Analysis and Synthesis of object overlap in Microscopy Images
Analysis and Synthesis of object overlap in Microscopy Images

Master Thesis report

June, 2012

Authors: Amir Etbaeitabari and Mekuria Tegegne

Supervisor: Dr. Stefan Karlsson
Co-Supervisor: Prof. Josef Bigun
Examiner: Prof. Antanas Verikas
Contents

Acknowledgements iii

Abstract iv

List of Figures v

1 Introduction 1
  1.1 Problem Statement 2

2 Background 5
  2.1 Related Works 6
    2.1.1 Related Classifier Algorithms 8

3 Method 11
  3.1 Synthesis 11
    3.1.1 Multi-Layered Microscopy 11
    3.1.2 Depth of Focus 15
    3.1.3 Analogy with Confocal Microscopy 22
  3.2 Analysis 23
    3.2.1 Parametric Description of DOF 23
CONTENTS

3.2.2 Feature Extraction ........................................... 25
3.2.3 Feature Selection ............................................. 27
3.2.4 Normalization ............................................... 28
3.2.5 Classification ............................................... 28
3.2.6 Watershed ...................................................... 33
3.2.7 Cell counting and overlap-bisection ......................... 35
3.2.8 Flowcharts for Synthesis and Analysis ................. 38

4 Experiments and Results ........................................ 41
  4.1 Experiment 1 (overlap detection by state of art algorithm) ... 43
  4.2 Experiment 2 (overlap detection by our algorithm) .......... 43
  4.3 Experiment 3 (counting cells by watershed and our algorithm) 46
  4.4 Result .......................................................... 50
    4.4.1 Result 1 (overlap detection by state of art algorithm) . 56
    4.4.2 Result 2 (overlap detection by our algorithm) ........... 57
    4.4.3 Result 3 (counting cells by watershed and our algorithm) 59

5 Conclusion and Future work .................................... 63
Acknowledgements

We want to thank for Dr. Stefan Karlsson for his direct supervision, motivation and continuous feedback and suggestion during the thesis writing period plays a great role for completing the thesis.

Furthermore, we would like to thank also Prof. Josef Bigun for his co-supervision, suggestion and advise, and we would like to pass our gratitude also to Prof. Antanas Verikas and A/Prof. Stefan Byttner for their suggestion and feedback.
Abstract

We propose a test-bed application for synthesis and analysis of multi-layered microscopy data with variation in depth of focus (DOF), where we consider the problem of detecting object overlap.

For the synthesis part, the objects are elliptical in appearance with the possibility of setting different parameters like noise, resolution, illumination, circularity, area and orientation.

For the analysis part, the approach allows the user to set several parameters, including sensitivity for error calculation and classifier type for analysis.

We provide a novel algorithm that exploits the multi-layered nature of the object overlap problem in order to improve recognition. The variation of gray value for each pixel in different depth is used as feature source for classification. The classifier divides the pixels in three different groups: background pixels, pixels in single cells and pixels in overlapping parts.

We provide experimental results on the synthesized data, where we add noise of different density. In non-noisy environments the performance for accuracy of overlapping positions is 93% and the performance of the missed overlaps is around 99.98% for density of 150 cells.
List of Figures

2.1 A Confocal Microscope hardware design and working principle 6

3.1 Geometrical representation of Distance function 13
3.2 Figure for Sigmoid function 14
3.3 Illustration of aliasing and anti-aliasing effect 15
3.4 16
3.5 A Synthesized Microscopic Pure Data 17
3.6 Pure Data with Gaussian Noise 18
3.7 Pure Data with Salt & Pepper Noise 19
3.8 Gaussian filters with different standard deviations(SD) 21
3.9 24
3.10 A 3D Volume Data 25
3.11 one single artificial neuron 31
3.12 A simple feed forward network with one hidden layer 32
3.13 Identifying overlap and drawing bisection line 36
3.14 Overlap cell after bisection 37
3.15 38
3.16 39
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.17</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>Illustration of Dice and overlapping pixel areas</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Binary Image from Ground Truth</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Binary Image from Linear Classifier</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>Binary Image from Artificial Neural Network Classifier</td>
<td>45</td>
</tr>
<tr>
<td>4.5</td>
<td>Input Binary Image from classifier containing 150 cells</td>
<td>46</td>
</tr>
<tr>
<td>4.6</td>
<td>Watershed segmentation result in Low resolution image</td>
<td>48</td>
</tr>
<tr>
<td>4.7</td>
<td>Comparing the segmentation result of our method with watershed</td>
<td>49</td>
</tr>
<tr>
<td>4.8</td>
<td>Noise Free and Normal Image</td>
<td>51</td>
</tr>
<tr>
<td>4.9</td>
<td>Image with 0.1 Noise Density</td>
<td>52</td>
</tr>
<tr>
<td>4.10</td>
<td>Image with 0.4 Noise Density</td>
<td>53</td>
</tr>
<tr>
<td>4.11</td>
<td>Image with down sample factor 2</td>
<td>54</td>
</tr>
<tr>
<td>4.12</td>
<td>Image with down sample factor 4</td>
<td>55</td>
</tr>
<tr>
<td>4.13</td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>4.14</td>
<td></td>
<td>57</td>
</tr>
<tr>
<td>4.15</td>
<td></td>
<td>58</td>
</tr>
<tr>
<td>4.16</td>
<td>Performance versus Resolution using Linear network</td>
<td>59</td>
</tr>
<tr>
<td>4.17</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>4.18</td>
<td>Performance versus Resolution using Artificial Neural network</td>
<td>61</td>
</tr>
<tr>
<td>4.19</td>
<td>Performance versus number of cells</td>
<td>61</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Counting white blood cells from microscopic images is a tool for physicians to diagnose human body health condition. A complete blood cell count (CBC) is a very common blood test that doctors ask for diagnosing some diseases. It counts the different types of cells in the blood and provides information regarding their size, shape and numbers. There are 3 different types of blood cells in blood stream, red cells that contain haemoglobin and carry oxygen, white cells that fight infections and platelets that help in blood clotting, which each of them contains different subsets.

White blood counting (WBC) can be a representative factor for different issues as infection amount in body. High amount of white cells can indicate an existing infection, leukaemia, or tissue damage while low amount indicates that a patient is in danger of infection.

In modern biology, visual analysis of cells by means of microscopes is a basic tool for identifying and counting of cells. Important factors like noisy environment, poor resolution and variant illumination affects the appearance of them critically. Moreover, when humans perform WBC estimation, they consider a smaller portion of a full image for estimating the total number of cells. However, in our case we used an automatic processing method for WBC detection and count, which is computationally fast and efficient approach.

Availability of a large clinical data set is necessary for Image acquisition and segmentation process. However, it is difficult to access large clinical datasets due to several issues, like, providing cost, privacy, security and the availability
Introduction

of resources. Even after accessing to datasets, experts are needed to analyse and obtain the "Golden Ground Truth" for evaluation of segmentation algorithms performance manually, which is too time demanding. In addition, there are negative factors like noise, poor image quality and variant illumination which make it harder to extract the ground truth. In real life image segmentation of medical images is a critical and difficult task due noise, poor image quality, non-distinct edges, imaging artifacts and occlusions.

Identifying overlapping particles and counting the number of objects from images containing small cells is one of the main challenging problems for automatic microscopy analysis. Unfortunately, some issues as connecting or overlapping objects, which would present themselves as one rigid object in a 2D image, can increase the error of total cell counting. Moreover, factors like noisy environment, low resolution photos and different illumination levels should be considered in real world.

A varying number of images are synthesized by a virtual camera by changing its simulated focal length to focus on different depths in a volume. The effect of "depth of focus" is considered by blurring the image of each cell with respect to its distance from its specific position in depth. The synthesizing algorithm provides an opportunity to analysers to experience and consider different conditions and environment properties by varying different factors in the algorithm for generating different cell models. For example, volume dimension, out of focus effect and cell types can be set by the analyser.

The main objective of the Analysis part is to identify overlapping areas and their positions. A secondary goal is counting total number of cells. In this approach, we considered different cell densities, noise distributions and resolutions.

1.1 Problem Statement

This Master thesis project was originally proposed by a Company, which works in the health care area in handling white blood cell and urine test. However, the company was regrettably unable to provide the data required to verify the synthesis application. As a result, we have formulated the project according to the company's description which is to enhance their current algorithm and/or to develop a new method for finding overlapping positions
of cells, separating overlapping objects (cells) and counting total number of cells within an entire volume of data by considering external factors like noise, resolution, illumination and so on.

Contribution

We developed a synthesis application that can produce Multi-Layered Microscopic Images containing different cell types. This synthesis application allows users to set different parameters with respect to their need for synthesized images. The Adjustable parameters in our synthesized model are: volume size, cell types, cell density, noise type, noise density, number of images and depth of focus effect.

An approach is proposed by parametric fitting function for deformable blob models to estimate a parameters of a cell which is not overlapping as well as its out of focus. Using single procedure we can extract different parameters of a single cell.

Different features from the volume were extracted and examined to choose a good feature subset for classification level. Besides, we have implemented a new bisection algorithm for bisecting overlapping cells which helps us to count total number of cells more accurately as mentioned in section 3.2.7.

We have compared the performance result among Linear Discriminant Analysis, Artificial Neural Network and state of art on images with different noise density and resolutions.

Finally, we have developed a test-bed application that is able to synthesize Microscopic Images and analyse them by allowing users to set different parameters in synthesizing level as well in choosing the method of analysis.
Introduction
Chapter 2

Background

Developing a robust algorithm for identifying and separating overlapping particles in microscopic images is one of the main concerns and research work for automatic microscopy analysis. Some of the researches for overlap separating approaches are based on prior particle shape and gray value intensity. For example, watershed\cite{2, 21} and mathematical morphology methods\cite{11, 25, 26}, which suffer from over segmentation and expensive of computational time.

According to\cite{7} they proposed a static template matching for a specific cell class knowing their prior knowledge of the cell shape and size. They construct a single cell template model to identify the overlapping particles. However, their approach could not be used for many cell classes which contain different single cell types as well as complex overlapping cells.

The approach for separating touching and overlapping objects by\cite{16} uses geometrical features as well as variation of intensity in touching and overlapping places. For noisy and poor resolution images the gray value feature will introduce an error in identifying overlapping objects.

Moreover, none of these mentioned algorithms are working on a sequence of images. Instead of fusing the information acquired from images of different depth they use a single image. We are not aware of any work in regular optical microscopy that has taken this novel approach.
2.1 Related Works

Confocal Microscopy

A closely connected, yet fundamentally different topic to our thesis work, is that of confocal microscopy. The term confocal refers to the condition where two lenses are arranged to focus on the same point or the final image has the same focus as or the focus corresponds to the point of focus in the object. When an object and its image are "confocal", the microscope is able to filter out the out-of-focus light from above and below of focus point towards to the object. Normally when an object is imaged in the fluorescence microscope, the produced image is from the full thickness of the specimen in a way that does not let most of it to be in focus for the observer. The confocal microscope eliminates this out-of-focus information by means of a confocal "pinhole" situated in front of the image plane which acts as a spatial filter and allows only the in-focus portion of light to be imaged. Light from above and below the plane of focus of the object is eliminated from the final image [1]. A diagram of the confocal diagram and principle is shown in figure 2.1.

![Figure 2.1: A Confocal Microscope hardware design and working principle](image)
The major optical difference between a conventional microscope and a confocal microscope is the presence of the confocal pinholes, which allow only light from the plane of focus to reach the detector. This forms the principle of a confocal microscope where "out of focus" light is removed from the image by the use of a suitably positioned pinhole. This produces images of exceptional resolution and clarity, and also allows the user to collect optical slices of the object to use in creating a 3D construction of the sample.

Using confocal microscope, it is possible to not only capture sharp images in various focal planes by scanning through the entire specimen, but also to eliminate scattered light from out-of-focus regions that can result in blurred images [15].

The confocal microscopy is a major advance upon normal light microscopy since it allows one to visualize not only deep into cells and tissues, but also create images in three dimensions [1]. There are many aspects of a confocal microscope that make it a much more versatile instrument than a conventional fluorescence microscope. Although confocal microscope is often thought of as an instrument that can create 3D images of live cells, the great versatility of this machine is not only having many creative ways for examining a structural details, but also the dynamics of cellular processes [1].

**Shape from Focus (SFF)**

When microscopic images are taken from an object by varying the depth of focus, it is possible to produce different blurring level on different regions. In order to solve this problem Image analysts use "shape from focus" algorithm for detecting sharp image regions and recovering of a three dimensional object [8].

The conventional shape-from-focus (SFF) [23, 19, 18, 29] use a sequence of images taken from a camera at different depth of focus to compute depth of the objects. Methods like Laplacian focus measure and gray-level variance focus measure are used for finding the best focused image slice in a specific location within the image volume.

In general, the conventional shape from focus approach works by calculating the contrast of neighborhood pixel for individual pixel location from a
sequence of images, where the maximum of the contrast value indicates to a scale (level) where the pixel is in focus and the scale relates to a distance near to an object point [8]. By applying those steps to every pixel points within the image sequences a sharp image will be obtained. Similarly, Gaussian interpolation approximation approach is used for computing a more accurate depth estimation by Nayar and Nakagawa [20].

Another better method than the conventional Shape from focus based on a new concept called focused image surface (FIS) was proposed by Subbarao and Choi [17], where FIS of an object is a surface containing a set of points at which the object points are focused from a camera. Furthermore, they are able to show a one-to-one relationship between a matching FIS and the object shape.

2.1.1 Related Classifier Algorithms

As we mentioned in section 3.2.5 we have used Linear Discriminant Analysis (LDA) as a simple classifier and Artificial Neural Networks (ANN) as a complex classifier in our work. Similarly, there are different kind of classifiers that are used currently in research fields.

Support Vector Machine (SVM)

Support Vector Machine builds a hyperplane that has the largest distance to the nearest training data point of any class. If the width of the discriminant boundary of the linear classifier could be widened; "Margin" is the maximum width which the boundary still does not intersect a data point. These nearest data points are called "support vectors". It is obvious that support vectors are the most important data samples in the training process and Support Vector Machine is supposed to find the linear classifier with the maximum margin.

Given a training set of l data samples \((x_i, y_i)\) while \(x_i\) contains n features and one target value \((y_i)\) equal to 1 or -1, the support vector machine requires the solution of the following optimization problem [6] to find \(W\) as shown in
\textbf{equation 2.1:}

\[\min_{(w,b,\epsilon)} \frac{1}{2} W^T W + C \sum_{i=1}^{l} \epsilon_i \]

\textbf{subject to}

\[y_i(W^T \phi(X_i + b) \geq 1 - \epsilon_i, \text{ for } \epsilon_i \geq 0 \]

where \(W\) is the classifier hyperplane normal vector and \(\epsilon_i\) is a slack variable added to allow miss-classification of difficult or noisy samples of the data \(X_i\), and parameter \(C\) is a variable for controlling over-fitting.

This linear classifier uses dot product between vectors \(K(x_i, x_j) = x_i^T x_j\). For non-linear classification, primarily every data point is mapped into a higher-dimensional space via some transformation \(\phi : x \rightarrow \phi(x)\), then dot product between vectors becomes: \(K(x_i, x_j) = \phi(x_i^T)\phi(x_j)\), which is a kernel function corresponding to an inner product in an expanded feature space.

There are different types of kernels to apply and the efficiency of SVM is related to the kernel type, the kernel’s parameters, and the soft margin parameter \(C\) which is a penalty term to control the over-fitting.

Scaling the data in the preprocessing level is essential before SVM applying and prevents the negative effect of greater numeric values of some features over ones with smaller values. For a classification problem with more than 2 classes, "one verses all" approach should be applied and that is, breaking down the multi-class problem to several binary classifications by separating each class from the rest and to assign the class corresponding to the highest classification function output to each data. In addition, there are many related works on image processing applying support vector machines [27, 28] and applying this classifier is suggested as a future work for this project.

\textbf{Random Forest}

The random forest classifier is made up of many decision trees in a way that each tree gives a classification "vote" to each observation and the classifier selects the class with the most votes [12].

To construct a tree in a situation where the number of training cases are \(N\) and the number of variables are \(M\), a constant number \(m \ll M\) variables is
Background

selected randomly for each node and the decision to split the node. Each
tree would grow to a maximum size as possible with no pruning.

This classifier is a precise classifier in many cases [24] as it works effectively
and fast on large data sets like our project and is a suitable choice for real
time problems [10] and can be considered as a future work for this project.
Chapter 3

Method

3.1 Synthesis

3.1.1 Multi-Layered Microscopy

In our model we have considered a 3D data which can contain different cells inside the volume with their specific x, y and z coordinates, where z is the DOF.

As we know in real world, most of the microscopic cells have a circular or ellipsoid like shapes. Hence, we have used a general ellipse equation in order to model the cells.

The standard equation form of an ellipse with centers at \((X_c,Y_c)\), where a and b are the lengths of the axes lying along the coordinate axes, is given by:

\[
a(x - X_c)^2 + b(y - Y_c)^2 = 1 \quad (3.1)
\]

In a more general way for an ellipse equation when its principal directions are along the x and y axes; if we apply a rotation to equation (3.1), we will have another general representation of an ellipse as follows:
Method

\[ r^T R_\theta^T \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} R_\theta r = 1 \]  \hspace{1cm} (3.2)

where \( r = \begin{pmatrix} x - X_c \\ y - Y_c \end{pmatrix} \) and \( R_\theta \) is the rotational matrix for the angle \( \theta \).

In our model we considered \( V \) different classes of cells having \( V \) different areas, and their area \( A \) is calculated from the major and minor axis as:

\[ A_i = \frac{\pi}{\sqrt{ab}}. \]

The area of a cell is independent of rotational direction \( \theta \) and roundness measure \( \beta \), where \( \beta \in (0,1] \) attains a value 1 for a circle and converges to 0 when the ellipse degenerates into a line structure.

An ellipse is given by \( A, \beta, \theta, X_c, Y_c \) and is conveniently expressed as:

\[ r^T R_\theta^T \begin{pmatrix} \beta^2 \frac{\pi}{A} & 0 \\ 0 & \beta^{-2} \frac{\pi}{A} \end{pmatrix} R_\theta r = 1 \]  \hspace{1cm} (3.3)

Points inside the ellipse are defined by:

\[ r^T R_\theta^T \begin{pmatrix} \beta^2 \frac{\pi}{A} & 0 \\ 0 & \beta^{-2} \frac{\pi}{A} \end{pmatrix} R_\theta r - 1 < 0 \]  \hspace{1cm} (3.4)

A distance function can thus be defined as:

\[ Dist(x, y) = r^T R_\theta^T \begin{pmatrix} \beta^2 \frac{\pi}{A} & 0 \\ 0 & \beta^{-2} \frac{\pi}{A} \end{pmatrix} R_\theta r - 1 \]  \hspace{1cm} (3.5)

which is a matrix representation of orbits with elliptical shape containing their pixel positions. All pixels located on one orbit perimeter have same gray value representing orbit distance to its boundary as shown in figure 3.1.

It is proven that no analytical expression exist for the true Euclidean distance to the boundary of an ellipse [9]. For high roundness distance is approximately equal to the orthogonal distance to the boundary, weighted with +1 outside the shape and -1 inside.
In order to limit the output of error function from 0 to 1 instead of -1 to 1, we have defined a sigmoid function which is an error function where its output is between 0 to 1 as follows:

\[
erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt
\]  \hspace{1cm} (3.6)

\[
\text{Sig}(x) = \frac{1 + erf(x)}{2}
\]  \hspace{1cm} (3.7)

where \(erf(x)\) is an error function which is twice the integral of the Gaussian distribution with 0 mean and variance of \(\frac{1}{2}\), and \(\text{Sig}(x)\) is a sigmoid function made from the error function.
Method

Hence, we used the sigmoid function to produce a cell at a specific location inside a volume as follows:

$$Cell(x, y) = Sig\left(\frac{-\text{Dist}(x, y)}{d}\right)$$ (3.8)

where $\text{Dist}(x, y)$ is the distance function containing all pixel positions of a cell as described in equation 3.5 and $d$ is the depth of focus parameter.

For low resolution settings especially, we get aliasing effects if we use equation (3.4) directly to generate cells. The cell function is used rather to generate cells in an anti-aliased way by using a sigmoidal function as shown in figure 3.2.

![Sigmoid function](image)

(a) with $d=1$

(b) with $d=\frac{1}{20}$

Figure 3.2: Figure for Sigmoid function

Aliasing happens when we are representing an image at a lower resolution as shown on bottom of figure 3.3. The visual distortions by jagged and sharp boundaries introduce error which is especially critical for low resolution images. Therefore, we used the sigmoid function with two parameters for anti-aliasing effect to make the gray value variation on boundaries much smoother. The first parameter presents the distance of ellipsoidal shapes from the cell center as shown in figure 3.1 to make a smooth change of the cell pixels gray values from the maximum in the cell center towards the boundaries. The second parameter, which is called "d" in this approach, determines the rate of gray value decreasing outward the cell center and consequently defines the sharpness of cell boundaries as shown on top of figure 3.3.
3.1.2 Depth of Focus

By considering the depth of focus (DOF) [22], we simulate a camera that captures N stack images at different depths, form the synthesized 3D data as shown in figure 3.4. For each captured images, the camera focuses on one hypothetical plane perpendicular to Z-axis, which is selected by camera focal length.

Cells which are placed in depths near to a specific DOF plane looks clear and have higher definition in the corresponding image. However, they are still visible in a lower definition quality inside images made by further focal lengths. Actually, blurring effect is stronger for a cell when camera focal point locates further from the cell position. To achieve the out of focus effect in images as shown in figure 3.5, we have used Gaussian low pass filter for blurring effect simulation.

During synthesising we have introduced external factors like noise and low resolution according to the project requirement for simulating real data. The figures on 3.6 and 3.7 presents one image captured with a specific DOF affected by Gaussian noise and Pepper & Salt noise respectively.
Figure 3.4: A 3D volume containing different slice of images, where the blue and red arrows are focusing to a specific single cells at different DOF (focal length).
Figure 3.5: A Synthesized Microscopic Pure Data
Figure 3.6: Pure Data with Gaussian Noise
We do not use the "d" parameter in sigmoid function to introduce the out of focus effect, instead we apply a Gaussian filter. However, we have used it for tuning the sharpness of cell boundaries and for anti-aliasing effect as shown in figure 3.9.

Gaussian filter is a low-pass blurring filter which passes low frequency while reduce high frequencies. Due to the high intensity change of pixel values around edges in spatial domain, which has high frequency behaviour in the frequency domain, boundaries look smoother after blurring.
Method

\[ G(x, y; \sigma) = \frac{1}{2\pi\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \] (3.9)

The function \( G \) is a two dimensional (2D) Gaussian filter as shown in figure 3.8(c) and 3.8(d) with center positioned in the origin, where \( x \) and \( y \) are the distances from the origin and \( \sigma^2 \) is the variance for the Gaussian filter that defines the blurring amount. The variance is linearly dependent on the difference between the cell depth \((Z_c)\) and the DOF \((Z_d)\) as shown in equation 3.11.

\[ Z_d = \frac{[1 + (Z_c - 1)(D - 1)]}{N} \] (3.10)

\[ \sigma^2 = K|Z_c - Z_d| \] (3.11)

where, \( K \) is a constant factor, \( Z_c \) is cell depth, \( Z_d \) is the DOF offset, \( N \) is the total number of images captured and \( D \) is the depth of the 3D volume in \( Z \) direction.

We have used a Gaussian Filter as a scale-spaced filtering \[13\] to produce a smoothed stack of images at different depth of focus in \( Z \) direction. The advantage of using the Gaussian Filter is that we can produce smoothed images without introducing any new structures as we go from one depth to another depth.

Separability is one of the important properties of Gaussian filter \((G)_{m \times n}\). A 2D filter kernel is separable if it can be obtained from the outer product of two one dimensional (1D) filters \( G_x \) and \( G_y \) as shown in figure 3.8(a) and 3.8(b) respectively. The outer product of two vectors is equal to the two dimensional convolution of those two vectors as

\[ G(x, y; \sigma) = G(x; \sigma) * G(y; \sigma) = G(x; \sigma) \times G(y; \sigma) \] (3.12)

According separability property, convolution of two 1D Gaussian filters can be used instead of applying a 2D Gaussian filter \( G \) in order to make filtering much faster. Convolving with a 2D filter needs \((x \times y)\) multiplication for each pixel, however applying 1D filters separately decrease the multiplications.
needed for each pixel to $(m + n)$, as a result the computational time will be reduced by a factor $\frac{(x \times y)}{(x+y)}$.

Figure 3.8: Gaussian filters with different standard deviations (SD)
3.1.3 Analogy with Confocal Microscopy

The working principle of confocal microscope is based on changing the level or plane at which the species sample is focused. With confocal microscopy, only the level which is in focus will be imaged while out-of-focused details will not appear in the image slices.

Such an image stack is a near 3D representation of the object. It could accurately be described as a 2-and-a-half-D representation (as there is still the issue of occlusion in the data). This near 3D information is considered as one of the main features and a major advantage of confocal microscopy over conventional optical microscopy [1].

In our project we have synthesized a sequence of microscopic images by varying the DOF to an object and scanning sequential X-Y planes by changing depth in the Z-direction along a 3D volume data.

We have used a Gaussian filter for out of focus blurring cells with respect to their position inside the 3D volume. The sum of all the Gaussian filter coefficients are always equal to one (independent of the $\sigma$ used). This is an expression of the principle:

objects blur as they go out of focus, but the intensity contribution of objects never disappear.

With confocal microscopy, however the principle is essentially different, as:

objects blur as they go out of focus, and the intensity contribution of objects decrease rapidly.

We introduced a simple change in our synthesis application for the out-of-focus blurring filters to accommodate this. Apart from changing the $\sigma$, we optionally also scale the entire filter so that the sum of coefficients go to zero as the z-distance to point of focus increase. With confocal microscopy only one of two occluding cells are visible at the same time. Only part of the back cell will be visible when focusing on it yet the problem of cell overlap is non-existent (area estimation of the cell would be off however). Synthesizing confocal data is entirely optional to do and the rest of thesis is focused not on confocal microscopy but on the regular optical version.
3.2 Analysis

3.2.1 Parametric Description of DOF

We did not use the sigmoid function we mentioned in section 3.1.1 to render out of focus blurring in images. The main drawback with this approach is that it cannot create an equal blurring in all directions as shown on left side of figure 3.9. It also does not provide as efficient an algorithm in terms of speed as does a simple Gaussian convolution. However, the sigmoid function provides us with a simple analytical function \( f \) that incorporates DOF offset \( d \), area of a cell \( A \), circularity \( \beta \) and orientation of a cell \( \theta \) as follows:

\[
f(\bar{X}, d, \beta, \theta, A) = \text{sig}(\frac{-\text{Dist}(x, y)}{d})
\]

(3.13)

where \( \bar{X} \) is a vector of pixel position in xy plane.

We emphasize that the parametric fitting function in equation 3.13 will not accurately describe ellipses with \( \beta<1 \), but it provides reasonable approximation for \( \beta \) close to 1. It is good to have a parametric fitting model for estimating a real image as it contains more parameters like depth of focus, circularity, area, orientation and position of the real data. Those parameters of the real cell can be approximated easily when the residual error of the real cell and sigmoid model cell is lowest at a specific position of an image. Similarly, we can use the Gaussian blurring function for estimating the parameters of a real image. However, it will be computationally expensive and requires to do blurring of a cell iteratively until we estimate the depth of focus parameter.

For clear understanding the difference between sigmoid blurring and Gaussian blurring we have shown in figure 3.9. The left side of the figure are made by sigmoidal blurring, while the the right side of the figure are made by Gaussian blurring using different \( \beta, d \) and \( \sigma \) values.
Method

(a) sigmoidal blurring using $\beta=0.5$ and $d=0.95$

(b) Gaussian blurring using $\beta=0.5$ and $\sigma=47$

(c) sigmoidal blurring using $\beta=0.9$ and $d=0.95$

(d) Gaussian blurring using $\beta=0.9$ and $\sigma=47$

(e) sigmoidal blurring using $\beta=0.5$ and $d=0.4$

(f) Gaussian blurring using $\beta=0.5$ and $\sigma=20$

(g) sigmoidal blurring using $\beta=0.9$ and $d=0.4$

(h) Gaussian blurring using $\beta=0.9$ and $\sigma=20$

Figure 3.9: Comparing sigmoidal blurring with Gaussian blurring, where $\sigma$ is a Gaussian filter smoothing factor and $\sigma=50d$
3.2.2 Feature Extraction

In the synthesizing stage a stack of $N$ images with $a \times b$ size, where $a$ is length and $b$ is width of the 2D image, is produced as shown in figure 3.10. Regarding classification, each vector has $N$ degrees of freedom from depth of focus. Each image position has one vector associated with it. There are three classes defined as background, single and overlapping. To assign a vector to the overlapping class, it should contain more than one cell intersecting that vector. According to the DOF concept, each pixel value in an image does not necessarily belong to a cell on XY plane with the same coordinates in the volume due to the blurring effect, which causes one of the main difficulties in analysis of the overlapping particles.

![Figure 3.10: A 3D Volume Data](image)

Due to our multi depth of focus images issue, the analysis of the gray value variation in different depths in three directions was considered as the main source of information for the classification problem. To extract features from a specific position, we consider the whole range of $z$ variation. This defines a $3 \times 3 \times N$ Rectangular Cuboid (taking into account also the neighbourhood of each position). Different types of features were extracted and imported to classifier for comparing their classification errors to choose the best features subset.

We considered $a \times b$ pixels in $N$ different depths as $N \times a \times b$ voxels, and
Method

convolve a three dimensional sobel filters $G_x$, $G_y$ and $G_z$ as shown in equation 3.14, 3.15 and 3.16 along $x$, $y$ and $z$ directions respectively to them which assigns a value to each voxel corresponding to the center of the filter. This value represents the gradient of the intensity in each voxel with respect to its 26 neighbouring voxels providing the direction and ratio of the change to darker area. We tried to find some information in each voxel considering its neighbour. Then other features were tried to extract by focusing on the center of the local volume. At this point, it was tried to focus on the center of the Rectangular Cuboid and variation of the gray value along the $z$ dimension. Using a $\times b$ vectors containing $N$ pixels having the same $XY$ coordinate with different position in $Z$ direction, we have calculated one dimensional derivation, magnitude of Discrete Fourier Transform and central moments for each vector respectively.

$$
G_x(:, :, -1) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad G_x(:, :, 0) = \begin{bmatrix} 2 & 4 & 2 \\ 0 & 0 & 0 \\ -2 & -4 & -2 \end{bmatrix} \quad G_x(:, :, 1) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}
$$

(3.14)

$$
G_y(:, :, -1) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y(:, :, 0) = \begin{bmatrix} 2 & 0 & -2 \\ 4 & 0 & -4 \\ 2 & 0 & -2 \end{bmatrix} \quad G_y(:, :, 1) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}
$$

(3.15)

$$
G_z(:, :, -1) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad G_z(:, :, 0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad G_z(:, :, 1) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{bmatrix}
$$

(3.16)

The reason for examining the magnitude of Fourier coefficients $|x_n|$ as shown in equation 3.17 is due to the property that a shift of the signal is described in the phase only. Thus, the magnitude should be expected to be less sensitive to the relative placement of the cells in $z$ direction. The symmetric positioning of the cells relative to a $XY$ plane located in the middle of the volume is considered the same in our algorithm; for example, two cells overlapping in
a way that the $N^{th}$ depth of focus image was located between them have the same scenario as if they place with the same distance of the $1^{st}$ image.

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i2\pi \frac{k}{N}}$$

(3.17)

where $N$ is the number images taken at $N$ different DOF and $X_k$ is the gray value in each depth.

The shift in the location of overlapping appears in the phase of the DFT coefficients values and was ignored by considering only the magnitude of DFT coefficients. According to real gray values and hermitian symmetry property, the first $\frac{N}{2}$ Fourier coefficient magnitudes are chosen.

So, in general for $N=15$ images we have used $N$ features from 3D sobel filter, $N-1$ features from 1D derivation by convolving each vector with a $[1\ -1]$ kernel, 4 features from central moments as shown in equation 3.18 ($2^{nd}$ to $5^{th}$ terms, where the $0^{th}$ moment is always equal to 1 and the $1^{st}$ moment is the average gray value which is already in the DFT component), $\frac{N}{2}$ features from magnitude of DFT.

$$\mu_k = E[(X - E[X])^k]$$

(3.18)

where $E$ is the expectation operator, $X$ is a vector of gray values of one pixel in different depths of focus and $\mu$ is the central moment for $k=2, 3, 4$ and $5$.

### 3.2.3 Feature Selection

Feature or variable selection is the process of choosing a subset of the existing data features which are more helpful and robust in the relevant learning model, through decreasing the size of the data by removing the non-relevant features which do not contain valuable information which leads to the decrease of classification error and a faster process. We expected features related to the high frequencies to be more vulnerable to noise. We used backward feature elimination for feature selection. Backward feature elimination algorithm starts by considering all M features and calculating the classification error. In the next step, the M error values of the model is calculated by removing of the features one by one. The least error which was caused
Method

by removing a specific feature shows that the corresponding feature is the worst one and would be eliminated. Afterwards the same process is applied on the remaining M-1 features until we get chosen number of features or the desired classification accuracy. In our case we focus on the classification error and continued the iterations until there was a considerable classification error jump.

3.2.4 Normalization

In a data preprocessing level, we usually make sure features are almost on a similar scale. By combining different features to the same scale, we are also allowing the comparison of different features. The inputs which are not related at all in scale, decrease the precision of our model. Furthermore, it is easier to analyse the inputs when they are in the same magnitude range. One of the typical problems is in algorithms using gradient descent to find the minimum of cost function. In practice, the contours of the cost function shape causes the gradient descent takes a longer time to get to global minimum. In contrast, after the normalization gradient descent converges faster.

During normalization, we calculated the mean normalization by subtracting each feature vector from its own average and changed the average of all features to zero. And the mean normalization part is necessary for applying feature reduction methods, such as Principal Component Analysis. Besides, the mean normalization value is divided by the variance of the data for negating the effect of scales in different features.

3.2.5 Classification

Classification is the problem of detecting which of a set of categories a new observation belongs based on a training data set which contains observations whose category belonging is already known. The main goal for the classifier is to provide a $\times b$ image size for a new volume presenting the position of each cell in XY plane regardless of their depth in Z axis and detect places where true cells overlapping happens. To be more precise, there are three categories for the output of the classifier, overlapping, single and background.
Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) which was firstly tried in our project, is a linear classifier identifying which of the three categories a new observation belongs to, based on training data set with known class labels. Linear classifying, suggests that the classes can be separated by a linear combination of the observations attributes. The classification goal is to assign an observation to the category with highest conditional probability. In a K class problem it can be assigned the observation $x$ to the $C_k$ where $P(C_k|x) > P(C_j|x), \forall j \neq k$.

In practice, finding a posteriori probability $P(C_k|x)$ for category $C_k$ is difficult to obtain but $P(x|C_k)$ the probability of getting a particular observation $x$ given that the observation comes from group $C_k$ is more easily computable. Therefore the Bayes theorem is applied to relate these two conditional probability:

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{\sum_{l=1}^{K} P(x|C_l)P(C_l)} \tag{3.19}$$

The prior probability $P(C_k)$ of class $k$ could be expressed as:

$$P(C_k) = \frac{t}{T} \tag{3.20}$$

where $t$ is number of training dataset samples in $C_k$ and $T$ is total number of training dataset samples.

Maximizing the aposteriori probability is equivalent to maximizing the nominator of the right side of equation 3.19 and we can have:

$$\hat{C}(x) = arg\max_k P(C_k|x) = arg\max_k P(x|C_k)P(C_k) \tag{3.21}$$

In linear Discriminant Analysis we assume that the X density, given its category follows a Gaussian distribution and it can be written as follows:

$$P(x|C_k) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T (\Sigma_k^{-1}x - \mu_k)} \tag{3.22}$$

where $p$ is the dimension and $\Sigma_k$ is the covariance matrix.
Method

In LDA it is assumed that the covariance matrix is identical for different classes which linearize the problem. After calculating the mean vector for class $k$ by averaging all feature vectors belonging to that class, the covariance matrix for three classes $K=3$ will be calculated using equation 3.23.

$$
\hat{\Sigma} = \sum_{k=1}^{K} \Sigma_{c_k} \frac{(x^i - \hat{\mu}_k)((x^i - \hat{\mu}_k)^T)}{N-K}
$$

where $\mu_k$ is the average feature vector for class $k$ and $N$ is the total number of data in training data set. By combining equation 3.21, equation 3.22 and equation 3.23 we will have another expression as follows:

$$
\hat{C}(x) = \arg\max_k P(C_k|x) = \arg\max_k P(x|C_k)P(C_k)
$$

$$
= \arg\max_k \log(P(x|C_k)P(C_k))
$$

$$
\hat{C}(x) = \arg\max_k [x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(P(c_k))]
$$

Equation 3.24 contains the the expression for linear discriminant function which is given by

$$
\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(P(c_k))
$$

Equation 3.25 and each new observation was classified in a category which made linear discriminant function expression maximized.

LDA is simpler and less accurate but much faster than the other classifier used in this project, Artificial Neural Networks(ANN). LDA was usually used as a benchmark for the classification problem in our project. Moreover, according to its computational time, it is a good tool to analyse different features or selecting a subset of features which are more suitable for the classification, which is described in feature selection section. However, the classification result by LDA as it is presented in the result section was acceptable to be used as a less accurate reference for other complex classifiers which can be used in our project.
Artificial Neural Networks

Artificial Neural Networks is an algorithm that initially motivated to have machines with the ability to simulate the human brain that deals with high training time and working with large data sets. Recently due high speed of new computers, neural networks are the state of art technique for many applications as Signal processing, Control Systems, Pattern Recognition, Financial applications, Game playing, and so on.

In artificial neural network, a simple presentation of a neuron (node) can be shown as a logistic unit where the name of the logistic unit comes from its sigmoid(logistic) activation function "F". Let there be n+1 input signals varying from $x_0$ to $x_n$ for the neuron and n+1 weights varying from $w_0$ to $w_n$ respectively; where the input signal $x_0$ is assigned to the value +1 and "$w_0 = bias" produces the bias term for the node as shown in figure 3.11 for one single artificial neuron.

The artificial neurons output is the summation of the multiplication of each input signal by a weight on its way to the node and the activation function "F" converts the sum of the weighted inputs to the neuron output.

Figure 3.11: one single artificial neuron
Method

The neural network is made by the combination of these nodes which themselves are positioned in a layer form, as the first layer is the input layer and the last layer is the output layer and if there are layers between them the input layer and the output layer called hidden layers. Different types of ANN are modified by selecting the layers interconnection architecture, the learning process for updating weights and the activation function.

Feedforward Neural Network is used in this project which is one of the common neural network architectures and is applied to many different tasks as Machine Vision, Signal Analysis, Data Mining, Data Compression, Robust Pattern Detection, Data Segmentation, Text recognition, Adaptive Control, Optimization and so on. In this architecture all nodes are connected forward to the next layer and there is no connection from a layer with a backward direction to its previous layers and consequently there are no cycles in this architecture. Here we showed in figure 3.12 a simple feed forward network with five nodes in input layer, one hidden layer with four nodes and two nodes in output layer is presented.

![Figure 3.12: A simple feed forward network with one hidden layer](image.png)
Feedforward Neural Network is an example of supervised learning. That is, the correct answers of some observations, called the train data, are valid for the system. The weights in the network are initialized randomly as they get updated each iteration by the information provided through comparing the network outputs and the correct answers. Therefore the derivative of the error with respect to the network weights is calculated each iteration and would be used in the gradient descent algorithm. This procedure continues until the system converges to a state with a small error by finding minima. Due to the gradient descent in our algorithm, the sigmoid activation function’s continuous derivative gives us the possibility to use it in the backpropagation.

The other important decision in designing the network is the best number of hidden layers and the nodes of them which can play roles in the classification error of the model. Conveniently, one hidden layer is considered for the model. Too few neurons layers will result in under fitting (high bias system) while too many neurons in the hidden layers can result in over fitting and high computational time. To avoid this problem the numbers of the hidden nodes in our model is decided by increasing them until the classification error decreases. The number of the input layer node is equal to the number of features in addition to, one node for the bias. And the number of nodes in the output layer is 3 owing to the 3 output classes.

Over fitting in the training level happens when the system gets complex (more than enough) and considers outliers in the training data dramatically in respect to the underlying function. In addition to the number of hidden nodes mentioned above one of the reasons for this issue is high number of iterations in the model training. To avoid this problem "early stopping" is used which ensures that the classification error for new observations which are not in the data set is acceptable. To do this in our project, our training set is split into new training set and validation set and the training process is done on the train data set and the error is calculated from the validation set. Training iterations stop when there is no improvement in the classifying the validation data.

3.2.6 Watershed

Watershed Transformation is a segmentation algorithm in image processing and it was firstly introduced by [4] for segmentation problems. In contrast
Method

with segmentation algorithms which find the local variations and boundaries, it mostly works by finding similar regions.

To present the idea, it is better to see a grey value image as a topographic surface; accordingly, the lighter pixels in image having larger gray value have higher altitudes and the darker pixels having smaller gray value have shorter altitudes in the surface. Now, we can visualize regional maximum pixels which are lighter from their surroundings as peaks and regional minimum pixels which are darker from their surroundings as the sinks. Accurately, a regional (local) minimum is a point if there is no ascending path in the surface starting from that point to its neighbourhood. In the same way, if there is no descending path in the surface starting from that point to its neighbourhood established for regional maximums. Catchment basins and watershed lines are two important terms in defining watershed area by flooding process. When we fill the holes (catchment basins) of the surface by water, the pixels on which dams can be made to avoid merging the adjacent catchment basins are watershed lines. Each catchment basin has its own local minima. For each pixel, if we imagine dropping of water on it, it will move toward to the nearest local minima in the topographic surface. To be precise, it goes toward the minimum located at the end of the steepest descent and we can say that the pixel belongs to the corresponding catchment basin of the local minima. A function $W$ is a watershed of function $F$ if and only if $W \leq F$ and $W$ preserves the contrast between the regional minima of $F$; where the contrast between two regional minima $M_1$ and $M_2$ is defined as the minimal altitude to which one must climb in order to go from $M_1$ to $M_2$ [3].

The main information for image segmentation using watershed transformation is from analysing the gradient magnitude of an image as a topographic surface. Watershed Transformation usually suffers from over segmentation. To overcome this problem, [5] proposed marker control segmentation which uses prior knowledge about objects locations from other sources for segmentation and the gradient image is modified in a way to keep only the most significant contours in the areas of interest between the markers.

In our project we have used watershed transformation for counting total number of cells to compare with our approach. By taking in to consideration that none of the images from different depths of the volume contains all information about the volume due to the DOF , noise and low resolution. We have used the ground truth data as an input to watershed transformation,
instead of the processed (output) data from neural classifier, which we used as an input in our algorithm. However, the counting result from watershed suffers from over segmentation which makes it worse than the result from our algorithm. In the first step, to make catchment basins for watershed algorithm, we found the distance transformation of the negative of ground truth image compliment. In other words, we find the distance of every pixel to the nearest nonzero-valued pixel, then after watershed algorithm is applied to it.

3.2.7 Cell counting and overlap-bisection

We have used the neural network output, overlapping areas (blobs) from a volume filled with defined density of cells, to generate a vector of overlapping area centroids in order to use them for bisecting overlapping cells. By marking all centroid points of overlapping areas of the overlapping cells, we identified the nearest background pixel with respect to the centroid of overlapping cell using Euclidean distance. The line crossing this pixel and the corresponding centroid point is considered as the bisecting line for those overlapping cells as shown in figure 3.13.

As, we mentioned in section 3.1.2 our model considers a noisy environment and images with different resolutions. It is likely to miss a few overlapping areas which leads to assign those overlapping cells as single cells.

To overcome such problems, we have also used some geometrical properties as additional features for the final step of cells counting. Experimentally, we have found that circularity, area and extent features of an object are very important and can be easily used to distinguish single cells from those overlapping cells which are left not bisected from the previous phase. Additionally, by comparing the output of our classifier model and the ground truth, some morphological was applied to the 2D output image to fairly compensate the classification error. Finally, by fusing the result from the bisecting cell method and the geometric features extraction method, we are able to count the total number of cells with in a volume filled with cells.
(a) overlapping cells from classifier output

(b) bisection line in an overlap cell

Figure 3.13: Identifying overlap and drawing bisection line

Bisecting Overlapping Cells

To be precise, two cells are overlapping if their mappings to xy plane capture same points in some parts and these overlapping cells are located in different
positions in z coordinates then bisecting the two overlapping cells gives the opportunity to count the cells in the volume correctly. In our algorithm, we use the classifier output to have a more precise bisection which leads to also a better cell counting.

In this approach, the zero moment of all pixels belonging to overlapping areas found by the classifier output, is calculated. The overlapping area between two cells is assumed to be symmetric when a line passes through the centre of the overlapping area. Although, there are an infinite lines that can cross the overlapping area centroid, we have found an appropriate line that bisects the overlapping area.

To draw the bisecting line, finding a second points was necessary which let us to draw a unique line, and by considering the ellipsoid shapes of the cells and analysing different cases of two cells with different sizes and different orientations overlapping, the nearest background cell in the output image to the center is selected as the second point in this approach as shown in figure 3.13(a) and figure 3.13(b). To present the bisecting better and cutting the overlapping cell to two single objects, the pixels on the overlapping area which are located on the bisecting line were assigned to the background. At this point, the number of cells can be easily counted by counting the separated objects in the output image as shown in figure 3.14.

![Overlap cell after bisection](image)

**Figure 3.14: Overlap cell after bisection**
3.2.8 Flowcharts for Synthesis and Analysis

In addition to the explanation for the method we used so far, we have included a flow diagrams that describes the entire process of both Synthesis and Analysis parts as shown in figure 3.15 for synthesis section, figure 3.16 and figure 3.17 for analysis section respectively.

Figure 3.15: Flow diagram for synthesising microscopic image, the synthesis part includes generating of cells within a volume, blurring of each cell w.r.t DOF and cell depth, adding external factors and preparing of Golden Ground Truth data.
Figure 3.16: Flow diagram for measuring overlapping performance, we used the classifier output to identify overlapping positions of cells using Dice index method and overlapping pixel areas.
Method

Figure 3.17: Flow diagram for counting number of cells, for cell counting stage we have compared the total number of cells using our method and watershed segmentation result.
Chapter 4

Experiments and Results

As we mentioned in section 1.1 we had originally planned to test on real data, provided by a company in the health care industry. However, their data and current algorithm was not provided for us. Therefore, we have assumed their algorithm uses the average of binary images from 15 different depths. Binary images are prepared from 15 original images by putting a threshold with a constant value.

We have used a Venn diagram for illustrating the overlapping positions using disc as shown in figure 4.1(a), where B and G are discs for our model output and ground truth respectively and $(B \cap G)$ is their intersection. For measuring the performance of different algorithms in finding the place where overlapping of particles occurs, we have used the Dice coefficient [14] which is mostly used for measuring spatial overlaps in binary images. The Dice coefficient $D$ is given by $D = \frac{2|B \cap G|}{|B| + |G|} \times 100$ and the values for the coefficients varies from 0 (non overlap) and 1 (perfectly overlapped).

A disc is placed on each overlapping area of model output and ground truth in a concentric manner as shown in figure 4.1(b), where the center of each overlapping area is computed by averaging the pixels coordinates belonging to that area. The disc radius $r$ introduces a bias for displacement of an overlapping area in model output comparing with ground truth which can still be approved. In other words, distance more than $2r$ between one specific overlapping area center in the output with respect to the ground truth is considered as 100% error. This method is used for defining the performance.
Experiments and Results

(a) disc using Venn diagram

(b) overlapping particles using disc

Figure 4.1: Illustration of Dice and overlapping pixel areas
of "overlapping positions" detection based on their centres regardless their size.

Due to focusing only on center of overlapping blobs regardless their size, the "disc" approach is sensitive to few so tiny overlapping blobs that can occur in the model output which are not in the ground truth. Similarly, another approach called "Area" is applied by means of whole pixels inside an overlapping blob instead of its corresponding disc. This approach analyses precisely each pixel in the output and does not consider the number of overlapping blobs as the former one.

4.1 Experiment 1 (overlap detection by state of art algorithm)

For Experiment 1 we used averaged image, obtained from 15 intermediate binary images with different DOF as an input. External factors like noise and resolution are considered. The performance measurement is done by using the disc and area approach.

The overlapping places are found by averaging all binary images and assigning the pixel with gray value greater than the threshold to overlapping class. It should be mentioned that the previously used threshold was chosen in a manner that it could able to find almost all the overlapping areas in non-noisy environment to make a fare enough comparison.

4.2 Experiment 2 (overlap detection by our algorithm)

For experiment 2, we modelled 2 classifiers to find the output for new volumes. Firstly, we did feature backward elimination on our data set and we conclude that the features extracted by 1D derivation and moments are not improving the classification accuracy. The selected features are the first 8 DFT coefficients magnitudes. As the number of input vectors belonging the overlapping class is fewer than the remaining classes, the number of the
training samples for background and single classes were chosen equally to the overlapping class. The whole data set for a volume contains 1 million samples. For a volume with 150 cells and 15 images, the whole train data set contains approximately 11000 samples. The number of the hidden nodes for the Artificial Neural Network was determined 12 nodes, by finding the least classification error.

The number of the hidden nodes for the Artificial Neural Network was assigned to 12 nodes by finding the least classification error. It was observed that ANN finds almost all the overlapping areas. However, it was noticed that the overlapping blobs were slightly expanded and sometimes few wrong tiny overlapping blobs were occurred as shown in figure 4.4. These tiny blobs does not make a large negative effect on finding the output accuracy considering the whole area of overlapping blobs, however it can considerably decrease the output accuracy calculated by disc method. For compensating this negative effect, the output binary image of the overlapping pixels was eroded by a structuring element of size which reduces the size of the expanded blobs and removes the tiny blobs.

In order to compare the classification error using LDA and ANN with respect to the ground truth, we have showed three different binary images from Ground Truth, Linear Classifier and ANN Classifier in figure 4.2 to figure 4.4 respectively.

![Figure 4.2: Binary Image from Ground Truth](image)
Figure 4.3: Binary Image from Linear Classifier

Figure 4.4: Binary Image from Artificial Neural Network Classifier
4.3 Experiment 3 (counting cells by watershed and our algorithm)

We have compared our approach with watershed algorithm for counting the total number of cells inside a volume containing 150 cells as shown in figure 4.5. In our algorithm, counting process was done on the classifier output. The classifier output is a 2D image containing pixels with gray values 0, 1 and 2 which presents the assigned class for corresponding pixel in different depth as shown in figure 4.3 and figure 4.4.

Figure 4.5: Input Binary Image from classifier containing 150 cells

In contrast to our algorithm which has the capability to fuse information of different depth, watershed works with one image. Because none of the images contain all information from different depths and for a fair comparison
between our algorithm and watershed in cell counting level, ground truth is fed to watershed.

For a fair comparison between watershed and our algorithm, which has the capability to fuse information of different depth, the ground truth binary image was fed to watershed. Although, we have given this option to watershed, its performance is much less than our algorithm because it suffers too much from oversegmentation as shown in figure 4.6 for Low Resolution image and in figure 4.7(a) for detailed segmentation result.

The graphs from 4.13 up to 4.18 illustrate the performance of finding overlapping particles in a noisy environment and different resolution images, while the bar graph 4.19 illustrates the performance of counting total number of cells.
Figure 4.6: watershed segmentation result in Low resolution image
(a) watershed segmentation for overlap cell

(b) segmentation using our method for overlap cell

Figure 4.7: Comparing the segmentation result of our method with watershed
4.4 Result

We have used two approaches for performance measuring the accuracy of finding overlapping cells, using the disc and areas of overlapping pixels. According to our experiment we have tried to show different Performance measurement results for the state of art, Linear Discriminant analysis and Artificial Neural Networks in both noisy environment and images with different resolution. Moreover, for the total cell counting we have used a bar graph for comparing our approach (bisection method) with the watershed algorithm.

We have compared the performance of different cases for images containing different noise densities and resolution as shown from figure 4.8 up to figure 4.12. The performance using the state of art is around 70% in noise free environment with highest resolution images (i.e. 1000 × 1000 pixels). By using our approach and ANN we have improved it to 93%. We made 70 different volumes which contains 1106 overlapping places. The undetected overlaps using our approach was 18 with respect to the ground truth. Hence, in non-noisy environment 99.98% of overlapping blobs were detected regardless of their shape and displacement.

Similarly, we have compared cell counting performance between watershed and our method in volumes containing different number of cells (i.e. 100, 150, 200, 250 and 300). The cell counting performance, in noise free and high resolution images, using watershed varies from 20% to 25%, while our method performance varies from 95% to 99%. The performance of counting cells for watershed is calculated in a situation that watershed algorithm was fed with the ground truth 2D image, for getting an opportunity to compare the watershed result in a fair way.
Figure 4.8: Noise Free and Normal Image
Figure 4.9: Image with 0.1 Noise Density
Figure 4.10: Image with 0.4 Noise Density
Figure 4.11: Image with down sample factor 2
Figure 4.12: Image with down sample factor 4
4.4.1 Result 1 (overlap detection by state of art algorithm)

For the state of art we have used the average gray value of pixels along Z direction from the 3D volume data. However, the performance result in non-noisy and normal image size is 70% using disc measurement and around 67% using areas for overlapping pixels as shown in figure 4.13 and figure 4.14 respectively.

![Measurements using Dice Index](image1)

![Measurements using Area of Overlapping Pixels](image2)

Figure 4.13: Performance versus Noise Density using State of art, we have compared the output result of Median, Average and Wiener(Adaptive) Filters after removing the noise, where the highest performance is around 70% in noise free environment.
4.4.2 Result 2 (overlap detection by our algorithm)

Unlike the state of art, we have chosen the magnitude of DFT coefficients as best feature subset for both LDA and ANN from the 3D volume data. The performance result in non-noisy and normal image size for the LDA is around 78% using disc measurement and around 75% using areas for overlapping pixels as shown in figure 4.15 and figure 4.16 respectively. Similarly the performance result for the ANN on the same environment is around 93% using disc measurement and around 89% using areas for overlapping pixels as shown in figure 4.17 and figure 4.18 respectively.
Exp erimen ts and Results

Figure 4.15: Performance versus Noise Density using Linear network, we have compared the output result of Median , Average and Wiener(Adaptive) Filters after removing the noise, where the highest performance is around 78% in noise free environment.
4.4.3 Result 3 (counting cells by watershed and our algorithm)

In addition to the performance measure for overlapping cells, we have compared watershed and bisection (our method) algorithms for counting total number of cells within the 3D volume as shown in figure 4.19.
Experiments and Results

Figure 4.17: Performance versus Noise Density using Artificial Neural network, we have compared the output result of Median, Average and Wiener (Adaptive) Filters after removing the noise, where the highest performance is around 93% in noise free environment.
Figure 4.18: Performance versus Resolution using Artificial Neural network

Figure 4.19: Performance versus number of cells
Experiments and Results
Chapter 5

Conclusion and Future work

Among different features extracted for classification problem, the magnitude of DFT was found to be the best feature subset extracted from synthesized data for classification. We observed that fusing information from different depth in the presence of salt and pepper noise in images can compensate considerably negative effect of information loss by using noise removing filters. We have also noticed that fusing data from different depths can increase classification performance encountering low resolution images. This ability presents the value of multi-layered microscopy data to improve the problem for developers to improve their work accuracy.

During counting cells, it was seen that our bisection algorithm which uses the overlapping areas found by classifier model have a much better performance than watershed algorithm even in a case that ground truth was provided for watershed algorithm. Our approach is not suffering from over segmentation as watershed does.

The proposed test-bed application for synthesis and analysis of multi-layered microscopy data is a useful tool which gives the opportunity to user to get realistic looking images with annotated ground truth data. This application should be beneficial for developers during developing phase for quick viewing the effect of set different parameters provided. Adding more features and parameters like complete three dimensional cells, different cell body patterns and variant illumination in environment can be a nice work for providing more realistic model, including real data for comparison with the synthesised
Conclusion and Future work

data in future. Moreover, providing a graphical user interface in future would make it easier to work with the test-bed for users. Also many classifier models can be added to test-bed for analysis approaches.

Our analysis work has been mainly data driven. However, the model of the kind provided by equation 3.13 is essentially a deformable blob model. It can be used for the purpose of estimating the parameters of a cell that is not overlapping. It would also estimate its out of focus (parameter d). This would be done by fitting the model to positions in an image in a procedure which is essentially a local regression approach. The parameters of depth of focus, cell circularity, area and orientation and position could be estimated in one single procedure. The residual error of such a fitting would be smallest over the position of cells, where its value would tell us something of how well the data conforms to the assumption of a blurred ellipse. An interesting topic for future work would be to investigate the possibility of extending this to occurrences of multiple cells in the same locality.
Bibliography


BIBLIOGRAPHY


