

Automated yeast cells segmentation and counting using a parallel U-Net based two-stage framework

YAN KONG,¹ HUI LI,² YONGYONG REN,¹ GEORGI Z. GENCHEV,^{1,4,5}
XIAOLEI WANG,¹ HONGYU ZHAO,^{1,3,6} ZHIPING XIE,^{2,7} AND HUI
LU^{1,4,8}

¹SJTU-Yale Joint Center for Biostatistics, Shanghai Jiao Tong University, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, China

²State Key Laboratory of Microbial Metabolism, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, China

³Department of Biostatistics, Yale University, New Haven, CT 06520, USA

⁴Center for Biomedical Informatics, Shanghai Children's Hospital, Shanghai 200240, China

⁵Bulgarian Institute for Genomics and Precision Medicine, Sofia 1000, Bulgaria

⁶hongyu.zhao@yale.edu

⁷zxie@sjtu.edu.cn

⁸huilu@sjtu.edu.cn

Abstract: Yeast fluorescence microscopic images are widely used to observe the living conditions and survival of yeast cells under experimental conditions. Accurate cell counting provides key quantitative feedback and plays key roles in biological research as well as in industrial and biomedical applications. Unfortunately, the commonly used manual counting method is time-intensive, poorly standardized, and non-reproducible. Here, we developed a two-stage framework using parallel modified U-Nets together with seed guided water-mesh algorithm for automatic segmentation and yeast cells counting. The proposed framework was tested with independent images, of which the ground truth of yeast cell number and locations was done by skilled technicians. Our method improved cell counting by reducing bias and demonstrated a 99.35% consistent recall rate of experienced manual counting, and decreased the time required from 5 minutes on average to only 5 seconds for each image.

© 2020 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Cell counts are an important indicator in cell function studies, such as cell cycle regulation, growth monitoring, and cell proliferation. However, establishing cell counts is often a time consuming and labor-intensive task. Traditionally, laboratories rely on mechanical techniques such as flow cytometry, to count cells; cell numbers can also be roughly estimated by using the specific light absorption or gradient dilution values of cells. In addition to mechanical methods, imaging technology has been widely used in modern cell counting studies. One typical imaging method to estimate cell numbers is the density-based method. Herein cells are manually counted on a local grid image view with the help of image processing software programs such as imageJ [1]; this step is subsequently repeated in different selected grids and the results are combined to calculate an average cell density value which is used to estimate the total number of cells.

Such manual methods can be beset by shortcomings such as poor sensitivity and specificity, low reproducibility, and a time-consuming process of detection and cell enumeration. This fact underscores the urgent need to develop and refine rapid and automated cell counting methods. Some automated workflows and software tools to perform the cell counting task are already in existence, but they can be costly, closed-source, and may not adequately address quantifying

error issues [2]. One straightforward approach for determining cell counts is to develop an object detection and segmentation model, which incorporates key determining characteristic combinations of morphological features such as cell size, color value, and cell spacing [3]. Since morphology characteristics may not account for the entire image and some of them such as shape vary widely across the same cells, it can be difficult to accurately detect and segment by using traditional machine learning methods.

Object detection and segmentation have been an import research focus in the computer vision field and many popular algorithms have been proposed in recent years such as Fast R-CNN [4], YOLO [5,6], and U-Net [7,8]. While biological image analysis has entered the era of artificial intelligence with the utilization of approaches such as computer vision methods and deep convolutional networks [9], because segmenting small objects is a notoriously difficult problem, few studies focus on the segmentation of objects at the cellular level [10–12].

In this study we developed an open-source two-stage framework to automate the process of cell counting. Our framework is based on a parallel modified U-Net, which has previously shown promising results when applied to the tasks of images segmentation [13], in combination with the seed guided watershed [14] method. We also compared our method with ImageJ, which has a lot of functional toolboxes. When applied to the independent verification image set, our method achieved improved segmentation accuracy results and improved convenience of use. We demonstrate that our proposed approach leads to a state-of-art performance and results in automated yeast cells counting in a fluorescent image.

2. Image collection, annotation, and preparation

2.1. Image collection

Fluorescent cell images used in this study were captured by Olympus IX83 (Olympus Life Sciences, Tokyo, Japan) inverted microscope. We manually selected 20 raw DIC (differential interference contrast) images which contained a number of yeast cells. Cell region and cell central region annotations of the yeast cells were manually done by experienced laboratory technicians.

2.2. Image annotation

Using the *OpenCV* Python library [15] and the DIC images, regions encompassing yeast cells were marked in black while other regions were marked in white. Then, from each raw DIC image, two annotation images were obtained. First, masked annotation images were obtained by setting the gray values of the white areas to grayscale = 1 by binarization method [16] while keeping the black regions grayscale = 0. In addition, center annotation images were obtained by manually marking the 10 pixels around the center of each yeast cell as grayscale value=1 and the other pixels as grayscale value = 0. At the end of the step, we obtained 40 annotated images: 20 of which were yeast region (masked annotation) images, and the remaining 20 were central region annotated (center annotation) images of yeast cells.

2.3. Image dataset preparation

The training set and the validation set were built by randomly selecting 10 raw DIC, and corresponding 10 masked annotated and 10 central region annotated images. In order to reduce the memory requirements of the training step, both the raw and the two corresponding annotation images were split into image tiles of size 224 by 224 pixels with a step stride of 65 and 33 pixels for the horizontal and vertical direction, respectively. After this, we got 4360 raw image tiles and the corresponding center annotation and masked annotation image tiles. From this set, 3310 tile triples (raw DIC image tiles, center annotation image tiles, masked annotation image tiles) were randomly selected as the training data set, and the remainder were considered as the validation set. All image processing was performed by using the *OpenCV* Python library.

When we use our framework (Fig. 1, white boxes) to count cells, the raw DIC fluorescent images are first transformed into a suitable size by splitting them into smaller images (tiles) with dimension of 224×224 pixels. This is used as the constant input image size in our proposed modified U-Net. It's worth noticing that there is a 100 pixels overlap between adjacent image tiles after the image splitting and the predicted masked images discards the 50 pixel edge of all images except in the boundary cases. The predicted annotated image is pieced together into a complete image of the same size as the re-sized raw image. The raw DIC images are annotated and clearly separated into two regions, the yeast cell or cell centers region, and the background region. Taking the annotated image of the yeast cell areas as a base, the cells were counted automatically using the method described in section 3.2.

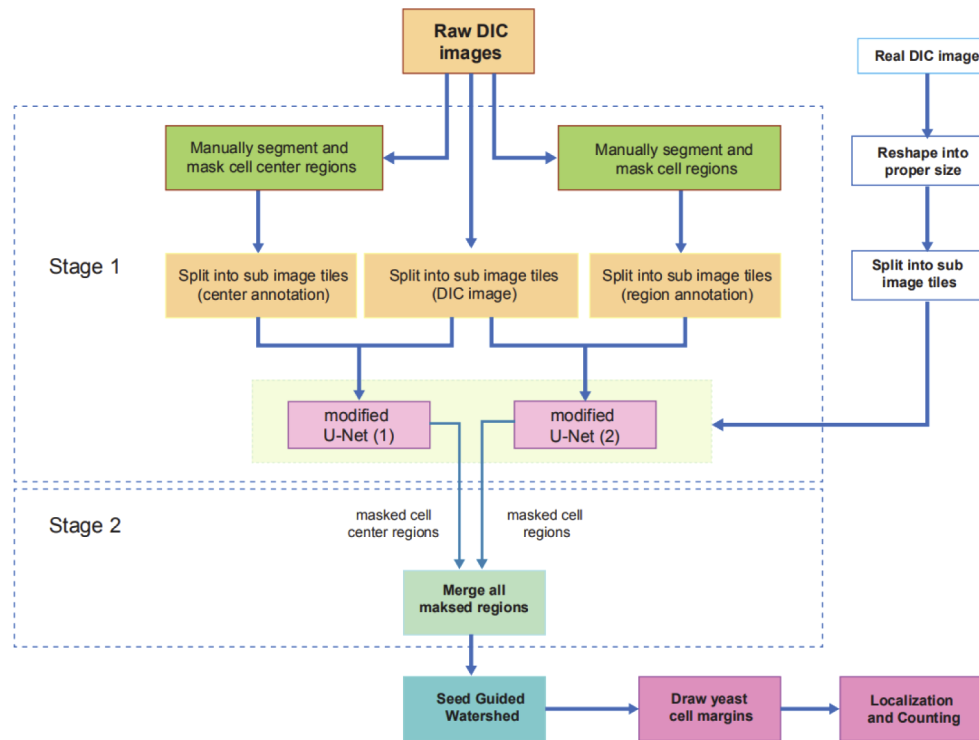


Fig. 1. The flowchart of the proposed two-stage method Right-hand side white blocks: pre-processing of the new DIC images when performing cell counting method (described in detail in section 2.3) Top dotted line- bound box: first stage -establishing a parallel modified N-Net (described in detail in section 3.1). Bottom dotted line-bound box: second stage-implementation of seed guided watershed algorithm (described in detail in section 3.2)

The raw test DIC images used in this study were sized $\sim 1002 \times 1998$ pixels; before feeding them into the algorithm, they were resized into 1092×2084 pixels so that each DIC image could be split into a grid of 8×16 image blocks. The image blocks were then fed into our modified U-Net.

3. Methodology

In this paper, a two-stage framework is proposed to automate yeast segmentation and cell counting. The flow of the proposed two-stage framework is shown in Fig. 1. The first stage consists of establishing a parallel modified U-Net (first dotted line- bound box display on the figure), and

the second stage consists of implementation of seed guided watershed algorithm (second dotted line-bound box is the figure). In this section, we present our proposed method in detail.

3.1. Parallel modified U-Net structure (stage 1)

The first step in the proposed segmentation approach is predicting the position information of each yeast cell using a convolutional neural network, thus the first stage of our proposed two-stage approach is building parallel deep neural convolution networks and training them. A suitable algorithm for the task - U-Net is a convolutional neural network and is well-known for fast and precise segmentation of biomedical images [7,17]; thus we begin by using a parallel modified U-Net to segment the yeast cells. Herein, a parallel modified U-Net was trained to predict yeast cell mask segmentation and yeast cell center segmentation respectively. To obtain additional characteristics, we modified the traditional U-Net for the yeast cell segmentation task by adding more convolutional layers in the down-sampling part - our modified U-Net consisted of 27 convolutional layers, (the full architecture of the network is displayed in Fig. 2 and details were displayed in Table 1). The VGG-16 network [18], which consisted of 13 convolutional neural network layers (and ReLU non-linearity) and 3 fully connected layers, was used in the down-sampling part, the last full connection layers were ignored, and followed by 4 de-convolution layers in the up-sampling stage. We then applied a softmax activation to the output layer to obtain the class-wise probabilities for each pixel. The output annotated images

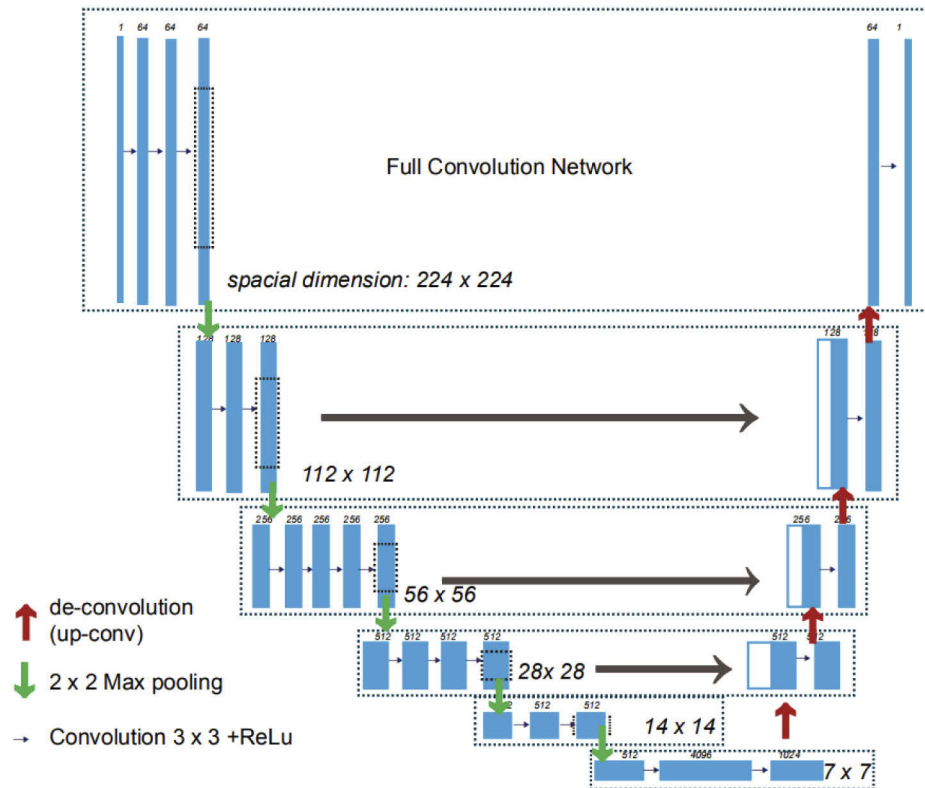


Fig. 2. Modified U-Net used architecture in this study. The down-sampling part was the same as VGG16 network but excluded the final full connection layer. The up-sampling layers were made up of 4 de-convolution layers. The size of feature maps of every layer is marked in the dotted box.

Table 1. The detailed architecture of proposed Modified U-Net model

layer	Feature Map	Size	Kernel size	Stride	Activation
Input	1	$224 \times 224 \times 3$	-	-	-
Convolution	64	$224 \times 224 \times 64$	3×3	2	relu
Convolution	64	$224 \times 224 \times 64$	3×3	2	relu
Convolution	64	$224 \times 224 \times 64$	3×3	2	relu
Max pooling	128	$112 \times 112 \times 128$	2×2	2	relu
Convolution	128	$112 \times 112 \times 128$	3×3	2	relu
Convolution	128	$112 \times 112 \times 128$	3×3	2	relu
Max pooling	256	$56 \times 56 \times 256$	2×2	2	relu
Convolution	256	$56 \times 56 \times 256$	3×3	2	relu
Convolution	256	$56 \times 56 \times 256$	3×3	2	relu
Convolution	256	$56 \times 56 \times 256$	3×3	2	relu
Convolution	256	$56 \times 56 \times 256$	3×3	2	relu
Max pooling	512	$28 \times 28 \times 512$	2×2	2	relu
Convolution	512	$28 \times 28 \times 512$	3×3	2	relu
Convolution	512	$28 \times 28 \times 512$	3×3	2	relu
Convolution	512	$28 \times 28 \times 512$	3×3	2	relu
Max pooling	512	$14 \times 14 \times 512$	2×2	2	relu
Convolution	512	$14 \times 14 \times 512$	3×3	2	relu
Convolution	512	$14 \times 14 \times 512$	3×3	2	relu
Max pooling	512	$7 \times 7 \times 512$	2×2	2	relu
Convolution	4096	$7 \times 7 \times 4096$	3×3	2	relu
Convolution	1024	$7 \times 7 \times 1024$	3×3	2	relu
De-convolution	512	$28 \times 28 \times 512$	3×3	2	relu
concatenate	512	$28 \times 28 \times 512$	-	2	-
Convolution	512	$28 \times 28 \times 512$	3×3	2	relu
De-convolution	256	$56 \times 56 \times 256$	3×3	2	relu
concatenate	256	$56 \times 56 \times 256$	-	2	-
Convolution	256	$56 \times 56 \times 256$	3×3	2	relu
De-convolution	128	$112 \times 112 \times 128$	3×3	2	relu
concatenate	128	$112 \times 112 \times 128$	-	2	-
Convolution	128	$112 \times 112 \times 128$	3×3	2	relu
De-convolution	64	$224 \times 224 \times 64$	3×3	2	relu
Convolution	64	$224 \times 224 \times 64$	3×3	2	relu
output	1	$224 \times 224 \times 1$	-	-	softmax

were the same size as the input images. Our U-Net included many layer weights to be optimized by the learning process. We used transfer learning [19] in the down-sampling part, in which the model variables were initialized by a pre-trained model (which was trained for ImageNet [20]), and then fine-tuned with our training data-set.

3.2. Outlining yeast cells with seeded watershed

Following the segmentation of yeast cells regions and center regions, the seed guided watershed algorithm was applied to find margins of yeast cells and separate each yeast cell. This is the second stage of our proposed method.

The watershed method [14,21,22], which segments and labels an image using a landscape analogy, is popular in the field of image segmentation. Studies have reported excellent research results in object detection and image segmentation using the method of combining watershed transform and deep neural networks [10,23]. Such segmentation shows improved performance if some foreground regions and background locations can be specified ahead of time, this notion underlies the theoretical basis of seeded (marker-controlled) watershed.

In stage 1 the parallel deep neural convolution network (U-net) returns both yeast cell region annotation and cell center annotation. Those cell center annotations serve as the seed regions in stage 2. In stage 2, seed watershed was implemented by running the skimage.feature package [24] and the skeletons of the yeast cells were obtained. Finally, we considered closed contours with areas larger than 5 pixels as yeast cells and labeled them with numbers.

3.3. Evaluation metric

We evaluate our method by using the metrics of precision, recall, F1 score [25], and miss rate (MR), which are defined as follows:

$$precision = \frac{TP}{TP+FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1_score = \frac{2TP}{2TP + FN + FP}$$

$$MR = \frac{FN}{TP + FN}$$

where TP, FP, and FN denote true positive, false positive, and false negative, respectively.

Aggregated Jaccard Index (AJI) [26] was also used as additional evaluation method. The AJI value is commonly used as an object-level evaluation metric, and is an extension of the Jaccard Index which is defined as:

$$AJI = \frac{\sum_{i=1}^K |GT_i \cap PRD_j^*(i)|}{\sum_{i=1}^K |GT_i \cap PRD_j^*(i)| + \sum_{l \in U} |PRD_l|}$$

where GT_i ($i=1,2,\dots,K$) is a pixel belonging to ground truth yeast cell region annotations, and PRD_j ($j=1,2,\dots,L$) is a pixel belonging to predicted annotations. $PRD_j^*(i)$ is the connected component object of the predicted annotations that have the maximum Jaccard Index with the ground truth annotations, and U is the union of predicted annotation objects that does not have intersections with any ground truth.

4. Results

4.1. Network implementation details

Both modified U-Net in the proposed framework were trained using the same strategy, and SGD (stochastic gradient descent) was used as an optimizer to minimize the loss function. The input patch size of both deep neural networks was 224×224 pixels. Networks of cell region segmentation and cell center segmentation were trained with batch size of 2 and 5, and consumed 8h and 5h on two NVIDIA Tesla P100 GPU cells for 50 epoches, respectively. Additionally, we employed various augmentation techniques during the training process such as image rotation, rotation, and horizontal flipping. All the experiments were conducted using Tensorflow [27].

4.2. Results and comparative analysis

In order to prove the effectiveness of our proposed two-stage framework (PM-UNet & seeded watershed), we built another deep learning-based pipeline for comparison. The comparison framework (SM-UNet & watershed) contained one modified U-Net, the same architecture as described in section 3.1, which was used to predict yeast cell regions, and the watershed algorithm which was used to segment cells. We also used *ImageJ* to count yeast cells as a comparison. Because using *imageJ* to automate the counting of cells involves very complex morphological changes, such as adjusting brightness and contrast, to simplify the comparison process, we directly used the images after background segmentation as the input file. We calculated the evaluation indicators mentioned in section 3.3 but skipped the AJI index for the ImageJ method.

All the 10 test DIC images contain 1562 complete yeast cells not including the cell whose body is outside the view of the DIC images. We segmented and counted yeast cells with three method and compared their performances. Our method was able to segment not only yeast cells with complete boundaries but also those with incomplete boundaries while the comparison two methods could not do. Our proposed two-stage framework showed good performance, achieving average Recall rate of 99.35%, AJI of 94.17%, F1-score of 99.54%, and MR of 0.65% on the test DIC images (Table 2). Detailed evaluation results can be found in Dataset 1 [28]. These evaluations values underscore the reliability of our method. The higher the AJI indicator, the better we isolate the yeast cell region from the background on the pixel level, which will prove the credibility of the area of yeast cells we have segmented. In addition, for deep learning methods, the images are feed into the program once and results are returned without additional operations, while *imageJ* software requires more manual involvement. Both our proposed deep learning-based method consumed less than 5 seconds for single DIC image while experienced lab members would take an average of 5 minutes at least.

Table 2. The performances of different methods of yeast cell counting on the test images

Method	Precision (%)	Recall (%)	F1-score (%)	MR (%)	AJI (%)
ImageJ	96.32	96.83	96.53	3.17	-
SM-UNet & watershed ^a	99.75	97.52	98.62	2.48	90.19
PM-UNet & watershed (proposed) ^b	99.74	99.35	99.54	0.65	94.17

^aSM-UNet & watershed: single modified U-Net together with watershed method

^bPM-UNet & watershed: our proposed two-stage frame, parallel modified U-Net together with seeded watershed method

We provided the segment and count results of the 10 test images with our proposed method in Fig. 3 and we also selected two image parts as examples to compare the results of different methods(Fig. 4).

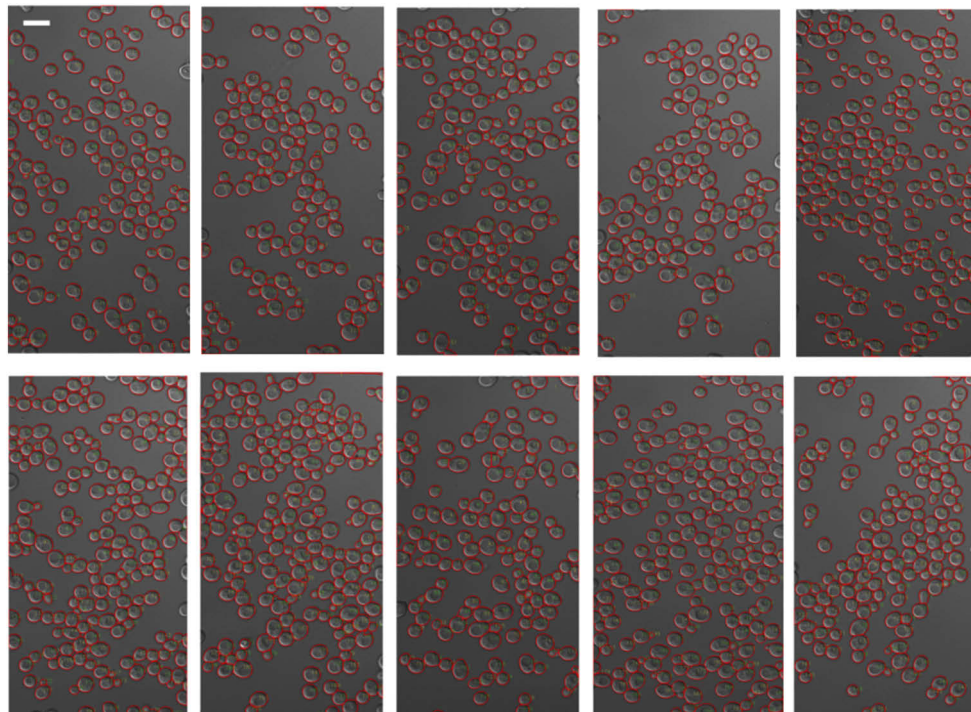


Fig. 3. Final results of 10 independent validation DIC images with the proposed method. Yeast cells detected were segmented and numbered. Scale bar: 10 μm .

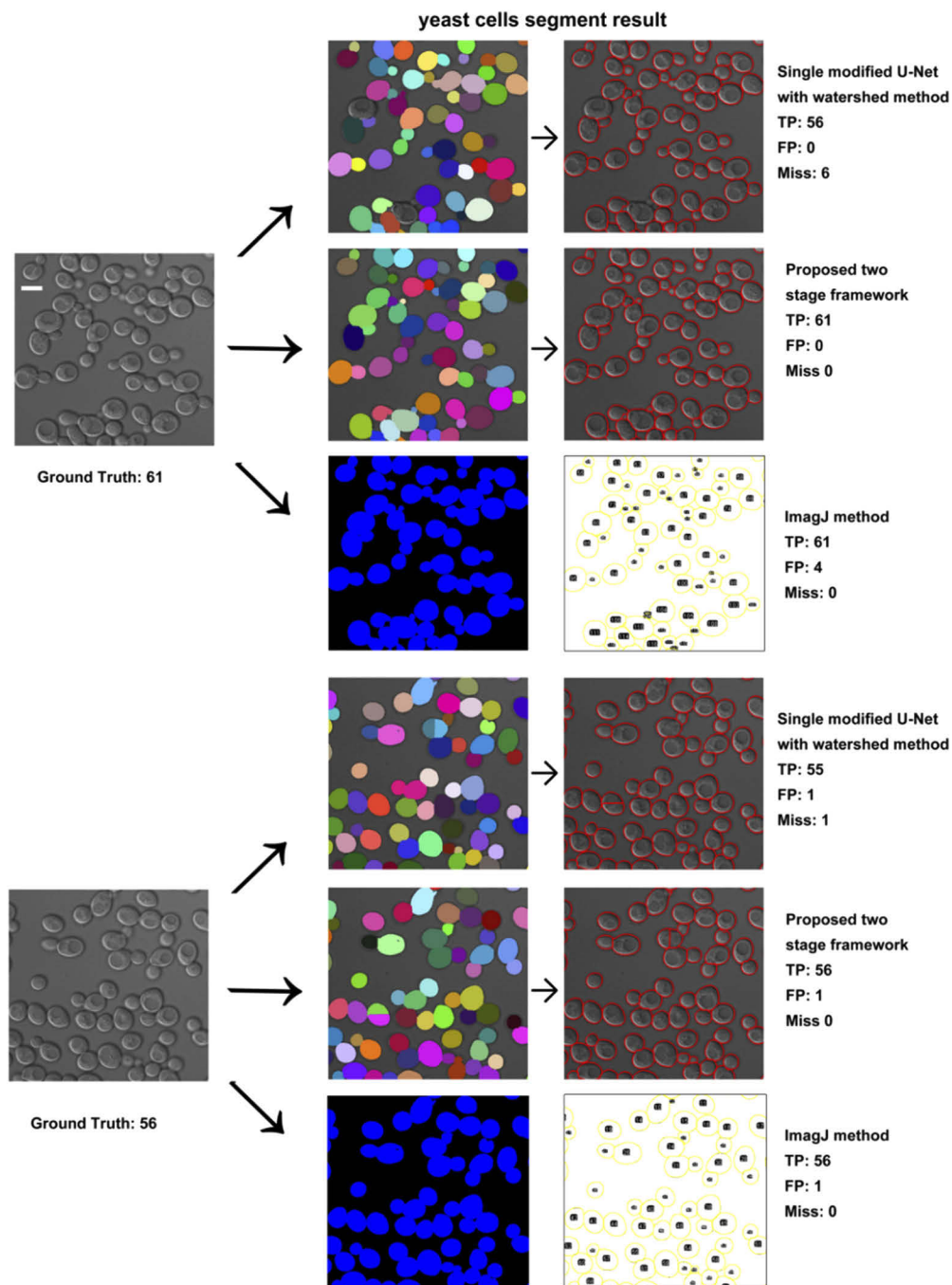


Fig. 4. Example of yeast cells segmentation. First column: raw DIC image, second, and third column: segmentation results of different methods. Scale bar: 5 μm .

5. Discussion

In this study, we combined the advantage of deep convolution networks and computer vision and proposed a parallel modified U-Net and seed guided watershed method-based two-stage framework for automatic segmentation and counting of yeast cells in fluorescent DIC images. Our method not only showed great reliability in automatic yeast cells counting but also showed acceptable time consumption. Some previous studies have proposed software and methods to automatically count cells, such as Cellprofiler [29,30] and ImageJ [1]. However, these methods appear to struggle to handle the fluorescent DIC images involved in this study. Because of the specificity of the DIC images, when we manually extract the features of yeast cells, we may get noise from nuclei and vacuoles as they have the same topology shapes as the yeast cells when captured by the microscope. The proposed modified U-Net was trained according to the established training sets which contained the whole image and cell region or cell center region segment annotation pairs and could obtain the overall features of all yeast cells. This was the key point and the first stage of our yeast counting framework.

We did experiment on additional test images, and checked those false negative (miss-detected) yeast cells. About 75% of the miss-detected yeast cells were new budding yeast cells (Fig. 5). Compared to mature yeast cells, new yeast cells are much smaller. Because of the limitations of microscopy, the camera does not capture the internal structure of the new- yeast cells. On the other hand, we also noticed that the miss-detected new yeast cells had low resolution which resulted in a blurring boundary which made them look less clearly defined than the mature ones. This indicates that our algorithm may also have some shortcomings in small object segmentation, which is a common problem. The accuracy of the small object detection has always been an important issue in the field of image segmentation [31,32]. The positive rate of recognition of newly budding yeast cells from fluorescence images will be an important goal in our future study, and an improved recognition rate will be of great utility for researchers that study the budding rate of yeast under extreme conditions. Finally, our two-stage framework, based on modified U-Net and combined with seeded watershed algorithm, is useful and reliable in automatic counting of yeast cells in fluorescence microscope images. Its use is not only limited to yeast cells; furthermore it can be employed in a variety of other cell experiments and in industry applications.

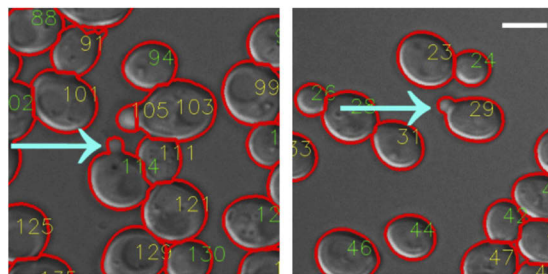


Fig. 5. Example of missed new budding yeast cells. Scale bar: 5 μm .

Funding

National Key Research and Development Program of China (2018YFC0910500); Neil Shen's SJTU Medical Research Fund; SJTU-Yale Collaborative Research Seed Fund.

Disclosures

The authors declare that they have no competing financial interests.

References

1. I. V. Grishagin, "Automatic cell counting with ImageJ," *Anal. Biochem.* **473**, 63–65 (2015).
2. M. Gamarra, E. Zurek, H. Jair, and L. Hurtado, "Biomedical Signal Processing and Control Split and merge watershed: A two-step method for cell segmentation in fluorescence microscopy images," *Biomed. Signal Process. Control* **53**, 101575 (2019).
3. S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk, "ilastik: interactive machine learning for (bio) image analysis," *Nat. Methods* **16**(12), 1226–1232 (2019).
4. R. Girshick, "Fast R-CNN," *Int. Conf. Comput. Vis.* 1440–1448 (2015).
5. R. Joseph and A. Farhadi, "YOLOv3: an incremental improvement," *arXiv Comput. Vis. Pattern Recognit.* (2018).
6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *arXiv Comput. Vis. Pattern Recognit.* (2015).
7. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Med. image Comput. Comput. Assist. Interv.* 9351, 234–241 (2015).
8. T. Falk, D. Mai, R. Bensch, Ahmed Abdulkadir, and Yassine Marrakchi, "U-Net: deep learning for cell counting, detection, and morphometry," *Nat. Methods* **16**(1), 67–70 (2019).
9. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large scale image recognition," *Int. Conf. Learn. Represent.* 1–14 (2015).
10. L. Han, T. H. N. Le, and M. Savvides, "An automatic cells detection and segmentation," *Proc. SPIE* **10137**, 101370W (2017).
11. M. J. Huttunen, C. W. McCloskey, J. Upham, B. C. Vanderhyden, R. W. Boyd, C. Vanderhyden, R. W. Boyd, and S. Murugkar, "Automated classification of multiphoton microscopy images of ovarian tissue using deep learning images of ovarian tissue using deep learning," *J. Biomed. Opt.* **23**(6), 066002 (2018).
12. D. Hong, G. Lee, N. C. Jung, and M. Jeon, "Fast automated yeast cell counting algorithm using bright-field and fluorescence microscopic images," *Biol. Proced. Online* **15**(1), 13 (2013).
13. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Comput. Vis. pattern Recognit.* **39**, 3431–3440 (2015).
14. A. Piniyaarachchi and C. Wahlby, "Seeded watershed for combined segmentation and tracking of cells," *Int. Conf. Image Anal. Process.* **3617**, 336–343 (2005).
15. "OpenCv," <https://opencv.org/> (n.d.).
16. N. K. Puneet and Garg, "Binarization Techniques used for Grey Scale Images," *Int. J. Comput. Appl.* **71**(1), 8–11 (2013).
17. S. K. Devalla, P. K. Renukanand, B. K. Sreedhar, G. Subramanian, L. Zhang, S. Perera, J.-M. Mari, K. S. Chin, T. A. Tun, N. G. Strouthidis, T. Aung, A. H. Thiery, and M. J. A. Girard, "DRUNET: a dilated-residual U-Net deep learning network to segment optic nerve head tissues in optical coherence tomography images," *Biomed. Opt. Express* **9**(7), 3244 (2018).
18. L. Wang, G. Sheng, W. Huang, and Y. Qiao, "Places205-VGGNet models for scene recognition," *arXiv Comput. Vis. Pattern Recognit.* 12–13 (2015).
19. P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," *neural Inf. Process. Syst.* 1990–1998 (2015).
20. Jia Deng, Wei Dong, R. Socher, Lijia Li, Kai Li, and Li Feifei, "ImageNet: a large-scale hierarchical image database," *IEEE Conf. Comput. Vis. Pattern Recognit.* 248–255 (2009).
21. J. B. T. M. Roerdink and A. Meijster, "The Watershed transform: definitions, algorithms and parallelization strategies," *Fundam. Informaticae* **41**(1,2), 187–228 (2000).
22. M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," *Comput. Vis. pattern Recognit.* 2858–2866 (2017).
23. G. Srinivasa and E. Society, "Multiresolution multiscale active mask segmentation of fluorescence microscope images," *Proc. SPIE* **74446**, 744603 (2009).
24. V. Der Walt, J. L. Sch, and J. Nunez-iglesias, "scikit-image: image processing in Python," *PeerJ* **2**, e453 (2014).
25. Y. Sasaki, "The truth of the F-measure," *Teach Tutor mater* 1–5 (2007).
26. N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE Trans. Med. Imaging* **36**(7), 1550–1560 (2017).
27. M. Abadi, P. Barham, J. Chen, Z. Chen, M. Devin, A. Davis, and J. Dean, "TensorFlow: a system for large-Scale machine learning," *Oper. Syst. Des. Implement.* 265–283 (2016).
28. Y. Kong, H. Li, Y. Ren, G. Z. Genchev, X. Wang, H. Zhao, Z. Xie, and H. Lu, "supplementary-table.xls," figshare (2020), <https://doi.org/10.6084/m9.figshare.11931450>
29. M. A. Bray, M. S. Vokes, and A. E. Carpenter, "Using cellProfiler for automatic identification and measurement of biological objects in images," *Curr. Protoc. Mol. Biol.* **1**(1), 1–13 (2016).
30. A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes," *Genome Biol.* **7**(10), R100–11 (2006).
31. L. Zheng and K. Tian, "Detection of small objects in sidescan sonar images based on POHMT and Tsallis entropy," *Signal Process.* **142**, 168–177 (2018).

32. R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka, "Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery," *Comput. Vis. Pattern Recognit.* 1–10 (2017).