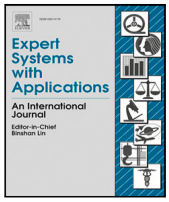
Expert Systems With Applications 238 (2024) 122094

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



DeepSeeded: Volumetric segmentation of dense cell populations with a cascade of deep neural networks in bacterial biofilm applications

Tanjin Taher Toma [a,*], Yibo Wang [b], Andreas Gahlmann [b,c], Scott T. Acton [a]

[a] Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, 22904, VA, USA
[b] Department of Chemistry, University of Virginia, Charlottesville, 22904, VA, USA
[c] Department of Molecular Physiology and Biological Physics, University of Virginia, Charlottesville, 22903, VA, USA

ARTICLE INFO

Keywords:
Dense cell segmentation
Deep learning
Seeded watershed
Multi-scale structural similarity loss
Bacterial biofilms

ABSTRACT

Accurate and automatic segmentation of individual cell instances in microscopy images is a vital step for quantifying the cellular attributes, which can subsequently lead to new discoveries in biomedical research. In recent years, data-driven deep learning techniques have shown promising results in this task. Despite the success of these techniques, many fail to accurately segment cells in microscopy images with high cell density and low signal-to-noise ratio. In this paper, we propose a novel 3D cell segmentation approach DeepSeeded, a cascaded deep learning architecture that estimates seeds for a classical seeded watershed segmentation. The cascaded architecture enhances the cell interior and border information using Euclidean distance transforms and detects the cell seeds by performing voxel-wise classification. The data-driven seed estimation process proposed here allows segmenting touching cell instances from a dense, intensity-inhomogeneous microscopy image volume. We demonstrate the performance of the proposed method in segmenting 3D microscopy images of a particularly dense cell population called bacterial biofilms. Experimental results on synthetic and two real biofilm datasets suggest that the proposed method leads to superior segmentation results when compared to state-of-the-art deep learning methods and a classical method.

1. Introduction

Cell segmentation from microscopy is an essential image processing task 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 and thus aid the drug development process (Kar et al., 2022; Vicar et al., 2019). 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. These conditions are exacerbated in 3D imaging.

There exist many classical approaches for cell segmentation, including thresholding methods followed by pixel-grouping via connected components (Phoulady, Goldgof, Hall, & Mouton, 2016; Shen et al., 2018), morphological methods based on the watershed transform (Atta-Fosu et al., 2016; Beucher & Meyer, 2018; Cheng et al., 2008), geometric active contour models (Acton & Ray, 2009; Mukherjee & Acton, 2014), and methods using graph cuts (Boykov & Funka-Lea, 2006; He et al., 2015). 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 (Wang, Zhang, Zhang, Wang, Gahlmann, & Acton, 2021). 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

* Corresponding author.
E-mail addresses: tt4ua@virginia.edu (T.T. Toma), yw9et@virginia.edu (Y. Wang), ag5vu@virginia.edu (A. Gahlmann), acton@virginia.edu (S.T. Acton).

https://doi.org/10.1016/j.eswa.2023.122094
Received 17 February 2023; Received in revised form 8 October 2023; Accepted 8 October 2023
Available online 14 October 2023

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 (Soille et al., 1999). 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 (Meyer & Beucher, 1990). Various approaches have been attempted for extracting these seeds, such as the h-minima (or maxima) based techniques (Jung & Kim, 2010; Koyuncu, Akhan, Ersahin, Cetin-Atalay, & Gunduz-Demir, 2016) and multilevel thresholding (Salem, Sobhy, & El Dosoky, 2016; Smołka, 2006; Xiong, Zhang, Li, & Zhang, 2020). Seed estimation following these approaches requires tuning specific thresholds or hyperparameters 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 first to perform pixel-wise segmentation (also known as semantic segmentation) using a deep network, e.g., a U-Net (Caicedo et al., 2019; Çiçek, Abdulkadir, Lienkamp, Brox, & Ronneberger, 2016; Prangemeier, Wildner, Françani, Reich, & Koeppl, 2022; Ronneberger, Fischer, & Brox, 2015), and later group pixels into isolated cell instances using classical algorithms, such as watershed or graph partitioning techniques (Eschweiler et al., 2019; Wang et al., 2022; Wolny, Cerrone, Vijayan, Tofanelli, Barro, Louveaux, Wenzl, Strauss, Wilson-Sánchez, Lymbouridou, et al., 2020; Zhang et al., 2020). 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 (Chen et al., 2021; Hatamizadeh, Nath, et al., 2022; Hatamizadeh, Tang, et al., 2022). Another notable approach, known as Cellpose (Stringer, Wang, Michaelos, & Pachitariu, 2021), 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 (Chen & Zhang, 2021; He, Gkioxari, Dollár, & Girshick, 2017; Prangemeier et al., 2022; Zhao et al., 2018) are widely used instance-wise segmentation approach. The original Mask R-CNN method (He et al., 2017) 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 (Abeyrathna, Rauniyar, Sani, & Huang, 2022; Ilyas et al., 2022; Schmidt, Weigert, Broaddus, & Myers, 2018).

More recently, transformer-based end-to-end instance-wise detection and segmentation methods have been proposed (Carion et al., 2020; Prangemeier, Reich, & Koeppl, 2020). 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 (Wei, Liu, Liu, Wang, & Meijering, 2022; Yang, Liu, Wang, Zhang, & Meijering, 2021). Other soma segmentation approaches include a ray-shooting model combined with Long Short-Term Memory (LSTM)-based network (Jiang, Chen, Liu, Wang, & Meijering, 2020), and 3D U-Net-based approaches (Li & Shen, 2019; Li et al., 2021).

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 (Li, Wang, Tang, Fan, & Yu, 2019; Wang et al., 2019). 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, Löffler, Böhland, and Mikut (2020) 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. (2022) 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 (Scherr et al., 2020; Zhang et al., 2022), 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. A comprehensive description of the contributions of the proposed method is provided in Section 1.1.

We demonstrate the performance of the proposed method in the segmentation of bacteria cells from 3D microscopy images of densely packed biofilms. Bacterial biofilms are complex biological systems that play critical roles in infectious diseases, as well as in many industrial and ecological processes (Bjarnsholt et al., 2013; Chaturvedi & Verma, 2016; Drescher, Shen, Bassler, & Stone, 2013; Hall-Stoodley, Costerton, & Stoodley, 2004; Prince, 2002; Schultz, Bendick, Holm, & Hertel, 2011). 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.

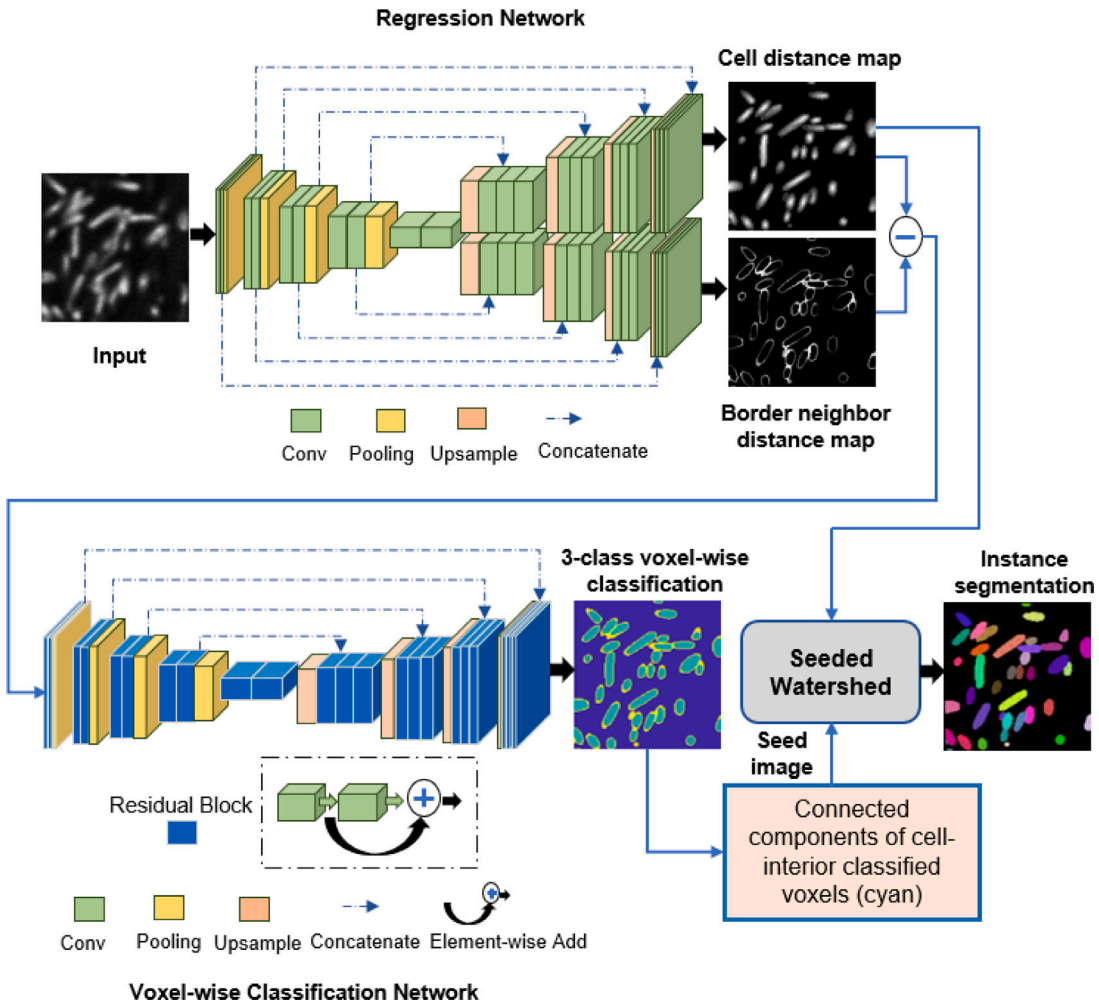


Fig. 1. Overview of the *DeepSeeded* segmentation workflow. The input and output images demonstrated in this artwork correspond to a 2D slice of a 3D biofilm stack.

1.1. 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 hyper-parameters 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 paper is organized as follows: Section 2 presents the theory of the proposed approach. Section 3 includes details of the experimental setup and dataset, evaluation metrics, and comparative methods. Experimental results are presented and discussed in Section 4. Finally, Section 5 offers concluding remarks. A number of symbols used in the paper are listed in Table 1.

Table 1
Description of symbols.

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

2. Theory

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. 1.

2.1. 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 (Çiçek et al., 2016) to estimate these two maps.

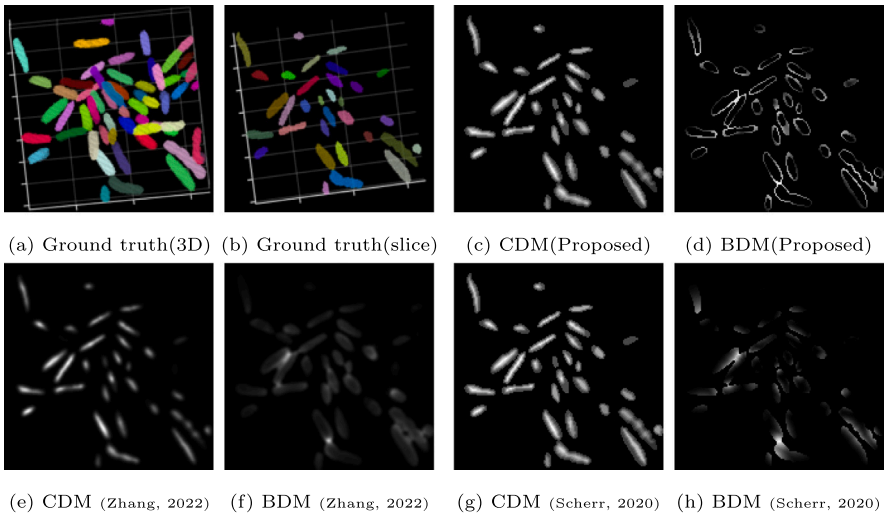


Fig. 2. Qualitative comparison of cell distance map (CDM) and border distance map (BDM) between the proposed method and the competing methods. The maps in Figs. 2(c)–2(h) correspond to the 2D slice in Fig. 2(b).

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 $l = \{0, 1, \ldots, L\}$ 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 Scherr et al. (2020), Zhang et al. (2022). In Scherr et al. (2020), a neighbor distance map is computed pixel-wise for each cell from a ground-truth instance-labeled image. The approach in Zhang et al. (2022) 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. The weight matrix is calculated by subtracting the cell distance map from the binary mask. 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. Such a representation can mislead the network into confusing the interior voxels with the border voxels. In this paper, 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. 2, we qualitatively compare the ground-truth distance maps from the proposed method with those in Scherr et al. (2020), Zhang et al. (2022). The maps are computed in 3D from a ground truth instance-labeled image of an *E.coli*-biofilm, shown in Fig. 2(a). We demonstrate the maps for a particular 2D slice (Fig. 2(b)) of the image volume. From the figure, it is evident that the border distance map of the proposed method captures cell border information more effectively compared to the corresponding maps from Scherr et al. (2020), Zhang et al. (2022). Also, it is seen that the proposed cell distance map highlights the cell interior as effectively as in Scherr et al. (2020).

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,

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

In Eq. (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 (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 nth voxel.

$$C_{\text{SL1}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{N}\sum_{n=1}^{N} \text{SL1}(n) \tag{2}$$

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

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,

$$C_{\text{MS-SSIM}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{N}\sum_{n=1}^{N}[1 - \text{MS-SSIM}(n)] \tag{3}$$

The computation of MS-SSIM involves computing the SSIM metric at multiple scales/resolutions (Wang, Simoncelli, & Bovik, 2003). The SSIM for each pixel/voxel n is defined as follows,

$$\begin{aligned} \text{SSIM}(n) &= \frac{2\mu_{\boldsymbol{x}}\mu_{\tilde{\boldsymbol{x}}} + C_1}{\mu_{\boldsymbol{x}}^2 + \mu_{\tilde{\boldsymbol{x}}}^2 + C_1} \cdot \frac{2\sigma_{\boldsymbol{x}\tilde{\boldsymbol{x}}} + C_2}{\sigma_{\boldsymbol{x}}^2 + \sigma_{\tilde{\boldsymbol{x}}}^2 + C_2} \\ &= l(n) \cdot cs(n) \end{aligned}$$

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,

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

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

$$\text{MS-SSIM}(n) = l_M^{\beta}(n)\, cs_M^{\beta}(n) \prod_{j=1}^{M-1} cs_j^{\delta_j}(n) \tag{4}$$

The hyperparameters β and $\{\delta_j\}$ in Eq. (4) and α in Eq. (1) are set empirically during the offline training stage of the network and have been chosen on a validation set of images.

2.2. 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 (Hatamizadeh, Nath, et al., 2022) and focal loss (Lin, Goyal, Girshick, He, & Dollár, 2017) as follows,

$$Loss, L_C = \frac{1}{T} \sum_{t=1}^{T} [C_{\text{Dice}}(\boldsymbol{v}_t, \tilde{\boldsymbol{v}}_t) + C_{\text{focal}}(\boldsymbol{v}_t, \tilde{\boldsymbol{v}}_t)] \tag{5}$$

where,

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

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

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

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. A.6.

2.3. 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 $\tilde{\mathfrak{s}}$. We then apply the watershed function on the cell interior-enhanced image $\tilde{\boldsymbol{x}}^c$. Starting with the seed locations in image $\tilde{\mathfrak{s}}$, 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.

Algorithm 1 Compute Cell Distance Map

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

3. 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.

Algorithm 2 Compute Border Neighbor Distance Map

```
Input: Instance labeled image x^l
Output: Border neighbor distance map x^b
x^b ← zeros(size(x^l))                          ▷ initialize as matrix of zeros
for l = 1, ...., L do
    c^l ← l^th cell
    c^b ← boundary voxels of c^l
    for p in c^b do
        i, j, k ← location of p in x^l
        for m = 1, ...., L and m ≠ l do
            c^m ← m^th cell
            for q in c^m do
                d_pq ← E(p, q)                  ▷ Euclidean distance
            end for
        end for
        x^b(i, j, k) ← 1 − min(d_pq)
    end for
end for
```

3.1. 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. 1. The encoder and each of the two decoders consist of five consecutive convolution layers. In the encoding path, each convolution layer performs two $3 \times 3 \times 3$ convolutions with ReLU activation and batch normalization, followed by a $2 \times 2 \times 2$ 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 $2 \times 2 \times 2$ strides, followed by two $3 \times 3 \times 3$ 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 (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 (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. 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 (Cardoso et al., 2022).

3.2. 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 (Toma et al., 2022), 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 (Zhang et al., 2019) of two kinds of real bacteria species, *Escherichia coli* and *Shewanella oneidensis*. We have exploited an *Escherichia coli* image dataset from previously published works (Toma et al., 2022; Zhang et al., 2020). 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 min and another with an interval of 30 s. 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. 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 min 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 performance 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.

3.3. 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.

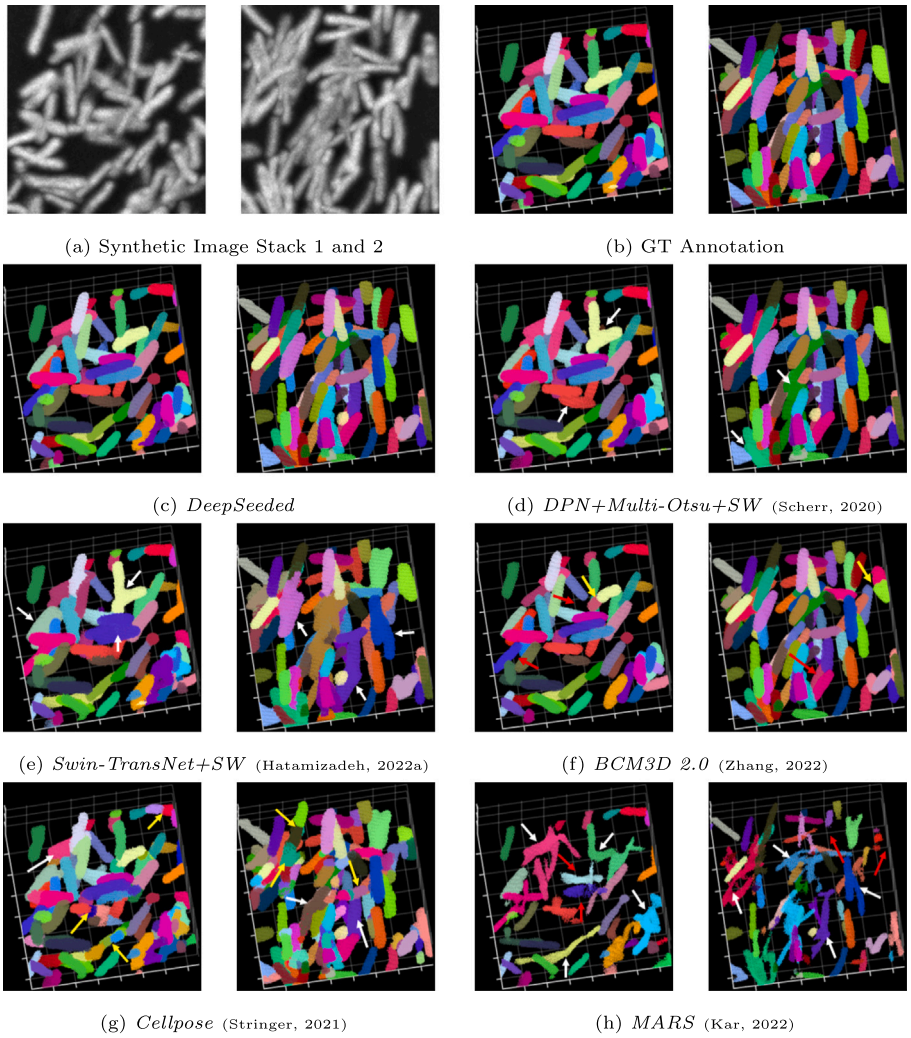
$$SCF1 = \frac{1}{N_{match}} \sum_{l=1}^{N_{match}} \frac{2 \times TP^l}{2 \times TP^l + FP^l + FN^l}$$

To further evaluate the accuracy of the segmentation in separating touching cells, we also compute a single-cell boundary F1 score (Wang et al., 2021) (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. 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. (2020), 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 (Liao et al., 2001) (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



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

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

(e) *Swin-TransNet+SW* (Hatamizadeh, 2022a) (f) *BCM3D 2.0* (Zhang, 2022)

(g) *Cellpose* (Stringer, 2021) (h) *MARS* (Kar, 2022)

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

segmentation followed by a seeded watershed-based post-processing. While such methods mentioned in the literature (Eschweiler et al., 2019; Kar et al., 2022; Kucharski & Fabijańska, 2021) exploit a standard U-Net convolutional network, we have adopted a more recent network architecture, Swin Transformer-based U-Net (Hatamizadeh, Nath, et al., 2022) 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 (Stringer et al., 2021). We have further compared against the latest deep learning-based 3D biofilm segmentation approach named *BCM3D 2.0* (Zhang et al., 2022), 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 (Kar et al., 2022) 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.

4. Experimental results and discussion

We demonstrate the qualitative comparison of the segmentation results on two synthetic biofilm test stacks in Fig. 3. The images shown in Fig. 3(a) are the maximum intensity projections (MIPs) of 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(f) 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(c), Fig. 3(f), and Fig. 3(d), 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(h) that the classical *MARS* approach suffers heavily in separating the touching cells and preserving the cell volume.

In Table 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

Table 2
Quantitative evaluation on 100 synthetic 3D biofilms.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW (Scherr, 2020)	0.893 ± 0.04	0.827 ± 0.08	0.883 ± 0.02	0.957 ± 0.02
Swin-TransNet+SW (Hatamizadeh, 2022a)	0.844 ± 0.06	0.485 ± 0.14	0.664 ± 0.03	0.686 ± 0.04
MARS (Kar, 2022)	0.651 ± 0.08	0.017 ± 0.01	0.377 ± 0.05	0.427 ± 0.02
BCM3D 2.0 (Zhang, 2022)	0.877 ± 0.05	0.863 ± 0.06	0.881 ± 0.03	0.962 ± 0.02
Cellpose (Stringer, 2021)	0.810 ± 0.06	0.440 ± 0.09	0.663 ± 0.03	0.743 ± 0.03
DeepSeeded	**0.948 ± 0.02**	**0.915 ± 0.05**	**0.904 ± 0.02**	**0.980 ± 0.01**

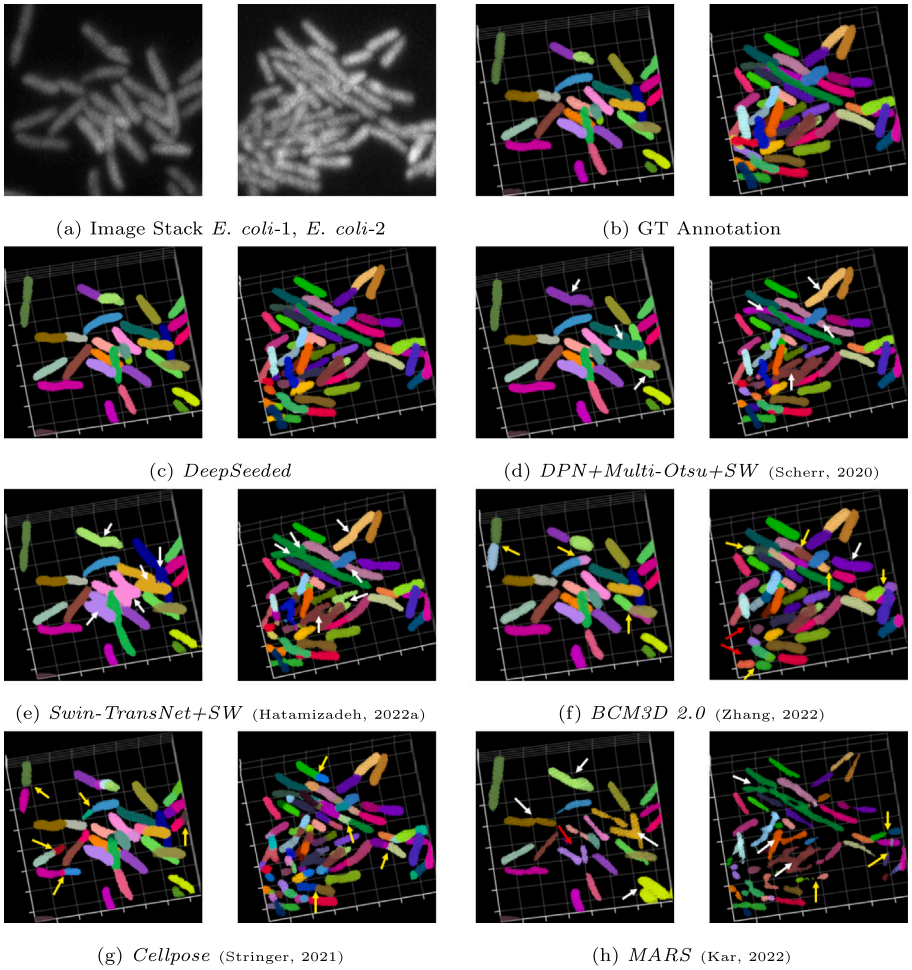


Fig. 4. Qualitative evaluation on two 3D *E.coli* images. The white, yellow, and red arrows indicate various locations of touching, broken, and missing cells, respectively.

visual observation from Fig. 3. The proposed method achieves higher average scores for each quantitative segmentation accuracy measure.

We also demonstrate the qualitative segmentation results on real biofilm stacks in Figs. 4 and 5. The images shown in Figs. 4(a) and 5(a) 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, 4, 5, and 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 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/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 qualitative test results on another real

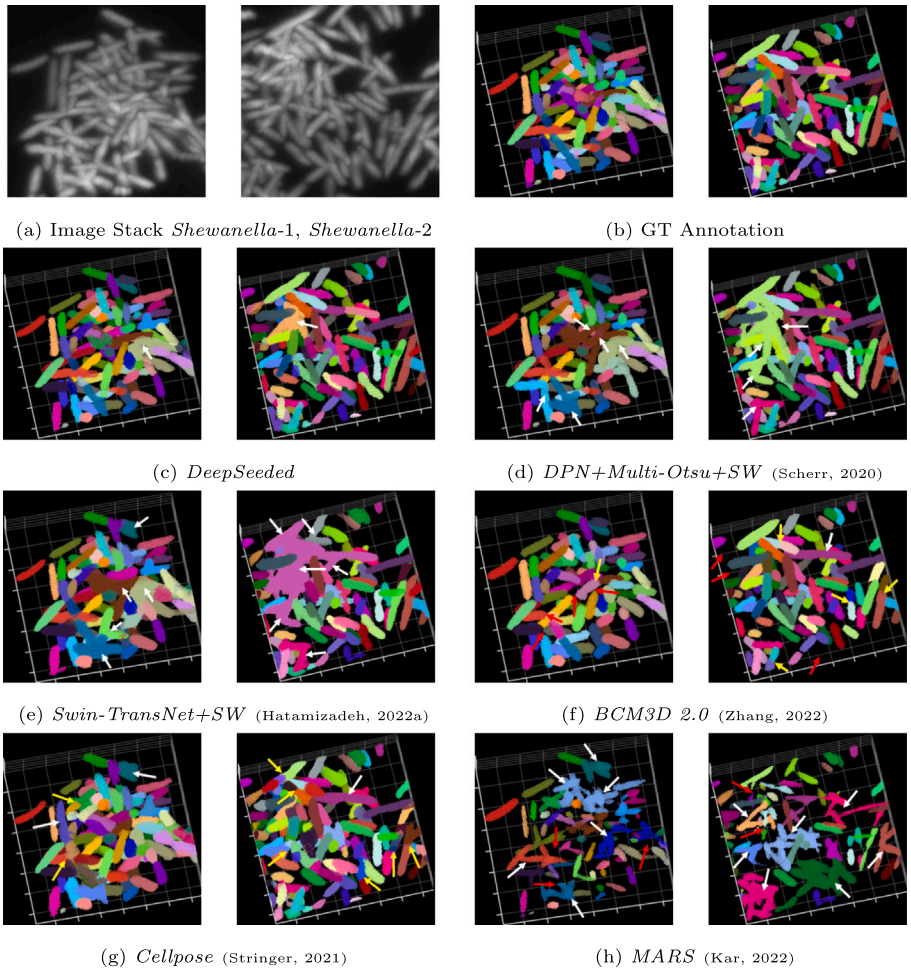


Fig. 5. Qualitative evaluation on two 3D *Shewanella* images. The white, yellow, and red arrows indicate various locations of touching, broken, and missing cells, respectively.

biofilm stack, denoted as "*Shewanella*-3" with dimensions of $200 \times 200 \times 50$, are presented in Fig. A.7 of the appendix. 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 s per epoch during training, resulting in a total training time of 5.0 h. The average testing time on a single instance was 1.7 s. As for the *DPN+Multi-Otsu+SW* method, the per epoch training time averaged at 90.5 s, leading to an overall training time of 6.3 h. The average testing time on a single instance was 4.6 s. Regarding the *Swin-TransNet+SW* method, each epoch's training time averaged at 20.8 s, resulting in a total training time of 1.4 h. The average testing time on a single instance was 3.1 s. For the *Cellpose* method, the per epoch training time was approximately 60.5 s, leading to an overall training time of 4.2 h. The average testing time on a single instance was 10.0 s. 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 s. In our proposed *DeepSeeded* method, the per epoch training time for the regression network (*Net-1*) averaged at 92.1 s, resulting in a total training time of 6.4 h. The per epoch training time for the voxel-wise classification network (*Net-2*) was approximately 17.5 s, leading to an overall training time of 1.2 h. The average testing time on a single instance was 4.9 s.

4.1. 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 7 and 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.

4.2. 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

Table 3
Quantitative evaluation on stack *E.coli*-1.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW (Scherr, 2020)	0.931	0.828	0.795	0.898
Swin-TransNet+SW (Hatamizadeh, 2022a)	0.852	0.407	0.674	0.734
MARS (Kar, 2022)	0.836	0.173	0.514	0.580
BCM3D 2.0 (Zhang, 2022)	0.921	0.825	0.793	0.899
Cellpose (Stringer, 2021)	0.800	0.286	0.584	0.624
DeepSeeded	**1.000**	**0.840**	**0.853**	**0.909**

Table 4
Quantitative evaluation on stack *E.coli*-2.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW (Scherr, 2020)	0.855	0.327	0.618	0.713
Swin-TransNet+SW (Hatamizadeh, 2022a)	0.800	0.434	0.635	0.752
MARS (Kar, 2022)	0.600	0.074	0.327	0.351
BCM3D 2.0 (Zhang, 2022)	0.842	0.316	0.593	0.666
Cellpose (Stringer, 2021)	0.671	0.197	0.560	0.658
DeepSeeded	**0.937**	**0.829**	**0.800**	**0.912**

Table 5
Quantitative evaluation on stack *Shewanella*-1.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW (Scherr, 2020)	0.821	0.680	0.786	0.883
Swin-TransNet+SW (Hatamizadeh, 2022a)	0.800	0.450	0.670	0.770
MARS (Kar, 2022)	0.562	0.123	0.446	0.541
BCM3D 2.0 (Zhang, 2022)	0.864	0.722	0.750	0.866
Cellpose (Stringer, 2021)	0.825	0.402	0.673	0.783
DeepSeeded	**0.885**	**0.874**	**0.964**	**0.967**

Table 6
Quantitative evaluation on stack *Shewanella*-2.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
DPN+Multi-Otsu+SW (Scherr, 2020)	0.834	0.717	0.811	0.925
Swin-TransNet+SW (Hatamizadeh, 2022a)	0.775	0.539	0.705	0.822
MARS (Kar, 2022)	0.577	0.110	0.427	0.538
BCM3D 2.0 (Zhang, 2022)	0.833	0.660	0.756	0.876
Cellpose (Stringer, 2021)	0.817	0.435	0.650	0.771
DeepSeeded	**0.918**	**0.900**	**0.976**	**0.977**

biofilms. However, there are several areas where further improvement can be made. In the proposed cascaded deep learning framework, the two networks have been trained separately on two different loss functions: one is for the regression task of two distance map estimations, and another is for the semantic segmentation task for classifying the cell seeds. Such separate training might not be optimal in terms of smooth information flow between the two networks. Therefore, in future development, the *DeepSeeded* framework can be further extended to train the two networks jointly. Such joint training can be achieved in an alternative optimization fashion, such as optimizing the regression loss by adjusting the weights of the regression network, similar to the current methodology; however, optimizing the voxel-wise (semantic) classification loss by adjusting the weights of both networks. The joint training strategy, which has been adopted in recent deep learning frameworks (Lee, Cho, & Kim, 2019; Wang & Zhang, 2022), can further enhance the learning of the *DeepSeeded* by ensuring better gradient flow during backpropagation.

Furthermore, 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 (Brügger, Baumgartner, & Konukoglu, 2019; Mescheder, Oechsle, Niemeyer, Nowozin, & Geiger, 2019; Qi, Yi, Su, & Guibas, 2017; Reich, Prangemeier, Cetin, & Koeppl, 2021). The memory-efficient CNN approaches leverage implicit 3D representations, known as occupancy values (Mescheder et al., 2019), 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 while still retaining the key benefits of our method, including effective cell border representation, specialized image quality-oriented loss, and the two-staged cascaded workflow.

5. Conclusion

This paper 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

Table 7
Ablation study result on stack *E.coli*-2.

Methods	$CCF1$		$SCF1$	$SCBF1$
	$IOU = 0.1$	$IOU = 0.5$		
Net-1+Multi-Otsu+SW	0.850	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 8
Ablation study result on stack *Shewanella*-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**

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 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. In the future, to further strengthen the ability of DeepSeeded, we aim to optimize the training strategy by performing joint optimization of the two deep networks in an alternative minimization fashion. Additionally, to enable efficient learning on large 3D training datasets, we intend to replace the traditional U-Net schemes used in our approach with memory-efficient U-Net approaches. By achieving fast and accurate 3D segmentation in densely packed cell images, we anticipate enabling more robust analysis of cell populations, including tasks such as tracking individual cell instances over time and quantifying their growth and division rates. The code of the presented work will be available at "https://engineering.virginia.edu/viva/viva-research".

CRediT authorship contribution statement

Tanjin Taher Toma: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing – original draft. **Yibo Wang:** Data curation, Investigation, Writing – original draft. **Andreas Gahlmann:** Project administration, Writing – review & editing. **Scott T. Acton:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported in part by the U.S. National Institute of General Medical Sciences under NIH Grant No. 1R01GM139002. For this study, no ethical approval was required. The authors have no conflicts of interest.

Appendix

In Fig. A.6, we present the intermediate maps generated from the two networks in the proposed *DeepSeeded* approach for an example *E.coli* image stack. The predicted cell distance map and the border neighbor distance map from the regression network are demonstrated in Figs. A.6(b) and A.6(c), 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. A.6(d). 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. A.6(e), which is referred to as the seed-labeled image. Finally,

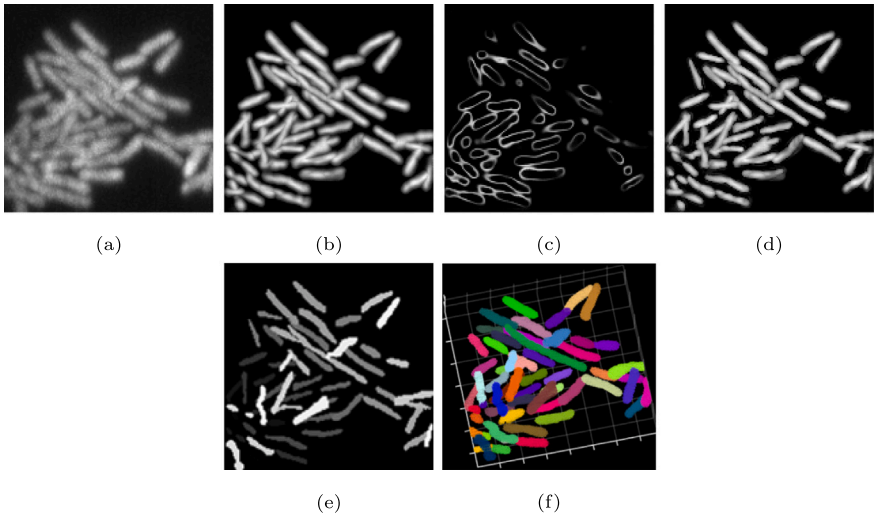
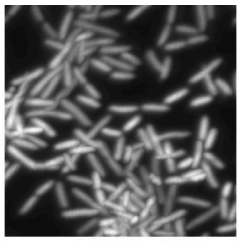
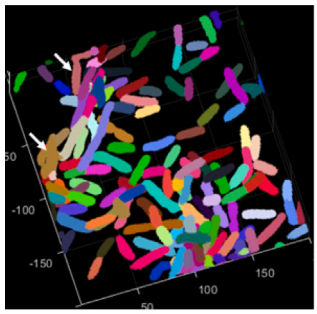


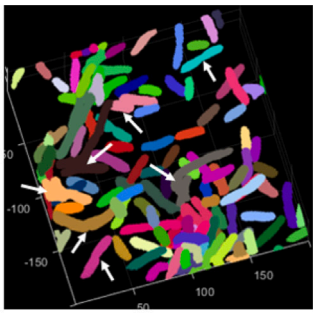
Fig. A.6. (a) Given a raw image stack, the intermediate maps from the two networks and the final segmentation by *DeepSeeded* approach. Here, (a) input stack (MIP), (b) predicted cell distance map (MIP), (c) predicted border neighbor distance map (single slice), (d) predicted difference map (MIP), (e) predicted seed labeled image (MIP), and (f) final segmentation (3D point cloud). The term MIP refers to the maximum intensity projection.



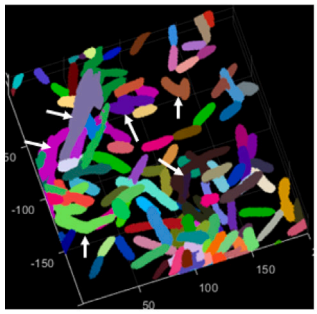
(a) Image Stack *Shewanella*-3



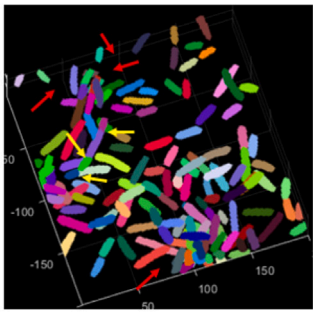
(b) *DeepSeeded*



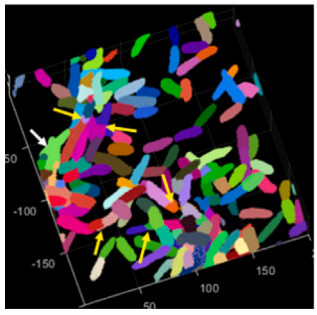
(c) *DPN+Multi-Otsu+SW* (Scherr, 2020)



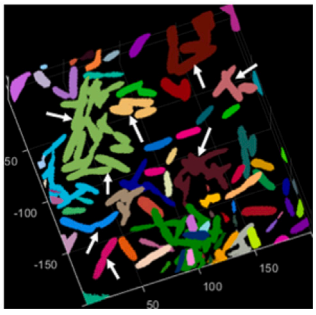
(d) *Swin-TransNet+SW* (Hatamizadeh, 2022a)



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



(f) *Cellpose* (Stringer, 2021)



(g) *MARS* (Kar, 2022)

Fig. A.7. Qualitative evaluation on a 3D *Shewanella* image. The white, yellow, and red arrows indicate various locations of touching, broken, and missing cells, respectively.

the instance segmentation result after applying the seeded watershed algorithm is illustrated in Fig. A.6(f).

In Fig. A.7, we qualitatively compare the segmentation results obtained by different approaches on the *Shewanella*-3 image. To highlight the segmentation errors in various methods, we have included arrows in the figures. 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 attributed 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.

References

Abeyrathna, D., Rauniyar, S., Sani, R. K., & Huang, P.-C. (2022). A morphological post-processing approach for overlapped segmentation of bacterial cell images. *Machine Learning and Knowledge Extraction*, *4*(4), 1024–1041.

Acton, S. T., & Ray, N. (2009). Biomedical image analysis: Segmentation. In *Synthesis lectures on image, video, and multimedia processing. Vol. 4. No. 1* (pp. 1–108). Morgan & Claypool Publishers.

Atta-Fosu, T., Guo, W., Jeter, D., Mizutani, C. M., Stopczynski, N., & Sousa-Neves, R. (2016). 3D clumped cell segmentation using curvature based seeded watershed. *Journal of Imaging*, *2*(4), 31.

Beucher, S., & Meyer, F. (2018). The morphological approach to segmentation: The watershed transformation. In *Mathematical morphology in image processing* (pp. 433–481). CRC Press.

Bjarnsholt, T., Alhede, M., Alhede, M., Eickhardt-Sørensen, S. R., Moser, C., Kühl, M., et al. (2013). The in vivo biofilm. *Trends in Microbiology*, *21*(9), 466–474.

Boykov, Y., & Funka-Lea, G. (2006). Graph cuts and efficient ND image segmentation. *International Journal of Computer Vision*, *70*(2), 109–131.

Brügger, R., Baumgartner, C. F., & Konukoglu, E. (2019). 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. Vol. 22* (pp. 429–437). Springer.

Caicedo, J. C., Roth, J., Goodman, A., Becker, T., Karhohs, K. W., Broisin, M., et al. (2019). Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, *95*(9), 952–965.

Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., et al. (2022). MONAI: An open-source framework for deep learning in healthcare. arXiv preprint arXiv:2211.02701.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer.

Chaturvedi, V., & Verma, P. (2016). Microbial fuel cell: A green approach for the utilization of waste for the generation of bioelectricity. *Bioresources and Bioprocessing*, *3*(1), 1–14.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., et al. (2021). Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306.

Chen, J., & Zhang, B. (2021). Segmentation of overlapping cervical cells with mask region convolutional neural network. *Computational and Mathematical Methods in Medicine*, *2021*.

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

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention* (pp. 424–432). Springer.

Drescher, K., Shen, Y., Bassler, B. L., & Stone, H. A. (2013). 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.

Eschweiler, D., Spina, T. V., Choudhury, R. C., Meyerowitz, E., Cunha, A., & Stegmaier, J. (2019). CNN-based preprocessing to optimize watershed-based cell segmentation in 3D confocal microscopy images. In *2019 IEEE 16th international symposium on biomedical imaging* (pp. 223–227). IEEE.

Hall-Stoodley, L., Costerton, J. W., & Stoodley, P. (2004). Bacterial biofilms: from the natural environment to infectious diseases. *Nature Reviews Microbiology*, *2*(2), 95–108.

Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H. R., & Xu, D. (2022). Swin unetr: Swin transformers for semantic segmentation of brain tumors in MRI images. In *International MICCAI brainlesion workshop* (pp. 272–284). Springer.

Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., et al. (2022). Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 574–584).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

He, Y., Gong, H., Xiong, B., Xu, X., Li, A., Jiang, T., et al. (2015). iCut: An integrative cut algorithm enables accurate segmentation of touching cells. *Scientific Reports*, *5*(1), 1–17.

Ilyas, T., Mannan, Z. I., Khan, A., Azam, S., Kim, H., & De Boer, F. (2022). TSFD-Net: Tissue specific feature distillation network for nuclei segmentation and classification. *Neural Networks*, *151*, 1–15.

Jiang, Y., Chen, W., Liu, M., Wang, Y., & Meijering, E. (2020). 3D neuron microscopy image segmentation via the ray-shooting model and a DC-BLSTM network. *IEEE Transactions on Medical Imaging*, *40*(1), 26–37.

Jung, C., & Kim, C. (2010). Segmenting clustered nuclei using H-minima transform-based marker extraction and contour parameterization. *IEEE Transactions on Biomedical Engineering*, *57*(10), 2600–2604.

Kar, A., Petit, M., Refahi, Y., Cerutt, G., Godin, C., & Traas, J. (2022). Benchmarking of deep learning algorithms for 3D instance segmentation of confocal image datasets. *PLoS computational biology*, *18*(4), e1009879.

Koyuncu, C. F., Akhan, E., Ersahin, T., Cetin-Atalay, R., & Gunduz-Demir, C. (2016). Iterative h-minima-based marker-controlled watershed for cell nucleus segmentation. *Cytometry Part A*, *89*(4), 338–349.

Kucharski, A., & Fabijańska, A. (2021). CNN-watershed: A watershed transform with predicted markers for corneal endothelium image segmentation. *Biomedical Signal Processing and Control*, *68*, Article 102805.

Lee, J., Cho, S., & Kim, M. (2019). An end-to-end joint learning scheme of image compression and quality enhancement with improved entropy minimization. arXiv preprint arXiv:1912.12817.

Li, Q., & Shen, L. (2019). 3D neuron reconstruction in tangled neuronal image with deep networks. *IEEE Transactions on Medical Imaging*, *39*(2), 425–435.

Li, X., Wang, Y., Tang, Q., Fan, Z., & Yu, J. (2019). Dual U-Net for the segmentation of overlapping glioma nuclei. *IEEE Access*, *7*, 84040–84052.

Li, Q., Zhang, Y., Liang, H., Gong, H., Jiang, L., Liu, Q., et al. (2021). Deep learning based neuronal soma detection and counting for Alzheimer's disease analysis. *Computer Methods and Programs in Biomedicine*, *203*, Article 106023.

Liao, P.-S., Chen, T.-S., Chung, P.-C., et al. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, *17*(5), 713–727.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4460–4470).

Meyer, F., & Beucher, S. (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation*, *1*(1), 21–46.

Mukherjee, S., & Acton, S. T. (2014). Region based segmentation in presence of intensity inhomogeneity using legendre polynomials. *IEEE Signal Processing Letters*, *22*(3), 298–302.

Phoulady, H. A., Goldgof, D. B., Hall, L. O., & Mouton, P. R. (2016). Nucleus segmentation in histology images with hierarchical multilevel thresholding. In *Medical imaging 2016: Digital pathology. Vol. 9791* (pp. 280–285). SPIE.

Prangemeier, T., Reich, C., & Koeppl, H. (2020). Attention-based transformers for instance segmentation of cells in microstructures. In *2020 IEEE international conference on bioinformatics and biomedicine* (pp. 700–707). IEEE.

Prangemeier, T., Wildner, C., Françani, A. O., Reich, C., & Koeppl, H. (2022). Yeast cell segmentation in microstructured environments with deep learning. *Biosystems*, *211*, Article 104557.

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

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems. Vol. 30*.

Reich, C., Prangemeier, T., Cetin, Ö., & Koeppl, H. (2021). OSS-Net: memory efficient high resolution semantic segmentation of 3D medical data. arXiv preprint arXiv:2110.10640.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Salem, N., Sobhy, N. M., & El Dosoky, M. (2016). A comparative study of white blood cells segmentation using otsu threshold and watershed transformation. *Journal of Biomedical Engineering and Medical Imaging*, *3*(3), 15.

Scherr, T., Löffler, K., Böhland, M., & Mikut, R. (2020). Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. *PLoS One*, *15*(12), Article e0243219.

Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018). 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. Vol. 11* (pp. 265–273). Springer.

Schultz, M. P., Bendick, J., Holm, E., & Hertel, W. (2011). Economic impact of biofouling on a naval surface ship. *Biofouling*, *27*(1), 87–98.

Shen, S. P., Tseng, H.-a., Hansen, K. R., Wu, R., Gritton, H. J., Si, J., et al. (2018). Automatic cell segmentation by adaptive thresholding (ACSAT) for large-scale calcium imaging datasets. *Eneuro*, *5*(5).

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

Soille, P., et al. (1999). *Morphological image analysis: Principles and applications. Vol. 2. No. 3*. Springer.

Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: A generalist algorithm for cellular segmentation. *Nature Methods*, *18*(1), 100–106.

Toma, T. T., Wu, Y., Wang, J., Srivastava, A., Gahlmann, A., & Acton, S. T. (2022). Realistic-shape bacterial biofilm simulator for deep learning-based 3D single-cell segmentation. In *2022 IEEE 19th international symposium on biomedical imaging* (pp. 1–5). IEEE.

Vicar, T., Balvan, J., Jaros, J., Jug, F., Kolar, R., Masarik, M., et al. (2019). Cell segmentation methods for label-free contrast microscopy: Review and comprehensive comparison. *BMC Bioinformatics*, *20*(1), 1–25.

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003. Vol. 2* (pp. 1398–1402). IEEE.

Wang, J., Tabassum, N., Toma, T. T., Wang, Y., Gahlmann, A., & Acton, S. T. (2022). 3D GAN image synthesis and dataset quality assessment for bacterial biofilm. *Bioinformatics*, *38*(19), 4598–4604.

Wang, W., Taft, D. A., Chen, Y.-J., Zhang, J., Wallace, C. T., Xu, M., et al. (2019). Learn to segment single cells with deep distance estimator and deep cell detector. *Computers in Biology and Medicine*, *108*, 133–141.

Wang, J., & Zhang, M. (2022). Geo-SIC: Learning deformable geometric shapes in deep image classifiers. arXiv preprint arXiv:2210.13704.

Wang, J., Zhang, M., Zhang, J., Wang, Y., Gahlmann, A., & Acton, S. T. (2021). Graph-theoretic post-processing of segmentation with application to dense biofilms. *IEEE Transactions on Image Processing*, *30*, 8580–8594.

Wei, X., Liu, Q., Liu, M., Wang, Y., & Meijering, E. (2022). 3D soma detection in large-scale whole brain images via a two-stage neural network. *IEEE Transactions on Medical Imaging*, *42*(1), 148–157.

Wolny, A., Cerrone, L., Vijayan, A., Tofanelli, R., Barro, A. V., Louveaux, M., et al. (2020). Accurate and versatile 3D segmentation of plant tissues at cellular resolution. *Elife*, *9*, Article e57613.

Xiong, L., Zhang, D., Li, K., & Zhang, L. (2020). The extraction algorithm of color disease spot image based on Otsu and watershed. *Soft Computing*, *24*(10), 7253–7263.

Yang, B., Liu, M., Wang, Y., Zhang, K., & Meijering, E. (2021). Structure-guided segmentation for 3D neuron reconstruction. *IEEE Transactions on Medical Imaging*, *41*(4), 903–914.

Zhang, J., Wang, Y., Donarski, E. D., Toma, T. T., Miles, M. T., Acton, S. T., et al. (2022). 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.

Zhang, M., Zhang, J., Wang, J., Achimovich, A. M., Aziz, A. A., Corbitt, J., et al. (2019). 3D imaging of single cells in bacterial biofilms using lattice light-sheet microscopy. *Biophysical Journal*, *116*(3), 25a.

Zhang, M., Zhang, J., Wang, Y., Wang, J., Achimovich, A. M., Acton, S. T., et al. (2020). Non-invasive single-cell morphometry in living bacterial biofilms. *Nature Communications*, *11*(1), 1–13.

Zhao, Z., Yang, L., Zheng, H., Guldner, I. H., Zhang, S., & Chen, D. Z. (2018). Deep learning based instance segmentation in 3D biomedical images using weak annotation. In *International conference on medical image computing and computer-assisted intervention* (pp. 352–360). Springer.