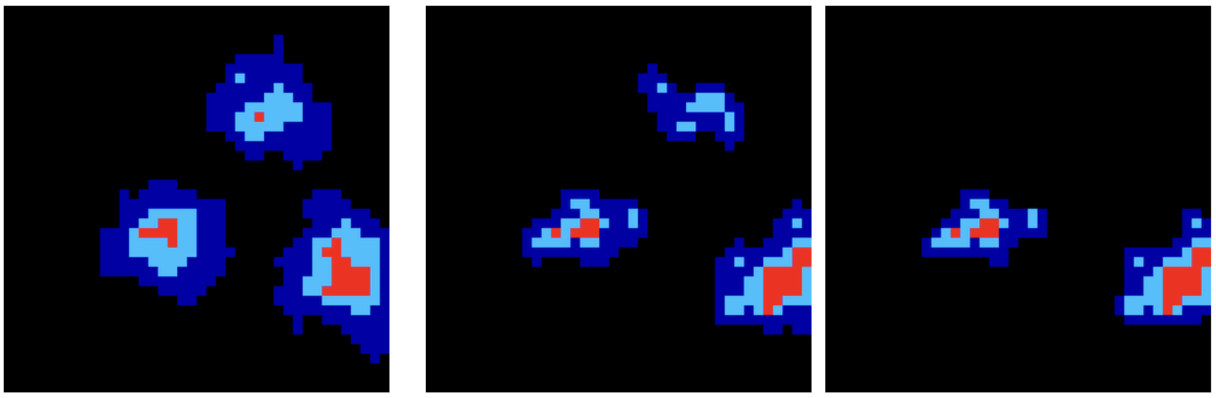


Figure 4.10: Three segmentation maps for the ground truth (left), lr1 with image reconstruction (mid) and lr1 without image reconstruction (right). Here an object that is found when using image reconstruction is completely missed when not using it.

4.3 Sensitivity Analysis

For our sensitivity analysis we have looked at how variations in the move factor variable influences the precision, recall, and more importantly the F1-score. As fig 4.11a and fig 4.11b demonstrate, both the precision and recall show expected behaviour: precision increases when λ increases, while recall decreases. This is because when increasing move factor we essentially decrease object area, reducing the number of false positives but increasing the number of false negatives. For the F1-Score, to get the most complete picture, we have tested this for three different image sets, including a best 4.11c, worst 4.11d and average 4.11e scenario. Each line corresponds to an F1-Score for varying move factors, while the dashed vertical line corresponds to the move factor found by our optimization process. We can see that for the best and average case scenario, our dashed line is extremely close to the peak of each line. Meanwhile, our worst case scenario only has the lr1 peak far away from our chosen parameter value. This shows that for all scenarios, our optimization manages to find the parameters that are fairly close to the most optimal ones. Upon further inspection we can defer several interesting conclusions from these plots. For one, each dwell time shows a clear peak that is either extremely flat (lr5 in best case scenario) or extremely narrow (lr1 in worst case scenario). It also shows that each peak is in order from lowest to highest dwell time, which is in line with our expectation of move factor and dwell time being positively related to each other. Finally we see that the λ has an enormous influence on the final performance of our segmentation map: a move factor of 3 results in a score of zero for all lr1 images, while a move factor of 0 gives results ranging from 0.4-0.8 for the higher dwell times.

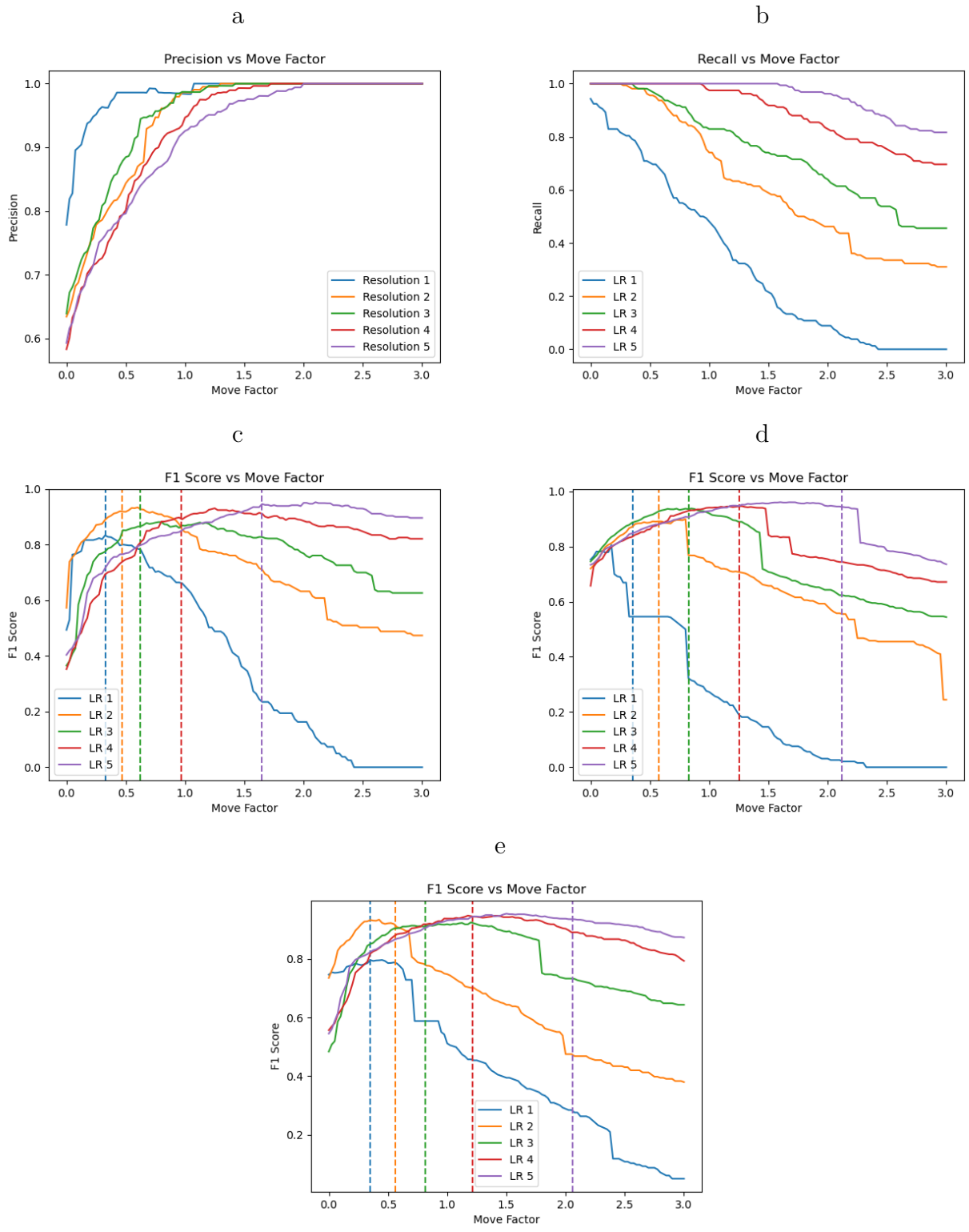


Figure 4.11: Sensitivity analysis for various metrics highlighting the behaviour of the move factor variable. (a) Precision metric, (b) Recall metric, (c) F1-Score metric in the best case scenario, (d) F1-Score metric in the worst case scenario, and (e) F1-Score metric in the average scenario.

4.4 Mapping

In this section, we attempt to ascertain the relationship between the pixel intensities of the high and low dwell time images, as well as explore any potential correlation in the amount of noise within the pairs. To achieve this, we have mapped each low dwell time image against its corresponding high dwell time image, as shown in Fig. 4.12. We observe that for lr1, there are only a few possible pixel values, resulting in a relationship between their intensities that is only somewhat linear. However, as the dwell time increases, we see a clear increase in linearity. This trend is supported by Table 4.13, which displays the R-values for all pairs. Here, we note that while lr1 exhibits only a weak linear relationship, this steadily increases until we observe a very strong relationship between lr5 and hr. This indicates that there is indeed some linear mapping between the lower dwell time and high-resolution images, providing us with more information about our ground truth.

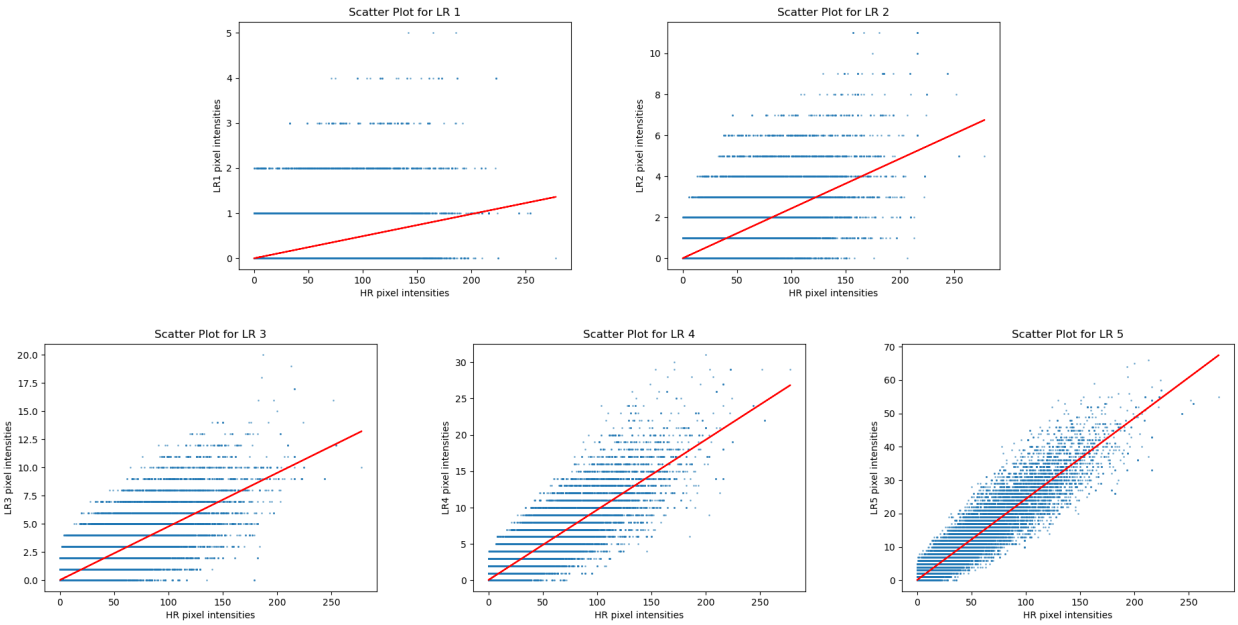


Figure 4.12: Pixel intensity mapping, matching all lower dwell time images lr-lr5 to their corresponding high dwell time image. It shows how there is a linear relationship between pixel intensity for the pairs, increasing as dwell time increases.

Figure 4.13: r values for all high and low dwell time image pairs. This shows that as the dwell time increases, the linear relationship also increases rapidly.

LR	**(r)**
lr1	0.399
lr2	0.683
lr3	0.789
lr4	0.876
lr5	0.939

When we look at noise patterns found in our pairs we have found several interesting observations. Firstly, when viewing fig 4.14 which shows the residuals of our linear mapping for lr5, we can see that they seem to be increasing as the pixel intensity increases. This indicates that with an increase in intensity, the noise also increases. To further explore this, we have created multiple intensity bins in order to plot variance against pixel intensity, found in fig 4.15. We can see that this line (besides above 200 intensity which has a lack of data points) is extremely linear. This means that noise increases linearly with variance and has a square root relationship with intensity, giving us even more insight in the relationship and characteristics of these images.

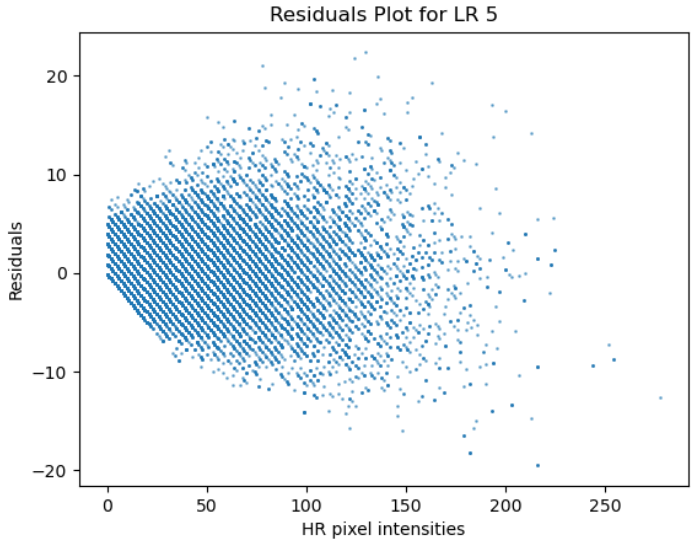


Figure 4.14: Residual mapping between the lr5 images and the hr images. As intensity increases, the spread of the residuals also increases.

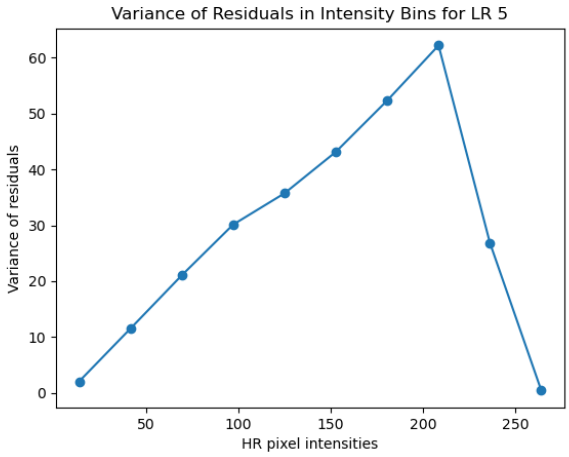


Figure 4.15: A plot of the variance from bins for the lr5 images and the hr images. It show how a linear relationship between variance and pixel intensity, showing that noise increases linearly with variance.

4.5 Secondary Experiments

Finally, we will also discuss the results for our final two secondary experiments. Our first experiment, pertaining the generalization to higher resolution STED images, showed mixed results. In fig 4.16 we see several segmentation maps that show pristine segmentation maps, clearly showing the right cell area and properly denoting higher intensity areas. However, in fig 4.17 we also found several segmentation maps that found a large number of small objects around our main cell. These errors are likely a result of faulty background estimation, partly due to a highly fluctuating background which contains both background noise but also noise distributed by the cell itself.

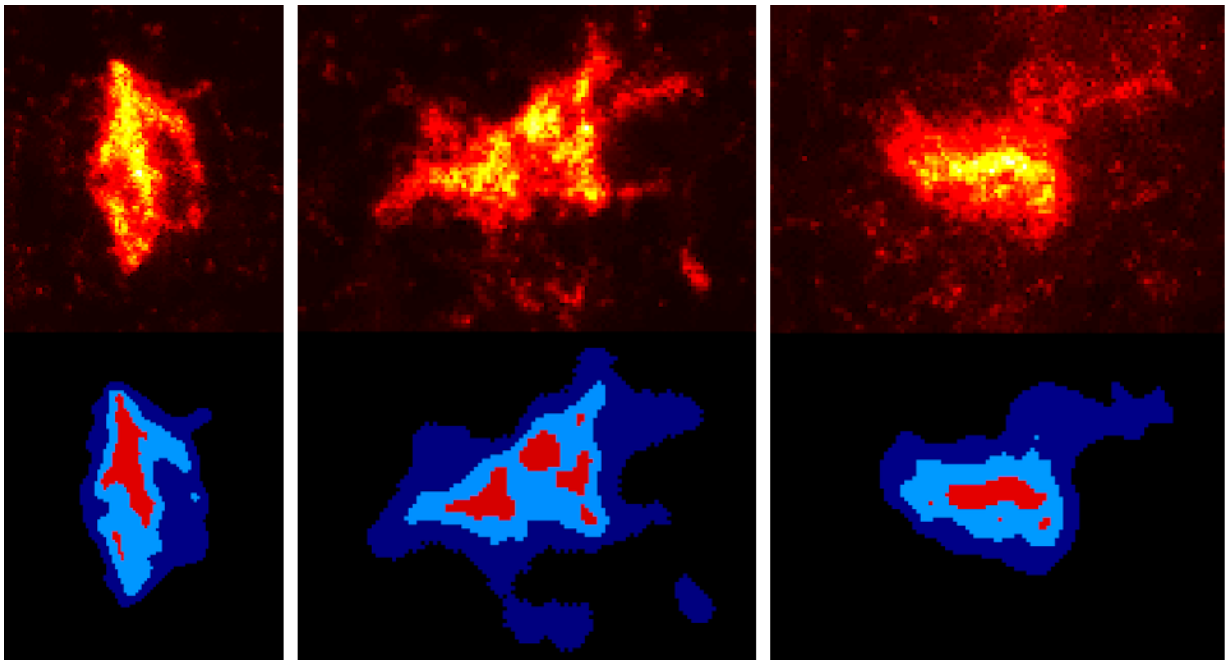


Figure 4.16: Some promising results when applying our optimized pipeline to a new higher resolution STED image data set. It shows a near perfect segmentation map for both area and inner-segmentation.

Our second experiment, the object detection between STED and confocal objects, showed some interesting insights. For this experiment we have artificially lowered the move factor for the confocal images, as the SNR is much lower for this technology, to make sure we find as much objects as possible. Some images showcased fairly good overlap, managing the find a majority of objects. Fig 4.18 is a great example for this, where the STED segmentation map (left), the confocal map (middle) and their matches (right) are shown. Here we can see that a large number of STED objects are also found in the confocal map, with the caveat that a sizable amount are found in merged fashion. However, due to the confocals low resolution, when a large number of objects are close together it merges these all into one. While we still retain some information compared to the STED map, for most cases this map will find much more distinct object, along with their accurate cell areas. To summarize our results, we have documented the average number of STED objects and confocal matches (both single and merged) per image, shown in table 4.19. Here it shows out of all objects found within the STED image, only about

60% can also be found in the confocal image. Furthermore, out of the found objects, about 58% are part of some merged object, where multiple STED objects can be found within a single confocal object. It shows that a slight majority of objects can be found, while also a slight majority of those found are merged together.

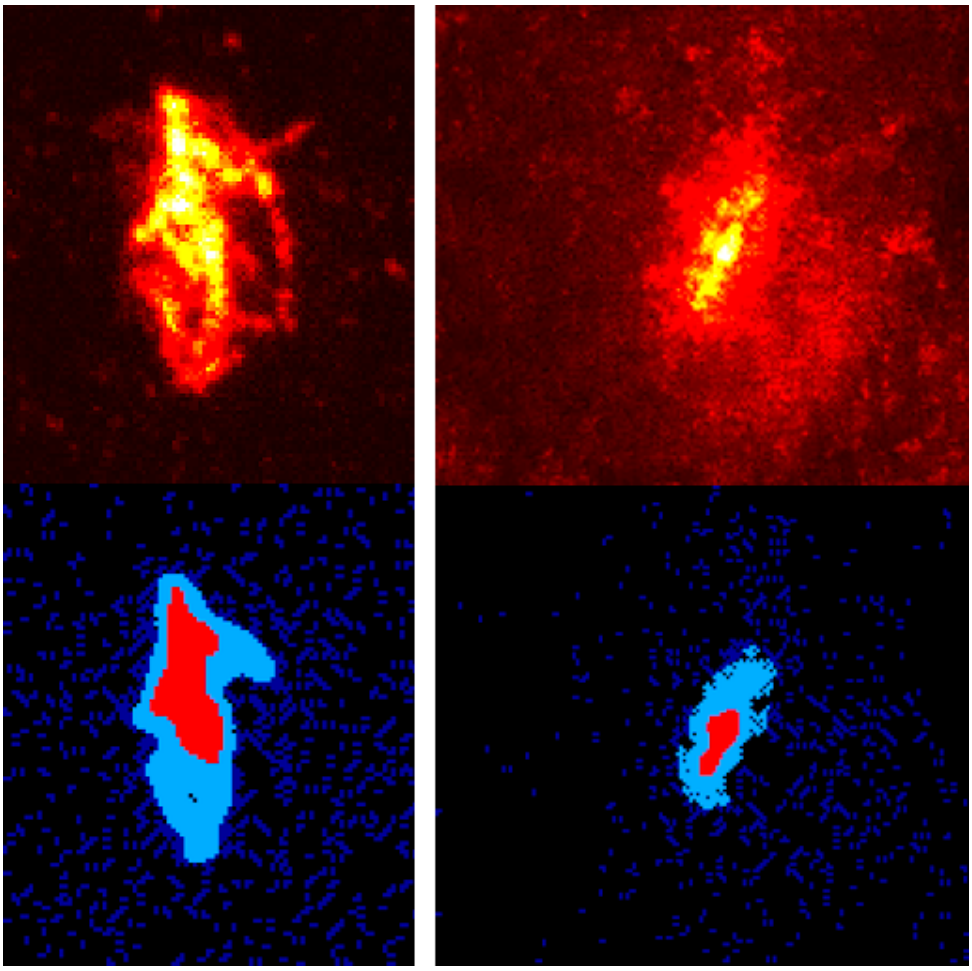


Figure 4.17: Suboptimal results when applying our optimized pipeline to a new higher resolution STED image data set. It shows how the program has trouble differentiating what is noise and what are objects.

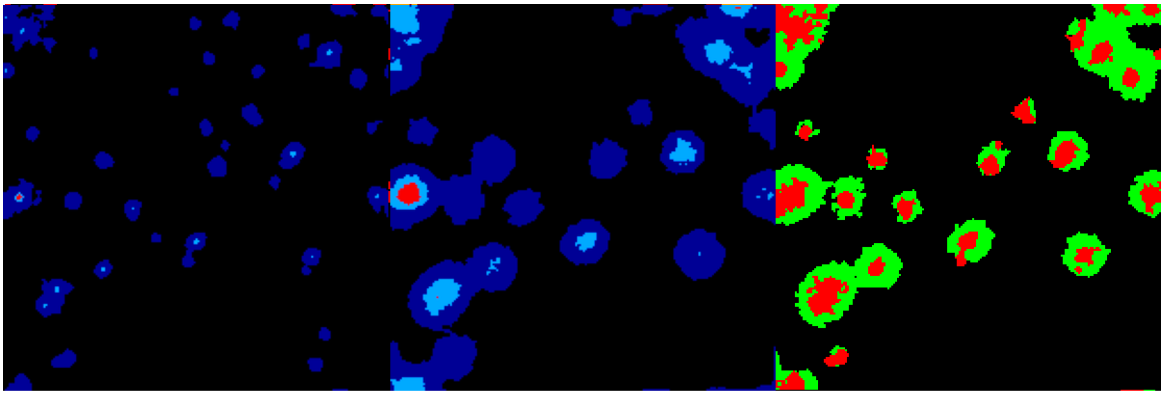


Figure 4.18: The segmentation map for the STED image (left), the confocal image (middle) and their matched objects (right). A majority of objects found in the STED map are also found in the confocal map, even though a large number of objects have been merged.

Figure 4.19: Summary of average STED metrics per pair. Merged hits are a subset of matched objects.

Metric	**Value**
Average STED objects per pair	31.60
Average Matched STED objects per pair	19.05
Average Merged hits per pair	11.09
Average Detection Rate	0.60

Chapter 5

Conclusion & Future work

5.1 Conclusion

In this study, we developed and optimized a segmentation pipeline for low dwell time STED microscopy images, aiming to preserve as much essential information as possible, while minimizing the negative effect caused by photobleaching and phototoxicity. By adapting the MTObjects segmentation tool, we successfully developed a pipeline tailored to live-cell imaging. By implementing Gaussian and Perona-Malik blurring techniques, along with reconstruction and new segmentation tools, our approach demonstrated robust performance in preserving cell area and enhancing inner-segmentation in low dwell time images.

Our findings have shown that using both Perona-Malik, for its ability to maintain edges, with Gaussian blurring for inner-segmentation has shown the best results. Great performance was shown for all discussed metrics, most notably showing F1-Scores ranging between 0.86 and 0.93 for lr2-lr5. While lr1's performance remained somewhat lacking, it still showed promise in giving only the most essential information about the cell. We have also provided a clear overview of how much information is retained at each dwell time, enabling informed decisions when selecting the appropriate dwell time. Furthermore, by using our adaptive bin based inner-segmentation method, this part of the segmentation map is highly adaptable. This adaptability is further illustrated by the fact that the ground truth can be chosen at will, with the pipeline designed to find the closest matching map for the lower dwell time images. Finally, we have also showcased that our image reconstruction technique drastically boost performance, allowing us to find much more faint objects accurately in the lowest dwell time images.

Our auxiliary experiments have led to several interesting conclusions. Firstly, our sensitivity analysis demonstrated that our optimization method consistently identifies parameters that are either optimal or very close to the optimal values, while also highlighting the critical role of the move factor variable in segmentation performance. Secondly, our mapping revealed a distinct linear correlation between pixel intensities for low and high dwell time images, as well as a linear increase in noise with variance. These insights provide us with a deeper understanding of both the images and their interrelationships. Finally, we have shown that our pipeline can be effectively applied to both secondary datasets. Although the results were mixed, the potential is evident, and this paper can serve as a foundation for further research into both areas.

In conclusion, this study presents an effective approach to optimizing low dwell time STED microscopy images, focusing on information retention and accurate segmentation mapping. The optimized pipeline, combining Perona-Malik and Gaussian blurring techniques, successfully balances image quality with the need to minimize dwell time. The methodologies developed in this work demonstrate a reliable strategy for enhancing the analysis of low dwell time images and provide a strong foundation for future advancements in cellular imaging.

5.2 Future Work

For our future work we have summarized a few novel approaches that might add or improve this work:

1. **Segmentation map analysis:** In this paper we have assumed that our ground-truth segmentation map represents an optimal segmentation map. While considerable thought has gone into determining what we believe to be the best possible map, this may not yield the best results for all purposes. A detailed segmentation map analysis could provide a better ground-truth, which would also result in lower dwell time maps that would be more ideal.
2. **(improved) Further generalization:** Our research has presented two data sets that tested in one way or another how well our optimized pipeline performed on new data. We saw that for our higher resolution images we got mixed results, with segmentation maps that closely match our expectations and others that resulted in some sub optimal images. Future work will focus on two main areas: addressing the issues encountered with current generalization and identifying additional datasets to further test and refine our generalization capabilities.
3. **Parameter matching:** During our optimization process we have matched each of our input paramaters to the average pixel intensity found within the image. While this metric gives direct information about an important feature of this image and is heavily correlated to dwell time, this might not be the ideal variable to use in all cases. An example for this could be large images with only a few cells, which would result in low average pixel intensity while this might not say much about the intensity of these cells. Possible adaptations could be using the average background noise as this should almost have a perfect linear relationship with dwell time or non-zero average pixel intensity which gives great information about the intensity of the cells found within the image. Exploring additional parameter matching could lead to better performance and enhanced generalization.
4. **Adaptive image reconstruction:** Finally, a final future implementation could be the addition of adaptive image reconstruction. While our implementation has shown great promise in the lower dwell time images, it was implemented fairly static, with the input values for all images. By making this process adaptable, the level of reconstruction could be adjusted based on specific image traits, thereby broadening the applicability and effectiveness of the technique.

Bibliography

[1] S. W. Hell and J. Wichmann, "Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy," *Optics Letters*, vol. 19, no. 11, pp. 780–782, 1994.

[2] S. Hell, K. Willig, M. Dyba, and S. Jakobs, "Nanoscale resolution with focused light: Sted and other resolft microscopy concepts," *ResearchGate*, 2006.

[3] C. Lee, D. Wallace, and P. Burke, "Photobleaching and phototoxicity of mitochondria in live cell fluorescent super-resolution microscopy," *Mitochondrial Communications*, 2024.

[4] T. Liu, T. Stephan, P. Chen, *et al.*, "Multi-color live-cell sted nanoscopy of mitochondria with a gentle inner membrane stain," *Proceedings of the National Academy of Sciences*, vol. 119, no. 37, p. e2215799119, 2022.

[5] G. Vicidomini, P. Bianchini, and A. Diaspro, "Sted super-resolved microscopy," *Nature methods*, vol. 15, no. 3, pp. 173–182, 2018.

[6] I. Coto Hernández, M. Castello, L. Lanzanò, and M. d'Amora, "Two-photon excitation sted microscopy with time-gated detection," *Scientific Reports*, vol. 6, p. 19419, 2016.

[7] J. Kwon, M. Elgawish, and S. Shim, "Bleaching-resistant super-resolution fluorescence microscopy," *Advanced Science*, vol. 9, no. 15, p. e2101817, 2022.

[8] T. Staudt, B. Harke, J. Fölling, A. Engelhardt, and S. W. Hell, "Rescue sted nanoscopy with reduced background and improved resolution at low light levels," *Nature Methods*, vol. 8, pp. 571–573, 2011.

[9] U. Bottanelli, R. Fernández-Miñán, F. Zeuschner, *et al.*, "Minfield-sted: Minimally invasive fluorescence intensity depletion sted nanoscopy," *eLife*, vol. 6, p. e22286, 2017.

[10] C. Li, S. Liu, W. Wang, W. Liu, and C. Kuang, "Recent research on stimulated emission depletion microscopy for reducing photobleaching," *Journal of Microscopy*, vol. 269, no. 1, pp. 69–78, 2018.

[11] V. Ebrahimi, T. Stephan, J. Kim, P. Carravilla, C. Eggeling, S. Jakobs, and K. Y. Han, "Deep learning enables fast, gentle sted microscopy," *Communications Biology*, vol. 6, p. 674, 2023.

[12] C. Haigh, N. Chamba, A. Venhola, R. Peletier, L. Doorenbos, M. Watkins, and M. H. Wilkinson, "Optimising and comparing source-extraction tools using objective segmentation quality criteria," *Astronomy & Astrophysics*, vol. 645, p. A107, 2021.

[13] E. Bertin and S. Arnouts, "Sextractor: Software for source extraction," *Astronomy and Astrophysics Supplement Series*, vol. 117, pp. 393–404, 1996.

[14] A. S. G. Robotham, I. K. Baldry, and e. a. Phillipps, Steven, "Profound: Source extraction and application to galaxy and mass assembly (gama)," *Monthly Notices of the Royal Astronomical Society*, vol. 476, no. 3, pp. 3137–3154, 2018.

[15] M. Akhlaghi, "Noisechisel and the use of asymmetry to robustly identify faint structure in astronomical images," *Astronomy and Computing*, vol. 12, pp. 122–135, 2015.

[16] M. C. Teeninga, E. Valentijn, and M. Bárcenas, "Mtobjects: An automated multithresholding object detection software for astrophysical images," *Astronomy and Computing*, vol. 16, pp. 67–77, 2016.

[17] S. Calovi, F. Soria, and J. Tønnesen, "Super-resolution sted microscopy in live brain tissue," *Neurobiology of Disease*, vol. 151, p. 105269, 2021.

[18] MathWorks, "Generate image from segmentation map using deep learning," 2023. Accessed: 2024-07-08.

[19] P. Teeninga, U. Moschini, S. C. Trager, and M. H. Wilkinson, "Statistical attribute filtering to detect faint extended astronomical sources," *Mathematical Morphology: Theory and Applications*, vol. 1, pp. 100–115, 2016.

[20] B. Turcotte, A. Bilodeau, F. Lavoie-Cardinal, *et al.*, "pysted: a sted microscopy simulation tool for machine learning training," in *Proceedings of the Workshop on AI to Accelerate Science and Engineering (AI-2-ASE)*, 2022.

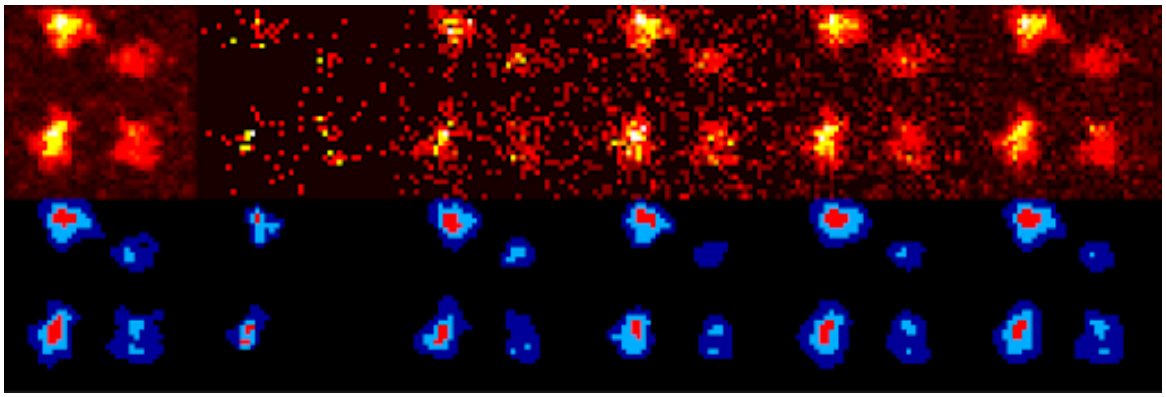
[21] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

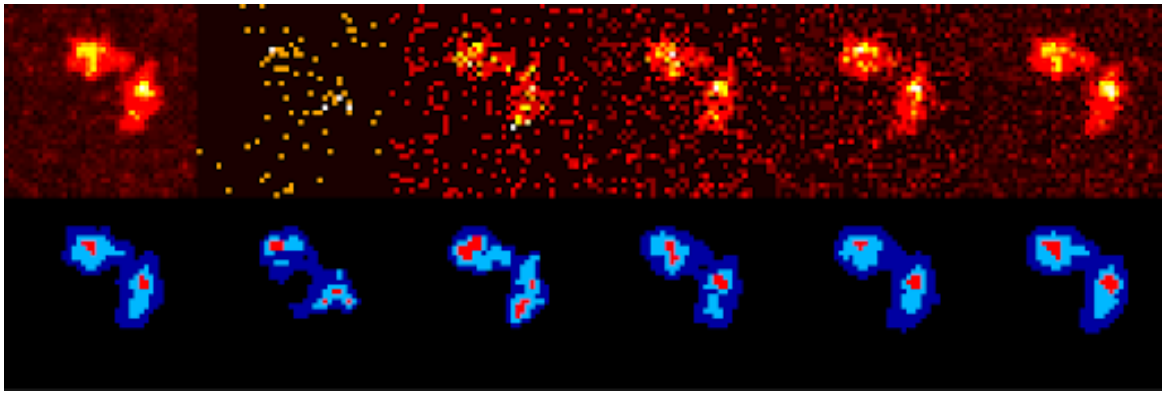
[22] P. Pan and D. Schonfeld, "Image reconstruction and multidimensional field estimation from randomly scattered sensors," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 94–99, 2008.

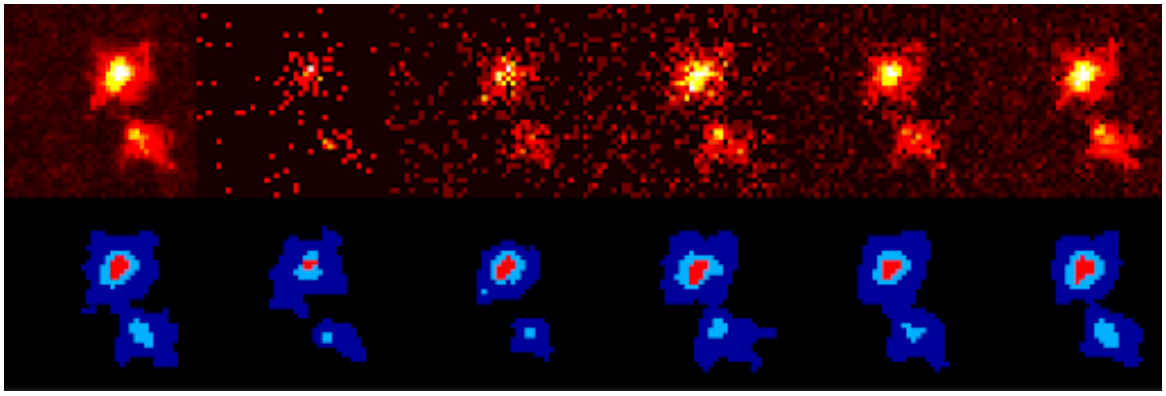
[23] S. Beucher and F. Meyer, "The watershed transformation applied to image segmentation," in *Conference on Signal and Image Processing in Microscopy and Microanalysis*, pp. 299–314, 1992.

Appendices

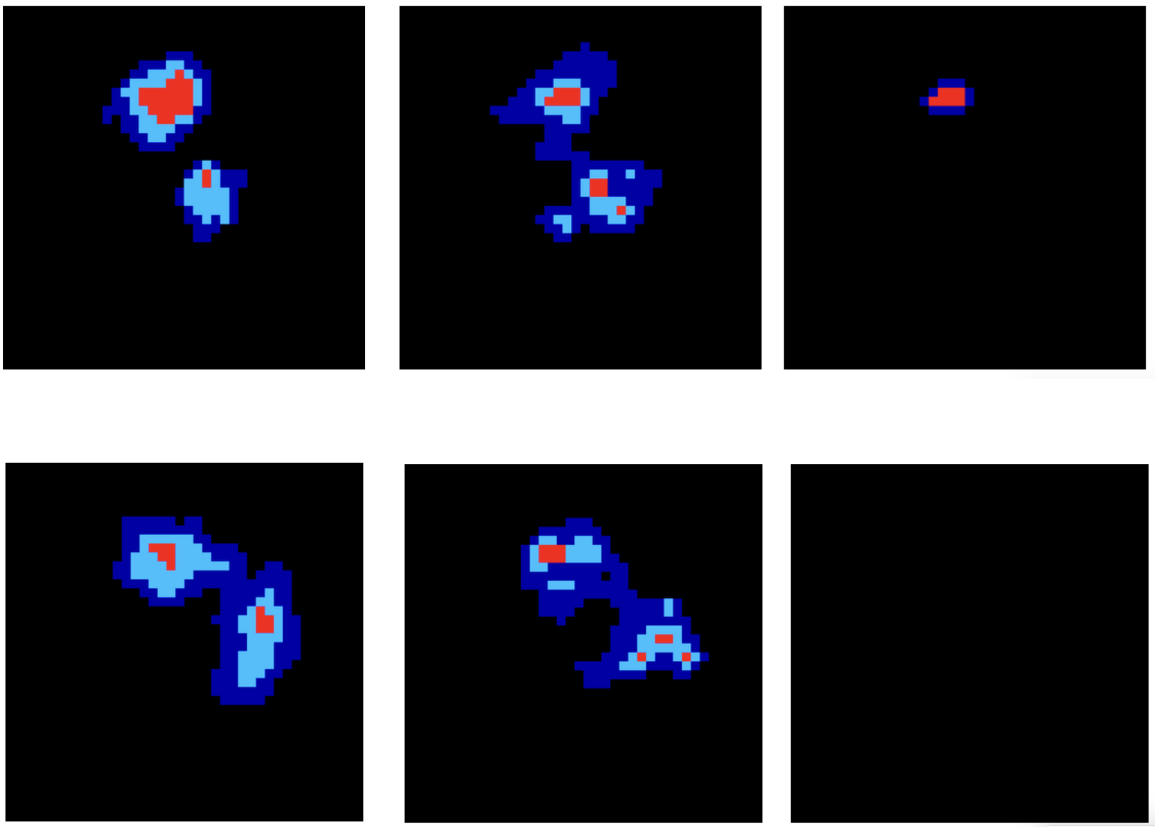
A Additional Segmentation Maps







B Additional Image Reconstruction Comparisons



Additional segmentation maps showing the ground truth (left), with image reconstruction (middle) and no image reconstruction (right).