Computer-Aided Analysis of Scanning Electron Microscopy Images with Illumination Inhomogeneity and Touching/Crossing Cells

by
Ali Memariani

A dissertation submitted to the Department of Computer Science,

College of Natural Sciences and Mathematics

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Computer Science

Chair of Committee: Ioannis A. Kakadiaris, Ph.D

Committee Member: Kevin Garey, Ph.D

Committee Member: Stephen Huang, Ph.D

Committee Member: Nikolaos V. Tsekos, Ph.D

University of Houston
August 2020

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisor Dr. Ioannis A. Kakadiaris for his insightful guidance and mentorship during my Ph.D study. I would also like to extend my deepest gratitude to Dr. Kevin Garey for his encouragement and support of my research. Furthermore, I am grateful to the rest of my Ph.D committee, Dr. Nikolaos V. Tsekos and Dr. Stephen Huang for their comments and questions. In addition, I would like to thank Dr. Bradley T. Endres and Dr. Long Chang for image acquisition; Dr. Bradley T. Endres and Dr. Eugenie Basseres for correcting the primary set of annotations and Dr. Jahangir Alam for providing the second set of annotations.

ABSTRACT

Clostridioides difficile infection (CDI) is a significant cause of death and morbidity due to infectious gastroenteritis in the USA. Treatments for CDI are being developed and comparison of the treatments is of paramount importance. Conventional microbiology methods investigate the effectiveness of treatments on the macro-level, and a phenotypic investigation has not been performed. Phenotypic features (e.g., length, shape deformation) of CDI cells in scanning electron microscopy (SEM) images indicate critical information about cell health in CDI research studies. However, analysis of SEM images is challenging due to the following issues: (1) inhomogeneous illumination, which causes shadows on the cells and bright areas around the cells, and (2) the presence of touching and crossing cells. Therefore, there is an urgent critical need to develop methods for the segmentation of the CDI cells to extract phenotypic information. This work presents a deep learning pipeline to provide instant-level segmentation of CDI cells in scanning electron microscopy images. The components are: (i) an adversarial region proposal network to compute cell candidate bounding boxes, and (ii) an instance-level segmentation network extracting features from bounding boxes, and computing the segmentation masks of isolated, touching, and crossing cells. The pipeline provides a computational tool for analysis of scanning electron microscopy images which is critical to compare the efficacy of CDI treatments. Finally, the performance is evaluated and compared to the state-of-the-art in instant-level object segmentation. The results indicate that the proposed computational tool out-performs the state-of-the-art method Mask-RCNN in detection (mean average precision) and segmentation (dice score) of CDI cells in SEM images.

TABLE OF CONTENTS

LIST OF TABLES

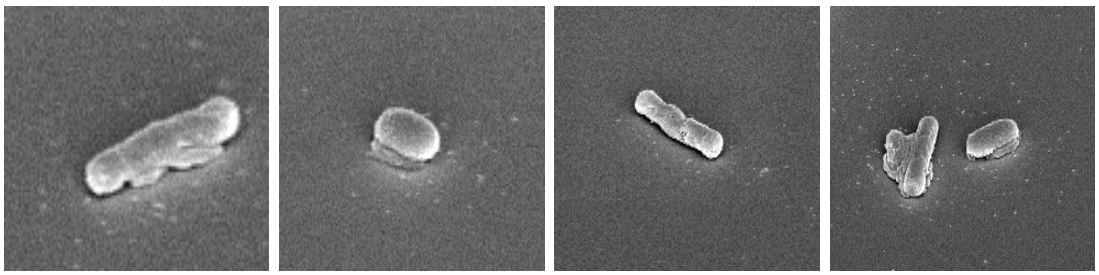
LIST OF FIGURES

1 Introduction

1.1 Motivation

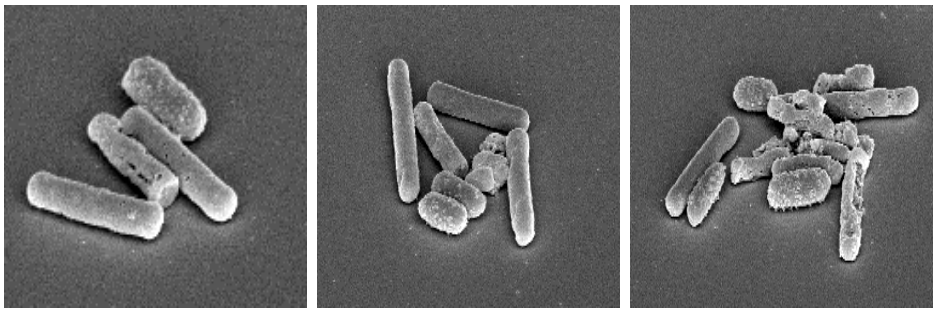
Developments in scanning electron microscopy (SEM) have facilitated the acquisition of digital images of micron-level cells, leading to improvements in cell quantification for pharmaceutical and medical research studies [16].

Clostridioides difficile infection (CDI) is the most common cause of death due to infectious gastroenteritis in the USA and a significant source of morbidity [14]. Extraction of cell-related information (e.g., length, location, deformation) in scanning electron microscopy (SEM) images is an essential task for comparison of treatments in CDI research studies [16]. However, analysis of SEM images is challenging due to inhomogeneous illumination and the presence of clustered cells. Existing computer vision methods have not considered the problem of instance-based segmentation with challenges above. Therefore, the automatic quantification of the efficacy of CDI treatments could not be addressed using the existing computer vision methods. This work addresses the problem of detection and segmentation of CDI cells in SEM images using a deep adversarial pipeline out-performing the state-of-the-art. Furthermore, the detection of cells in SEM images could potentially be used in the analysis of phenotypic information of other infectious diseases such as Coronaviruses, helping pharmaceutical researchers and society.



(a) (b) (c) (d)

Figure 1: Depiction of inhomogeneous illumination effecting SEM cell images, causing: (a,b) bright spots on the cell, and (c,d) shadows around the cells.



(a) (b) (c)

Figure 2: Depiction of samples of: (a) touching cell with clear cell walls, (b) with overlapping cell walls, and (c) Occluded cell bodies.

1.2 Challenges

Microscopic images may have inhomogeneous illumination and are often degraded due to noise. Inhomogeneous illumination causes bright spots on the cells as well as shadows around the cells. Figure 1 depicts samples of the challenges caused by inhomogeneous illumination.

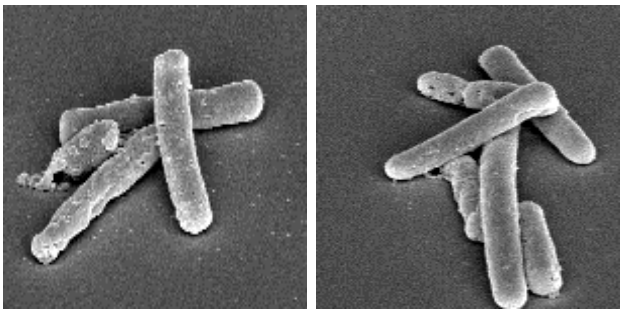


Figure 3: Depiction of samples of a crossing cell occluding one (L), or multiple cells (R).

Furthermore, the cells of various sizes are clustered together, making the problem of cell detection challenging. Clustered cells consist of touching and crossing cells. Cells may be touching with a clear boundary, or a portion of the cell walls may be overlapped. In some cases, a portion of the cell body is occluded. Figure 2 depicts samples of touching cells in SEM images. Moreover,

crossing cells cluster may occur where a cell crosses on top of one or multiple cells. Figure 3 depicts samples of crossing cells in SEM images.

1.3 Goal

The goal of this research is to develop a computational tool that analyzes scanning electron microscopy images of elongated cells, addressing challenges of inhomogeneous illumination, touching cells, and crossing cells, to provide instant-level segmentation of cells in the image.

1.4 Objectives

The objectives of this work are to:

1. Generate a dataset of synthesized SEM cell images with 6,000 images;
2. Develop and evaluate a pipeline for segmentation of cell images with inhomogeneous illumination;
3. Modify the pipeline to address the challenges of inhomogeneous illumination and separating touching cells;
4. Modify the pipeline to address the challenges of inhomogeneous illumination, separating touching cells, and separating crossing cells;
5. Evaluate the computational tools that provide specific cell-related information for all the cells present in the SEM image and compare the result to conventional manual annotation methods.
    - 5.1 Determine the correlation between manual cell counts of CDI compared to the computational tools developed in this dissertation, of determining the number of cells (vegetative and spores).
    - 5.2 Determine the proportion of spores vs vegetative cells using conventional microbiologic technique compared to the computational tools developed.

1.5 Contributions

Contributions to pharmaceutical research

This dissertation, provides a computational tool to automatically detect and count CDI cells in SEM images and computes the proportion of spores to vegetative cells. Furthermore, the computational tool provides a mask for every detected instance that could be used to extract phenotypic information which is important in comparing CDI treatments in pharmaceutical research.

Contributions to computer science

This dissertation, has the following computer science contributions:

1. An image synthesizer algorithm for background and cell (ISABC) is developed to provide the training data for a deep cell detection pipeline. ISBAC is capable of increasing the number of training samples in order of magnitude which is essential in training deep networks for the analysis of microscopy images which are expensive and time-consuming to obtain (Chapter 2).

2. A semantic segmentation pipeline, SoLiD (Segmentation of *clostridioides difficile* cells in the presence of inhomogeneous iLlumInation using a Deep adversarial network) is developed to separate CDI cells form the background in SEM images, addressing the challenge of inhomogeneous illumination where changes in illumination are modeled as an adversarial attack (Chapter 3).

3. The adversarial pipeline designed for semantic segmentation is extended for instance-based segmentation, providing a segmentation mask, a bounding box, and a class label for every cell in the image. The extended pipeline, DETCID (Detection of Elongated Touching Cells with Inhomogeneous illumination using a Deep adversarial network) is trained with synthesized images of isolated, touching, and crossing cells and 10-fold cross-validation is performed to measure the performance.

   To detect crossing cells, the ISABC algorithm is used to synthesize UH-S-cdiff1 where multiple pairs of touching and crossing cells are clustered together similar to the acquired images,

increasing the number of occluded samples. Furthermore, a modified IoU is developed based on the mask overlaps to detect touching cells with partial overlaps and crossing cells with occlusion. 10-fold cross-validation is performed over UH-S-cdiff1 to evaluate the performance on the synthetic images. Then, UH-S-cdiff1 is used to train DETCID and the trained model is evaluated on the acquired images (UH-A-cdiff1). A Bland-Altman analysis is performed to compare the error with the error between two human annotators (Chapter 4).

4. At the beginning of this work, the-state-of-the-art in cell detection was based on shallow methods. Therefore, two shallow methods (DeTEC and DETCIC) were developed before the deep pipeline was formed. The shallow methods were useful to identify the challenges of the dataset as well as providing insight on how to address them (Chapter 5).

5. Developed and evaluated a module to achieve rotational invariance via a parametric representation using convolutional capsules. Deep ConvNets have complex architectures and are difficult to interpret. A parametric representation is critical to interpreting the features used to detect the objects. A capsule-based representation is developed and evaluated to detect cells in various orientations (Chapter 6).

In this dissertation,, the developed methods are detailed for each objective and the performance of the methods is evaluated and compared with the state-of-the-art in cell detection and segmentation. Finally, conclusions and future work are discussed in Chapter 7.

1.6 Publications

Journal publications

1. **A. Memariani** and I. A. Kakadiaris, "Mask IoU Loss: Detection of Clustered Cells with Inhomogeneous Illumination", IEEE Transaction on Medical Imaging (In preparation).

2. **A. Memariani** and I. A. Kakadiaris, "DETCID: Detection of Elongated Touching Cells with Inhomogeneous Illumination Using a Deep Adversarial Network", IEEE Journal of Biomedical and Health Informatics (Under review 2020).

Conference publications

1. **A. Memariani**. B. T. Endres, E. Bassères, K. W. Garey, and I. A. Kakadiaris, "RISEC: Rotational Invariant Segmentation of Elongated Cells in SEM Images with Inhomogeneous Illumination"; In Proc. International Symposium of Visual Computing, Lake Tahoe, NV, USA, Oct 7–9, 2019.

2. **A. Memariani** and I. A. Kakadiaris, "SoLiD: Segmentation of clostridioides difficile cells in the presence of inhomogeneous iLlumInation using a Deep adversarial network"; In Proc. International Conference on Machine Learning in Medical Imaging a MICCAI Workshop, Spain, Sep 16, 2018, 285–293.

3. **A. Memariani**, C. Nikou, B. T. Endres, E. Bassères, K. W. Garey, I. A. Kakadiaris, DETCIC: "Detection of Elongated Touching Cells with Inhomogeneous Illumination Using a Stack of Conditional Random Fields"; In Proc. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Funchal, Spain, Jan 27–29, 2018, 574–580.

4. **A. Memariani**, C. Nikou, B. T. Endres, E. Bassères, K. W. Garey, I. A. Kakadiaris, "DeTEC: Detection of Touching Elongated Cells in SEM Images"; In Proc. International Symposium of Visual Computing, Las Vegas, NV, USA, Dec 12–14, 2016, 288–297.

Other publication

1. B. T. Endres, E. Bassères, **A. Memariani**, L. Chang, M. J. Alam, R. J. Vickers, I. A. Kakadiaris, K. W. Garey, "A Novel Method for Imaging the Pharmacological Effects of Antibiotic Treatment on Clostridium Difficile", Anaerobe, 40, 2016, 10–14.

2 Objective 1: Generate a dataset of synthesized SEM cell images

Deep networks require large numbers of training data. More specifically, deep learning architectures proposed for biomedical image analysis are hindered by the lack of large amounts of training data. Therefore, generating synthetic images becomes important in the analysis of biomedical images, since their acquisition is expensive and time-consuming. A simple solution would be to avoid training by applying a pre-trained model. However, the solution is limited to cases where a pre-trained network with similar training data exits [65]. To train a deep segmentation model, we developed an image synthesis algorithm capable of synthesizing cell images with inhomogeneous illumination where cells could be isolated, touching, or crossing.

2.1 Related work

Deep learning frameworks such as TensorFlow provide simple image augmentation functions such as translation, rotation, cropping, flipping, and scaling [1]. These basic augmentation techniques often create black areas in the image which could be filled with interpolation or more complex image in-painting techniques. The basic augmentation methods mentioned above have been applied to increase the number of training samples, and add invariance to challenges such as rotation of objects in the image [11, 32, 38, 42, 49, 65, 73, 81].

Increasing the amount of data by orders of magnitude is essential in training deep models to analyze biomedical images. However, using basic augmentation methods to increase the amount of data by orders of magnitude results in a high correlation between the images in the training dataset. U-net applied image warping to the cell images, creating images with slightly different cell and backgrounds [58]. Nevertheless, warping the whole image with the same warping transformation would limit the number of synthesized images.

2.2 Methods

This section presents ISABC, an Image Synthesizer Algorithm for Background and Cell, capable of synthesizing large numbers of images with the same background texture and cell shapes in

the images captured by SEM. The image quilting technique by Efros and Freeman is applied to synthesize similar background images [54]. Then, the cells are randomly warped and placed into the image. Warping the cells ensures that the training data are different from the testing data. Algorithm 1 depicts the steps of the algorithm and the notations used in Algorithm 1 are defined in Table 1.

2.2.1 Annotation

Two sets of manually annotated masks were collected using COCO-annotator [9], a cross-platform image annotation tool designed to create segmentation masks for instance-level object segmentation. The tool also allows us to specify disconnect masks to annotate the occluded objects. Coco-annotator provides JSON outputs with standard MS-COCO style annotations for masks and bounding boxes. In the frontend, the JavaScript GUI provides a polygon tool to trace the boundaries of the object and create the segmentation masks. In the backend, a python web framework communicates with the browser and a database management system based on MongoDB to store the segmentation masks.

Annotating biomedical images requires cross-platform tools to communicate between the biomedical expert and computer vision researchers. Docker provides a state-of-the-art virtualization platform using a single operating system kernel. Every software is packaged in a virtual environment called a container. The containers have independent configuration and libraries and may communicate via docker channels. A docker-engine hosts the containers and communicates with the operating system so that all containers share the same kernel. Therefore, docker provides a more lightweight platform compared to conventional virtual machines. A docker user group may be defined to allow the user to launch their containers without administrative privileges that restrict the software installations on the same machine.

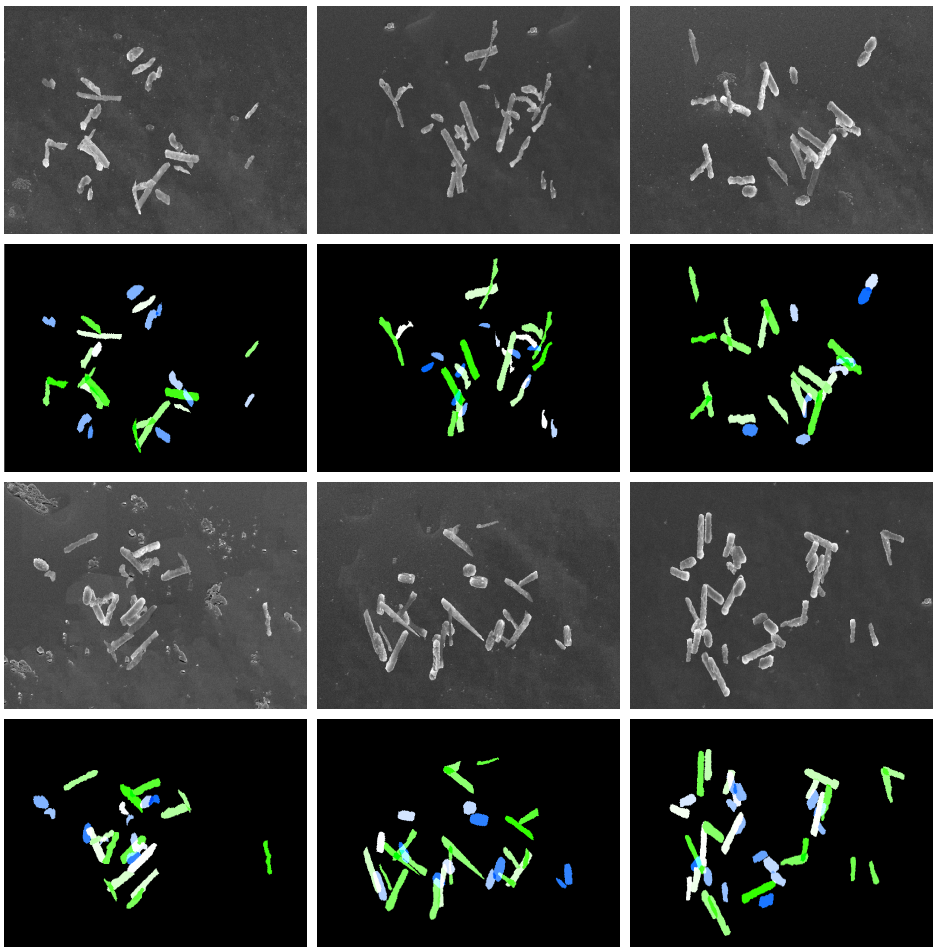


Figure 4: Depiction of samples of the synthetic cell images in UH-S-cdiff1 and their corresponding ground truth masks where vegetative cells are depicted in shades of green and spores are depicted in shades of blue.

Table 1: Definition of notations used in Algorithm 1.

Notation	Type	Definition
ϕ	Scalar	Angle between the major axes of a crossing pair of cells (degrees)
θ	Scalar	Angle of the major axis of the cell (degrees)
(x, y)	Scalar	Centroid of a cell (pixels)
e, f	Scalar	Constants
η	Scalar	Height of a cell mask (pixels)
σ	Scalar	Horizontal shear parameter
(ρ, κ)	Scalar	Image size for acquired and synthetic images (pixels)
δ	Scalar	Maximum change of the random orientation of a cell (degrees)
χ	Scalar	Maximum horizontal shift of the second cell in a touching/crossing pair (pixels)
ψ	Scalar	Maximum vertical shift of the second cell in a touching/crossing pair (pixels)
a	Scalar	Number of acquired images
n	Scalar	Number of cells in an acquired image
c	Scalar	Number of crossing pairs of cells in a synthetic image
o	Scalar	Number of isolated cells in a synthetic image
t	Scalar	Number of touching pairs of cells in a synthetic image
z	Scalar	Number of synthetic images
ϵ	Scalar	Random number in the range [0,1]
ω	Scalar	Width of a cell mask (pixels)
w	Scalar	Window size for texture synthesis (same width and height) (pixels)
$\mathbf{A}^k$	2D tensor	Acquired image k
$\mathbf{B}_i^k$	2D tensor	Annotation mask for image k and cell i with size (ρ, κ)
$\mathbf{J}_j^l$	2D tensor	Synthetic ground truth mask for image l and cell j with size (ρ, κ) (pixels)
$\mathbf{C}$	2D tensor	Image of a cell with size (ρ, κ) (pixels)
$\mathbf{C}^m$	2D tensor	Binary mask of a cell with size (ρ, κ) (pixels)
$\mathbf{I}^l$	2D tensor	The l^{th} synthetic image
$\mathbf{T}$	2D tensor	3×3 geometric transformation matrix
$\mathbf{B}^k$	3D tensor	Annotation mask for acquired image k with size (n, ρ, κ) (pixels)
$\mathbf{J}^l$	3D tensor	Synthetic ground truth mask for image l with size $(o + 2t + 2c, \rho, \kappa)$ (pixels)
$\mathscr{A}^m$	Set	Set of manually annotated masks
$\mathscr{A}^r$	Set	Set of acquired images
$\mathscr{J}^g$	Set	Set of synthetic ground truth masks
$\mathscr{J}^s$	Set	Set of synthetic cell images

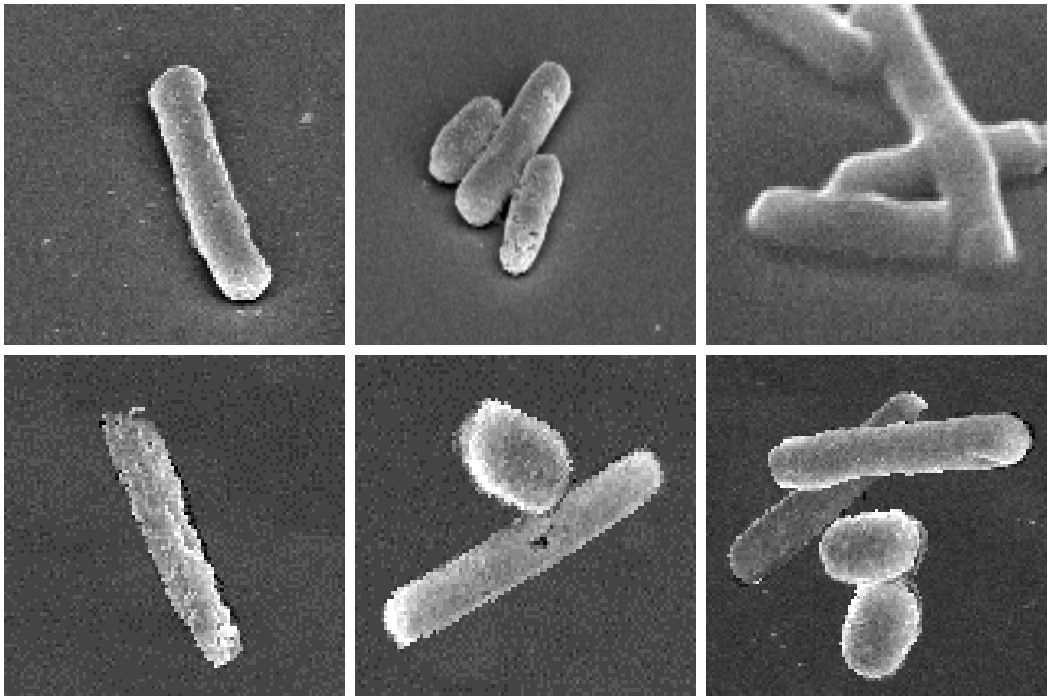


Figure 5: Depiction of samples of the acquired (T) and the synthesized (B) images. The first column depicts isolated cells, the second column includes samples of touching cells, and the third column depicts both touching and crossing cells.

Algorithm 1 Image synthesis algorithm generating images with isolated, touching, and crossing cells. The notations are defined in Table 1

Input : A set of a acquired images $\mathscr{A}^r = \{\mathbf{A}^1, ..., \mathbf{A}^a\}$ and their manually annotated masks $\mathscr{A}^m = \{\mathbf{B}^1, ..., \mathbf{B}^a\}$, number of images to be synthesized z, window size m

Output: Synthetic cell images $\mathscr{I} = \{\mathbf{I}^1, ..., \mathbf{I}^z\}$, synthetic ground truth masks $\mathscr{J} = \{\mathbf{J}^1, ..., \mathbf{I}^z\}$

Function `AddCell(`$\mathbf{A}^k$, $\mathbf{B}_i^k$, $\mathbf{I}^l$, $\mathbf{J}_j^l$, θ, x, y`):`

- Extract the i^{th} cell in the k^{th} acquired image $\mathbf{A}^k$ using its annotated mask $\mathbf{B}_i^k$: $\mathbf{C} \leftarrow \mathbf{A}^k \circ \mathbf{B}_i^k$, $\mathbf{C}^m \leftarrow \mathbf{B}_i^k$
- Translate the center of the annotation mask of the cell in $\mathbf{C}$ and its mask $\mathbf{C}^m$ to the center of the image
- Initialize a 3×3 geometric transformation matrix $\mathbf{T}$ as an identity matrix and add a random noise to the transformation parameters (e.g., horizontal shear: $\sigma = f + e\epsilon$);
- Warp the masked cell image $\mathbf{C}$ and its annotation mask $\mathbf{C}^m$ with transformation $\mathbf{T}$ and resize to size (ρ, κ)
- Compute the angle ϕ of the major axis of the cell corresponding to the horizontal axis
- Rotate $\mathbf{C}$ and $\mathbf{C}^m$ to align with orientation θ and crop to the size (ρ, κ);
- Translate the object centroid in $\mathbf{C}$ and $\mathbf{C}^m$ to location (x, y) (move the object inside if the cell is partially outside the image boundaries);
- Overlay $\mathbf{C}$ on $\mathbf{I}^l$ and $\mathbf{J}_j^l \leftarrow \mathbf{C}^m$

return $\mathbf{I}^l$, $\mathbf{J}_j^l$

for $l = 1, ..., z$ **do**

- Randomly select an image $\mathbf{A}^k$ from the acquired images $\mathscr{A}^r$ and its annotation mask $\mathbf{B}^k$ from $\mathscr{A}^m$
- Apply the image inpainting algorithm [33] to remove the cells from the image and store the background to $\mathbf{I}^l$
- Randomly select a background patch of size $w \times w$ from image $\mathbf{I}^l$
- Synthesize a background image with the same resolution as $\mathbf{I}^l$ using the texture synthesis algorithm [54] and replace with $\mathbf{I}^l$
- Randomly select the number of isolated cells o, touching pairs t, and crossing pairs c to be placed into the image $(o, t, c \in \{1, ..., n\})$
- Initialize $\mathbf{J}^l$ with zeros as a 3D tensor of size $(o + 2t + 2c, \rho, \kappa)$

end

end

for $p = 1, ..., o$ **do**

- Randomly select a cell mask i ($i \in 1, ..., n$) with annotation tensor $\mathbf{B}_i^k$ in the acquired image $\mathbf{A}^k$
- Randomly generate $\theta \in [0°, 180°]$, $x_1 \in [\frac{\omega}{2}, ..., \rho - \frac{\omega}{2}]$, $y_1 \in [\frac{\eta}{2}, ..., \kappa - \frac{\eta}{2}]$
- Add the cell into the image:
  $\mathbf{I}^l$, $\mathbf{J}_j^l \leftarrow$ `AddCell(`$\mathbf{A}^k$, $\mathbf{B}_i^k$, $\mathbf{I}^l$, $\mathbf{J}_j^l$, θ, x_1, y_1`)`

for $q = 1, ..., t$ **do**

- Randomly select a cell mask i ($i \in 1, ..., n$) with annotation tensor $\mathbf{B}_i^k$ in the acquired image $\mathbf{A}^k$
- Randomly generate $\theta \in [0°, 180°]$, $x_1 \in [\frac{\omega}{2}, ..., \rho - \frac{\omega}{2}]$, $y_1 \in [\frac{\eta}{2}, ..., \kappa - \frac{\eta}{2}]$
- Add the first touching pair into the image the first output:
  $\mathbf{I}^l$, $\mathbf{J}_j^l \leftarrow$ `AddCell(`$\mathbf{A}^k$, $\mathbf{B}_i^k$, $\mathbf{I}^l$, $\mathbf{J}_j^l$, θ, x_1, y_1`)`
- Update $(x_1, y_1) \leftarrow$ centroid of $\mathbf{J}_j^l$
- Randomly select a cell mask u with annotation tensor $\mathbf{B}_u^k$ in the acquired image $\mathbf{A}^k$ as the second cell
- Let ω be the width and (x_1, y_1) the centroid of the cell mask $\mathbf{C}^m$. Randomly select a location (x_2, y_2):
  $x_2 \in \left[x_1 - \omega, x_1 - \frac{\omega}{2}\right] \cup \left[x_1 + \frac{\omega}{2}, x_1 + \omega\right]$ and $y_2 \in [y_1 - \psi\epsilon, y_1 + \psi\epsilon]$
- Add the second touching pair into the image:
  $\mathbf{I}^l$, $\mathbf{J}_v^l \leftarrow$ `AddCell(`$\mathbf{A}^k$, $\mathbf{B}_u^k$, $\mathbf{I}^l$, $\mathbf{J}_v^l$, θ, x_2, y_2`)`

for $r = 1, ..., c$ **do**

- Randomly select a cell mask i ($i \in 1, ..., n$) with annotation tensor $\mathbf{B}_i^k$ in the acquired image $\mathbf{A}^k$
- Randomly generate $\theta \in [0°, 180°]$, $x_1 \in [\frac{\omega}{2}, ..., \rho - \frac{\omega}{2}]$, $y_1 \in [\frac{\eta}{2}, ..., \kappa - \frac{\eta}{2}]$
- Add the first crossing pair into the image: $\mathbf{I}^l$, $\mathbf{J}_j^l \leftarrow$ `AddCell(`$\mathbf{A}^k$, $\mathbf{B}_i^k$, $\mathbf{I}^l$, $\mathbf{J}_j^l$, θ, x_1, y_1`)`
- Update $(x_1, y_1) \leftarrow$ centroid of $\mathbf{J}_j^l$
- Randomly select a cell mask u with annotation tensor $\mathbf{B}_u^k$ in the acquired image $\mathbf{A}^k$ as the second cell
- Let η be the height and (x_1, y_1) the centroid of the cell mask $\mathbf{C}^m$. Randomly select a location (x_2, y_2):
  $y_2 \in \left[y_1 - \eta, y_1 - \frac{\eta}{2}\right] \cup \left[y_1 + \frac{\eta}{2}, y_1 + \eta\right]$ and $x_2 \in [x_1 - \chi\epsilon, x_1 + \chi\epsilon]$
- Randomly select an angle ϕ: $\phi = \theta + 90 \pm \delta\epsilon$
- Add the second crossing pair into the image: $\mathbf{I}^l$, $\mathbf{J}_v^l \leftarrow$ `AddCell(`$\mathbf{A}^k$, $\mathbf{B}_u^k$, $\mathbf{I}^l$, $\mathbf{J}_v^l$, ϕ, x_2, y_2`)`

2.2.2 Estimate of the number of training instances

Deep neural networks have a large number of parameters. Learning the values for such parameters requires a large number of training data. Even though there is no closed-form formula for the minimum number of instances required to train a neural network, there are heuristics that help us estimate the number of training samples by simplifying the problem. In this section, two analogies are mentioned to provide an estimate and the effect of regularization is discussed.

The first analogy compares training of a neural network to solving a system of linear equations where the linear variables are the parameters of the network and each linear equation represents a constraint (pattern) that needs to be satisfied (recognized). Each sample provides insight into the pattern in a real-world example and each variable provides a degree of freedom to find a solution that satisfies the equations. According to the numerical methods such as Gauss–Jordan elimination, the number of training samples should be at least equal to the parameters of the network [84]. However, this heuristic does not consider the non-linearity in neural networks. The second analogy compares the training to fitting a polynomial to a set of points. For instance, fitting a polynomial of degree n with less than n points results in over-fitting [84]. Even though, the two analogy mentioned above does not accurately describe the training process. They can provide an intuitive idea of the importance of having a large training dataset.

Lack of training data may lead to over-fitting. However, regularization and dropout techniques prevent over-fitting, allowing for training more complex architecture [52]. The concept is similar to dictionary learning where a function is approximated by a combination of an over-complete basis while sparsity is enforced to reduce the redundancy.

Common architectures such as ResNet50 (25.6M), ResNet101 (44.5M), and VGG16 (138M) that have achieved the state-of-the-art, were trained on MS-COCO [40] and ImageNet [13] datasets with much less number of images. ImageNet has 1.2 million images of 1,000 synonym sets of instances referring to a unique concept (synsets) for object detection and MS-COCO has more than 200K labeled images of 1.5 million object instances for 80 object categories for object detection and segmentation.

ImageNet statistics [13] has resulted in a rule of thumb, collecting 1,000 samples for each synset in the test set. Furthermore, transferring the learned parameters of a network such as ResNet50 trained on ImageNet or MS-COCO is commonly used in biomedical image analysis where data is not abundant. Fast convergence has been reported in the analysis of pathology images after initialization with MS-COCO pretrained weights [36].

Given the ImageNet rule of thumb, three synsets were defined to represent the challenges of detecting CDI cells in SEM images, namely, isolated cells, pairs of touching cells, and pairs of crossing cells for each class (vegetative and spore). Therefore, 6,000 images were selected as the required number of training samples.

2.3 Results

The synthesized images are based on the acquired SEM dataset UH-A-cdiff1. UH-A-cdiff1 consists of 20 C. diff cell images (197 vegetative cells and 111 spores) acquired via scanning electron microscopy. Image dimensions are 411×711 pixels with 10,000x magnification. Moreover, many cells are touching or crossing each other with the existence of debris. Also, the cells were partially deformed and cell walls are damaged due to a laboratory treatment, making the detection more challenging. Two sets of annotations were provided labeling every cell as a binary mask for every cell in the image. Algorithm 1 is applied to synthesize three synthetic datasets described below, similar to UH-A-cdiff1 and the primary set of annotations were used to synthesize the ground truth labels.

1. UH-S-cdiff1: Algorithm 1 is applied to synthesize 6,000 images of 411×711 pixel dimensions. The synthetic images include two to four pairs of the following scenarios: a pair of two vegetative cells touching, a pair of two vegetative cells crossing, and a vegetative cell touching a spore. Moreover, two to four single isolated cells of each type are added into the image, creating a variety of possible overlaps between cells in various orientations. Figure 4 depicts samples of the synthetic cell images in UH-S-cdiff1 and their synthesized ground truth.

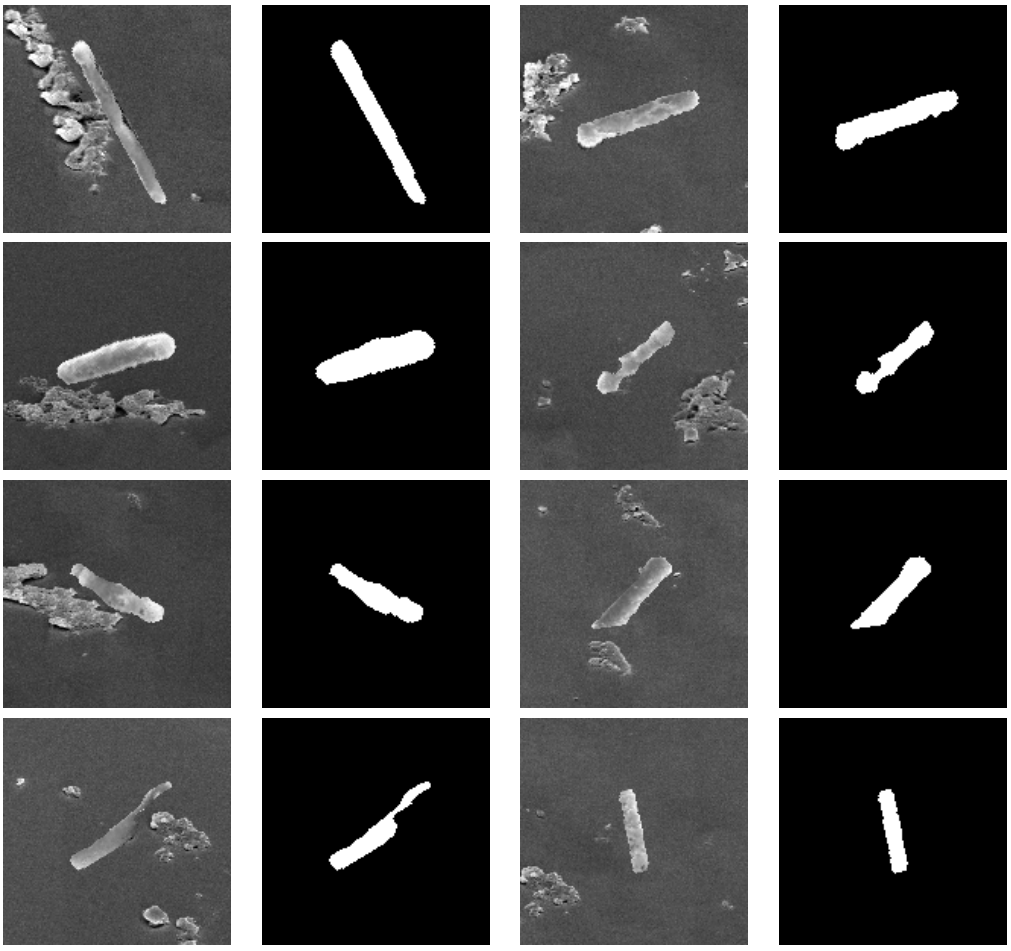


Figure 6: Depiction of samples of the synthesized isolated cells with SEM background with their ground truth annotations in presence of debris and artifacts.

2. UH-S-cdiff0: The initial dataset to train the adversarial pipeline included 2,000 images with 411×711 pixels were the cells were randomly placed into the image with no restriction on the number of touching and crossing pairs. The images were divided into patches of 150x150 to provide more than 17,000 training samples. Furthermore, the dataset was synthesized for semantic segmentation where the target was to segment the cell areas from the background. Figure 5 depicts samples of the isolated, touching, and crossing cells in UH-S-cdiff0.

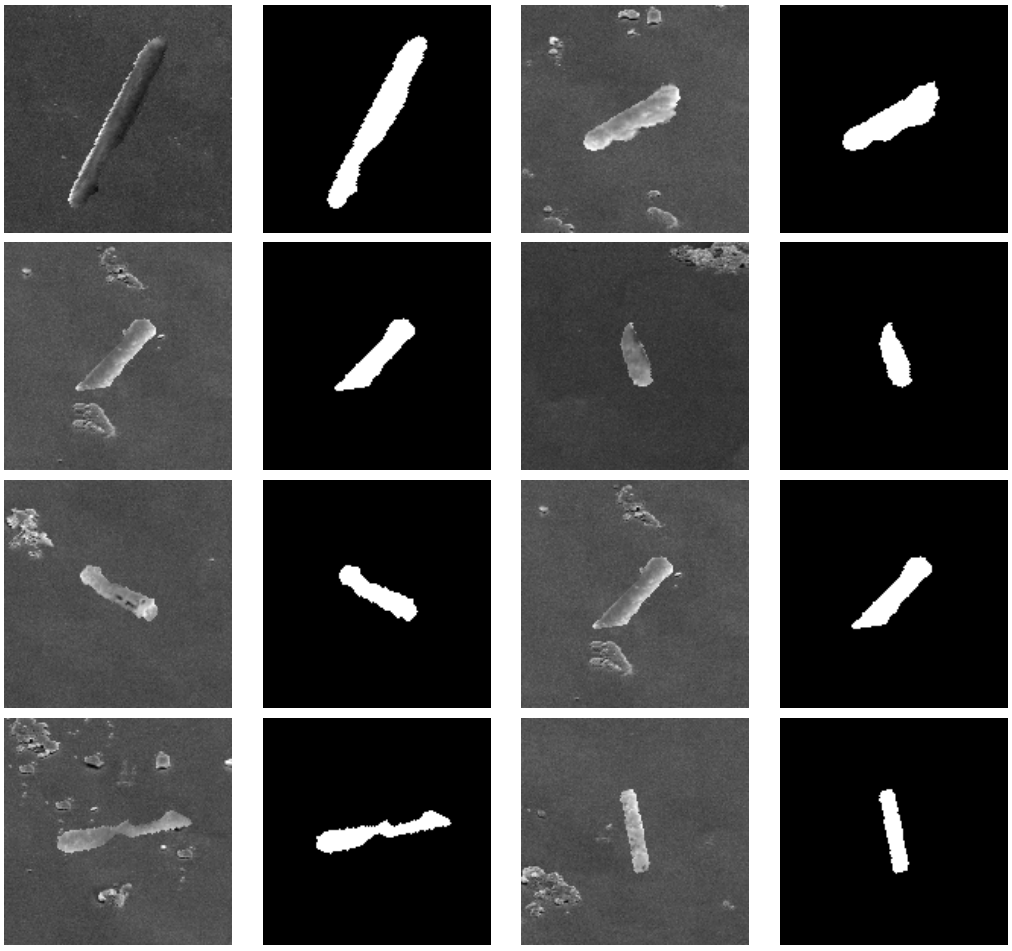


Figure 7: Depiction of samples of the synthesized isolated cells with SEM background with their ground truth annotations with inhomogeneous illumination.

3. UH-P-cdiff1: A synthetic dataset with image patches of 150x150 resolution was created containing three categories of isolated, touching, and crossing cells. For each category 2000 samples and their masks are generated with distinct random seed values (6,000 total). Each category is divided into 10 folds for cross-validation. For instance, in the first cross-validation experiment is to predict the result for the first fold. Therefore, the first fold is reserved for validation, and folds 2-10 are used for training. Figure 4 depicts samples of the synthetic images with isolated, touching, and crossing cells with inhomogeneous illumination.

Figure 6 depicts samples of the synthetic isolated cells in presence of debris and artifacts. Figure 7 depicts samples of the synthetic isolated cells in presence of inhomogeneous illumination. Figure 8 depicts samples of the synthetic touching cells with a narrow background between the cells. Figure 9 depicts samples of the synthetic touching cells with inhomogeneous illumination where the cells have overlaps over the cell boundaries. Figure 10 depicts samples of the synthetic crossing cells with inhomogeneous illumination where the cells are aligned with vertical and horizontal axes. Figure 11 depicts samples of the synthetic crossing cells with inhomogeneous illumination where the cells are rotated.

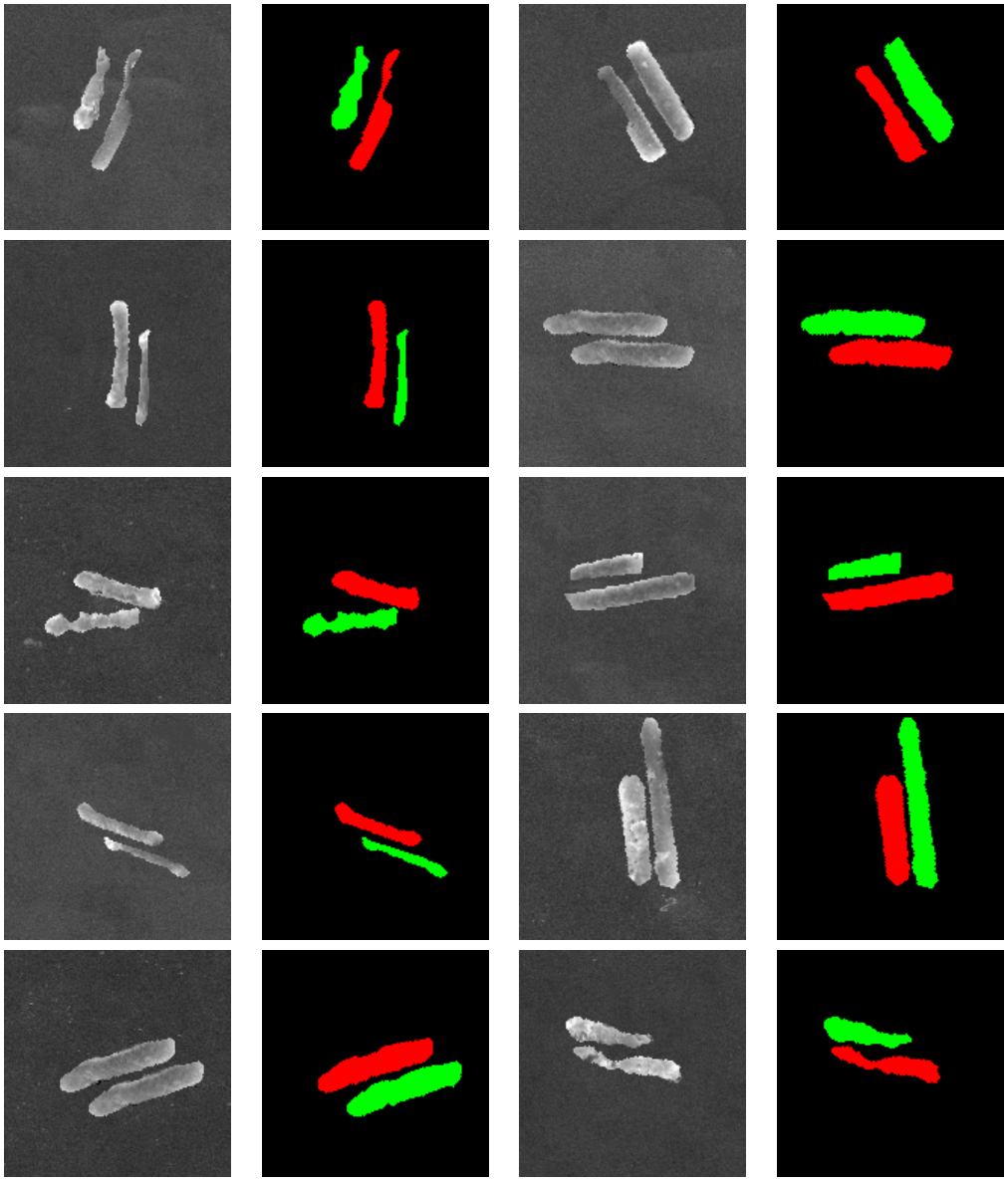


Figure 8: Depiction of samples of the synthesized touching cells with SEM background with their ground truth annotations with inhomogeneous illumination where there is a narrow background between the cells.

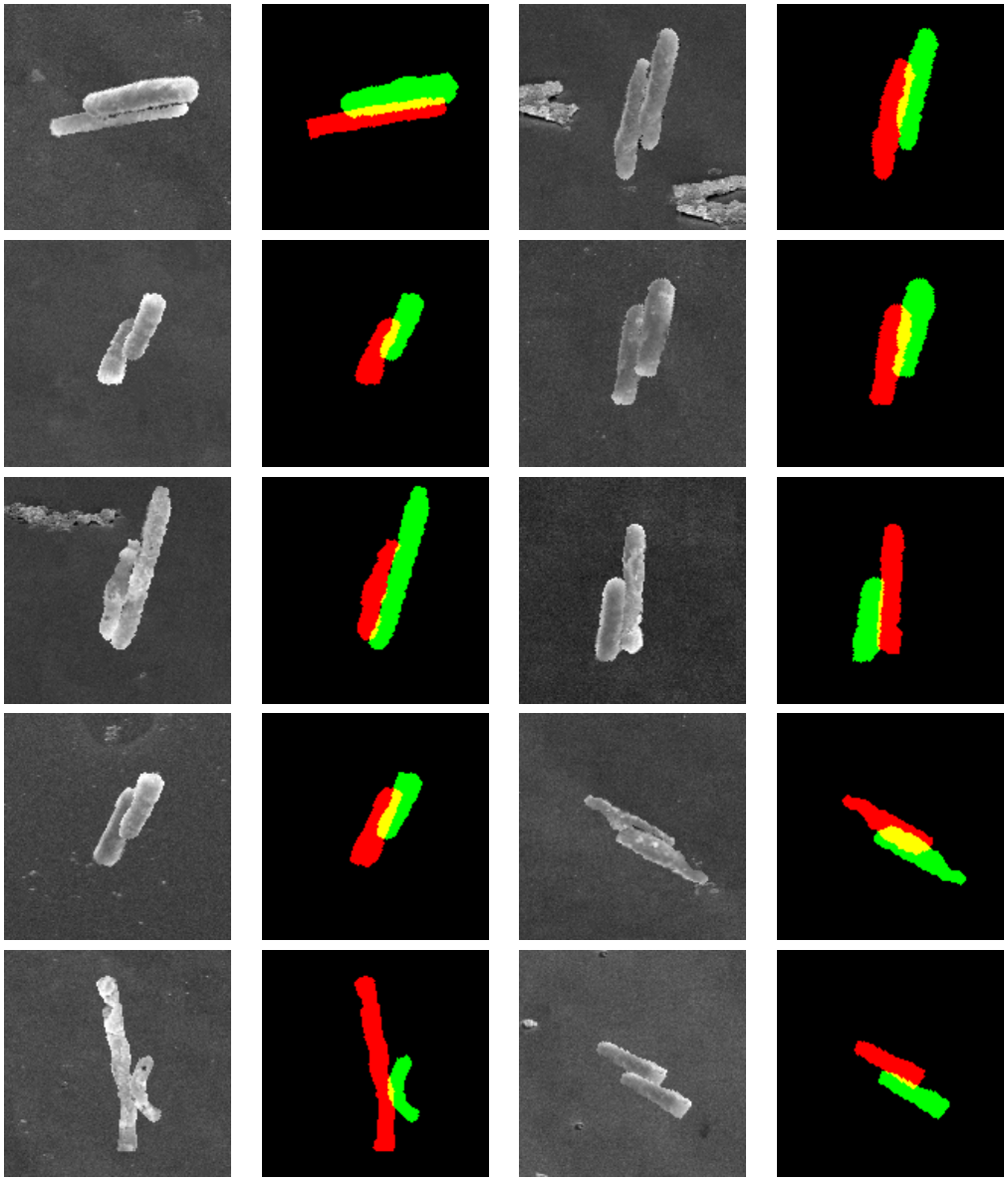


Figure 9: Depiction of samples of the synthesized touching cells with SEM background with their ground truth annotations with inhomogeneous illumination where the cells have overlaps over the cell boundaries.

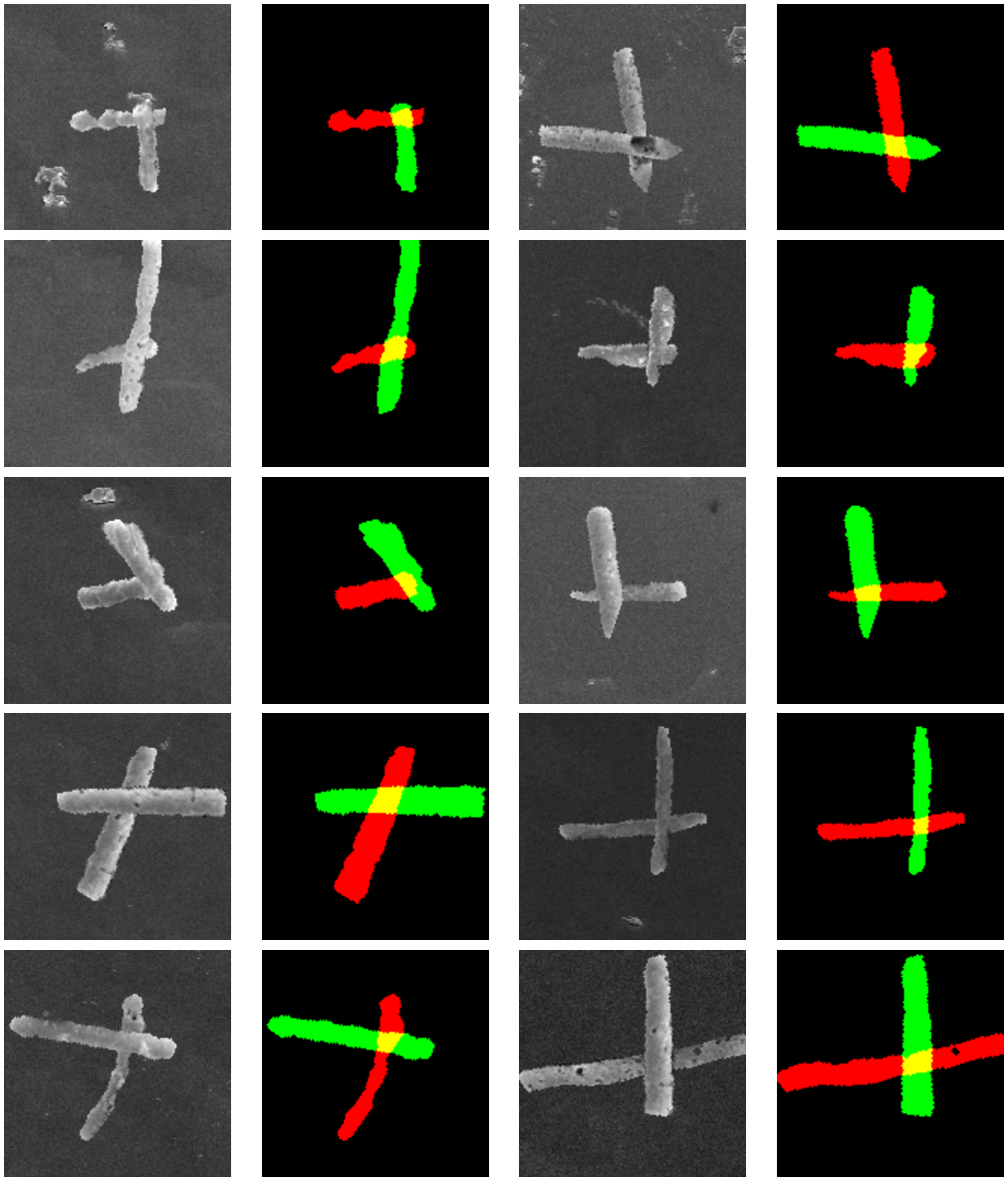


Figure 10: Depiction of samples of the synthesized touching cells with SEM background with their ground truth annotations with inhomogeneous illumination where the cells are aligned with vertical and horizontal axes.

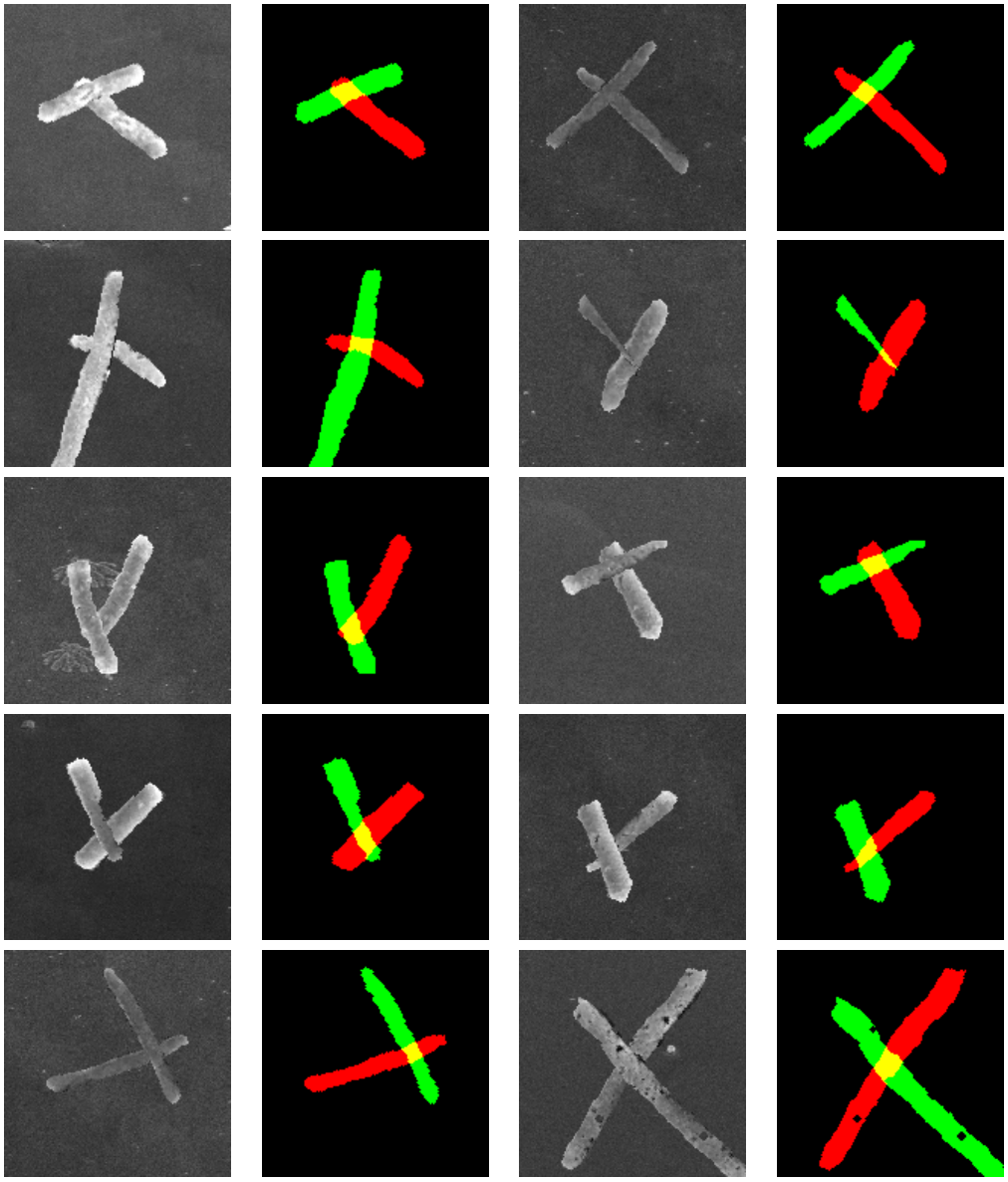


Figure 11: Depiction of samples of the synthesized crossing cells with SEM background with their ground truth annotations with inhomogeneous illumination where the cells are rotated.

3 Objective 2: A pipeline for the segmentation of cell images with inhomogeneous illumination

Analysis of CDI cell images via SEM is a challenging task due to the intrinsic properties of cells, their micron level size, or imperfect image acquisition process. The images are degraded by inhomogeneous illumination creating shadows on the cells, bright areas around the cells, and intensity non-uniformity in the background. Inhomogeneous illumination results in shadows on the cells. The resulting shadows on the cells cause variation of intensities across pixels belonging to the same cell that is similar to the bias field artifact in medical imaging modalities such as computed tomography, and magnetic resonance imaging.

3.1 Related work

3.1.1 Inhomogeneous illumination in biomedical images

Electron microscopy images are degraded by inhomogeneous illumination. Therefore, addressing inhomogeneous illumination retrospectively is of paramount importance in the analysis of CDI images. Previous illumination normalization approaches could be categorize into: Filtering-based methods [5, 64, 67, 82]. and optimization-based methods [90, 91, 37, 29, 46, 71, 86, 87].

Filtering-based Methods: Filtering-based methods model the intensity inhomogeneity as low-frequency artifacts. Although these methods have low computational cost, they are not effective since they eliminate the texture of the cells and create filtering artifacts near the edges. Some approaches [67, 82] applied homomorphic filtering as an illumination normalization technique in the log domain that can reduce the intensity inhomogeneity and increase the image contrast:

$$\mathbf{I}^n = \exp\left(\log\left(\mathbf{I}\right) - \log\left(\mathbf{I}\right) * \mathbf{S}\right) + c \tag{1}$$

where, the bias field is computed by filtering the original image $\mathbf{I}$ with a low-pass filter $\mathbf{S}$. Then, the bias field is subtracted from the image in the log domain followed by adding a constant c to obtain

the illumination normalized image $\mathbf{I}^n$. Other approaches suggested computing an approximation of the $\mathbf{I}^n$ [5, 64].

Optimization-based methods: Optimization-based approaches proposed several objective function to estimate the true reflectance of objects such as quadratic fidelity of the reflectance gradient with respect to the observed image gradients along with sparsity and fidelity priors [90, 91], MAP estimations [37, 29, 46], gradient, and intensity distributions [71, 86, 87].

3.1.2 Cell segmentation

Cell detection methods fall into two categories: Region-based methods and deep convolutional networks.

Region-based methods: Region-based cell detection methods first detect a collection of cell candidate regions based on shape or statistical texture descriptors. Then, the best candidates are selected via correlation clustering [85], optimization-based [8, 10, 7, 45], or heuristic methods [29, 62]. Some approaches applied feature extractors on image patches. Then, the extracted features were forwarded to a classifier, such as random forests, to identify the cell centroids [28] using several distance metrics for the classification score [77, 75, 61, 48]

Deep ConvNet based methods: Recently, deep convolutional networks have outperformed state of the art in many biomedical image processing tasks [65]. Fully convolutional networks have been applied for image segmentation [41]. Specifically, U-net is widely used for biomedical image segmentation tasks [58]. However, to the best of our knowledge inhomogeneous illumination has prevented the introduction of a deep network for automatic segmentation in SEM images.

Generative adversarial training has been applied to medical images, improving the image segmentation by producing label maps that are similar to the manual ground truth [42]. Recently, adversarial networks have gained more attention in the segmentation of MRI images [11, 32, 38, 49, 81] where the datasets and the annotations are available.

3.2 Methods

This Section, presents SoLiD: "Segmentation of *Clostridioides difficile* cells in the presence of inhomogeneous iLlumInation using a Deep adversarial network" [44]. SoLiD applies a fully convolutional U-net based network $\mathcal{S}$ as the segmenter and a deep ConvNet $\mathcal{D}$ as the discriminator for the adversarial training. The segmenter predicts a label map for the pixels while the discriminator distinguishes between the predicted label maps and the ground truth. Figure 12 depicts the segmenter and the discriminator networks in the SoLiD pipeline.

The input to the segmenter is a cell image. The segmenter includes a convolution path and a deconvolution path similar to U-net. The convolution path extracts a feature map for segmentation using convolution layers while the deconvolution path increases the resolution, creating a label map. The generated label map may differ significantly from the ground truth since the segmenter does not consider the smoothness of the labels, resulting in a non-continuous segmentation.

The discriminator is used to train the segmenter to produce label maps similar to the ground truth. The discriminator is a regular ConvNet classifier trained on the ground truth and predicted segmentation masks. During training, it learns to classify the input image into two classes: "artificially generated" or "ground truth", and backpropagates the gradients.

3.2.1 Segmenter Network

The segmenter network consists of six convolutional units: the first three units include a 3x3 convolution layer, a ReLU layer, and a 2x2 max pooling layer with a stride of two, downsampling the image (contracting units). The next three units (expanding units) include an upsampling of the features followed by a 2x2 deconvolution. Each contracting unit doubles the number of feature channels while each expanding unit halves the number of channels. The segmenter minimizes a loss function L_S:

$$L_S = \mathbf{w_c} * L_C\left(\mathcal{S}(\mathbf{I}), \mathbf{G}\right) + L_C\left(\mathcal{D}\left(\mathcal{S}(\mathbf{I})\right), 1\right) \tag{2}$$

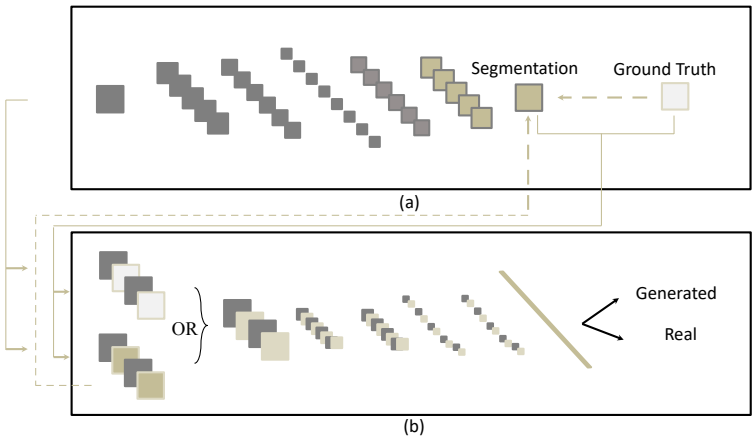


Figure 12: Depiction of the adversarial architecture of SoLiD. (a) The segmenter network predicts a label map and feeds it to the discriminator. (b) The discriminator distinguishes between the predicted label maps and the ground truth. The adversarial loss is then back-propagated through the network to update both networks.

where $L_C\Big(\mathcal{S}(\mathbf{I}), \mathbf{G}\Big)$ is a cross-entropy term between the predicted labels S corresponding to the image $\mathbf{I}$ and the ground truth $\mathbf{G}$. The second term $L_C\Big(\mathcal{D}\big(\mathcal{S}(\mathbf{I})\big), 1\Big)$ is the adversarial loss term, computed by the discriminator. The label map of image $\mathbf{I}$ generated by the segmenter is denoted by $\mathcal{S}(\mathbf{I})$ and $\mathcal{D}$ is the discriminator network described in the next section. The adversarial loss forces the segmenter to produce label maps that would be considered as ground truth by the discriminator. To distinguish touching cells, the segmenter considers the boundaries of the cells (cell wall) as a separate class. Hence, the segmenter loss is one-hot encoded with three classes. The number of cell wall samples is considerably less compared to the other classes. To compensate for the bias in the training set, the segmenter cross-entropy loss is weighted ($\mathbf{w_c}$). The minority class receives a higher classification weight.

The segmenter network may misclassify a large portion of cells due to inhomogeneous illumination. We present the adversarial training to evaluate such misclassifications and improve the segmenter (Algorithm 2). A discriminator ConvNet is applied to compute the likelihood of the predicted segmentation map being an actual label map.

Algorithm 2 SoLiD training

Input : Augmented training cells, Training labels
Output: Trained segmenter network, Trained discriminator network

begin
  for *number of pretraining iterations* **do**
    Select a batch of labels $\mathbf{G}$
    Train the discriminator with the cross-entropy loss $L_C\big(\mathcal{D}(G), 1\big)$
  end
  for *number of adversarial iterations* **do**
    Select a batch of training images and their labels $\{\mathbf{I}, \mathbf{G}\}$
    Feedforward the batch to the segmenter and predict the segmentation $\mathcal{S}(\mathbf{I})$. Compute the segmentation cross-entropy loss $L_C(\mathcal{S}(\mathbf{I}), \mathbf{G})$
    Feed the predicted labels to the discriminator and compute the adversarial loss $L_C\big(\mathcal{D}(\mathcal{S}(\mathbf{I})), 0\big)$
    Given the labels $\mathbf{G}$, compute the discriminator cross-entropy loss $L_C\big(\mathcal{D}(\mathbf{G}), 1\big)$
    Compute L_D and backpropogate the discriminator (Eq. 17) Compute L_S and backpropogate the segmenter (Eq. 23)
  end
end

3.2.2 Discriminator Network

The discriminator improves the generated labels by sending feedback to the generator if the segmentation labels are significantly different from the ground truth. It does not increase the complexity of the network since it is used only during training. It consists of five convolutional layers with valid padding, followed by ReLU activations and average pooling. Furthermore, two fully connected layers are placed at the end of the discriminator.

To avoid saturation, the last layer of the discriminator does not have a thresholding operator, so it produces an unscaled output. Computing scores between zero and one may cause the discriminator to generate values close to 0 for generated label maps, in which case the gradient would be too small to update the generator and eventually saturate the network [22].

The discriminator $\mathcal{D}$ computes the cross-entropy of the ground truth label maps $\mathbf{G}$ and 1, and the cross-entropy of the generated label maps $\mathcal{S}(\mathbf{I})$ and 0, minimizing the following loss function:

$$L_D = L_C\bigg(\mathcal{D}(G), 1\bigg) + L_C\bigg(\mathcal{D}(\mathcal{S}(\mathbf{I})), 0\bigg). \tag{3}$$

During the training, the discriminator improves the segmenter network, penalizing the segmentation labels that do not look like manual labels. Therefore, the adversarial result has properties such as smoothness and robustness to inhomogeneous illumination.

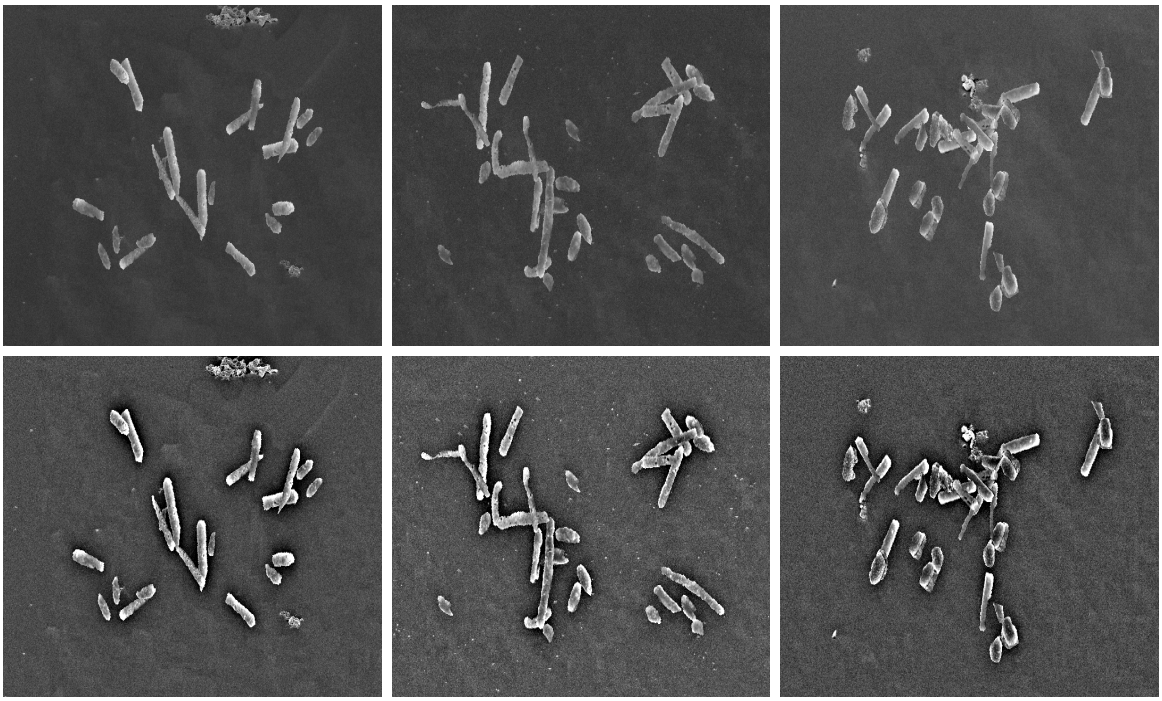


Figure 13: (T) Depiction of samples of the synthesized images of UH-S-cdiff0 and (B) their illumination normalized images.

3.3 Results

UH-S-cdiff0 dataset is used to train SoLiD and U-net as the baseline. Since the target is to segment the cell areas in SEM images, only the original SEM images were included in the test set. Three baseline models were trained based on U-net. In the first baseline, U-net is trained with UH-S-cdiff0. The second baseline is a trained U-net with illumination normalized version of the images in UH-S-cdiff0. An illumination normalization method [90] is applied to form the second training set. Figure 13 depicts samples of the illumination normalized images. For the third baseline, the training set in synthesized with warping the original images with horizontal shear. Then, the images

Table 2: Comparative results between the segmentation performance of SoLiD and the state-of-the-art in semantic segmentation by U-net. The dice score represents the performance of the methods segmenting the cells correctly while AUC depicts the performance over the whole image.

Method	Cell Dice Score	AUC
U-net [58]	0.50	0.93
U-net (trained with illumination normalized images only)	0.07	0.82
U-net (trained with warped images only)	0.49	0.71
SoLiD	**0.72**	**0.99**

are cropped and resized to the original size. Five-fold cross-validation is performed to ensure that the train and test set are different images.

Figure 14 depicts the qualitative comparison of the segmentation obtained by SoLiD and the three baselines. The effect of inhomogeneous illumination can be observed as bright areas around the cells, bright spots on the cell body, and shadows in the background.

Pixel labels are assigned according to the maximum score values obtained by the segmenter network. One-vs-all is applied to obtain binary masks for the detected cells. Then, the dice scores were computed for the cells to measure the performance of cell segmentation. The area under the curve (AUC) was computed for the entire image to measure the classification performance over all three classes. Table 2 compares the performance of SoLiD with the baselines.

3.4 Discussion

In this Chapter, an adversarial pipeline was developed and evaluated for the semantic segmentation of CDI cells in SEM images. Inhomogeneous illumination is modeled as an adversarial attack and a discriminator ConvNet is used to improve the segmentation performance. The results indicated that the pipeline improved the segmentation of CDI cells from the background by at least 44 percent compared to U-net. Furthermore, U-net trained with illumination normalized images resulted in significantly lower dice scores due to added high-frequency information and destroyed texture caused by illumination normalization. Moreover, SoLiD and U-net trained with UH-S-cdiff0 achieved higher dice scores compared to the training set where only warping is used for data augmentation.

Therefore, Algorithm 1 resulted in a better dice score compared to data augmentation with warping.

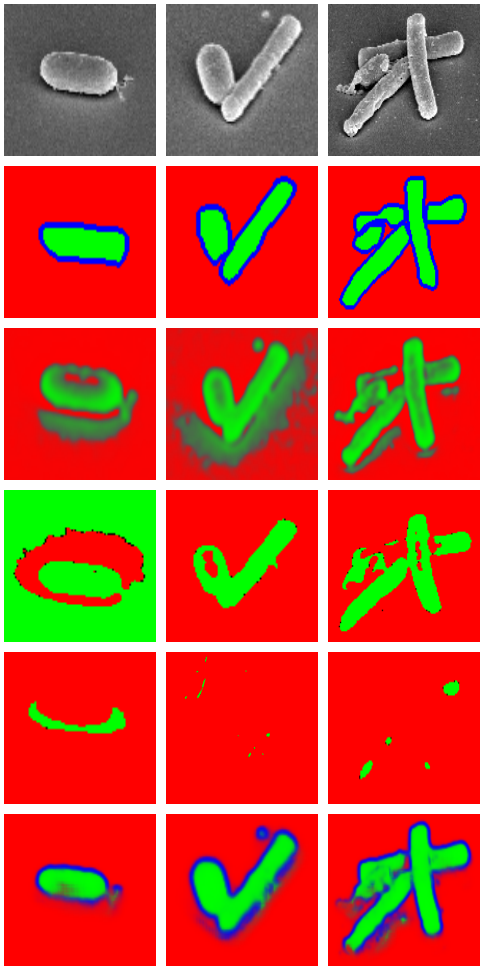


Figure 14: The segmentation results from SoLiD are qualitatively compared to the result from U-net: The first and second rows depict the original image and their annotated segmentation respectively. The third row depicts the segmentation results by U-net [58]. The fourth row depicts the result of U-net trained with illumination normalized images. The fifth row depicts the result of U-net trained with warp only images. The last row illustrates the segmentation by SoLiD. The figure is best viewed in color.

4 Objective 3: Addressing the challenge of touching/crossing cells

This Chapter presents DETCID (Detection of Elongated Touching C. difficile cells in the presence of inhomogeneous Illumination using a Deep adversarial network) a deep cell detection algorithm to detect C. diff cells in SEM images. Similar to SoLiD [44], DETCID models the inhomogeneous illumination as an adversarial attack. SoLid [44] demonstrated that adversarial training could improve the semantic segmentation performance of U-net in presence of inhomogeneous illumination by 44% where the adversarial loss penalizes the segmentation output to be similar to the ground truth without increasing the complexity of the network during deployment.

DETCID expands the semantic segmentation method in SoLiD to an instance-level segmentation method where several bounding boxes are selected to extract features relevant features from a deep feature pyramid network such as ResNet50. Furthermore, the mask-based non-max suppression method detects the clusters of touching cells in various orientations. An image synthesis algorithm is developed to generate clustered cell images to train the network.

4.1 Related Work

This section categorizes the related work on cell detection into shallow and deep detection methods. Then, it presents the findings from a review of the literature on deep segmentation methods addressing inhomogeneous illumination.

4.1.1 Shallow cell detection methods

Shallow methods select cell candidate regions using intensity thresholding or energy minimization. Then, an optimization algorithm is applied [7, 70], a machine learning regressor is trained over a set of hand-crafted features to select the cell from a set of candidates [28, 62, 21, 3], or a level set segmentation is used [89, 88]. Finally, size filtering or hole filling is applied to refine the results. To address the inhomogeneous illumination, pre-processing steps are proposed. However, these pre-processing steps may remove texture information, making the detection more challenging [47]. Ulman *et. al.* [72] performed an objective comparison of many shallow cell detection algorithms

with deep convolutional networks. Shallow methods do not require expensive computing hardware to train and are interpretable. However, they are outperformed by deep learning methods.

4.1.2 Deep cell detection methods

Deep learning algorithms have outperformed the state of the art in many biomedical image processing tasks [65]. Shi *et. al.* [66] applied a cascade of Quaternion Grassmann average layers to develop an unsupervised deep network for the segmentation of histology cells. Others have applied deep auto-encoders for cell segmentation [27]. Shen *et. al.* [65] reviewed many unsupervised, or CNN-based approaches combined with hand-crafted features for segmentation of biomedical images. Roopa *et. al.* [25] trained a CNN with hand-craft features as input to classify white blood cells in peripheral blood smear images. Hand-crafted cell nuclei boundary masks are also used as a shape prior to filter the detection of CNNs [69]. Others applied CNNs for cell detection with pixel-level classification for each patch in the images[80, 78, 68]. Hofener *et. al.* [26] applied post-processing to smooth the scores derived by CNNs to improve the cell nuclei detection in histology images. However, patch-based approaches need to run the network for every patch resulting in redundant computations.

To reduce the computations by sharing the computations over the overlapping patches, fully convolutional networks were introduced for image segmentation [41]. Specifically, U-net is widely used for biomedical image segmentation tasks [58]. Xie *et. al.* [79] evaluated the performance of U-net on multiple pathology datasets. Ramesh *et. al.* [55] added an unsupervised pre-processing layer with logistic sigmoid functions to U-net to separate clustered image patches from each other. However, U-net is sensitive to inhomogeneous illumination which increases the false positive for segmentation of SEM images [44]. Xie *et. al.* [78] applied two U-shape network architecture without skip connections to compute cell spatial density maps. Spatial densities are applied to detect cell overlaps. However, the application is limited to round shape objects only [78, 51].

Gu *et. al.* [23] proposed a pyramid of residual blocks to capture spatial information in multiple resolutions to detect histology cells in various sizes. Li *et. al.* [36] replaced the head layers

in Mask R-CNN with a conditional random field (CRF) to impose smoothness on the boundaries of segmented patches.

4.1.3 Deep methods addressing inhomogeneous illumination

Deep networks, such as U-net, are sensitive to inhomogeneous illumination. Wan *et. al.* [74] proposed an iterative process where a U-net is applied to provide a preliminary segmentation followed by a convolution layer to estimate the bias field in magnetic resonance (MR) images. Next, the bias field corrected image is again sent to the U-net for the next iteration, improving the segmentation. However, the iterative process involves many passes through the network, increasing the complexity of the method.

Generative adversarial networks (GANs) have been used to add robustness to adversarial attacks to the deep networks [22]. Adversarial training can improve image segmentation by producing label maps that are similar to a target image [42]. Adversarial networks have been applied in the segmentation of MR images [38, 49, 81] where the datasets and the annotations are available. However, the application is limited since the adversarial training requires a large training set to train both the segmenter and the discriminator networks. Lee *et. al.* [34] proposed an unsupervised image deconvolution method using a cycle-consistent adversarial network to improve the quality of blurred and noisy fluorescence microscopy images without labeled data. The adversarial network in DETCID models the illumination as an adversarial attack without increasing the complexity of the network during deployment.

4.2 Methods

DETCID compromises of two parts: A deep adversarial region proposal network (ARPN) and a feature extraction network (FEN). Figure 15 depicts the overview of the pipeline. The input to ARPN is a cell image and the output is a label map. The FEN is fully convolutional and produces a probability map for the presence of cells in the image. An RoI alignment layer combines the output of the two networks and aligns the extracted features with the input. RoIs are passed through two

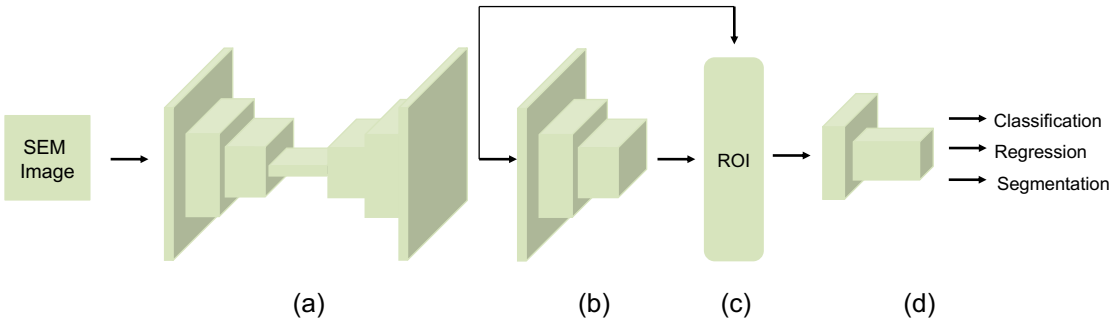


Figure 15: Depiction of DETCID Pipeline: (a) An adversarial region proposal network (ARPN) selects the cell region to be classified. (b) an FEN extracts the features from cell candidate regions. (c) detected region of interests (ROI) are aligned using bilinear interpolation. (d) two convolution layers are applied to produce the final mask.

convolution layers to produce the final segmentation mask.

4.2.1 Adversarial region proposals network

The ARPN consists of two deep ConvNets, namely the segmenter and the discriminator. The segmenter predicts a label map for the pixels while the discriminator distinguishes between the predicted label maps and the ground truth.

The input to the segmenter is a cell image. The segmenter includes a convolution path and a deconvolution path similar to U-net. The convolution path extracts a feature map for segmentation using convolution layers while the deconvolution path increases the resolution, creating a label map. The generated label map may differ significantly from the ground truth since the segmenter does not consider the smoothness of the labels, resulting in a non-continuous segmentation.

A second ConvNet (discriminator) is used to train the segmenter to produce label maps similar to the ground truth. The discriminator is a regular ConvNet classifier trained on the ground truth and predicted segmentation masks. During training, it learns to classify the input image into two classes: “artificially generated” or “ground truth”, and backpropagates the gradients.

4.2.2 Segmenter Network

The segmenter network consists of six convolutional units: the first three units include a 3x3 convolution layer, a ReLU layer, and a 2x2 max pooling layer with a stride of two. The next three units include an upsampling of the features followed by a 2x2 deconvolution. Each contracting unit doubles the number of feature channels while each expanding unit halves the number of channels. The segmenter minimizes a loss function L_S:

$$L_S = \mathbf{w_c} * L_C\Big(\mathcal{S}(\mathbf{I}), \mathbf{G}\Big) + L_C\Big(\mathcal{D}\big(\mathcal{S}(\mathbf{I})\big), 1\Big), \tag{4}$$

where $L_C\Big(\mathcal{S}(\mathbf{I}), \mathbf{G}\Big)$ is a cross-entropy term between the predicted labels S corresponding to the image $\mathbf{I}$ and the ground truth $\mathbf{G}$. The second term $L_C\Big(\mathcal{D}\big(\mathcal{S}(\mathbf{I})\big), 1\Big)$ is the adversarial loss term, computed by the discriminator. The label map of image $\mathbf{I}$ generated by the segmenter is denoted by $\mathcal{S}(\mathbf{I})$ and $\mathcal{D}$ is the discriminator network described in the next section. The adversarial loss forces the segmenter to produce label maps that would be considered as ground truth by the discriminator. To distinguish touching cells, the segmenter considers the boundaries of the cells (cell wall) as a separate class. Hence, the segmenter loss is one-hot encoded with three classes. The number of cell wall samples is considerably less compared to the other classes. To compensate for the bias in the training set, the segmenter cross-entropy loss is weighted ($\mathbf{w_c}$). The minority class receives a higher classification weight.

The segmenter network may misclassify a large portion of cells due to inhomogeneous illumination. We applied the adversarial training to evaluate such misclassifications and improve the segmenter. A discriminator ConvNet is applied to compute the likelihood of the predicted segmentation map being an actual label map.

4.2.3 Discriminator Network

The discriminator improves the generated labels by sending feedback to the generator if the segmentation labels are significantly different from the ground truth. It does not increase the complexity

of the network since it is used only during training. It consists of five convolutional layers with valid padding, followed by ReLU activations and average pooling. Furthermore, two fully connected layers are placed at the end of the discriminator.

To avoid saturation, the last layer of the discriminator does not have a thresholding operator so it produces an unscaled output. Computing scores between 0 and 1 may cause the discriminator to generate values close to 0 for generated label maps, in which case the gradient would be too small to update the generator and eventually saturate the network [22].

The discriminator ($\mathcal{D}$) computes the cross-entropy of the ground truth label maps ($\mathbf{G}$) and 1, and the cross-entropy of the generated label maps ($S(\mathbf{I})$) and 0, minimizing the following loss function:

$$L_D = L_C\Big(\mathcal{D}(\mathbf{G}), 1\Big) + L_C\Big(\mathcal{D}(\mathcal{S}(\mathbf{I})), 0\Big). \tag{5}$$

During the training, the discriminator improves the segmenter network, penalizing the segmentation labels that do not look like manual labels. Therefore, the adversarial result has properties such as smoothness and robustness to inhomogeneous illumination.

4.2.4 Instance-level cell segmentation

A ResNet50 architecture [39] is applied to extract features from images. The computation is shared for all cell bounding boxes for efficiency. Generating a large number of bounding box proposals is one of the major drawbacks of algorithms such as MaskRCNN [24] and Faster R-CNN [57]. Even with feature sharing using ResNet, region-based methods computational complexity is not comparable with one-shot detection algorithms such as YOLO [56]. Unlike Mask R-CNN, DETCID uses the APRN to generate the proposal bounding boxes to reduce the number of the proposals. An average pooling is applied to the result of the ARPN for each anchor type (i.e., horizontal, vertical, and square box). If the average value is greater than a threshold t, then anchor boxes are centered at that location. The value of the threshold is determined by the size of the smallest cell in the training set.

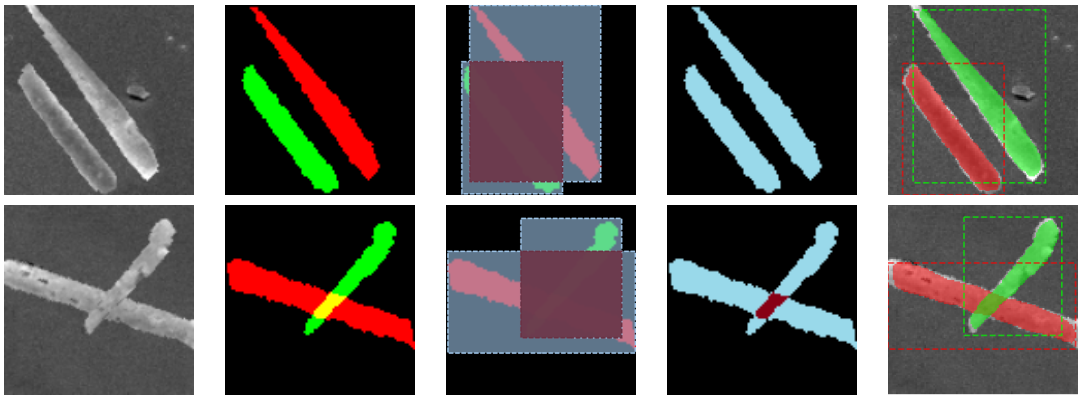


Figure 16: Visualisation of mask and bounding box overlaps in rotated touching (T), and crossing (B) cells: (a) original image, (b) ground truth mask, (c) the intersection and union of the cells using their bounding boxes, (d) the intersection and union of the cells using their masks, (e) detection results. Computing IoU using the masks distinguishes the detected candidates that correspond to different cells.

Manual annotation assigns a single bounding box to a cell. However, small variations of the bounding box in length and height may also get a high classification score by the detection method due to smoothness property. Setting all such variations to the background will be confusing for the network. Furthermore, the region proposal network may not always propose the finest bounding box around the object. Therefore, for each proposed region, similar bounding boxes with variations in length and height are passed to a fully-connected layer for refinement.

ResNet features are extracted for each anchor box and are fed to the ROI alignment. A network head is applied similar to head in Mask-RCNN to compute the masks, bounding boxes, and detection probabilities. The cells may appear in various orientations. Therefore, applying non-max suppression based on the bounding box overlaps is not suitable for the detection of cells in SEM images. DETCID modifies the computation of intersection over union (IoU) based on the masks overlaps and the area of the cell masks. Non-max suppression is then applied to the modified IoU values. Figure 16 depicts the effect of the modified IoU on the detection of rotated touching and crossing cells.

Table 3: Quantitative results of the performance of DETCID, the state-of-the-art in cell detection by Mask-RCNN, FCRN, and a shallow method by Kainz *et. al.* [28] on the acquired (UH-A-cdiff1) and the synthetic (UH-S-cdiff1) images. A 10-fold cross validation is performed on the synthetic dataset and the result is reported with a 95% CI.

Dataset	Method	Vegetative Cell		Spore	
		mAP	Dice	mAP	Dice
UH-A-cdiff1	Mask RCNN	0.52	**0.88**	**0.69**	**0.88**
	FCRN	0.41	0.60	0.13	0.46
	Kainz *et. al.* [28]	0.50	0.34	0.02	0.31
	DETCID	**0.65**	0.85	0.65	0.87
UH-S-cdiff1	Mask RCNN	0.52±0.03	0.86±0.01	0.45±0.05	**0.90±0.01**
	FCRN	0.36±0.1	0.62±0.03	0.46±0.11	0.62±0.03
	DETCID	**0.66±0.02**	**0.88±0.01**	**0.69±0.02**	0.89±0.01

The loss function includes a classification term to detect the cells and a regression term to identify the bounding box:

$$L_F = L_C(p, p^*) + p^* L_R(\mathbf{r}, \mathbf{r}^*) + L_C(\mathbf{M}, \mathbf{G}), \tag{6}$$

where, p and p^* denote the classification score and label respectively, r, r^* denote the prediction and annotated bounding box parameters, and $\mathbf{M}$ and $\mathbf{G}$ are the predicted and the ground truth masks respectively labeling the pixels as background or cell candidate.

4.3 Results

To evaluate DETCID for cell detection and segmentation, an acquired SEM dataset UH-A-cdiff1 [47], and a synthetic dataset UH-S-cdiff1 were used. Mean average precision (mAP) and dice score metrics were used to evaluate the performance of detection and segmentation, respectively.

4.3.1 Baseline comparisons

The results were compared with two state-of-the-art methods in cell detection and segmentation:

Table 4: Overall comparison of the performance of DETCID, the state-of-the-art in cell detection by Mask-RCNN, FCRN, and a shallow method by Kainz *et. al.* [28] on the acquired (UH-A-cdiff1) and the synthetic (UH-S-cdiff1) images. A 10-fold cross validation is performed on the synthetic dataset and the result is reported with a 95% CI.

Dataset	Method	Overall	
		mAP	Dice
UH-A-cdiff1	Mask RCNN	0.54	**0.88**
	FCRN	0.23	0.60
	Kainz *et. al.* [28]	0.20	0.34
	DETCID	**0.65**	0.85
UH-S-cdiff1	Mask RCNN	0.49±0.02	0.83±0.01
	FCRN	0.25±0.05	0.60±0.03
	DETCID	**0.67±0.01**	**0.88±0.01**

4.3.2 Mask-RCNN

Mask-RCNN [24] was developed by Facebook AI Research (FAIR) Lab as an instance-based object segmentation method capable of providing mask and bounding box for every object in the image. ResNet50 is used as the backbone with COCO pre-trained weights to be consistent with the backbone used for DETCID. A fully-connected region proposal network predicts rectangular bounding box regions. The regions of interest were resized and passed to the network head to compute the mask and refine the bounding box. We selected Mask-RCNN as a baseline since Mask-RCNN has achieved the state-of-the-art in instance-segmentation of cell nuclei in microscopy images [17] and segmentation of epithelial cells in pathology images with overlapping cells [36] using a TensorFlow implementation provided by Matterport which is used for comparison with DETCID [2] and is initialized with COCO pretrained weights.

4.3.3 Fully convolutional regression network (FCRN)

Fully convolutional regression network (FCRN) is a U-net based cell detection method developed by Visual Geometry Group (VGG) at the University of Oxford [78]. FCRN does not rely on a bounding box region of proposals and therefore is not sensitive to the IoU threshold for detecting overlapping cells in various orientations. The network architecture includes a convolution and a deconvolution path predicting a density map for the image. The ground truth for every cell is

defined as a 2D Gaussian where the pick is at the cell center. The label images were obtained by filtering the binary masks where the cell is white and the background is zero. The predicted local maxima in the image were found using Laplacian of Gaussian (LoG) and the watershed algorithm is used to detect the cell boundaries. We selected FCRN as a baseline because FCRN is capable of detecting cell clumps in various orientations using only synthetic training data.

4.3.4 Comparison with a shallow method trained on acquired images only

A shallow method [28] trained on real SEM images is used as a baseline to compare the performance with the deep model trained with synthetic images. The selected shallow baseline has been successfully applied to detect overlapping cells in histopathology images.

Various hand-crafted features were extracted including gradient magnitude, first and second-order gradients in x and y directions, oriented gradients, and histogram equalized intensities. Cell centers are dot-annotated and the target labels are computed as a smooth function of the inverted distance transform of the annotations. Furthermore, a regression random forest is trained on the extracted features given an image patch to predict a proximity score map.

Five-fold cross-validation is performed on 20 acquired images of UH-A-cdiff1. Feature vectors are extracted from patches of size 9×9 for both datasets. Then a random forest is trained to predict the centerline of the cells. Finally, a watershed transform is applied to compute the masks.

4.3.5 Implementation details

The implementation of the adversarial region proposal network is based on SoLiD [44] discussed in Chapter 3. Random patches of size 256×256 are passed to the ResNet50 [39] for training. The ResNet50 architrave applies scale anchors of size $\{32^2, 64^2, 128^2\}$ to extract features, and is initialized with COCO pretrained weights. To allow finer ROIs, five percent variations in length and height of the bounding boxes are considered and bounding boxes with less than five percent of potential cell areas are filtered before pooling ROIs. The above thresholds are found empirically. Adam optimization is applied to optimize the overall loss function on a cluster with 4 NVIDIA

Table 5: Depiction of the quantitative result of 10-fold cross-validation of the state-of-the-art Mask-RCNN over UH-P-cdiff1.

Fold #	Isolated		Touching		Crossing		Overall mAP	Overall Dice
	mAP	Dice	mAP	Dice	mAP	Dice		
1	0.99	0.93	0.89	0.81	0.94	0.85	0.94	0.86
2	1.00	0.94	0.91	0.83	0.94	0.86	0.95	0.87
3	1.00	0.93	0.92	0.84	0.94	0.85	0.95	0.87
4	0.59	0.92	0.93	0.84	0.91	0.83	0.81	0.86
5	0.58	0.92	0.92	0.84	0.91	0.83	0.80	0.87
6	0.78	0.93	0.90	0.82	0.89	0.81	0.86	0.85
7	0.66	0.91	0.91	0.82	0.93	0.84	0.83	0.86
8	0.64	0.92	0.89	0.81	0.92	0.83	0.82	0.85
9	0.71	0.93	0.93	0.85	0.94	0.85	0.86	0.88
10	0.58	0.92	0.91	0.83	0.93	0.85	0.81	0.87
Overall	0.75	0.93	0.91	0.83	0.93	0.84	0.86	0.86

Table 6: Depiction of the quantitative result of 10-fold cross-validation of DETCID over UH-P-cdiff1.

Fold #	Isolated		Touching		Crossing		Overall mAP	Overall Dice
	mAP	Dice	mAP	Dice	mAP	Dice		
1	0.99	0.94	0.95	0.87	0.99	0.91	0.97	0.90
2	1.00	0.95	0.95	0.88	0.99	0.91	0.98	0.91
3	1.00	0.93	0.94	0.85	0.97	0.88	0.97	0.89
4	0.99	0.93	0.97	0.89	0.99	0.91	0.98	0.91
5	1.00	0.93	0.97	0.89	0.96	0.88	0.98	0.90
6	0.99	0.93	0.98	0.90	0.98	0.90	0.98	0.91
7	0.98	0.92	0.97	0.88	0.99	0.90	0.98	0.90
8	1.00	0.93	0.97	0.89	0.99	0.91	0.98	0.91
9	0.99	0.94	0.94	0.86	0.96	0.87	0.96	0.89
10	1.00	0.94	0.98	0.90	1.00	0.91	0.99	0.92
Overall	0.99	0.93	0.96	0.88	0.98	0.90	0.98	0.90

GeForce GTX GPU of 12 GB capacity.

4.3.6 Performance comparison

UH-S-cdiff1 is used to perform 10-fold cross-validation to evaluate and compare the performance. Furthermore, UH-S-cdiff1 is used to train DETCID to evaluate on the acquired dataset UH-A-cdiff1. Mean average precision (mAP) and dice scores were computed to measure the performance of the detection and segmentation respectively. Table 3 summarizes the quantitative performance evaluations. The results indicate that DETCID outperforms the state-of-the-art in the detection of C. diff cells in the acquired SEM images UH-A-cdiff1 (*P=0.04; 95% CI 0.001-Inf*). However, DETCID achieves comparable results to the state-of-the-art in the detection of spores in UH-A-cdiff1 (*P=0.36; 95% CI-0.11-0.28*) which is more challenging since they could be misclassified as debris due to their smaller size. DETCID also achieved comparable results in segmentation of the vegetative cells (*P =0.63; 95% CI -0.071-0.11*) and spores (*P=0.15; 95% CI -0.04-0.23*).

Moreover, Table 4 indicates that DETCID achieved significant improvements in detection (*P<0.01, 95% CI -0.18-Inf*) and comparable results in segmentation (*P=0.23; 95% CI -0.03-0.13*) of UH-S-cdiff1 images where the number of touching clustered cells are higher. Therefore, DETCID outperforms the state-of-the-art in the detection of C. diff cells where touching cells are clustered together.

Figure 17 depicts qualitative comparisons between DETCID performance and the state-of-the-art. The qualitative results indicate that Mask-RCNN has better performance in the segmentation of isolated objects. However, DETCID outperforms Mask-RCNN when multiple cells are touching and in the presence of debris resulting in lower mAP. FCRN is able to separate the touching cells. However, FCRN is highly sensitive to inhomogeneous illumination and the presence of debris, resulting in poor mAP and dice score.

4.3.7 Bland–Altman analysis

A Bland-Altman analysis is performed to compute the agreement between the result of DETCID and the primary manual annotation as well as the agreement two sets of annotations for cell detection and segmentation. To evaluate the quality of the synthetic training dataset UH-S-cdiff1, an acceptance rubric is defined as the performance of cell counting on the acquired images. Figure 18 depicts the Blond-Altman plots for the number of detected cells and their segmentation masks. The Bland-Altman plots reveal no evidence of proportional bias for cell counts and segmentation differences. Furthermore, the number of detected cells by DETCID correlated with the primary set of annotations on the number of vegetative cells ($R = 0.92, P < 0.01$) and spores ($R = 0.83, P < 0.01$) compared to the correlation between the two sets of annotations of vegetative cells ($R = 0.91, P < 0.01$) and spores ($R = 0.84, P < 0.01$). The proportion of the number of spores over the number of the vegetative cell by DETCID correlated with the the primary set of annotations ($R = 0.66, P < 0.01$) compared the correlation of the ratio between the two sets of annotations ($R = 0.74, P < 0.01$). Accordingly, DETCID performance trained on the synthetic dataset UH-S-cdiff1 differs from the primary set of manual annotations in the detection of CDI cells as the two sets of manual annotations differ from themselves. Therefore, the training set UH-S-cdiff1 satisfies the acceptance rubric.

Furthermore, Table 3 depicts no significant performance difference between UH-S-cdiff1 and UH-A-cdiff1 on the detection (*P=0.85; 95% CI -0.18, 0.21*) and the segmentation (*P=0.20; 95% CI -0.11-0.03*) of CDI cells in SEM images. Therefore, UH-S-cdiff1 is a good surrogate for assessing the performance of the methods.

Finally, The area of the segmentation masks computed by DETCID also correlated with the area of the annotated masks in the primary set ($R^2 = 0.89$) compared to the area of the masks between the two set of annotations ($R^2 = 0.94$).

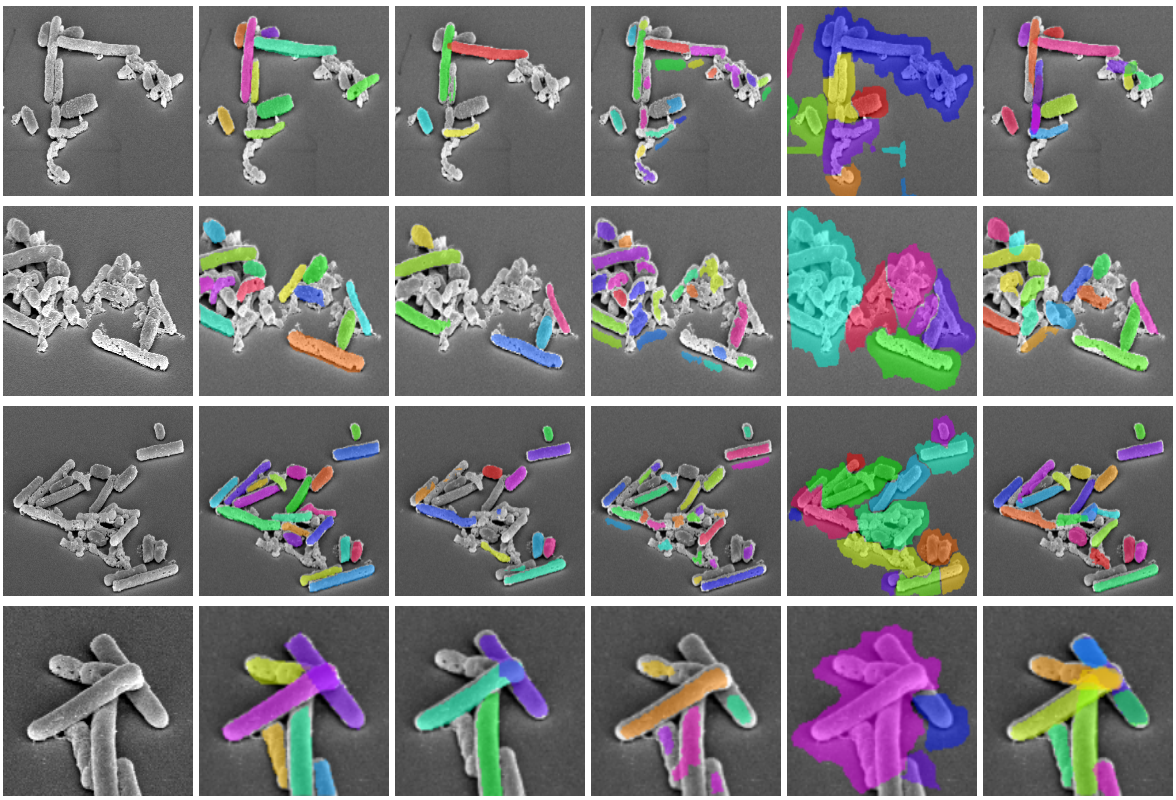


Figure 17: Depiction of the segmentation results: (a) original image, (b) ground truth labels, (c) Mask-RCNN, (d) FCRN, (e) a shallow cell detection method [28], and (f) DETCID segmentation. Mask-RCNN is more accurate in detecting isolated cells. However, Mask-RCNN does not detect cells in the presence of debris or cell clusters. FCRN is sensitive to inhomogeneous illumination and the presence of debris and results in false positives. DETCID is able to detect cells when touching cells are clustered together.

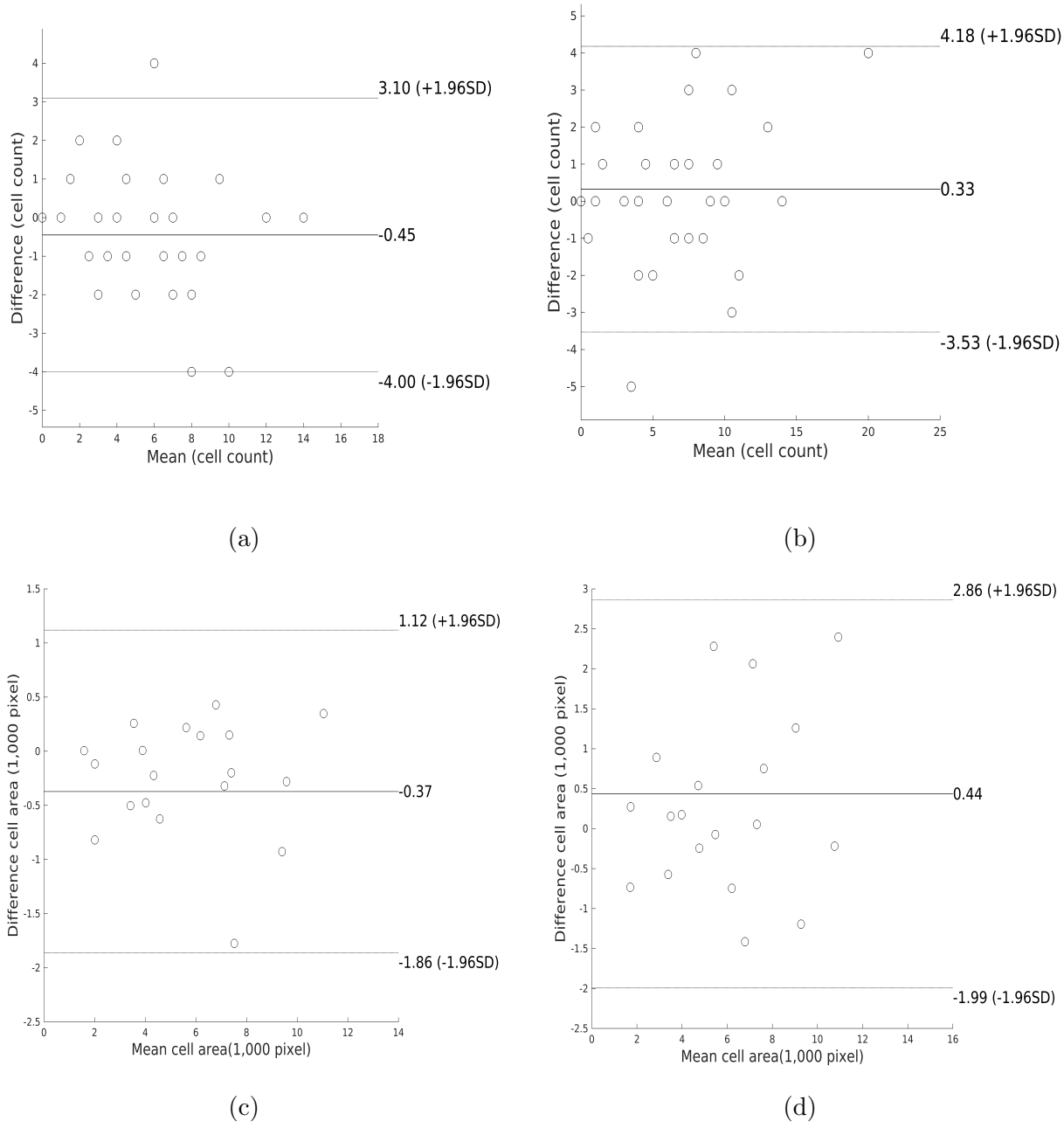


Figure 18: Depiction of Bland-Altman plots, comparing DETCID performance with the performance of the secondary set of human expert annotations: (a) the agreement between the two sets of annotations on the number of annotated cells in image, (b) the agreement between the primary set of annotations and DETCID on the number of annotated cells in image, (c) the agreement between the two sets of annotations on the annotated masks, and (d) the agreement between the annotated in the primary set of annotations and the computed masks by DETCID.

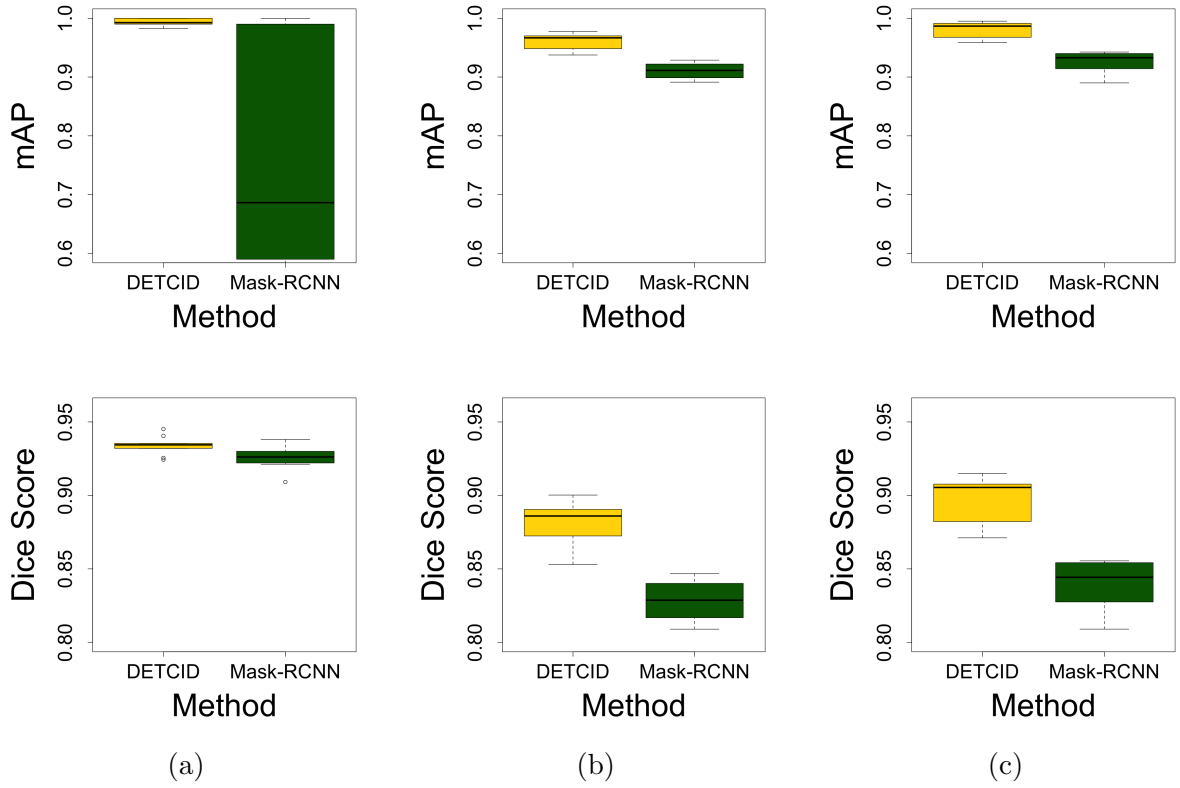


Figure 19: Visual comparison of the performance of the detection (mAP Top row) and segmentation (Dice score bottom row) between DETCID (Yellow) and Mask-RCNN (Green) for (a) isolated, (b) touching, and (c) crossing cells.

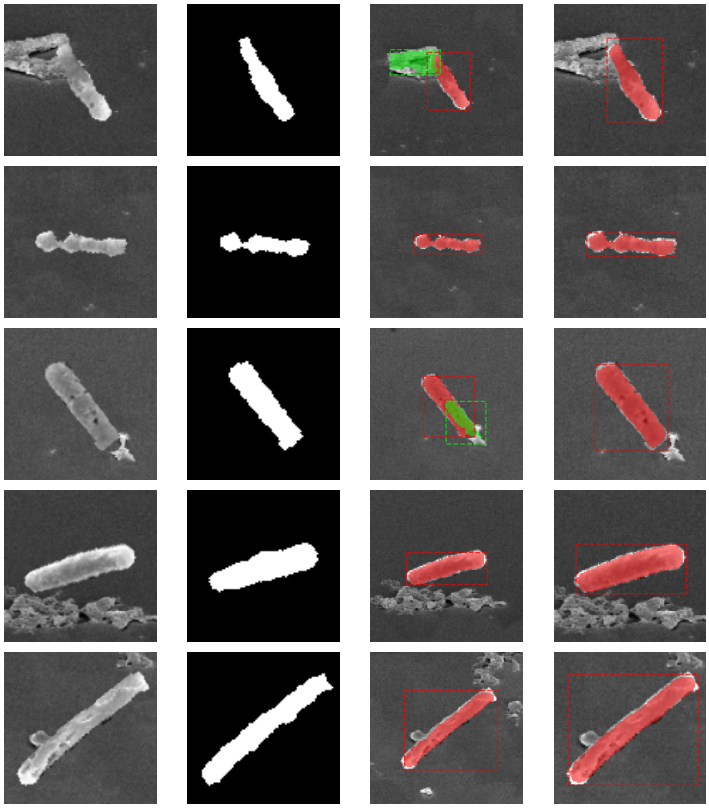


Figure 20: Depiction of segmentation results in UH-P-cdiff1 isolated cells (from left to right: Original image, Segmentation ground truth, Mask-RCNN outcome, and DETCID outcome). DETCID is able to detect isolated cells. However, Mask-RCNN is sensitive to the presence of debris and artifacts.

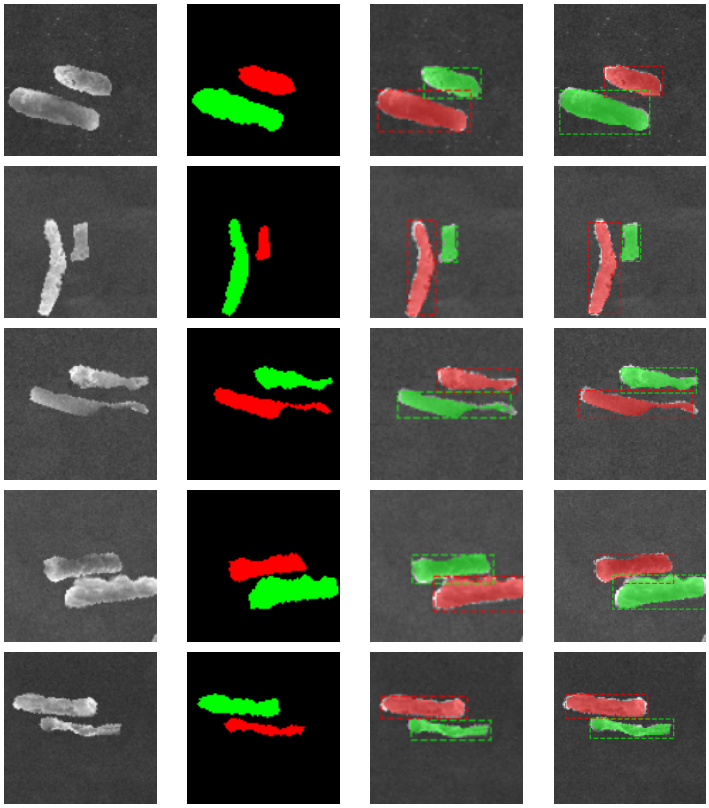


Figure 21: Depiction of segmentation results in UH-P-cdiff1 touching cells in special cases that at least one of the cells is vertical or horizontal (from left to right: Original image, Segmentation ground truth, Mask-RCNN outcome, and DETCID outcome). Applying the bounding box to compute IoU limits the performance of the algorithm to the detection of such special cases.

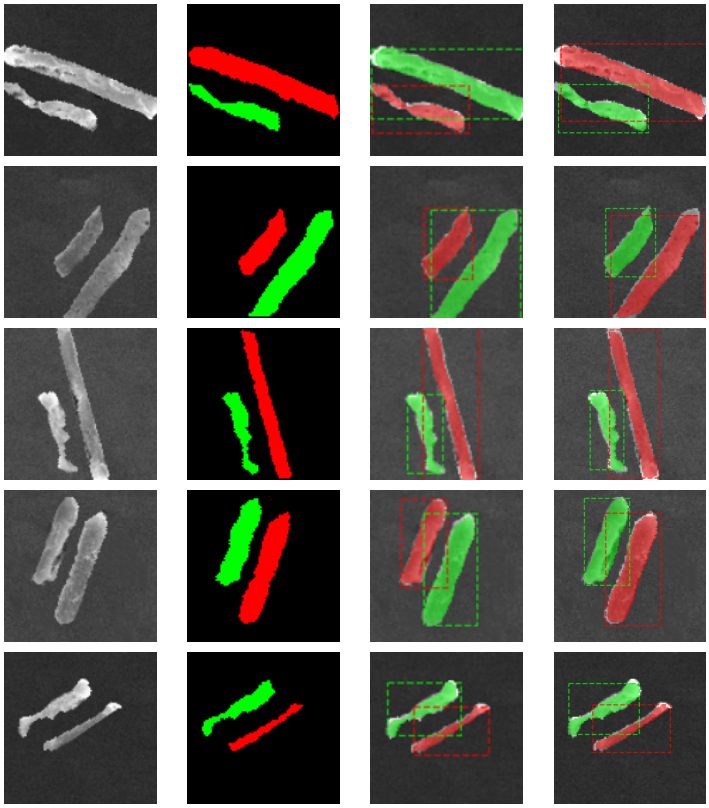


Figure 22: Depiction of DETCID segmentation results compared with Mask-RCNN in UH-P-cdiff1 touching cells in various orientations (from left to right: Original image, Segmentation ground truth, Mask-RCNN outcome, and DETCID outcome). Even though the two cells have no overlaps their bounding boxes partially intersects, reducing the detection accuracy.

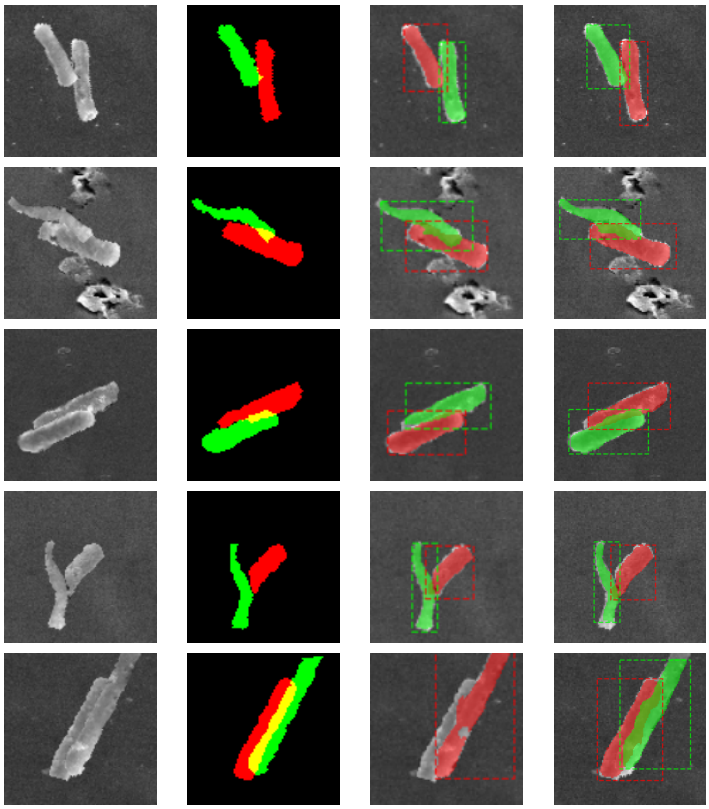


Figure 23: Depiction of segmentation results in UH-P-cdiff1 touching cells in various orientations (from left to right: Original image, Segmentation ground truth, Mask-RCNN outcome, and DET-CID outcome). Touching cells may have small overlaps (Yellow). However, their bounding boxes overlaps are significant, making the detection result sensitive to the IoU threshold. Computing IoU using the masks is a more accurate metric to estimate the overlaps of the cell bodies.

4.3.8 Cross validation on synthetic isolated, touching, and crossing cells

Cross-validation is performed based on the synthetic images in UH-P-cdiff1. Mean average precision (mAP) and dice scores were computed for each fold. Figure 19 depicts the boxplots comparing the performance of detection (mAP) and segmentation (dice score) between DETCID with MASK-RCNN. Table 5 depics the mAP and the dice scores for Mask-RCNN and Table 6 depicts the mAP and the dice scores related to DETCID performance.

Table 7 depicts the P-values comparing the difference between the performance of DETCID with state-of-the-art Mask-RCNN. The results indicated that DETCID out-performed the state-of-the-art in detection ($P - value < 0.001$) and segmentation ($P - value < 0.001$) of touching and crossing and obtained comparable performance in detection and segmentation of isolated CDI cells in SEM images.

Figure 20 depicts the segmentation results of isolating cells in UH-P-cdiff1 with inhomogeneous illumination in the presence of debris. Figure 21 depicts the segmentation results of touching cells in UH-P-cdiff1 with a narrow background between the cells in the special case that at least one cells in aligned with horizontal or vertical axes. Figure 22 depicts the segmentation results of touching cells in UH-P-cdiff1 where the cells are rotated without overlapping cell boundaries. Figure 23 depicts the segmentation results of touching cells in UH-P-cdiff1 where the cells are rotated and the cell boundaries overlap, making the detection of cells challenging due to occlusion. Furthermore, non-max suppression may remove the occluded cells in case the overlap is above the IoU threshold.

Figure 24 depicts the segmentation results of crossing cells in UH-P-cdiff1 where at least one cell in the crossing pair is aligned with horizontal or vertical axis. Figure 25 depicts the segmentation results of crossing cells in UH-P-cdiff1 where the crossing pair is rotated.

Table 7: P-values of the test for difference in mean of the performance measures between DETCID and Mask-RCNN

	Detection	Segmentation
Isolated	0.002	0.017
Touching	3.095E-007	2.906E-007
Crossing	4.484E-007	8.379E-008
Overall	3.177E-005	0.0001486

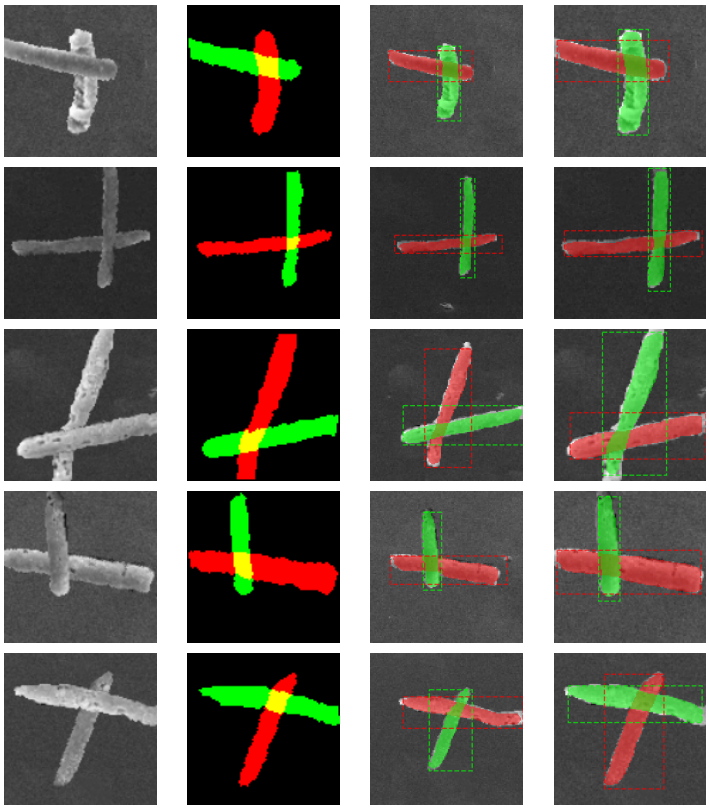


Figure 24: Depiction of segmentation results in special case crossing cells where at least one cell is horizontal or vertical (from left to right: Original image, Segmentation ground truth, Mask-RCNN outcome, and DETCID outcome). Applying horizontal or vertical bounding boxes to compute IoU limits the performance of the state-of-the-art in instant level object segmentation.

4.4 Discussion

In this Chapter, DETCID was proposed to detect and segment CDI cells in SEM images. An adversarial region proposal network was implemented to address the challenge of inhomogeneous illumination. Furthermore, a modified IoU metric is used for non-max suppression for detecting clusters of touching cells. A data augmentation algorithm was developed to provide a large number of training images suitable for training deep feature extraction architectures such as ResNet. The performance is compared to both deep region-based method and U-net based methods. DETCID outperforms the state-of-the-art in the detection of touching cells and provide comparable result in

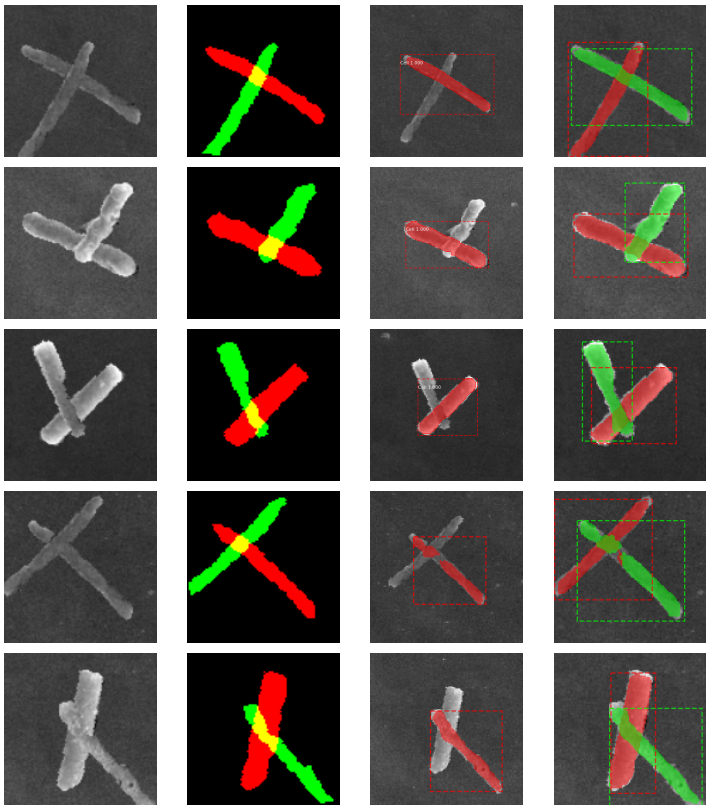


Figure 25: Depiction of DETCID segmentation results in UH-P-cdiff1 rotated crossing cells. Rotation increases the overlaps between the bounding boxes and in many cases non-max supression using bounding box IoU filters a cell increasing the false negative detections.

the segmentation of cells.

5 Shallow methods for separation of touching cells

In this chapter, we discuss the algorithms proposed and developed to address the challenge of touching *Clostridioides* difficile cells and spores in SEM images in the presence of inhomogeneous illumination. We developed DeTEC: "Detection of Touching Elongated Cells", using a hierarchy of Markov random fields (MRFs) to separate the touching cells. DeTEC was an unsupervised model. We further expanded the algorithm to its supervised version, DETCIC: "Detection of Elongated Touching Cells with inhomogeneous Illumination using a stack of Conditional random fields".

5.1 Related work

Many cell detection methods assume that the cell walls are clearly acquired by microscopy imaging and their intensity/color significantly distinguishes them from the cell body. Therefore, the cell separation is done connecting the detected cell walls [18, 28, 35]. Some methods applied an optimization algorithm on clusters of cells to identify the best cell candidates based on statistical texture and appearances [6, 7, 8, 12, 30, 63] or correlation clustering [85].

5.2 Methods

DeTEC comprises the following steps:

- Learning a cell wall probability map by a random forest regression (off-line).
- Segmentation of the image to superpixels and selection of superpixels that form candidate cell regions using an MRF applied to superpixels.
- Selection of superpixel boundary segments that form elongated cell walls using a second MRF applied to superpixel boundaries.

As a pre-processing stage, we apply a random forest regression to estimate the probability of a pixel belonging to a cell wall (Fig. 26(c)). To train the random forest, we compute a feature vector containing a set of rotation invariant local binary patterns (LBP)[53], the response of the images

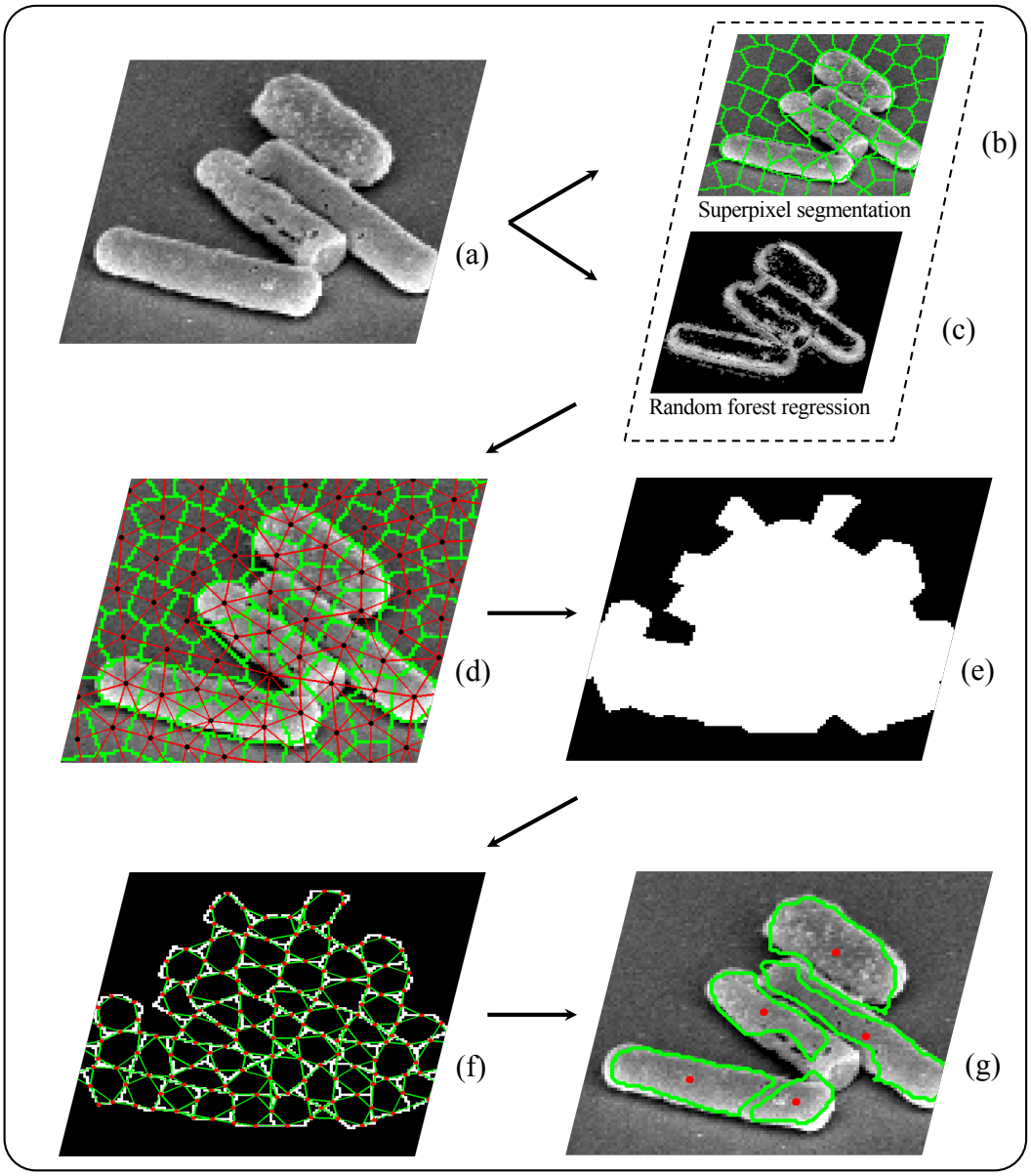


Figure 26: Overview of the two-layer hierarchical method (the figure is best seen in color). (a) A cell cluster in the original image. (b) Superpixel map and (c) cell wall probabilities predicted by random forest regression (Top). (d) A random field defined over the superpixels provides potential cell regions (the nodes are represented by black dots and the edges by red lines). (e) Output superpixel area provided by the random field in (c). (f) A second random field defined over the remaining superpixel boundaries detects elongated cells (the nodes are represented by red dots and the edges by green lines). (g) Detected centroids (red), and cell walls (green) are shown

to the difference of Gaussians of varying width ratios, and a vessel enhancement filter [20]. A few images were manually annotated to provide the labels for training the random forest.

The next step involves developing a hierarchical method based on two random fields: the first random field imposes texture smoothness while the second random field imposes smoothness on the continuity of superpixel boundary segments. At first, a cell image is divided into superpixels [50] and an MRF separates the cells from the background at the superpixel level. However, a standard MRF may not separate clustered cells.

Cell walls have a key role in the detection of cells and the separation of adjacent cells. Every superpixel boundary segment has a likelihood of belonging to a cell wall. Moreover, neighboring superpixel boundary segments are more likely to have a small variance in orientation if they form an elongated cell wall. These two observations are key-issues in DeTEC.

Algorithm 3 Detection algorithm for DeTEC

Input : Original Image
Output: Cell centroids

Compute the superpixel map.
Compute the cell wall probability map.
begin Layer 1: Detecting potential cell regions
    For every superpixel i compute feature vector $\mathbf{f_i^s}$, $i = 1, ..., n^s$.
    Apply the Gaussian mixture model on F^s space to compute parameter set $\mathscr{T}$.
    For every superpixel i compute the unary potentials as the negative log of the Gaussian probability densities with parameter $\mathscr{T}$, $i = 1, ..., n^s$.
    For every boundary component b_{ij} compute the cell wall score π_{ij}, $b_{ij} = 1, ..., n_1^b$.
    Apply graph cut to find the set of superpixel labels $\mathscr{L}$ that minimizes $E^1(\mathscr{L})$.
    For every superpixel s selected in $\mathscr{L}$ record the indexes of the boundary components $b_{ss'}$ in the adjacency matrix. s' could be any neighbor of s.
end
begin Layer 2: Elongated cell separation
    For every boundary component k, selected in the first layer compute feature vector $\mathbf{f_k^b}$, $k = 1, ..., n_2^b$.
    Apply the Gaussian mixture model on space F_b to compute parameter set $\mathscr{O}$.
    For every superpixel boundary component k compute unary potentials as the negative log of the Gaussian mixture model with parameter set $\mathscr{O}$, $k = 1, ..., n_2^b$.
    Apply graph cut to find the set of boundary component labels $\mathscr{L}$ that minimizes $E^2(\mathscr{L})$.
    Compute the centroids of the closed contours
end

5.2.1 Cell candidate detection

Algorithm 3 depicts the steps of DeTEC. The first random field is imposed onto the superpixels adjacency graph (Fig. 26(d)). A graph cut provides the binary segmentation of superpixels with the following objective function:

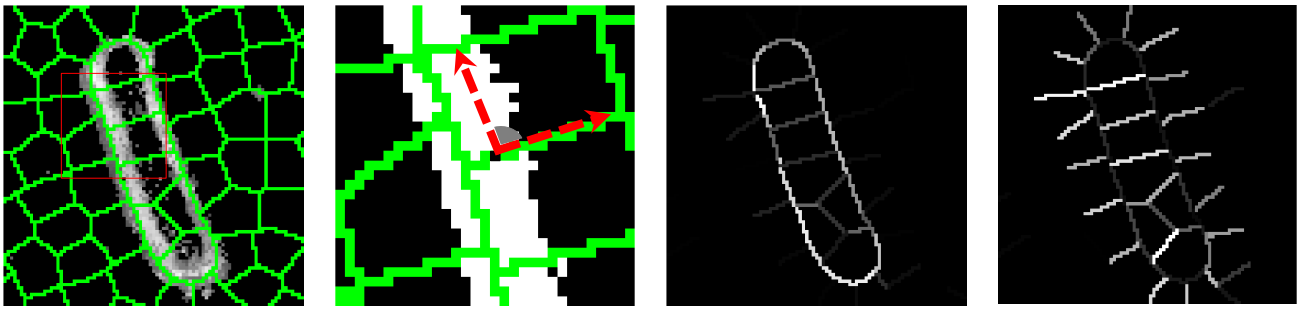
$$E^1 = \sum_i u_i^s(\mathbf{f_i^s}|\mathscr{L},\mathscr{T}) + \sum_i \sum_{j\in\mathscr{G}_i^1} v_{i,j}^s(l_i^s, l_j^s), \tag{7}$$

the first term is the sum of unary potentials u_i^s, consisting of a mixture of two Gaussians with parameter set $\mathscr{T} = \{\theta_0, \theta_1\}$, modeling the foreground and the background with superpixel label set $\mathscr{L} = \{l_i^s \in \{0,1\} | i = 1, ..., n_s\}$. The feature vector $\mathbf{f_i^s}$ comprises a set of orientation invariant LBPs, along with the mean, median, and standard deviation of pixels belonging to the i^{th} superpixel. The second term is the pairwise potential where $\mathscr{G}_i^1$ is the set of superpixel neighbors of the i^{th} superpixel.

In the standard MRF formulation, the pairwise term enforces the superpixels to have the same labels as their neighbors. However, when two cells are close to each other but not touching (e.g., they are separated by a small number of background pixels), the pairwise term forces the small background region between the two cells to be labeled as part of a cell. To avoid these false positives, we define a new pairwise penalty involving the probability of the boundary separating adjacent superpixels to be part of a cell wall [4, 83]. Therefore, we define the pairwise potential between neighboring superpixel labels l_i^s and l_j^s by:

$$v_{i,j}^s(l_i^s, l_j^s) = \begin{cases} -\log(\pi_{ij}+1) & , \quad \text{if } l_i^s \neq l_j^s \\ 0 & , \quad \text{if } l_i^s = l_j^s \end{cases}, \tag{8}$$

where π_{ij} is the probability indicating whether the boundary between the i^{th} and j^{th} superpixels is on a cell wall:



(a) (b) (c) (d)

Figure 27: (a) The superpixel map (green) is overlaid onto the cell wall probability map. (b) Zoomed visualization of the area inside the red square in (a). The cell wall probabilities are projected onto the superpixel boundary segments based on the angle between the largest connected component in the probability map (white) and the superpixel boundary segments (green). (c) The projected mean cell wall probabilities π_{ij} (14) of the image in (a). (d) The standard deviations of cell wall probabilities. The standard deviations and the projected mean of cell wall probabilities scores are used to obtain the unary potentials in the second MRF. Boundaries with high standard deviations are less likely to belong to cell walls

$$\pi_{ij} = \frac{1}{|\mathcal{N}_{ij}|} \sum_{x \in \mathcal{N}_{ij}} p_x . \cos \alpha_{ij}, \tag{9}$$

where, $\mathcal{N}_{ij}$ is the set of all pixels at the border of the two superpixels indexed by i and j, and p_x is the probability of a pixel at position x belonging to a cell wall. This value is obtained from the random forest (Fig. 26(c)).

In Eq. (14), p_x is projected onto the superpixel map; α_{ij} is the angle between the superpixel boundary component and the corresponding connected component in the probability map in a neighborhood around position x (Fig. 27). This projection ensures that a superpixel boundary receives a high cell wall score when it is parallel to a real cell wall. If the boundary component is more likely to be part of a cell wall, then the two touching superpixels are less likely to have the same labels.

This MRF model segments the cell regions from the background (Fig. 26(e)). However, when the cells are clumped together, every cluster of cells is segmented as one connected component. The

second MRF takes the boundary segments of the cell superpixels and detects a set of boundary components that are more likely to form an elongated cell wall to detect elongated cells and separate the clustered cells. The second random field considers only the selected candidate superpixels. Therefore, the number of boundary segments in the second layer is much smaller than the total number of boundary segments in the original superpixel map.

Elongated cell separation

The second random field is defined over the remaining superpixel boundary segments (Fig. 26(f)). The goal is to cluster these boundaries into two categories: boundaries that belong to elongated cell walls, and the rest of the boundaries. The energy function to be minimized is:

$$E^2 = \sum_k u_k^b(\mathbf{f_k^b}|\mathscr{L},\mathscr{O}) + \sum_k \sum_{k' \in \mathscr{G}_k^2} v_{k,k'}^b(l_k^b, l_{k'}^b). \tag{10}$$

The unary term represents the potential of the superpixel boundary component to be a part of a cell wall. Similar to the first layer, u_k^b is modeled by a Gaussian mixture model parameterized by $\mathscr{O}$ and $\mathscr{L} = \{l_k^b \in \{0,1\} | k = 1, ..., n_b\}$. The feature vector $\mathbf{f_k^b}$ comprises the mean π_{ij} (Eq. 14) and the standard deviation of the cell wall probabilities for the k^{th} superpixel boundary components (see Fig. 27.). The second term is the pairwise potential enforcing the elongation of the k^{th} boundary segment with respect to its neighbors in $\mathscr{G}_k^2$:

$$v_{k,k'}^b(l_k^b, l_{k'}^b) = \begin{cases} \cos\beta_{kk'} & \text{if } l_k^b \neq l_{k'}^b \\ 0 & \text{if } l_k^b = l_{k'}^b \end{cases}, \tag{11}$$

where $\beta_{kk'}$ is the angle between boundaries k and k'. When two adjacent boundary components have different orientations, they are less likely to have the same label. The extracted superpixel boundary components form the detected cell walls that separate the cell regions (Fig. 26(g)).

DeTEC applies a sequence of two Markov random fields (MRF) to detect touching elongated cells. The first MRF segments the cells from the background using texture features. The second MRF separates the touching cells by estimating the cell walls. However, DeTEC has the following

drawbacks:

(i) It relies only on texture features and cell wall probabilities to separate cells from their background. Since the algorithm is unsupervised, the features have the same level of importance. However, inhomogeneous illumination may alter the local texture and hence decrease the accuracy of the segmentation.

(ii) It applies a number of edge detectors to train a random forest, estimating the cell wall probabilities. However, edge detectors are not robust to noise. In case a cell is eroded due to a laboratory treatment, the random forest detects the erroneous cell walls.

(iii) Noisy estimation of cell wall probabilities leads to the poor classification of cell walls. (iv) It relies on superpixels; Inhomogeneous illumination hinders the extraction of superpixels whose boundaries match with the cell walls.

To address the mentioned drawbacks, we proposed DETCIC: "Detection of Elongated Touching Cells with Inhomogeneous Illumination Using a Stack of Conditional Random Fields", which improves the performance of DeTEC with respect to the drawbacks mentioned above.

(i) DETCIC considers shading along with texture for feature extraction.

(ii) it employs a shearlet based edge detector [31] that is robust to noise to enhance the detection of the cell wall pixels.

(iii) DETCIC applies a stack of two conditional random fields, which is a supervised method, in contrast to the MRF formulation of DeTEC.

(iv) DETCIC applies illumination normalization, reducing the effect of inhomogeneous illumination.

5.2.2 Cell separation with a stack of conditional random fields

DETCIC consists of a stack of two conditional random fields (CRF): the first CRF selects the cell candidates from the background while the second CRF separates the touching cells. Estimating the cell walls is an important step for both CRFs. This section describes how the cell walls can be estimated and how the cell wall probabilities can be applied to form the potentials of the two

CRFs.

Estimation of the cell walls

Inhomogeneous illumination hampers the detection of the cell walls. The illumination component is estimated by smoothing the original image in the logarithmic domain using a Gaussian filter. Then, the illumination normalized image is obtained by dividing the image intensities with the estimated illumination in every image $\mathbf{I}$:

$$\mathbf{I}^n = \exp\left(\log\left(\mathbf{I}+1\right) - \log\left(\mathbf{I}+1\right) * G\right), \tag{12}$$

where, G is a Gaussian filter with standard deviation σ_G. The underlying assumption in Eq (12) is the Retinex model [90] of illumination which states that an acquired image $\mathbf{I}$ is a pointwise product of illumination and reflectance. The illumination component is present mainly in coarse scales, and it can be estimated by appropriately smoothing the image. The reflectance component captures structures lying, in general, in finer scales.

The illumination normalization highlights the cell walls, reducing the effect of inhomogeneous illumination. A shearlet-based total variation method is applied to obtain the denoised image $\mathbf{D}$, retaining the cell boundaries [15].

A random forest estimates the probability of a pixel belonging to a cell wall in $\mathbf{D}$. We compute a matrix of edge detector features $\mathbf{F}^r$, including, a difference of Gaussian, a vessel enhancement filter [19], Roberts, and a shearlet-based edge detectors [31]. The first two edge detectors are selected because they create a narrow line for cell walls though they may include some noise. On the contrary, the last two features preserve the edges, which have the shape of a curve, but they cover a thicker area around the actual cell walls (Figure 28). The random forest combines all the edge detectors to provide robust boundaries representing the cell walls.

Next, a sequence of two CRFs is described in which the first CRF finds the cell candidate regions and the second CRF separates cells by estimating their cell walls.

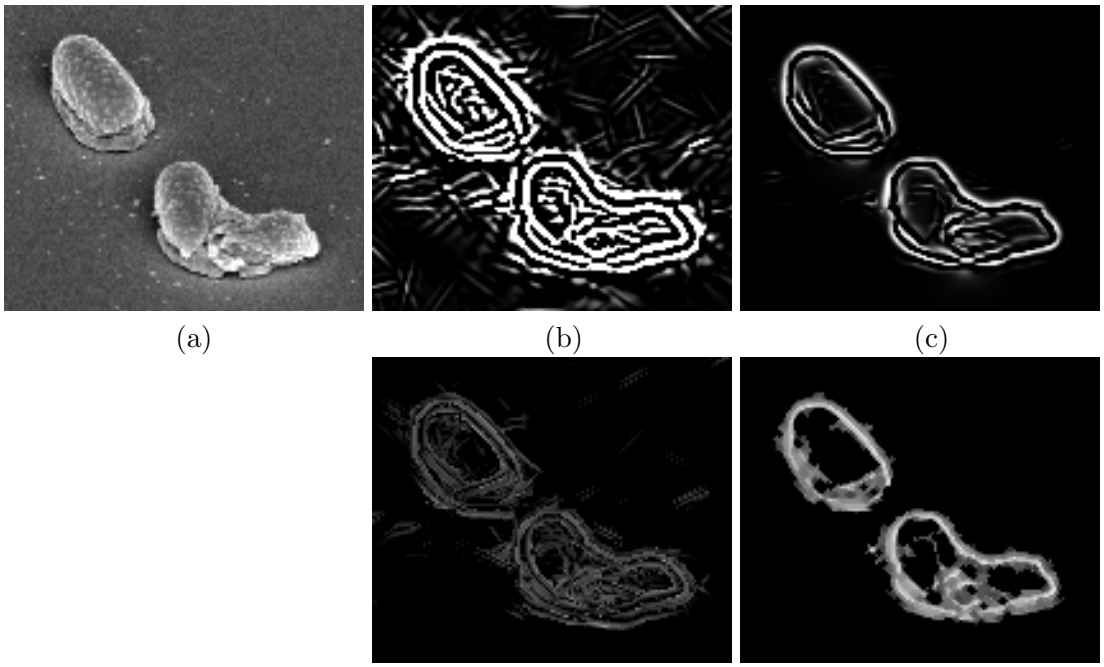


Figure 28: Depiction of edge detector features used for estimation of cell wall probabilities: (a) Original image, (b) Difference of Gaussians, (c) Application of a vessel enhancement filter [19], (d) Roberts edge detector, and (e) A shearlet-based edge detector [31].

5.2.3 Cell candidate segmentation

The denoised image $\mathbf{D}$ is divided into superpixels [50]. A CRF is applied onto the superpixels with the following objective function:

$$E^1 = \sum_t \left(\sum_i u^1_{ti}(\mathbf{f}^1_{ti}, \lambda^1_{ti}; \mathbf{w}^1) + \sum_i \sum_{j \in \mathscr{G}^1_{ti}} v^1_{tij}(\lambda^1_{ti}, \lambda^1_{tj}, \mathbf{P}^1_{tij}; \mathbf{w}^1) \right). \tag{13}$$

The unary u^1_{ti} and pairwise v^1_{tij} potentials are considered linear in the parameter $\mathbf{w}^1$. The feature vector $\mathbf{f}^1_{ti}$ contains the mean of the shading [90] and intensity values of the i^{th} superpixel.

The pairwise potential v^1_{tij} adds a penalty if the neighboring superpixels have different labels. The pairwise penalty is reduced if the boundary segment between the superpixels i and j has a high probability of belonging to cell wall:

$$\mathbf{P}^1_{tij} = \frac{1}{|\mathscr{N}^t_{ij}|} \sum_{x \in \mathscr{N}^t_{ij}} p_x . \cos \alpha_{tij}, \tag{14}$$

where $\mathscr{N}^t_{ij}$ is the set of all pixels separating the superpixels i and j in the image t of the training set, and p_x is the probability of a pixel at position x belonging to a cell wall obtained by the trained random forest. The angle α_{tij} is the angle between the superpixel boundary component (SBC) and the corresponding connected component estimated by the random forest when the cell wall probability map is superimposed on the superpixel map (Figure 27).

The first CRF separates the cell regions from the background by predicting the superpixel labels λ^1_{ti}. However, the cells may be clustered together. Thus, A second CRF is imposed onto the SBCs of the selected superpixels to estimate the cell walls and separate cells.

Algorithm 4 DETCIC training

Input : Training images, cell annotations
Output: Trained random forest, CRF weight parameters

begin
  For every image $\mathbf{I}_t$ $(t = 1, ..., n_t)$ in the training set, compute the illumination normalized image $\mathbf{I}^n_t$, shearlet denoised image $\mathbf{D}_t$, superpixel map $\mathbf{S}_t$, and edge detector feature map $\mathbf{F}^r_t$.
  Given the feature map $\mathbf{F}^r$ train a random forest to estimate the cell wall probability $\mathbf{P}^1$
  Given $\mathbf{P}^1$ and $\mathbf{S}_t$ train the first CRF on superpixels, minimizing E^1 to obtain weights $\mathbf{w}^1$
  For every $\mathbf{S}_t$ $(t = 1, ..., n_t)$, extract SBCs that belong to a cell wall.
  Train the second CRF on SBCs, minimizing E^2 to learn the weights $\mathbf{w}^2$.
end

5.2.4 Elongated cell separation

The second CRF is defined over the SBCs extracted from the first CRF. The objective function aims to select SBCs that are probable to belong to a cell wall and are elongated:

Algorithm 5 DETCIC inference

Input : A new image $\mathbf{I}_d$, the parameters of the random forest and CRFs
Output: Cell centroids

begin
  For the cell image $\mathbf{I}_d$, compute the illumination normalized image $\mathbf{I}_d^n$, the shearlet denoised image $\mathbf{D}_d$, the superpixel map $\mathbf{S}_d$, and the edge detector features $\mathbf{F}_d^r$
  Input $\mathbf{F}_d^r$ to the trained random forest to compute the cell wall probability map $\mathbf{P}_d$
  Given $\mathbf{P}^1$, $\mathbf{S}_d$, and $\mathbf{w}^1$, apply graph cut to obtain a segmentation on superpixels.
  Extract the SBCs from the selected superpixels.
  Given $\mathbf{P}^2$, and $\mathbf{w}^2$, apply graph cut on SBCs to estimate cell walls.
  Use the estimated cell walls to create morphological connected components.
  Compute the cell centroids.
end

$$E^2 = \sum_t \left(\sum_q u_{tk}^2(\mathbf{f}_{tq}^2, \lambda_q^2, \mathbf{w}^2) + \sum_q \sum_{r \in \mathscr{G}_{tq}^2} v_{tqr}^2(\lambda_q^2, \lambda_r^2, \mathbf{f}_{tq}^2, \mathbf{f}_{tr}^2, \mathbf{B}_{tqr}, \mathbf{w}^2) \right). \tag{15}$$

Similar to the first CRF, the unary and the pairwise terms are linear combinations of features and weight parameters that minimize the energy function E^2. The unary feature vector $\mathbf{f}_{tq}^2$ includes the mean and standard deviation of the cell wall probabilities $\mathbf{P}_{tpq}^2$. The pairwise feature vector includes the difference between the two unary features and the cosine of the angle $\mathbf{B}_{tqr}$ between SBCs q and r. The pairwise potential v_{tqr}^2 penalizes the objective function if the predicted labels λ_q^2 and λ_r^2 are different. However, the penalty is reduced if the two SBCs have different unary features or do not form an elongated structure.

5.2.5 DETCIC training and inference

The DETCIC training set includes images $\mathbf{I}_t$ $(t = 1, ..., n_t)$, which are annotated manually. Cell wall labels to train the random forest are the boundaries of the annotations. Moreover, the CRF objective function E^1 is trained with the superpixel label set $\mathscr{L}_t^1 = \{l_{ti}^2 \in \{0, 1\} | i = 1, ..., n_s\}$, where n_s is the number of superpixels in the image. The first CRF selects superpixels that are likely to

belong to a cell. The second CRF is trained with the label set $\mathscr{L}_t^2 = \{l_{tp}^3 \in \{0,1\} | p = 1, ..., n_b\}$, where n_b is the number of SBCs extracted from the cell candidate superpixels in the image t in the training set. Label sets $\mathscr{L}_t^1$ and $\mathscr{L}_t^2$ are computed from the manual annotations. Algorithm 4 outlines the training steps for both CRFs. A graph cut provides the labels for each CRF while a gradient-based optimization method selects the best parameter configuration $\mathbf{w}$ that minimizes the objective function E.

Algorithm 4 learns the parameters $(\mathbf{w}^1, \mathbf{w}^2)$. Given a new image I_d, computing the cell wall probabilities $\mathbf{P}_d$ requires computing the illumination normalized image $\mathbf{I}_d^n$ and denoised image $\mathbf{D}_d$ similar to the training images. Then, DETCIC performs two graph cuts: the first is applied to a rough segmentation of the cells from the background and the second is applied to the SBCs to determine the cell walls. Algorithm 5) depicts the steps for DETCIC inference.

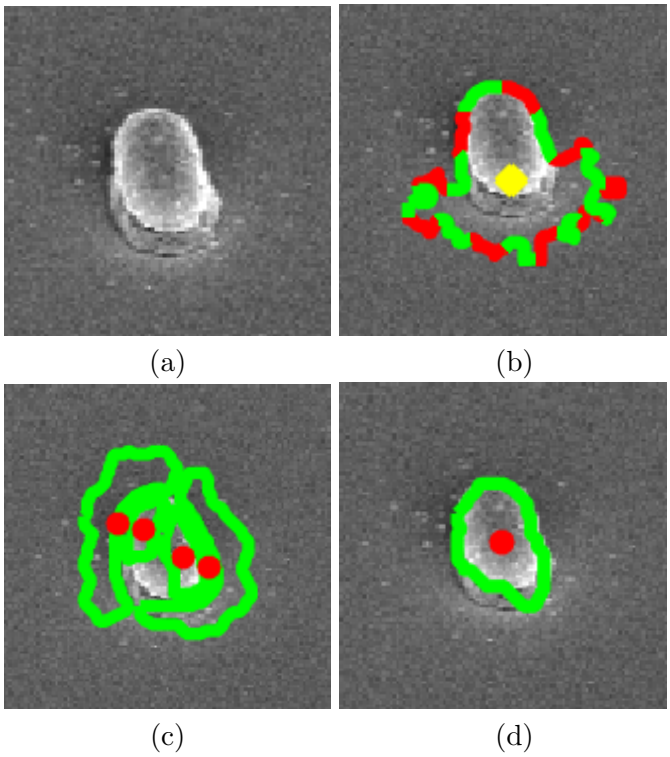


Figure 29: Depiction of the effect of inhomogenous illumination: (a) Original image, (b) CellDetect [8], (c) DeTEC [45], and (d) DETCIC.

Table 8: Comparative results between DETCIC, DeTEC [45], and CellDetect [8], where the acceptable distance of detected centroids from the ground truth is set to the length of the major axis of the smallest cell in the dataset.

Method	Precision	Recall	F-score
CellDetect	**0.80**	0.23	0.36
DeTEC	0.50	**0.88**	0.63
DETCIC	0.68	0.83	**0.75**

5.3 Results

To evaluate the detection, cell centroids were computed from the annotation providing the ground truth. A cell is considered to be detected if the detected centroid lies within a distance d from the ground truth. The distance is set to the length of the smallest cell in the dataset. Precision, recall, and F-score were computed to measure the performance of detection.

Table 8 provides a comparison of the performance of DETCIC with CellDetect and DeTEC. The training was based on leave-one-out cross-validation. CellDetect is a supervised region-based cell detection method which applies extremal regions to detect candidate cell regions [43]. Then, a statistical model selects the best extremal regions. However, CellDetect fails to detect a fair amount of cells, assuming there should exist some extremal regions that can represent the cells [7]. Therefore, CellDetect achieves a lower recall index compared to the other two methods. DeTEC is an unsupervised region-based method that applies an MRF to segment the cell candidates, and a second MRF to separate the best cell walls to detect the centroids. Although DeTEC detects most cells, the detected cell walls are sensitive to erosion which may be caused by a pharmaceutical treatment. Therefore, some cells are broken into smaller pieces, increasing the number of false positives which leads to low precision. DETCIC significantly improves the cell breakdowns due to a better estimation of cell wall probabilities which are used to train the second CRF.

Figure 29 depicts an instance where inhomogeneous illumination created shadows on the cell body as well as the area around the cell. CellDetect falsely includes shadows around the cell as part of the cell body. Furthermore, the shadow on the cell body creates two bright sides on both

sides of the cell. DeTEC considers these sides as separate cells and fails to detect the entire cell. However, DETCIC is able to reduce the effect of illumination and detect the cell wall accurately.

Figure 30 depicts examples of detected cells. CellDetect does not detect many cells while failing to separate clusters of touching cells. On the contrary, DETCIC is able to detect most cells. However, a few cells are missing due to large shadows which make the cells merge into the background.

DeTEC is able to detect most cells or a portion of them. However, DeTEC fails to estimate the correct boundaries in many cases. Also, DeTEC fails to distinguish between cells and small background regions surrounded by cells due to its unsupervised nature.

Furthermore, DeTEC is more sensitive to inhomogeneous illumination compared to DETCIC. More specifically, DeTEC fails to clearly distinguish between cells and background in image regions where cell walls are covered by shadows. Figure 29 depicts the detection of a cell affected by inhomogeneous illumination.

5.4 Discussion

In recent years deep ConvNets outperformed shallow methods in computer vision. However, shallows methods are useful in the early stages of designing deep architectures. For instance, shallow methods point out the challenges of a real-world problem with minimum data acquisition and annotation effort which is critical in preparing the training set, designing the deep architecture, and choice of loss terms. In this Chapter two shallow methods namely, DeTEC (unsupervised) and DETCIC (supervised) were developed and evaluated to detect CDI cells in SEM images with a sequence of two random fields. Both methods outperformed the shallow methods in the literature.

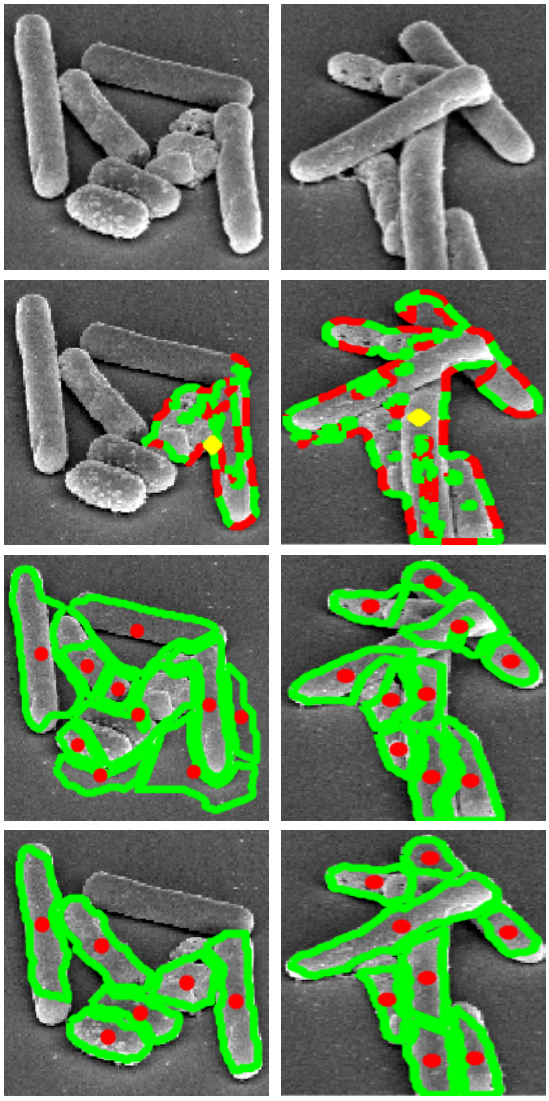


Figure 30: Depiction of the detected cell centroids and their estimated cell walls, from top to bottom: Original image, CellDetect [8], DeTEC [45], and DETCIC [47].

6 Addressing the challenge of rotational invariance

Elongated CDI cells are in various orientations which is a major drawback for deep ConvNets, making the segmentation more challenging. Deep neural networks are able to segment a variety of objects (e.g., cars, pedestrians, animals) that do not tend to change their vertical or horizontal poses. A primitive solution to address rotation invariance is to acquire images of rotated objects and train a deeper network with larger tensor volumes. However, training such networks considering all possible orientations of the same objects, dramatically increases the computational complexity. Deep models in biomedical image analysis (e.g., fully convolutional networks [41, 65], and U-net [58]) have not considered the segmentation of elongated objects with orientation invariance.

6.1 Related work

Rotation changes the order of the input feature. Sending appropriate features to the next layer in the network is called feature routing. Recently, CapsNet [59, 60] and Harmonic network [76] proposed feature routing models with rotation invariance. However, feature routing is hindered by the image resolution. Computing the appropriate features need an extra loop inside each epoch. Image resolution increases the number of capsules in each layer and thus the number of part-whole connections between layers. Furthermore, every capsule is connected to all the capsules in the next layer. Therefore, increasing image resolution increases the computations between capsule until routing convergence.

The learned representation is passed to a decoder to segment instances of two cell types (i.e. vegetative and spore) in various orientations based on their shape. The segmentation masks provided by RISEC could be used to quantify the shape information of cells such as the length of the major axis to quantify the efficacy of the treatments. Such quantifiable measurements provide a unique opportunity for clinical research where the treatments are developing (specifically CDI infection) to compare the efficacy of the treatments.

6.2 Methods

RISEC consists of two parts: a region proposal network and a segmentation network. The region proposal network selects the potential cell regions and models the inhomogeneous illumination as an adversarial variation. The segmentation network learns a shape representation via dynamic routing [59] and predicts the cell type and the corresponding segmentation mask. Figure 31 depicts the architecture of the pipeline.

6.2.1 Region proposal network

The region proposal network (RPN) is similar to U-net with an adversarial loss. The adversarial loss models the illumination as an adversarial attack and gives feedback to RPN to improve the segmentation.

The RPN consists of six convolutional units: the first three units include a 3x3 convolution layer, a ReLU layer and a 2x2 max pooling layer with a stride of two, downsampling the image. The next three units include an upsampling of the features followed by a 2x2 deconvolution. RPN loss function is the cross-entropy loss combined with an adversarial loss:

$$L_S = \mathbf{w_c} * L_C\Big(\mathcal{S}(\mathbf{I}), \mathbf{G}\Big) + L_C\Big(\mathcal{D}\big(\mathcal{S}(\mathbf{I})\big), 1\Big), \tag{16}$$

where $L_C\big(\mathcal{S}(\mathbf{I}), \mathbf{G}\big)$ denotes a cross-entropy term between the predicted labels $\mathbf{S}$ corresponding to the image $\mathbf{I}$ and the ground truth $\mathbf{G}$. Since cell areas are smaller than the background portion of the image, the segmenter cross-entropy loss is weighted by $\mathbf{w_c}$. The minority class has a higher weight in the loss function. The second term $L_C\big(\mathcal{D}\big(\mathcal{S}(\mathbf{I})\big), 1\big)$ is the adversarial loss term, computed by the discriminator. The label map of image $\mathbf{I}$ generated by the RPN is denoted by $\mathcal{S}(\mathbf{I})$ and $\mathcal{D}$ denotes the discriminator network [44]. The adversarial loss penalizes the RPN to produce label maps similar to the ground truth, reducing the effect of the illumination. The discriminator is a conventional ConvNet classifier trained on the ground truth and predicted segmentation provided by the RPN. The discriminator improves the generated labels by sending feedback to the generator if the RPN

labels are significantly different from the ground truth. It does not increase the complexity of the network since it is used only during training. It consists of five convolutional layers with valid padding, followed by ReLU activations and average pooling. Furthermore, two fully connected layers are placed at the end of the discriminator. The discriminator $\mathcal{D}$ computes the cross-entropy of the ground truth label maps $\mathbf{G}$ and 1, and the cross-entropy of the generated label maps $\mathcal{S}(\mathbf{I})$ and 0, minimizing the following loss function:

$$L_D = L_C\Big(\mathcal{D}(G), 1\Big) + L_C\Big(\mathcal{D}(\mathcal{S}(\mathbf{I})), 0\Big). \tag{17}$$

During the training, the discriminator improves the RPN, penalizing the candidate regions that do not look like manually label areas (e.g., with misclassified shadows on top and bright areas around the cell). Therefore, the adversarial training reduces the effect of the inhomogeneous illumination. A fixed window size is defined that can contain the largest cell in our dataset. The RPN provides a primary segmentation of images of any size larger than the window size. The primary segmentation is then filtered to select the candidate regions where the sum of the pixel probabilities is above a threshold t_c. Non-max suppression is applied to reduce the number of candidate regions.

6.2.2 Dynamic routing for rotational invariant segmentation

ConvNets are sensitive to object transformation such as rotation since they change the order of the low-level features [59]. Feature routing is the process to put such features to the appropriate order. Max-pooling is a primitive feature routing to overcome translation. However, max-pooling hinders the equivariance property of the network. Therefore, local information about object shape and pose are lost throughout the network, resulting in the ConvNet to be sensitive to the transformation of the object.

To accomplish rotation invariance, dynamic feature routing [59] is applied to put the rotated parts (features) together to create the shape of an object. If part i belongs to object j then i and j are coupled together with a coupling coefficient c_{ij} close to one where $c_{ij} \in [0, 1]$. A two-layer

Algorithm 6 Depiction of RISEC Training

Input : Augmented training images, Segmentation ground truth, and class labels
Output: Trained network

begin
  for *number of pretraining iterations* **do**
    Select a batch of labels $\mathbf{G}$
    Train the discriminator with the cross-entropy loss $L_C\big(\mathcal{D}(\mathbf{G}), 1\big)$
  end
  for *number of adversarial iterations* **do**
    Select a batch of training images and their labels $\{\mathbf{I}, \mathbf{G}\}$
    Predict the segmentation $\mathcal{S}(\mathbf{I})$. Compute the segmentation cross-entropy loss $L_C(\mathcal{S}(\mathbf{I}), \mathbf{G})$
    Compute the adversarial loss $L_C\big(\mathcal{D}(\mathcal{S}(\mathbf{I})), 0\big)$
    Given the labels $\mathbf{G}$, compute the discriminator cross-entropy loss $L_C\big(\mathcal{D}(\mathbf{G}), 1\big)$
    Compute $L_{\mathcal{D}}$ and backpropagate the gradients through the discriminator and the segmenter.
    Compute $L_{\mathcal{S}}$ and backpropagate the gradients through the segmenter.
  end
  For all i in the first capsule layer and j in the second capsule layer: Set $b_{ij} = 0$
  for *number of dynamic routing iterations* **do**
    Compute $c_{ij} = softmax(b_{ij})$
    Compute $\mathbf{v_i}$ (Eq. 18) and $\mathbf{v_{j|i}}$ for every (i, j) (Eq. 19)
    Compute $\mathbf{s_j}$ (Eq. 20) and $\mathbf{v_j}$ (Eq. 18)
    Compute the agreement a_{ij} between capsules (i, j) (Eq. 21)
    Update $b_{ij} \leftarrow b_{ij} + a_{ij}$
    Compute the loss L_R (Eq. 22) and backpropagate the gradients to update the tensor $\mathbf{W_{ij}}$
  end
end

capsule architecture is used in which the first layer capsules represent the local pose information and the second layer capsules represent the overall shape and orientation of a cell, based on the overall agreement between the local pose information. First, two layers of convolution are applied to a candidate region. Then, the primary capsule layer is formed by reshaping the output of the convolution volume so that every bloc (e.g., $8 \times 8 \times 1$) in the convolution volume represent a 64×1 capsule vector s_i. The length of the vector $\mathbf{v_i}$ is then squashed to $[0, 1]$:

$$\mathbf{v_i} = \frac{\|\mathbf{s_i}\|^2}{1 + \|\mathbf{s_i}\|^2} . \frac{\mathbf{s_i}}{\|\mathbf{s_i}\|} . \tag{18}$$

Next, the likelihood $\mathbf{v_{j|i}}$ of capsule j in the second capsule layer, representing a vegetative cell or a

spore, is estimated as:

$$\mathbf{v_{j|i}} = \mathbf{W_{ij}}.\mathbf{v_i}\,, \tag{19}$$

where the weight matrix $\mathbf{W_{ij}}$ is learned during training. Then, the weighted sum of inputs i over capsule j is computed as:

$$\mathbf{s_j} = \sum_i c_{ij}.\mathbf{v_{j|i}}\,, \tag{20}$$

where c_{ij} is the coupling coefficient, acting as a prior, for the likelihood $\mathbf{v_{j|i}}$.

The coupling coefficient c_{ij} is defined as the softmax of b_{ij}, where b_{ij} is computed as the cumulative sum of agreements between capsules over routing iterations. In every routing iteration, the agreement a_{ij} is added to the log prior b_{ij} until convergence. The agreement a_{ij} between two capsules is defined as the scalar product of the squashed mean v_j and the likelihood $\mathbf{v_{j|i}}$:

$$a_{ij} = \mathbf{v_j}.\mathbf{v_{j|i}}\,. \tag{21}$$

Alternatively, a clustering algorithm could be applied over each capsule layer and define the proximity of a capsule vector to the mean of the cluster as the agreement [60]. Algorithm 1 depicts the training steps of RISEC.

A decoder is added to the output of the second capsule layer to compute the segmentation mask by increasing the resolution of the feature map using two deconvolution steps. Finally, the routing loss function,

$$L_R = L_M + L_C(\mathbf{I}, \mathbf{G})\,, \tag{22}$$

consists of two terms: classification loss L_M and $L_C(\mathbf{I}, \mathbf{G})$ cross-entropy loss of the segmentation. The classification loss L_M enforces that the output vector of capsule representing class k has a large length if and only if an object of class k is present:

$$L_M = \sum_k t_k.max(0, m^+ - \|\mathbf{v_k}\|)^2 + \lambda(1 - t_k)max(0, \|\mathbf{v_k}\| - m^-)^2 \tag{23}$$

Table 9: Comparative results between the segmentation performance of RISEC and the state-of-the-art in biomedical instance segmentation by U-net and CapsNet.

Method	Dice (Vegetative)	Dice (Spore)	Dice (Total)	AUC
CapsNet [59]	0.56	0.68	0.62	0.80
U-net [58]	**0.86**	0.60	0.73	0.94
RISEC	0.78	**0.83**	**0.81**	**0.97**

where $t_k = 1$ indicates the presence of object of class k and m^+, m^+, and λ denote the hyper-parameters of the model. The lambda parameter prevents the initial learning from shrinking the activity vector for the absent classes.

6.3 Results

6.3.1 Dataset

UH-S-Cdiff0 a dataset of CDI cell images acquired via SEM imaging with pixel dimensions 411×711 and 10,000x magnification with two CDI cell types [47], is applied to train and validate RISEC, with two classes of vegetative cell and spore. A training set of synthesized images is created by synthesizing background images of size 150×150. Then, cells were randomly selected and placed into the center of the image. The cells were slightly warped to generate variant images. A dataset of 12,000 samples is divided into a training set with 1,000 images (5,000 samples of each cell type) and a validation set with 2,000 images (1,000 samples for each cell type). The same training set is applied to train RISEC, and our baselines, U-net, and CapsNet. The validation set is not seen by the network during the training and includes cells in various random orientations.

6.3.2 Performance evaluation

Table 9 summarizes the segmentation performance of RISEC with the results of U-net and CapsNet. Figure 33 depicts the ROC curve of RISEC with CapsNet and U-net. The dice scores were computed to measure the segmentation performance. Figure 34 depicts sample results of the segmentation. U-net produces accurate boundaries but is highly sensitive to illumination and artifacts in the

image.

6.4 Discussion

The black-box nature of deep ConvNets has been one of the major drawbacks in areas where required clear explanations about the features used in the decision-making process such as medical and financial applications. Therefore, learning a parametric representation of detected objects is important. Capsule networks have been successfully applied to learn parametric representations on MNIST dataset for digit recognition. However, the application of capsule-based architectures is limited to images with small sizes due to their computational complexity. In this Chapter, a capsule-based parametric model was developed and evaluated to detect CDI cells in various orientations that is capable of detecting cells in images with size 150×150. An adversarial region proposal network is applied on images to separate the cells from the background a two-layer capsule architecture is applied to segment and classify the cells in various orientations. The result indicated an 11 percent improvement of dice score compared to U-net in the segmentation of CDI cells in SEM images.

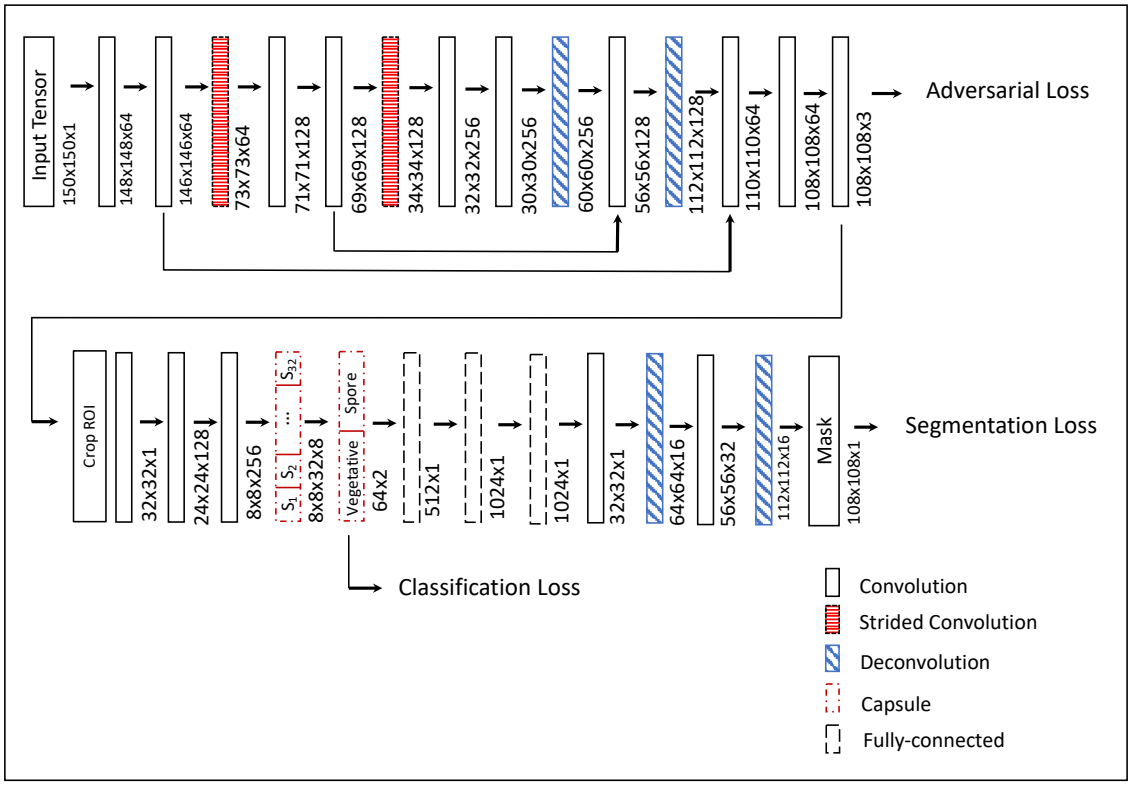


Figure 31: Depiction of RISEC pipeline. First, the adversarial segmenter network separates the potential cell areas. Then, regions of interest are passed through two layers of convolution. The resulting volume is reshaped to form capsules, representing the local shape properties of the cells. The capsules in the primary layer are fully connected to the capsules in the secondary layer, performing dynamic feature routing. The secondary capsule layer learns two shape representation for vegetative cells and spores. Finally, the representation is passed to a decoder to provide the segmentation.

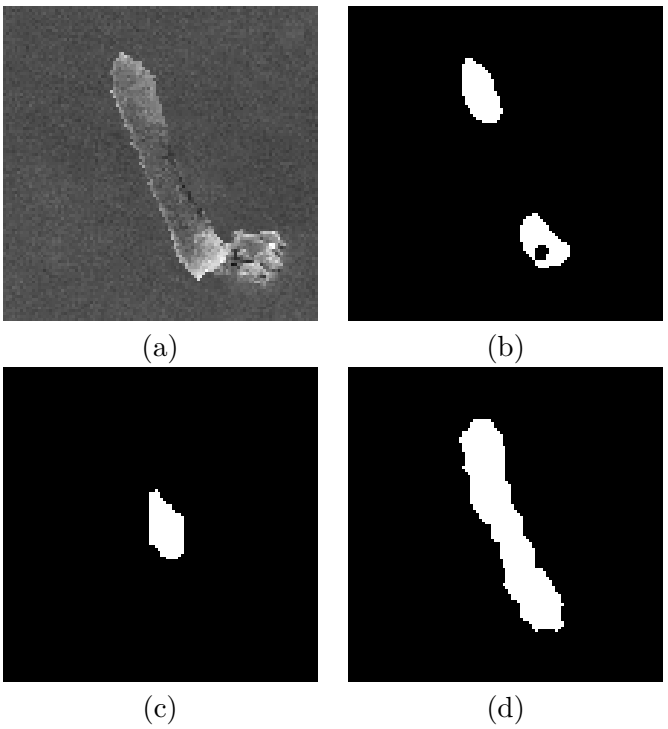


Figure 32: Effect of inhomogeneous illumination is depicted. (a) Shadows and bright spots have divided a cell into different parts. (b) U-net detected a vegetative cell as two spores since different parts of the cell are inhomogeneously illuminated. (c) CapsNet is able to detect the rotation of the cell but fails to segment the entire cell. (d) RISEC has inferred that inhomogeneously illuminated parts are likely to belong to a single vegetative cell based on their shape and orientation. The detected boundaries could be improved.

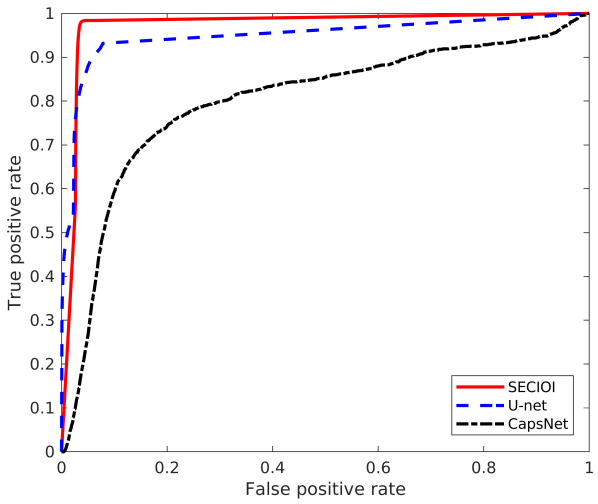


Figure 33: The ROC curve indicates that RISEC outperforms U-net [58] and CapsNet [59] in segmenting the cells.

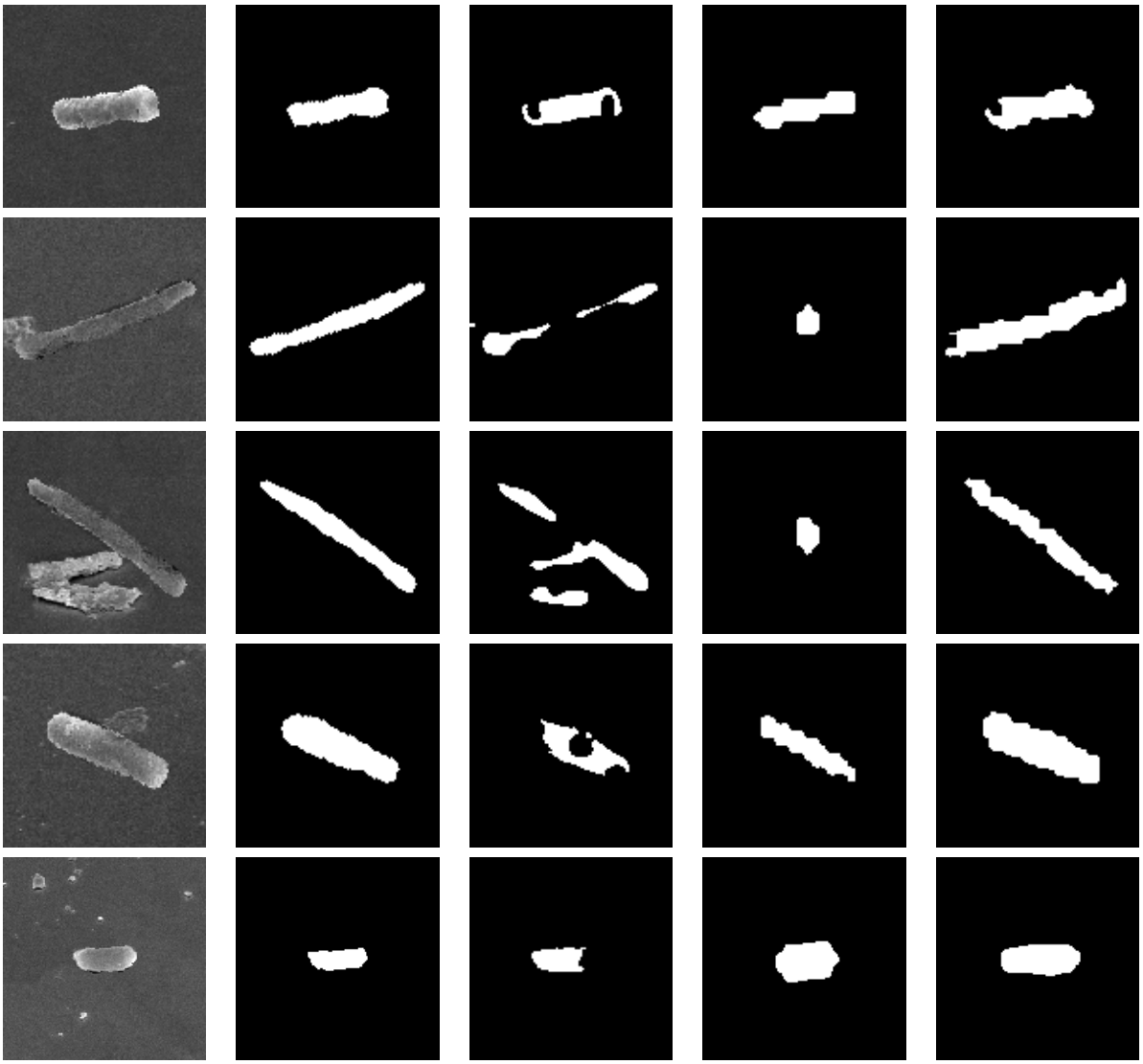


Figure 34: Qualitative depiction of RISEC segmentation results compared to the results from U-net, and CapsNet. From left to right: Original image, ground truth, U-net, CapsNet, RISEC. Inhomogeneous illumination results in partial segmentation of objects.

7 Conclusions and future work

7.1 Conclusions

In this thesis, a series of methods were proposed to detect CDI cells in SEM images. A computational tool is proposed, developed, evaluated, and compared with state-of-the-art to provide instance-level-segmentation masks for the detection of CDI cells in SEM images. The computational tool is necessary for the analysis and quantification of the efficacy of treatments on CDI cells in SEM images.

To obtain a large synthesized training set, the image synthesizer algorithm, ISABC, was developed to provide synthetic images with isolated, touching, and crossing cells with SEM backgrounds needed to train and evaluate the deep ConvNets used in the pipeline.

Furthermore, to overcome the challenge of inhomogeneous illumination, a deep adversarial network is developed to segment the cell candidate regions from the background. A discriminator network improves the result of the segmenter network without increasing the complexity during testing. The performance of semantic segmentation was improved by 44 percent compared to U-net.

To address the challenge of touching cells, a deep ConvNet is applied to learn shared features from the candidate regions. The features are aligned and fed to fully connected layers for classification and bounding box prediction. A network head including layers of convolution and a deconvolution layer is applied to produce the mask. Non-max suppression is applied to filter the duplicated detected cells. While the state-of-the-art in object detection and instance-level segmentation relies on bounding boxes to compute IoU values, this work applied the masks overlaps to compute the IoU. The modified IoU metric is capable of detecting cluttered touching and crossing cells in various orientations. The overall detection performance was improved by at least 20 percent compared to the state-of-the-art. A Bland-Altman analysis is performed to compare the result with manual annotations. The detection results correlated with the primary set of annotations similar to how the primary and secondary set of annotations correlated with each other, indicating that the method has the same error as human-level performance in the detection of CDI cells in SEM

images. The modified IoU could be used as an additional loss term to improve the detection and segmentation of the clustered cells.

7.2 Future work

Deep instance-level object segmentation models are divided into two categories: region-based and Unet-based models. Variations of U-net are proposed for the segmentation of biomedical objects with less challenging datasets and fewer classes of objects present in the image [55, 79, 23]. Region-based methods are more accurate in segmentation since their deeper architecture allows for learning more complex functions [36]. However, region-based rely on applying a large number of region proposals, resulting in multiple proposals per object. Non-max suppression is conventionally applied to remove the redundant proposal but in case of overlapping objects, non-max suppression is sensitive to the IoU threshold.

The future works could contribute to the problem of instance-based cell segmentation by the following objectives:

1. Reduce the false positive in cell detection due to occlusion or presence of debris

2. Improve the segmentation of overlapping cells by adding a penalty on detected cell overlaps

The modified IoU developed in this thesis could be used to improve the segmentation of clustered cells via penalty terms that target sources of the error mentioned above. Detected masks are often fragmented due to occlusion or the presence of debris. Therefore, the partially detected cell fragments lead to a higher false-positive rate. Accordingly, the target ground truth is selected for a detected mask with maximum mask IoU. Then, the mask IoUs of that ground truth with other detected masks are penalized. Similarly, an overlap between a detected mask with all ground truth other than its target is penalized. The second source of error is due to overlaps between the detected masks. Mask-IoU loss penalizes the mask overlap between proposals with different targets. Therefore, the mask IoU between the two proposals with different target ground truth should be minimum.

Bibliography

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., and Others. TensorFlow: A system for large-scale machine learning. In *Proc. Symposium on Operating Systems Design and Implementation* (November 2–4 2016), pp. 265–283.

[2] Abdulla, W. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Available: https://github.com/matterport/Mask-RCNN, 2017. Last accessed: 2019-11-22.

[3] Ahmad, A., Asif, A., Rajpoot, N., Arif, M., Amir, F., and Minhas, A. Correlation filters for detection of cellular nuclei in histopathology images. *Journal of Medical Systems 42*, 1 (2018), 7–15.

[4] Andres, B., Kappes, J. H., Beier, T., Kothe, U., and Hamprecht, F. A. Probabilistic image segmentation with closedness constraints. In *Proc. International Conference on Computer Vision* (Barcelona, Spain, November 6-13 2011), pp. 2611–2618.

[5] Ardizzone, E., Pirrone, R., Gambino, O., and Vitabile, S. Illumination correction on biomedical images. *Computing and Informatics 33*, 1 (2014), 175–196.

[6] Arteta, C., Lempitsky, V., Noble, A., and Zisserman, A. Learning to detect partially overlapping instances. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (Portland, OR, February 2013), pp. 3230–3237.

[7] Arteta, C., Lempitsky, V., Noble, A., and Zisserman, A. Detecting overlapping instances in microscopy images using extremal region trees. *Medical Image Analysis 27* (2016), 3–16.

[8] Arteta, C., Lempitsky, V., Noble, J. A., and Zisserman, A. Learning to detect cells using non-overlapping extremal regions. In *Proc. Medical Image Computing and Computer Assisted Intervention* (Berlin, Heidelberg, October 2012), pp. 348–356.

[9] Brooks, J. COCO Annotator. https://github.com/jsbroks/coco-annotator/, 2019.

[10] Browet, A., Vleeschouwer, C. D., Jacques, L., Mathiah, N., Saykali, B., and Migeotte, I. Cell segmentation with random ferns and graph-cuts. In *Proc. IEEE International Conference on Image Processing* (Phonix, AZ, September 25-28 2016), pp. 4145–4149.

[11] Dai, W., Doyle, J., Liang, X., Zhang, H., Dong, N., Li, Y., and Xing, E. P. SCAN: Structure Correcting Adversarial Network for Organ Segmentation in Chest X-rays. In *Proc. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (Granada, Spain, September 20 2018), pp. 263–273.

[12] Daněk, O., Matula, P., Ortiz-de Solórzano, C., Muñoz-Barrutia, A., Maška, M., and Kozubek, M. Segmentation of touching cell nuclei using a two-stage graph cut model. In *Proc. Scandinavian Conference on Image Analysis* (Oslo, Norway, June 15-18 2009), pp. 410–419.

[13] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proc. Computer Vision and Pattern Recognition* (Miami Beach, FL, June 22-24 2009), pp. 248–255.

[14] DuPont, H. L., Garey, K., Caeiro, J.-P., and Jiang, Z.-D. New advances in Clostridium difficile infection: changing epidemiology, diagnosis, treatment and control. *Current Opinion in Infectious Diseases 21*, 5 (2008), 500–507.

[15] Easley, G. R., Labate, D., and Colonna, F. Shearlet-based total variation diffusion for denoising. *IEEE Transactions on Image Processing 18*, 2 (2009), 260–268.

[16] Endres, B. T., Bassères, E., Memariani, A., Chang, L., Alam, M. J., Vickers, R. J., Kakadiaris, I. A., and Garey, K. W. A novel method for imaging the pharmacological effects of antibiotic treatment on clostridium difficile. *Anaerobe 40* (2016), 10–14.

[17] Feng, L., Song, J. H., Kim, J., Jeong, S., Park, J. S., and Kim, J. Robust nucleus detection with partially labeled exemplars. *IEEE Access 7* (2019), 162169–162178.

[18] Fiaschi, L., Koethe, U., Nair, R., and Hamprecht, F. A. Learning to count with regression forest and structured labels. In *Proc. International Conference on Pattern Recognition* (Tsukuba, Japan, November 11-15 2012), pp. 2685–2688.

[19] Frangi, A. F., Niessen, W. J., Vincken, K. L., and Viergever, M. A. Multiscale vessel enhancement filtering. In *Proc. Medical Image Computing and Computer Assisted Intervention* (Cambridge, MA, October 1998), vol. 1496, pp. 130–137.

[20] Frangi, A. F., Niessen, W. J., Vincken, K. L., and Viergever, M. A. Multiscale vessel enhancement filtering. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention* (Cambridge, MA, October 11-13 1998), pp. 130–137.

[21] Funke, J., Hamprecht, F. A., and Zhang, C. Learning to segment: training hierarchical segmentation under a topological loss. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich, Germany, October 5-9 2015), pp. 268–275.

[22] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proc. Advances in neural information processing systems* (Montréal, Canada, December 8-13 2014), pp. 2672–2680.

[23] Gu, J., Zhu, Y., Yang, B., Yang, J. J., Yang, J. W., Yang, J. Y., and Yang, W. Z. A Multi-scale Pyramid of Fully Convolutional Networks for Automatic Cell Detection. In *International Conference on Bioinformatics and Biomedicine* (Madrid, Spain, December 3-6 2018), pp. 633–636.

[24] He, K., Gkioxari, G., Dollar, P., and Girshick, R. Mask R-CNN. In *Proc. International Conference on Computer Vision* (Venice, Italy, October 22-29 2017), pp. 2961–2969.

[25] Hegde, R. B., Prasad, K., Hebbar, H., and Singh, B. M. K. Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. *Biocybernetics and Biomedical Engineering 39*, 2 (2019), 382–392.

[26] Höfener, H., Homeyer, A., Weiss, N., Molin, J., Lundström, C. F., and Hahn, H. K. Deep learning nuclei detection: A simple approach can deliver state-of-the-art results. *Computerized Medical Imaging and Graphics 70* (2018), 43–52.

[27] Hou, L., Nguyen, V., Kanevsky, A. B., Samaras, D., Kurc, T. M., Zhao, T., Gupta, R. R., Gao, Y., Chen, W., Foran, D., and Others. Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern Recognition 86* (2019), 188–200.

[28] Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., and Lepetit, V. You should use regression to detect cells. In *Proc. Medical Image Computing and Computer-Assisted Intervention* (Munich, Germany, October 5-9 2015), pp. 276–283.

[29] Keuper, M., Schmidt, T., Rodriguez-Franco, M., Schamel, W., Brox, T., Burkhardt, H., and Ronneberger, O. Hierarchical markov random fields for mast cell segmentation in electron microscopic recordings. In *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (Chicago, IL, March 2011), pp. 973–978.

[30] Keuper, M., Schmidt, T., Rodriguez-Franco, M., Schamel, W., Brox, T., Burkhardt, H., and Ronneberger, O. Hierarchical markov random fields for mast cell segmentation in electron microscopic recordings. In *Proc. International Symposium on Biomedical Imaging* (Chicago, IL, March 30 - April 2 2011), pp. 973–978.

[31] King, E., Reisenhofer, R., Kiefer, J., Lim, W., Li, Z., and Heygster, G. Shearlet-based edge detection: flame fronts and tidal flats. In *Proc. SPIE Applications of Digital Image Processing* (San Diego, CA, August 2015), vol. 9599, pp. 1–11.

[32] Kohl, S., Bonekamp, D., Schlemmer, H.-P., Yaqubi, K., Hohenfellner, M., Hadaschik, B., Radtke, J.-P., and Maier-Hein, K. Adversarial Networks for the Detection of Aggressive Prostate Cancer. *arXiv preprint arXiv:1702.08014* (2017).

[33] Korman, S., and Avidan, S. Coherency sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38*, 6 (2015), 1099–1112.

[34] Lee, S., Han, S., Salama, P., Dunn, K. W., and Delp, E. J. Three Dimensional Blind Image Deconvolution for Fluorescence Microscopy using Generative Adversarial Networks. In *Proc. International Symposium on Biomedical Imaging* (Venice, Italy, April 8-11 2019), pp. 538–542.

[35] Lempitsky, V., and Zisserman, A. Learning to count objects in images. In *Proc. Advances in Neural Information Processing Systems* (Vancouver, Canada, December 6-11 2010), pp. 1324–1332.

[36] Li, W., Li, J., Sarma, K. V., Ho, K. C., Shen, S., Knudsen, B. S., Gertych, A., and Arnold, C. W. Path R-CNN for prostate cancer diagnosis and gleason grading of histological images. *IEEE Transactions on Medical Imaging 38*, 4 (2018), 945–954.

[37] Li, X., Li, L., Lu, H., and Liang, Z. Partial volume segmentation of brain magnetic resonance images based on maximum a posteriori probability. *Medical Physics 32*, 7 (2005), 2337–2345.

[38] Li, Z., Wang, Y., and Yu, J. Brain tumor segmentation using an adversarial network. In *Proc. International MICCAI Brainlesion Workshop* (Quebec City, QC, Canada, September 10-14 2017), pp. 123–132.

[39] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proc. Computer Vision and Pattern Recognition* (Honolulu, HI, July 21-26 2017).

[40] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision* (Zurich, Switzerland, September 6-12 2014), pp. 740–755.

[41] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proc. Computer Vision and Pattern Recognition* (Boston, MA, June 8-10 2015), pp. 3431–3440.

[42] Luc, P., Couprie, C., Chintala, S., and Verbeek, J. Semantic segmentation using adversarial networks. In *Proc. NIPS Workshop on Adversarial Training* (Barcelona, Spain, December 5-10 2016), pp. 1–9.

[43] Matas, J., Chum, O., Urban, M., and Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing 22*, 10 (2004), 761–767.

[44] Memariani, A., and Kakadiaris, I. A. SoLiD: segmentation of clostridioides difficile cells in the presence of inhomogeneous illumination using a deep adversarial network. In *International Workshop on Machine Learning in Medical Imaging* (Granada, Spain, September 16 2018), pp. 285–293.

[45] Memariani, A., Nikou, C., Endres, B. T., Bassères, E., Garey, K. W., and Kakadiaris, I. A. DeTEC: detection of touching elongated cells in SEM images. In *Proc. International Symposium on Visual Computing* (Las Vegas, NV, December 12-14 2016), pp. 288–297.

[46] Memariani, A., Nikou, C., Endres, B. T., Bassères, E., Garey, K. W., and Kakadiaris, I. A. DeTEC: Detection of touching elongated cells in SEM images. In *Proc. Advances in Visual Computing* (December 12-14 2016), pp. 288–297.

[47] Memariani, A., Nikou, C., Endres, B. T., Bassères, E., Garey, K. W., and Kakadiaris, I. A. DETCIC: Detection of elongated touching cells with inhomogeneous illumination using a stack of conditional random fields. In *Proc. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (Funchal Madeira Portugal, January 27-29 2018), pp. 574–580.

[48] Minaee, S., Fotouhi, M., and Khalaj, B. H. A geometric approach to fully automatic chromosome segmentation. In *Proc. IEEE Signal Processing in Medicine and Biology Symposium* (Philadelphia, PA, December 2014), pp. 1–6.

[49] Moeskops, P., Veta, M., Lafarge, M. W., Eppenhof, K. A. J., and Pluim, J. P. W. Adversarial training and dilated convolutions for brain MRI segmentation. In *Proc. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support.* Quebec City, QC, Canada, September 14 2017, pp. 56–64.

[50] Mori, G. Guiding model search using segmentation. In *Proc. International Conference on Computer Vision* (Beijing, China, October 17-20 2005), pp. 1417–1423.

[51] Naylor, P., Laé, M., Reyal, F., and Walter, T. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE Transactions on Medical Imaging 38*, 2 (2018), 448–459.

[52] Ng, A. Machine learning yearning: technical strategy for AI engineers in the era of deep learning. Available: https://www.mlyearning.org, 2019. Last accessed: 2019-9-15.

[53] Ojala, T., Pietikainen, M., and Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 7 (July 2002), 971–987.

[54] Proc. Efros, A. A., and Freeman, W. T. Image quilting for texture synthesis and transfer. In *Computer Graphics and Interactive Techniques* (New York, NY, August 2001), pp. 341–346.

[55] Ramesh, N., and Tasdizen, T. Cell segmentation using a similarity interface with a multi-task convolutional neural network. *IEEE Journal of Biomedical and Health Informatics 23*, 4 (2019), 1457–1468.

[56] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: unified, real-time object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, June 26 – July 1 2016).

[57] Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Information Processing Systems*. Montreal, Canada, October 22-29 2015, pp. 91–99.

[58] Ronneberger, O., Fischer, P., and Brox, T. U-net: convolutional networks for biomedical image segmentation. In *Proc. Medical Image Computing and Computer-Assisted Intervention* (Munich, Germany, October 5-9 2015), pp. 234–241.

[59] Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (Long Beach, CA, December 2017), pp. 3856–3866.

[60] Sabour, S., Frosst, N., and Hinton, G. E. Matrix capsules with EM routing. In *International Conference on Learning Representations* (Vancouver Canada, February 2018).

[61] Saiyod, S., and Wayalun, P. A hybrid technique for overlapped chromosome segmentation of g-band matasprea images automatic. In *Proc. International Conference on Digital Information and Communication Technology and its Applications* (Bangkok, Thailand, May 2014), pp. 400–404.

[62] Santamaria-Pang, A., Rittscher, J., Gerdes, M., and Padfield, D. Cell segmentation and classification by hierarchical supervised shape ranking. In *Proc. IEEE International Symposium on Biomedical Imaging* (Brooklyn, NY, April 16-19 2015), pp. 1296–1299.

[63] Santamaria-Pang, A., Rittscher, J., Gerdes, M., and Padfield, D. Cell segmentation and classification by hierarchical supervised shape ranking. In *Proc. International Symposium on Biomedical Imaging* (Brooklyn, NY, April 16-19 2015), pp. 1296–1299.

[64] Santha Kumari, T., Suresh, B., Yashwanth, P., and Rao, R. R. Modified histogram based contrast enhancement using unsharp masking filter for medical images. *International Journal of Research in Computer and Communication Technology 4*, 2 (2015), 137–140.

[65] Shen, D., Wu, G., and Suk, H.-I. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering 19* (2017), 221–248.

[66] Shi, J., Zheng, X., Wu, J., Gong, B., Zhang, Q., and Ying, S. Quaternion Grassmann average network for learning representation of histopathological image. *Pattern Recognition 89* (2019), 67–76.

[67] Sreenivasan, K. R., Havlicek, M., and Deshpande, G. Nonparametric hemodynamic deconvolution of fmri using homomorphic filtering. *IEEE Transactions on Medical Imaging 34*, 5 (May 2015), 1155–1163.

[68] Tiwari, P., Qian, J., Li, Q., Wang, B., Gupta, D., Khanna, A., Rodrigues, J. J. P. C., and de Albuquerque, V. H. C. Detection of subtype blood cells using deep learning. *Cognitive Systems Research 52* (2018), 1036–1044.

[69] Tofighi, M., Guo, T., Vanamala, J. K. P., and Monga, V. Deep networks with shape priors for nucleus detection. In *Proc. International Conference on Image Processing* (Athens, Greece, October 7-10 2018), pp. 719–723.

[70] Türetken, E., Wang, X., Becker, C. J., Haubold, C., and Fua, P. Network flow integer programming to track elliptical cells in time-lapse sequences. *IEEE Transactions on Medical Imaging 36*, 4 (2017), 942–951.

[71] Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., and Gee, J. C. N4itk: improved n3 bias correction. *IEEE Transactions on Medical Imaging 29*, 6 (2010), 1310–1320.

[72] Ulman, V., Maška, M., Magnusson, K. E. G., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., Smal, I., Rohr, K., Jaldén, J., Blau, H. M., Dzyubachyk, O., Lelieveldt, B., Xiao, P., Li, Y., Cho, S.-Y., Dufour, A. C., Olivo-Marin, J.-C., Reyes-Aldasoro, C. C., Solis-Lemus, J. A., Bensch, R., Brox, T., Stegmaier, J., Mikut, R., Wolf, S., Hamprecht, F. A., Esteves, T., Quelhas, P., Demirel, Ö., Malmström, L., Jug, F., Tomancak, P., Meijering, E., Muñoz-Barrutia, A., Kozubek, M., and Ortiz-de Solorzano, C. An objective comparison of cell-tracking algorithms. *Nature Methods 14*, 12 (2017), 1141.

[73] Van Valen, D. A., Kudo, T., Lane, K. M., Macklin, D. N., Quach, N. T., DeFelice, M. M., Maayan, I., Tanouchi, Y., Ashley, E. A., and Covert, M. W. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Computational Biology 12*, 11 (2016), e1005177.

[74] Wan, F., Smedby, O., and Wang, C. Simultaneous MR knee image segmentation and bias field correction using deep learning and partial convolution. In *Proc. Medical Imaging: Image Processing*, pp. 61–67.

[75] Wayalun, P., Chomphuwiset, P., Laopracha, N., and Wanchanthuek, P. Images enhancement of g-band chromosome using histogram equalization, {OTSU} thresholding, morphological dilation and flood fill techniques. In *Proc. 8$ˆ{th}$ International Conference on Computing and Networking Technology* (Gueongju, China, February 2012), pp. 163–168.

[76] Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In *In Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 5028–5037.

[77] Wu, B., and Nevatia, R. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *International Journal of Computer Vision 82* (2009), 185–204.

[78] Xie, W., Noble, J. A., and Zisserman, A. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 6*, 3 (2018), 283–292.

[79] Xie, Y., Xing, F., Shi, X., Kong, X., Su, H., and Yang, L. Efficient and robust cell detection: A structured regression approach. *Medical Image Analysis 44* (2018), 245–254.

[80] Xue, Y., Ray, N., Hugh, J., and Bigras, G. Cell counting by regression using convolutional neural network. In *Proc. European Conference on Computer Vision* (Amsterdam, The Netherlands, October 8-16 2016), pp. 274–290.

[81] Xue, Y., Xu, T., Zhang, H., Long, L. R., and Huang, X. SegAN: Adversarial network with multi-scale L-1 loss for medical image segmentation. *Neuroinformatics*, 3-4, 383–392.

[82] Yang, D., Gach, H., Li, H., and Mutic, S. Tu-h-206-04: An effective homomorphic unsharp mask filtering method to correct intensity inhomogeneity in daily treatment mr images. *Medical Physics 43*, 6 (2016), 3774–3774.

[83] Yarkony, J., Ihler, A., and Fowlkes, C. C. Fast planar correlation clustering for image segmentation. In *Proc. European Conference on Computer Vision* (Florence, Italy, October 7-13 2012), pp. 568–581.

[84] Zelinsky, D. *A first course in linear algebra*. Academic Press, 2014.

[85] Zhang, C., Yarkony, J., and Hamprecht, F. A. Cell detection and segmentation using correlation clustering. In *Proc. Medical Image Computing and Computer-Assisted Intervention* (Boston, September 2014), Springer, pp. 9–16.

[86] Zheng, W., Chee, M. W., and Zagorodnov, V. Improvement of brain segmentation accuracy by optimizing non-uniformity correction using n3. *Neuroimage 48*, 1 (2009), 73–83.

[87] Zheng, Y., Grossman, M., Awate, S. P., and Gee, J. C. Automatic correction of intensity nonuniformity from sparseness of gradient distribution in medical images. In *Proc.*

International Conference on Medical Image Computing and Computer-Assisted Intervention (London, UK, September 20-24 2009), pp. 852–859.

[88] Zhi, X.-H., and Shen, H.-B. Saliency driven region-edge-based top down level set evolution reveals the asynchronous focus in image segmentation. *Pattern Recognition 80* (2018), 241–255.

[89] Zhou, S., Wang, J., Zhang, M., Cai, Q., and Gong, Y. Correntropy-based level set method for medical image segmentation and bias correction. *Neurocomputing 234* (2017), 216–229.

[90] Zosso, D., Tran, G., and Osher, S. A unifying retinex model based on non-local differential operators. In *Proc. SPIE Computational Imaging* (Burlingame, CA, February 5-7 2013), vol. 8657, pp. 865702–865716.

[91] Zosso, D., Tran, G., and Osher, S. J. Non-local retinex—a unifying framework and beyond. *SIAM Journal on Imaging Sciences 8*, 2 (2015), 787–826.