



# LCU-Net: A novel low-cost U-Net for environmental microorganism image segmentation<sup>☆</sup>



Jinghua Zhang<sup>a</sup>, Chen Li<sup>a,\*</sup>, Sergey Kosov<sup>b</sup>, Marcin Grzegorzek<sup>c</sup>, Kimiaki Shirahama<sup>d</sup>, Tao Jiang<sup>e</sup>, Changhao Sun<sup>a,f</sup>, Zihan Li<sup>a</sup>, Hong Li<sup>a</sup>

<sup>a</sup> Microscopic Image and Medical Image Analysis Group, College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China

<sup>b</sup> Faculty of Data Engineering, Jacobs University Bremen, Bremen, Germany

<sup>c</sup> Institute of Medical Informatics, University of Luebeck, Luebeck, Germany

<sup>d</sup> Faculty of Science and Engineering, Kindai University, Higashiosaka, Osaka, Japan

<sup>e</sup> Control Engineering College, Chengdu University of Information Technology, Chengdu, China

<sup>f</sup> Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China

## ARTICLE INFO

### Article history:

Received 9 January 2020

Revised 19 September 2020

Accepted 31 January 2021

Available online 13 February 2021

### Keywords:

Environmental miroorganisms

Image segmentation

Deep convolutional neural networks

Low-cost

## ABSTRACT

In this paper, we propose a novel *Low-cost U-Net* (LCU-Net) for the *Environmental Microorganism* (EM) image segmentation task to assist microbiologists in detecting and identifying EMs more effectively. The LCU-Net is an improved Convolutional Neural Network (CNN) based on U-Net, Inception, and concatenate operations. It addresses the limitation of single receptive field setting and the relatively high memory cost of U-Net. Experimental results show the effectiveness and potential of the proposed LCU-Net in the practical EM image segmentation field.

© 2021 Elsevier Ltd. All rights reserved.

## 1. Introduction

In industrialized countries, industrial pollution has been threatening human health and environments. Thus, different methods of controlling, reducing, and eliminating pollution are being established. The methods of eliminating environmental pollution generally include three major categories: chemical, physical, and biological approaches. By contrast, the biological method is more harmless and efficient [1]. *Environmental Microorganisms* (EMs) are very tiny living beings in our surroundings, which are natural decomposers and indicators. For example, *Actinophrys* can digest the organic waste in sludge and increase the quality of freshwater; *Rotifera* can decompose rubbish in the water and reduce eutrophication. Therefore, EM research plays a significant role in managing environmental pollution [2], and the identification of EMs is the basic step for related research.

Generally, there are four traditional types of EM identification methods. The first is the “chemical method”, which is highly

accurate but often results in secondary pollution of chemical reagents [3]. The second strategy is the “physical method”. It also has high accuracy but requires expensive equipment [3]. The third is the “molecular biological method”, which distinguishes EMs by sequence analysis of gene [4]. This strategy needs expensive equipment, massive time consumption, and professional researcher. The fourth is the “morphological method”, which needs a skillful operator to observe EMs under a microscope and give the EM identities by their shape characteristics [1]. Hence, these traditional methods have their respective disadvantages in practical work.

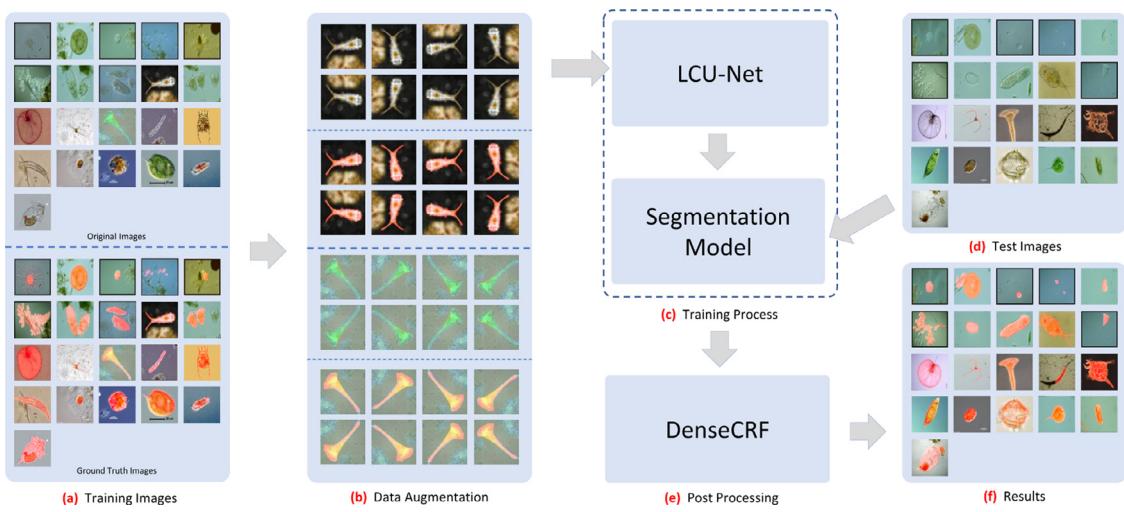
Due to the drawbacks of the traditional methods and the excellent performance on segmentation tasks of CNNs, the widely used U-Net [5] is firstly considered in our work. The training strategy used in U-Net relies on the strong use of data augmentation to use the available annotated samples more efficiently [5]. Besides, the end-to-end structure used in U-Net can retrieve the shallow information of the network [5]. Therefore, U-Net provides the high possibility to achieve high accuracy with our small EM training data set.

The cost and reliability of CNN are extremely important [6]. However, the adaptability of U-Net is limited by its single receptive field setting, and the memory cost of U-Net is relatively high in practical work, so we propose a novel *Low-cost U-Net* (LCU-Net)

\* This work is supported by the “National Natural Science Foundation of China” (No. 61806047).

\* Corresponding author.

E-mail address: [lichen201096@hotmail.com](mailto:lichen201096@hotmail.com) (C. Li).



**Fig. 1.** The workflow of the proposed EM image segmentation method using LCU-Net.

for the EM image segmentation task to assist microbiologists to identify EMs more effectively. The LCU-Net is an improved CNN based on U-Net [5], Inception [7], and concatenate operations [8]. In contrast to the original U-Net, it increases the overall segmentation performance and reduces the memory cost. Besides, we apply fully connected Conditional Random Field (Dense CRF) [9], which can obtain the spatial information between pixels in an image, as a post-processing step to enhance the segmentation result of LCU-Net. The workflow of the proposed LCU-Net segmentation approach is shown in Fig. 1.

In Fig. 1, (a) Training Images: The training set contains 21 categories of EM images and their corresponding ground truth (GT) images. (b) Data Augmentation: To solve a small training set problem, the size of data set is increased. (c) Training Process: LCU-Net is trained to perform the segmentation task and generate a segmentation model. (d) Test Images: The test set contains 21 categories of EM images. (e) Post-processing: Dense CRF is used to enhance the segmentation results by spatial information among pixels in an image.

The main contributions of this paper are as follows:

- We propose LCU-Net for EM image segmentation to assist microbiologists, and it achieves better segmentation performance than that of U-Net.
- LCU-Net newly applies thinner convolution filters and concatenate operations to reduce the memory cost to less than one-third of U-Net.

The structure of this paper is as follows: Section 2 is the related work about existing EM segmentation methods. Section 3 gives a detailed description of LCU-Net. Section 4 introduces experiment settings, evaluation methods, and results. Section 5 closes this paper with a brief conclusion.

## 2. Related work

In this section, related works about microorganism image segmentation techniques are briefly summarized in Table 1, including classical and machine learning methods. For more details, please refer to our previous survey in [10].

### 2.1. Classical segmentation methods

As shown in Table 1, threshold-based, edge-based, and region-based methods are usually used for the EM image segmentation task. The basic idea of threshold-based methods is to find out

**Table 1**  
Microorganism image segmentation methods.

Category	Subcategory	Related work
Classical	Threshold-based Methods	[11–15]
	Edge-based Methods	[16–19]
	Region-based Methods	[20–23]
Machine Learning	Unsupervised Methods	[24–26]
	Supervised Methods	[2,27–30]

the optimal threshold value of an image first and then binarize this image with the threshold value [31]. Otsu thresholding algorithm [32] is one of the most representative threshold-based methods. Edge detection methods transform an original image into an edge image benefiting from the changes of gray tones in the image [33], such as the Canny edge detection method [34]. Region-based methods divide the entire image into subregions, such as the Watershed algorithm [35].

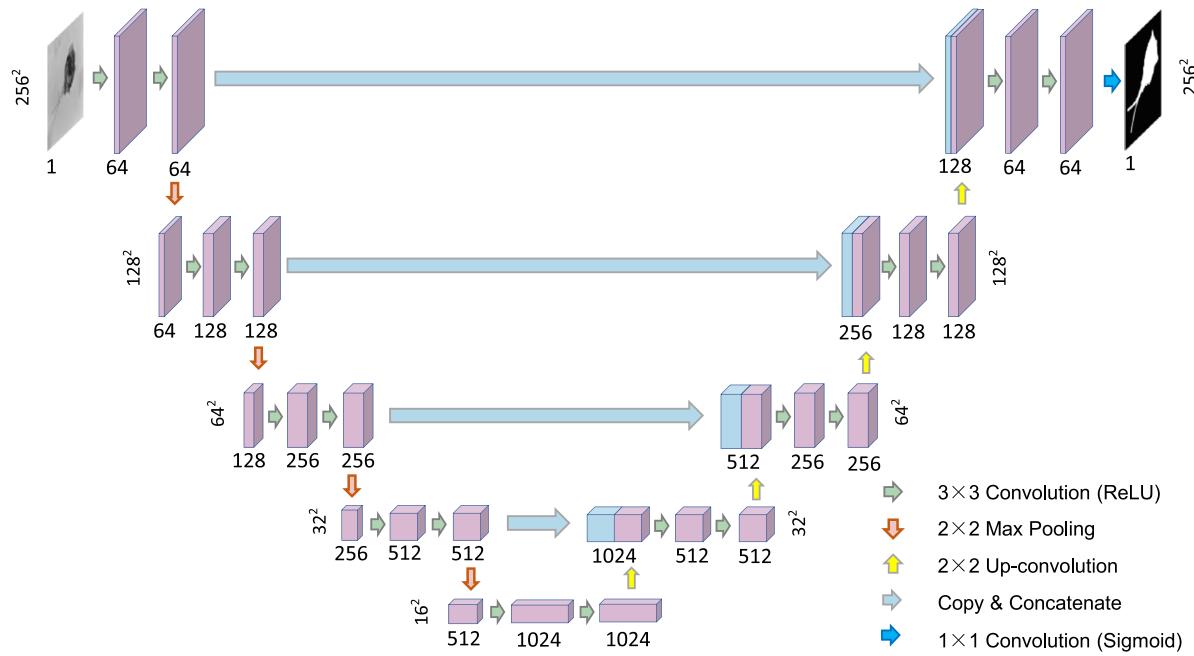
The works [11–15] use different threshold-based methods for microorganism image segmentation. For instance, [11] shows a comparison between threshold-based segmentation methods for evaluating biomass distribution in heterogeneous biofilms. The last result shows that the iterative selection method is superior. In [14], different algorithms based on Otsu thresholding are applied for the segmentation of floc and filaments to enhance the monitoring of activated sludge in wastewater treatment plants.

The works [16–19] use edge-based microorganism image segmentation methods. For example, a segmentation work, which uses Canny as the basic step, is introduced to identify individual microorganisms from a group of overlapping bacteria in [16]. In [19], a segmentation method based on active contour is proposed to segment large size images of zooplankton.

The works [20–23] use region-based methods for microorganism image segmentation. For example, in [20], the segmentation is performed on gray-level images using a marker controlled watershed method. In [22], after converting the color mode and using morphological operations to denoise, a seeded region-growing watershed algorithm is applied for segmentation.

### 2.2. Machine learning based segmentation methods

As shown in Table 1, the machine learning methods for microorganism image segmentation are grouped into two categories: Unsupervised and supervised learning. There are some representa-



**Fig. 2.** The network structure of U-Net.

tive unsupervised methods for microorganism image segmentation, such as *k*-means [36] and Markov Random Field (MRF) [37]. Besides, U-Net [5] is one of the most representative supervised segmentation methods for microorganism image segmentation.

The works [24–26] use unsupervised methods. For example, [24] evaluates clustering and threshold segmentation techniques on tissue images containing tuberculosis bacilli. The final result shows that *k*-means clustering ( $k = 3$ ) is outstanding. In [25], a comparison between CRFs and region-based segmentation methods is presented. The final result shows that these two kinds of microorganism segmentation methods have an average recognition rate higher than 80%.

The related works [27–30] use supervised methods in their works. For example, [29] proposes a segmentation system to monitor the algae in the water. First, image enhancement (sharpening) is applied using a Retinex filtering technique. Then, segmentation is done by using the Support Vector Machine (SVM). In [30], Rift Valley Virus is the segmentation subject. Because of the insufficient data set, they use data augmentation to assist U-Net in the segmentation work.

In our previous work [2], a supervised approach named Local-Global CRF for EM image segmentation is proposed. VGG-16 is trained first with EM images to generate pixel-level and global-level features, then these features are used to train Random Forest (RF) classifiers, and after that, these trained RF classifiers are used as unary potentials by a CRF model. Finally, with the pairwise potentials, the CRF model is applied to perform the EM image segmentation task.

We find that few existing studies use neural networks for microorganism image segmentation from our investigation of existing methods. Since our data set contains 21 EM classes, the images of different EM classes vary greatly. Therefore, methods like Otsu, Canny, *k*-means, and other methods may not perform well. Besides, in recent years, with the extensive use of neural networks in image segmentation, image segmentation performance has been greatly improved. Therefore, U-Net, one of the most representative neural networks in image segmentation, is first considered in our work.

### 3. LCU-Net based EM image segmentation method

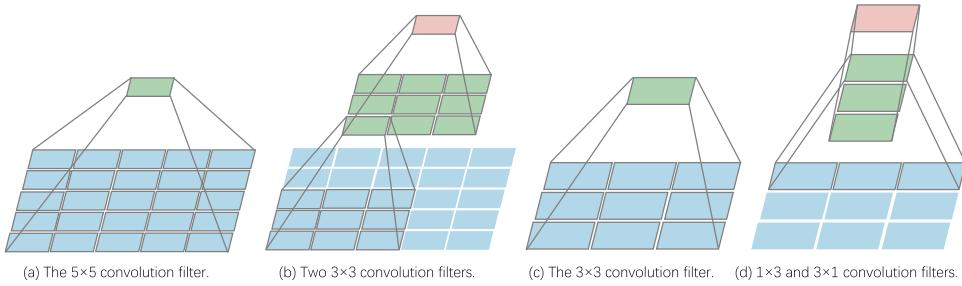
#### 3.1. LCU-Net

Although U-Net is widely used, its adaptability is limited by its single receptive field setting, and its memory cost is also relatively high in practical work. To address U-Net's limitation, we propose LCU-Net, a CNN based on U-Net [5], Inception-V3 [7], and concatenate operations [8].

##### 3.1.1. Basic knowledge of U-Net

As the structure is shown in Fig. 2, U-Net is a CNN that is initially used to perform the microscopic image segmentation task. The training strategy of U-Net relies on the strong use of data augmentation to make more effective use of the available annotated samples. [5]. Besides, the end-to-end structure of U-Net can retrieve the shallow information of the network [5].

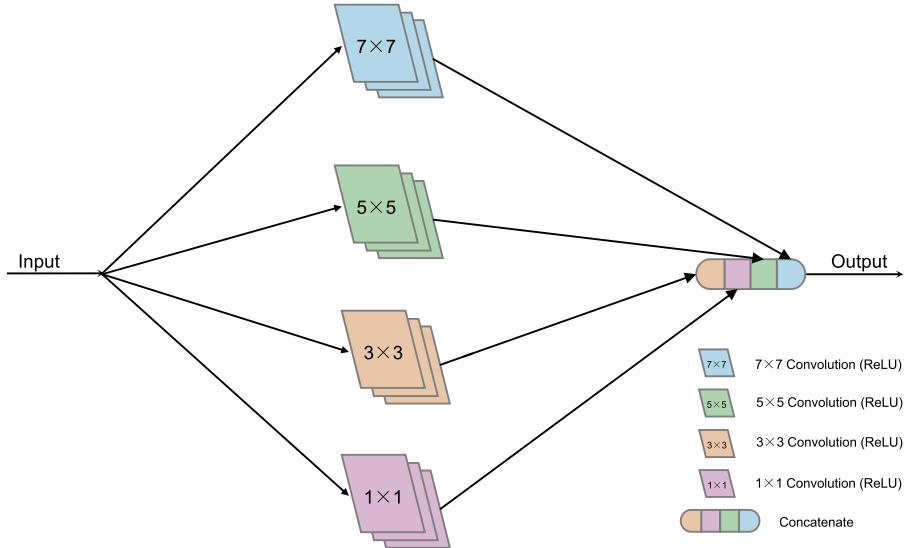
The structure of U-Net is symmetrical, consisting of a contracting path (left side) and an expansive path (right side) [5]. In the contracting path, each downsampling step contains a sequence of two  $3 \times 3$  convolution operations (each has a rectified linear unit (ReLU)) followed by a max-pooling operation with the size of  $2 \times 2$  and stride of 2 pixels. Pooling operation with the stride of 2 pixels can change the feature map's size into half of the original. In the contracting path, the downsampling step is repeated four times, and as the number of repetitions increases, the number of convolution filters increases by twice. As a result, the number of feature map channels is double. In the expansive path, there are three main operations in each upsampling step. The first is  $2 \times 2$  up-convolution operation (a  $2 \times 2$  upsampling operation followed by a  $2 \times 2$  convolution operation). The second is copy and concatenate. It copies the feature map generated by the corresponding layer from the contracting path and connects it with the feature map generated from up-convolution. This operation can help the network retrieve the spatial information lost by pooling operations [38]. The third operation is a sequence of two  $3 \times 3$  convolution operations (each has a ReLU). The upsampling step is repeated four times. The segmentation result is generated by the network's



**Fig. 3.** The strategies used by Inception-V2 and Inception-V3 to replace the big filter in the original Inception structure.



**Fig. 4.** An example of EM images with multiple scales.



**Fig. 5.** The structure of BLOCK-I.

final layer, a  $1 \times 1$  convolution operation with an activation function named Sigmoid.

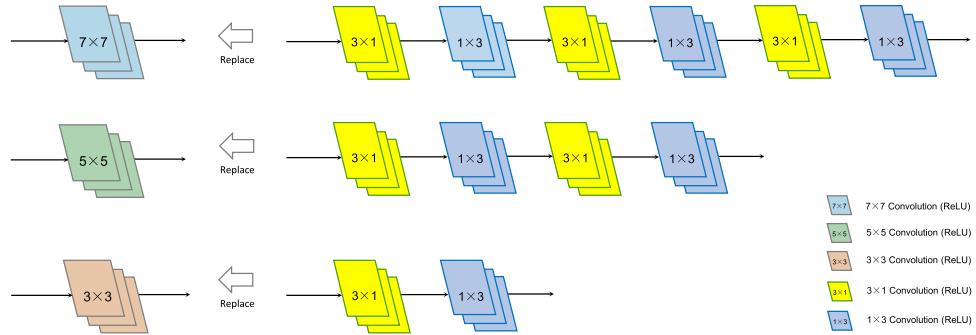
### 3.1.2. Basic knowledge of inception

The original Inception, which jointly uses filters of different sizes ( $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$ ), is proposed in GoogleNet [39]. Due to the utilization of these filters, Inception can adapt to multi-scale objects. However, there are also some disadvantages, such as, the increasing of parameters, overfitting, and vanishing gradient. To address these problems, Inception-V2 combines a couple of  $3 \times 3$  convolution filters to replace a  $5 \times 5$  convolution filter [7]. As shown in Fig. 3(a) and 3(b), a  $5 \times 5$  filter and a sequence of two  $3 \times 3$  filters have the same receptive fields, where the former has more parameters than the latter [7]. For further optimization, Inception-V3 proposes a better approach, which combines a  $1 \times N$  convolution filter and a  $N \times 1$  convolution filter instead of a  $N \times N$  convolution filter [7]. As shown in Fig. 3(c) and 3(d), a  $N \times N$  filter and a sequence of  $1 \times N$  and  $N \times 1$  filters have the same receptive field, where the former has more parameters than the latter.

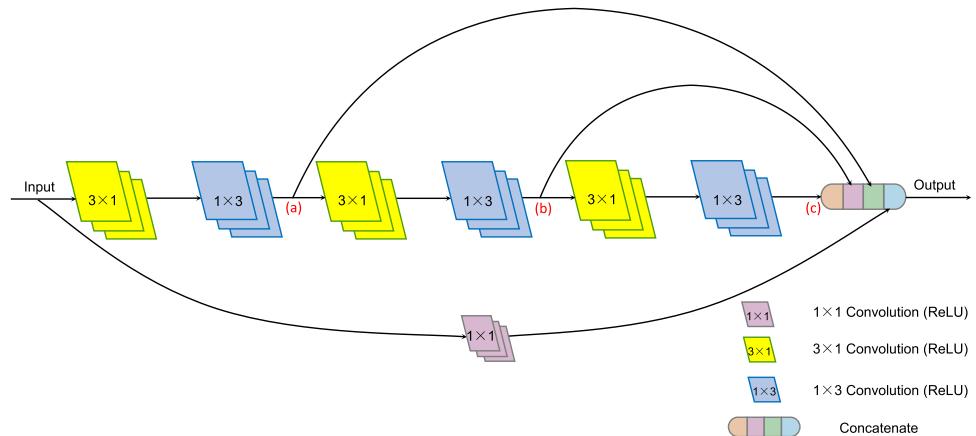
### 3.1.3. The structure of LCU-Net

There are multi-scale objects in EM images, as shown in Fig. 4. Considering that U-Net is difficult to adapt to this situation of multi-scale objects in EM images as shown in Fig. 4, we propose LCU-Net to address this problem. As the U-Net structure shown in Fig. 2, there is a sequence of two  $3 \times 3$  convolution operations before each pooling operation, each up-convolution operation, and the final convolution operation with Sigmoid, so the receptive field is limited. In contrast, in Inception series, convolution filters of different sizes are used to obtain various receptive fields. Hence, we propose BLOCK-I as shown in Fig. 5, which incorporates  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  convolution filters in parallel [40].

Although BLOCK-I can improve the adaptability of the network, it increases the memory cost. As mentioned in Section 3.1.2, Inception-V2 uses a sequence of two  $3 \times 3$  convolution filters to replace a  $5 \times 5$  convolution filter, and Inception-V3 uses a sequence of  $1 \times N$  and  $N \times 1$  convolution filters to replace a  $N \times N$  convolution filter [7]. Considering these strategies in Inception-V2 and Inception-V3, a sequence of three  $3 \times 3$  convolution filters has the same receptive field as a  $7 \times 7$  convolution filter. Further, a sequence of  $1 \times 3$  and  $3 \times 1$  convolution filters is used to replace a  $3 \times 3$  convolution filter. Therefore, the  $7 \times 7$  convolution filter of



**Fig. 6.** The examples of using multiple sequences of  $1 \times 3$  and  $3 \times 1$  convolution filters to replace convolution filters of different sizes.



**Fig. 7.** The structure of BLOCK-II.

BLOCK-I can be replaced by three sequences of  $1 \times 3$  and  $3 \times 1$  convolution filters. Similarly, the  $5 \times 5$  convolution filter of BLOCK-I can be replaced by two sequences of  $1 \times 3$  and  $3 \times 1$  convolution filters. The examples are shown in Fig. 6.

In Fig. 6, the  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  convolution filters can be replaced by one, two, and three sequences of  $1 \times 3$  and  $3 \times 1$  convolution filters, respectively. We find that the former layers of the three sequences of  $1 \times 3$  and  $3 \times 1$  convolution filters, which are used to replace the  $7 \times 7$  convolution filter, can be used to obtain the same receptive fields as  $3 \times 3$  and  $5 \times 5$  convolution filters by concatenate operations. By this idea, we propose BLOCK-II as shown in Fig. 7.

In Fig. 7, there are three nodes annotated as (a), (b), and (c) in BLOCK-II, respectively. The concatenate operation is applied to nodes (a), (b), and (c). It can obtain the same receptive fields as  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  convolution filters, respectively. Simultaneously, the above approach avoids the excessive use of the sequence of  $1 \times 3$  and  $3 \times 1$  convolution filters. Therefore, this approach can reduce more memory cost. Because of the lower cost of memory with BLOCK-II, we newly deploy BLOCK-II in LCU-Net. The structure of the proposed network is shown in Fig. 8.

Besides, we add a batch normalization layer after each convolution layer and convolution transpose layer to reduce the internal covariate shift [41] in LCU-Net, and the padding in mode “same” is applied to each convolution filter. The details of the LCU-Net are shown in Table 2. To make a comparison with BLOCK-I, we also provide the details of the network based on BLOCK-I, which is called U-Net-BI, in Table 2.

### 3.2. Post-processing with dense CRF

Although CNNs show good performance on the image segmentation task, there are still some shortages. It cannot take the de-

pendency among local variables into consideration [42]. The CNN works through the receptive field of the convolution filter. That is why the size of the convolution filter is so important to the performance of CNN. In [9], the fully connected CRF (Dense CRF) can establish pairwise potentials on all pairs of pixels in an image. As the Dense CRF workflow is shown in Fig. 9, when we use this approach as post-processing, it can effectively obtain the global information of the whole image to enhance the segmentation result.

The Dense CRF model of [9] employs the energy function, which is the sum of unary potential and pairwise potential. The function is shown in Eq. (1).

$$E(\mathbf{x}) = \sum_i U(x_i) + \sum_{i,j} P(x_i, x_j) \quad (1)$$

In Eq. (1),  $\mathbf{x}$  is the label assignment of pixel.  $U(x_i)$  represents the unary potential, which measures the inverse likelihood of the  $i$ -th pixel taking the label  $x_i$ , and  $P(x_i, x_j)$  means the pairwise potential, which measures the cost of assigning labels  $x_i, x_j$  to  $i$ -th pixel,  $j$ -th pixel simultaneously [43].

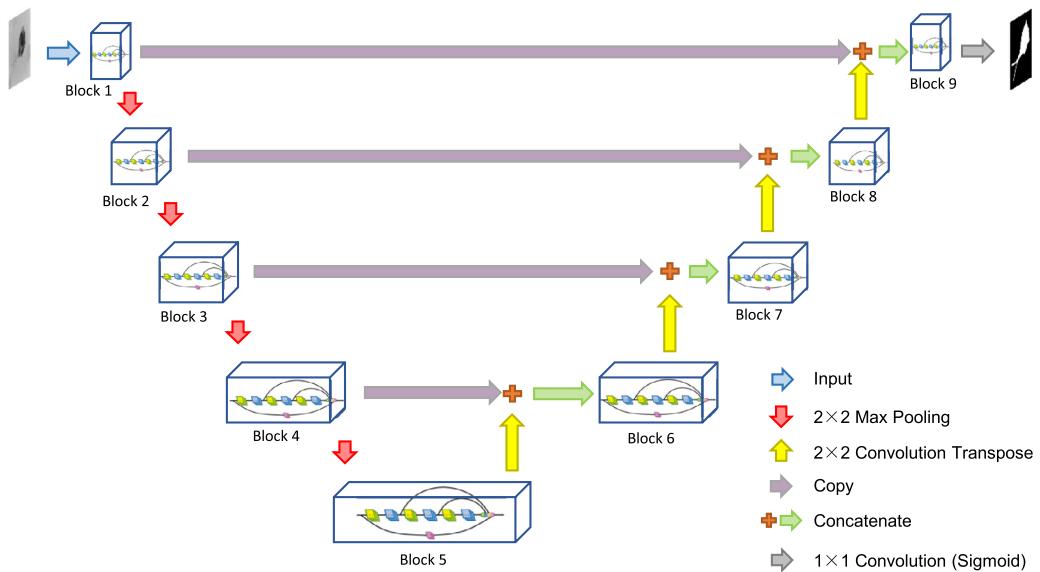
In our model, we use Eq. (2) as unary potential, where  $L(x_i)$  is the label assignment probability at the  $i$ -th pixel as computed by our neural networks [44].

$$U(x_i) = -\log L(x_i) \quad (2)$$

The pairwise potential is defined as Eq. (3),

$$P(x_i, x_j) = \phi(x_i, x_j) \underbrace{\sum_{m=1}^M \omega^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)} \quad (3)$$

where  $\phi(x_i, x_j)$  is a penalty term on the labelling [42]. As explained in [9],  $\phi(x_i, x_j)$  is given by the Potts model. If the  $i$ -th pixel

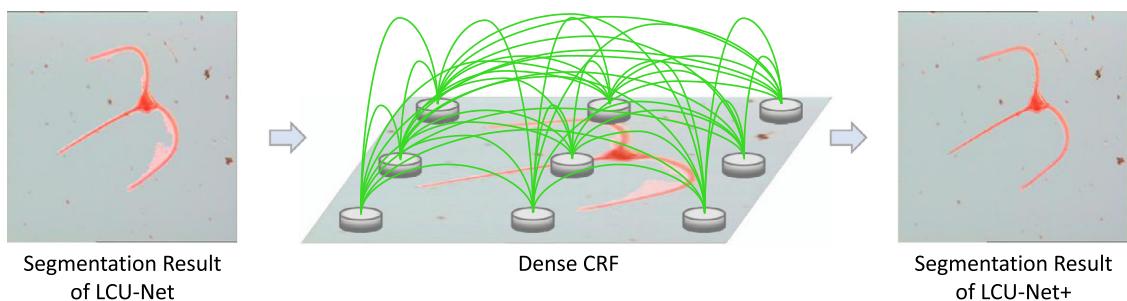


**Fig. 8.** The network structure of the LCU-Net. The details of Blocks (Block 1, Block 2,..., Block 9) are shown in [Table 2](#).

**Table 2**

A comparison of LCU-Net and U-Net-BI. Con2D is 2D convolution operation provided in Keras.

Block	Model		Filter Number	Block	Model		Filter Number
	U-Net-BI	LCU-Net			U-Net-BI	LCU-Net	
Block 1 & Block 9	Con2D(3,3)	Con2D(3,1) Con2D(1,3)	16	Block 2 & Block 8	Con2D(3,3)	Con2D(3,1) Con2D(1,3)	32
	Con2D(5,5)	Con2D(3,1)			Con2D(5,5)	Con2D(1,3)	
	Con2D(7,7)	Con2D(1,3) Con2D(3,1)			Con2D(7,7)	Con2D(3,1) Con2D(1,3)	
Block 3 & Block 7	Con2D(1,1)	Con2D(1,1)	64	Block 4 & Block 6	Con2D(1,1)	Con2D(1,1)	128
	Con2D(3,3)	Con2D(3,1) Con2D(1,3)			Con2D(3,3)	Con2D(3,1) Con2D(1,3)	
	Con2D(5,5)	Con2D(3,1)			Con2D(5,5)	Con2D(1,3)	
Block 5	Con2D(7,7)	Con2D(1,3) Con2D(3,1)	256		Con2D(7,7)	Con2D(3,1) Con2D(1,3)	
	Con2D(1,1)	Con2D(1,1)			Con2D(1,1)	Con2D(1,1)	
	Con2D(3,3)	Con2D(3,1) Con2D(1,3)			Con2D(3,3)	Con2D(3,1) Con2D(1,3)	
	Con2D(5,5)	Con2D(1,3)			Con2D(5,5)	Con2D(1,3)	
	Con2D(7,7)	Con2D(3,1)			Con2D(7,7)	Con2D(3,1) Con2D(1,3)	
	Con2D(1,1)	Con2D(1,1)			Con2D(1,1)	Con2D(1,1)	



**Fig. 9.** The workflow of Dense CRF for EM image segmentation post-processing. The network with Dense CRF as post-processing is abbreviated as Net+, such as LCU-Net with Dense CRF as post-processing is abbreviated as LCU-Net+.

and the  $j$ -th pixel have the same label, the penalty term is equal to zero; otherwise, it is equal to one. This function is shown as Eq. (4).

$$\phi(x_i, x_j) = \begin{cases} 0 & x_i = x_j \\ 1 & x_i \neq x_j \end{cases} \quad (4)$$

As Eq. (3) shows, each  $k^{(m)}$  is the Gaussian kernel, which depends on the feature vectors  $\mathbf{f}_i$ ,  $\mathbf{f}_j$  of the  $i$ -th pixel and the  $j$ -th pixel, and is weighted by  $\omega^{(m)}$ . In [9],  $k(\mathbf{f}_i, \mathbf{f}_j)$  uses contrast-sensitive two-kernel potentials, defined in terms of the color vectors  $I_i$  and  $I_j$  and positions  $p_i$  and  $p_j$ . It is shown as Eq. (5).

$$k(\mathbf{f}_i, \mathbf{f}_j) = \underbrace{\omega_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right)}_{\text{appearance kernel}} + \underbrace{\omega_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right)}_{\text{smoothness kernel}} \quad (5)$$

The first appearance kernel depends on both pixel positions (denoted as  $p$ ) and pixel color intensities (denoted as  $I$ ). The second smoothness kernel only depends on pixel positions. The parameters  $\sigma_\alpha$ ,  $\sigma_\beta$  and  $\sigma_\gamma$  control the scale of Gaussian kernels. The first kernel forces pixels with similar color and position to have similar labels, while the second kernel only considers spatial proximity when enforcing smoothness [44].

## 4. Experiments

### 4.1. Experimental settings

#### 4.1.1. Image dataset

In our work, we use *Environmental Microorganism Dataset 5th Version* (EMDS-5), which is a newly released version of EMDS series [45], containing 21 classes of EMs. Each EM class contains 20 original microscopic images and their corresponding ground truth (GT) images. Since the microscopic images have multiple scales, we convert all the image sizes into  $256 \times 256$  pixels uniformly, as shown in Fig. 10.

#### 4.1.2. Training, validation and test data setting

We randomly divide each class of EMDS-5 into training, validation, and test data sets in a ratio of 1:1:2. Thus, we have 105 original images and their corresponding GT images for training and validation, respectively, and there are 210 original images for testing.

Furthermore, in the training process, data augmentation can effectively improve the lack of training images. Considering the method proposed in [30] and our pre-tests, we augment the 105 training images with rotations of 0, 90, 180, and 270 degrees, which result in  $105 \times 4 = 420$  images. After that, these rotated images are flipped over by the mirror, and  $420 \times 2 = 840$  images are obtained.

#### 4.1.3. Experimental environment

The experiment is conducted by Python 3.6.8 in Windows 10 operating system. The models we use in this paper are implemented by Keras 2.24 [46] framework with Tensorflow 1.12.0 [47] as the backend. Our experiment uses a workstation with Intel(R) Core(TM) i7-8700 CPU with 3.20GHz, 32GB RAM, and NVIDIA GEFORCE RTX 2080 8GB.

**Table 3**  
The definitions of evaluation metrics.

Metric	Definition	Metric	Definition
Dice	$\text{Dice} = \frac{2 \times  V_{\text{Pred}} \cap V_{\text{GT}} }{ V_{\text{Pred}}  +  V_{\text{GT}} }$	Jaccard	$\text{Jaccard} = \frac{ V_{\text{Pred}} \cap V_{\text{GT}} }{ V_{\text{Pred}} \cup V_{\text{GT}} }$
Precision	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$	Recall	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
Accuracy	$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$	VOE	$\text{VOE} = 1 - \frac{ V_{\text{Pred}} \cap V_{\text{GT}} }{ V_{\text{Pred}} \cup V_{\text{GT}} }$

### 4.1.4. Hyper parameters

The segmentation task is to predict the individual pixels whether they represent a point of interest (foreground EMs) or the background. Thus, the task can be seen as a pixel-level binary classification. Hence, as the loss function of the network, we take the binary cross-entropy function and minimize it [48]. The binary cross-entropy loss for the image is defined as Eq. (6).

$$L_1(X, Y, \hat{Y}) = - \sum_{i \in X} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (6)$$

In Eq. (6), for the image  $X$ ,  $Y$  is the corresponding GT image, and  $\hat{Y}$  represents the predicted segmentation result. For the  $i$ -th pixel in image  $X$ , the network predicts  $\hat{y}_i$ , whereas the GT value in the model is  $y_i$  [48].

For a batch with  $N$  images inside, the loss function  $J_1$  is defined by Eq. (7).

$$J_1 = \frac{1}{N} \sum_{i=1}^N L_1(X_i, Y_i, \hat{Y}_i) \quad (7)$$

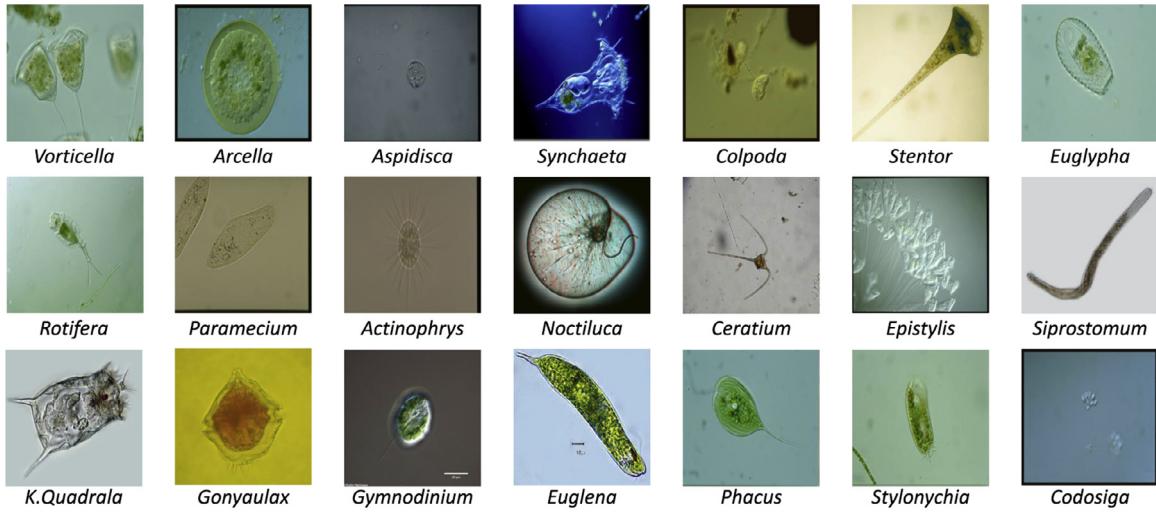
Besides, we use Adam optimizer, which can dynamically adjust the learning rate in the training process [49], with  $1.5 \times 10^{-4}$  learning rate and set the batch size to 2 in our training process. As the loss and accuracy curves of these models are shown in Fig. 11, after 40 iterations, the loss and accuracy curves level off. Therefore, considering the computational performance of the workstation, we finally set 50 epochs for training.

### 4.2. Evaluation metrics

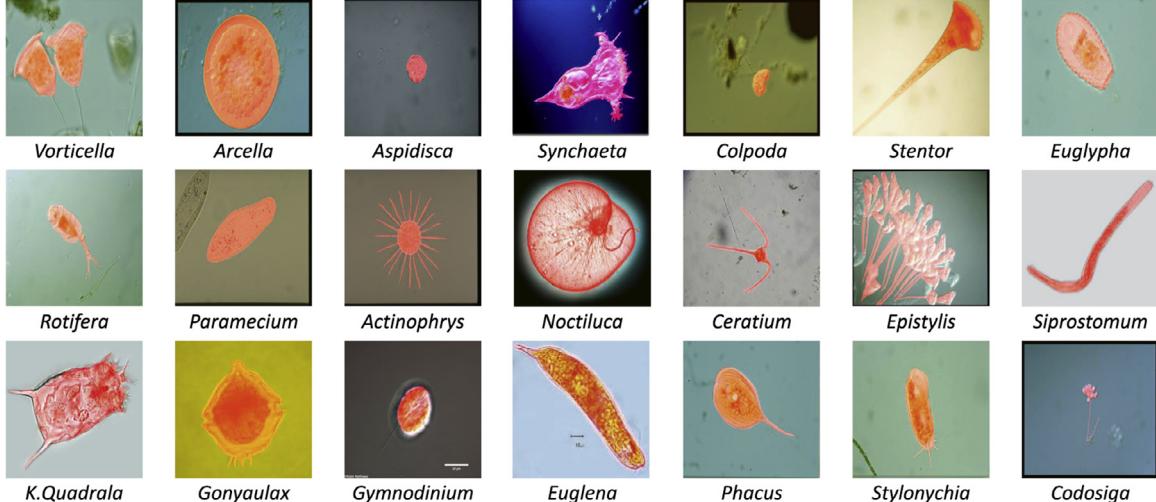
To compare the performance of various methods, different metrics are used to evaluate the segmentation results. In our previous work [2], Recall and Accuracy are used to measure the segmentation results. Besides that, we further employ Dice, Jaccard, Precision, and VOE (volumetric overlap error) to evaluate the segmentation results.

The Dice coefficient [50] is a widely used metric to evaluate segmentation performance. In addition to the comparison between predicted results and GT images, the Dice is usually used to measure reproducibility, and it is mathematically equivalent to F1-score [51]. The Jaccard [52], also named the IoU (intersection over union), is defined as the intersection between two sets divided by their union [51]. Recall measures the portion of positive pixels (foreground) in the ground truth images that are also identified as positive pixels by the segmentation results [51]. Nevertheless, we cannot judge the segmentation results only according to the value of Recall. The reason is well discussed in Section 4.3.2. Precision reflects the proportion of the real positive pixels among the positive pixels predicted by the model. Accuracy measures the sum of the portions of positive pixels (foreground) and negative pixels (background) in the GT images that are also identified as positive and negative in the segmentation images. VOE (volume overlap error) is the complement of the Jaccard coefficient [53].

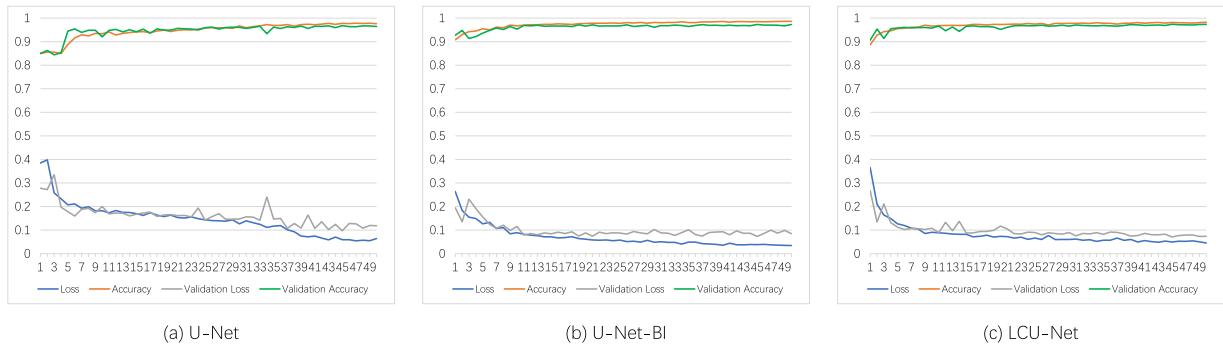
The definitions of these evaluation metrics are provided in Table 3.  $V_{\text{Pred}}$  represents the foreground predicted by the model.  $V_{\text{GT}}$  means the foreground in the GT image. TP (True Positive), FN (False Negative), FP (False Positive), and TN (True Negative) are concepts in the confusion matrix, and to facilitate understanding,



(a) Original Images

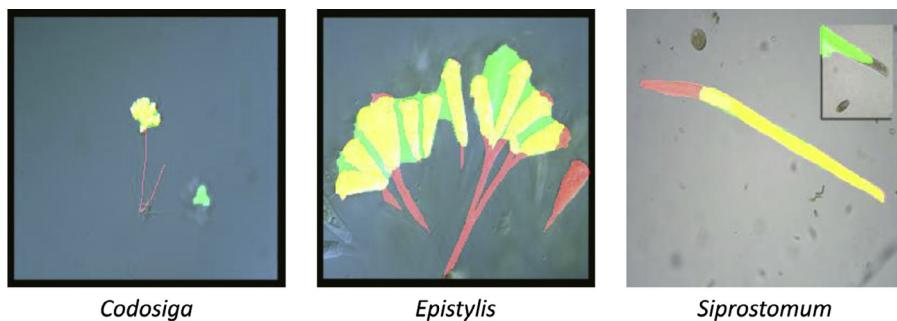


(b) Ground Truth Images

**Fig. 10.** Examples of the images in EMDS-5. (a) shows the original EM images and (b) provides the corresponding GT images.**Fig. 11.** The loss and accuracy curves of training process.

we provide a visual illustration in Fig. 12. The red and green masks represent the foregrounds of the GT images and LCU-Net+ segmentation results, respectively, and the yellow masks are the overlapping parts of them. Meanwhile, the red, green, and yellow masks also represent FN, FP, and TP in the confusion matrix, respectively, and the regions without any mask in the image represent TN.

From the metrics shown in Table 3, the higher the values of the first four metrics (Dice, Jaccard, Precision, Recall, and Accuracy) are, the better the segmentation results are. On the contrary, the lower the value of the final metric (VOE) is, the better the segmentation result is.



**Fig. 12.** The visual illustration on TP, FN, FP, and TN used in confusion matrix with examples of EM images.

**Table 4**

The memory and time costs of U-Net, U-Net-BI and LCU-Net.

Model	Memory Cost	Time Cost	
		Training Time	Average Test Time
U-Net	372,551,680 Byte (355 MB)	2145.38s (35.76min)	0.05s
U-Net-BI	426,868,736 Byte (407 MB)	4694.09s (78.24min)	0.13s
LCU-Net	108,560,384 Byte (103 MB)	2191.25s (36.52min)	0.15s

#### 4.3. Evaluation of segmentation methods

To prove the effectiveness of the proposed LCU-Net method for EM image segmentation, we compare its segmentation results with other classical and state-of-the-art methods mentioned in Section 2.

##### 4.3.1. Evaluation of different BLOCKs

In this part, we make a comparison between BLOCK-I and BLOCK-II. To this end, we carry out a series of experiments on U-Net, U-Net-BI, and LCU-Net.

*Evaluation of Memory and Time Costs* To compare the memory and time costs among U-Net, U-Net-BI, and LCU-Net, we provide the details in Table 4.

From Table 4, we can find that the memory cost of U-Net is 355 MB, the training time of U-Net is around 36 minutes for 840 EM images, and the average testing time is 0.05s for each test image. In contrast, the memory cost of LCU-Net is significantly reduced to 103 MB. From Table 4, we can also find that U-Net and LCU-Net have closed training time around 36 minutes. Furthermore, the proposed LCU-Net has only 0.1s test time longer than that of U-Net. Therefore, in contrast to U-Net, the LCU-Net saves 70.99% memory cost and has very close training time. Although the average test time is 0.1s longer than U-Net, the difference is acceptable in the practical EM segmentation work.

*Evaluation of Segmentation Performance* To compare the overall performance of the segmentation methods, we provide the average evaluation indices in Fig. 13.

From Fig. 13, we can find that U-Net achieves good performance, but the results of U-Net-BI and LCU-Net are even better. Especially, compared with U-Net, the average Dice value of LCU-Net is increased by around 1.5%; the average Jaccard value of LCU-Net makes 1.79% improvement; the improvement of the average Recall value made by LCU-Net is 5.14%; for the average Accuracy, the improvement of LCU-Net is 0.15%; the average VOE of LCU-Net is reduced by 2.32%. Hence, from these evaluation indices by LCU-Net, the overall segmentation performance is effectively improved. Although U-Net-BI makes slightly better segmentation performance than LCU-Net, the memory cost of LCU-Net is only about a quarter of that of U-Net-BI. Besides, the segmentation performance of LCU-Net is also better than the original U-Net.

To further improve the segmentation performance, we apply Dense CRF as the post-processing after obtaining the segmenta-

tion results of our networks. Dense CRF effectively improves the segmentation performance. We provide the average evaluation indices of U-Net+ (U-Net with Dense CRF as post-processing), U-Net-BI+ (U-Net-BI with Dense CRF as post-processing), and LCU-Net+ (LCU-Net with Dense CRF as post-processing) in Fig. 13. We can find that Dense CRF effectively assists these networks to further improve the segmentation performance from Fig. 13. Besides, U-Net-BI+ and LCU-Net+ still perform better than U-Net+.

After evaluating the overall performance of these methods, we also provide the detailed indices and examples of the segmentation results of each EM category under these methods in Table 5 and Fig. 14, respectively.

From Table 5 and Fig. 14, we can find that all the segmentation results of the improved methods can cover the main parts of the EMs. Besides, these improved methods have better EM detail acquisition capabilities than U-Net. The problems of under-segmentation and over-segmentation are much less in the segmentation results by these improved methods. Thus, because of the minimum memory cost, low time cost, and good segmentation performance with Dense CRF as the post-processing, we apply LCU-Net+ in the EM image segmentation task.

##### 4.3.2. Comparison with other methods

In this part, we conduct some comparative experiments on the task of EM image segmentation. We mainly adopt some representative segmentation methods mentioned in Section 2.1 and Section 2.2, including Otsu (threshold-based method), Canny (edge-based method), Watershed (region-based method), Region Growing (region-based method), MRF (unsupervised learning method),  $k$ -means (unsupervised learning method), and SegNet [54] (supervised learning method). Because the results are often insufficient, we need some post-processing for the results. For example, when we use edge-based methods to perform the segmentation task, it often needs post-processing, which includes dilation, erosion, and so on, to assist it [55]. To show better segmentation results of these methods, we uniformly used post-processing operations, which include dilation, erosion, flood-fill, and saving the largest region, to improve the segmentation results [55]. To evaluate these methods' overall performance, we provide the average evaluation indices of these methods in Fig. 15.

From the average indices in Fig. 15, we can find that the results of our methods are better than that of other methods. However, we can see that the Recall values in Fig. 15(d) are higher than

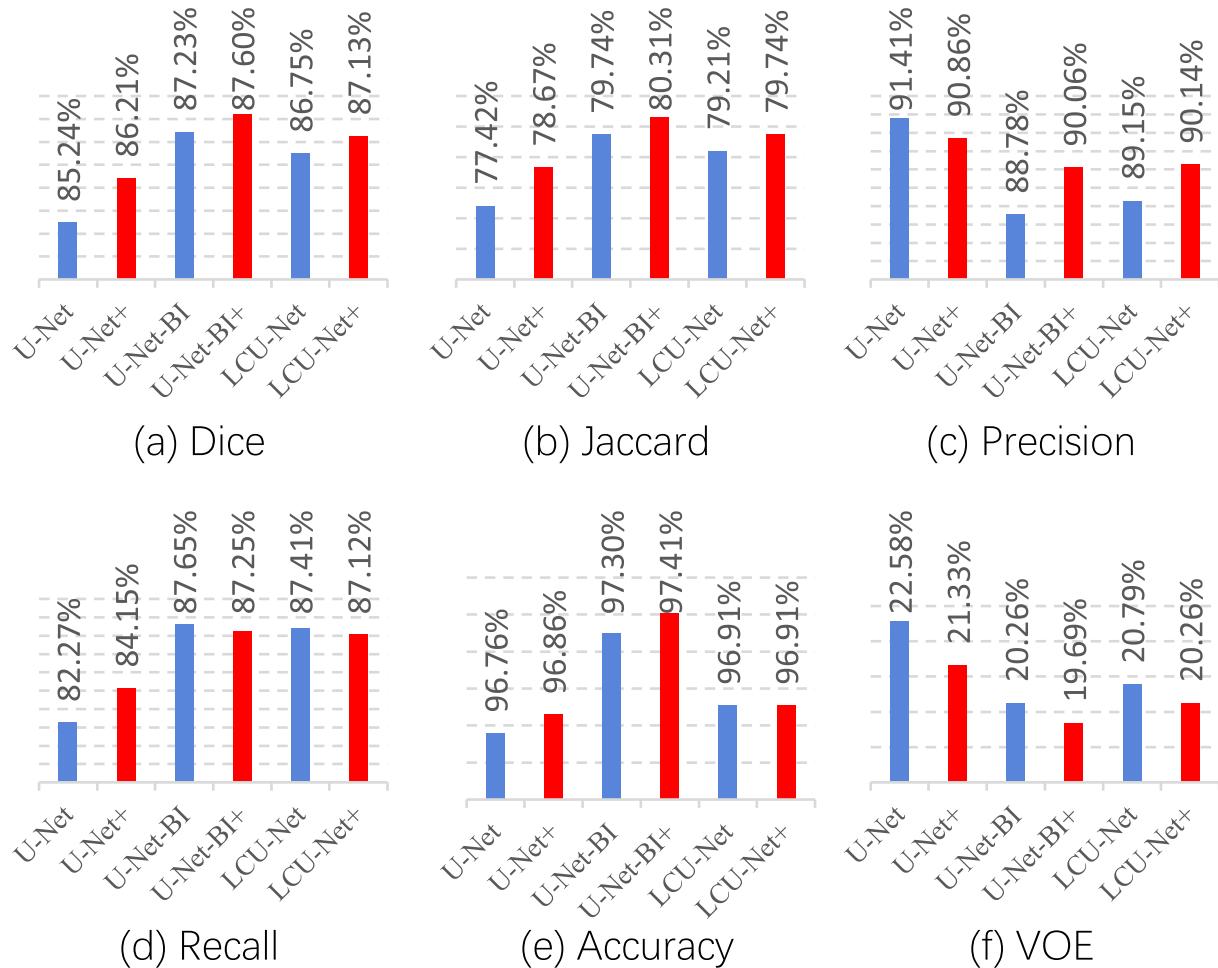
**Table 5**

The average segmentation indices for each EM category. For short, Region Growing, Dice, Jaccard, Precision, Recall, Accuracy, and VOE are abbreviated to RG, D, J, P, R, A, and V, respectively. (In %).

EM	Methods	Evaluation Metrics						EM	Methods	Evaluation Metrics						EM	Methods	Evaluation Metrics					
		D	J	P	R	A	V			D	J	P	R	A	V			D	J	P	R	A	V
		U-Net	71.80	57.47	95.19	59.13	97.53	42.53	U-Net	94.87	91.18	98.58	92.54	97.50	8.82	U-Net	94.06	88.86	96.24	92.19	99.70	11.14	
(a)	U-Net-BI	72.00	57.61	91.10	60.98	97.42	42.39	U-Net-BI	96.63	93.69	95.67	97.94	98.15	6.31	U-Net-BI	93.38	87.66	92.40	94.61	99.64	12.34		
	LCU-Net	72.17	57.82	92.49	60.76	97.50	42.18	LCU-Net	97.31	94.98	96.15	98.78	98.49	5.02	LCU-Net	94.28	89.29	95.32	93.44	99.71	10.71		
	U-Net+	71.83	57.46	95.30	59.27	97.52	42.54	U-Net+	95.19	91.73	98.87	92.77	97.66	8.27	U-Net+	93.97	88.74	94.88	93.41	99.69	11.26		
	U-Net-BI+	71.30	56.76	93.78	59.19	97.41	43.24	U-Net-BI+	97.47	95.24	96.91	98.28	98.63	4.76	U-Net-BI+	93.02	87.09	93.41	93.11	99.62	12.91		
	LCU-Net+	72.16	57.86	95.80	59.52	97.57	42.14	LCU-Net+	97.69	95.69	96.98	98.67	98.71	4.31	LCU-Net+	94.30	89.38	96.14	92.82	99.71	10.62		
	Otsu	31.12	24.88	32.60	81.96	41.41	75.12	(b)	Otsu	73.32	63.24	64.26	98.64	71.46	36.76	(c)	Otsu	4.49	2.31	2.31	89.49	14.31	97.69
	Canny	27.42	19.37	19.54	99.16	36.52	80.63		Canny	76.29	66.33	66.88	99.13	77.18	33.67		Canny	13.90	10.14	10.15	99.96	25.33	89.86
	Watershed	32.05	26.13	32.14	83.53	44.12	73.87		Watershed	67.02	58.11	62.61	85.35	74.54	41.89		Watershed	12.22	9.87	10.58	69.11	44.02	90.13
	k-means	14.73	10.85	13.04	67.74	40.37	89.15		k-means	52.41	47.40	48.51	68.88	70.18	52.60		k-means	4.38	2.25	2.25	90.00	11.97	97.75
	MRF	30.79	24.17	26.05	95.69	33.50	75.83		MRF	65.40	57.31	57.37	89.97	63.60	42.69		MRF	4.37	2.24	2.24	90.00	12.80	97.76
	RG	32.92	22.67	42.43	78.93	43.08	77.33		RG	87.22	81.51	82.97	98.52	85.31	18.49		RG	11.58	7.84	11.96	95.67	12.16	92.16
	SegNet	71.80	57.21	90.94	61.09	97.42	42.79		SegNet	96.11	92.96	96.41	96.49	97.92	7.04		SegNet	92.91	86.84	93.46	92.63	99.64	13.16
(d)	U-Net	48.83	38.24	62.33	44.24	96.64	61.76	U-Net	87.46	78.83	85.99	91.19	97.25	21.17	U-Net	55.43	40.56	66.41	50.04	89.00	59.44		
	U-Net-BI	58.27	44.86	65.14	56.97	97.17	55.14	U-Net-BI	84.30	74.51	79.46	94.14	97.62	25.49	U-Net-BI	68.38	52.90	69.14	70.82	90.67	47.10		
	LCU-Net	58.18	45.13	64.40	61.85	96.83	54.87	LCU-Net	80.44	70.59	78.79	90.35	97.03	29.41	LCU-Net	63.30	48.02	62.56	71.42	87.94	51.98		
	U-Net+	48.33	37.89	58.51	45.64	96.28	62.11	U-Net+	87.49	78.98	85.04	92.56	97.22	21.02	U-Net+	60.79	45.52	60.58	62.28	88.38	54.48		
	U-Net-BI+	58.16	44.90	65.49	57.02	97.12	55.10	U-Net-BI+	86.38	77.74	81.42	95.20	98.09	22.26	U-Net-BI+	69.29	53.89	67.94	72.69	90.74	46.11		
	LCU-Net+	59.20	46.29	64.21	60.44	96.68	53.71	LCU-Net+	82.28	72.93	79.88	91.74	97.22	27.07	LCU-Net+	64.01	48.94	58.86	76.74	87.11	51.06		
	Otsu	4.49	2.32	2.32	90.00	11.32	97.68	(e)	Otsu	31.09	24.68	27.66	96.45	29.63	75.32	(f)	Otsu	24.72	14.75	14.77	90.01	22.98	85.25
	Canny	7.08	3.76	3.83	96.78	17.31	96.24		Canny	38.84	33.16	33.21	99.91	50.24	66.84		Canny	36.87	23.56	23.67	99.25	39.81	76.44
	Watershed	7.11	4.17	11.19	62.70	41.28	95.83		Watershed	35.94	28.92	33.82	82.44	58.59	71.08		Watershed	30.75	19.25	22.36	81.74	39.34	80.75
	k-means	4.49	2.32	2.32	90.00	11.33	97.68		k-means	14.17	10.11	12.98	61.92	41.06	89.89		k-means	24.72	14.75	14.77	90.01	22.99	85.25
	MRF	4.67	2.41	2.41	90.00	13.17	97.59		MRF	27.95	22.31	22.64	96.18	30.01	77.69		MRF	23.95	14.40	14.41	80.01	32.32	85.60
	RG	5.55	2.88	2.88	100.00	2.88	97.12		RG	19.71	14.97	15.08	99.88	15.20	85.03		RG	31.78	19.71	21.66	95.85	24.91	80.29
	SegNet	56.59	44.05	62.00	56.77	96.48	55.95		SegNet	90.57	83.42	86.77	95.19	98.52	16.58		SegNet	61.32	46.01	55.87	73.80	86.30	53.99
(g)	U-Net	90.11	82.93	86.89	93.82	98.41	17.07	U-Net	92.77	87.46	92.05	93.74	97.84	12.54	U-Net	86.57	80.98	97.01	83.51	97.55	19.02		
	U-Net-BI	88.18	80.19	82.57	95.73	97.92	19.81	U-Net-BI	93.84	88.97	92.47	95.46	98.12	11.03	U-Net-BI	88.22	81.00	86.92	89.82	96.98	19.00		
	LCU-Net	88.43	80.20	81.22	98.08	98.06	19.80	LCU-Net	94.26	89.38	91.73	97.33	98.11	10.62	LCU-Net	88.87	81.90	86.46	91.67	96.93	18.10		
	U-Net+	89.87	82.56	85.07	95.79	98.28	17.44	U-Net+	92.99	87.83	91.77	94.41	97.88	12.17	U-Net+	90.37	84.30	97.29	86.66	97.96	15.70		
	U-Net-BI+	88.53	80.59	82.10	97.21	97.99	19.41	U-Net-BI+	94.05	89.39	93.38	94.90	98.19	10.61	U-Net-BI+	88.99	82.23	89.57	88.72	97.11	17.77		
	LCU-Net+	88.96	81.04	82.03	98.21	98.16	18.96	LCU-Net+	94.59	90.00	92.73	96.97	98.23	10.00	LCU-Net+	89.38	82.49	88.47	90.88	97.01	17.51		
	Otsu	47.39	41.79	43.36	86.30	66.37	58.21	(h)	Otsu	48.37	38.58	39.34	98.88	45.03	61.42	(i)	Otsu	70.77	62.37	68.10	91.17	82.17	37.63
	Canny	50.29	40.19	40.28	99.49	64.59	59.81		Canny	56.84	45.71	46.31	99.04	59.01	54.29		Canny	59.88	53.94	54.63	68.78	89.70	46.06
	Watershed	63.96	55.62	69.74	79.39	79.31	44.38		Watershed	64.85	55.60	69.83	85.37	71.92	44.40		Watershed	69.53	57.41	76.13	73.35	91.25	42.59
	k-means	5.90	3.46	4.22	26.19	73.90	96.54		k-means	14.27	8.53	8.53	50.00	50.66	91.47		k-means	70.40	63.18	71.23	79.46	88.32	36.82
	MRF	34.49	26.47	26.74	86.95	50.06	73.53		MRF	49.80	40.69	40.90	94.44	48.86	59.31		MRF	62.29	54.19	56.04	96.48	72.24	45.81
	RG	15.39	8.55	10.73	92.94	15.52	91.45		RG	40.59	29.67	31.55	97.73	33.01	70.33		RG	61.43	50.09	66.63	83.11	66.07	49.91
	SegNet	89.44	81.90	84.05	96.32	98.11	18.10		SegNet	92.58	86.66	91.45	94.02	97.60	13.34		SegNet	86.48	79.34	87.13	87.92	97.06	20.66
(j)	U-Net	91.36	84.30	88.07	95.13	98.97	15.70	U-Net	88.48	82.47	98.09	83.70	92.27	17.53	U-Net	83.32	73.21	92.84	76.63	96.91	26.79		
	U-Net-BI	90.49	82.89	85.56	96.26	98.56	17.11	U-Net-BI	94.96	90.66	97.07	93.27	96.08	9.34	U-Net-BI	81.56	72.48	87.23	80.01	96.73	27.52		
	LCU-Net	92.39	85.96	89.11	96.10	99.04	14.04	LCU-Net	90.90	84.57	98.02	86.44	93.20	15.43	LCU-Net	85.45	76.71	93.99	81.18	97.56	23.29		
	U-Net+	94.87	90.30	93.41	96.41	99.34	9.70	U-Net+	89.39	83.64	98.09	84.92	92.91	16.36	U-Net+	83.62	74.03	90.34	79.37	96.84	25.97		
	U-Net-BI+	93.97	88.79	92.71	95.52	99.02	11.21	U-Net-BI+	96.51	93.41	98.05	95.31	97.19	6.59	U-Net-BI+	81.60	72.11	89.23	79.57	96.65	27.89		
	LCU-Net+	94.71	90.01	93.98	95.52	99.34	9.99	LCU-Net+	92.08	86.35	98.78	87.49	93.95	13.65	LCU-Net+	86.78	77.63	96.70	80.49	97.53	22.37		
	Otsu	40.02	30.44	35.81	92.00	49.12																	

**Table 5 (continued)**

	EM	Methods	Evaluation Metrics						EM	Methods	Evaluation Metrics						EM	Methods	Evaluation Metrics					
			D	J	P	R	A	V			D	J	P	R	A	V			D	J	P	R	A	V
(m)	U-Net	88.76	80.63	95.40	84.50	97.25	19.37		U-Net	84.62	74.52	88.08	83.31	97.76	25.48		U-Net	93.06	87.23	95.23	91.36	97.38	12.77	
	U-Net-BI	91.02	83.78	91.55	91.03	97.67	16.22		U-Net-BI	81.60	72.46	82.87	80.83	97.44	27.54		U-Net-BI	93.07	87.34	92.84	93.66	97.46	12.66	
	LCU-Net	89.14	81.95	87.43	92.19	96.39	18.05		LCU-Net	84.86	75.70	88.70	83.73	97.96	24.30		LCU-Net	92.67	86.50	93.37	92.63	97.32	13.50	
	U-Net+	91.46	84.65	95.51	88.38	97.86	15.35		U-Net+	86.63	77.51	87.12	87.77	97.98	22.49		U-Net+	93.43	87.89	95.18	92.02	97.59	12.11	
	U-Net-BI+	92.23	85.80	93.23	91.60	98.02	14.20		U-Net-BI+	83.60	74.85	83.06	84.45	97.60	25.15		U-Net-BI+	92.96	87.26	93.96	92.26	97.46	12.74	
	LCU-Net+	89.87	83.24	88.76	92.61	96.44	16.76		LCU-Net+	85.68	76.69	89.49	84.58	98.06	23.31		LCU-Net+	93.12	87.25	95.14	91.62	97.49	12.75	
	Otsu	51.66	41.92	45.70	93.14	60.62	58.08	(n)	Otsu	50.54	40.88	41.58	96.80	66.80	59.12	(o)	Otsu	81.89	72.00	78.66	91.81	86.40	28.00	
	Canny	72.96	59.94	60.50	98.60	89.60	40.06		Canny	58.66	48.92	49.54	78.62	91.03	51.08		Canny	82.68	84.44	84.44	96.10	96.33	15.56	
	Watershed	58.66	45.73	62.14	66.26	86.71	54.27		Watershed	50.63	41.24	53.39	62.44	87.21	58.76		Watershed	86.87	77.41	88.06	87.17	95.14	22.59	
	k-means	32.19	24.88	26.46	66.89	63.76	75.12		k-means	43.26	35.96	37.04	67.48	81.26	64.04		k-means	53.16	45.87	53.93	61.56	77.84	54.13	
(l)	MRF	46.34	37.67	45.09	63.34	70.34	62.33		MRF	37.39	28.75	29.57	93.29	48.98	71.25		MRF	78.00	68.09	68.89	97.83	83.00	31.91	
	RG	51.27	39.86	45.21	92.88	54.15	60.14		RG	64.98	57.03	69.86	86.28	71.19	42.97		RG	58.39	46.76	56.30	90.00	56.66	53.24	
	SegNet	88.40	80.01	86.04	91.83	96.89	19.99		SegNet	79.92	71.45	76.32	85.24	96.59	28.55		SegNet	92.68	86.44	91.69	94.15	97.14	13.56	
	U-Net	89.80	82.23	97.22	84.41	97.59	17.77		U-Net	89.28	83.16	87.97	94.22	97.95	16.84		U-Net	93.08	87.27	98.93	88.16	95.07	12.73	
	U-Net-BI	92.89	87.08	97.77	88.93	98.27	12.92		U-Net-BI	87.21	80.41	82.19	97.77	97.95	19.59		U-Net-BI	95.23	90.96	97.91	92.84	96.47	9.04	
	LCU-Net	92.98	87.18	97.57	89.15	98.33	12.82		LCU-Net	86.19	80.22	83.02	96.35	97.34	19.78		LCU-Net	94.18	89.07	97.92	90.85	95.81	10.93	
	U-Net+	90.04	82.64	97.17	84.85	97.64	17.36		U-Net+	88.65	82.43	87.38	94.21	97.69	17.57		U-Net+	93.71	88.31	99.13	89.06	95.58	11.69	
	U-Net-BI+	91.90	85.43	97.76	87.18	98.06	14.57		U-Net-BI+	86.76	79.87	83.43	95.46	97.84	20.13		U-Net-BI+	94.89	90.35	98.51	91.69	96.48	9.65	
	LCU-Net+	92.40	86.18	97.74	87.93	98.22	13.82		LCU-Net+	84.94	78.99	82.92	95.22	96.79	21.01		LCU-Net+	94.12	88.99	98.57	90.18	95.82	11.01	
(p)	Otsu	58.89	48.05	57.65	77.64	75.02	51.95	(q)	Otsu	30.04	23.52	26.21	97.30	29.84	76.48	(r)	Otsu	73.25	63.89	70.33	93.31	68.78	36.11	
	Canny	65.46	54.89	56.50	97.17	77.66	45.11		Canny	68.81	60.51	61.93	88.19	87.95	39.49		Canny	93.34	87.71	95.55	91.57	95.46	12.29	
	Watershed	83.95	76.86	86.78	89.09	88.97	23.14		Watershed	75.71	70.45	75.08	84.80	91.40	29.55		Watershed	90.56	83.16	91.75	90.27	93.60	16.84	
	k-means	35.76	29.06	36.01	52.81	73.97	70.94		k-means	22.88	16.54	17.31	89.23	26.35	83.46		k-means	44.09	35.65	41.06	64.40	57.64	64.35	
	MRF	72.88	62.65	74.85	86.65	82.23	37.35		MRF	71.57	62.06	67.63	93.71	80.57	37.94		MRF	71.04	59.07	83.40	74.37	77.65	40.93	
	RG	66.66	54.98	70.07	83.61	73.16	45.02		RG	37.21	29.03	36.23	92.70	35.78	70.97		RG	62.75	48.82	60.65	87.78	56.71	51.18	
	SegNet	92.31	86.21	95.29	90.45	98.11	13.79		SegNet	83.97	77.62	80.59	96.35	96.64	22.38		SegNet	93.57	88.03	97.22	90.54	95.43	11.97	
(s)	U-Net	91.56	85.20	99.72	85.43	98.37	14.80		U-Net	80.01	68.72	97.60	70.06	93.57	31.28		U-Net	94.86	90.32	99.80	90.49	97.52	9.68	
	U-Net-BI	95.03	90.82	98.50	92.17	98.94	9.18		U-Net-BI	89.79	82.25	97.45	84.23	96.16	17.75		U-Net-BI	95.76	91.95	98.51	93.21	97.85	8.05	
	LCU-Net	92.20	86.37	98.76	87.50	98.20	13.63		LCU-Net	87.07	78.74	96.75	81.18	95.18	21.26		LCU-Net	96.44	93.18	98.44	94.59	98.19	6.82	
	U-Net+	91.61	85.18	99.71	85.40	98.35	14.82		U-Net+	81.43	70.60	97.82	71.89	93.90	29.40		U-Net+	94.64	89.93	99.81	90.09	97.46	10.07	
	U-Net-BI+	93.63	88.37	99.64	88.66	98.72	11.63		U-Net-BI+	89.79	82.30	98.73	83.22	96.17	17.70		U-Net-BI+	94.71	90.08	98.99	90.93	97.43	9.92	
	LCU-Net+	90.97	84.36	99.15	85.12	97.99	15.64		LCU-Net+	87.32	79.11	97.64	80.82	95.27	20.89		LCU-Net+	95.24	91.01	98.99	91.91	97.73	8.99	
	Otsu	40.28	33.47	40.85	81.99	48.48	66.53	(t)	Otsu	45.84	37.62	50.22	66.33	65.49	62.38	(u)	Otsu	83.78	75.20	84.21	89.96	86.09	24.80	
	Canny	57.15	50.79	53.33	97.16	61.44	49.21		Canny	65.65	55.66	59.43	95.10	74.53	44.34		Canny	94.61	89.95	92.67	96.90	97.29	10.05	
	Watershed	63.62	56.80	71.38	75.32	81.57	43.20		Watershed	53.94	41.84	75.68	64.57	73.45	58.16		Watershed	74.09	69.24	77.64	71.37	90.15	30.76	
	k-means	40.64	33.98	43.14	80.83	49.76	66.02		k-means	21.65	16.86	22.86	43.63	62.27	83.14		k-means	58.84	52.88	61.99	59.87	84.82	47.12	
	MRF	53.15	48.06	52.74	95.29	52.46	51.94		MRF	63.25	54.90	68.43	75.36	72.29	45.10		MRF	75.49	67.62	68.60	98.93	71.03	32.38	
	RG	70.00	59.52	76.51	82.98	74.94	40.48		RG	69.03	58.15	77.54	79.05	76.89	41.85		RG	58.44	46.41	49.23	97.17	48.64	53.59	
	SegNet	91.79	85.61	97.23	87.92	98.32	14.39		SegNet	86.82	78.32	97.58	79.91	95.13	21.68		SegNet	96.31	92.92	98.17	94.59	98.13	7.08	



**Fig. 13.** The evaluation indices of U-Net, U-Net+, U-Net-BI, U-Net-BI+, LCU-Net and LCU-Net+. The blue columns show the evaluation of segmentation results generated by the networks. The red columns show the evaluation of segmentation results generated by the networks with Dense CRF as post-processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the Recall values generated by U-Net, U-Net-BI, LCU-Net, and LCU-Net+. This is because some of the segmentation results by other methods have many background parts partitioned into the foreground [2]. What is more serious is that the whole picture is partitioned into the foreground. We can easily find this situation from Fig. 14. From the definition of Recall shown in Table 3, we can realize that as long as the foreground in the segmentation result contains the entire real foreground in GT, the value of Recall is 1 regardless of whether the over-segmentation problem is existing or not. Therefore, when we evaluate the segmentation results, we should not judge them by the value of Recall alone. From the above, we should consider multiple indices when we evaluate the segmentation results.

To observe the performance of these methods better, we provide not only the overall average indices in Fig. 15 but also the detailed indices of the segmentation results of each category of EMs under these methods in Table 5. Besides, we also provide examples of the segmentation results under these methods in Fig. 14.

#### 4.3.3. Comparison with local-Global CRF segmentation

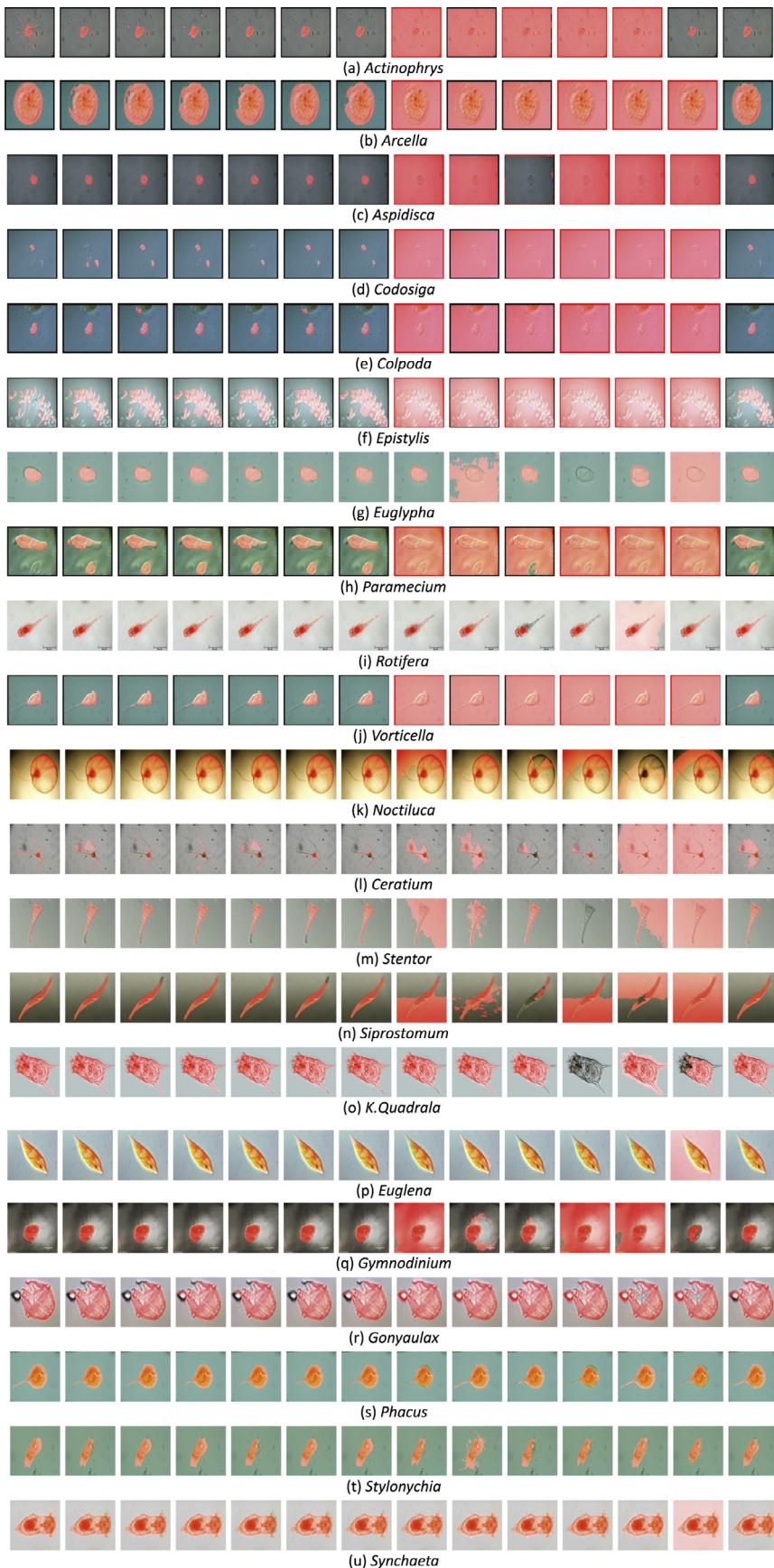
In our previous work of Local-Global CRF [2], we use the EMDS-4 data set with 20 categories of EMs. In contrast to EMDS-4, we have one more EM category (*Gymnodinium*) in EMDS-5 in this paper. Therefore, we evaluate the segmentation results obtained by LCU-Net+ without *Gymnodinium* here. Furthermore, there are six

models for segmentation in [2] to compare: Per-pixel RF (noEdges), CRF with Potts pairwise potentials (Potts), CRF with contrast-sensitive Potts model (PottsCS), fully connected CRF with Gaussian pairwise potentials (denseCRF), fully connected CRF on segmentation results by the original DeepLab method [44] (denseCRForg), fully convolutional network (FCN). Considering the evaluation metrics used in our previous work, we use Average Recall and Overall Accuracy to evaluate the performance of the segmentation results in [2]. The Average Recall and Overall Accuracy values of LCU-Net+ and our previous models are shown in Fig. 16.

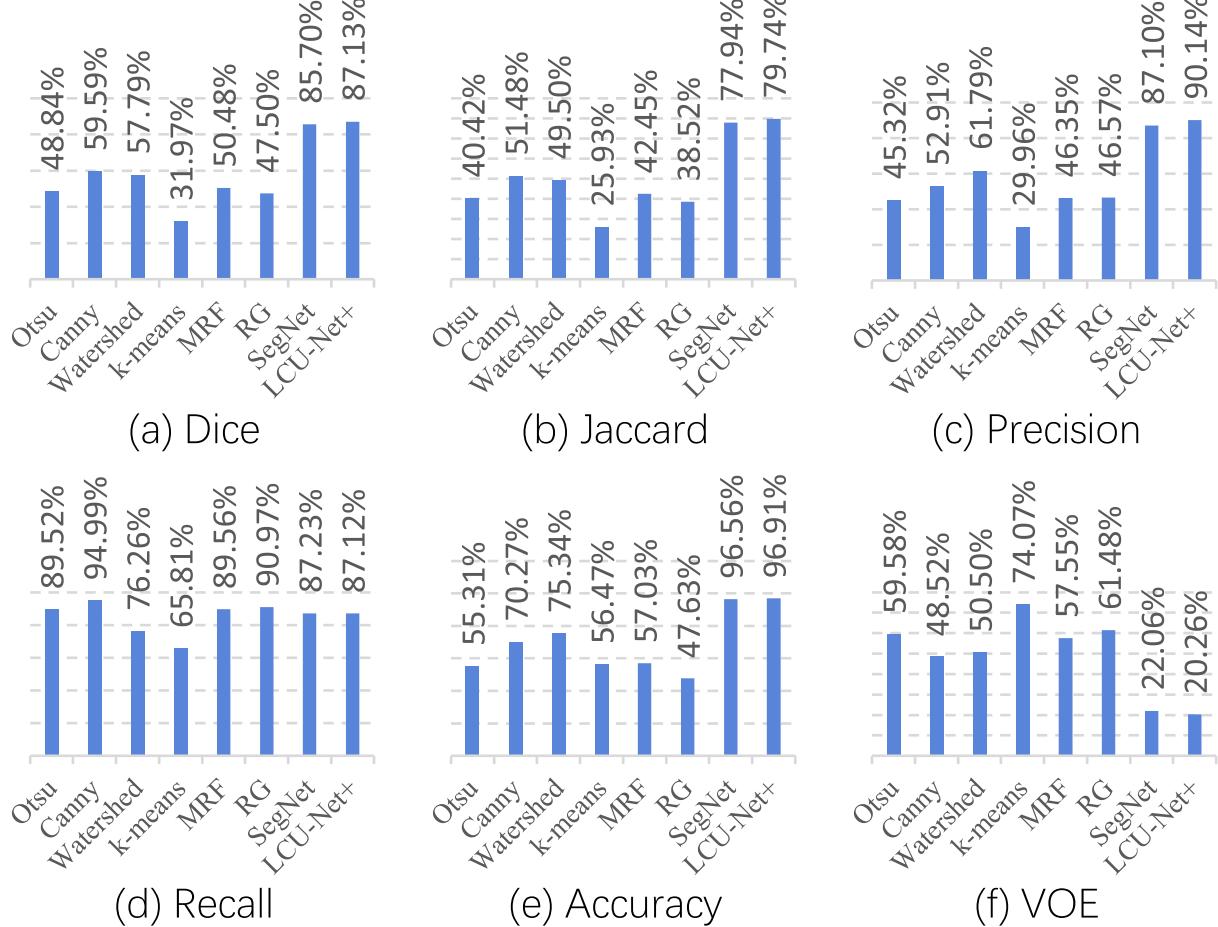
From Fig. 16, we can find that compared with our previous models, the Average Recall is improved by more than 7%, and the Overall Accuracy increases by at least 1%. From this, we can realize that the proposed LCU-Net+ in this paper performs better than the models in our previous work [2].

#### 4.3.4. Repeatability tests

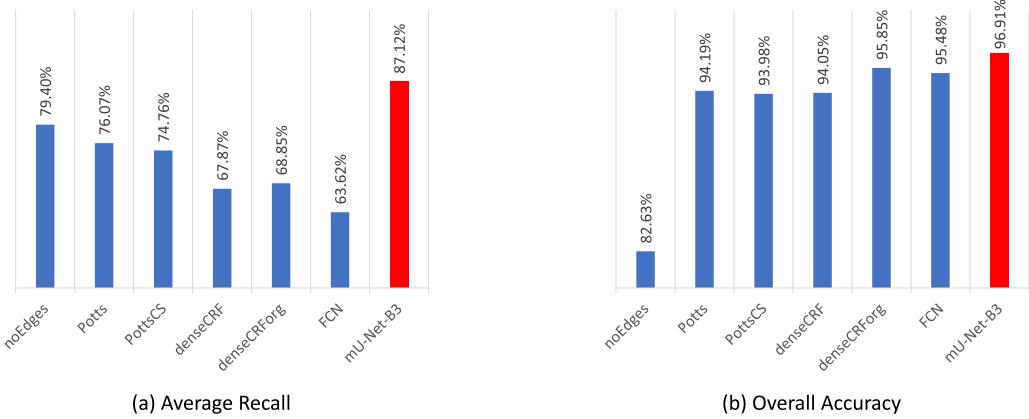
In the repeatability tests, we add four more repeatability LCU-Net+ experiments based on the original experiment. The overall evaluation indices of these segmentation results are provided in Table 6. From Table 6, we find that all five LCU-Net+ experiments perform better than the original U-Net and the evaluation indices of the five experiments are approximate. Therefore, it is proved that LCU-Net+ has good stability and performance on the EM image segmentation task.



**Fig. 14.** An example of GT and segmentation results for each EM category by U-Net, U-Net-BI, LCU-Net, U-Net+, U-Net-BI+, LCU-Net+, Otsu, Canny, Watershed, k-means, MRF, RegionGrowing and SegNet (from the left to the right). The red regions are GT and the segmentation results by different methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 15.** The average evaluation indices of Otsu, Canny, Watershed, *k*-means, MRF, RegionGrowing (RG), SegNet and LCU-Net+ based EM image segmentation.



**Fig. 16.** The Average Recall and Overall Accuracy of LCU-Net+ and Local-Global CRF in [2].

**Table 6**

The evaluation indices of Repeatability Tests. For short, Repetition is abbreviated to Re. (In %).

Model	Dice	Jaccard	Precision	Recall	Accuracy	VOE
LCU-Net+	87.13	79.74	90.14	87.12	96.91	20.26
LCU-Net+ (Re 1)	87.12	79.60	91.78	84.83	97.05	20.40
LCU-Net+ (Re 2)	87.15	79.94	88.97	87.96	97.00	20.06
LCU-Net+ (Re 3)	87.12	79.69	89.46	88.03	97.00	20.31
LCU-Net+ (Re 4)	87.27	79.95	88.90	88.51	96.94	20.05
U-Net	85.24	77.42	91.41	82.27	96.76	22.58

## 5. Conclusion and future work

In this paper, we propose LCU-Net for the EM image segmentation task. LCU-Net is a CNN based on U-Net, Inception, and concatenate operations. Furthermore, the Dense CRF is applied as post-processing to obtain global information to enhance the segmentation results (LCU-Net with Dense CRF as post-processing is abbreviated to LCU-Net+). The proposed method not only performs better than the original U-Net but also reduces the memory cost from 355 MB to 103 MB. In the evaluation of segmentation results by LCU-Net+, the values of evaluation indices Dice, Jaccard, Precision, Recall, Accuracy, and VOE (volume overlap error) are 87.13%, 79.74%, 90.14%, 87.12%, 96.91%, and 20.26%, respectively. Compared with U-Net, these indices are improved by 1.89%, 2.32%, 4.84%, and 0.14%, respectively, and Precision and VOE is decreased by 1.27% and 2.32%. Besides, compared with our previous Local-Global CRF model in [2], the performance of segmentation results is significantly improved.

In the future, we plan to increase the number of images in our data set and optimize the time cost of LCU-Net to improve segmentation performance. Meanwhile, we will also consider more new methods in our future work. For example, the strategy used in AdderNet [56] is considered to optimize our network, where instead of multiplication operation, addition operation is applied to reduce the computational complexity of CNNs. We also consider using the fuzzy binarization proposed in [57] to produce multiple image layers and use them to train separate deep learning models to improve segmentation performance. Besides, we can apply GAN to perform the data augmentation task [58].

## 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.

## Acknowledgements

We thank Prof. Beihai Zhou, Dr. Fangshu Ma from the University of Science and Technology Beijing, PR China, and Prof. Yanling Zou from Freiburg University, Germany, for their previous cooperation in this work. We thank Miss Zixian Li and Mr. Guoxian Li for their important discussion. We also thank B.E. Xuemin Zhu from Johns Hopkins University, US and B.E. Bolin Lu from Huazhong University of Science and Technology, PR China, for their careful work in the EMDS-5 image data preparation. Especially, Prof. Dr.-Ing. Chen Li works as the co-first author and corresponding author in this paper.

## References

- [1] R. Maier, I. Pepper, C. Gerba, *Environmental microbiology*, Academic Press, 2015.
- [2] S. Kosov, K. Shirahama, C. Li, et al., Environmental microorganism classification using conditional random fields and deep convolutional neural networks, *Pattern Recognit* 77 (2018) 248–261.
- [3] C. Li, K. Wang, N. Xu, A survey for the applications of content-based microscopic image analysis in microorganism classification domains, *Artif Intell Rev* 51 (4) (2019) 577–646.
- [4] T. Yamaguchi, S. Kawakami, M. Hatamoto, et al., In situ DNA-hybridization chain reaction (HCR): facilitated *in situ* HCR system for the detection of environmental microorganisms, *Environ. Microbiol.* 17 (2015) 2532–2541.
- [5] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2015, Springer, 2015, pp. 234–241.
- [6] S. Ghosh, A. Pal, S. Jaiswal, et al., Segfast-v2: semantic image segmentation with less parameters in deep learning for autonomous driving, *Int. J. Mach. Learn. Cybern.* 10 (11) (2019) 3145–3154.
- [7] C. Szegedy, V. Vanhoucke, S. Ioffe, et al., Rethinking the inception architecture for computer vision, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, IEEE, 2016, pp. 2818–2826.
- [8] B. Zhang, S. Qi, P. Monkam, et al., Ensemble learners of multiple deep CNNs for pulmonary nodules classification using CT images, *IEEE Access* 7 (2019) 110358–110371.
- [9] P. Krähenbühl, V. Koltun, Efficient inference in fully connected Crfs with Gaussian edge potentials, Proc. of 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011, Curran Associates Inc., 2011, pp. 109–117.
- [10] F. Kulwa, C. Li, X. Zhao, et al., A state-of-the-art survey for microorganism image segmentation methods and future potential, *IEEE Access* 7 (2019) 100243–100269.
- [11] X. Yang, H. Beyenal, G. Harkin, et al., Evaluation of biofilm image thresholding methods, *Water Res.* 35 (5) (2001) 1149–1158.
- [12] J. Yerly, Y. Hu, S. Jones, et al., A two-step procedure for automatic and accurate segmentation of volumetric CLSM biofilm images, *J. Microbiol. Methods* 70 (3) (2007) 424–433.
- [13] D. Rojas, L. Rueda, A. Ngom, et al., Image segmentation of biofilm structures using optimal multi-level thresholding, *Int J Data Min Bioinform* 5 (3) (2011) 266–286.
- [14] M. Khan, H. Nisar, C. Ng, et al., Local adaptive approach toward segmentation of microscopic images of activated sludge flocs, *J. Electron Imaging* 24 (6) (2015) 061102.
- [15] M. Khan, H. Nisar, N. Aun, et al., Iterative region based Otsu thresholding of bright-field microscopic images of activated sludge, Proc. of 2016 IEEE-EMBS Conference on Biomedical Engineering and Sciences, IECBES 2016, IEEE, 2016, pp. 533–538.
- [16] M. Dubuisson, A. Jain, M. Jain, Segmentation and classification of bacterial culture images, *J. Microbiol. Methods* 19 (4) (1994) 279–295.
- [17] M. Forero, G. Cristobal, J. Alvarez-Borrego, Automatic identification techniques of tuberculosis bacteria, Proc. of Applications of Digital Image Processing XXVI, SPIE 2003, SPIE, 2003, pp. 71–81.
- [18] M. DaneshPanah, B. Javidi, Segmentation of 3D holographic images using bivariate jointly distributed region snake, *Opt Express* 14 (12) (2006) 5143–5153.
- [19] E. Gutzeit, C. Scheel, T. Dolereit, et al., Contour based split and merge segmentation and pre-classification of zooplankton in very large images, Proc. of 9th International Conference on Computer Vision Theory and Applications, VISAPP 2014, SciTePress, 2014, pp. 417–424.
- [20] P. Hiremath, P. Bannigidad, M. Hiremath, Automated identification and classification of Rotavirus-A particles in digital microscopic images, Proc. of Nat. Conf. on Recent Trends in Image Processing and Pattern Recognition, RTIPPR 2010, IJCA, 2010, pp. 69–73.
- [21] F. Long, J. Zhou, H. Peng, Visualization and analysis of 3D microscopic images, *PLoS Comput. Biol.* 8 (6) (2012) e1002519.
- [22] M. Chayadevi, G. Raju, Automated colour segmentation of tuberculosis bacteria thru region growing: a novel approach, Proc. of 5th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2014, IEEE, 2014, pp. 154–159.
- [23] C. Xu, D. Zhou, T. Guan, et al., A segmentation algorithm for mycobacterium tuberculosis images based on automatic-marker watershed transform, Proc. of 2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014, IEEE, 2014, pp. 94–98.
- [24] M. Osman, M. Mashor, H. Jaafar, Performance comparison of clustering and thresholding algorithms for tuberculosis bacilli segmentation, Proc. of 2012 International Conference on Computer, Information and Telecommunication Systems, CITS 2012, IEEE, 2012, pp. 1–5.
- [25] M. Kemmler, B. Fröhlich, E. Rodner, et al., Segmentation of microorganism in complex environments, *Pattern Recognit Image Anal.* 23 (4) (2013) 512–517.
- [26] R. Rulaningtyas, A. Suksmono, T. Mengko, et al., Multi patch approach in k-means clustering method for color image segmentation in pulmonary tuberculosis identification, Proc. of 4th International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering, ICICI-BME 2015, IEEE, 2015, pp. 75–78.
- [27] C. Suzuki, J. Gomes, A. Falcao, et al., Automatic segmentation and classification of human intestinal parasites from microscopy images, *IEEE Trans. Biomed. Eng.* 60 (3) (2012) 803–812.
- [28] D. Nie, E. Shank, V. Jovic, A deep framework for bacterial image segmentation and classification, Proc. of 6th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB 2015, ACM, 2015, pp. 306–314.
- [29] K. Dannemiller, K. Ahmadi, E. Salari, A new method for the segmentation of algae images using retinex and support vector machine, Proc. of IEEE International Conference on Electro/Information Technology, EIT 2015, IEEE, 2015, pp. 361–364.
- [30] D. Matuszewski, I. Sintorn, Minimal annotation training for segmentation of microscopy images, Proc. of 15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, IEEE, 2018, pp. 387–390.
- [31] K. Bhargavi, S. Jyothi, A survey on threshold based segmentation technique in image processing, *International Journal of Innovative Research and Development* 3 (12) (2014) 234–239.
- [32] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans Syst Man Cybern* 9 (1) (1979) 62–66.
- [33] N. Senthilkumar, R. Rajesh, Edge detection techniques for image segmentation-A survey of soft computing approaches, *International Journal of Recent Trends in Engineering* 1 (2) (2009) 250–254.
- [34] J. Canny, A computational approach to edge detection, *IEEE Trans Pattern Anal Mach Intell* 8 (6) (1986) 679–698.

- [35] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, *IEEE Trans Pattern Anal Mach Intell* 13 (6) (1991) 583–598.
- [36] J. Hartigan, M. Wong, Algorithm AS 136: A K-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28 (1) (1979) 100–108.
- [37] S. Li, Markov random field models in computer vision, *Proc. of European conference on computer vision, ECCV 1994*, Springer, 1994, pp. 109–117.
- [38] M. Drozdzal, E. Vorontsov, G. Chartrand, et al., The importance of skip connections in biomedical image segmentation, *Deep Learning and Data Labeling for Medical Applications* (2016) 179–187.
- [39] C. Szegedy, W. Liu, Y. Jia, et al., Going deeper with convolutions, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, IEEE, 2015, pp. 1–9.
- [40] C. Li, J. Zhang, X. Zhao, et al., MRFU-Net: a multiple receptive field U-Net for environmental microorganism image segmentation, *Proc. of 8th International Conference on Information Technologies in Biomedicine, ITIB 2020*, Springer, 2020, pp. 27–40.
- [41] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, *Proc. of 32nd International Conference on Machine Learning, ICML 2015, IMLS, 2015*, pp. 448–456.
- [42] Y. Cao, Z. Wu, C. Shen, Estimating depth from monocular images as classification using deep fully convolutional residual networks, *IEEE Trans. Circuits Syst. Video Technol.* 28 (11) (2017) 3174–3182.
- [43] S. Zheng, S. Jayasumana, B. Romera-Paredes, et al., Conditional random fields as recurrent neural networks, *Proc. of 15th IEEE International Conference on Computer Vision, ICCV 2015*, IEEE, 2015, pp. 1529–1537.
- [44] L. Chen, G. Papandreou, I. Kokkinos, et al., Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE Trans Pattern Anal Mach Intell* 40 (4) (2017) 834–848.
- [45] Y. Zou, C. Li, K. Shirahama, et al., Environmental microorganism image retrieval using multiple colour channels fusion and particle swarm optimisation, *Proc. of 23rd IEEE International Conference on Image Processing, ICIP 2016*, IEEE, 2016, pp. 2475–2479.
- [46] A. Gulli, S. Pal, Deep learning with keras, Packt Publishing Ltd, 2017.
- [47] M. Abadi, P. Barham, J. Chen, et al., Tensorflow: a system for large-scale machine learning, *Proc. of 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, USENIX Association, 2016, pp. 265–283.
- [48] N. Ibtehaz, M. Rahman, Multiresunet: rethinking the U-Net architecture for multimodal biomedical image segmentation, *Neural Networks* 121 (2020) 74–87.
- [49] D. Kingma, J. Ba, Adam: a method for stochastic optimization, *Proc. of 3rd International Conference on Learning Representations, ICLR 2015, ICLR, 2015*, pp. 1–8.
- [50] L. Dice, Measures of the amount of ecologic association between species, *Ecology* 26 (3) (1945) 297–302.
- [51] A. Taha, A. Hanbury, Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool, *BMC Med Imaging* 15 (1) (2015) 29.
- [52] P. Jaccard, The distribution of the flora in the alpine zone. 1, *New Phytol.* 11 (2) (1912) 37–50.
- [53] P. Christ, F. Ettlinger, F. Grün, et al., Automatic liver and tumor segmentation of CT and MRI volumes using cascaded fully convolutional neural networks arXiv: 1702.05970, 2017.
- [54] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, *IEEE Trans Pattern Anal Mach Intell* 39 (12) (2017) 2481–2495.
- [55] R. Gonzalez, R. Woods, *Digital Image Processing*, Pearson Education, 2011.
- [56] H. Chen, Y. Wang, C. Xu, et al., AdderNet: do we really need multiplications in deep learning?, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, IEEE, 2020, pp. 1468–1477.
- [57] K. Santosh, L. Wendling, S. Antani, et al., Overlaid arrow detection for labeling regions of interest in biomedical images, *IEEE Intell Syst* 31 (3) (2016) 66–75.
- [58] H. Xu, C. Li, M.M. Rahaman, Y. Yao, Z. Li, J. Zhang, F. Kulwa, X. Zhao, S. Qi, Y. Teng, An enhanced framework of generative adversarial networks (EF-GANs) for environmental microorganism image augmentation with limited rotation-Invariant training data, *IEEE Access* 8 (2020) 187455–187469.



brane computing.

**Chen Li** received his B.E. degree from the University of Science and Technology Beijing, China in 2008, M.Sc. degree from the Northeast Normal University, China in 2011, and Dr.- Ing. degree from the University of Siegen, Germany in 2016. From 2016 to 2017, he worked as a postdoctoral researcher in the Johannes Gutenberg University Mainz, Germany. Currently, he is working as an associate professor in the Northeastern University, China. He is the head of the Research Group for Microscopic Image and Medical Image Analysis in the Northeastern University. His research interests are microscopic image analysis, medical image analysis, machine learning, pattern recognition, machine vision, multimedia retrieval and mem-



**Sergey Kovos** received his Diploma in Applied Mathematics from the Kirgiz-Russian Slavic University, Kirgystan in 2004, and M.Sc. degree in Computer Science from the Saarland University, Germany in 2008. From 2008 to 2013, he worked as a researcher in the Max Plank Institute for Informatics and Leibniz University, Germany. From 2013 he worked for industrial companies and in 2017 joined Pattern Recognition Group in University of Siegen as an external Ph.D. student. He received Dr.-Ing. degree in the University of Siegen, Germany in 2018. Currently he is a lecturer at Jacobs University, Germany. His research interests include classification with conditional random fields and deep neural networks, motion estimation with optical flow, 3-D reconstruction as well as ray tracing algorithms for realistic image synthesis.



**Marcin Grzegorzek** received his doctor of engineering degree in statistical pattern recognition from the University of Erlangen-Nürnberg in 2007, worked as postdoc for the Queen Mary University of London and the University of Koblenz-Landau, was assistant professor of pattern recognition at the University of Siegen. Currently, he is professor of medical informatics at the University of Lübeck. His scientific interests include medical data science, pattern recognition, machine learning and pervasive computing.



**Kimiaki Shirahama** received his B.E., M.E. and D.E degrees in Engineering from Kobe University, Japan in 2003, 2005 and 2011, respectively. After working as an assistant professor in Muroran Institute of Technology, Japan, he worked as a postdoctoral researcher at Pattern Recognition Group in University of Siegen, Germany from 2013 to 2018. Since 2018, he is working as an associate professor at Kindai University, Japan. His research interests include multimedia data processing, machine learning, data mining and sensor-based human activity recognition. He is a member of ACM SIGKDD, ACM SIGMM, the Institute of Image Information and Television Engineers in Japan (ITE), Information Processing Society of Japan (IPSJ) and the Institute of Electronics, Information and Communication Engineering in Japan (IEICE).



**Tao Jiang** received his Ph.D. degree from the University of Siegen, Germany, in 2013. He is currently a full professor with the Chengdu University of Information Technology (CUIIT), China. He is also the Dean with the Control Engineering College of CUIIT. His research interests include machine vision, artificial intelligence, robot control, self-driving auto, and membrane computing.



**Jinghua Zhang** was born in 1996. He received his B.E. degree from Hefei University, PR China, in 2018. He is currently pursuing the master's degree with the Research Group for Microscopic Image and Medical Image Analysis, Northeastern University, China. He will pursue the Ph.D. degree at National University of Defense Technology from the fall of 2021. His research interests are microorganism image analysis and deep learning.



**Changhao Sun** was born in 1995. He received his B.E. degree from the Northeastern University, China, in 2018. From 2018 till now, he is a master student in the Research Group for Microscopic Image and Medical Image Analysis in the Northeastern University, China. His research interests are microscopic image segmentation and deep learning.



**Hong Li** is currently an associate professor of the Northeastern University, China, and she is a young director of the Artificial Intelligence Branch of the Biomedical Engineering Society. She received her B.E. and M.Sc. degrees from Dalian University of Technology, China and received her Ph.D. degree from the Northeastern University, China. She also worked as an engineer in the Neusoft Medical System Co., Ltd. Her research areas include machine learning and application, neuroinformatics, biomedical imaging.



**Zihan Li** was born in 1995. He received his B.E. degree from the Northeastern University, China, in 2018. From 2018 till now, he is a master student in the Research Group for Microscopic Image and Medical Image Analysis in the Northeastern University, China. His research interests are microorganism image retrieval and deep learning.