

Brightfield Cell Segmentation Without Labels or Learning

Xénia Richnáková^{1,*}, Viktória Hodorová² and Jiří Hladůvka¹

¹Faculty of Mathematics, Physics and Informatics, Comenius University Bratislava

²Faculty of Natural Sciences, Comenius University Bratislava

Abstract

Deep learning dominates image segmentation, but its reliance on large labeled datasets and GPU resources may limit its applicability. We introduce a lightweight, handcrafted pipeline tailored for brightfield microscopy in settings with scarce to no labeled data. Our method employs classical image-processing techniques to segment cells, eliminating the need for manual annotations or training.

Visual inspection and comparison with the deep-learning model Cellpose [1] confirm reliable performance across a variety of imaging conditions. Notably, our approach maintains high-quality, smooth boundaries by operating at full resolution, avoiding the jagged edges introduced by downsampling and upscaling required by Cellpose. The pipeline is interpretable, fast, and resource-efficient, offering a practical alternative when deep models are either unnecessary or infeasible.

Keywords

cell segmentation, handcrafted pipeline, image processing, resource efficiency

1. Introduction

Cell segmentation is a fundamental task in bioimage analysis, enabling downstream applications such as cell counting, morphology quantification, tracking, or texture-based cell analysis. While deep learning has become the dominant approach to segmentation in recent years, its adoption often comes with practical barriers. Whether trained from scratch or adapted from pre-trained models, deep networks typically require large amounts of annotated data and access to GPU resources – both of which may be unavailable in laboratory settings.

Brightfield images are often acquired alongside other modalities such as fluorescence, phase contrast, or confocal channels. In our setting, for example, segmentation based on brightfield images (Figure 1a) is preferable, as fluorescence channels (Figure 1b) often exhibit blurred or less distinct cell boundaries. This highlights the importance of robust, label-free segmentation methods that rely only on brightfield input.

We present a lightweight, handcrafted pipeline for cell segmentation in brightfield images, designed specifically for contexts where manual cell annotations are unavailable. Our approach uses classical image processing techniques such as filtering, thresholding, and morphological operations to segment cells without the need for training or annotations. Designed with interpretability and simplicity in mind, the pipeline runs efficiently on standard hardware and demonstrates reliable segmentation across diverse conditions, as confirmed by visual inspection. It offers a practical alternative in settings where deep learning is unnecessary, infeasible, or undesirable.

2. Materials

Magnusiomyces magnusii CBS 234.85 is an ascomycetous yeast belonging to the family Dipodascaceae (subphylum Saccharomycotina). Its giant cells containing multiple nuclei make this species a suitable model organism for cell biology studies.

ITAT'25: Information Technologies – Applications and Theory, September 26-30, 2025, Telgárt, Slovakia

*Corresponding author.

✉ richnakova4@uniba.sk (X. Richnáková); hodorova4@uniba.sk (V. Hodorová); jiri.hladuvka@uniba.sk (J. Hladůvka)

ORCID 0000-0003-4152-8744 (V. Hodorová); 0000-0002-5947-8813 (J. Hladůvka)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

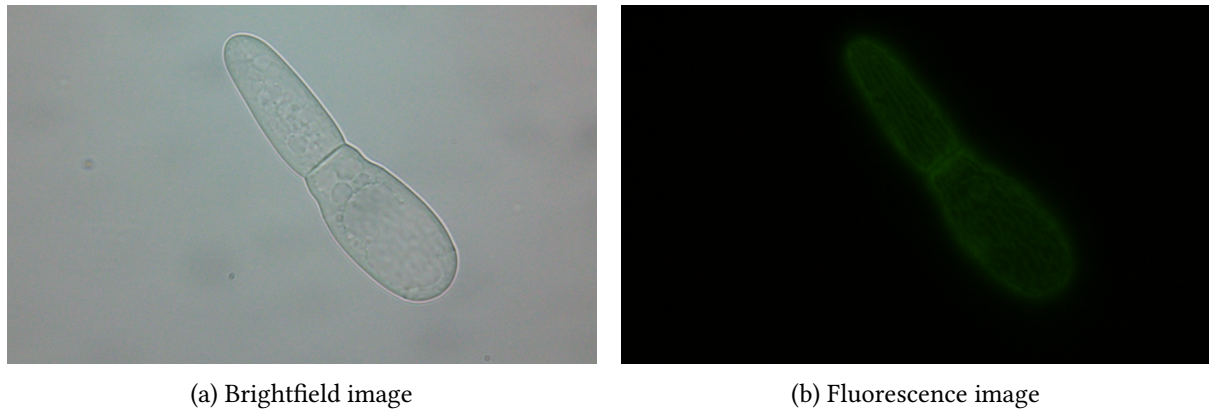


Figure 1: Comparison of Brightfield vs. Fluorescence.

For microscopy, the cultures of *M. magnusii* CBS 234.85 were grown overnight in yeast extract-peptone-dextrose (YPD) medium (1% [wt/vol] yeast extract, 2% [wt/vol] peptone, 1% [wt/vol] glucose) at 28 °C on a rotary shaker. Cells were observed in visible light using an Olympus BX50 microscope and a Zeiss Axio Imager.Z2 microscope.

On the Olympus BX50, images were captured at 1920×1200 resolution with a pixel size of $0.059 \mu\text{m}$ in true-color RGB (8 bits per channel). On the Zeiss Axio Imager.Z2, images were acquired at 2752×2208 resolution as 16-bit grayscale frames at a pixel size of $0.045 \mu\text{m}$.

3. Methods

3.1. Challenges of Automated Cell Segmentation

Cell segmentation from images in the brightfield microscopy is not a trivial task, particularly for the following reasons:

- **Irregular cell shape and size.** Cells can vary significantly at different stages of their life cycle, so classic preset mask shapes often do not suffice.
- **Noise and background artifacts.** Biological samples often contain tissue residues, small debris, or dust, which can be mistakenly identified as cells during binarization.
- **Uneven illumination.** The sample may not be evenly illuminated, leading to variable contrast—some cells appear darker, others are significantly overexposed (Figure 4) .
- **Cell overlap.** In dense cultures or tissues, cells can partially or completely overlap, complicating the extraction of contours.

Our goal, is to avoid deep learning approaches, which imply a black-box solution. We aim to develop a robust pipeline that:

1. automatically handles fluctuating illumination and contrast (using adaptive thresholding),
2. removes noise and artifacts without losing detail at cell boundaries,
3. refines object contours and fills internal cavities to produce binary masks with correct separation of cells from background.

3.2. Cell Segmentation Methods

The chosen segmentation methodology combines adaptive thresholding with morphological image processing. The entire process can be divided into several steps:

1. Image preprocessing - noise smoothing and normalization (as detailed in section 4)
2. Local thresholding to create a binary mask
3. Morphological mask cleaning (removal of small objects, hole filling, gap closing)
4. Identification and measurement of properties of individual segmented objects
5. Selection of the segment that best corresponds to the target cell (the most regular object)
6. Adaptive threshold refinement to improve the segmented shape (if necessary)

3.3. Thresholding

Thresholding is a fundamental segmentation technique: it converts a grayscale image into a binary one based on a threshold value. Initially, we experimented with Otsu's thresholding [2] — a global method that selects a single threshold by maximizing the between-class variance of the image histogram.

However, because Otsu's method applies one fixed threshold to the entire image, it cannot cope with the local variations in illumination and contrast present in our micrographs, and thus produced less satisfying results than adaptive (local) thresholding.

Adaptive thresholding computes a threshold $T(x, y)$ independently for each neighborhood around pixel (x, y) . In this work, we employ the Niblack's thresholding method [3] which is well suited for inhomogeneous backgrounds [4]:

Niblack threshold $T(x, y)$ is computed by:

$$T(x, y) = m(x, y) + k \cdot s(x, y) \quad (1)$$

where $m(x, y)$ is the local mean intensity (brightness) and $s(x, y)$ is the local standard deviation within a window around (x, y) . The parameter k allows shifting the threshold relative to the mean—usually downward into darker regions. While for text documents Niblack recommended a negative k to capture dark letters on a light background, in the context of cell images it is more appropriate to choose a positive k . This is because a positive value slightly raises the threshold above the local mean, helping to reduce false background merging while still preserving the weaker (darker) parts of the cell as foreground. In our case, we empirically set $k = 0.2$ within a 15×15 -pixel window. This parameter can be optionally adjusted (e.g., increased or decreased by 0.1) whenever the cell boundary is not fully enclosed.

The resulting local threshold values are stored in matrix $T(x, y)$ for each pixel. By comparing the original image to this matrix, a binary image (mask) is produced, in which pixels with intensity above $T(x, y)$ are marked as foreground (cell) and the rest as background.

Adaptive thresholding using Niblack's method ensured that the cells were correctly detected even under uneven illumination, unlike the global Otsu threshold, which in our tests either missed the weaker parts of the cells or included excessive background noise. This switch from Otsu to Niblack thresholding was therefore crucial and significantly improved mask quality.

3.4. Morphological Operations

The raw binary mask after thresholding usually contains small artifacts (isolated pixels or small noise regions) and may have holes within objects. To clean the mask, we apply a set of morphological operations:

- (a) removal of small objects
- (b) filling of small holes
- (c) closing of narrow gaps
- (d) optional dilation

Operations (a) and (b) were carried out by removing all connected components smaller than a given area or by filling all holes smaller than a given area, respectively. For our images, we chose a threshold of 500 pixels, based on empirical tuning. This value reliably removed small specks and background noise without eliminating faintly cells.

Next, we applied the morphological closing [5] with a disk-shaped structuring element of radius 3 px. Closing corresponds to dilation followed by erosion; its effect is to join nearby regions and close narrow gaps or slits within objects. This was followed by dilation with a disk of radius 1 px, slightly expanding the resulting cell outline.

These steps aim to obtain as clean mask as possible, where each cell is represented by a single white area without small protrusions or holes.

3.5. Segment Analysis and Object Selection

After morphological cleaning, the next step is to identify individual segmented objects and select those corresponding to target cells. The cleaned binary mask may contain several connected components.

Each connected object is assigned a unique label. Then, for each object, a set of properties is computed. The most important features in our logic are the object's area, the area of its convex hull, and the object's Euler number.

- **Euler number** E of a connected region is defined as one minus the number of holes it contains. An ideal cell segment should therefore have the Euler number equal to 1 [6].
- **Convex area** The area of the convex hull, i.e. the area of the smallest convex polygon that contains (encloses) the region [7].

Feature Usage To select the segment representing the target cell, we assume the cell will be among the larger objects in the mask and have a relatively compact (regular) shape. We first filter the list of all objects by convex area larger than 15000 px, thus ignoring segments which we are not considered as whole cells.

The remaining objects are then sorted. Preference is given to the object with an Euler number closest to 1, and if there are multiple such objects, the one with the largest convex area is chosen. Such an object is considered the most regular – it has a large area and almost no holes, which roughly corresponds to the characteristics of a fully segmented cell.

This procedure yields the label of the selected object, which is then marked as the candidate for the final cell mask. If there is only one cell in the frame, this algorithm naturally selects it (if it meets the minimum size); if there are multiple cells or fragments, it automatically chooses the one with the most intact shape.

3.6. Adaptive Threshold Refinement

After selecting the best object (cell), an additional threshold refinement step is added to remove remaining imperfections, especially if the selected segment contains holes. If the selected object's Euler number differs from 1, the algorithm attempts to fill the hole by further adjusting the threshold. Adaptation consists of iteratively reducing the offset responsible for noise suppression.

We originally set this Δ to 0.05 (5 % intensity). By increasing Δ , we tighten the condition for foreground pixels – a larger Δ means a higher intensity above the threshold is required, and conversely, by decreasing Δ we allow slightly darker parts to pass into the foreground.

This refinement process consists of iteratively decreasing the offset Δ by 0.005 in each step and re-segmenting the image with the new threshold value; iterations continue as long as the selected segment's Euler number improves. Specifically, if the segment had $E = 0$, after a slight decrease of Δ that hole can be filled (the part that originally had slightly lower intensity than the original threshold now becomes white) and the Euler number increments. The algorithm checks whether the object's Euler number approaches 1; if not, it proceeds to another decrease of Δ .

This iterative process stops when the Euler number reaches 1 or when further decreasing Δ no longer improves the Euler number, or when Δ reaches the minimum set value (0.005 in our implementation). This adaptive refinement ensures that the final cell mask is as compact as possible and contains no internal holes caused by an overly strict threshold.

4. Implementation

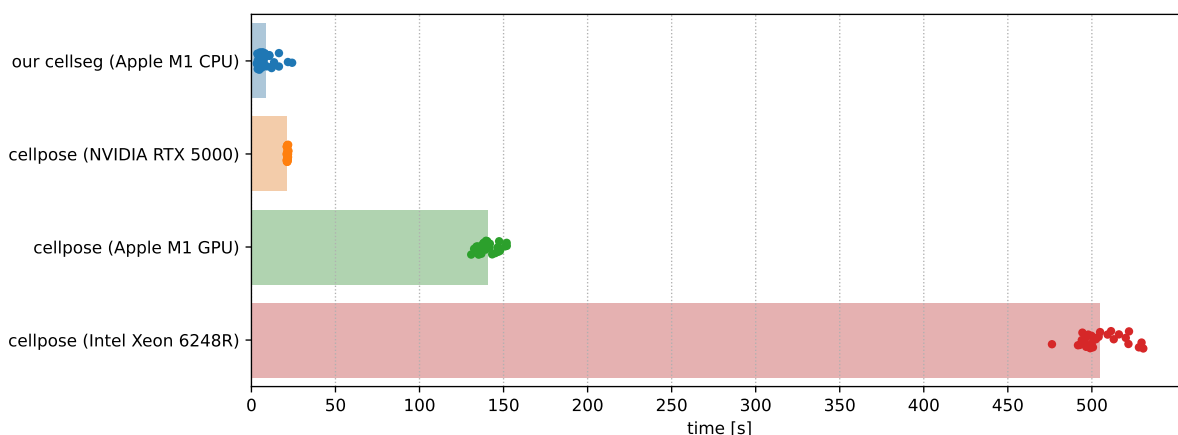


Figure 2: Runtime comparison of our tool with cellpose across hardware configurations. Top-to-down mean times in seconds: 8.7 ± 5.4 , 21.3 ± 0.2 , 140.6 ± 5.8 , 505.0 ± 12.9 .

The implementation is carried out in Python using the *scikit-image* and SciPy/NumPy libraries. The main script processes either a single image (*.tif*, *.png* or *.czi*) or an entire folder of images, and saves or visualizes the resulting masks.

1. **Loading and preprocessing:** Depending on the format, the image is loaded and converted to grayscale (*rgb2gray*). A Gaussian filter [8] (*filters.gaussian*, $\sigma \approx 0.4$ – 1.6 adjusted to specific microscope) is applied and the result is normalized to $[0, 1]$ via min–max scaling, so that the darkest pixel maps to 0 and the brightest to 1. Smaller values of σ gently suppress high-frequency noise while preserving fine cell-boundary detail, whereas larger values more aggressively smooth background variations.
2. **Adaptive thresholding:** The smoothed image is thresholded with *threshold_niblack* (window 15×15), producing the binary mask `binary = gray_smooth > thresh`.
3. **Morphological cleaning:** The binary mask is cleaned in sequence:
 - `remove_small_objects(binary, min_size=500)` – removal of noise,
 - `remove_small_holes(..., area_threshold=500)` – hole filling,
 - `binary_closing(..., disk(3))` – closing of narrow gaps,
 - `binary_dilation(..., disk(1))` – slight outline dilation.
4. **Labeling and segment selection:** Connected components are labeled (using *ndimage.label* of SciPy), and for each component we compute region properties (via *regionprops* of *skimage*).

Among the components with convex area ≥ 15000 , we select the one whose Euler number is closest to 1.

5. **Threshold refinement:** If the selected segment still contains holes ($E < 1$), the offset Δ is iteratively decreased by 0.005 during thresholding and the mask is re-cleaned until the Euler number improves or the offset reaches 0.005.

5. Results

We tested the proposed method on multiple brightfield cell images from two microscopes of varying quality and content - images with a single dominant cell, images with multiple cells or fragments in the background, images with uneven illumination. In most cases the algorithm successfully identified the cell regions and produced plausible segmentation masks (see Figures 3a, 3c, 4a, and 4e).

We consider an output successfully segmented if the mask follows the contours of all cells in the original image, with no disconnected regions within cell outline. Examples when this is not fulfilled can be seen in Figure 3e (triangular shape between the 3 cells) and 4c (the central cell not segmented correctly).

Based on this criterion and visual inspection, out of 50 images, 46 were successfully segmented, indicating a success rate of 92%.

For those images that were not correctly segmented on the first attempt, one can fine-tune the k parameter in Niblack thresholding or adjust the σ value in the Gaussian filter to achieve proper segmentation.

5.1. Comparison with Cellpose

As we lack ground-truth segmentations, standard quantitative validation is not possible. Nonetheless, we assessed how consistently our handcrafted segmentation pipeline agrees with a pre-trained deep-learning, state-of-the-art model.

Cellpose [1] is a widely used generalist segmentation model designed to work out-of-the-box on various microscopy images without retraining or parameter tuning. It offers both a GUI and API suited for high-throughput analysis.

Although neither our nor Cellpose segmentation clearly represents the ground truth, comparing them can offer valuable insights into their relative behavior and agreement.

We quantified this agreement using the Intersection over Union (IoU), also known as the Jaccard index [9]. Let \mathbf{O} and \mathbf{C} denote the sets of segmentation pixel coordinates of our method and Cellpose, respectively. Then:

$$IoU(\mathbf{O}, \mathbf{C}) = \frac{|\mathbf{O} \cap \mathbf{C}|}{|\mathbf{O} \cup \mathbf{C}|} \in \langle 0, 1 \rangle \quad (2)$$

This metrics yields values from the unit interval with 1 indicating a perfect overlap. In many biomedical segmentation benchmarks, IoUs in the 0.85–0.95 range indicate excellent agreement.

Across the dataset, we observed a high average intersection over union (IoU) of 0.91 (standard deviation: 0.17), indicating that both approaches segment largely overlapping regions. This high concordance suggests that our handcrafted pipeline produces results comparable in coverage to a pre-trained deep network, while retaining advantages in interpretability, computational simplicity, and independence from labeled training data

Further visual inspection revealed that Cellpose tends to produce suboptimal segmentation on our full-resolution scans. Plausible segmentations were only achieved after appropriate downsampling, whether via image pyramids or by specifying the average cell diameter. While downsampling typically improves Cellpose’s ability to delineate cell bodies, rescaling the resulting masks back to original resolution introduces jagged, pixelated boundaries, particularly evident when zooming into Figures 3 and 4. Thus, the segmentation boundary smoothness becomes a trade-off against detection plausibility.

In contrast, our handcrafted pipeline operates natively at full resolution, producing smooth, high-quality boundaries without requiring rescaling. This demonstrates a clear advantage in preserving boundary integrity – suggesting that when high-resolution delineation is important, our method may offer more reliable results.

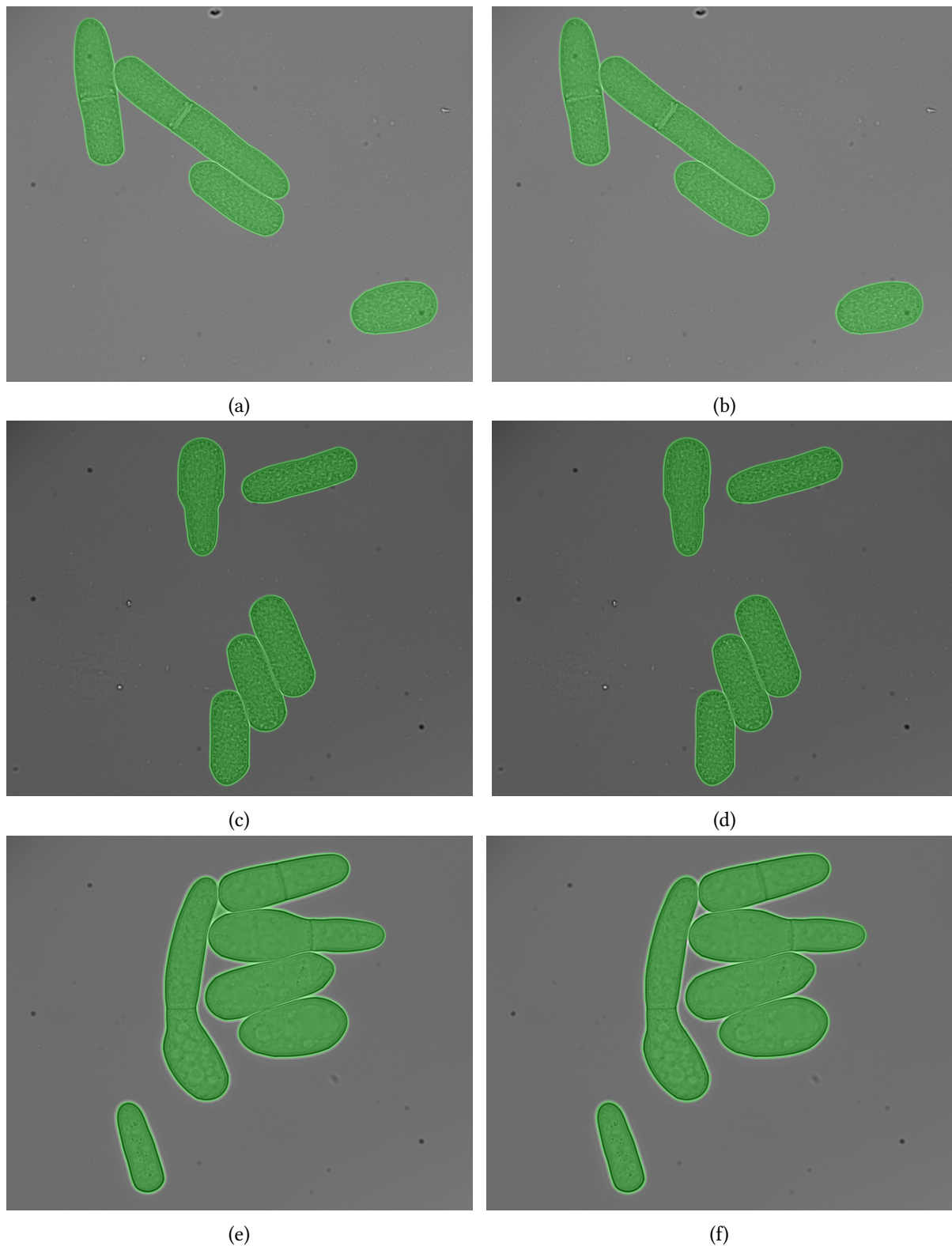
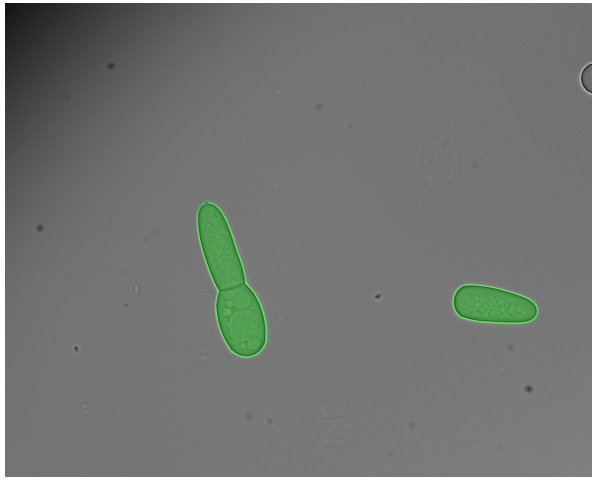
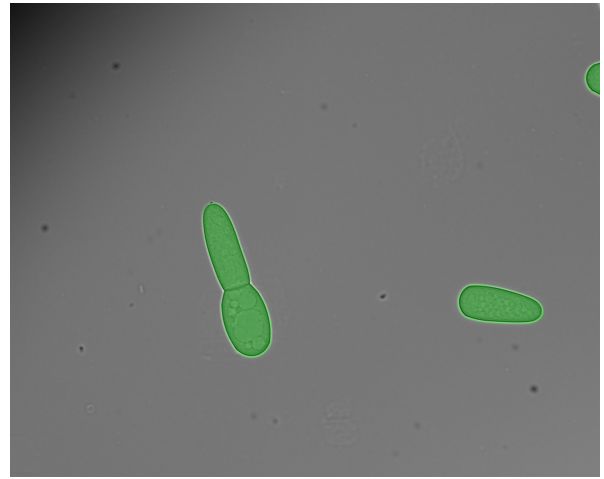


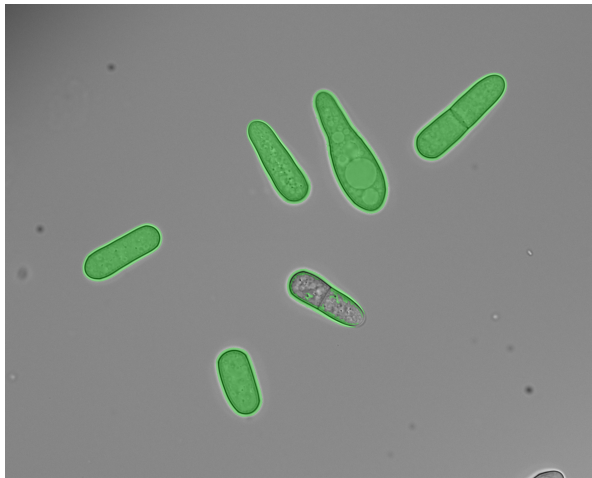
Figure 3: Segmentation examples where our algorithm (a,c,e) and Cellpose [1] (b,d,f) are in concordance. The IoU for the three examples are 0.99, 0.99, and 0.98, respectively.



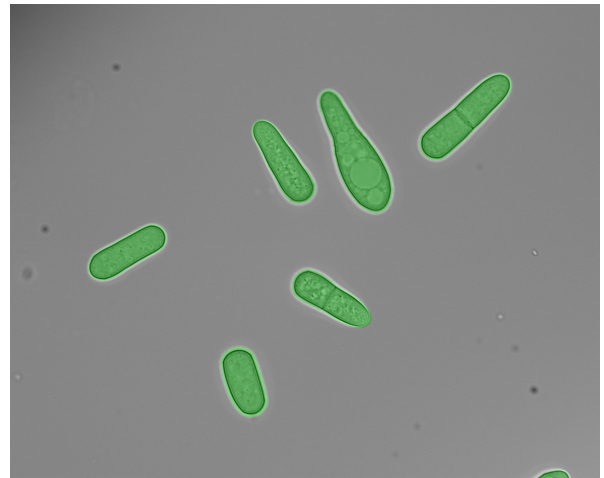
(a)



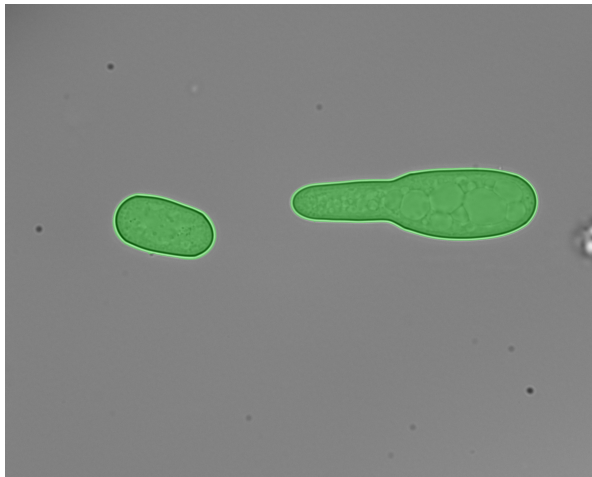
(b)



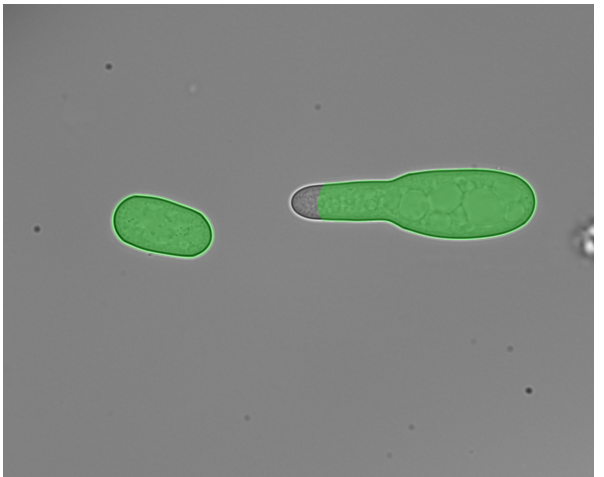
(c)



(d)



(e)



(f)

Figure 4: Segmentation examples where our algorithm and Cellpose [1] differ and eventually fail. Top-to-down IoU scores are 0.91, 0.84, and 0.92, respectively. Cellpose may produce partial or fragmented masks at image borders (b),(d), or even under-segment (f). Our method may be sensitive to intensity gradient along the cell boundary (c).

6. Discussion, Conclusions, and Future Work

When deploying Cellpose in real-world biomedical labs, several practical limitations become evident.

First, GPU acceleration is a necessity (see Figure 2). On CPU alone, processing a single high-resolution image can take over 500 seconds, which is impractically slow for routine tasks. Even using GPUs, performance disparities are significant: on Apple M1 hardware, it still takes more than 140 seconds per image, whereas the pricey high-end NVIDIA RTX-5000 brings this down to around 20 seconds.

Second, despite its generalist claim, Cellpose still may require parameter tuning or downsampling large scans to achieve optimal segmentation for specific samples.

Furthermore, the high computational demands and the need for manual parameter adjustments make Cellpose inadequate for fast-paced, GPU-limited laboratory environments.

On the other hand, our handcrafted brightfield segmentation pipeline offers several compelling benefits over Cellpose when applied to high-resolution microscopy images.

Full-resolution precision Unlike Cellpose – which downscales inputs to match its trained cell sizes – our pipeline operates natively at full resolution, preserving smooth, accurate cell boundaries without jagged artifacts.

Resource efficiency Our pipeline runs on standard CPU hardware, requires no GPU and performs well without extensive parameter tuning. This is in contrast to Cellpose, which demands GPU acceleration.

Interpretability and flexibility Based on classical image processing (filters, thresholding, morphology), our approach is transparent and customizable. In fact there are currently two parameters to adjust, should the segmentation be implausible. Cellpose on the other hand relies on a complex, hardly-interpretable deep model.

Strong agreement Despite its simplicity, our method achieves high overlap with Cellpose segmentations, demonstrating it produces biologically plausible masks comparable to a state-of-the-art pretrained deep network.

Overall, we show that for brightfield microscopy with limited or even no ground truth segmentation, a well-tuned handcrafted pipeline can deliver segmentation quality on par with deep learning – while being faster, more transparent, and finely preserving boundary detail at native resolution. This makes it a valuable, practical alternative when deep models are unnecessary or unavailable.

Limitations Our method may fail when parts of the cell wall are severely out of focus or exhibit very low local contrast; in such cases the object boundary can be mis-segmented. In many images, this failure mode can be alleviated by modest tuning of the Niblack parameter k and/or the Gaussian smoothing σ prior to thresholding. We also observe occasional under-segmentation in dense, irregular clusters: when neighboring cells are separated by only a narrow or low-contrast gap, this gap may not be detected and adjacent cells may merge into a single component.

Future work We still consider our cell segmentation tool a work in progress that requires further refinement to enhance its usability and practicality. In particular, we foresee that incorporating the image intensity derivatives would precisely define cell walls with an inhomogeneous appearance, such as the central cell in Figure 4c.

Acknowledgments

This work was supported by the European Union’s NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project 09I03-03-V04-00363.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4.5 for grammar and spelling checks.

References

- [1] C. Stringer, T. Wang, M. Michaelos, M. Pachitariu, Cellpose: a generalist algorithm for cellular segmentation, *Nature Methods* 18 (2021) 100–106. doi:10.1038/s41592-020-01018-x.
- [2] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1979) 62–66. doi:10.1109/TSMC.1979.4310076.
- [3] W. Niblack, An introduction to image processing, in: *Prentice-Hall International Series in Computer Graphics and Geometric Modeling*, Prentice-Hall, 1986. Chapter on Local Thresholding.
- [4] P. Nagabhushan, S. Nirmala, Text extraction in complex color document images for enhanced readability, *Intelligent Information Management* 2 (2010) 120–133.
- [5] A. Challa, S. Danda, B. S. Daya Sagar, *Morphological Closing*, Springer International Publishing, Cham, 2020, pp. 1–2. doi:10.1007/978-3-030-26050-7_211-1.
- [6] J. Ohser, W. Nagel, K. Schladitz, The Euler number of discretized sets: On the choice of adjacency in homogeneous lattices, in: K. R. Mecke, D. Stoyan (Eds.), *Morphology of Condensed Matter*, volume 600 of *Lecture Notes in Physics*, Springer, Berlin, Heidelberg, 2002, pp. 275–298.
- [7] R. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Information Processing Letters* 1 (1972) 132–133. doi:10.1016/0020-0190(72)90045-2.
- [8] I. T. Young, L. J. van Vliet, Recursive implementation of the gaussian filter, *Signal Processing* 44 (1995) 139–151. doi:10.1016/0165-1684(95)00020-E.
- [9] A. A. Taha, A. Hanbury, Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool, *BMC Medical Imaging* 15 (2015) 29. doi:10.1186/s12880-015-0068-x.