Deep Learning and Physics-inspired Modeling for Cell Segmentation and Tracking with Application to Bacterial Biofilms

---

A

Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Tanjin Taher Toma

May 2024

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Tanjin Taher Toma

This Dissertation has been read and approved by the examing committee:

Advisor: Scott T. Acton

Advisor:

Committee Member: Zongli Lin

Committee Member: Miaomiao Zhang

Committee Member: Andreas Gahlmann

Committee Member: Yangfeng Ji

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

May 2024

Deep Learning and Physics-inspired Modeling for Cell Segmentation and Tracking with Application to Bacterial Biofilms

Abstract

Synthetic cell image generation, as well as cell segmentation and tracking from microscopy, are essential in biology and biomedical research for advancing scientific understanding of a cell population. Synthetic image generation allows the testing of algorithms and the training of data-driven methods. The segmentation task allows for identifying, isolating, and analyzing individual cells from images, while tracking enables the analysis of cell behavior over multiple frames of an image sequence. Developing cell segmentation and tracking algorithms is an active research domain that propels drug discovery, disease diagnosis and treatment, tissue engineering, and basic research in cell biology.

In this dissertation, we introduce novel synthetic image generation, cell segmentation, and tracking algorithms to study a particularly challenging cell population called bacterial biofilms from lattice light-sheet microscopy 3D images and videos. Biofilms are complex biological systems that have critical functions in diverse fields, including the production of bioelectricity and the development of infectious diseases. High cell density and intra-cellular intensity inhomogeneity in the microscopy images of biofilms pose significant challenges to the existing algorithms in identifying individual bacterial cells and tracking their movements over time. The dissertation achieves three main objectives. (1) We present a simulation framework designed to produce synthetic biofilm images and videos featuring cells with realistic curvilinear morphology. This framework is demonstrated to be useful for training supervised deep learning models for cell segmentation and tracking purposes. (2) We propose a novel deep learning-based cell segmentation approach that involves enhancing the cell interior and border information leveraging Euclidean distance transforms and then detecting cell seeds for a classical watershed segmentation through voxel-wise classification. (3) We introduce an innovative cell tracking framework that incorporates a deep temporal sequence classification network to predict the probability of potential associations between consecutive frames, followed by a one-to-one matching optimization to establish accurate matches. The tracking approach also considers the detection of cell division events via an Eigen decomposition-based strategy.

To my beloved family.

Acknowledgements

I extend my sincere appreciation to the Department of Electrical and Computer Engineering at the University of Virginia for granting me the invaluable opportunity to embark on my doctoral studies. I express profound gratitude to my advisor, Scott T. Acton, for his unwavering guidance, encouragement, and strong support throughout my Ph.D. journey.

Furthermore, I would like to acknowledge and thank the members of my dissertation committee, Zongli Lin, Miaomiao Zhang, Andreas Gahlmann, and Yangfeng Ji, for their valuable feedback and insightful suggestions that significantly contributed to refining my research and enhancing the quality of my thesis.

A special thanks is owed to my present and former lab mates at VIVA Lab for creating a supportive environment that inspired me to pursue my research goals with passion and dedication. I also express my gratitude to my collaborating lab, Gahlmann Lab, for sharing resources and providing essential suggestions for completing this dissertation.

Lastly, I extend my heartfelt thanks to my family and friends. My deepest gratitude goes to my husband and my parents for their enduring love, support and encouragement throughout my academic journey.

This thesis work is supported by the U.S. National Institute of General Medical Sciences under NIH Grant No. 1R01GM139002. I am grateful to NIH for providing financial support that made this research possible.

Table of Contents

List of Tables

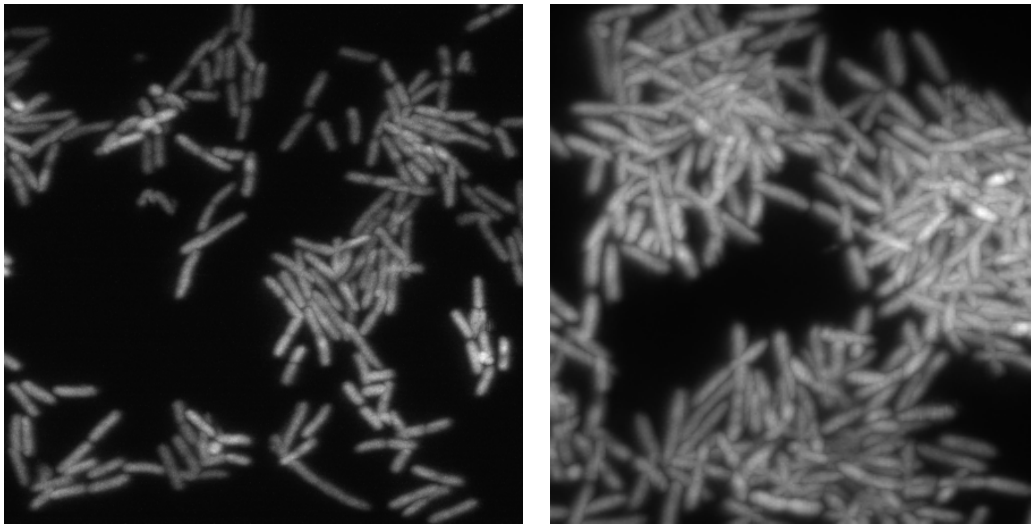
List of Figures

Chapter 1

Introduction

Automatic cell segmentation and tracking is an active field of research in the area of biomedical image processing. Segmentation of individual cell instances from raw microscopy images enables researchers to localize cells and extract many representative cellular attributes [1]. Additionally, tracking cells over time in a microscopy image sequence provides information regarding cell motion, growth and division rates, cell appearance, and death rates within the imaging field of view [2]. Developing cell segmentation and tracking algorithms is essential for gaining insights into the underlying dynamics of a given cell population.

In this thesis, we focus on developing automatic cell segmentation and tracking approaches to study 3D microscopy images of a particularly challenging cell population known as bacterial biofilms. Biofilms are large multicellular communities comprising spatially dense allocation of bacterial cells. Bacterial biofilms play an immense role in regulating many ecological processes, such as recycling soil nutrients, and assisting plant growth [3, 4]. Biofilms are also useful in producing fuel cells, which can help meet the energy requirements [5, 6]. On the contrary, undesired growth of biofilms may cause infectious diseases or degrade process efficiencies in an industrial setting [7, 8]. Thus, analysis of bacterial biofilms is necessary to exploit their potential for human benefit as well as to control their undesired growth. Fig. 1.1 presents examples of lattice light-sheet microscopy images of two types of bacterial biofilms. High cell density in biofilm images poses a significant challenge for existing approaches in accurately identifying individual cell instances and tracking their movements over time. Hence, developing effective segmentation and tracking methods is essential for automatic and accurate analyses of bacterial cell populations.



(a) *E. coli* (b) *S. oneidensis*

Figure 1.1: Maximum intensity projection of the image stacks of two kinds of biofilms

1.1 Cell Segmentation

Single cell segmentation from microscopy images of biofilms is highly susceptible to undersegmentation and oversegmentation issues, primarily due to the presence of numerous touching cells and instances with intensity inhomogeneities. Traditional model-based methods, including thresholding, level sets, graph cuts, and morphological watershed techniques [9, 10, 11, 12], are not optimal for effective biofilm segmentation. Some of these approaches demand an initial contour or coarse segmentation of individual instances, others require tuning numerous hyperparameters, and some are iterative in nature, making them computationally expensive when applied to segmenting a large number of cells in densely packed biofilms.

In recent years, deep learning-based approaches have shown superior performance in many cell segmentation tasks. A commonly adopted strategy involves initially conducting pixel-wise segmentation through a deep network, such as U-Net or its variants [13, 14, 15, 16], and subsequently aggregate pixels into isolated cell instances using classical methods, such as watershed or graph partitioning techniques [17, 18]. However, performing pixel-wise segmentation directly on low-contrast input images is observed to be less effective in detecting subtle boundary changes between touching cells. Additionally, deep learning-based segmentation methods that involve the estimation of spatial gradient maps [19] can be easily affected by intra-cellular intensity inhomogeneity, leading to the segmentation of broken cells. Furthermore, despite the effectiveness of region-based convolutional neural networks, namely Mask-RCNN and its variants [20, 21], in many segmentation applications,

these methods are found to struggle in the presence of touching instances due to non-maximum suppression operations.

Recently, distance transform prediction-based approaches have become very popular owing to their superior performance in segmenting individual cell regions [22, 23, 24]. Such methods enhance individual cell regions performing U-Net-based distance transform estimation, followed by a classical seeded watershed segmentation. However, the lack of effective distance transform representation, difficulty in learning high-quality distance maps from low contrast input, and the need for multiple threshold tuning during seed selection in the watershed segmentation stage often hinder successful segmentation. Inspired by the promising performance of these approaches in certain cell segmentation tasks, we propose a superior distance transform-based segmentation approach to effectively segment bacterial cells from dense biofilm images. The proposed segmentation approach addresses both oversegmentation and undersegmentation errors by effectively extracting cell interior and border information through Euclidean distance transforms, and then estimating cell seeds for classical seeded watershed through voxel-wise classification. The efficacy of the proposed approach has been evaluated using synthetic and real 3D lattice light-sheet microscopy images of biofilms.

1.2 Cell Tracking

As individual cell instances are segmented in a biofilm image stack, we can address the problem of tracking these instances and their offspring from a temporal sequence of image stacks. In the presence of high cell density in such a biofilm video, frame-by-frame association of the instances and subsequent cell division detection becomes a challenging task. Various classical model-based approaches have been applied to address cell tracking problems, including nearest neighbor-based tracking, graph matching techniques, motion filter-based methods, and multiple hypothesis-based probabilistic approaches [25, 26, 27, 28, 29]. However, these approaches often require manual tuning of many parameters and rely on simplistic assumptions about cell behavior, such as the choice of a particular cell motion model, which may not always hold true. Some also necessitate a user-defined similarity function to associate similar instances between consecutive frames.

In recent years, various deep learning-based cell tracking approaches have been introduced. Such methods include a graph neural network method for complete cell trajectory estimation [30], a deep reinforcement learning method [31], and a Siamese networks pipeline for frame-by-frame associations [32]. However, these methods lack the incorporation of temporal history for predicting associations in the next frame, potentially leading to inaccuracies when cells are poorly imaged

or segmented. Additionally, they do not explicitly enforce one-to-one matching between successive frames, allowing for erroneous associations between one-to-multiple instances. In our proposed approach, we address these limitations and develop an effective tracking strategy by incorporating deep learning with model-based techniques for tracking bacterial cells over time and detecting their offspring in crowded image scenarios.

1.3 Thesis Overview and Outline

This thesis involves the development of data-driven solutions that integrate deep learning and image processing to facilitate the segmentation and tracking of bacterial cells in microscopy images of densely packed biofilms. The thesis accomplishes three main research objectives: (1) In chapter 2, we propose a simulation framework to generate 3D synthetic biofilm images consisting of realistic-shaped bacterial cells [33]. We exploit an elastic shape analysis framework to model realistic bacteria shapes in 3D. The proposed simulation framework can be useful for producing realistic-shaped synthetic images and videos to train deep learning models for cell segmentation and tracking purposes. (2) In chapter 3, we propose a new 3D cell segmentation method called DeepSeeded, a cascaded deep learning architecture to segment densely packed bacterial cells in biofilm microscopy images [34]. We incorporate effective distance transform representation, design a loss function for learning high-quality distance maps, and adopt a data-driven seed estimation to avoid hyperparameter tuning. (3) In chapter 4, we introduce a cell tracking framework named DenseTrack [35] for tracking bacterial cells in 3D image sequences of densely populated biofilms. The proposed tracking approach formulates the frame-by-frame association problem as a deep learning-based temporal sequence classification task, followed by an optimization-based one-to-one matching. Additionally, we propose an effective cell division detection method involving the Eigen decomposition of the coordinates of the unmatched instances in the next frame.

Chapter 2

Realistic-shape Bacterial Biofilm Simulator for Deep Learning-based 3D Single-Cell Segmentation

2.1 Introduction

This chapter presents an image simulation tool that can produce 3D synthetic biofilm images consisting of realistic-shaped bacteria cells [33]. We model realistic bacteria shapes exploiting a 3D elastic shape analysis framework known as square-root normal field (SRNF) representation [36]. These newly modeled curvilinear-shaped bacteria cells replace the conventional rod-shaped cells in synthetic biofilms generated through a biofilm modeling software called CellModeller [37]. We then demonstrate that training a deep segmentation network using the resulting synthetic biofilms noticeably improves the single-cell segmentation performance on real biofilm data when compared to training with conventional rod-shaped synthetic biofilms.

2.2 Related Works

Single-cell segmentation in dense 3D biofilms is a challenging task. Traditional model-based image segmentation approaches, such as thresholding, watershed, and level-set methods, often fail to accurately segment intensity-inhomogeneous 3D bacteria cells from the noisy background [38, 39, 40]. The high density of cells in a biofilm further degrades segmentation performance where most al-

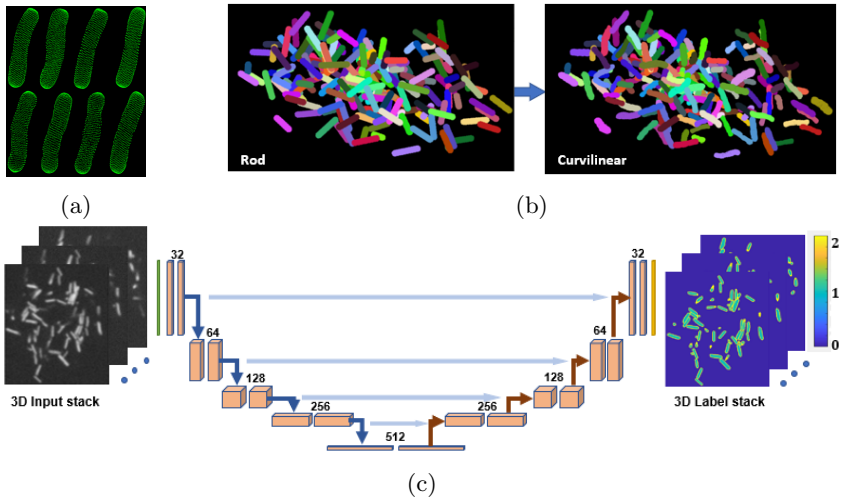


Figure 2.1: Illustration of the proposed method: (a) Examples of curvilinear-shaped synthetic bacteria, (b) Transformation of a rod-shaped simulated biofilm into a curvilinear-shaped simulated biofilm, (c) Training 3D U-Net with curvilinear-shaped biofilm dataset.

gorithms struggle to separate the neighbouring touching cells [41]. In contrast, data-driven deep learning techniques have recently shown superior potential in cell segmentation and detection problems [42, 43, 19] by overcoming various limitations of the conventional model-based approaches. However, deep networks typically require a large amount of ground-truth training data for the optimal performance. We note that it is difficult, if not impossible, to manually produce ground-truth annotations of individual cells from a large number of dense 3D biofilms. Thus, synthetic training data is employed in recent studies to train deep segmentation networks for 3D cell images [44, 17]. There exist computational algorithms to simulate synthetic bacterial biofilms, which model the growth and division of bacterial cells in a biofilm according to some predefined biological and chemical rules [37, 45, 46]. Such simulation models represent bacterial cells in a simple shape geometry, such as a rod-shaped cell, which consists of a cylindrical body and two hemispherical ends. The rod-shaped cellular morphology is mostly used to model bacterial biofilms of the *Bacillus* species, such as *Escherichia coli*, *Bacillus cereus*, and *Salmonella bongori* microbial populations. However, the synthetic biofilms represented with rod-shaped cells do not correspond very well with real biofilms from the microscope, where the bacteria cells have more irregular curvilinear-shaped morphology.

2.3 Proposed Approach

In this section, we explain how to model realistic bacteria shapes using shape analysis, exploit these shapes to simulate 3D synthetic biofilms, and train a basic U-Net with the simulated biofilms. An overview of the proposed method is demonstrated in Fig 2.1.

2.3.1 Modeling Realistic Bacteria Shapes

We use the square-root normal field (SRNF) representation [36] to analyze a small number of bacteria cells manually segmented from real biofilms and generate new synthetic bacteria shapes. We manually segment 50 bacterial cells from several real microscopy biofilms which are not used to validate the segmentation performance. The segmented cells are represented in point clouds in $\mathbb{R}^3$. The point clouds are transformed into spherically-parameterized surfaces by applying a 3D surface parameterization representation method called SPHARM [47]. Each bacterial shape is represented as a parameterized function: $f_i : \mathbb{S}^2 \to \mathbb{R}^3 (i = 1, ..., 50), f_i \in \mathcal{F}$, where $\mathcal{F}$ is the set of surfaces.

To analyze the shapes of bacterial surfaces, we utilize the elastic shape analysis approach [48]. Let Γ be the set of all orientation-preserving diffeomorphisms of $\mathbb{S}^2$. For any parameterized surface $f \in \mathcal{F}$ and a $\gamma \in \Gamma$, the composition $f \circ \gamma$ is simply a re-parameterization of f and has the same shape as f. Equivalently, for any two maps f_1 and f_2, elements of γ help in a dense registration of points across two maps. Initially, for any $s \in \mathbb{S}^2$, we say that the point $f_1(s)$ on f_1 is registered to the point $f_2(s)$ on f_2. If we re-parameterize f_2 by γ, then the point $f_1(s)$ is now registered to the point $f_2(\gamma(s))$. Thus, γ becomes a tool for controlling the registration between surfaces. In order to quantify the differences in the *shapes* of surfaces, we need a metric that is invariant to their rigid motions, global scaling, and re-parameterization.

To accomplish this, we utilize a special mathematical representation called the square-root normal field (SRNF) as follows. For an $s \in \mathbb{S}^2$, where $s = (u, v)$ is a point on the sphere $\mathbb{S}^2$, the partial derivatives f_u and f_v denote two orthogonal tangent vectors to the surface f at the point $f(s)$. The vector $\mathbf{n}(s) = f_u \times f_v$ denotes the (unnormalized) normal vector at points s. Then, SRNF $q : \mathbb{S}^2 \to \mathbb{R}^3$ of f is defined as: for $s \in \mathbb{S}^2$, $q(s) = n(s)/\left|n(s)\right|^{\frac{1}{2}}$, where $|\cdot|$ denotes the vector norm. As described in [49], the $\mathbb{L}^2$-metric under SRNF representation is invariant to shape-preserving transformations, and can be used to compare shapes of surfaces. The SRNF is invariant to translations since it involves only derivatives of f. We separate the scaling variability by re-scaling all surfaces to unit area: $f = f/\sqrt{\alpha_f}$, where $\alpha_f = \int_{\mathbb{S}^2} |n(s)| ds$ is the surface area of f. In this modeling problem, we

will separately preserve, analyze, and include the size information of bacteria as it is an important feature.

Now, let $O \in SO(3)$ denotes all 3D rotations. The SRNF of the rotated and re-parameterized surface $O(f \circ \gamma)$ becomes $O(q \star \gamma) \equiv O\sqrt{\mathbf{J}_\gamma}(q \circ \gamma)$, where $\mathbf{J}_\gamma$ is the determinant of the Jacobian of γ. Hence, we can define a *shape metric* between f_1 and f_2 as follows,

$$d_s(f_1, f_2) = \inf_{(O,\gamma) \in SO(3) \times \Gamma} \|q_1 - O(q_2 \star \gamma)\| \ , \tag{2.1}$$

for all $O \in SO(3)$ and all $\gamma \in \Gamma$. Here, the optimal rotation O^* is solved by Procrustes analysis, and the optimal re-parameterization γ^* is solved using a gradient-descent approach. The details of the algorithms are presented in [48].

We can use this framework to define statistical summaries of bacteria shapes. We define the Karcher mean of all given surfaces $f_1, f_2, \ldots, f_{50}$ as the shape μ that minimizes the sum of square of distances to the given shapes under the shape metric, i.e., $\mu = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} d_s(f, f_i)^2$. Note that in the process of computing the Karcher mean, we also get all surfaces f_i that are aligned to the mean shape. With the Karcher mean and the aligned surfaces, we perform principal component analysis (PCA) to capture essential shape variability, and derive low-dimensional representations of shapes for use in generating synthetic bacteria. We start by computing the covariance matrix C for surfaces, $C = \sum_{i=1}^{n} V_i V_i^T$, where matrix V contains $f_i - \mu$ in its ith column V_i. By performing singular value decomposition (SVD) of the covariance matrix $C = U\Sigma U^T$, we obtain the left singular vectors as the columns of the unitary matrix U. These columns form the principal directions of shape variability in the data. Next, we compute the principal scores for all surfaces based on $z_{i,d} = \langle f_i - \mu, U(:,d) \rangle$, where $z_{i,d}$ denotes the surface f_i's principal score on the d-th principal direction. Thus, a high-dimensional object f_i is now represented by a d-dimensional vector $z_i \in \mathbb{R}^d$. In this study, the first 19 principal components can explain over 95% of the variability in shapes, so we use $d = 19$. It is important to note that this representation is invertible and we can reconstruct the surfaces according to: $\widehat{f_i} = \mu + \sum_{d=1}^{n} z_{i,d} U(:,d)$. After computing the areas and shape principal scores of manually segmented bacterial surfaces, we generate some synthetic shapes. We compute the mean μ_α and the variance σ_α^2 of areas α_{f_i} of all given surfaces. We also compute the variance σ_d^2 of shape principal scores $z_{i,d}$, where $i = 1, 2, ..., 50$ and $d = 1, 2, ..., 19$. For synthesis, we randomly generate areas $\widehat{\alpha}_{f_j}$ and principal scores $\widehat{z}_{j,d}$ according to the normal distributions $\mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$ and $\mathcal{N}(\mu_d, \sigma_d^2)$, where $j = 1, 2, \ldots, 500$. The reconstructed curvilinear bacteria surface is

represented as follows,

$$\widehat{f_j} = \widehat{\alpha}_{f_j}(\mu + \sum_{d=1}^{n} \widehat{z}_{j,d} U(:,d)), \quad \forall j \tag{2.2}$$

Overall, we have generated 500 synthetic bacteria surfaces. Finally, we transform these synthetic surfaces back into point clouds in $\mathbb{R}^3$. A few examples of such generated shapes are shown in Fig 2.1a.

2.3.2 Simulating 3D Biofilms

With the generated curvilinear bacteria shapes, next step is to simulate 3D synthetic bacterial biofilms. First, we model the biofilm growth for different spatial arrangement of cells using the biofilm modeling software, named Cell-Modeller [37]. An output from Cell-Modeller can be considered as a biofilm consisting of numerous adjacent bacteria cell units, where each unit is represented by a simple rod-shaped structure.

Let a rod-shaped biofilm with m rod-shaped cells be represented as a collection of point clouds, $B_r = \{P_r^{(1)}, ..., P_r^{(m)}\}$, where each rod-shaped cell is represented as a point cloud, $P_r^{(j)} = \{r_i\}_{i=1}^{n}$ of n points in $\mathbb{R}^3$. We aim to replace $P_r^{(j)}$ for all j with a synthetic curvilinear-shaped point cloud, $P_c^{(j)} = \{c_i\}_{i=1}^{n}$, $c_i \in \mathbb{R}^3$. First, we shift both $P_r^{(j)}$ and $P_c^{(j)}$ to the center of the coordinate system by subtracting the means of the corresponding point clouds. Next, $P_c^{(j)}$ is rotated to align with the rod-shaped point cloud $P_r^{(j)}$ using the iterative closest point (ICP) algorithm [50]. To make the size of $P_c^{(j)}$ consistent with $P_r^{(j)}$, we scale the cell $P_c^{(j)}$ by applying: $\widehat{P}_c^{(j)} = \|P_r^{(j)}\|_2 (\frac{P_c^{(j)}}{\|P_c^{(j)}\|_2})$. Finally, $\widehat{P}_c^{(j)}$ is translated to the original coordinates of $P_r^{(j)}$. We repeat these steps for all j to replace all rod-shaped cells in B_r with the curvilinear-shaped cells from our set of generated shapes, and obtain a curvilinear-shaped synthetic biofilm, $B_c = \{\widehat{P}_c^{(1)}, ..., \widehat{P}_c^{(m)}\}$. The transformation of a rod-shaped synthetic biofilm into a curvilinear-shaped one is depicted in Fig 2.1b.

The continuous coordinates of a synthetic biofilm B_c are quantized to turn it into a discrete 3D volume. To generate a training label for the segmentation network, the voxels within the discrete volume are labelled into three categories, background (class 0), cell-interior (class 1), and cell-boundary (class 2). Such multi-class labelling can provide better segmentation performance compared to a binary labeling [42, 17]. To generate the fluorescence image volume as a training input data, we first simulate fluorophore distributions within the discrete volume, and then convolve it with the experimentally-acquired point spread function (PSF) [17]. Finally, Gaussian and Poisson-distributed noise is added to the convolved image volume.

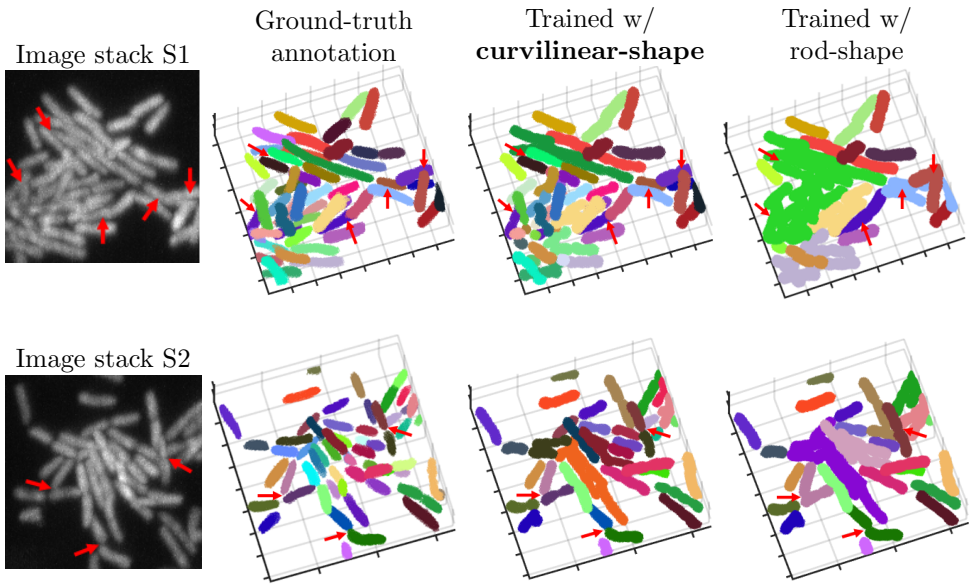


Figure 2.2: Qualitative evaluation of single-cell segmentation on real 3D microscopy stacks illustrated in point clouds. Red arrows indicate comparative performance of the models in terms of touching cell separation.

2.3.3 Training Segmentation Network

We train a 3D U-Net convolutional neural network [51] with the simulated curvilinear-shaped biofilms. The network architecture is shown in Fig 2.1c. The network has an encoding path followed by a decoding path with each path consisting of five layers. In the encoding path, each layer conducts two $3 \times 3 \times 3$ convolutions each followed by an instance normalization and parametric rectified linear unit (PReLU) activation. A $2 \times 2 \times 2$ max-pooling operation is performed between the consecutive layers. The decoding path performs transposed operations of the encoding path. Further, skip connections are added between the corresponding layers of encoding and decoding paths to combine features from different spatial resolutions. In Fig 2.1c, orange boxes represent the feature maps. The number of channels in each of the two features maps of a layer is also denoted in the figure. To train the network, we exploit a hybrid loss function combining Dice loss and focal loss [52]. While the Dice loss makes the network learn the voxel distribution between different classes, focal loss term mostly forces the network better learn the poorly predicted boundary voxels. The Adam optimization algorithm is used to minimize the loss function.

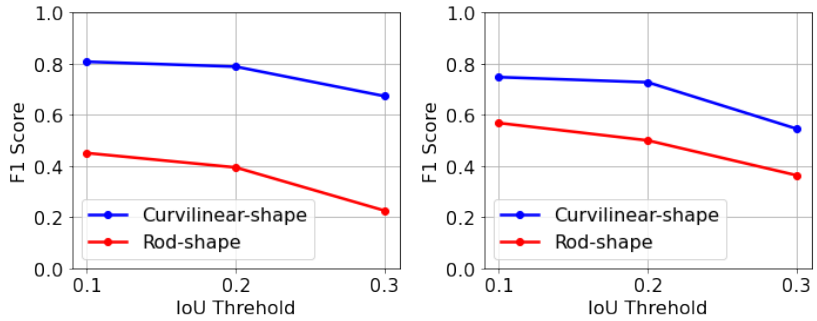
Test Stacks	Train Dataset	Precision	Recall	F1 Score
Stack S1	Rod-shape	0.9412	0.2963	0.4507
	Curvilinear-shape	0.8400	0.7778	**0.8077**
Stack S2	Rod-shape	0.9259	0.4098	0.5682
	Curvilinear-shape	0.9737	0.6066	**0.7475**

Table 2.1: Quantitative evaluation on real test stacks

2.4 Experimental Setup

We perform experiments to analyze how the U-Net segmentation network performs segmentation task on real microscopy data while training the network with curvilinear-shaped synthetic dataset vs. rod-shaped synthetic dataset. We simulated 25 rod-shaped 3D biofilms with different cell arrangements and density from the CellModeller software [37]. For each rod-shaped biofilm of dimensions $500 \times 500 \times 100$ (x-y-z), the corresponding curvilinear-shaped counterpart is generated using the proposed framework. Each biofilm is further randomly cropped into four ROI volumes of spatial dimensions $160 \times 160 \times 96$ (x-y-z). Overall, there are $25 \times 4 = 100$ training input-label pairs in both datasets. The network is trained on the MONAI (Medical Open Network for AI) platform [53], which is an open-source PyTorch-based framework for image segmentation. For training with either dataset, we have used the same training parameters with train batch size=2, learning rate=0.0001 in Adam optimizer, and 250 training epochs. The network is trained on a machine with NVIDIA RTX GPU with 24 GB memory.

We test the trained models on two real lattice light-sheet microscopy *Escherichia coli* image volumes obtained from [54]. The dimensions of the two volumes are $153 \times 154 \times 51$ and $164 \times 166 \times 51$, respectively. For these two volumes, ground-truth cell annotations are also provided. To evaluate segmentations of these two test image volumes, the raw volumes are first pre-processed with deconvolution operation and then fed into the network. The network outputs a three-class segmentation mask (background, cell interior, and cell boundary). The voxels corresponding to the cell interior class are grouped using connected-component analysis and then dilated to obtain the final instance-labeled segmentation result (S). We qualitatively and quantitatively evaluate the segmentation outputs (S) with respect to the ground-truth annotations (G). For quantitative evaluation of the single-cell segmentation, we compute precision, recall, and F1 score for different intersection-over-union (IoU) thresholds between the segmentation outputs (S) and the ground-truths (G), where $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$, $F_1 = \frac{2 \times precision \times recall}{precision + recall}$, and IoU $= \frac{S \cap G}{S \cup G}$. Here, TP (true positive) denotes the number of accurately detected cells, FP (false positive) denotes



(a) Stack S1 (b) Stack S2

Figure 2.3: F1 scores vs. IoU thresholds on two real test stacks while training with curvilinear vs rod-shaped data.

the number of incorrectly detected cells, and FN (false negative) denotes the number of missing cells.

2.5 Experimental Results And Discussion

We demonstrate qualitative comparison between the segmentation results from the curvilinear-shaped trained model vs. the rod-shaped trained model on real biofilm stacks S1 and S2 in Fig 2.2. In each row of Fig 2.2, the first column is the maximum intensity projection (MIP) of the raw fluorescence image volume. The second column is the ground-truth annotation of individual cells. The third and fourth columns show the segmentation outputs from curvilinear-shaped trained network and rod-shaped trained network, respectively. The cells which are correctly identified in the segmentation outputs with respect to ground-truth annotation (true positives), are annotated with same color for proper visual comparison. From both rows in Fig 2.2, it is observed that the segmentation output from curvilinear-shaped model is characterized by more correctly identified cells and less touching cells when compared to the corresponding output from rod-shaped trained model. The red arrows on the images indicate locations on the biofilm where the proposed curvilinear-shaped trained model successfully separate the touching cells while the comparative rod-shaped trained model failed to distinguish between them.

In Table 2.1, we demonstrate precision, recall, and F1 score on the segmentation outputs of the stacks S1 and S2. Here we consider that the intersection-over-union (IoU) between a segmentation output and ground-truth annotation is 0.1, i.e., a cell is detected if the overlap between the segmentation mask and the ground-truth mask is 10% or more. From the table, it is evident that for both stacks (S1 and S2), the proposed curvilinear-shaped trained model renders better F1 scores than the rod-shaped trained model. Although in case of S1, the rod-shaped trained model attains

better precision than the curvilinear-shaped trained model, it performs very poorly in terms of recall yielding very low F1 score. The reason behind this result is the presence of many touching cells in the output of the rod-shaped trained model (top row in Fig. 2.2). It is evident that the presence of numerous touching cells results in higher number of false negatives (i.e., lower number of true positives), and at the same time lower number of false positives. Therefore, the model yields low recall score although the precision is high. In case of S2, the proposed curvilinear-shaped trained model outperforms the rod-shaped trained model in terms of precision, recall, and F1 scores. We also plot F1 scores with respect to multiple IoU thresholds for both models in Fig 2.3. From the figure, we see that the F1 scores drop as the IoU threshold values increase. This trend is expected since the segmented cells with volumes smaller than the threshold are not detected as true positives. However, curvilinear-shaped trained outputs maintained significantly higher F1 scores than the rod-shaped trained outputs even at greater IoU thresholds of 0.2 and 0.3 on both stacks.

2.6 Conclusion

We proposed a framework for simulating synthetic 3D biofilms comprising of realistic cellular morphology. We demonstrated that training a simple segmentation network with these realistic-shaped synthetic biofilms significantly improves the single-cell segmentation accuracy on real biofilm images compared to training with conventional rod-shaped synthetic biofilms.

Chapter 3

Volumetric Segmentation of Dense Cell Populations with a Cascade of Deep Neural Networks in Bacterial Biofilm Applications

3.1 Introduction

This chapter presents a solution to the problem of segmenting densely-packed cells in 3D microscopy images. Cell segmentation is a crucial task in image processing that facilitates the understanding of the characteristics of a cellular population. Given a segmentation, the microscopist is able to localize and track cells over time, detect cell division and growth rates, trace cell lineages, and extract volume, shape, and other representative information. These quantitative details can provide insights regarding cellular health and cellular response to certain drugs [55]. While many automatic segmentation approaches have been developed over the years, cell segmentation still remains challenging in certain conditions, such as low signal-to-noise ratio, intra-cellular intensity inhomogeneity, and high cell density, especially in 3D imaging.

We introduce a novel cell segmentation method called DeepSeeded [34], which uses a combination of two deep convolutional networks to predict the initial seeds required by the traditional seeded watershed algorithm for cell segmentation. Our proposed technique has been tested on 3D

microscopy images of bacterial biofilms. Through experiments conducted on a synthetic dataset and two real biofilm datasets, we have shown that proposed approach outperforms existing deep learning methods and a traditional method in terms of various measures of segmentation accuracy.

3.2 Related Works

There exist many classical approaches for cell segmentation, including thresholding methods followed by pixel-grouping via connected components [9, 56], morphological methods based on the watershed transform [57, 10, 38], geometric active contour models [58, 11], and methods using graph cuts [59, 12]. Among these approaches, thresholding methods often suffer from over-segmentation or under-segmentation errors that result in broken cells, rough region boundaries, and clumps of touching cells. Unlike thresholding, active contour models are able to address the intensity inhomogeneity problems and provide smooth segmentation results; however, they face difficulty in separating touching cells in the absence of an initial contour for each cell. The graph cut and watershed-based methods are more suitable approaches when dealing with densely packed overlapping cells. The graph cut techniques first require an initial detection or coarse segmentation of the cell regions from the background and then attempt to split the touching cells into isolated cells by cutting graphs based on conditions such as minimum similarity of node features [60]. While such graph cut methods can improve touching cell separation, their performance can degrade if the initial detection stage fails to detect cells in the regions of heterogeneous brightness. Also, the iterative graph optimization in 3D for large input volumes becomes computationally expensive. In contrast, marker-controlled or seeded watershed methods feature reduced computation and also require tuning a smaller number of hyperparameters compared to graph cuts and active contour methods. However, accurate cell seed estimation is also very challenging in low-contrast and densely packed cell images. Precise seed generation demarcates the desired regions in the image and hence is crucial to successful segmentation by the seeded watershed segmentation [61]. Accurate estimation of seed markers helps avoid over-segmentation and under-segmentation errors, enhancing the algorithm's ability to handle noise and improving the overall robustness [62]. Various approaches have been attempted for extracting these seeds, such as the h-minima (or maxima) based techniques [63, 64] and multilevel thresholding [65, 66, 67]. Seed estimation following these approaches requires tuning specific thresholds or hypereparameters that are not adaptive to new data.

Over the recent years, various data-driven deep learning techniques have been proposed for cell segmentation. One widely adopted approach is to first perform pixel-wise segmentation (also known

as semantic segmentation) using a deep network, e.g., a U-Net [13, 51, 14, 68], and later group pixels into isolated cell instances using classical algorithms, such as watershed or graph partitioning techniques [69, 70, 17, 18]. The U-Net-based pixel-wise segmentation directly performed on low-contrast input images is not very effective in detecting the subtle boundary changes between touching cells, often causing inaccurate classification of the boundary pixels. Moreover, with U-Net results, the later post-processing stage involves various tunable hyper-parameters and hence cannot resolve the touching-cell problem in a data-adaptive fashion. In recent developments, attention-based transformer encoders have also been employed within the U-Net architecture for pixel-wise segmentation tasks [15, 71, 16]. Another notable approach, known as Cellpose [19], estimates spatial gradient maps from the input image and later performs gradient tracking to achieve final instance-wise segmentation. However, such a gradient feature-based method can be easily affected by the noise and heterogeneous illumination present in the input.

Furthermore, various methods have been proposed to perform end-to-end instance-wise segmentation. The region-based convolutional neural networks, namely Mask-RCNN and its variants [20, 21, 72, 68] are widely used instance-wise segmentation approach. The original Mask R-CNN method [20] consists of several key components: a CNN backbone, a region proposal network (RPN) with non-maximum suppression, a RoIAlign layer, and individual prediction heads for instance-wise segmentation. These methods output a bounding box, classification label, and pixel/voxel-wise mask per detected instance. While the Mask-RCNN-based methods have demonstrated significant performance gain in many applications, these methods struggle in situations with many touching/overlapping objects in space due to greedy non-maximum suppression post-processing, as mentioned and demonstrated in the literature [73, 74, 75].

More recently, transformer-based end-to-end instance-wise detection and segmentation methods have been proposed [76, 77]. These approaches employ a combination of transformer encoder-decoder, CNN backbone encoder, and CNN decoder to produce bounding box predictions, class labels, and masks for detected instances. These methods have demonstrated their effectiveness in 2D object detection and cell segmentation tasks. However, their suitability for predicting separate bounding boxes for individual instances in dense 3D cell environments is an area that requires further exploration. In addition, several techniques have been suggested with a specific emphasis on segmenting the cellular soma regions from microscopy images of neuronal cells, for instance, approaches such as the scale fusion segmentation network and the structure-guided segmentation network [78, 79]. Other soma segmentation approaches include a ray-shooting model combined with Long Short-Term Memory (LSTM)-based network [80], and 3D U-Net-based approaches [81, 82].

Recently cell segmentation methods that incorporate the concept of CNN-based distance map prediction, followed by seeded watershed segmentation, have demonstrated great success in segmenting images of densely packed cell populations. Such methods train a convolutional neural network (CNN) to estimate a cell distance map from a low-contrast input image [23, 83]. In this cell distance map, the cell interior pixels are more enhanced than the boundary pixels. However, in the case of many touching cells, an additional map representing the cell border information was found to be more effective. Scherr *et al.* [22] proposed a neighbor distance map in addition to the cell distance map, which utilizes not only touching cells but also close cells in the CNN training process. Similarly, Zhang *et al.* [24] proposed a CNN-based dual distance map prediction approach to estimate a more effective cell border map. Both these approaches perform the final segmentation task by exploiting the seeded watershed algorithm, where the seeds are obtained by thresholding the estimated maps from the CNN. While these methods can improve cell segmentation accuracy by enhancing the cell interior and border from the low-contrast input, the subsequent seed selection stage for the watershed-based segmentation involves tuning various parameters, such as intensity, size, or shape-based thresholding parameters. These thresholds may not be readily applicable to other datasets. Furthermore, in the presence of heterogeneity of intensity, size, or shape among the cells, the choice of global image thresholds may not be appropriate for extracting the cell seeds accurately.

The proposed method, *DeepSeeded*, overcomes several limitations of existing solutions. Firstly, we utilize a CNN for the image regression task, estimating two distance maps from the low-contrast input image stack. However, compared to existing distance map-based solutions [22, 24], we incorporate an effective distance map representation to facilitate the separation of touching cells. Additionally, we propose a specialized loss function to enhance the quality of distance map estimations. Secondly, we leverage another CNN for voxel-wise classification (also known as semantic segmentation), which automatically estimates the seeds required for the seeded watershed algorithm. This additional network eliminates the need for sub-optimal thresholding-based seed estimation. We demonstrate the performance of the proposed method in the segmentation of bacteria cells from 3D microscopy images of densely packed biofilms. Segmentation of individual instances of bacteria from a biofilm image is challenging due to the presence of many touching cells and due to intra-cellular intensity inhomogeneity, which lead to under-segmentation and over-segmentation errors, respectively.

Symbols	Description
$\boldsymbol{x}$	Input 3D cell image
$\tilde{\boldsymbol{x}}^c$	Cell interior-enhanced image
$\tilde{\boldsymbol{x}}^b$	Cell border-enhanced image
$\tilde{\boldsymbol{v}}$	Voxel-wise classified map
$\tilde{\boldsymbol{x}}^l$	Instance labeled segmentation
T	Number of training samples
N	Number of voxels in an image

Table 3.1: Description of symbols

Our Contribution

The main contributions of the proposed method are mentioned as follows:

- We propose an automatic seed estimation approach for the seeded watershed algorithm using a cascade of two deep networks, an image regression network, and a voxel-wise image classification network. Such an approach eliminates the need to tune any hyperparameters during the online/testing phase of the segmentation workflow.
- We propose a novel cell border representation, the 'border neighbor distance map,' to be learned by the regression network for a precise estimation of the border voxels. Such a representation is beneficial for separating touching cells in a densely packed volume.
- We utilize a 3D multi-scale structural similarity index measure (MS-SSIM) as a loss term in combination with an error-based loss to train the regression network. Such a loss function formulation ensures superior image quality of the cell interior and border estimation maps.

This chapter is organized as follows: Section 3.3 presents the theory of the proposed approach. Section 3.4 includes details of the experimental setup and dataset, evaluation metrics, and comparative methods. Experimental results are presented and explained in Section 3.5. Limitations and future potentials are discussed in Section 3.6. Finally, Section 3.7 offers concluding remarks. A number of symbols used in the chapter are listed in Table 3.1.

3.3 Proposed Approach

The proposed segmentation approach is an instance-based segmentation approach that labels every cell in the input image. The segmentation problem is formulated as finding the seeds of a classical watershed algorithm using deep learning. An overview of the proposed approach is demonstrated in Fig 3.1. We then explain the details of each component of the proposed cell segmentation framework.

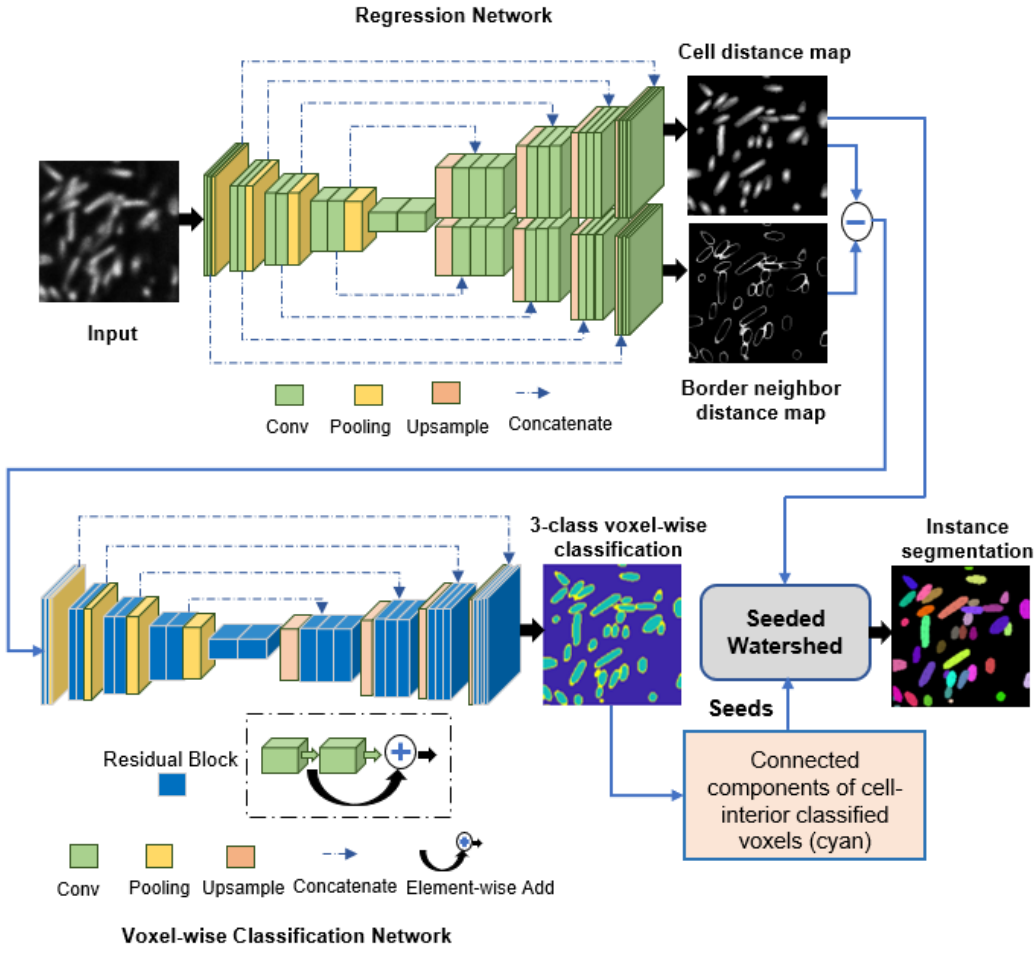


Figure 3.1: Overview of the *DeepSeeded* segmentation workflow. The input and output images demonstrated in this artwork correspond to a 2D slice of a 3D biofilm stack.

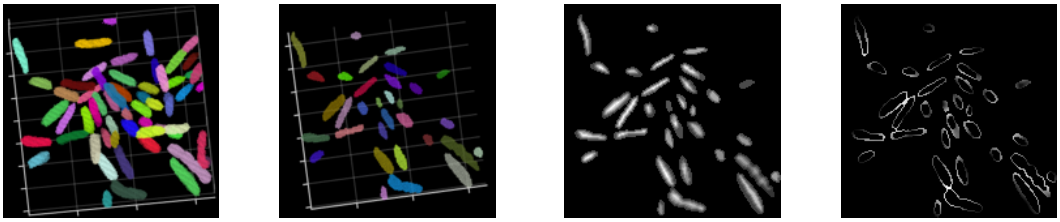
3.3.1 Image Regression Network

Given a potentially low-contrast 3D microscopy image $\boldsymbol{x}$, we produce two new 3D images, $\tilde{\boldsymbol{x}}^c$ and $\tilde{\boldsymbol{x}}^b$, where $\tilde{\boldsymbol{x}}^c$ represents a cell interior-enhanced image and $\tilde{\boldsymbol{x}}^b$ represents a cell border-enhanced image. We implemented a modified two-decoder version of the original single-decoder 3D U-Net [51] to estimate these two maps.

To train the network with groups containing one input and two target images, the ground truth images for two targets $\{\boldsymbol{x^c}, \boldsymbol{x^b}\}$ are generated from a ground truth instance-labeled image $\boldsymbol{x}^l$ of $\boldsymbol{x}$ where $l = \{0, 1, .., L\}$ with L cell instances and 0 as background. We refer to $\boldsymbol{x^c}$ and $\boldsymbol{x^b}$ as 'cell distance map' and 'border neighbor distance map,' respectively. The 'cell distance map' $\boldsymbol{x^c}$ is computed from $\boldsymbol{x}^l$ by calculating the Euclidean distance transform for each of the L cells independently. To compute the 'border neighbor distance map' $\boldsymbol{x^b}$, we first find the border voxels of each cell, and

then for each border voxel, we compute the inverse normalized distance to the nearest neighbor voxel. The detailed steps of computing $\boldsymbol{x^c}$ and $\boldsymbol{x^b}$ are provided in Algorithms 1 and 2.

We propose a precise border map representation to be learned by a regression network in contrast to representations in [22, 24]. In [22], a neighbor distance map is computed pixel-wise for each cell from a ground-truth instance-labeled image. The approach in [24] refines this representation by multiplying the neighbor distance map with a weight matrix so that the boundary pixels/voxels are more enhanced than the cell interior. Although this representation enhances cell boundaries, a percentage of cell-interior voxels can still be highlighted in a cell border representation, especially in dense neighborhoods, such as within a bacterial biofilm. In this work, we propose a 'border neighbor distance map' that computes the neighbor distance only for the border voxels of each cell. This approach yields a sharper border representation. In Fig. 3.2, we qualitatively compare the ground-truth distance maps from the proposed method with those in [22, 24]. The maps are computed in 3D from a ground truth instance-labeled image of an *E. coli*- biofilm, shown in Fig. 3.2a. We demonstrate the maps for a particular 2D slice (Fig. 3.2b) of the image volume.



(a) GT annotation (3D) (b) GT annotation (slice) (c) Proposed CDM (d) Proposed BDM

(e) [24] CDM (f) [24] BDM (g) [22] CDM (h) [22] BDM

Figure 3.2: Qualitative comparison of cell distance map (CDM) and border distance map (BDM) between the proposed method and the competing methods. The maps in the figures (3.2c-3.2h) correspond to the 2D slice in Fig. 3.2b.

We propose a hybrid loss function to train the regression network by incorporating an image quality-based loss term in combination with the error-based loss. The proposed loss function consists of multiscale SSIM loss (MS-SSIM) and smooth L1 loss. While the smooth L1 loss minimizes the error between the ground truth and prediction, the MS-SSIM loss especially helps to maximize the image quality of the prediction with respect to the ground truth. With T number of training samples, our loss term is the sum of the losses to estimate the two maps,

$$
\begin{aligned}
Loss, L_R &= \frac{1}{T}\sum_{t=1}^{T}\left[Cost(\boldsymbol{x}_t^c, \tilde{\boldsymbol{x}}_t^c) + Cost(\boldsymbol{x}_t^b, \tilde{\boldsymbol{x}}_t^b)\right] \\
&= \frac{1}{T}\sum_{t=1}^{T}[\alpha\, C_{\text{SL1}}(\boldsymbol{x}_t^c, \tilde{\boldsymbol{x}}_t^c) + (1-\alpha)\, C_{\text{MS-SSIM}}(\boldsymbol{x}_t^c, \tilde{\boldsymbol{x}}_t^c) \\
&\quad + \alpha\, C_{\text{SL1}}(\boldsymbol{x}_t^b, \tilde{\boldsymbol{x}}_t^b) + (1-\alpha)\, C_{\text{MS-SSIM}}(\boldsymbol{x}_t^b, \tilde{\boldsymbol{x}}_t^b)]
\end{aligned} \tag{3.1}
$$

In equation (3.1), C_{SL1} refers to smooth L1 cost term and $C_{\text{MS-SSIM}}$ represents MS-SSIM cost term. The parameter α is used to control the balance between these two terms. The smooth L1 cost term is further defined in (3.2), where N is the number of voxels in the image. The terms $\tilde{\boldsymbol{x}}(n)$ and $\boldsymbol{x}(n)$ correspond to the predicted distance map and the ground truth distance map values, respectively, at the n^{th} voxel.

$$
C_{\text{SL1}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{N}\sum_{n=1}^{N} \text{SL1}(n) \tag{3.2}
$$

$$
\text{SL1}(n) = \begin{cases} 0.5\,[x(n) - \tilde{x}(n)]^2, & \text{if } |x(n) - \tilde{x}(n)| < 1 \\ |x(n) - \tilde{x}(n)| - 0.5, & \text{otherwise} \end{cases}
$$

Since MS-SSIM is an image quality-based measure that we aim to maximize, the MS-SSIM cost term is computed to minimize the following term,

$$
C_{\text{MS-SSIM}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{N}\sum_{n=1}^{N}[1 - \text{MS-SSIM}(n)] \tag{3.3}
$$

The computation of MS-SSIM involves computing the SSIM metric at multiple scales/resolutions [84]. The SSIM for each pixel/voxel n is defined as follows,

$$
\text{SSIM}(n) = \frac{2\mu_{\boldsymbol{x}}\mu_{\tilde{\boldsymbol{x}}} + C_1}{\mu_{\boldsymbol{x}}^2 + \mu_{\tilde{\boldsymbol{x}}}^2 + C_1} \cdot \frac{2\sigma_{\boldsymbol{x}\tilde{\boldsymbol{x}}} + C_2}{\sigma_{\boldsymbol{x}}^2 + \sigma_{\tilde{\boldsymbol{x}}}^2 + C_2} = l(n) \cdot c(n)
$$

Here, $\mu_{\boldsymbol{x}}$, $\sigma_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}\tilde{\boldsymbol{x}}}$ denote the mean of $\boldsymbol{x}$, the variance of $\boldsymbol{x}$, and the covariance of $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$, respectively. To ensure numerical stability, small constants C_1 and C_2 are used. Means, standard deviations, and covariance are computed with a 3D Gaussian filter of standard deviation σ_G. The terms $l(n)$ and $cs(n)$ represents luminance and contrast sensitivity measures, respectively.

To utilize the SSIM-based image quality measure as a loss function, we especially apply rectified linear unit (ReLU) activation function on those two terms to avoid the negative values in the loss function,

$$
l(n) := max(0, l(n)); \quad c(n) := max(0, c(n))
$$

Algorithm 1 Compute Cell Distance Map

Input: Instance labeled image $\boldsymbol{x}^l$
Output: Cell distance map $\boldsymbol{x}^c$
$\boldsymbol{x}^c \leftarrow zeros(size(\boldsymbol{x}^l))$ ▷ initialize as matrix of zeros
$\boldsymbol{O} \leftarrow$ voxel locations of $\boldsymbol{x}^l$ where $l = 0$
for $l = 1, ..., L$ **do**
    $c^l \leftarrow l^{th}$ cell ▷ coordinates of l^{th} cell
    for p in c^l **do**
        $i, j, k \leftarrow$ location of p in $\boldsymbol{x}^l$
        for q in O **do**
            $d_{pq} \leftarrow E(p, q)$ ▷ Euclidean distance
        end for
        $\boldsymbol{x}^c(i, j, k) \leftarrow min(d_{pq})$
    end for
end for

Algorithm 2 Compute Border Neighbor Distance Map

Input: Instance labeled image $\boldsymbol{x}^l$
Output: Border neighbor distance map $\boldsymbol{x}^b$
$\boldsymbol{x}^b \leftarrow zeros(size(\boldsymbol{x}^l))$ ▷ initialize as matrix of zeros
for $l = 1, ..., L$ **do**
    $c^l \leftarrow l^{th}$ cell
    $c^b \leftarrow$ boundary voxels of c^l
    for p in c^b **do**
        $i, j, k \leftarrow$ location of p in $\boldsymbol{x}^l$
        for $m = 1, ..., L$ and $m \neq l$ **do**
            $c^m \leftarrow m^{th}$ cell
            for q in c^m **do**
                $d_{pq} \leftarrow E(p, q)$ ▷ Euclidean distance
            end for
        end for
        $\boldsymbol{x}^b(i, j, k) \leftarrow 1 - min(d_{pq})$
    end for
end for

Finally, the MS-SSIM for each voxel n is computed over a pyramid of M different resolutions as follows,

$$\text{MS-SSIM}(n) = l_M^{\beta}(n)\, c_M^{\beta}(n) \prod_{j=1}^{M-1} c_j^{\delta_j}(n) \tag{3.4}$$

The hyperparameters α, β, and $\{\delta_j\}$ are set empirically during the offline training stage of the network and have been chosen on a validation set of images.

3.3.2 Voxel-wise Classification Network

The difference map $\tilde{\boldsymbol{d}} = \tilde{\boldsymbol{x}}^c - \tilde{\boldsymbol{x}}^b$ of the two predicted maps from the regression network is provided as input to the classification network. We learn a mapping $F : \tilde{\boldsymbol{d}} \rightarrow \tilde{\boldsymbol{v}}$ to predict the class label k of each voxel in $\tilde{\boldsymbol{d}}$ using a 3D residual U-Net. Each voxel $\tilde{v}_n$ denotes the probability of being classified as class 0, 1, or 2, representing the background, cell interior, and cell border classes, respectively. In order to train the network, the target voxel-wise labeled image $\boldsymbol{v}$ is generated from the corresponding ground truth instance-wise labeled image $\boldsymbol{x}^l$ with L cell instances. In such a target image $\boldsymbol{v}$, a voxel of a cell is considered a border voxel if any of its neighbors has a different cell label. The remaining voxels of that cell are considered cell interior voxels. The cell interior and border voxels are labeled as 1 and 2, respectively, while all background voxels are labeled as 0. With the ground truth and predicted maps, we train the network using a loss function combining soft Dice loss [15] and focal loss [85] as follows,

$$Loss, L_C = \frac{1}{T}\sum_{t=1}^{T}[C_{\text{Dice}}(\boldsymbol{v}_t, \tilde{\boldsymbol{v}}_t) + C_{\text{focal}}(\boldsymbol{v}_t, \tilde{\boldsymbol{v}}_t)] \tag{3.5}$$

where,

$$C_{\text{Dice}}(\boldsymbol{v}, \tilde{\boldsymbol{v}}) = 1 - \frac{2}{K}\sum_{k=0}^{K-1}\frac{\sum_{n=1}^{N} v_k(n)\tilde{v}_k(n)}{\sum_{n=1}^{N}[v_k^2(n) + \tilde{v}_k^2(n)]}$$

$$C_{\text{focal}}(\boldsymbol{v}, \tilde{\boldsymbol{v}}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=0}^{K-1} v_k(n)[1 - \tilde{v}_k(n)]^{\gamma}\log\tilde{v}_k(n)$$

$$\text{and,} \quad \tilde{v}_k(n) = \frac{e^{\tilde{v}_k(n)}}{\sum_{j=0}^{K-1} e^{\tilde{v}_j(n)}}$$

Here, $K = 3$ for three-class voxel-wise classification. The Dice loss enables the network to maximize the overlap of voxels between the ground truth and segmentation. The focal loss mainly aims to minimize the segmentation error on the hard examples, such as cell border voxels.

We illustrate the predicted intermediate maps from the two networks in the proposed *DeepSeeded* approach for an example *E. coli* image stack in Fig. 3.3. The predicted cell distance map and the border neighbor distance map from the regression network are demonstrated in Fig. 3.3b and 3.3c, respectively. It is important to note that we show the border neighbor distance map for a single slice only, as the maximum intensity projection (MIP) view does not provide suitable border visualization. The difference map of the cell distance map and the border neighbor distance map is shown in Fig. 3.3d. The observations from this figure indicate that the background is more distinct, and the cells are better separated compared to the cell distance map alone. Furthermore,

we present the output of the voxel-wise classification network after performing connected components in Fig. 3.3e, which is referred to as the seed-labeled image. Finally, the instance segmentation result after applying the seeded watershed algorithm is illustrated in Fig. 3.3f.

3.3.3 Seeded Watershed

From the voxel-wise classified output $\tilde{\boldsymbol{v}}$, the voxels belonging to the cell interior class (class 1) are exploited to compute the seeds of the watershed algorithm. We perform connected component analysis to label the 1-classified voxels as seeds. The resulting seed-labeled image is denoted as $\tilde{\boldsymbol{s}}$. We then apply the watershed function on the cell interior-enhanced image $\tilde{\boldsymbol{x}}^c$. Starting with the seed locations in image $\tilde{\boldsymbol{s}}$, the seeded watershed algorithm attributes each voxel in $\tilde{\boldsymbol{x}}^c$ to a particular seed. The resulting output is an instance labeled image $\tilde{\boldsymbol{x}}^l$ with L detected cells.

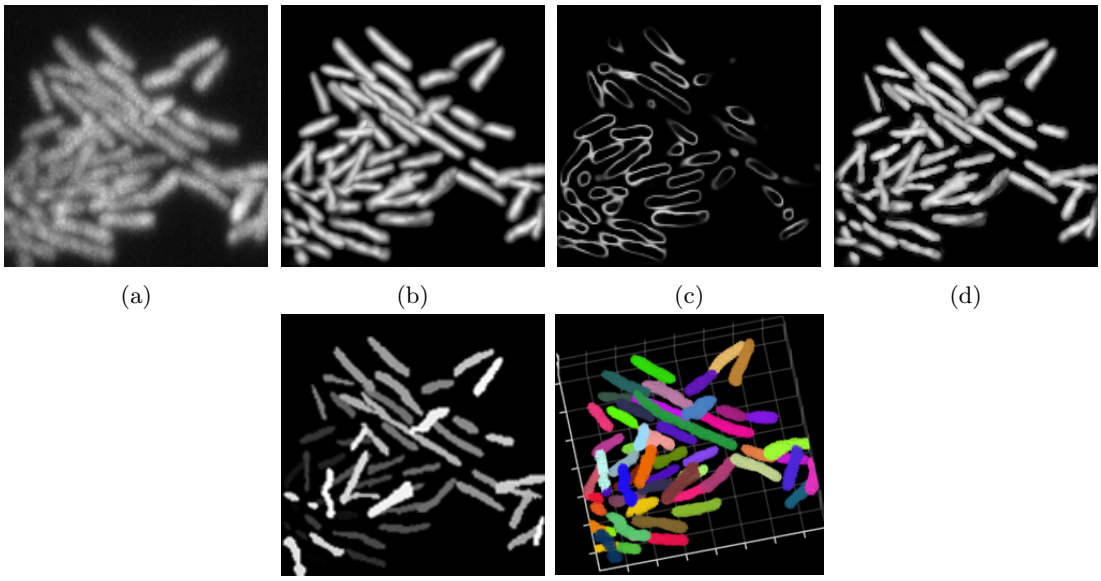


Figure 3.3: (a) Given a raw image stack, the intermediate maps from the two networks and the final segmentation by *DeepSeeded* approach. Here, (a) input stack (MIP), (b) predicted cell distance map (MIP), (c) predicted border neighbor distance map (single slice), (d) predicted difference map (MIP), (e) predicted seed labeled image (MIP), and (f) final segmentation (3D point cloud). The term MIP refers to the maximum intensity projection.

3.4 Experimental Setup

In this section, we provide the implementation details of the cascaded deep learning framework, the description of the dataset, the evaluation metrics, and an account of the comparative methods.

3.4.1 Implementation Details

The regression network has been implemented by modifying the original single encoder-decoder 3D U-Net into two decoders and a single encoder architecture shown in Fig 3.1. The encoder and each of the two decoders consist of five consecutive convolution layers. In the encoding path, each convolution layer performs two $3\times3\times3$ convolutions with ReLU activation and batch normalization, followed by a $2\times2\times2$ max pooling with strides of two. The feature maps used in the five convolution layers of the encoder are 32, 64, 128, 256, and 512. The same number of feature maps are used for both decoders but in reverse order. In each of the two decoding paths, there is a transposed convolution with $2\times2\times2$ strides, followed by two $3\times3\times3$ convolutions along with similar activation and normalization. We have implemented the network in the PyTorch framework. The MS-SSIM loss has been self-implemented for the 3D images. For the smooth L1 loss, PyTorch's built-in function has been exploited. The hyperparameter α in the loss function equation (3.1) is set to 0.4. In SSIM computation, the means, standard deviations, and covariance are calculated using a 3D Gaussian filter with a kernel size of $11 \times 11 \times 11$ and a standard deviation of $\sigma_G = 1.5$. Further, in the MS-SSIM loss term equation (3.4), we have set M=5, $\beta = 0.1333$, and $\{\delta_j\}_{j=1}^{4} = \{0.0448, 0.2856, 0.3001, 0.2363\}$. The network was trained for a maximum of 250 epochs using a batch size of 2. If there is no change in the loss values for 30 consecutive epochs, the network stops training. The initial learning rate is set at 5×10^{-4}, and it is gradually reduced at a rate of 0.25 to a minimum value of 10^{-5}. The Adam optimization is used for adjusting the weights of the network.

The voxel-wise classification network has been implemented by incorporating residual blocks within a 3D U-Net architecture shown in Fig 3.1. Like the regression network, this network consists of five convolutional layers in both encoding and decoding paths with 32, 64, 128, 256, and 512 feature maps. We have included two residual blocks in each of the convolutional layers of the network. The parametric ReLU (PReLU) activation function and instance normalization have been applied after the convolution operation. The kernel sizes for convolution and transposed convolution operations are set to be similar to those used in the regression network. The focal loss parameter is empirically set to $\gamma = 1$. The network has been trained for a batch size of 2, the learning rate of 10^{-4}, and a maximum epoch of 250. We have implemented the network using the open-source PyTorch-based framework MONAI [86].

3.4.2 Dataset

We exploit a 3D synthetic biofilm dataset and a few real biofilm 3D images in our experiment. The synthetic dataset has been generated using a biofilm simulation framework developed in our previous work [33], which can simulate 3D synthetic biofilms consisting of realistic-shaped bacteria cells. We simulated 40 synthetic biofilm stacks of dimensions $x \times y \times z$, where $x, y \in [300, 500]$ and $z \in [100, 200]$. Among them, 10 stacks have been separated as test stacks, 5 stacks for validation, and the rest of the stacks have been used for training. The training and validation set images have been further subdivided into multiple smaller patches of $128 \times 128 \times 64$ by random cropping and data augmentation operations. Also, we have generated 100 synthetic test volumes of $150 \times 150 \times 64$ by random cropping from the original larger test stacks.

We have performed experiments on lattice light-sheet microscopy images [87] of two kinds of real bacteria species, *Escherichia coli* and *Shewanella oneidensis*. We have exploited an *Escherichia coli* image dataset from previously published works [17, 33]. Further, we have acquired fluorescence images of a *Shewanella oneidensis* biofilm, which has a considerably higher cell density than an *Escherichia coli* biofilm. The biofilm of *Shewanella oneidensis* was observed under two different conditions: one with a temporal interval of 5 minutes and another with an interval of 30 seconds. In both cases, each 2D slice was acquired at an exposure time of 10 ms. The resolution is approximately 230 nm in x and y, and 370 nm in z, assuming green fluorescent protein (GFP) excitation and emission. Because manually labeling cells to produce ground truth annotation from dense 3D biofilm images is very laborious and challenging, we created ground truth cell labeling for three *E. coli* and two *S. oneidensis* stacks cropped from the original larger stacks. Two of the *E. coli* stacks have dimensions of $164 \times 166 \times 51$ and $153 \times 154 \times 51$, and the third has a dimension of $150 \times 150 \times 25$. These cropped stacks correspond to three different time points in an *E. coli* image sequence. Among the three stacks, the first stack was used in the training set along with its multiple augmented versions by flip and transpose operations in x-y-z dimensions. The rest of the two *E. coli* stacks were used for testing. The ground truth annotations of the *E. coli* stacks were generated by manually tracing the bacteria cells slice-by-slice in 3D.

For the dense *S. oneidensis* stacks, ground truth annotations were generated in a semi-automatic fashion by manually tracing cell seeds or centroids slice-by-slice in 3D and then applying seeded watersheds on their cell distance maps obtained from the regression network shown in Fig 3.1. The two *S. oneidensis* stacks with ground truth annotations have dimensions of $150 \times 150 \times 25$ and they correspond to two different time points of the *S. oneidensis* sequence with 5 minutes interval.

To further assess the robustness of the models in handling variations in segmentation imagery, we verified segmentation performance qualitatively on additional data. We demonstrated segmentation on another *S. oneidensis* stack with larger dimensions of $200 \times 200 \times 50$. This stack corresponds to a temporal frame of the *S. oneidensis* sequence captured at a 30-second interval.

3.4.3 Evaluation Metrics

We evaluate the cell counting accuracy of our segmentation output S with respect to the ground truth annotation G using per-image cell counting F1 score as follows,

$$CCF1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

If we denote the number of detected cells as N_S and the number of ground truth cells as N_{GT}, TP represents the number of correctly detected cells, $FP = N_S - TP$ represents the number of detected cells that do not exist in GT and $FN = N_{GT} - TP$ represents the number of missing cells in S. We compute $CCF1$ for a range of intersection-over-union (IoU) values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ A cell is considered TP if the percentage of overlapped voxels between S and GT is above a certain IoU threshold. By computing $CCF1$ over a range of IoU values, we can understand how much cell counting accuracy is affected if more cell-volume overlapping is expected.

We also compute the single-cell F1 score, denoted as $SCF1$, to evaluate cell segmentation accuracy. $SCF1$ provides an assessment of the number of voxels that are correctly classified per instance on average in the segmentation result. To calculate $SCF1$, each instance in the segmentation result S is compared with the closest instance in the ground-truth mask G based on their spatial overlap. From this comparison, we determine the true positive voxels (TP^l), false positive voxels (FP^l), and false negative voxels (FN^l) for each matched instance l. The number of matching instances, denoted as N_{match}, can be less than or equal to the total number of cells in the ground-truth mask. The $SCF1$ score indicates how well the segmentation result preserves the cell volume.

$$SCF1 = \frac{1}{N_{match}} \sum_{l=1}^{N_{match}} \frac{2 \times TP^l}{2 \times TP^l + FP^l + FN^l}$$

To further evaluate the accuracy of the segmentation in separating touching cells, we also compute a single-cell boundary F1 score [60] (denoted as $SCBF1$). The score $SCBF1$ tells us per cell how many boundary points match with the contour of the corresponding ground truth instance. In the

following expression of $SCBF1$, subscript b represents the boundary voxels.

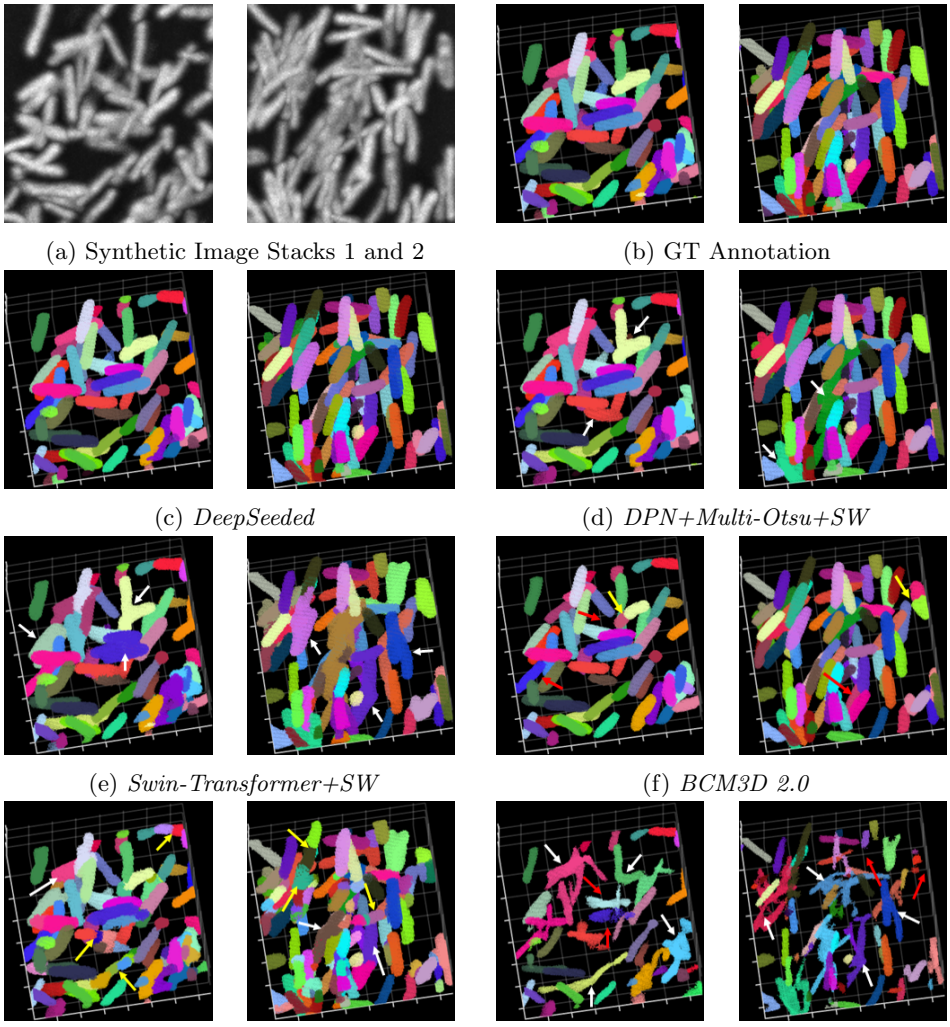
$$SCBF1 = \frac{1}{N_{match}} \sum_{l=1}^{N_{match}} \frac{2 \times TP_b^l}{2 \times TP_b^l + FP_b^l + FN_b^l}$$

3.4.4 Comparative Methods

The performance of the proposed method has been assessed by comparing it against four state-of-the-art deep learning techniques and a classical segmentation approach. We have compared against the popular distance prediction-based cell segmentation method by Scherr et al. [22], which predicts two distance prediction maps using a two-decoder U-Net and later performs the seeded watershed segmentation using the predicted maps. For better comparison, unlike performing the empirical thresholding-based seed selection approach mentioned in the paper, we have performed automatic multi-class Otsu thresholding [88] (three-class in this case) to obtain the seeds. Hence, we call this method a distance prediction network with multi-class Otsu and the seeded watershed, *DPN+Multi-Otsu+SW*. Also, the original paper performs 3D segmentation using a 2D network in a slice-by-slice fashion, whereas we have compared against fully 3D distance predictions by modifying the original 2D network into 3D. We have also compared against a method consisting of a CNN-based pixel-wise segmentation followed by a seeded watershed-based post-processing. While such methods mentioned in the literature [70, 89, 55] exploit a standard U-Net convolutional network, we have adopted a more recent network architecture, Swin Transformer-based U-Net [15] to perform the 3D pixel-wise classification task. We call this method *Swin-TransNet+SW*. The proposed method has also been compared against the popular cell-instance segmentation network *Cellpose* pipeline [19]. We have further compared against the latest deep learning-based 3D biofilm segmentation approach named *BCM3D 2.0* [24], which first performs dual distance transform predictions using a regression CNN, followed by a multi-stage thresholding-based seed selection for the seeded watershed segmentation. Finally, we have compared against a classical segmentation approach exploited in a recent paper [55] named *MARS*, which performs seeded watersheds using the h-minima (or maxima) operator. We used the publicly available code repositories mentioned in the corresponding papers to execute the comparative methods.

3.5 Experimental Results And Discussion

We demonstrate the qualitative comparison of the segmentation results on two synthetic biofilm test stacks in Fig. 3.4. The images shown in Fig. 3.4a are the maximum intensity projections (MIPs) of



(a) Synthetic Image Stacks 1 and 2 (b) GT Annotation

(c) *DeepSeeded* (d) *DPN+Multi-Otsu+SW*

(e) *Swin-Transformer+SW* (f) *BCM3D 2.0*

(g) *Cellpose* (h) *MARS*

Figure 3.4: Qualitative evaluation on synthetic 3D biofilm images. The white, yellow and red arrows indicate various locations of touching cells, broken cells, and missing cells, respectively.

the original 3D inputs. The ground truth annotations and the corresponding segmentation outputs are visualized in 3D. The cells which are correctly identified in the segmentation result are annotated with the same color as in GT annotation for proper visual comparison. From the figure, we observe that the *DeepSeeded* method can effectively separate the touching cells and prevent the individual cell from breaking into multiple segments. We also observe in Fig. 3.4f that the *BCM3D 2.0* method effectively addresses the touching cell separation for these synthetic image stacks. However, the results from *BCM3D 2.0* also contain a few broken and missing cells, which may result from the multiple stages of thresholding in the seed selection process. Moreover, one may notice that the results from the *DPN+Multi-Otsu+SW* method contain several unresolved touching cells. We also find that compared to the results from these distance prediction-based methods in Fig. 3.4c, Fig. 3.4f,

and Fig. 3.4d, the results from the *Cellpose*, *Swin-TransNet+SW*, and *MARS* method contain more errors. The enhancement of the cell interior and border information through distance predictions appears to make the subsequent segmentation task easier. The results also indicate that the *Cellpose* method mostly suffers from over-segmentation errors resulting in broken cell segments. Since the method is based on estimating spatial gradient features, the intra-cellular intensity inhomogeneity may lead to over-segmentation errors. Further, we observe in Fig. 3.4h that the classical *MARS* approach suffers heavily in separating the touching cells and preserving the cell volume.

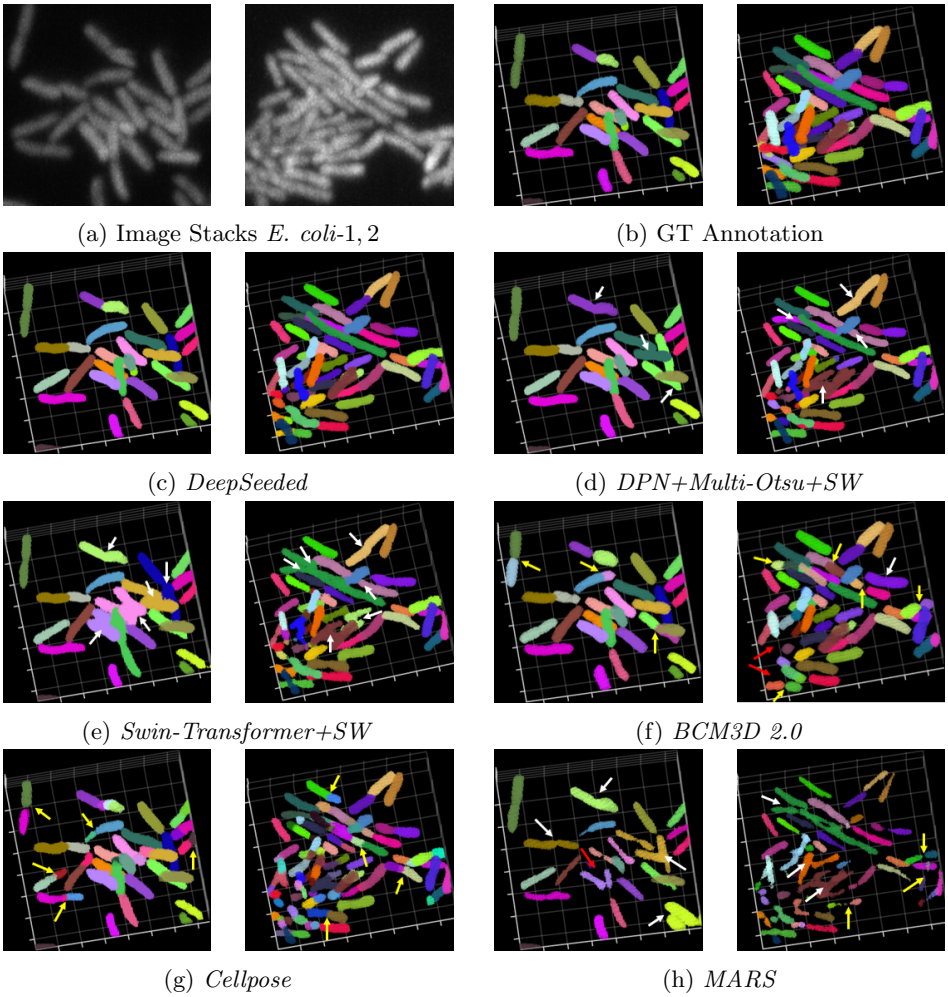
In Table 3.2, we report the mean and standard deviation of the quantitative evaluation measures on 100 synthetic biofilm test stacks. The $CCF1$ scores are reported for IoU values of 0.1 and 0.5. From the table, we notice that all three quantitative scores comply with our visual observation from Fig. 3.4. The proposed method achieves higher average scores for each quantitative segmentation accuracy measure.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW	0.893 ± 0.043	0.827 ± 0.076	0.883 ± 0.024	0.957 ± 0.020
Swin-Transformer+SW	0.844 ± 0.058	0.485 ± 0.143	0.664 ± 0.034	0.686 ± 0.041
MARS	0.651 ± 0.084	0.017 ± 0.010	0.377 ± 0.049	0.427 ± 0.021
BCM3D 2.0	0.877 ± 0.047	0.863 ± 0.058	0.881 ± 0.028	0.962 ± 0.018
Cellpose	0.810 ± 0.057	0.440 ± 0.087	0.663 ± 0.025	0.743 ± 0.031
DeepSeeded	**0.948 ± 0.024**	**0.915 ± 0.046**	**0.904 ± 0.023**	**0.980 ± 0.014**

Table 3.2: Quantitative evaluation on 100 synthetic 3D biofilms

We also demonstrate the qualitative segmentation results on real biofilm stacks in Fig. 3.5 and Fig. 3.6. The images shown in Fig. 3.5a and Fig. 3.6a are the maximum intensity projections (MIPs) of the corresponding 3D inputs. The GT annotations and the segmentation results from different methods are visualized in 3D. Overall, the *DeepSeeded* method outperforms competing approaches in segmenting individual bacteria cells from two kinds of real microscopy biofilms. Further, we notice that the *BCM3D 2.0* method causes more broken and missing cells on these real biofilm stacks compared to its results on synthetic data. It is also observed that the *DPN+Multi-Otsu+SW* and *Swin-TransNet+SW* methods result in many touching cells in segmenting the dense *Shewanella* stacks. The higher cell density of the *Shewanella* biofilms makes single-cell segmentation more challenging. In addition, the *Cellpose* and *MARS* methods also produce less effective segmentations for the real biofilm stacks causing broken and touching cells.

In Tables 3.3, 3.4, 3.5, and 3.6, we also report the quantitative measures on these four real biofilm volumes. The differences in the challenges posed by each type of biofilm (*Shewanella* with higher cell density and *E. coli* with lower image resolution) require separate reporting of the segmentation



(a) Image Stacks *E. coli*-1, 2 (b) GT Annotation

(c) *DeepSeeded* (d) *DPN+Multi-Otsu+SW*

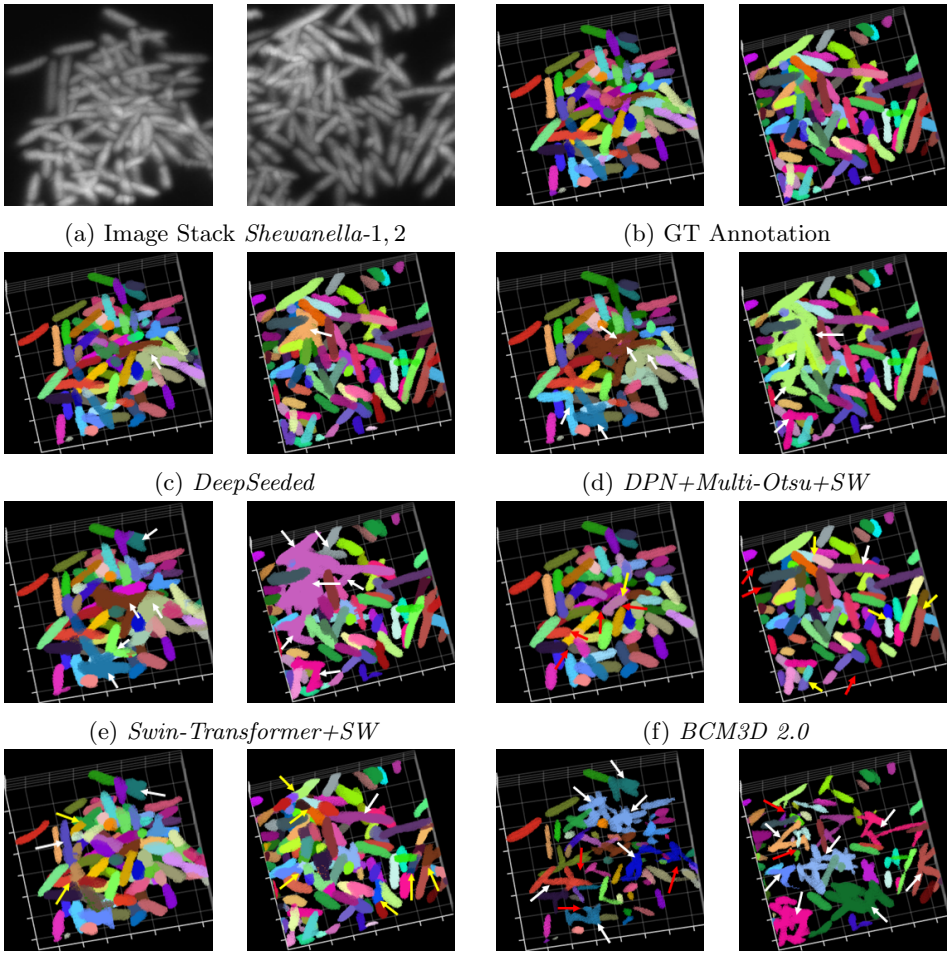
(e) *Swin-Transformer+SW* (f) *BCM3D 2.0*

(g) *Cellpose* (h) *MARS*

Figure 3.5: Qualitative evaluation on two 3D *E. coli* images. The white, yellow and red arrows indicate various locations of touching cells, broken cells, and missing cells, respectively.

results for each biofilm type, resulting in individual tables for specific biofilm stacks instead of a consolidated table. From the results presented in these tables, it is evident that the *DeepSeeded* method achieves higher scores in all three quantitative measures on each of the four image stacks. Also, we observe that the difference between the $CCF1$ scores at IoU values of 0.1 and 0.5 is small for the proposed method on all four stacks, while the competing methods have a larger difference between the corresponding $CCF1$ scores at IoU of 0.1 and 0.5. This reflects that the proposed method not only separates individual cells but also preserves the size or volume of the cells. This size information can be valuable in comprehending cellular characteristics and tracking cell behavior over time.

In order to provide further evidence of the effectiveness of the *DeepSeeded* method, additional test results on another real biofilm stack, denoted as "*Shewanella*-3" with dimensions of $200 \times 200 \times 50$, are presented in Fig. 3.7. Here, we qualitatively compare the segmentation results obtained by



(a) Image Stack *Shewanella*-1, 2 (b) GT Annotation

(c) *DeepSeeded* (d) *DPN+Multi-Otsu+SW*

(e) *Swin-Transformer+SW* (f) *BCM3D 2.0*

(g) *Cellpose* (h) *MARS*

Figure 3.6: Qualitative evaluation on two 3D *Shewanella* images. The white, yellow and red arrows indicate various locations of touching cells, broken cells, and missing cells, respectively.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW	0.931	0.828	0.795	0.898
Swin-Transformer+SW	0.852	0.407	0.674	0.734
MARS	0.836	0.173	0.514	0.580
BCM3D 2.0	0.921	0.825	0.793	0.899
Cellpose	0.800	0.286	0.584	0.624
DeepSeeded	**1.000**	**0.840**	**0.853**	**0.909**

Table 3.3: Quantitative evaluation on stack *E. coli*-1

different approaches on the *Shewanella*-3 image. To highlight the segmentation errors in various methods, we have included arrows in the figures.

From Fig. 3.7, overall, our observations indicate that the *DeepSeeded* method performs better than the competing approaches in accurately segmenting individual bacteria cells. We have also noticed that the *BCM3D 2.0* method results in several broken and missing cells, which could be at-

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW	0.855	0.327	0.618	0.713
Swin-Transformer+SW	0.800	0.434	0.635	0.752
MARS	0.600	0.074	0.327	0.351
BCM3D 2.0	0.842	0.316	0.593	0.666
Cellpose	0.671	0.197	0.560	0.658
DeepSeeded	**0.937**	**0.829**	**0.800**	**0.912**

Table 3.4: Quantitative evaluation on stack *E. coli*-2

tributed to multiple thresholding steps during seed selection. Additionally, the segmentation results obtained using the *DPN+Multi-Otsu+SW* and *Swin-TransNet+SW* methods exhibit numerous instances of touching cells. The gradient-based *Cellpose* method tends to oversegment, leading to broken cells in the output. Lastly, the classical *MARS* method produces less effective segmentation output, resulting in a high number of touching cells. Integrating an image quality-specific loss term and a refined cell border representation into the training of the regression network, along with a data-driven seed estimation using an additional network, contributed to the success of the proposed method in dense cell segmentation compared to competing approaches.

We also report the time taken for model building (i.e., offline training stage) and the online testing stage for the proposed method and other competing approaches. All deep learning-based methods were trained for 250 epochs using a machine equipped with an NVIDIA TITAN RTX GPU with 24 GB memory. The *BCM3D 2.0* method required an average of 72.1 seconds per epoch during training, resulting in a total training time of 5.0 hours. The average testing time on a single instance was 1.7 seconds. As for the *DPN+Multi-Otsu+SW* method, the per epoch training time averaged at 90.5 seconds, leading to an overall training time of 6.3 hours. The average testing time on a single instance was 4.6 seconds. Regarding the *Swin-TransNet+SW* method, each epoch's training time averaged at 20.8 seconds, resulting in a total training time of 1.4 hours. The average testing time on a single instance was 3.1 seconds. For the *Cellpose* method, the per epoch training time was approximately 60.5 seconds, leading to an overall training time of 4.2 hours. The average testing time on a single instance was 10.0 seconds. Since the *MARS* method is a classical approach, it does not require a training phase. The average testing time on a single instance was 1.5 seconds. In our proposed *DeepSeeded* method, the per epoch training time for the regression network (*Net-1*) averaged at 92.1 seconds, resulting in a total training time of 6.4 hours. The per epoch training time for the voxel-wise classification network (*Net-2*) was approximately 17.5 seconds, leading to an overall training time of 1.2 hours. The average testing time on a single instance was 4.9 seconds.

Ablation Study

In order to understand the individual contribution of each of the two networks in the proposed method, we also demonstrate the results of an ablation study in Tables 3.7 and 3.8 on the real biofilm stacks *E. coli*-2 and *Shewanella*-2, respectively. In both tables, the first row lists the segmentation scores exploiting the regression network (*Net-1*) combined with the multi-class Otsu thresholding and seeded watershed. The second row lists the scores corresponding to residual U-Net as the voxel-wise classification network (*Net-2*) followed by the seeded watershed. From the quantitative scores presented in these two tables, it is clear that the proposed architecture *DeepSeeded* provides the best segmentation performance in terms of all three quantitative measures, irrespective of the type of biofilm images. We also see from the results that the *Net-1+Multi-Otsu+SW* method achieves better scores than the *Net-2+SW* method. The superiority is due to the enhancement of the cell interior and border by the regression network, which makes the subsequent segmentation task easier than direct segmentation on the raw inputs.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW	0.821	0.680	0.786	0.883
Swin-Transformer+SW	0.800	0.450	0.670	0.770
MARS	0.562	0.123	0.446	0.541
BCM3D 2.0	0.864	0.722	0.750	0.866
Cellpose	0.825	0.402	0.673	0.783
DeepSeeded	**0.885**	**0.874**	**0.964**	**0.967**

Table 3.5: Quantitative evaluation on stack *Shewanella*-1

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW	0.834	0.717	0.811	0.925
Swin-Transformer+SW	0.775	0.539	0.705	0.822
MARS	0.577	0.110	0.427	0.538
BCM3D 2.0	0.833	0.660	0.756	0.876
Cellpose	0.817	0.435	0.650	0.771
DeepSeeded	**0.918**	**0.900**	**0.976**	**0.977**

Table 3.6: Quantitative evaluation on stack *Shewanella*-2

3.6 Limitations and Future Potentials

The proposed method *DeepSeeded* demonstrates significant performance gain compared to existing popular solutions when segmenting touching instances in dense cellular environments, such as in bacterial biofilms. However, there are several areas where further improvement can be made. Since the proposed segmentation framework addresses cell segmentation in 3D, the memory requirement

during training increases with more training data, even when trained with smaller training patches. Such limitation can be addressed by incorporating memory-efficient CNN architectures as introduced in recent literature [90, 91, 92, 93]. The memory-efficient CNN approaches leverage implicit 3D representations, known as occupancy values [91], to overcome the high computational complexity of traditional 3D CNNs. By learning a continuous decision boundary in a function space instead of a dense voxelized representation, these networks become significantly more memory efficient than traditional CNNs on 3D data. In our proposed *DeepSeeded* framework, we can incorporate such memory-efficient architectures instead of traditional U-Net-based CNNs for the regression and semantic segmentation tasks. Additionally, our two loss functions can be jointly learned in a multi-task framework using hypernetworks [94]. Such hypernetworks can optimize the weights of a single network for multiple tasks on hand. In the future, such additional features can be included in the proposed segmentation approach while still retaining the key benefits of the method, including effective cell border representation and specialized image quality-oriented loss in an initial enhancement task and later voxel-wise classification for seed estimation of the watershed.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
Net-1+Multi-Otsu+SW	0.850	0.679	0.770	0.882
Net-2+SW	0.786	0.394	0.664	0.760
DeepSeeded	**0.937**	**0.829**	**0.800**	**0.912**

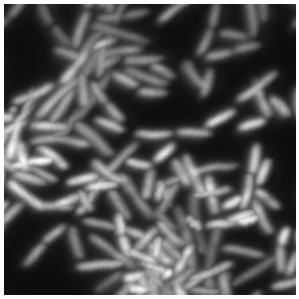
Table 3.7: Ablation study result on stack *E. coli*-2

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
Net-1+Multi-Otsu+SW	0.854	0.806	0.933	0.934
Net-2+SW	0.836	0.531	0.690	0.800
DeepSeeded	**0.918**	**0.900**	**0.976**	**0.977**

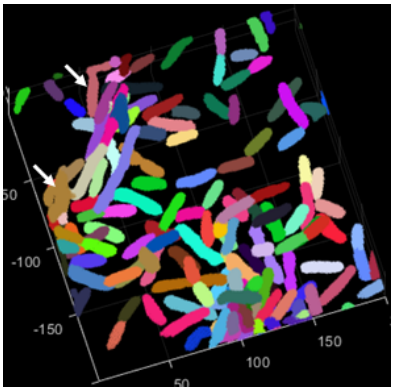
Table 3.8: Ablation study result on stack *Shewanella*-2

3.7 Conclusion

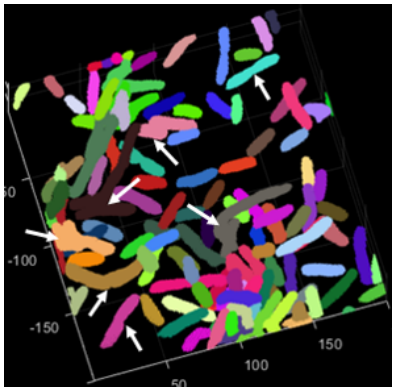
This chapter introduced a novel deep learning-based 3D cell segmentation approach DeepSeeded to effectively segment touching cells in a densely packed microscopy image volume. We devised the segmentation problem as estimating the seeds of a classical watershed algorithm using a hybrid deep-learning model consisting of an image regression network followed by a voxel-wise image classification network. The regression network incorporates a specialized image quality-specific loss term and a refined cell border representation during training, resulting in highly enhanced cell interior and



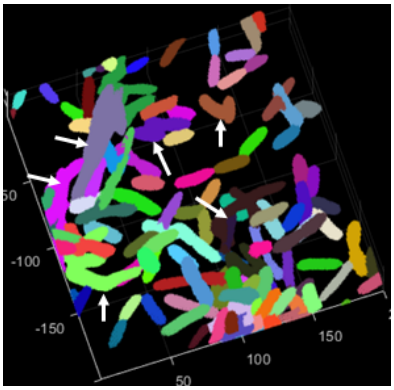
(a) Image Stack *Shewanella*-3



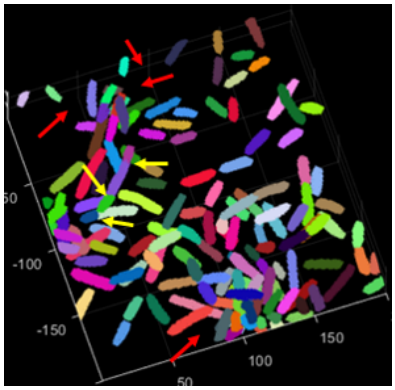
(b) *DeepSeeded*



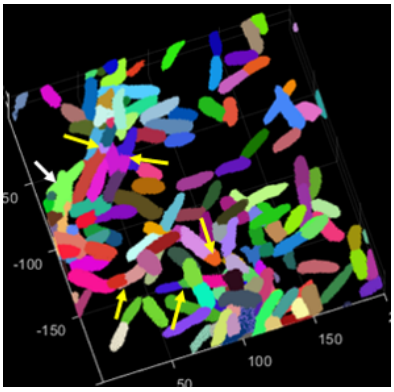
(c) *DPN+Multi-Otsu+SW* (Scherr, 2020)



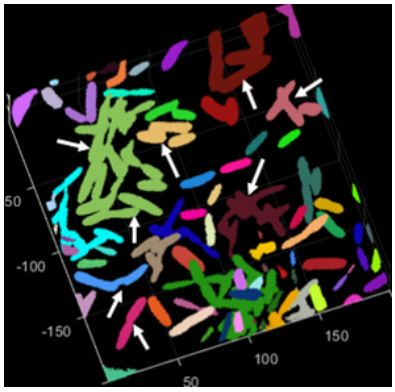
(d) *Swin-TransNet+SW* (Hatamizadeh, 2022a)



(e) *BCM3D 2.0* (Zhang, 2022)



(f) *Cellpose* (Stringer, 2021)



(g) *MARS* (Kar, 2022)

Figure 3.7: Qualitative evaluation on a 3D *Shewanella* image. The white, yellow, and red arrows indicate various locations of touching, broken, and missing cells, respectively.

border estimation maps. The voxel-wise classification network enables data-adaptive prediction of cell seeds for the watershed algorithm, eliminating the need for sub-optimal thresholding. We showed experimental results in segmenting bacteria cells from 3D microscopy images of densely packed biofilms. The proposed method achieved better segmentation results in qualitative comparison and in terms of all the adopted quantitative evaluation measures against the state-of-the-art cell segmentation methods.

Chapter 4

Deep Temporal Sequence Classification for Automatic Cell Tracking in Dense 3D Microscopy Videos of Bacterial Biofilms

4.1 Introduction

This chapter proposes a solution to the cell tracking problem in 3D microscopy videos of densely populated bacterial biofilms. Cell tracking in time-lapse microscopy image sequences is a challenging multi-object tracking task that is essential for biologists interested in controlling and studying the behavior of a cell population under investigation. Since a large number of cell instances needs to be tracked to draw statistically significant conclusions in biological studies, accurate and robust automatic tracking approaches are required. An automatic tracking process involves identifying and linking instances of the same biological cell and their offspring in consecutive frames of a microscopy video. The accurate reconstruction of observed cell trajectories enables researchers extract various biophysical parameters, such as velocity, acceleration, cell division rate, cell appearance, and death rate, which provide quantitative insights into the underlying dynamic processes of the cell population [2, 95]. The cell tracking problem often becomes challenging to solve in the presence of high

cell density, high cell migration rate, frequent division events, non-smooth cellular motion, and low frame rates.

In this chapter, we propose a data-driven cell tracking approach *DenseTrack* [35] to track bacterial cells in 3D lattice light-sheet image sequences of biofilms. The proposed method integrates deep learning with model-based techniques to formulate an effective solution for tracking bacterial cells and detecting their offspring in crowded image scenarios.

4.2 Related Works

There are two main categories of automatic cell tracking methods: tracking by contour evolution and tracking by detection. The contour evolution-based methods involve finding the object contour in the current frame given an initial contour from the previous frame [96, 97, 98, 99]. They solve the segmentation and tracking tasks simultaneously by solving an iterative PDE-based energy functional. In contrast, tracking by detection approach separates the segmentation and tracking task by first performing the segmentation of the individual instances in all the frames and then establishing the temporal associations between the segmented cells of consecutive frames [100, 101, 102]. While tracking by contour evolution are effective in certain scenarios, for instance, where morphological changes of cells are imaged in high magnification, detection based approaches are more suitable with lower frame rates, high cell density and frequent cell divisions scenarios [103, 104]. Also, less computational complexity allows the detection based methods to be widely adopted in real-time tracking of a larger number of cells over longer period of time. In this chapter, we focus on tracking by detection, and present an algorithm that can be used to effectively track bacteria cells over time from 3D temporal image sequences of bacterial biofilms.

Over the years, numerous tracking by detection approaches has been proposed. The simplest methods use basic nearest-neighbor techniques to match cells between frames based on features such as intensity distribution, morphology, and size [25, 105]. More complex features, such as features of the cell's neighborhood [106] or features derived from a graph structure [107] have also been exploited for instance matching. However, nearest-neighbor methods rely on a user-defined Euclidean distance function or an exponential similarity function for correspondence matching, which may not always provide the correct matches between frames. There also exist graph-based tracking approaches where cells are represented as nodes in a graph, and association hypotheses are represented as edges linking the nodes [26, 108, 109, 27]. This allows the tracking problem to be formulated as a graph-matching problem. Further, probabilistic approaches for correspondence finding have also

been proposed. These include joint probabilistic data association (JPDA) [110, 111] and multiple hypothesis-based tracking (MHT) [29, 112, 113, 114]. The classical Kalman filter or its probabilistic variants have also been used to predict the position of the cells in the next frame [115, 28], which have shown promise in tracking high motility cell populations. While these traditional methods have been useful, they often require manual tuning of many parameters and make simplistic assumptions about cell behavior (e.g., choice of particular cell motion model, probability of object appearance, and disappearance in the field of view) that may not always hold true.

Recently, several deep learning-based tracking methods have been proposed for cell tracking, which are computationally much more efficient compared to the traditional methods. One such approach modeled cell tracking as an edge classification problem in a direct graph using a graph neural network [30]. However, this method can be challenging for a dense cell population with a long temporal sequence. Another approach used two separate U-Nets for cell likelihood detection and motion estimation [116]. Other recent approaches include a deep reinforcement learning method [31] and a pipeline of Siamese networks [32]. However, these methods do not incorporate temporal history to predict the association in the next frame, which may be necessary as a cell may be poorly imaged or segmented in some frames but better detected in neighboring frames. Also, these methods do not explicitly enforce one-to-one matching between successive frames, which can prevent erroneous associations between one-to-multiple instances. In our proposed approach, we address these limitations to develop an effective tracking strategy for tracking bacterial cells over time within a dense environment.

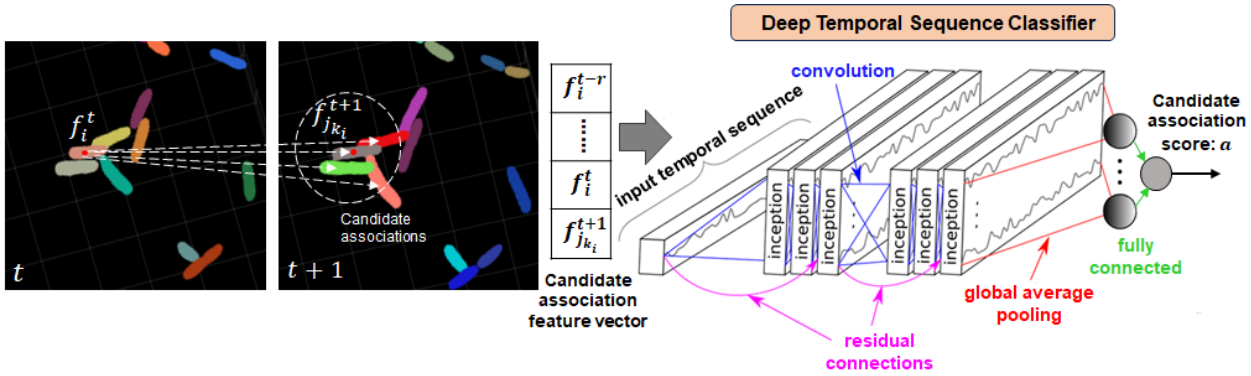
Our Contribution

The key contributions of the proposed method are outlined below:

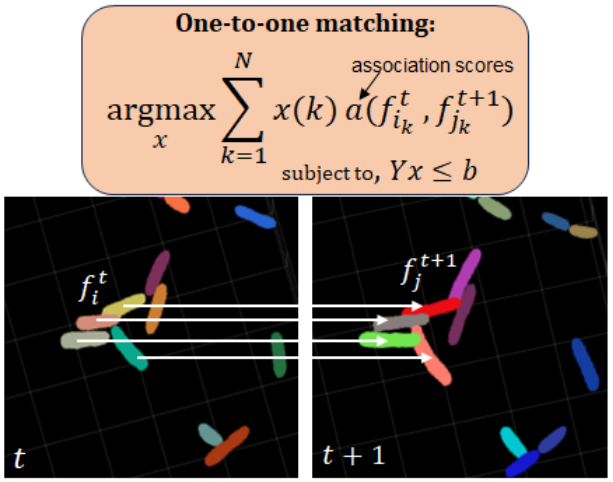
- Our approach to frame-by-frame association incorporates a deep learning-based temporal sequence classifier, which computes the association scores for the potential matches in the subsequent frame. We then solidify one-to-one matching by leveraging the confidence scores provided by the classifier.

- We leverage the near-temporal history to calculate an association score for the potential matches, rather than solely relying on features from the current frame and the next frame.

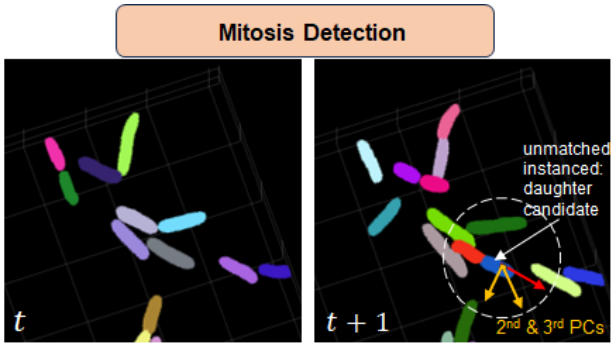
- In the context of cell division detection, we introduce a novel strategy that entails Eigen decomposition of unmatched instances in the following frame.



(a)



(b)



(c)

Figure 4.1: Overview of the proposed tracking approach *DenseTrack*. In (a) and (b), we illustrate our frame-by-frame matching technique that involves computing deep learning-based association scores and utilizing these scores in an one-to-one matching optimization. (c) represents that a cell division event can be detected by finding the neighbor instance with minimum projection along 2^{nd} and 3^{rd} principal components of the unmatched instance in frame $t+1$.

This chapter is organized as follows: Section 4.3 presents the theory of the proposed approach. Section 4.4 includes details of the experimental setup. Experimental results are presented and explained in Section 4.5. Finally, Section 4.6 offers concluding remarks.

4.3 Proposed Approach

In this section, first we define the cell tracking problem and then propose an effective strategy to track cells in 3D time-lapse microscopy videos of biofilms. An overview of the proposed approach is illustrated in Fig. 4.1.

4.3.1 Problem Statement

Let us consider an image sequence, denoted by $\boldsymbol{S} = \{\boldsymbol{F}^t\}_{t=1}^{T}$, which comprises T frames. Let L be the number of biological cells present in this sequence. The cell tracking problem can be stated as follows, (1) determine the trajectory of each biological cell and (2) identify the parent of each biological cell in cases where cell existence is due to cell division. For each biological cell, we need to calculate a set of information represented by $\mathcal{T}_l = \{t^l_{init}, t^l_{fin}, \boldsymbol{C}^l, P(l)\}$. Here t^l_{init} and t^l_{fin} refer to the first and last time points in which the l^{th} cell appears in the sequence, respectively. $\boldsymbol{C}^l$ represents the set of coordinates of the l^{th} cell from the first frame t^l_{init} to the last frame t^l_{fin}. Finally, $P(l)$ is a function that identifies the parent cell of the l^{th} cell, where $P(l) = l^{'}$ if $l^{'}$ is the parent of cell l, and $P(l) = 0$ if the cell appearance is not due to cell division. The objective of cell tracking is to obtain the set $\{\mathcal{T}_1, ..., \mathcal{T}_L\}$.

4.3.2 Tracking Solution

To solve the problem, our method involves initially matching cell instances across consecutive frames, followed by the detection of division events and the establishment of complete trajectories.

Frame-by-Frame Association

Let $\boldsymbol{F}^t = \{\boldsymbol{f}^t_i | i = 1, 2, .., m\}$ and $\boldsymbol{F}^{t+1} = \{\boldsymbol{f}^{t+1}_j | j = 1, 2, .., n\}$ denote two consecutive frames with m and n cell instances, respectively, where each instance is represented by a feature vector $\boldsymbol{f}$. For each instance $\boldsymbol{f}^t_i$, in frame t, there exist several matching candidates in frame $t+1$, represented by the set $\boldsymbol{M}_i = \{(\boldsymbol{f}^t_i, \boldsymbol{f}^{t+1}_{j_{k_i}}) | k_i = 1, 2, ..., N_i\}$. These candidates are selected from the neighborhood of the projected location of $\boldsymbol{f}^t_i$ in frame $t+1$. Our objective is to determine the likelihood that a candidate association is correct. For each of the candidate k_i associations, we create a spatio-temporal feature vector, $\boldsymbol{f}^{tem}_{i,(j_{k_i})} = [\boldsymbol{f}^{t-r}_i, ..., \boldsymbol{f}^t_i, \boldsymbol{f}^{t+1}_{j_{k_i}}]$. This vector is formed by concatenating the feature vector at time t with the feature vectors from the preceding r time frames and the feature vector of the candidate at time $t+1$. Representative features to characterize $\boldsymbol{f}_i$ at a specific time point include

3D spatial coordinates and 3D bounding box measures around the instance. With a chosen value of $r = 2$, the resulting $\boldsymbol{f}^{tem}_{i,(j_{k_i})} = [\boldsymbol{f}^{t-2}_i, \boldsymbol{f}^{t-1}_i, \boldsymbol{f}^{t}_i, \boldsymbol{f}^{t+1}_{j_{k_i}}]$ is a 36-dimensional feature vector.

By leveraging the spatio-temporal information of the k^{th}_i association, we propose computing the probability that this association is correct, denoted as $P[y = 1|\boldsymbol{f}^{tem}_{i,(j_{k_i})}]$, using a deep temporal sequence classification network. The confidence score of the classifier serves as the association probability or association score, expressed as $a(\boldsymbol{f}^t_i, \boldsymbol{f}^{t+1}_{j_{k_i}}) = \mathcal{R}(\boldsymbol{f}^{tem}_{i,(j_{k_i})}; \boldsymbol{\Theta})$, where the network $\mathcal{R}$ has been trained to differentiate between correct and incorrect associations ($y = 1$ or 0), with its parameters represented by $\boldsymbol{\Theta}$. InceptionTime [117], a widely adopted time-series convolutional neural network model based on the Inception architecture, has been chosen for this classification task. By integrating Inception modules along with residual connections, the InceptionTime architecture aims to mitigate overfitting and vanishing gradient issues. Moreover, through the stacking of multiple Inception modules consecutively, the network can capture latent hierarchical features at various resolutions. With N_i possible associations for the i^{th} instance, there are a total of $N = \sum_{i=1}^{m} N_i$ possible associations between frame t and $t + 1$, such that $\boldsymbol{M} = \cup_{i=1}^{m} \boldsymbol{M}_i$ exist. The network $\mathcal{R}$ is employed to compute an association score for all N associations.

Now, we enforce one-to-one matching between frames t and $t + 1$ by solving a constrained optimization problem. The objective is to choose the associations from the N potential associations that maximize the sum of the association scores. Mathematically, the optimal matching approach involves searching for a solution represented by a binary vector $\boldsymbol{x}_0 = \{0, 1\}^N$ that maximize the objective function presented in equation (4.1),

$$\boldsymbol{x}_0 = \underset{\boldsymbol{x}\in\{0,1\}^N}{\arg\max} \sum_{k=1}^{N} \left(\boldsymbol{x}(k)\, a(\boldsymbol{f}^t_{i_k}, \boldsymbol{f}^{t+1}_{j_k})\right) \tag{4.1}$$

The matching constraint that ensures bi-directional one-to-one correspondence for the optimization in (4.1) can be expressed as follows,

$$\boldsymbol{Y}\boldsymbol{x} \leq \boldsymbol{b} \tag{4.2}$$

where $\boldsymbol{Y}$ represents a $(m+n)\times N$ dimensional system matrix and $\boldsymbol{b}$ represents a $(m+n)$ dimensional vector of ones. The system matrix $\boldsymbol{Y}$ is designed as follows,

$$\boldsymbol{Y}(q, k) = \begin{cases} 1, & \text{if } q = i_k \text{ or } j_k \\ 0, & \text{otherwise} \end{cases}; \quad q = 1, 2, ..., (m + n) \tag{4.3}$$

The entries of the k^{th} column of $\boldsymbol{Y}$ indicate which cell instances in frame t and $t+1$ correspond to the k^{th} possible match $(\boldsymbol{f}_{i_k}^t, \boldsymbol{f}_{j_k}^{t+1})$ in $\boldsymbol{M}$, where $k = 1, 2, ..., N$. The solution to the optimization problem in equation (4.1) can be obtained by adopting an iterative search-based approach. The algorithm iterates for each instance in frame t to identify its matching candidate in $t+1$ with the highest association score. In cases where two instances from t are matched with a single instance in $t+1$, the instance with the higher association score is considered correct, and the other instance is assigned to its candidate with the next highest association score. The search algorithm continues until all matched candidates in $t+1$ are unique cell IDs. The computational complexity of the matching algorithm is $\mathcal{O}(m \log n)$. The pseudocode of the proposed one-to-one matching algorithm is presented in Algorithm 3.

After performing frame-by-frame association between any two consecutive frames t and $t+1$, the matched instances ($\boldsymbol{x}(k) = 1$) in frame $t+1$ are assigned the same identification numbers or cell ids as their corresponding instances in frame t. The unmatched instances ($\boldsymbol{x}(k) = 0$) in frame $t+1$ are labeled with new cell ids. This matching process is performed across the entire sequence of frames in the video.

Cell Division Detection

To identify division events, we examine the unmatched instances identified throughout the video sequence since these instances may result from a cell division event or indicate the appearance of a new cell in the field-of-view. To determine whether an unmatched instance is a candidate daughter cell, we perform Eigen decomposition of the spatial coordinates of the instance.

Let the coordinates of the instance be denoted by $X \in \mathcal{R}^{p \times 3}$ with p representing the number of 3D points. The covariance matrix can be expressed as $A = X^T X$, and we perform the singular-value decomposition, $[U, S, V] = svd(A)$. The Eigenvector matrix, $V = [v_1, v_2, v_3]$ with $v_i \in \mathcal{R}^3$ contains three principal components, each with a dimension of 3. We then take a neighborhood around X and project each neighboring cell $Y_i \in \mathcal{R}^{q \times 3}$ onto the 2^{nd} and 3^{rd} principal components. The corresponding projection matrix is expressed as $PM_i = Y_i V_{2,3}$ and a single projection value is computed as $PV_i = norm(mean(PM_i))$. The neighboring cell Y_i with minimum projection value, $\arg\min\{PV_i\}$, is considered as the other candidate daughter cell of X, denoted by $X^{'}$.

Now, to further ensure that instances X and $X^{'}$ in any frame $t+1$ result from the cell division of a parent cell in frame t, we compare the volume of the other candidate daughter cell in current frame, $vol(X_{t+1}^{'})$, against its volume in the preceding time frame, $vol(X_t^{'})$. As cell division typically results in the parent cell dividing into two daughter cells, each with approximately half the volume

of the parent cell, we examine whether the ratio $\frac{vol(X_t^{'})}{vol(X_{t+1}^{'})} \approx 50\%$. If the condition is satisfied, it suggests that X and $X^{'}$ are the daughters of a parent cell from the previous frame. In such cases, we assign a distinct new cell ID to the other daughter cell X' to differentiate it from its parent in the previous frame.

Generate Complete Trajectories

Following the frame-by-frame association and cell division detection, we can compute the complete trajectories of the labeled cell instances in the sequence. This process begins by identifying the unique instance ids in the relabeled sequence. For each unique instance id l, we traverse the sequence to determine the initial and final time points at which the instance appears, represented by t^l_{init} and t^l_{fin}, respectively. Additionally, we can extract the coordinates of the l^{th} instance at each time frame between t^l_{init} and t^l_{fin} and store them in a set of coordinates denoted by $\boldsymbol{C}^l$. Furthermore, we record whether the instance is a parent cell ($P(l) = 0$) or a daughter cell ($P(l) = l'$).

4.4 Experimental Framework

In this section, we provide the description of the dataset, the implementation details of the method, the evaluation metrics, and an account of the competing approaches.

4.4.1 Dataset

We evaluated the proposed cell tracking method on both synthetic and real 3D microscopy videos of bacterial biofilms. The synthetic biofilm sequences were generated using a simulation framework [33] which models biofilm formation following biophysical rules and also represents bacterial cells with realistic curvilinear morphology. In these synthetic videos, starting with one or multiple seed cells, biofilm continues to form as the cells grow and divide over a period of time. We simulated multiple synthetic sequences with a different number of initial clusters where the seed cell are placed at random spatial allocations and orientations. These sequences were generated at a frame interval of 10 seconds. Each synthetic video sequence has a dimension of $450 \times 450 \times 150 \times 40$ in x-y-z-t. The tracking challenge here is to linking cell instances within a very dense environment as well as detecting frequent division events.

For cell tracking in real biofilm sequences, we acquired lattice light-sheet microscopy [87] videos of two kinds of bacteria species, *Escherichia coli* and *Shewanella oneidensis*. The resolution of each frame in the video is approximately 230 nm in x and y, and 370 nm in z, assuming green fluorescent

Algorithm 3 One-to-One Matching between Frames t and $t+1$

Input: Cell ids for all candidate associations, $\boldsymbol{C}_{N\times 2}$; association scores, $\boldsymbol{a}_{N\times 1}$
Output: Association prediction $\boldsymbol{x}_{N\times 1} \in \{0,1\}$
$k_i \leftarrow$ no. of nearest neighbors in $t+1$ for i^{th} instance in t (set to 4)
$\boldsymbol{c}_0 \leftarrow$ unique cell ids from frame t
$\boldsymbol{c}_1 \leftarrow$ unique cell ids from frame $t+1$
$\boldsymbol{x} \leftarrow zeros_{N\times 1}$ ▷ initialize
$conflict \leftarrow 1$ ▷ initialize
$\boldsymbol{D}_0\left[\boldsymbol{c}_0[i]\right] \leftarrow$ -1 $\quad \forall\, i = \{0,1,..,(len(\boldsymbol{c}_0)\text{-}1)\}$ ▷ initialize a dictionary for unique ids of t
$\boldsymbol{D}_1\left[\boldsymbol{c}_1[j]\right] \leftarrow$ -1 $\quad \forall\, j = \{0,1,..,(len(\boldsymbol{c}_1)\text{-}1)\}$ ▷ initialize a dictionary for unique ids of $t+1$
while $conflict > 0$ **do**
    $conflict \leftarrow 0$
    for $i = 0$ to $(len(\boldsymbol{c}_0)\text{-}1)$ **do**
        if $\boldsymbol{D}_0\left[\boldsymbol{c}_0[i]\right] =$ -1 **then**
            $max\_loc \leftarrow \arg\max \boldsymbol{a}_{k_i\times 1}$ ▷ select association with max score among k_i candidate scores
            if $\boldsymbol{D}_1\left[\boldsymbol{C}[max\_loc, 1]\right] =$ -1 **then**
                $\boldsymbol{D}_1\left[\boldsymbol{C}[max\_loc, 1]\right] \leftarrow max\_loc$ ▷ update with new association location
                $\boldsymbol{D}_0\left[\boldsymbol{C}[max\_loc, 0]\right] \leftarrow max\_loc$ ▷ update with new association location
            else
                $conflict \leftarrow 1$
                if $\boldsymbol{a}[max\_loc] > \boldsymbol{a}\left[\boldsymbol{D}_1\left[\boldsymbol{C}[max\_loc, 1]\right]\right]$ **then**
                    $\boldsymbol{a}\left[\boldsymbol{D}_1\left[\boldsymbol{C}[max\_loc, 1]\right]\right] \leftarrow 0$ ▷ indicates no association
                    $\boldsymbol{D}_0\left[\boldsymbol{C}[\boldsymbol{D}_1\left[\boldsymbol{C}[max\_loc, 1]\right]]\right] \leftarrow$ -1 ▷ indicates no association
                    $\boldsymbol{D}_0\left[\boldsymbol{C}[max\_loc, 0]\right] \leftarrow max\_loc$ ▷ update with new association location
                    $\boldsymbol{D}_1\left[\boldsymbol{C}[max\_loc, 1]\right] \leftarrow max\_loc$ ▷ update with new association location
                else
                    $\boldsymbol{a}[max\_loc] \leftarrow 0$ ▷ indicates no association
                end if
            end if
        end if
    end for
end while
$\boldsymbol{x}\left[\boldsymbol{D}_0\left[\boldsymbol{c}_0[i]\right]\right] \leftarrow 1 \quad \forall\, i = \{0,1,..,(len(\boldsymbol{c}_0)\text{-}1)\} \wedge \boldsymbol{D}_0\left[\boldsymbol{c}_0[i]\right] \neq$ -1 ▷ obtain final association prediction

protein (GFP) excitation and emission. The *S. oneidensis* video was captured at 30 seconds frame interval for a total period of 15 minutes, while the *E. coli* sequence was captured at frame interval of 5 minutes over a period of 50 minutes. The *Shewanella* bacteria species has high motility and cell density, hence tracking individual cells over time becomes quite challenging. On the other hand, the *E. coli* data introduces frequent division events where cells divide with orientation change and spatial displacement into the next frame, thus poses significant challenges to detect those events.

4.4.2 Implementation Details

The proposed tracking method has one module that requires training, which is the temporal sequence classification network. The other modules are entirely solved in the online test stage. To train the network, we have used the synthetic biofilm sequences. From a training sequence, we randomly sampled trajectories $\boldsymbol{f}^{tem}_{i,(j_{k_i})}$ between any frame pairs t and $t+1$ with corresponding association labels of correct or incorrect associations ($y = 1$ or 0). We then train the network to minimize a binary cross-entropy loss, $\mathcal{L} = -\sum_{i=1}^{m} y^{(i)} \log \hat{y}^{(i)} - (1 - y^{(i)})(1 - \log \hat{y}^{(i)})$, where $\hat{y}$ is the association probability for i^{th} trajectory with a total of m training trajectories. The association network is implemented using the InceptionTime architecture available in the open-source timeseriesAI (tsai) framework [118].

We performed tracking experiments on six synthetic sequences and two real biofilm sequences. For synthetic sequences, the experiments were performed in a leave-one-out fashion, that is, the temporal sequence classification network was pre-trained on five sequences, while the tracking algorithm was evaluated on the remaining sequence. For the real image sequences of two different biofilm species, the tracking algorithm was executed using a pre-trained association network on the synthetic sequences.

Since the proposed method is a tracking-by-detection approach, prior to performing tracking task, the segmentation was performed on each 3D frame of the video using the proposed segmentation method *DeepSeeded* [34] in chapter 3.

4.4.3 Evaluation Measures

We evaluated the tracking performance using two already established cell tracking performance measures. Both of these measures are full-reference, hence compares the estimated tracks from the tracking algorithm with respect to the reference tracks. One measure is called tracking accuracy or TRA, which is widely adopted by the Cell Tracking Challenge. This metric, based on representing tracks as an acyclic oriented graph [119], calculates the cost associated with transforming a computed graph into the reference one. The cost, referred to as $AOGM$ (Acyclic Oriented Graph Metric), is computed as $AOGM = w_{ED}ED + w_{EA}EA + w_{EC}EC$. Here, ED represents the cost of adding edges (resulting from missing links), EA represents the cost of deleting edges (resulting from redundant links), and EC represents the cost of altering edge semantics (resulting from incorrect division detection). The weights w associated with these cost terms are typically set to 1. In essence,

TRA provides a relative cost compared to the expense of creating the reference graph from scratch, denoted as $AOGM_0$. Mathematically, the TRA measure is expressed as:

$$TRA = 1 - \frac{\min(AOGM, AOGM_0)}{AOGM_0}$$

Also, we separately evaluated the cell division detection accuracy in datasets with frequent division events using a F1 score named *Division-F1* [120] represented as follows,

$$\textit{Division-F1} = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$$

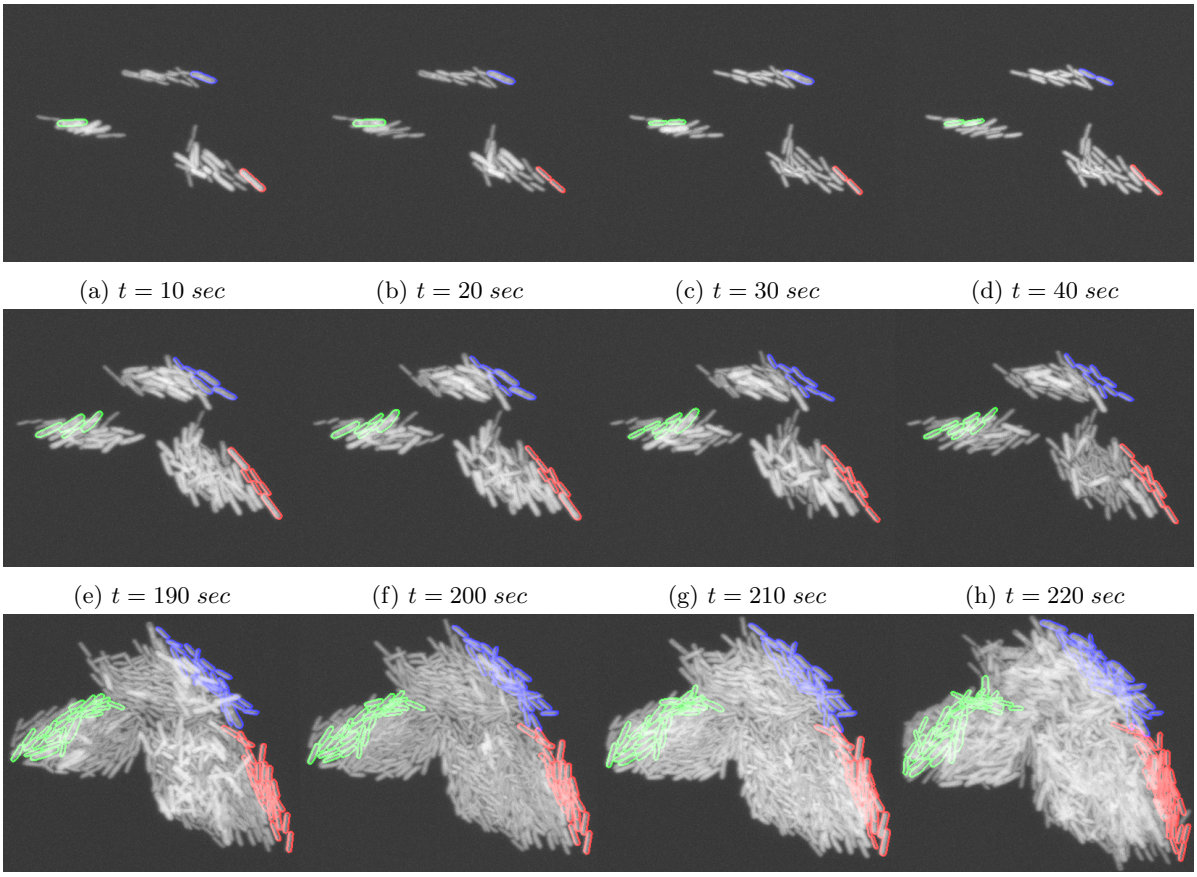
Here, $\textit{precision} = \frac{TP}{TP+FP}$ and $\textit{recall} = \frac{TP}{TP+FN}$, where TP represents the track splitting events detected within time distance t ($t = \pm 1$) of ground truth (GT) events, FP denotes the difference between total detected events and TP events, and FN indicates the difference between total GT events and TP events. Both of these quantitative metrices are computed using a publicly available repository [121].

4.4.4 Competing Approaches

The proposed cell tracking method *DenseTrack* has been evaluated against four competing approaches. We selected three recent methods that have demonstrated state-of-the-art performance in Cell Tracking Challenge datasets and have publicly available implementations. One of these methods is called *Ultrack*, which utilizes ultrametric contours of detected instances for linking them between adjacent frames through a multiple hypotheses-based technique [114]. Another approach, referred to as the *GraphOpt* approach, is a graph-based cell tracking method where segmented objects are assigned to tracks by solving a model-based graph optimization problem [122]. Additionally, we considered a recent deep learning-based cell tracking approach named *GNN*, which constructs cell trajectories using a graph neural network [30]. Finally, we compared the proposed method against a biofilm-specific tracking approach [24] known as the *NearestNbr* tracking method, which performs frame-by-frame association using Euclidean distance of the extracted features.

4.5 Results and Discussion

In this section, we present both qualitative and quantitative tracking results on synthetic and real biofilm sequences. In Fig. 4.2, we visualize the tracking results of a synthetic biofilm sequence

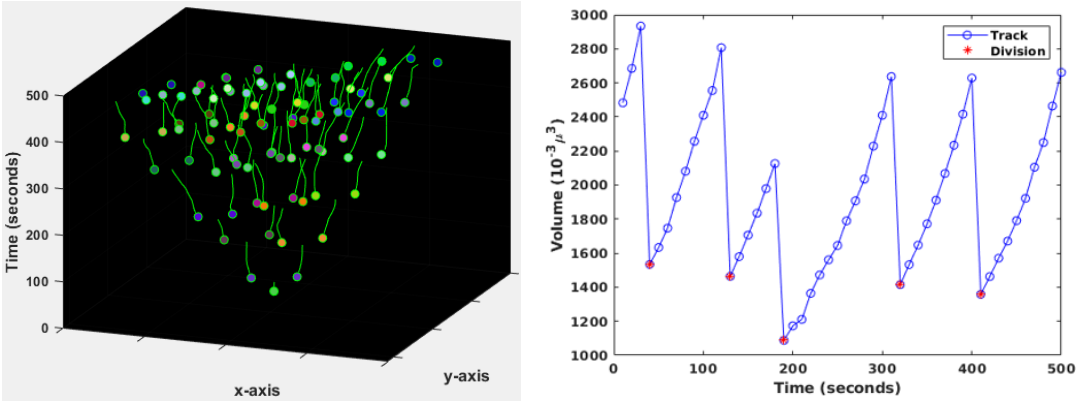


(a) $t = 10\ sec$ (b) $t = 20\ sec$ (c) $t = 30\ sec$ (d) $t = 40\ sec$

(e) $t = 190\ sec$ (f) $t = 200\ sec$ (g) $t = 210\ sec$ (h) $t = 220\ sec$

(i) $t = 420\ sec$ (j) $t = 440\ sec$ (k) $t = 460\ sec$ (l) $t = 500\ sec$

Figure 4.2: Qualitative visualization of tracking cells in a synthetic biofilm sequence with 50 frames captured at 10 seconds frame interval. We demonstrate tracking of three cells at several frames in the sequence. Each 3D frame is displayed as a maximum intensity projection along z axis.



(a) (b)

Figure 4.3: Evidence of effective cell division detection over time through (a) space-time plot and (b) volume-over-time plot, demonstrated for the 'blue' cell in the synthetic sequence in Fig. 4.2

Methods	TRA	Division-F1
DenseTrack	**0.942 ± 0.018**	**0.911 ± 0.022**
Ultrack [114]	0.919 ± 0.021	0.864 ± 0.024
GraphOpt [122]	0.915 ± 0.019	0.886 ± 0.022
NearestNbr [24]	0.840 ± 0.022	0.648± 0.025
GNN [30]	0.818 ± 0.026	0.637 ± 0.033

Table 4.1: Quantitative tracking evaluation on six synthetic biofilm videos

obtained from the proposed *DenseTrack* method. For the clarity of visual observation in a dense environment, we demonstrate the predicted matched instances from three particular cells over the length of the video, displayed in red, blue, and green. From the figure, it is noticeable that the *DenseTrack* method can successfully associate the same instance of a cell over consecutive time points, even in such a crowded neighborhood. Furthermore, it is evident that the cell division events are also accurately detected by the proposed method, which is essential for an effective tracking outcome in this dataset involving frequent division events.

In Fig. 4.3, we present additional support of the effectiveness of the proposed method in cell division detection using a space-time plot and a volume-over-time plot. We demonstrate these two plots for the 'blue' cell of the displayed sequence in Fig. 4.2. The space-time plot depicts the x and y coordinates of the centroid of the 'blue' cell and its matched instances over time. The green circle at the bottom represents the cell's location in the first frame. Pairs of circles in the same color indicate that the tracking algorithm detects two daughter cells in that space and time. The line growing out of the circle signifies the instance's growth until it divides again. The space-time plot derived from the proposed method also indicates that division of the bacterial cell follows a geometric progression, such as 2, 4, 8, 16, and so forth. On the other hand, the volume-over-time plot is generated by considering the volume of only one daughter cell at each division event from the corresponding parent cell. The sawtooth pattern of the plot ensures that the cell divisions are detected properly by the tracking method, as the volume increases when the cell grows and decreases as it splits into daughter cells.

In Table 4.1, we present the quantitative tracking performance for six synthetic biofilm image sequences in our dataset. These videos contain an average of 1400 ground-truth division events. The comparison of tracking methods is based on the overall tracking accuracy (TRA) and the division-specific accuracy metric (*Division-F1*). The results indicate that the proposed *DenseTrack* method outperforms other methods in both performance measures. Additionally, *Ultrack* and *GraphOpt* ex-

hibit reasonable performance in tracking bacterial cells within a dense biofilm environment. However, the nearest neighbor-based technique *NearestNbr*, employing a simplistic Euclidean distance-based frame-by-frame matching, and the graph neural network-based approach *GNN*, predicting one directed graph for the entire sequence, exhibit lower tracking accuracy in both measures.

We also present the visualization of tracked cell instances by the proposed method for a real biofilm sequence of *S. oneidensis* in Fig. 4.4, which was captured at a 30-second frame interval. The matched instances of the same cell over time are displayed in the same color. The figure demonstrates that the proposed method accurately tracks most of the cell instances. Additionally, we visualize the accuracy of predicted trajectories in comparison to corresponding ground-truth trajectories in Fig.4.5. Thirty manually generated ground-truth trajectories are plotted in x-y-z on top of the estimated trajectories by the tracking algorithm. We compare such trajectory plots from the proposed *DenseTrack* method and the best competing method, *Ultrack*, in Fig.4.5a and 4.5b. Observing the figures, it is evident that predicted trajectories from *DenseTrack* exhibit more overlap with the ground-truth, indicating greater accuracy compared to the *Ultrack* method.

In Fig. 4.6, we present a comparative analysis of quantitative tracking performance based on the aforementioned thirty ground-truth trajectories. Since the *S. oneidensis* sequence exhibits very few cell division events (only three ground-truth division events in the thirty trajectories), we have opted not to separately present the *Division-F1* measure and instead focus on reporting the overall tracking score TRA. The figure reveals that, similar to the results obtained from synthetic videos, the proposed *DenseTrack* approach excels in tracking highly motile *Shewanella oneidensis* bacterial cells. While the *Ultrack* method also demonstrates reasonable performance with approximately 90% tracking accuracy, the *GraphOpt* method struggles to track these motile cells, resulting in an approximately 60% TRA score. Additionally, it is observed that the performance of the *GNN* method on this real biofilm sequence further declined, possibly due to its lack of scalability to a data distribution that relatively differs from the distribution of the synthetic training sequences.

We then showcase the qualitative tracking results of our proposed approach on an *E. coli* image sequence in Fig. 4.7, which is captured at a larger frame interval of 5 minutes. In this figure, we observe that even in a lower frame-rate video with very frequent division events, the proposed method performs reasonably well in tracking the cells and their offspring. Furthermore, we provide a qualitative comparison of spatial trajectory plots between the proposed method and the *Ultrack* method with respect to ten manually generated ground-truth trajectories in Fig.4.8. It is observed that, in comparison to the trajectory plot corresponding to the *S. oneidensis* video in Fig.4.5a,

the proposed method exhibits more deviations from the ground-truth in Fig.4.8a. However, these deviations are still fewer than those seen in Fig.4.8b from the *Ultrack* method.

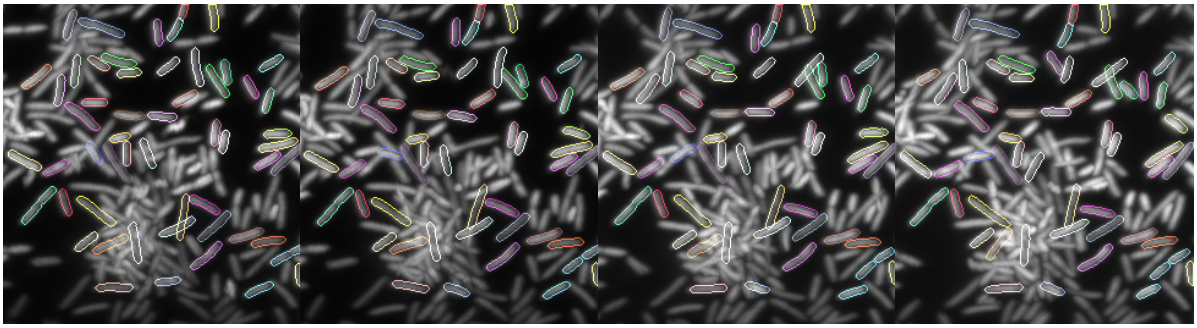
In Table 4.2, we provide the TRA and *Division-F1* scores for the comparative methods based on the ten manually generated trajectories mentioned earlier. These trajectories encompass 54 ground-truth division events. The table highlights that our proposed method excels in tracking bacterial cells even in a lower frame rate video, outperforming the competing methods. Additionally, it is observed that the comparative methods exhibit poorer performance in cell division detection, leading to lower *Division-F1* scores compared to the corresponding scores for synthetic image sequences. This reduced performance in cell division prediction may be attributed to the presence of division events with orientation changes and spatial displacement into the next frame.

Ablation Study

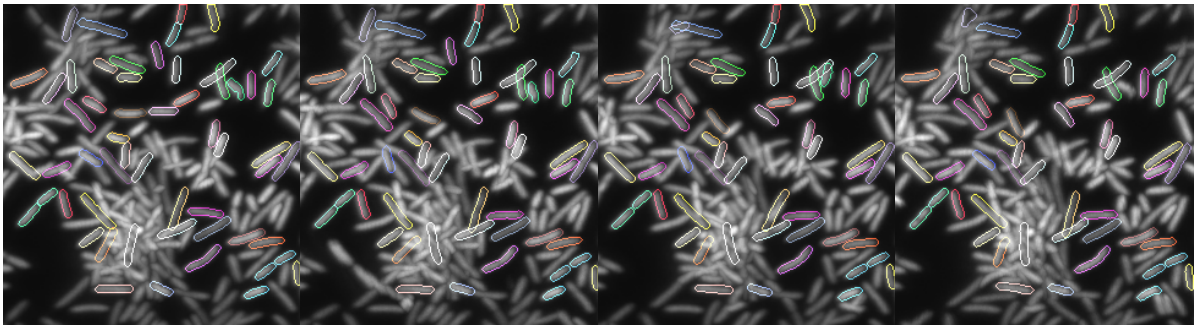
To comprehend the distinct contributions of various components in our proposed method, we conduct ablation studies in this section. In Table 4.3, we present quantitative support for our selection of the *InceptionTime* classifier in the temporal sequence classification task for frame-by-frame association. The classification performance reported here is an average over 10 different frame pairs of a synthetic biofilm video in our dataset.

In the first column, we report the temporal sequence classification accuracy of predicting correct versus wrong associations using different classifiers. It is evident that the proposed choice of the classifier *InceptionTime* outperforms other classifiers in this task. However, this classification result includes one-to-many mapping errors, such as one instance from any frame t being associated with multiple instances from frame $t+1$, or vice versa. Therefore, the results from the second column of the table show that, instead of directly using the classifier output in frame-by-frame matching, employing the classifier's confidence scores in a one-to-one matching optimization, as done in the *DenseTrack* framework, further improves the classification performance, as observed for all listed classifiers.

In Fig. 4.9, we also highlight the importance of leveraging near-temporal history in temporal sequence classification, as implemented in our proposed approach, rather than solely relying on cellular attributes from the present frame and the next frame. The significance is measured in terms of the overall tracking accuracy measure TRA. In Section 4.3, we mentioned the use of a spatio-temporal feature vector, $\boldsymbol{f}^{tem}_{i,(j_{k_i})} = [\boldsymbol{f}^{t-r}_i, ..., \boldsymbol{f}^{t}_i, \boldsymbol{f}^{t+1}_{j_{k_i}}]$, formed by concatenating the feature vector at time t with the feature vectors from the preceding r time frames and the feature vector at time $t+1$. The figure illustrates the effect of using $r = 2$ as in our proposed method versus the effect of using

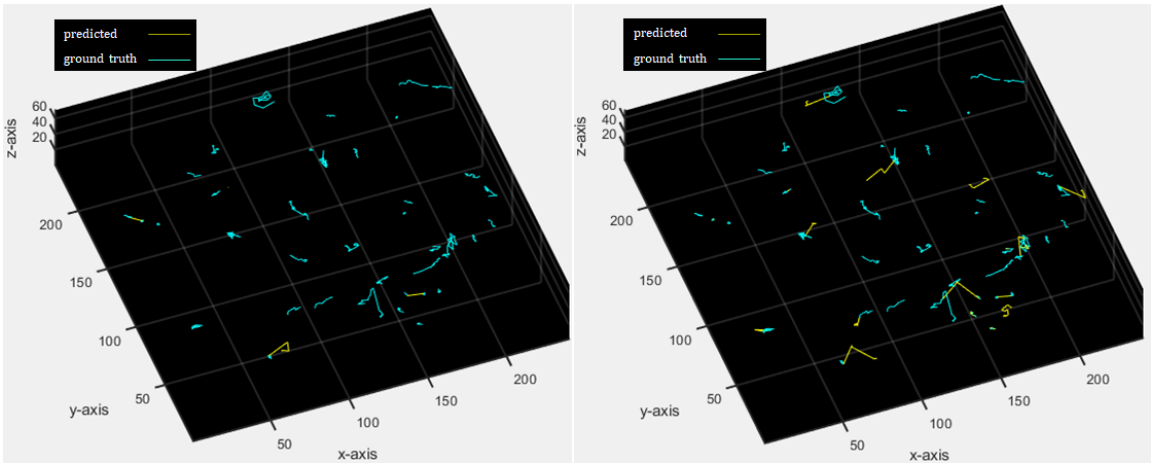


(a) $t = 0.5\ min$ (b) $t = 2.5\ min$ (c) $t = 5\ min$ (d) $t = 7.5\ min$



(e) $t = 10\ min$ (f) $t = 12.5\ min$ (g) $t = 13.5\ min$ (h) $t = 15\ min$

Figure 4.4: Qualitative observation of tracking cells in a *S. oneidensis* real biofilm sequence with 30 frames captured at 30 seconds frame interval. We display 55 cell trajectories over several time points of the video, each with a distinct color.



(a) (b)

Figure 4.5: Visualizing some predicted trajectories of the *S. oneidensis* sequence using the methods (a) *DenseTrack* and (b) *Ultrack* with respect to the corresponding ground-truth trajectories.

$r = 0$. In Fig. 4.9a, we observe such a comparison for a synthetic biofilm video, while in Fig. 4.9b, we observe it for a *S. oneidensis* video. The figures indicate that utilizing near-temporal history ($r = 2$) improves tracking accuracy for both the synthetic sequence and the real biofilm sequence, with a more pronounced improvement observed in the real biofilm example.

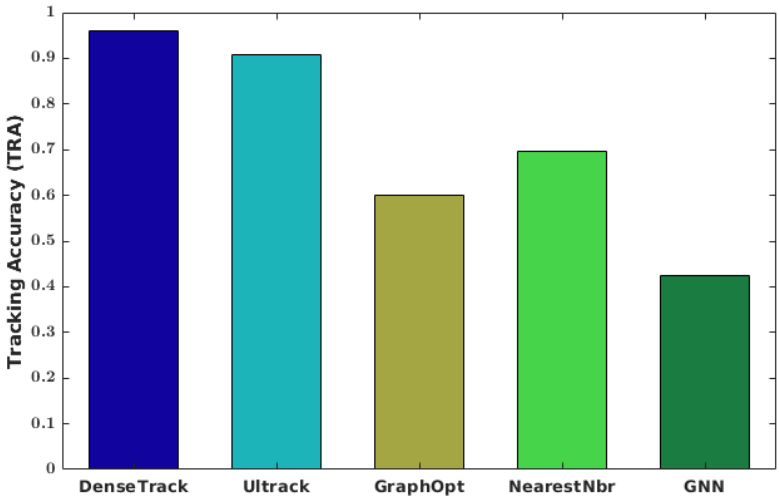


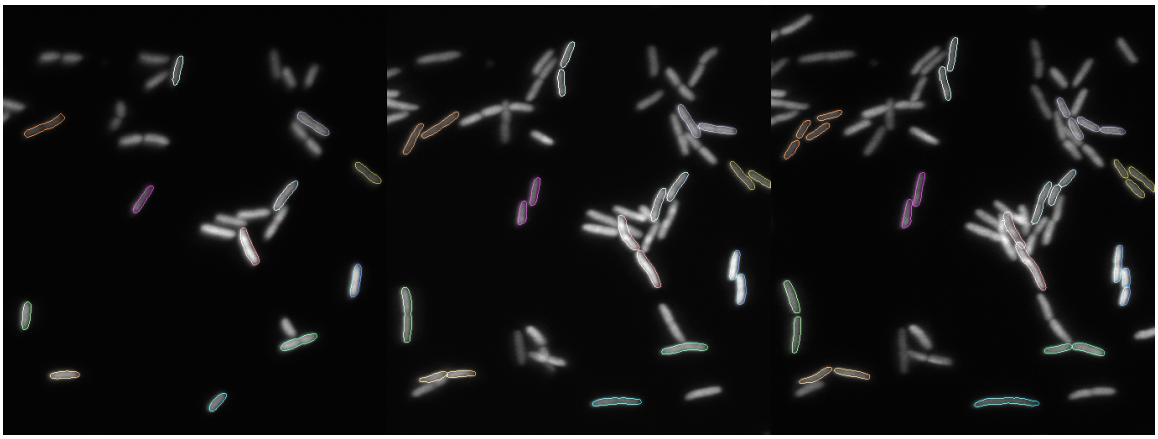
Figure 4.6: Quantitative tracking evaluation on a *S. oneidensis* biofilm video.

Methods	TRA	Division-F1
DenseTrack	**0.904**	**0.877**
Ultrack [114]	0.823	0.652
GraphOpt [122]	0.764	0.410
NearestNbr[24]	0.512	0.391
GNN [30]	0.477	0.297

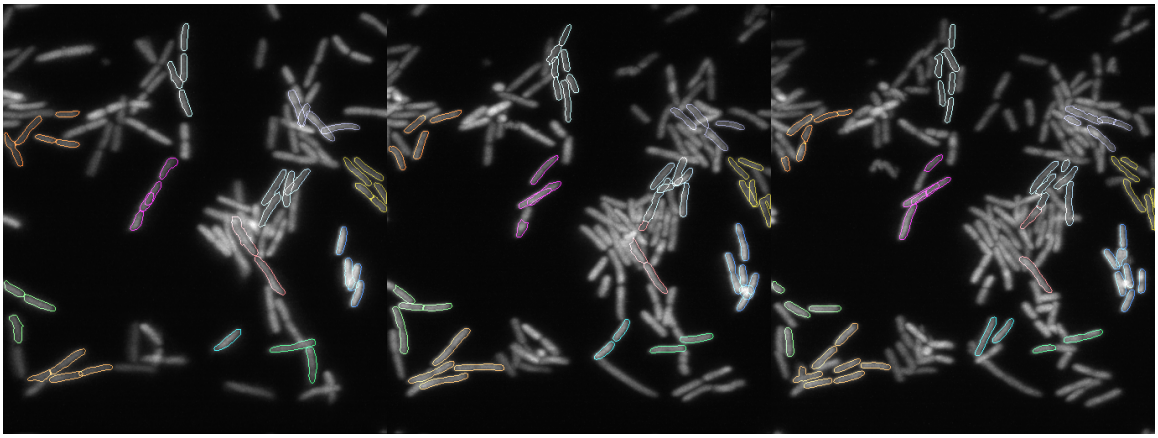
Table 4.2: Quantitative tracking evaluation on an *E. coli* biofilm video

Methods	Classifier	Classifier+OTOM
InceptionTime [117]	**0.964 ± 0.007**	**0.998 ± 0.003**
TST [123]	0.886 ± 0.032	0.940 ± 0.013
LSTM-FCN [124]	0.914 ± 0.024	0.958 ± 0.006
GRU-FCN [125]	0.919 ± 0.018	0.952± 0.007
Res-CNN [126]	0.804 ± 0.031	0.909 ± 0.009

Table 4.3: Binary classification accuracy on temporal sequence classification using various classifiers, and using classifier's confidence scores in an one-to-one matching ($OTOM$) optimization.

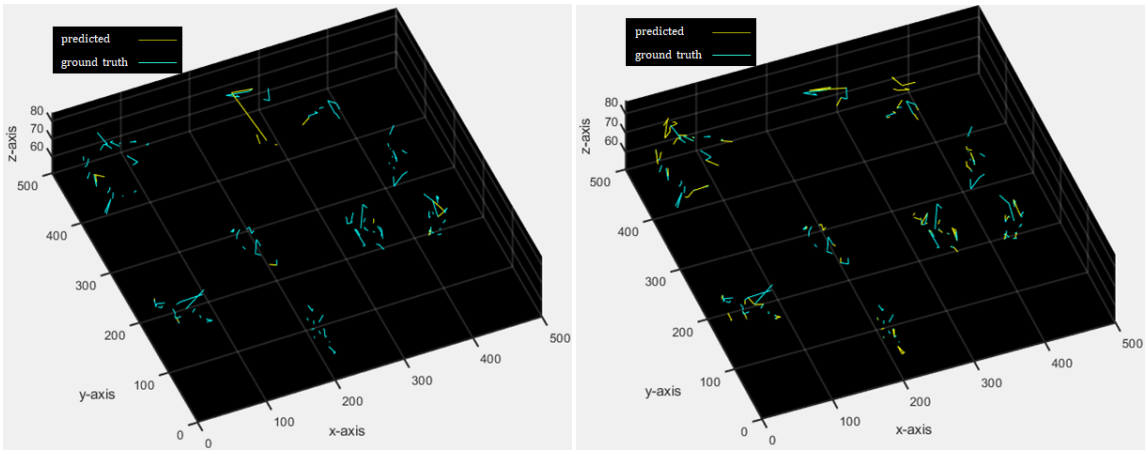


(a) $t = 5\ min$ (b) $t = 20\ min$ (c) $t = 25\ min$



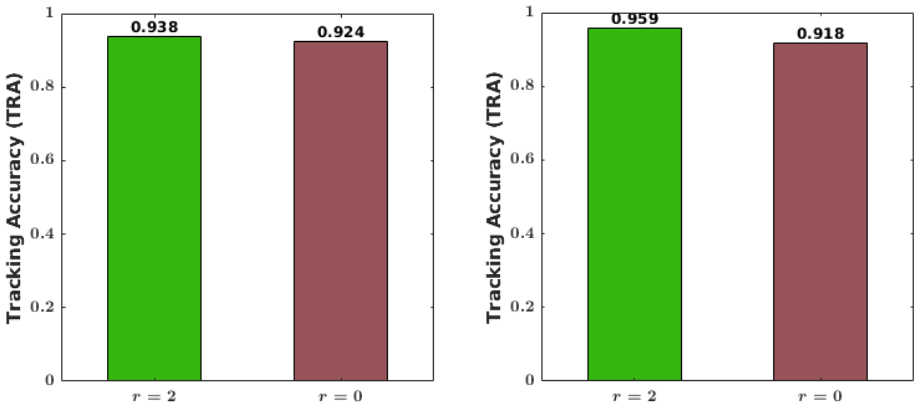
(d) $t = 35\ min$ (e) $t = 45\ min$ (f) $t = 50\ min$

Figure 4.7: Qualitative visualization of tracking cells in a *E. coli* biofilm video with 10 frames captured at 5 minutes frame interval. Trajectories corresponding to 12 cells in the first frame are demonstrated over the length of the video, each with a distinct color.



(a) (b)

Figure 4.8: Visualizing some predicted trajectories of the *E. coli* sequence using the methods (a) *DenseTrack* and (b) *Ultrack* with respect to the corresponding ground-truth trajectories.



(a) (b)

Figure 4.9: Evidence of exploiting near-temporal history ($r = 2$) in tracking performance, on a (a) synthetic biofilm video, and a (b) real biofilm video of *S. oneidensis*.

4.6 Conclusion

This chapter introduced a novel data-driven cell tracking approach to effectively track cell instances and their offspring in dense 3D time-lapse microscopy image sequences. We formulated the cell tracking problem as a frame-by-frame matching task exploiting a deep temporal sequence classifier's confidence scores in a one-to-one optimization framework. Utilizing a data-driven deep-learning-based classifier as opposed to a fixed distance or similarity-based measure yields better association scores for the potential matches between frame pairs. Additionally, an effective one-to-one matching optimization formulation with proper constraints presented in this work ensures superior performance in associating cell instances within a crowded environment. To detect cell division events with high accuracy, we also proposed an Eigen decomposition-based strategy that can identify division events even when daughter instances change orientation and displace spatially during dividing from the parent instance. We demonstrate the effectiveness of the proposed method in tracking bacterial cells from 3D lattice light-sheet image sequences of biofilms. The proposed method achieved better tracking results in terms of both qualitative and quantitative evaluation measures against recent and state-of-the-art cell tracking approaches.

Chapter 5

Discussion and Future Work

Cell segmentation and tracking are essential image processing tasks that facilitate scientists in gaining insights into the biophysics of a cell population. Manual analysis is laborious and prone to subjective errors, especially when dealing with a large number of cell instances, and becomes even more challenging in 3D datasets. Consequently, the development of automatic segmentation and tracking algorithms has been a thriving area of research. While many existing cell segmentation and tracking algorithms exist in the literature, they are not readily applicable to addressing the underlying challenges associated with a particular cell population. One such challenge involves solving segmentation and tracking problems in 3D images and videos of densely populated bacterial biofilms. In this thesis, we have proposed effective algorithms to address such problems. Our main contributions are presented in three distinct chapters.

In Chapter 2, we presented a simulation framework that can generate synthetic biofilm images consisting of bacterial cells with realistic curvilinear morphology. We utilized an elastic shape analysis framework known as square-root normal field (SRNF) to model such realistic bacterial shapes. These simulated curvilinear bacterial shapes were integrated into a biofilm modeling software called CellModeller. We demonstrated that the realistically shaped 3D synthetic biofilm images and videos generated by the proposed simulator are useful for training deep learning-based segmentation and tracking methods. In future, the simulation framework can be further improved by incorporating a realistic motion model of the bacterial cells during the growth and division process of a biofilm.

In Chapter 3, we proposed a novel cell segmentation approach aimed at effectively segmenting touching cell instances in crowded image scenarios, such as within a dense 3D biofilm. Our approach involved designing a hybrid deep-learning model that comprised an image regression network followed

by a voxel-wise image classification network. This model was utilized to estimate cell seeds for a subsequent seeded watershed segmentation. Through our regression network, we successfully generated enhanced maps of cell interior and borders, contributing to superior seed estimation by the voxel-wise classification network. Experimental results were showcased, demonstrating the application of our method in segmenting bacterial cells from both synthetic biofilm images and lattice light-sheet microscopy images of real biofilms. In qualitative and quantitative evaluations, our proposed method exhibited better segmentation results in comparison to state-of-the-art cell segmentation methods.

In the future, several improvements can be incorporated into the proposed segmentation workflow while still retaining the key benefits of our method, including effective Euclidean distance map representation, specialized image quality-oriented loss, and a multi-task scheme comprising enhancement and semantic segmentation tasks. With the success of denoising diffusion models in many image processing tasks, the diffusion model [127] can be integrated into our U-Net-based regression and voxel-wise classification networks. Further, to optimize the memory requirement while training the networks on large 3D biofilm datasets, we can incorporate such memory-efficient architectures [90, 91] instead of traditional U-Net-based CNNs for the regression and semantic segmentation tasks. Additionally, our two loss functions can be jointly learned in a multi-task framework using hypernetworks [94]. Lastly, instead of performing segmentation separately on each frame, a video-based segmentation workflow can be implemented while still leveraging the key features of the proposed segmentation technique.

In Chapter 4, we introduced a novel cell tracking approach designed to effectively track cell instances and their offspring in dense 3D time-lapse microscopy videos. The proposed tracking approach involves an efficient frame-by-frame matching procedure that leverages a deep learning-based temporal sequence classifier. This classifier computes association scores for potential matches in the subsequent frame. The association scores obtained from the network are then utilized in a constrained optimization framework to achieve one-to-one matching between consecutive time frames. Furthermore, for the task of cell division detection, we presented a novel strategy involving the Eigen decomposition of unmatched instances in the subsequent frame. Assuming the unmatched instance as a candidate daughter cell, the neighboring cell instance with the minimum projection along the second and third principal components of the unmatched instance was considered as the second daughter cell. We demonstrated the effectiveness of the proposed method in tracking bacterial cells from 3D microscopy videos of biofilms. Our method achieved superior tracking results in terms of qualitative and quantitative evaluation measures against existing cell tracking approaches.

In the future, improving the proposed tracking method could involve integrating a segmentation error correction stage and a tracklet stitching stage. Additionally, to boost the performance of frame-by-frame matching, a more effective approach could be achieved by merging temporal sequence classification and one-to-one matching optimization into a unified pipeline. Finally, to improve overall tracking accuracy, training the temporal sequence classifier with a broader range of diverse and realistic simulated biofilm sequences, including those with realistic motion features, could be beneficial.

References

[1] Tomas Vicar, Jan Balvan, Josef Jaros, Florian Jug, Radim Kolar, Michal Masarik, and Jaromir Gumulec. Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison. *BMC bioinformatics*, 20(1):1–25, 2019.

[2] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature methods*, 14(12):1141–1152, 2017.

[3] Luanne Hall-Stoodley, J William Costerton, and Paul Stoodley. Bacterial biofilms: from the natural environment to infectious diseases. *Nature Reviews Microbiology*, 2(2):95–108, 2004.

[4] Thomas Bjarnsholt, Maria Alhede, Morten Alhede, Steffen R Eickhardt-Sørensen, Claus Moser, Michael Kühl, Peter Østrup Jensen, and Niels Høiby. The in vivo biofilm. *Trends in microbiology*, 21(9):466–474, 2013.

[5] Michael P Schultz, JA Bendick, ER Holm, and WM Hertel. Economic impact of biofouling on a naval surface ship. *Biofouling*, 27(1):87–98, 2011.

[6] Venkatesh Chaturvedi and Pradeep Verma. Microbial fuel cell: a green approach for the utilization of waste for the generation of bioelectricity. *Bioresources and Bioprocessing*, 3(1):1–14, 2016.

[7] Knut Drescher, Yi Shen, Bonnie L Bassler, and Howard A Stone. Biofilm streamers cause catastrophic disruption of flow with consequences for environmental and medical systems. *Proceedings of the National Academy of Sciences*, 110(11):4345–4350, 2013.

[8] Alice S Prince. Biofilms, antimicrobial resistance, and airway infection. *New England Journal of Medicine*, 347(14):1110–1111, 2002.

[9] Simon P Shen, Hua-an Tseng, Kyle R Hansen, Ruofan Wu, Howard J Gritton, Jennie Si, and Xue Han. Automatic cell segmentation by adaptive thresholding (acsat) for large-scale calcium imaging datasets. *Eneuro*, 5(5), 2018.

[10] Jierong Cheng, Jagath C Rajapakse, et al. Segmentation of clustered nuclei with shape markers and marking function. *IEEE Transactions on Biomedical Engineering*, 56(3):741–748, 2008.

[11] Suvadip Mukherjee and Scott T Acton. Region based segmentation in presence of intensity inhomogeneity using legendre polynomials. *IEEE Signal Processing Letters*, 22(3):298–302, 2014.

[12] Yong He, Hui Gong, Benyi Xiong, Xiaofeng Xu, Anan Li, Tao Jiang, Qingtao Sun, Simin Wang, Qingming Luo, and Shangbin Chen. iCut: an integrative cut algorithm enables accurate segmentation of touching cells. *Scientific reports*, 5(1):1–17, 2015.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[14] Juan C Caicedo, Jonathan Roth, Allen Goodman, Tim Becker, Kyle W Karhohs, Matthieu Broisin, Csaba Molnar, Claire McQuin, Shantanu Singh, Fabian J Theis, et al. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, 95(9):952–965, 2019.

[15] Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in MRI images. In *International MICCAI Brainlesion Workshop*, pages 272–284. Springer, 2022.

[16] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[17] Mingxing Zhang, Ji Zhang, Yibo Wang, Jie Wang, Alecia M Achimovich, Scott T Acton, and Andreas Gahlmann. Non-invasive single-cell morphometry in living bacterial biofilms. *Nature Communications*, 11(1):1–13, 2020.

[18] Jie Wang, Nazia Tabassum, Tanjin T Toma, Yibo Wang, Andreas Gahlmann, and Scott T Acton. 3D GAN image synthesis and dataset quality assessment for bacterial biofilm. *Bioinformatics*, 38(19):4598–4604, 2022.

[19] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] Jiajia Chen and Baocan Zhang. Segmentation of overlapping cervical cells with mask region convolutional neural network. *Computational and Mathematical Methods in Medicine*, 2021, 2021.

[22] Tim Scherr, Katharina Löffler, Moritz Böhland, and Ralf Mikut. Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. *Plos One*, 15(12):e0243219, 2020.

[23] Xieli Li, Yuanyuan Wang, Qisheng Tang, Zhen Fan, and Jinhua Yu. Dual U-Net for the segmentation of overlapping glioma nuclei. *Ieee Access*, 7:84040–84052, 2019.

[24] Ji Zhang, Yibo Wang, Eric D Donarski, Tanjin T Toma, Madeline T Miles, Scott T Acton, and Andreas Gahlmann. BCM3D 2.0: accurate segmentation of single bacterial cells in dense biofilms using computationally generated intermediate image representations. *npj Biofilms and Microbiomes*, 8(1):99, 2022.

[25] M Ali Akber Dewan, M Omair Ahmad, and MNS Swamy. Tracking biological cells in time-lapse microscopy: An adaptive technique combining motion and topological features. *IEEE Transactions on Biomedical Engineering*, 58(6):1637–1647, 2011.

[26] Dirk Padfield, Jens Rittscher, and Badrinath Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical image analysis*, 15(4):650–668, 2011.

[27] Martin Schiegg, Philipp Hanslovsky, Carsten Haubold, Ullrich Koethe, Lars Hufnagel, and Fred A Hamprecht. Graphical model for joint segmentation and tracking of multiple dividing cells. *Bioinformatics*, 31(6):948–956, 2015.

[28] Lee-Ling S Ong, Marcelo H Ang, and H Harry Asada. Tracking of cell population from time lapse and end point confocal microscopy images with multiple hypothesis kalman smoothing filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 71–78. IEEE, 2010.

[29] Nicolas Chenouard, Isabelle Bloch, and Jean-Christophe Olivo-Marin. Multiple hypothesis tracking for cluttered biological image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2736–3750, 2013.

[30] Tal Ben-Haim and Tammy Riklin Raviv. Graph neural network for cell tracking in microscopy videos. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXI*, pages 610–626. Springer, 2022.

[31] Junjie Wang, Xiaohong Su, Lingling Zhao, and Jun Zhang. Deep reinforcement learning for data association in cell tracking. *Frontiers in Bioengineering and Biotechnology*, 8:298, 2020.

[32] Andreas Panteli, Deepak K Gupta, Nathan Bruijn, and Efstratios Gavves. Siamese tracking of cell behaviour patterns. In *Medical Imaging with Deep Learning*, pages 570–587. PMLR, 2020.

[33] Tanjin Taher Toma, Yuexuan Wu, Jie Wang, Anuj Srivastava, Andreas Gahlmann, and Scott T Acton. Realistic-shape bacterial biofilm simulator for deep learning-based 3D single-cell segmentation. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2022.

[34] Tanjin Taher Toma, Yibo Wang, Andreas Gahlmann, and Scott T Acton. Deepseeded: Volumetric segmentation of dense cell populations with a cascade of deep neural networks in bacterial biofilm applications. *Expert Systems with Applications*, 238:122094, 2024.

[35] Tanjin Taher Toma, Yibo Wang, Andreas Gahlmann, and Scott T Acton. Deep temporal sequence classification for automatic cell tracking in dense 3D microscopy videos of bacterial biofilms. *preparing submission to Transactions on Computational Biology and Bioinformatics*, 2024.

[36] Hamid Laga, Qian Xie, Ian H Jermyn, and Anuj Srivastava. Numerical inversion of SRNF maps for elastic shape analysis of genus-zero surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2451–2464, 2017.

[37] Timothy J Rudge, Paul J Steiner, Andrew Phillips, and Jim Haseloff. Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology*, 1(8):345–352, 2012.

[38] Thomas Atta-Fosu, Weihong Guo, Dana Jeter, Claudia M Mizutani, Nathan Stopczynski, and Rui Sousa-Neves. 3D clumped cell segmentation using curvature based seeded watershed. *Journal of Imaging*, 2(4):31, 2016.

[39] Jie Wang, Rituparna Sarkar, Arslan Aziz, Andrea Vaccari, A Gahlmann, and Scott T Acton. Bact-3D: A level set segmentation approach for dense multi-layered 3D bacterial biofilms. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 330–334. IEEE, 2017.

[40] Raimo Hartmann, Hannah Jeckel, Eric Jelli, Praveen K Singh, Sanika Vaidya, Miriam Bayer, Daniel KH Rode, Lucia Vidakovic, Francisco Díaz-Pascual, Jiunn CN Fong, et al. Quantitative image analysis of microbial communities with BiofilmQ. *Nature Microbiology*, 6(2):151–156, 2021.

[41] Raimo Hartmann, Praveen K Singh, Philip Pearce, Rachel Mok, Boya Song, Francisco Díaz-Pascual, Jörn Dunkel, and Knut Drescher. Emergence of three-dimensional order and structure in growing biofilms. *Nature Physics*, 15(3):251–256, 2019.

[42] David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Computational Biology*, 12(11):e1005177, 2016.

[43] Linfeng Yang, Rajarshi P Ghosh, J Matthew Franklin, Simon Chen, Chenyu You, Raja R Narayan, Marc L Melcher, and Jan T Liphardt. NuSet: A deep learning tool for reliably separating and analyzing crowded cells. *PLoS Computational Biology*, 16(9):e1008193, 2020.

[44] Kenneth W Dunn, Chichen Fu, David Joon Ho, Soonam Lee, Shuo Han, Paul Salama, and Edward J Delp. Deepsynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data. *Scientific Reports*, 9(1):1–15, 2019.

[45] Jonathan Naylor, Harold Fellermann, Yuchun Ding, Waleed K Mohammed, Nicholas S Jakubovics, Joy Mukherjee, Catherine A Biggs, Phillip C Wright, and Natalio Krasnogor. Simbiotics: a multiscale integrative platform for 3D modeling of bacterial populations. *ACS Synthetic Biology*, 6(7):1194–1210, 2017.

[46] Tomas Storck, Cristian Picioreanu, Bernardino Virdis, and Damien J Batstone. Variable cell morphology approach for individual-based modeling of microbial communities. *Biophysical Journal*, 106(9):2037–2048, 2014.

[47] Ch Brechbühler, Guido Gerig, and Olaf Kübler. Parametrization of closed surfaces for 3-D shape description. *Computer Vision and Image Understanding*, 61(2):154–170, 1995.

[48] Ian H Jermyn, Sebastian Kurtek, Hamid Laga, and Anuj Srivastava. Elastic shape analysis of three-dimensional objects. *Synthesis Lectures on Computer Vision*, 12(1):1–185, 2017.

[49] Ian H Jermyn, Sebastian Kurtek, Eric Klassen, and Anuj Srivastava. Elastic shape matching of parameterized surfaces using square root normal fields. In *European Conference on Computer Vision*, pages 804–817. Springer, 2012.

[50] P.J. Besl and Neil D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[51] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 424–432. Springer, 2016.

[52] Wentao Zhu, Yufang Huang, Liang Zeng, Xuming Chen, Yong Liu, Zhen Qian, Nan Du, Wei Fan, and Xiaohui Xie. AnatomyNet: deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Medical Physics*, 46(2):576–589, 2019.

[53] Project MONAI: Medical Open Network for AI. https://docs.monai.io/en/latest/index.html. Accessed: 2021-09-30.

[54] J. Wang, M. Zhang, J. Zhang, Y. Wang, A. Gahlmann, and S. T. Acton. Graph-theoretic post-processing of segmentation with application to dense biofilms. *IEEE Transaction on Image Processing*, 2021.

[55] Anuradha Kar, Manuel Petit, Yassin Refahi, Guillaume Cerutt, Christophe Godin, and Jan Traas. Benchmarking of deep learning algorithms for 3d instance segmentation of confocal image datasets. *bioRxiv*, pages 2021–06, 2022.

[56] Hady Ahmady Phoulady, Dmitry B Goldgof, Lawrence O Hall, and Peter R Mouton. Nucleus segmentation in histology images with hierarchical multilevel thresholding. In *Medical Imaging 2016: Digital Pathology*, volume 9791, pages 280–285. SPIE, 2016.

[57] Serge Beucher and Fernand Meyer. The morphological approach to segmentation: the watershed transformation. In *Mathematical morphology in image processing*, pages 433–481. CRC Press, 2018.

[58] Scott T Acton and Nilanjan Ray. Biomedical image analysis: Segmentation. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 4(1):1–108, 2009.

[59] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

[60] Jie Wang, Mingxing Zhang, Ji Zhang, Yibo Wang, Andreas Gahlmann, and Scott T Acton. Graph-theoretic post-processing of segmentation with application to dense biofilms. *IEEE Transactions on Image Processing*, 30:8580–8594, 2021.

[61] Pierre Soille et al. *Morphological image analysis: principles and applications*, volume 2. Springer, 1999.

[62] Fernand Meyer and Serge Beucher. Morphological segmentation. *Journal of visual communication and image representation*, 1(1):21–46, 1990.

[63] Can Fahrettin Koyuncu, Ece Akhan, Tulin Ersahin, Rengul Cetin-Atalay, and Cigdem Gunduz-Demir. Iterative h-minima-based marker-controlled watershed for cell nucleus segmentation. *Cytometry Part A*, 89(4):338–349, 2016.

[64] Chanho Jung and Changick Kim. Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization. *IEEE transactions on biomedical engineering*, 57(10):2600–2604, 2010.

[65] Jakub Smołka. Multilevel near optimal thresholding applied to watershed grouping. *Annales Universitatis Mariae Curie-Sklodowska, sectio AI–Informatica*, 5(1):191, 2006.

[66] Lu Xiong, Dongbo Zhang, Kangshun Li, and Lixia Zhang. The extraction algorithm of color disease spot image based on otsu and watershed. *Soft computing*, 24(10):7253–7263, 2020.

[67] Nancy Salem, Noorhan M Sobhy, and Mohamed El Dosoky. A comparative study of white blood cells segmentation using otsu threshold and watershed transformation. *Journal of Biomedical Engineering and Medical Imaging*, 3(3):15, 2016.

[68] Tim Prangemeier, Christian Wildner, André O Françani, Christoph Reich, and Heinz Koeppl. Yeast cell segmentation in microstructured environments with deep learning. *Biosystems*, 211:104557, 2022.

[69] Adrian Wolny, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, Amaya Vilches Barro, Marion Louveaux, Christian Wenzl, Sören Strauss, David Wilson-Sánchez, Rena Lymbouridou, et al. Accurate and versatile 3d segmentation of plant tissues at cellular resolution. *Elife*, 9:e57613, 2020.

[70] Dennis Eschweiler, Thiago V Spina, Rohan C Choudhury, Elliot Meyerowitz, Alexandre Cunha, and Johannes Stegmaier. CNN-based preprocessing to optimize watershed-based cell segmentation in 3D confocal microscopy images. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 223–227. IEEE, 2019.

[71] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3D medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022.

[72] Zhuo Zhao, Lin Yang, Hao Zheng, Ian H Guldner, Siyuan Zhang, and Danny Z Chen. Deep learning based instance segmentation in 3d biomedical images using weak annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 352–360. Springer, 2018.

[73] Dilanga Abeyrathna, Shailabh Rauniyar, Rajesh K Sani, and Pei-Chi Huang. A morphological post-processing approach for overlapped segmentation of bacterial cell images. *Machine Learning and Knowledge Extraction*, 4(4):1024–1041, 2022.

[74] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 265–273. Springer, 2018.

[75] Talha Ilyas, Zubaer Ibna Mannan, Abbas Khan, Sami Azam, Hyongsuk Kim, and Friso De Boer. Tsfd-net: Tissue specific feature distillation network for nuclei segmentation and classification. *Neural Networks*, 151:1–15, 2022.

[76] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[77] Tim Prangemeier, Christoph Reich, and Heinz Koeppl. Attention-based transformers for instance segmentation of cells in microstructures. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 700–707. IEEE, 2020.

[78] Xiaodan Wei, Qinghao Liu, Min Liu, Yaonan Wang, and Erik Meijering. 3D soma detection in large-scale whole brain images via a two-stage neural network. *IEEE Transactions on Medical Imaging*, 42(1):148–157, 2022.

[79] Bo Yang, Min Liu, Yaonan Wang, Kang Zhang, and Erik Meijering. Structure-guided segmentation for 3D neuron reconstruction. *IEEE transactions on medical imaging*, 41(4):903–914, 2021.

[80] Yi Jiang, Weixun Chen, Min Liu, Yaonan Wang, and Erik Meijering. 3D neuron microscopy image segmentation via the ray-shooting model and a dc-blstm network. *IEEE Transactions on Medical Imaging*, 40(1):26–37, 2020.

[81] Qiufu Li and Linlin Shen. 3D neuron reconstruction in tangled neuronal image with deep networks. *IEEE transactions on medical imaging*, 39(2):425–435, 2019.

[82] Qiufu Li, Yu Zhang, Hanbang Liang, Hui Gong, Liang Jiang, Qiong Liu, and Linlin Shen. Deep learning based neuronal soma detection and counting for alzheimer's disease analysis. *Computer Methods and Programs in Biomedicine*, 203:106023, 2021.

[83] Weikang Wang, David A Taft, Yi-Jiun Chen, Jingyu Zhang, Callen T Wallace, Min Xu, Simon C Watkins, and Jianhua Xing. Learn to segment single cells with deep distance estimator and deep cell detector. *Computers in biology and medicine*, 108:133–141, 2019.

[84] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[85] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[86] M Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, et al. MONAI: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*, 2022.

[87] Mingxing Zhang, Ji Zhang, Jie Wang, Alecia M Achimovich, Arslan A Aziz, Jacqueline Corbitt, Scott T Acton, and Andreas Gahlmann. 3D imaging of single cells in bacterial biofilms using lattice light-sheet microscopy. *Biophysical Journal*, 116(3):25a, 2019.

[88] Ping-Sung Liao, Tse-Sheng Chen, Pau-Choo Chung, et al. A fast algorithm for multilevel thresholding. *J. Inf. Sci. Eng.*, 17(5):713–727, 2001.

[89] Adrian Kucharski and Anna Fabijańska. CNN-watershed: A watershed transform with predicted markers for corneal endothelium image segmentation. *Biomedical Signal Processing and Control*, 68:102805, 2021.

[90] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[91] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[92] Christoph Reich, Tim Prangemeier, Özdemir Cetin, and Heinz Koeppl. OSS-Net: memory efficient high resolution semantic segmentation of 3D medical data. *arXiv preprint arXiv:2110.10640*, 2021.

[93] Robin Brügger, Christian F Baumgartner, and Ender Konukoglu. A partially reversible u-net for memory-efficient volumetric image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*, pages 429–437. Springer, 2019.

[94] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*, 2021.

[95] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak MW Balak, et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014.

[96] Christophe Zimmer, Elisabeth Labruyere, Vannary Meas-Yedid, Nancy Guillén, and J-C Olivo-Marin. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing. *IEEE transactions on medical imaging*, 21(10):1212–1221, 2002.

[97] Nilanjan Ray, Scott T Acton, and Klaus Ley. Tracking leukocytes in vivo with shape and size constrained active contours. *IEEE transactions on medical imaging*, 21(10):1222–1235, 2002.

[98] Kang Li, Eric D Miller, Mei Chen, Takeo Kanade, Lee E Weiss, and Phil G Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical image analysis*, 12(5):546–566, 2008.

[99] Oleh Dzyubachyk, Wiggert A Van Cappellen, Jeroen Essers, Wiro J Niessen, and Erik Meijering. Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. *IEEE transactions on medical imaging*, 29(3):852–867, 2010.

[100] Nezamoddin N Kachouie, Paul Fieguth, John Ramunas, Eric Jervis, et al. Probabilistic model-based cell tracking. *International Journal of Biomedical Imaging*, 2006, 2006.

[101] Daniel H Rapoport, Tim Becker, Amir Madany Mamlouk, Simone Schicktanz, and Charli Kruse. A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters. *PloS one*, 6(11):e27315, 2011.

[102] Markus Rempfler, Valentin Stierle, Konstantin Ditzel, Sanjeev Kumar, Philipp Paulitschke, Bjoern Andres, and Bjoern H Menze. Tracing cell lineages in videos of lens-free microscopy. *Medical image analysis*, 48:147–161, 2018.

[103] Klas EG Magnusson, Joakim Jaldén, Penney M Gilbert, and Helen M Blau. Global linking of cell tracks using the viterbi algorithm. *IEEE transactions on medical imaging*, 34(4):911–929, 2014.

[104] Ryoma Bise, Zhaozheng Yin, and Takeo Kanade. Reliable cell tracking by global data association. In *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pages 1004–1010. IEEE, 2011.

[105] Fatima Boukari and Sokratis Makrogiannis. Automated cell tracking using motion prediction-based matching and event handling. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(3):959–971, 2018.

[106] Fuhai Li, Xiaobo Zhou, Jinwen Ma, and Stephen TC Wong. Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis. *IEEE transactions on medical imaging*, 29(1):96–105, 2009.

[107] Arunachalam Narayanaswamy, Amine Merouane, Antonio Peixoto, Ena Ladi, Paul Herzmark, Ulrich Von Andrian, Ellen Robey, and Badrinath Roysam. Multi-temporal globally-optimal dense 3-d cell segmentation and tracking from multi-photon time-lapse movies of live tissue microenvironments. In *Spatio-temporal Image Analysis for Longitudinal and Time-Series Image Data: Second International Workshop, STIA 2012, Held in Conjunction with MICCAI 2012, Nice, France, October 1, 2012. Proceedings 2*, pages 147–162. Springer, 2012.

[108] Bernhard X Kausler, Martin Schiegg, Bjoern Andres, Martin Lindner, Ullrich Koethe, Heike Leitte, Jochen Wittbrodt, Lars Hufnagel, and Fred A Hamprecht. A discrete chain graph model for 3d+ t cell tracking with high misdetection robustness. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12*, pages 144–157. Springer, 2012.

[109] Min Liu, Yalan Liu, Weili Qian, and Yaonan Wang. Deepseed local graph matching for densely packed cells tracking. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(3):1060–1069, 2019.

[110] William J Godinez and Karl Rohr. Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. *IEEE transactions on medical imaging*, 34(2):415–432, 2014.

[111] Seyed Hamid Rezatofighi, Stephen Gould, Richard Hartley, Katarina Mele, and William E Hughes. Application of the imm-jpda filter to multiple target tracking in total internal reflection fluorescence microscopy images. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012: 15th International Conference, Nice, France, October 1-5, 2012, Proceedings, Part I 15*, pages 357–364. Springer, 2012.

[112] Stefano Coraluppi and Craig Carthel. Multi-stage multiple-hypothesis tracking. *J. Adv. Inf. Fusion*, 6(1):57–67, 2011.

[113] Liang Liang, Hongying Shen, Pietro De Camilli, and James S Duncan. A novel multiple hypothesis based particle tracking method for clathrin mediated endocytosis analysis using fluorescence microscopy. *IEEE transactions on image processing*, 23(4):1844–1857, 2014.

[114] Jordão Bragantini, Merlin Lange, and Loïc Royer. Large-scale multi-hypotheses cell tracking using ultrametric contours maps. *arXiv preprint arXiv:2308.04526*, 2023.

[115] Min Liu, Yue He, Yangliu Wei, and Peng Xiang. Plant cell tracking using kalman filter based local graph matching. *Image and Vision Computing*, 60:154–161, 2017.

[116] Junya Hayashida and Ryoma Bise. Cell tracking with deep learning for cell detection and motion estimation in low-frame-rate. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*, pages 397–405. Springer, 2019.

[117] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[118] Ignacio Oguiza. tsai - a state-of-the-art deep learning library for time series and sequential data. Github, 2023.

[119] Pavel Matula, Martin Maška, Dmitry V Sorokin, Petr Matula, Carlos Ortiz-de Solórzano, and Michal Kozubek. Cell tracking accuracy measurement based on comparison of acyclic oriented graphs. *PloS one*, 10(12):e0144959, 2015.

[120] Kristina Ulicna, Giulia Vallardi, Guillaume Charras, and Alan R Lowe. Automated deep lineage tree analysis using a bayesian single cell tracking approach. *Frontiers in Computer Science*, 3:734559, 2021.

[121] Janelia-Trackathon-2023. Utilities for computing common accuracy metrics on cell tracking challenge solutions with ground truth. https://github.com/Janelia-Trackathon-2023/traccuracy, 2023.

[122] Katharina Löffler, Tim Scherr, and Ralf Mikut. A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction. *PloS one*, 16(9):e0249257, 2021.

[123] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[124] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

[125] Nelly Elsayed, Anthony S Maida, and Magdy Bayoumi. Deep gated recurrent and convolutional network hybrid model for univariate time series classification. *arXiv preprint arXiv:1812.07683*, 2018.

[126] Xiaowu Zou, Zidong Wang, Qi Li, and Weiguo Sheng. Integration of residual network and convolutional neural network along with various activation functions and global pooling for time series classification. *Neurocomputing*, 367:39–45, 2019.

[127] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.