

Bioimage informatics

# A convolutional neural network for segmentation of yeast cells without manual training annotations

Herbert T. Kruitbosch<sup>1</sup>, Yasmin Mzayek<sup>1</sup>, Sara Omlor<sup>1</sup>, Paolo Guerra<sup>2</sup> and Andreas Miliás-Argeitis<sup>1</sup> <sup>2,\*</sup>

<sup>1</sup>Center for Information Technology, University of Groningen, 9747 AJ Groningen, The Netherlands and <sup>2</sup>Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, 9747 AG Groningen, The Netherlands

\*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on April 11, 2021; revised on October 9, 2021; editorial decision on November 26, 2021; accepted on December 7, 2021

## Abstract

**Motivation:** Single-cell time-lapse microscopy is a ubiquitous tool for studying the dynamics of complex cellular processes. While imaging can be automated to generate very large volumes of data, the processing of the resulting movies to extract high-quality single-cell information remains a challenging task. The development of software tools that automatically identify and track cells is essential for realizing the full potential of time-lapse microscopy data. Convolutional neural networks (CNNs) are ideally suited for such applications, but require great amounts of manually annotated data for training, a time-consuming and tedious process.

**Results:** We developed a new approach to CNN training for yeast cell segmentation based on synthetic data and present (i) a software tool for the generation of synthetic images mimicking brightfield images of budding yeast cells and (ii) a convolutional neural network (Mask R-CNN) for yeast segmentation that was trained on a fully synthetic dataset. The Mask R-CNN performed excellently on segmenting actual microscopy images of budding yeast cells, and a density-based spatial clustering algorithm (DBSCAN) was able to track the detected cells across the frames of microscopy movies. Our synthetic data creation tool completely bypassed the laborious generation of manually annotated training datasets, and can be easily adjusted to produce images with many different features. The incorporation of synthetic data creation into the development pipeline of CNN-based tools for budding yeast microscopy is a critical step toward the generation of more powerful, widely applicable and user-friendly image processing tools for this microorganism.

**Availability and implementation:** The synthetic data generation code can be found at <https://github.com/prhbrt/synthetic-yeast-cells>. The Mask R-CNN as well as the tuning and benchmarking scripts can be found at <https://github.com/ymzayek/yeastcells-detection-maskrcnn>. We also provide Google Colab scripts that reproduce all the results of this work.

**Contact:** [a.milias.argeitis@rug.nl](mailto:a.milias.argeitis@rug.nl)

**Supplementary information:** [Supplementary data](#) are available at *Bioinformatics* online.

## 1 Introduction

The use of time-lapse microscopy to study cellular process dynamics at the single-cell level has become indispensable for systems biology research. While large volumes of time-lapse microscopy data can be generated relatively easily today, processing the resulting microscopy movies to extract high-accuracy single-cell information is still challenging. Two key steps in this processing pipeline are (i) the detection and outlining of each distinct cell appearing in an image (instance segmentation) and (ii) the tracking of individual cells across the frames of a microscopy movie. While genetically encoded fluorescent markers can greatly facilitate these tasks, they have several

drawbacks that limit their applicability (Versari *et al.*, 2017; Vicar *et al.*, 2019). Therefore, label-free imaging techniques such as brightfield are preferable wherever possible.

In the case of *Saccharomyces cerevisiae* (budding yeast), a model eukaryote of central importance both in fundamental research and applications, manual segmentation and tracking of large cell numbers in brightfield movies is very time-consuming. For this reason, several software tools have already been introduced to automate these tasks (Bredies and Wolinski, 2011; Carpenter *et al.*, 2006; Dimopoulos *et al.*, 2014; Gordon *et al.*, 2007; Pelet *et al.*, 2012; Versari *et al.*, 2017; Wang *et al.*, 2009; Wood and Donic, 2019). However, almost all of them are

based on classical image processing techniques (such as watershed, thresholding and contour tracing) for cell detection, and require careful fine-tuning to perform well. Moreover, their performance may fluctuate across datasets acquired under different imaging conditions, which limits their generalizability.

Convolutional neural networks (CNNs) have clearly demonstrated their power over traditional image processing techniques in several types of challenging biological applications (Angermueller et al., 2016; Moen et al., 2019). Yet, the use of CNNs for the processing of yeast microscopy movies was relatively limited until very recently, when several promising tools based on CNNs (Dietler et al., 2020; Lu et al., 2019; Prangemeier et al., 2020b; Salem et al., 2021) and attention-based transformers (Prangemeier et al., 2020a) appeared. A key advantage of these approaches over traditional image processing methods is that they are significantly more robust to changes in imaging conditions and experimental setups, while their accuracy on problematic cases can still be improved by retraining with data representative of those cases. On the negative side, these tools require large, diverse and high-quality annotated datasets to be trained (Moen et al., 2019; Vicar et al., 2019), and model performance depends heavily on the quality and size of the training dataset. Unfortunately, these training datasets typically need to be manually generated or curated from actual experimental data. For example, training a CNN for yeast segmentation requires manually drawing cell boundaries for thousands of single cells across different microscopy images, while ensuring that no cells are missed. Therefore, the generation of training data for such a CNN is a very time-consuming process, which may need to be repeated whenever the CNN has to be retrained to achieve better performance on new imaging setups.

Contrary to datasets constructed from real-world data, synthetic datasets are generated algorithmically. Compared to the time and effort required to generate manually annotated datasets, the automatic generation of large volumes of synthetic training data for CNNs is extremely time-efficient and requires only minimal human input. A synthetic dataset does not need to be a faithful reproduction of the real-world data, as long as it can reproduce key features of these data. Moreover, the annotation of synthetic data is fully accurate by construction. Synthetic data have already been used to create well-understood training datasets for various CNN applications such as 3D object segmentation (Danielczuk et al., 2019), text detection (Gupta et al., 2016), segmentation of agar plates (Andreini et al., 2020) and crop seed instance segmentation (Toda et al., 2019). Despite the relative simplicity of yeast cell shapes, synthetic generation of yeast microscopy images and the use of these images for CNN training has not been tried yet. Such an approach could greatly accelerate the development of deep learning-based tools for yeast microscopy.

In this work, we present a versatile tool for the automatic creation of synthetic brightfield images of yeast-like objects, together with a Mask R-CNN (He et al., 2020) trained on a synthetic dataset to perform cell segmentation. Despite the fact that our Mask R-CNN was only trained on synthetically generated data, it is able to segment yeast cells on actual brightfield microscopy images with remarkable accuracy. By feeding the output of the Mask R-CNN into a density-based spatial clustering algorithm (DBSCAN; Ester et al., 1996), we were able to automatically and accurately track individual yeast cells across microscopy movies. Synthetic data creation completely bypasses the laborious generation of manually annotated datasets, and can be easily adjusted to produce images with many different features.

We demonstrate the applicability of our approach by segmenting and tracking yeast cells grown inside a microfluidic device with pillar-like structures (Lee et al., 2012), as well as cells grown under agarose pads in two different microscopy setups. Using benchmarking datasets from the Yeast Image Toolkit (YIT; Versari et al., 2017; <http://yeast-image-toolkit.biosim.eu/>), we compare the performance of the Mask R-CNN to that of two recently presented CNNs: YeaZ (Dietler et al., 2020; the current state-of-the-art in the field) and YeastNet2 (Salem et al., 2021). The Mask R-CNN performed very similarly to YeaZ and better than YeastNet2, even though the

last two tools were trained on large, manually annotated datasets. Our results demonstrate that the incorporation of synthetic data into the development pipeline of CNN-based tools for budding yeast microscopy is an important step toward the generation of more powerful, widely applicable and user-friendly image processing tools.

## 2 Materials and methods

### 2.1 Synthetic training data generation

We focused on the generation of synthetic images of cells imaged in white light with brightfield microscopy, since brightfield imaging avoids phototoxicity and photobleaching artifacts, does not require genetic engineering of the cells, and can be readily acquired in every widefield microscopy setup.

We used OpenCV to draw random ellipses with blurred dark and bright boundaries to create slightly out-of-focus cell-like objects. To mimic the noisy background and cell interior, we used NumPy to create Gaussian random fields whose standard deviations were estimated based on real data. Finally, the image dataset was augmented using the PiecewiseAffine transformation of the `imgaug` image augmentation library (<https://imgaug.readthedocs.io/>), which slightly deformed the ellipsoidal cell-like objects.

### 2.2 Instance segmentation

To arrive at our final choice of instance segmentation model, we compared two alternatives: U-Net (Ronneberger et al., 2015) and Mask R-CNN (He et al., 2020).

A key challenge of U-Net is that it performs semantic segmentation for two or more classes, but does not separate instances, producing a single probability map instead. This in turn implies that a significant amount of post-processing is necessary to locate individual cells. In our tests with U-Net, we identified connected components as separate instances. Since the connected components do not accurately segment each cell, we used seam carving (Avidan and Shamir, 2007) to trace paths across the black and white rings on the cell boundary using a polar coordinate system centered on the centroid of each connected component, and determined the cell boundary as the average of these two paths. Still, U-Net detected many false positives, which required training a simple classifier on manually labeled detections based on features such as path circularness, the distance between the path endpoints and the enclosed area. Due to the high number of false positives and the large amount of post-processing required at the U-Net output, we did not further explore the use of U-Net.

Contrary to U-Net, Mask R-CNN directly performs segmentation of individual instances by applying a region proposal technique with many false positives to find potential bounding boxes for objects, and estimating the probability of each bounding box containing an object or nothing. Since we were interested in detecting yeast cells, we only had to estimate the probabilities  $p_{\text{background}}$  and  $p_{\text{yeast cell}}$  for each bounding box. Yeast cells were then detected based on a threshold on  $p_{\text{yeast cell}}$ , e.g.  $p_{\text{yeast cell}} \geq 0.8$ . Further technical details on the Mask R-CNN implementation used here can be found in [Supplementary Section S1](#).

### 2.3 Tracking

To track a segmented cell across the frames of a microscopy movie, we used the DBSCAN algorithm (Ester et al., 1996) to cluster together cell detections across frames. DBSCAN is a widely used clustering algorithm that allows transitive clustering of cell detections over long time spans whenever intermediate detections have enough overlap. DBSCAN was chosen because it operates in an intuitive manner, using quantities such as the minimum number of detections required to start tracking a cell, and the minimum required overlap to consider two detections close enough to belong to one cell. Contrary to clustering methods that optimize cluster compactness (such as k-means), DBSCAN favors cluster separation instead. Since cells in most imaging setups are not fully restricted, can move over

time and are present throughout a movie, cluster compactness cannot be expected. Instead, given that cells do not move much between frames relative to the space they occupy, distinguishing individual cells in adjacent frames by their lack of overlap is always possible, unless the frame rate is too low to permit reliable tracking.

To determine whether detected objects across adjacent or nearby frames correspond to the same cell, we defined a distance measure based on the amount of overlap [intersection over union (IoU)] of the objects and the time difference between frames. Our distance measure assigns infinite distance to (i) detections more than  $\Delta t_{\max}$  frames apart, (ii) detections which have no mutual overlap between two frames and (iii) detections in the same frame. All detections with non-infinite distance were assigned a distance equal to one minus IoU

$$d(x, y) = 1 - \left[ \frac{\text{area}(x \cap y)}{\text{area}(x \cup y)} \right],$$

where  $x$  and  $y$  denote the objects detected at frames  $t_x$  and  $t_y$ , respectively.

Based on this distance metric, DBSCAN determines dense regions of a given set based on the parameters  $\epsilon$  and  $\text{min\_samples}$ . More specifically, a set of cell detections with at least  $\text{min\_samples}$  members that overlap at least  $\epsilon$  with one member is considered a dense region and will belong to the same cluster. Dense regions that overlap at least  $\epsilon$ , as well as all detections that overlap at least  $\epsilon$  with one of the dense region members, are also considered to be the same cluster. In this way, clusters are first defined by their dense regions, and then nearby detections are added. Since the goal of our clustering method is not cluster compactness, detections with infinite distance over more than one frame can be assigned to one cluster via detections at intermediate frames. Finally, DBSCAN identifies outliers as detections which do not overlap enough with any cell of any dense region, and ignores them.

Given that each cluster of detections forms a single-cell trajectory, a trajectory could contain two detections at one time frame, e.g. when a cell gets assigned two detection masks due to a segmentation error. When this occurred, we only selected the most likely detection according to the Mask R-CNN and used it for the evaluation of tracking performance. At the same time, the additional detections were penalized as false positives for tracking.

## 2.4 Test set description

We evaluated the performance of our Mask R-CNN trained on our synthetic dataset using benchmark data from the YIT, a publicly available evaluation platform for comparing segmentation and tracking performance on yeast microscopy images (Versari et al., 2017). This platform provides 10 test sets of budding yeast microscopy movies with annotated ground-truth data (cell center locations and unique cell labels throughout each movie). We chose the seven test sets (TS1–7) that contain brightfield images to evaluate the segmentation and tracking performance of the Mask R-CNN and DBSCAN, as described below. TS1–2 contain isolated single cells and small growing colonies, whereas TS3–7 show larger, heavily clustered and merging colonies. Although our Mask R-CNN was trained on synthetic images with relatively sparse cell arrangements, TS3–7 were included in the tests to show how our algorithm generalizes to this type of data.

Besides the YIT datasets, we also used the brightfield images of wild-type cells from the YeaZ dataset (<https://www.epfl.ch/labs/lpbs/data-and-software/>) obtained at the lowest exposure level. These images contain pixel-level ground-truth annotations of cell masks, which were used to evaluate the quality of cell masks produced by the Mask R-CNN and compare it with the quality of YeaZ segmentations. We did not use images of mutant cells, as these cells have shapes that were not included in our synthetic training set.

## 2.5 Performance metrics

To assess the performance of the Mask R-CNN and the DBSCAN algorithm on the YIT datasets, we considered the different types of

outcomes that these two algorithms can produce. A detailed description of these outcomes and the types of errors that they generate can be found in the [Supplementary Sections S3 and S4](#) and [Figures S2 and S3](#).

The performance of the Mask R-CNN and DBSCAN was assessed via the  $F1$ -score and an accuracy measure defined in [Dietler et al. \(2020\)](#) for each of these two tasks and YIT test set. For the definition of the  $F1$ -score, the use of precision and recall is required, as shown below:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}),$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}).$$

$$F1 = 2 \cdot (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}),$$

where TP: true positive detections; FP: false positive detections; and FN: false negatives. Intuitively, recall reveals whether actual cells are missed, and precision whether noise is falsely picked up as a cell. The accuracy measure was defined as

$$\text{Accuracy} = \text{TP}/(\text{TP} + \text{FN} + \text{FP}).$$

The  $F1$ -score ranges between 0 and 1, and is equal to the harmonic mean of the precision and recall. This means that high precision and recall are required to obtain an  $F1$ -score close to 1. On the other hand, the accuracy measures the total number of correct detections versus the sum of the correct, incorrect and missed detections.

Segmentation quality on the YeaZ dataset was calculated via the IoU between detected objects and ground-truth cell masks.

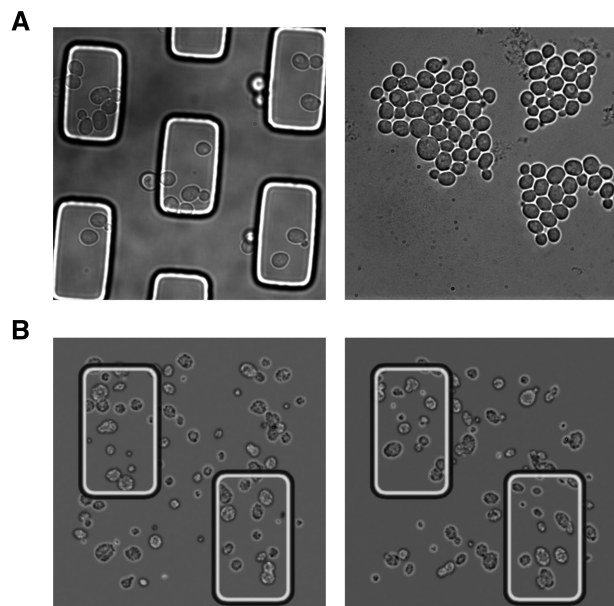
## 3 Results

### 3.1 Synthetic training dataset description

The generation of our training data was based on the observation that the geometry of a budding yeast cell in a brightfield image is relatively simple and can therefore be reproduced synthetically. As [Figure 1](#) shows, budding yeast cells imaged by brightfield microscopy appear as ellipsoids with a relatively narrow range of sizes, which display a specific light-dark pattern on their periphery. The range of sizes and the maximum eccentricity of our synthetic cell-like objects were based on rough estimates from real data.

Since synthetic cells are positioned independently of each other and overlaps are allowed, we selected the number of objects per frame to balance two conflicting requirements: on the one hand, a large amount of overlap between nearby objects is unrealistic and gives rise to label noise, which implies that synthetic cells should not overlap too much. On the other hand, having many objects in a frame allows the network to learn more from that frame, which increases training efficiency. We found that 100 synthetic cells per frame achieve a good balance between these two objectives.

To successfully train a model that generalizes to real data, we created 20,000  $512 \times 512$  images, each with 100 randomly arranged synthetic cells. For each cell, the boundary pattern was created by arranging a black elliptical ring within a white one. The width of these rings, their arrangement (outer-black, inner-white or vice versa) and the amount of Gaussian blurring were adjusted to mimic actual yeast images. Since real yeast cells are not perfect ellipses, we deformed the generated elliptical objects by applying a randomly generated piecewise affine transformation (cf. [Supplementary Section S2](#)). To simulate the effect of out-of-focus noise-like features seen in real images, we used a Gaussian random field with the standard deviation of the spatial correlation set to 2 pixels. Since there are more noise-like features in the cell interiors compared to the background, we scaled the standard deviation of the noise to 0.002 outside and 0.03 inside the cells, respectively, based on rough estimates obtained from a small sample of real images. Finally, we set the average background grayscale intensity to 0.4. The cell areas used to



**Fig. 1.** Actual versus synthesized microscopy images. (A; left) Brightfield image of budding yeast cells grown inside a microfluidic device with pillar-like rectangular structures (Lee et al., 2012). Cells get trapped and grow underneath these structures, while growth medium flows continuously through the device. Small cells, as well as cells that get dislodged from underneath the pillars, get washed away. (Right) Brightfield image of yeast cells growing inside a microfluidic device without microstructures. This image was taken from test set 3 (TS3) of the YIT. In both panels, image contrast was adjusted to improve their appearance in this figure. (B) Two synthetically generated images from the dataset used to train our Mask R-CNN. Each image contained 100 cell-like objects placed at random positions which allow overlap. The image background also contained the pillar-like structures of our microfluidic device. Cell-like objects were placed over the whole image area, to help the Mask R-CNN identify cells over the whole image and not only at specific locations. On each synthetic image, the pixels belonging to each cell-like object were annotated and used to train the Mask R-CNN for instance segmentation

train our instance segmentation model were defined by the inner boundary of each synthetic object, but their specification can be easily adapted to different imaging setups. Even though the resulting synthetic objects do not precisely mimic real yeast cells, they still capture important aspects of actual yeast images (size, ellipticity, membrane patterns, etc.), which can be learned by an instance segmentation algorithm. The generation of the dataset described above took 70 min on a 40-core machine, without significant code optimization.

In addition to generating cell-like objects, synthetic data creation algorithms can simulate the appearance of different features of the microfluidic chip in which the cells are cultured. These features are typically found in microfluidics containing single-cell traps (Prangemeier et al., 2020b) or pads under which a small number of cells can be trapped (Lee et al., 2012). Such microstructures are visible in the same field of view with the cells that need to be tracked, and can confuse CNNs unless they are explicitly accounted for in the training (Prangemeier et al., 2020b). Since our group works with such a microfluidics chip for long-term imaging of cells under constant nutrient conditions (Lee et al., 2012), we included the pillar-like structures of our chip in the synthetic images that we created for training of our Mask R-CNN. In this way, the Mask R-CNN was trained to ignore these structures to avoid false positives. Two samples of the resulting synthetic images with cell-like objects and the microfluidic chip structures are shown in Figure 1. More examples of the types of images that can be generated by tuning the parameters described above can be found on the corresponding GitHub page. Further technical details on synthetic image generation can be found in Supplementary Section S2 and Figure S1.

## 3.2 Hyperparameter tuning

We tested how the performance of our segmentation and tracking algorithms varies with respect to three key hyperparameters: (i) the segmentation threshold ( $p_{\text{thr}}$ ) of the Mask R-CNN, (ii) the epsilon hyperparameter ( $\epsilon$ ) of DBSCAN and (iii) the maximum allowed separation between frames in the DBSCAN distance function ( $\Delta t_{\text{max}}$ ). To investigate the effect of these hyperparameters on segmentation and tracking performance, we plotted the different performance metrics as a function of the hyperparameter values for the brightfield datasets from the YIT. With the resulting calibration curves, we tuned the Mask R-CNN and DBSCAN parameters to optimize the  $F1$ -score of segmentation and tracking, respectively.

### 3.2.1 Segmentation threshold tuning

By design, our Mask R-CNN generates 2000 region proposals, along with a probability estimate for each region. This probability expresses how likely it is that the region corresponds to a yeast cell ( $p_{\text{yeast cell}}$ ) or not ( $p_{\text{background}} = 1 - p_{\text{yeast cell}}$ ). From this large number of proposals, false positives need to be filtered out by choosing proposals for which  $p_{\text{yeast cell}}$  is larger than a user-defined threshold  $p_{\text{thr}}$ . With this threshold, one can choose to maximize precision (higher  $p_{\text{thr}}$ ), recall (lower  $p_{\text{thr}}$ ) or  $F1$ -score (intermediate  $p_{\text{thr}}$ ). Higher thresholds make it less likely that detected objects are false positives, while lower thresholds allow more low-confidence cells to be detected. The choice of  $p_{\text{thr}}$  can be made for a particular dataset by scanning through a range of threshold values and picking the one that performs best according to user objectives. This scanning is enabled by the fact that our image processing pipeline (Mask R-CNN + DBSCAN) achieves very fast runtimes. A detailed breakdown of running times for the YIT datasets can be found in Supplementary Table S1.

Supplementary Figure S4 shows the performance metrics of our Mask R-CNN with respect to  $p_{\text{thr}}$  for datasets TS1–7 from the YIT. As can be observed, the  $F1$ -score for TS1 and TS2 is maximized for threshold values around 0.97, as lower threshold values lead to a decrease in precision due to false positive detections. On the other hand, the  $F1$ -score for TS3–7 drops at threshold values close to 1 due to the decrease in recall. When the threshold is lowered to around 0.8; however, the  $F1$ -score becomes largely independent of the precise threshold value. To be able to use a common threshold of 0.8 for both sparse (TS1–2) and dense (TS3–7) YIT sets, we implemented a simple post-processing step to remove false positive detections based on their size. This was possible because the vast majority of false positives correspond to objects that are much smaller than cells and the smallest detectable buds. As shown in Supplementary Figure S5, use of a size threshold between 20 and 100 pixels greatly improved model precision for TS1 and TS2, while leaving model performance on TS3–7 unaffected. By implementing this simple filtering step using a size threshold of 50, we were able to make the performance of the Mask R-CNN largely independent of the threshold on all YIT datasets (Supplementary Fig. S5).

### 3.2.2 DBSCAN tuning

The `min_samples` parameter was set to 3 in all DBSCAN runs. To tune the other DBSCAN parameters ( $\epsilon$  and  $\Delta t_{\text{max}}$ ), we evaluated the tracking performance of the clustering algorithm for a range of parameter values for all YIT datasets (TS1–7). The calibration results (Supplementary Figs S6–S12) show that  $\epsilon = 0.6$  and  $\Delta t_{\text{max}} = 1$  provide good  $F1$ -score performance across all tested YIT datasets.

## 3.3 Benchmarking

We compared the segmentation and tracking performance of our Mask R-CNN with two recently presented networks based on the U-Net architecture: YeaZ (Dietler et al., 2020) and YeastNet2 (Salem et al., 2021). For the comparison, we used the seven datasets from the YIT described above (TS1–7). We did not carry out a comparison with YeastSpotter (Lu et al., 2019), another Mask R-CNN

for budding yeast, since this tool was not trained on yeast cell images and its performance appears inferior to YeaZ (Dietler *et al.*, 2020). All evaluation scripts are provided in our GitHub repository.

To run YeastNet2, we followed the instructions provided by the authors in their GitHub repository (<https://github.com/kaernlab/YeastNet>). YeaZ (downloaded from <https://github.com/lpbsscientist/YeaZ-GUI> and run without the GUI here) contains a tunable detection threshold parameter which operates at the pixel level before detected objects are separated, rather than the instance level, as the Mask R-CNN parameter does. While the YeaZ threshold mostly affects cell outlines and not false positive or false negative rates, a scan over different threshold values showed that YeaZ performance improves slightly at threshold values close to 1. We therefore performed all tests with YeaZ using a threshold of 0.95, noting that YeaZ developers use a default threshold of 0.5. Finally, for our Mask R-CNN, we chose a threshold value of 0.8 and a pixel area threshold of 50, as described in Section 3.2. The results of the comparison are displayed in Table 1. As can be seen, our Mask R-CNN outperforms YeastNet2 and performs similarly to YeaZ, even though the latter tool was trained on more than 10 000 high-quality segmented yeast cells (Dietler *et al.*, 2020).

More specifically, the Mask R-CNN achieved similar *F1*- and accuracy scores with YeaZ for all test sets with regard to segmentation (Table 1) and tracking (Table 2). False negatives in tracking are mostly due to the propagation of a segmentation error, which suggests that accurate segmentation is the main determinant of tracking accuracy. A detailed breakdown of the causes of false negatives and false positives in tracking is provided in Supplementary Table S2.

Since the YIT does not provide annotated cell areas but only centroid coordinates for the detected cells, it was necessary to further evaluate the quality of the segmentation achieved by our Mask R-CNN. Visual inspection of segmentations across different imaging setups (Fig. 2) showed that the Mask R-CNN is able to accurately detect cell boundaries in the majority of cases, even in crowded conditions. Besides, the addition of the pillar-like structures in the synthetic training data allowed the Mask R-CNN to ignore the structures of the microfluidic chip in the real images and to precisely segment cells that are very close to the edges of the pillars (Fig. 2C and D).

Accuracy in cell area estimation is needed in several applications of yeast time-lapse microscopy, such as the calculation of cell growth rates (Ferrezuelo *et al.*, 2012), or the estimation of fluorescent protein abundance over time (Cookson *et al.*, 2010). Examples of cell area time series obtained from the Mask R-CNN + DBSCAN and their comparison with manually curated results from BudJ

(Ferrezuelo *et al.*, 2012), a segmentation and tracking plugin of ImageJ, can be found in Supplementary Figure S11. To evaluate the segmentation accuracy of the Mask R-CNN in more quantitative terms, we used the annotated brightfield images of wild-type cells provided on the YeaZ website. The results presented in Table 3 show that the segmentation quality achieved by the Mask R-CNN is comparable to that of YeaZ. Visual comparisons of the YeaZ and Mask R-CNN output with the ground truth are provided in Supplementary Figures S12–S14.

## 4 Conclusions

Given the key role that budding yeast still plays in basic biological research and synthetic biology, the availability of robust and precise tools for automatic yeast cell segmentation and tracking is crucial for obtaining a better understanding of fundamental cellular processes such as growth, division and aging. Here, we presented a highly customizable software tool for the creation of synthetic images that mimic budding yeast cells imaged with brightfield microscopy. Using a dataset created by this tool, we trained a Mask R-CNN to segment the synthetically generated objects and tested its performance on actual microscopy images of yeast cells. Our Mask R-CNN performed exceptionally well on the actual images, despite not having been trained on them. The good quality of segmentation further allowed us to track yeast cells across microscopy movies by implementing a DBSCAN. The segmentation and tracking performance of the resulting combination of Mask R-CNN and DBSCAN were similar to that of YeaZ, the current state-of-the-art tool in the field. Moreover, the extracted area profiles of the segmented cells compared favorably with high-quality area profiles of the YeaZ dataset, confirming that the proposed combination of Mask R-CNN with DBSCAN can be used to automatically extract single-cell information from time-lapse microscopy movies.

We believe that synthetic training dataset generation for the task proposed here is an important methodological advance compared to previous approaches. Deep learning methods for image processing require very large amounts of high-quality annotated data, and neural networks for yeast cell segmentation are no exception. The generation of these training datasets in turn requires a considerable amount of time and experience from human annotators. Moreover, the annotation process has to be carried out whenever the neural network has to be retrained on new data types, such as altered imaging conditions. Synthetic data can be generated with minimal effort and can be easily adapted to reflect different imaging setups. To demonstrate this flexibility, our synthetic training images contained

**Table 1.** Segmentation performance metrics for Mask R-CNN, YeaZ and YeastNet2 evaluated on the brightfield test sets of the YIT

Test set	Precision			Dataset	Recall		
	Mask R-CNN	YeaZ	YeastNet2		Mask R-CNN	YeaZ	YeastNet2
TS1	94.7	100.0	100.0	TS1	93.5	91.9	79.8
TS2	100.0	100.0	100.0	TS2	94.5	94.5	84.8
TS3	99.9	98.9	100.0	TS3	89.2	92.2	71.7
TS4	99.9	99.3	100.0	TS4	84.9	86.5	75.5
TS5	98.4	98.2	99.3	TS5	87.9	82.6	72.8
TS6	99.0	99.7	99.7	TS6	86.9	87.8	71.9
TS7	98.0	99.8	98.4	TS7	88.4	89.8	77.7

Test set	F1-score			Dataset	Accuracy		
	Mask R-CNN	YeaZ	YeastNet2		Mask R-CNN	YeaZ	YeastNet2
TS1	94.1	95.8	88.8	TS1	88.8	91.9	79.8
TS2	97.2	97.2	91.8	TS2	94.5	94.5	84.8
TS3	94.2	95.5	83.5	TS3	89.1	91.3	71.7
TS4	91.8	92.5	86.1	TS4	84.9	86.0	75.5
TS5	92.8	89.7	84.0	TS5	86.6	81.4	72.4
TS6	92.5	93.4	83.6	TS6	86.1	87.6	71.8
TS7	92.9	94.5	86.8	TS7	86.8	89.6	76.7

Note: Highlighted cells denote the tool with the highest performance for each test set.

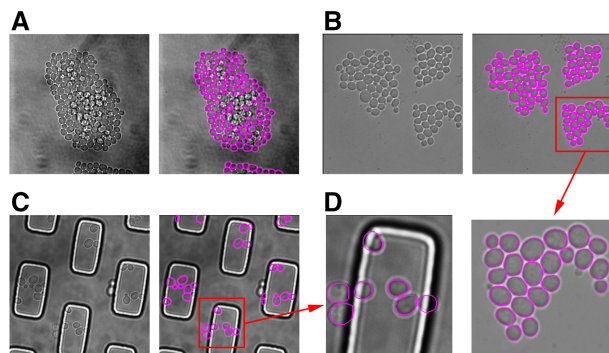
**Table 2.** Tracking performance metrics for Mask R-CNN, YeaZ and YeastNet2 evaluated on the brightfield test sets of the YIT

Test set	Precision			Dataset	Recall		
	Mask R-CNN	YeaZ	YeastNet2		Mask R-CNN	YeaZ	YeastNet2
TS1	94.7	100	97.0	TS1	93.4	91.9	77.1
TS2	100.0	100	100.0	TS2	93.5	94.2	84.2
TS3	99.7	97.9	82.3	TS3	83.7	91.3	58.6
TS4	99.6	97.4	82.3	TS4	75.5	84.9	62.1
TS5	98.6	96.6	86.2	TS5	85.0	81.7	63.5
TS6	99.4	99.4	99.3	TS6	80.7	86.8	71.0
TS7	98.7	99	97.4	TS7	82.4	89.4	76.8

Test set	F1-score			Dataset	Accuracy		
	Mask R-CNN	YeaZ	YeastNet2		Mask R-CNN	YeaZ	YeastNet2
TS1	94.1	95.8	85.9	TS1	88.8	91.9	75.3
TS2	96.7	97	91.4	TS2	93.5	94.2	84.2
TS3	91.0	94.5	68.4	TS3	83.5	89.5	52.0
TS4	85.9	90.7	70.8	TS4	75.2	83	54.8
TS5	91.3	88.6	73.1	TS5	84.0	79.5	57.7
TS6	89.0	92.7	82.8	TS6	80.3	86.3	70.6
TS7	89.8	94	85.9	TS7	81.5	88.7	75.3

Note: Highlighted cells denote the tool with the highest performance for each test set.



**Fig. 2.** Segmentation of yeast cells in different imaging setups. (A) Large, dense colony growing under a nutrient-infused agarose pad in our imaging setup. Cells in the center of the colony have been pushed vertically and are largely out of focus. Despite the large amount of crowding, our Mask R-CNN was able to accurately detect the majority of cells, even though it was not trained on such dense images. Objects detected by the neural network are marked with magenta outlines. Cells that were not detected do not carry an outline. (B) Cells growing inside the microfluidic device used in Uhlendorf et al. (2012). (C) Cells growing inside the microfluidic device used in our group. In such sparse cell configurations, our Mask R-CNN is able to detect a wide range of cell sizes, from large, aged mother cells, to young growing buds. (D) Inset showing close-up views of cell boundaries detected by the Mask R-CNN. In all panels, contrast was adjusted to improve their appearance in this figure; no contrast adjustments were made to the images that were provided to the Mask R-CNN

features of a microfluidic chip used in our experiments, and the inclusion of these features prevented the Mask R-CNN from incorrectly recognizing the chip features.

Several future improvements and extensions will make synthetic data generation even more powerful, leading to further improvements in segmentation and tracking performance. For example, the recall of the Mask R-CNN and the detection of small growing buds in dense cell configurations can be improved by generating densely packed cell-like objects to mimic the appearance of crowded microscopy fields. Such configurations are difficult to obtain with our current data creation tool due to the fact that object positions are generated independently at random. More realistic synthetic data

**Table 3.** Average IoU of true positive instances in the annotated brightfield images of wild-type cells from the YeaZ dataset

Test set	wtF2BF	wtF3BF	wtF4BF	wtF5BF	wtF6BF
Mask R-CNN	81.6	82.0	84.6	84.5	81.2
YeaZ	85.8	85.3	87.2	88.0	84.2
Test set	wtF7BF	wtF8BF	wtF9BF	wtF10BF	wtF11BF
Mask R-CNN	81.7	78.7	75.7	79.8	84.5
YeaZ	81.8	82.9	80.2	85.3	85.8
Test set	wtF12BF	wtF13BF	wtF14BF	wtF15BF	
Mask R-CNN	78.9	84.0	83.4	83.8	
YeaZ	84.2	86.0	86.4	86.0	

Note: The YeaZ dataset contains images obtained at six different exposure levels. For all the tests performed here, the lowest exposure level was used. The two models were run using the same threshold values as in the YIT tests.

could be produced using an algorithm for the generation of dense object packings (Delaney et al., 2005), or by simulating the growth of yeast microcolonies using existing physics-based simulators (Jönsson and Levchenko, 2005; Wang et al., 2017). In the latter case, CNNs such as TrackR-CNN (Voigtlaender et al., 2019) could be trained to simultaneously perform cell segmentation and tracking using synthetically generated microscopy movies. Additional steps to improve data realism could include the generation of non-random internal structures inside the cell-like objects (e.g. vacuoles), as well as cell objects mimicking budding cells. At the same time, the generation of synthetic images that imitate other transmitted light imaging modalities, e.g. phase contrast and differential image contrast, can further expand the range of applicability of our Mask R-CNN.

## Funding

A.M.-A. was supported by the Dutch Research Council (NWO) through an NWO-VIDI grant (project number 016.Vidi.189.116).

Conflict of Interest: none declared.

## References

- Andreini, P. *et al.* (2020) Image generation by GAN and style transfer for agar plate image segmentation. *Comput. Methods Programs Biomed.*, **184**, 105268.
- Angermueller, C. *et al.* (2016) Deep learning for computational biology. *Mol. Syst. Biol.*, **12**, 878.
- Avidan, S. and Shamir, A. (2007) Seam carving for content-aware image resizing. In: *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*. ACM Press, New York, NY, p. 10.
- Bredies, K. and Wolinski, H. (2011) An active-contour based algorithm for the automated segmentation of dense yeast populations on transmission microscopy images. *Comput. Vis. Sci.*, **14**, 341–352.
- Carpenter, A.E. *et al.* (2006) CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.*, **7**, R100–R111.
- Cookson, N.A. *et al.* (2010) Cell cycle-dependent variations in protein concentration. *Nucleic Acids Res.*, **38**, 2676–2681.
- Danielczuk, M. *et al.* (2019) Segmenting unknown 3D objects from real depth images using Mask R-CNN trained on synthetic data. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., pp. 7283–7290.
- Delaney, G. *et al.* (2005) Random packing of elliptical disks. *Philos. Mag. Lett.*, **85**, 89–96.
- Dietler, N. *et al.* (2020) A convolutional neural network segments yeast microscopy images with high accuracy. *Nat. Commun.*, **11**, 1–8.
- Dimopoulos, S. *et al.* (2014) Accurate cell segmentation in microscopy images using membrane patterns. *Bioinformatics*, **30**, 2644–2651.
- Ester, M. *et al.* (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, pp. 226–231.
- Ferrezuelo, F. *et al.* (2012) The critical size is set at a single-cell level by growth rate to attain homeostasis and adaptation. *Nat. Commun.*, **3**, 1012.
- Gordon, A. *et al.* (2007) Single-cell quantification of molecules and rates using open-source microscope-based cytometry. *Nat. Methods*, **4**, 175–181.
- Gupta, A. *et al.* (2016) Synthetic data for text localisation in natural images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2315–2324.
- He, K. *et al.* (2020) Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.*, **42**, 386–397.
- Jönsson, H. and Levchenko, A. (2005) An explicit spatial model of yeast micro-colony growth. *Multiscale Model. Simul.*, **3**, 346–361.
- Lee, S.S. *et al.* (2012) Whole lifespan microscopic observation of budding yeast aging through a microfluidic dissection platform. *Proc. Natl. Acad. Sci. USA*, **109**, 4916–4920.
- Lu, A.X. *et al.* (2019) YeastSpotter: accurate and parameter-free web segmentation for microscopy images of yeast cells. *Bioinformatics*, **35**, 4525–4527.
- Moen, E. *et al.* (2019) Deep learning for cellular image analysis. *Nat. Methods*, **16**, 1233–1246.
- Pelet, S. *et al.* (2012) An integrated image analysis platform to quantify signal transduction in single cells. *Integr. Biol.*, **4**, 1274–1282.
- Prangemeier, T. *et al.* (2020a) Attention-based transformers for instance segmentation of cells in microstructures. In: *2020 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 700–707.
- Prangemeier, T. *et al.* (2020b) Multiclass yeast segmentation in microstructured environments with deep learning. In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2020)*. Institute of Electrical and Electronics Engineers Inc.
- Ronneberger, O. *et al.* (2015) U-Net: convolutional networks for biomedical image segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds.), *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*. Springer International Publishing, Cham, pp. 234–241.
- Salem, D. *et al.* (2021) YeastNet: deep-learning-enabled accurate segmentation of budding yeast cells in bright-field microscopy. *Appl. Sci.*, **11**, 2692.
- Toda, Y. *et al.* (2020) Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Commun. Biol.*, **3**, 173.
- Uhlendorf, J. *et al.* (2012) Long-term model predictive control of gene expression at the population and single-cell levels. *Proc. Natl. Acad. Sci. USA*, **109**, 14271–14276.
- Versari, C. *et al.* (2017) Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the Evaluation Platform. *J. R. Soc. Interface*, **14**, 20160705.
- Vicar, T. *et al.* (2019) Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison. *BMC Bioinform.*, **20**, 360.
- Voigtlaender, P. *et al.* (2019) MOTS: multi-object tracking and segmentation. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, pp. 7934–7943.
- Wang, Q. *et al.* (2009) Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. *Cytometry Part A*, **77**, 101–110.
- Wang, Y. *et al.* (2017) A modeling study of budding yeast colony formation and its relationship to budding pattern and aging. *PLoS Comput. Biol.*, **13**, e1005843.
- Wood, N.E. and Doncic, A. (2019) A fully-automated, robust, and versatile algorithm for long-term budding yeast segmentation and tracking. *PLoS One*, **14**, e0206395.