Automatic segmentation of microscopic cell images

ADRIAN MARIUS MUNTEANU, University of Twente, The Netherlands

Protoplasts are plant cells without cell walls. They provide a unique single-cell system to underpin several aspects of modern biotechnology. They are important in researching fundamental aspects of cell physiology and cell surface interaction with toxins and pathogens. The segmentation of protoplast cell images proves to be a difficult task even for the current state-of-the-art machine learning models. Even though some of them show good results, most lack accuracy when it comes to automatically segmenting cells from an image. A prevalent challenge in cell imaging is the occurrence of out-of-focus blur, which arises because cells are positioned at different levels along the Z-axis on the chamber slide. This study seeks to enhance automatic segmentation accuracy by training a U-net model with microscopy image annotation masks generated using SAM's segmentation tools.

Additional Key Words and Phrases: Image segmentation, microscopic imagery, machine learning, U-net model

1 INTRODUCTION

The research of cells is a vital topic for the accurate assessment of health conditions. Understanding cell behaviour can lead to higher accuracy in the detection of certain diseases and better comprehension of numerous biological processes. For example, protoplasts (plant cells) have the potential to regenerate a cell wall and are proven to be very useful when it comes to creating experimental systems for many kinds of plant genetic manipulation [3]. Normally, cell images are segmented manually by pathologists, but this method is extremely time-consuming. Therefore, the need for a cell segmentation algorithm came up pretty naturally. However, the process of cell segmentation comes as a significant problem due to cells existing in three-dimensional space. This challenge is amplified by the use of multiple focal planes during imaging, leading to varying levels of blurring across cells. As a result, some cells appear more focused while others are entirely blurred. Consider Figures 1 and 2, which depict cells in similar states. A noticeable difference arises between the clarity of the cells taken at various z-levels, showcasing the impact of different focal planes on cell blurring. In this particular case, a z-level 18 (Figure 2) provides a clearer separation of cells.

With the evolution of deep learning techniques, computer vision technologies play an important role in cell segmentation since it is a classical problem of image processing where an image is divided into different regions according to a set of features such as shape, colour or texture. Convolutional neural networks (CNN) attract the most interest since they allow the development of different image segmentation models with high accuracy[21]. One big disadvantage of these networks is that they require large amounts of high-quality labelled data to be trained upon, and in the case of image segmentation, this process tends to be time-consuming.

TScIT 41, July 5,2024, Enschede,The Netherlands

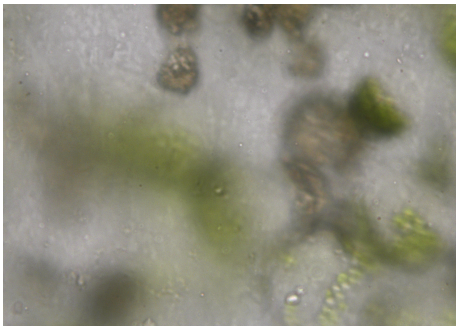


Fig. 1. Cell image taken at z-level 12

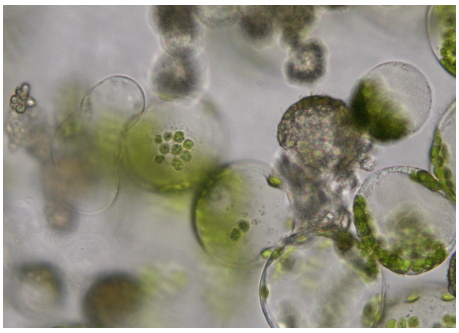
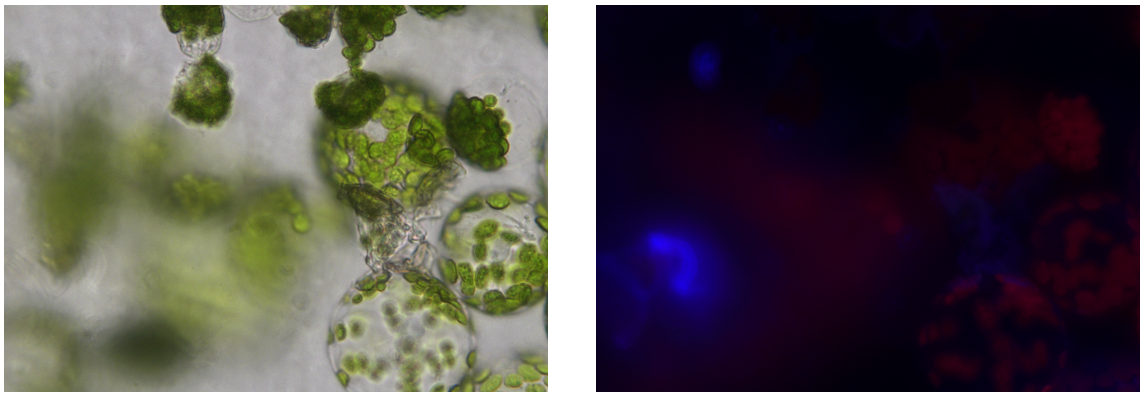


Fig. 2. Cell image taken at z-level 18

Currently, one of the biggest segmentation models is SAM (Segment Anything Model)[8]. Its most important feature is that it allows for instance segmentation rather than semantic segmentation. The latter classifies pixels based on their semantic meaning, treating all objects within the same category as one entity, while the former distinguishes between different objects of the same class, allowing for better identification and differentiation of the object. Moreover, SAM was designed to be a foundation model. It shows great results regarding zero-shot learning, meaning that it manages to segment images from domains it was not trained on. SAM outputs a 2D array mask, where pixels that do not belong to an object are labelled as 0 and pixels that are used in the representation of a segmented instance are labelled with a corresponding numeric value to that instance. Besides having an automatic segmentation function, the model provides an interactive segmentation [11] feature that allows users to use point prompts to highlight the object that should be segmented. The prompts are of two types, positive or negative, and help the model distinguish the region that should be segmented from the image.

2 PROBLEM STATEMENT

This research aims to study various methods to obtain accurate segmentation results from cell images taken at various focal levels.



(a) Bright-field image at z-level 16 (b) Fluorescent image at z-level 16

Fig. 3. Comparison between a bright-field (a) and a fluorescent (b) image

Due to the different focal planes, some cells appear blurred, while others are clearer and more defined. Due to this phenomenon, there is no way to create an annotation mask based only on one single image. Moreover, low contrast and blurred edges have always caused problems affecting cell segmentation performance. There were also issues in detecting the edges of cells that overlap with each other. One of the advantages of fluorescent images is that they clearly show the outline of the cells, ignoring other structures that can be found in a bright-field image, thus simplifying the segmentation task. For example, Figure 3 shows this difference. As you can see, the fluorescent image is darker and shows some blue outlines that represent the cell walls and ignores the green chloroplast formations that can be seen in the bright-field image. Therefore, it makes the job easier for the SAM model to predict the location of the cells.

Figure 4 shows the output of the automatic segmentation using the "vit-b" image encoder of the SAM model and Figure 5 the results of the labelled image using the interactive segmentation feature of SAM that shows our correct masks. It is evident that there is a significant discrepancy between the two figures; the automatic segmentation algorithm failed to deliver an accurate outcome, resulting in the image being divided into an excessive number of objects. U-Net [18] is one of the most popular machine learning architectures used for numerous segmentation tasks. Various research has shown the U-Net model's efficacy, establishing it as one of the most commonly used models with exceptional performance in cell segmentation. As stated in [12], fine-tuning hyper-parameters and employing efficient image pre-processing techniques can allow a U-Net model to deliver state-of-the-art results. Consequently, I decided to train my own U-Net model to assess its performance in my particular case.

2.1 Research Questions:

The main questions that this research aims to answer are:

(1) **How to effectively generate annotation masks for cells from multimodal images taken on different focal planes using interactive segmentation tools?**

(2) **What is the effectiveness of automatic semantic segmentation of a trained U-net model on cell images taken on different focal planes?**

3 RELATED WORK

In this section, we analyze previous work done on image segmentation and specifically cell segmentation. In 2022, Tingxi et. al.[21] reviewed most of the techniques used for image segmentation. The methods are split into two categories: traditional methods and deep learning methods.

3.1 Traditional Methods

3.1.1 Threshold segmentation method[14]. Represents a simple method of separating objects from their background. It involves setting a threshold value. Pixels in the image are given intensity values and the ones that are above the threshold are classified as belonging to one group, while those below it belong to another (typically the background). This method proved to be extremely accurate when there is a big contrast between the object of interest and the background.

3.1.2 Region growing method[6]. This method relies on grouping neighboring pixels based on a certain criterion such as intensity or colour. The algorithm starts from a given pixel or region, and then neighbouring pixels are iteratively added to the region if they meet the criteria. This technique is effective for segmenting regions with homogeneous properties.

3.1.3 Active contour segmentation[16]. This process outlines object boundaries in images, thus resolving image overlaps effectively. It works by initializing a curve near the object boundary and then iteratively deforming it to fit the boundary. The contour is attracted to features such as edges or intensity gradients, while also resisting factors such as noise or occlusions. The deformation is guided by energy minimization principles, where an energy function is defined to balance between fitting the contour to the image features and maintaining smoothness.

Fig. 4. Automatic segmentation result

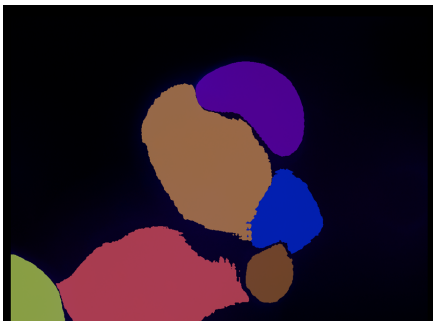


Fig. 5. Interactive segmentation result

3.2 Deep learning methods and cell segmentation strategies

These methods leverage Convolutional Neural Networks[15] for analyzing pathological images. CNN processes images through multiple layers trying to extract the crucial features for classification or segmentation such as edges, shapes, and various patterns. Architectures like AlexNet[9] and VGG[19] pioneered image classification while FCNs[13] and U-Net[18] revolutionized image segmentation, particularly in medical imaging. CNN-based models excel over traditional approaches significantly benefiting medical research by providing more precise cell image segmentation results. A notable method for cell segmentation was presented by C. Hernández et.al.[5]. He described a two-step process using convolutional neural networks to count cells in microscopy images. Firstly a feature pyramid network(FPN)[10] is used to generate a cell mask. The FPN is designed to learn features at different scales, which improves the accuracy of the segmentation mask. Secondly, a VGG-11-based network is used to count the cells in the foreground mask generated in the previous step. The authors achieved impressive results having over 80% of ground truth counts fall within the model's predicted 95% interval.

4 METHODOLOGY

In this section, we will present and discuss each step of our process. Figure 6 represents the workflow diagram of our methodology.

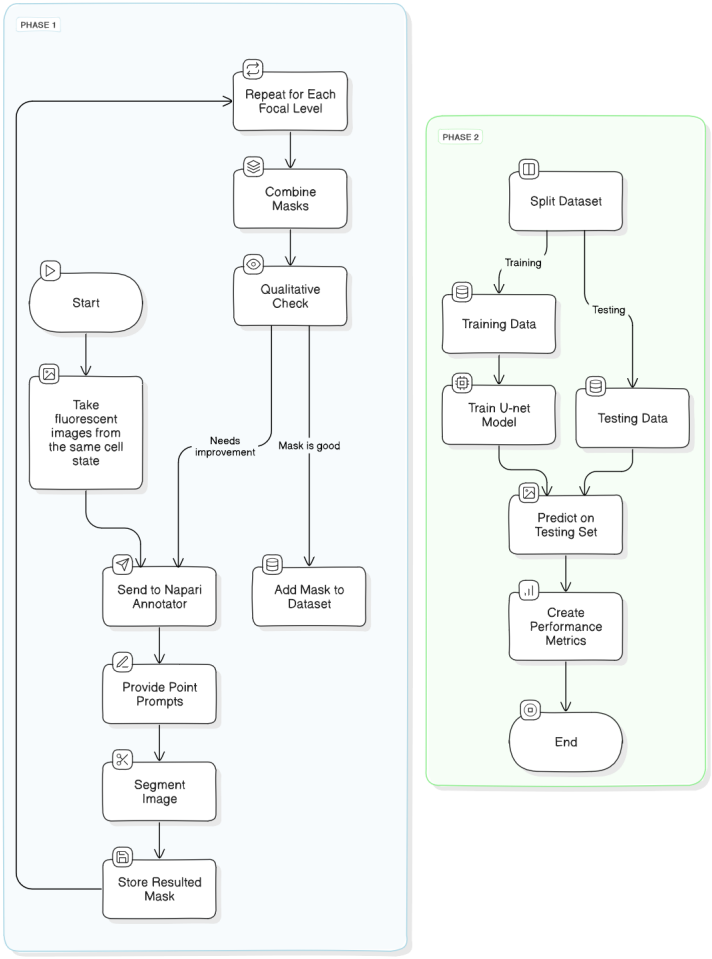


Fig. 6. Workflow diagram

4.1 Creating the annotation masks

As seen in Figure 6 the first phase in our research starts by creating the annotation masks for our dataset. They will act as a ground truth in the training process of the model. The dataset contains brightfield and fluorescent cell images taken on 7 different focal planes for every specific state. The images are systematically named based on key acquisition parameters:

- **Date**: The day the image was captured (e.g., "day1").
- **Time Point**: Indicates the specific time point within the experiment (e.g. "t1").
- **Imaging Modality**:Denoted by "c" with "c1" representing bright-field imaging and "c2" representing fluorescent imaging.
- **XY Position**: The location of the cells within the sample, based on the x and y axes of the microscope stage (e.g., "xy10").
- **Z-Stack Level**: The specific focal plane captured within the z-stack (e.g., "z15").

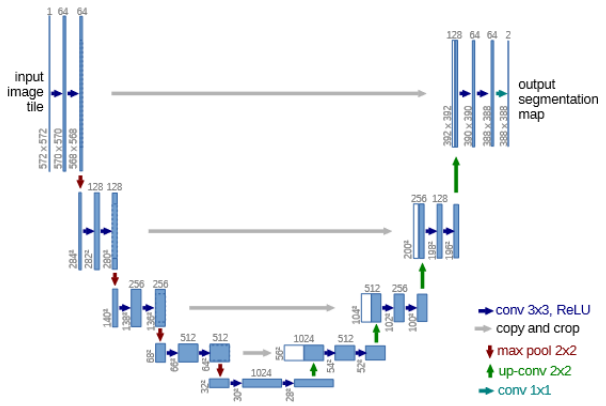


Fig. 7. U-net architecture diagram [18]

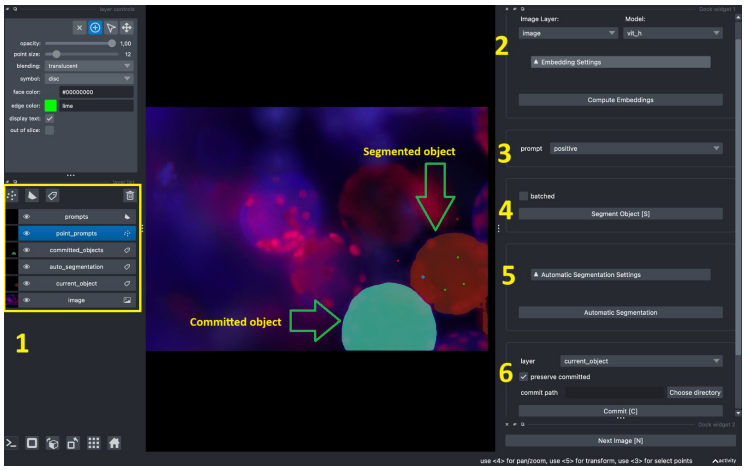


Fig. 8. 2d-annotator from micro-sam [2]

Thus, an image named "day1t1c1xy10z15" represents a bright-field image of cells taken on day 1 of the experiment, at time point 1, at position 10 on the XY stage, and at focal plane 15 within the z-stack. For the dataset, we acquired 2030 images (1015 fluorescent and 1015 bright-field), which correspond to 145 unique states

$$(1015\ unique\ images) \div 7\ different\ focal\ planes = 145\ unique\ states$$

To create the masks, we use the 2D Micro-SAM annotator [1]. Micro-SAM provides an annotator built as a Napari [20] plugin using Meta's SAM model. It also contains new fine-tuned models based on publicly available microscopy data, which are also available on BioImage.IO [1], however for this research we will stick to using only the SAM's "vit-h" model. The reason we choose to use the annotator is for its simplicity. Figure 8 depicts an image of the annotator itself. It contains the following features:

(1) The napari layers for the segmentations and prompts:
   - **prompts**: Shape layer that is used to provide box prompts to the model.
   - **point_prompts**: point layer that is used to provide point prompts to the model. Positive points(green) for marking the object you want to segment and negative points (red) to mark the background.
   - **committed_objects**: label layer that contains objects that have been segmented.
   - **auto_segmentation**: label layer that contains results from the automatic instance segmentation operation.
   - **current_object**: label layer for the object that is currently being segmented.
(2) The embedding menu. Used to select the image that has to be processed and the model that you want to use to make the segmentation and compute the embeddings.
(3) The prompt menu. Allows for choosing the type of the currently selected point prompt.(positive or negative)
(4) Segmentation button. By clicking the "Segment object" the segmentation process will run based on the given prompts. The result is displayed in the **current_object layer**.
(5) Automatic Segmentation menu. Clicking the "automatic segmentation" button will segment all objects detected in the image. The results will be displayed in the **auto_segmentation** layer. Additionally, different parameters can be tuned directly from the menu.
(6) The menu for committing the segmentation. When clicking "Commit", the results from the selected layer (**current_object** or **automatic_segmentation**) will be transferred to the **commited_objects** layer. When a "commit_path" is given, the results will be automatically stored there.

The annotation procedure will focus exclusively on fluorescent images, leveraging their ability to outline important features of cell structures while disregarding irrelevant elements that are present in the bright-field images. The annotator allows images to be processed in batches, making it easier to annotate images from the same state. After annotating the masks, they will be combined into one general mask reflecting the ground truth for the corresponding cell state. Every mask will be saved as a *tiff* file. Moreover, each cell instance will also be saved as an array, allowing faster quality improvements on the masks.

4.2 Formatting of the dataset

With the generation of the mask completed, we proceed to the second phase of our research. In previous sections we mentioned the two different types of images, recall the bright-field and fluorescent images. Our goal is to determine if the fluorescent images can have a positive impact on the results. Therefore, we decide to build two separate datasets, once containing only bright-field images, and a second one with both bright-field and fluorescent images.

Images at the same xy level were organized into distinct folders since they all correspond to the same annotation mask. To ensure effective training, the data set was split based on the xy location. Randomly splitting the data is not advisable as it could result in images from the same state appearing in both the training and test sets, potentially causing overfitting. For this study, we designate all

[1]https://bioimage.io/

images from position xy-10 as our test data, while the remaining images will be used for training.

4.3 Training the U-Net Model

U-Net is a deep learning architecture that was introduced to combat the challenge of limited annotated data in the medical field [18]. As we can see in Figure 7, the U-Net model has two main parts, the contracting part in which the image is being downsized and the expansive part in which the image is up-scaled back to the original size. The first part is similar to the feed-forward layers of other CNN networks; the main idea here is for the model to try and detect important features of the image such as edges and corners while downsizing the image to try and capture more abstract features. The expansive part is responsible for decoding the image and up-scaling it back to its original size, while also locating relevant features.

Data Pre-processing. Our images and masks are stored using Python NumPy arrays [4]. NumPy library offers easy-to-use methods to quickly store and load data in arrays. To read the image we use OpenCV[7] library and resize each one of them from $2880X2048$ to a size of $512X512$ using linear interpolation. Afterwards, we normalize the images to the pixel values between [0,1].

The masks underwent a similar process. After resizing, the resulting masks are converted into binary masks by setting all pixels greater than 0 equal to 1, and the rest to 0. The conversion to binary masks is a necessary step since we are training our model for semantic segmentation.

Training. We trained two U-Net models for each dataset using two different loss functions. The training is done for 50 epochs using the Adam optimizer with a batch size of 16 and a cycling learning rate that starts at $1e-4$ and goes up to $1e-2$. The loss functions used are:

- **Binary cross-entropy** is a commonly used loss function also known as log loss.

$$BCE = -(y * log(pred) + (1-y) * log(1-pred))$$

It compares each of the predicted probabilities with the actual class output, which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value.

- **Mean Squared Error loss** quantifies the extent of the error between the prediction of the machine learning model and the ground truth by taking the average of the squared difference between the predictions and the target values. Squaring the difference leads to a bigger penalty for large errors, making the model more sensitive to outliers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Here, n represents the size of the data, y the true segmentation of an image and $\hat{y}$ the predicted segmentation.

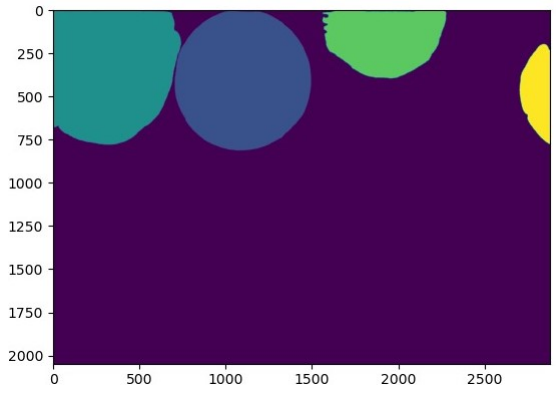


Fig. 9. Final resulted annotation mask

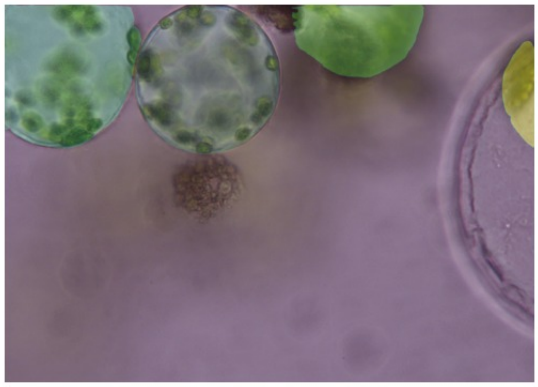


Fig. 10. Overlay of the mask over the cell image for quality check inspection

4.4 Performance metrics

With the training complete, automatic segmentation is performed on the test data using newly trained models. We compare the segmentation results using the following criteria:

- the *Model Accuracy*

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- the *F1 score.*

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- the *Jaccard index (IoU).*

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

5 RESULTS AND DISCUSSION

In this section, we show and discuss the results obtained by our experiment.

5.1 Image annotation

As previously mentioned, a total of 145 annotation masks were meticulously crafted using the Micro-SAM annotation tool in conjunction with the "vit-h" model. As each cell state encompasses seven distinct focal planes (z-12, z-14, z-16, z-18, z-20, z-22, and z-24), the annotation process is strategically organized into batches, each containing seven images representing one focal plane. To ensure the fidelity of the annotations, each mask is visually inspected by overlaying it onto the corresponding original image immediately after its generation. This allows for prompt identification and correction of any misalignment with the cell outlines. An example of the resulting high-quality annotation mask is presented in figure 9. During the annotation process, we encountered challenges regarding the cell overlapping. In certain cases, a larger cell would emerge across different z-levels, partially or completely obscuring another cell. Figure 10 shows an example of such a case. This phenomenon negatively impacts the quality of the masks because the research approach involves using a single mask for all z-levels, allowing models to recognize cells even in blurred and out-of-focus images. In this case, the smaller cells are discarded and only the big cell is added to the final annotation mask. With the help of fluorescent images, a large number of dead cells and foreign structures are discovered to be present in the brightfield images and are not introduced in the final mask with the purpose of teaching the model to ignore them completely.

5.2 U-Net models results

The results of the U-net model can be seen in table 1 and 2. Both tables showcase a comparison between the performance of a machine learning model when using BCE and MSE loss functions across all of the Z-levels.Table 1 shows the results of the model trained on both bright-field and fluorescent images whereas table 2 shows the results of the model trained only on the bright-field images. Just by looking at the two tables, we can see that when training the model on both types of images we get better results no matter the loss functions we use. The biggest difference between the models can be noticed when looking at the F1 scores (3% improvement) and the IoU score(4% improvement). The biggest benefit of training the model on all images is not necessarily the metric's overall growth, but that it will be able to make some predictions when given fluorescent images as well. Next, we will focus on the results from the table 1 since they are better. The overall result shows that regardless of which loss function we use the model performs similarly. However, when using BCE it has a 2% lead in accuracy. The Figures 11 and 12 show an example with some predictions that our model makes. When looking at Figure 12, both models appear to over-segment the image, identifying more regions as cells than are present in the ground truth. Nevertheless, the BCE model's prediction seems more aligned with the ground truth, although it misses some smaller cells. It can be noticed that both models manage to ignore the foreign structures from the top of the image. The low IoU scores of the models also point to the fact that they perform poorly when they try to identify the cells and that there is a low overlap between the prediction and the ground truth mask. Analyzing the performance across the z-levels we find out that the models perform worst on images from z12. This outcome is expected since cells at z12 levels are mostly fully blurred whereas z20 (highest accuracy) shows clearer cell images. Images 1 and 2 show this difference. More results of model prediction can be seen in figures 13,14,15 in the appendix. You can notice how the prediction gets more accurate across the z-levels.

5.3 Discussion and Future Work

After analysing the results achieved using the annotation masks, it is obvious that the model encounters problems predicting overlapping cells. A future research would be to investigate how a Mask R-CNN model would perform using the current annotation masks. Studies show that it can achieve good results when it comes to segmenting overlapping objects[17], therefore it might be a potential candidate for this particular research. An alternative approach to improve the accuracy of ground truth masks involves a change in how they are created and used. Recognizing that larger cells may fully occlude smaller ones at certain z-levels, it becomes necessary to avoid using an annotation mask containing the larger cell for images where it hasn't started appearing. A better strategy for generating annotation masks might be creating individual binary masks for each cell encountered throughout the z-stack. This approach allows for the construction of a unique binary mask at each z-level, seamlessly aggregating the masks of previously identified cells based on their presence in the corresponding image. By adopting this methodology, the limitations of using a single annotation mask per cell state across all z-levels are avoided. Consequently, the ground truth masks become more faithfully representative of the specific image they correspond to, potentially leading to a significant improvement in accuracy.

Z-level	BCE			MSE		
	Accuracy	F1-Score	IoU	Accuracy	F1-Score	IoU
z12	0.68	0.68	0.52	0.67	0.71	0.55
z14	0.70	0.71	0.55	0.69	0.72	0.56
z16	0.72	0.73	0.57	0.71	0.74	0.58
z18	0.73	0.74	0.59	0.71	0.75	0.60
z20	0.74	0.76	0.61	0.72	0.75	0.60
z22	0.73	0.75	0.60	0.71	0.75	0.60
z24	0.71	0.73	0.58	0.69	0.73	0.58
Overall	0.72	0.73	0.58	0.70	0.73	0.58

Table 1. Performance Comparison Across Z-levels of the model trained on all images

6 CONCLUSION

This research is divided into two phases, each attributed to the corresponding research question. The first phase consists of creating annotation masks for a series of microscopic cell images with the help of a state-of-the-art interactive segmentation model. Afterwards, a manual analysis of the quality of the masks and an assessment of whether the SAM model is effective in the creation of the masks. During this part of the research, we answered **RQ1** by showing that it is indeed possible to generate high-quality masks

Z-level	BCE			MSE		
	Accuracy	F1-Score	IoU	Accuracy	F1-Score	IoU
z12	0.69	0.65	0.46	0.69	0.65	0.49
z14	0.71	0.69	0.52	0.71	0.69	0.53
z16	0.73	0.71	0.55	0.72	0.71	0.55
z18	0.73	0.72	0.57	0.73	0.72	0.56
z20	0.73	0.73	0.57	0.72	0.72	0.56
z22	0.72	0.71	0.55	0.70	0.70	0.54
z24	0.70	0.69	0.52	0.70	0.67	0.53
Overall	0.71	0.70	0.54	0.71	0.70	0.54

Table 2. Performance Comparison Across Z-levels of the model trained only on bright-field images

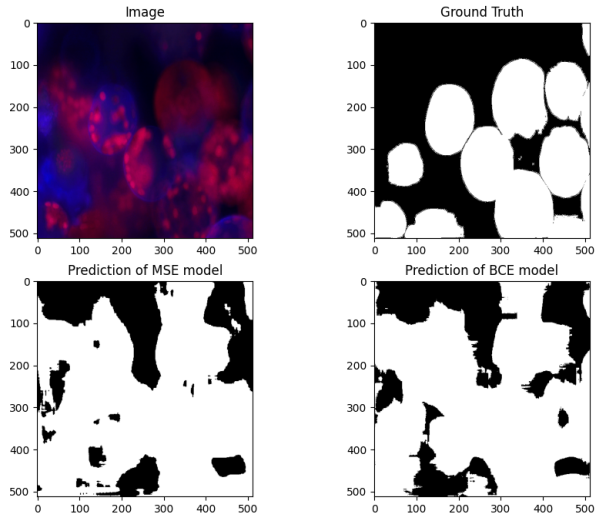


Fig. 11. Prediction of the model on a fluorescent image

using advanced image segmentation models. The second part of our research involves training U-Net models on 2 different datasets (one containing only bright-field images and another one containing fluorescent images as well), to see how well they would perform the automatic segmentation task based on the masks generated in the previous part. To answer our **RQ2**, both models showed similar results, with an overall accuracy of 70%, thus showing that there could be room for improvement. Additionally, we learned that training the model on different types of images does not harm its accuracy, but can actually improve it. The insights gained from this research pave the way for future investigations, including exploring alternative models such as Mask R-CNN and refining annotation strategies to address issues such as cell overlap.

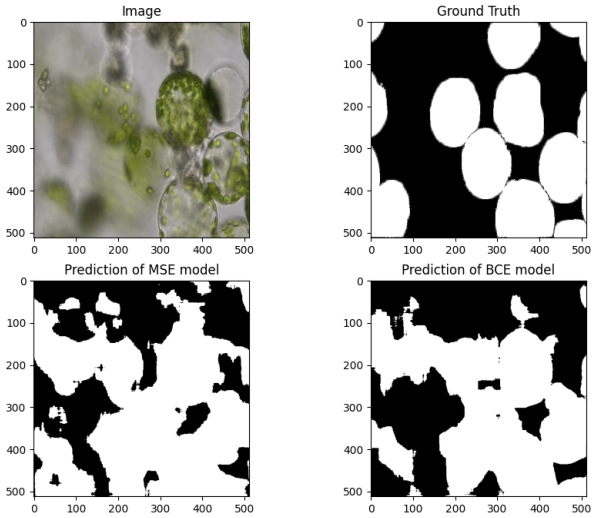


Fig. 12. Prediction of the model on a bright-field image

REFERENCES

[1] Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. 2023. Segment Anything for Microscopy. https://doi.org/10.1101/2023.08.21.554208

[2] Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. 2023. Segment Anything for Microscopy. https://computational-cell-analytics.github.io/micro-sam/micro_sam.html#annotator-2d

[3] Michael R. Davey, Paul Anthony, J. Brian Power, and Kenneth C. Lowe. 2005. Plant protoplasts: status and biotechnological perspectives. *Biotechnology Advances* 23, 2 (March 2005), 131–171. https://doi.org/10.1016/j.biotechadv.2004.09.008

[4] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[5] Carlos X. Hernández, Mohammad M. Sultan, and Vijay S. Pande. 2018. Using Deep Learning for Segmentation and Counting within Microscopy Data. https://doi.org/10.48550/arXiv.1802.10548 arXiv:1802.10548 [cs, q-bio].

[6] S. Hojjat and J Kittler. 1998. Region Growing: A New Approach. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 7 (Feb. 1998), 1079–84. https://doi.org/10.1109/83.701170

[7] Itseez. 2015. Open Source Computer Vision Library. https://github.com/itseez/opencv.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 3992–4003. https://doi.org/10.1109/ICCV51070.2023.00371

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (May 2017), 84–90. https://doi.org/10.1145/3065386

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. http://arxiv.org/abs/1612.03144 arXiv:1612.03144 [cs].

[11] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. 2020. Interactive Image Segmentation With First Click Attention. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 13336–13345. https://doi.org/10.1109/CVPR42600.2020.01335

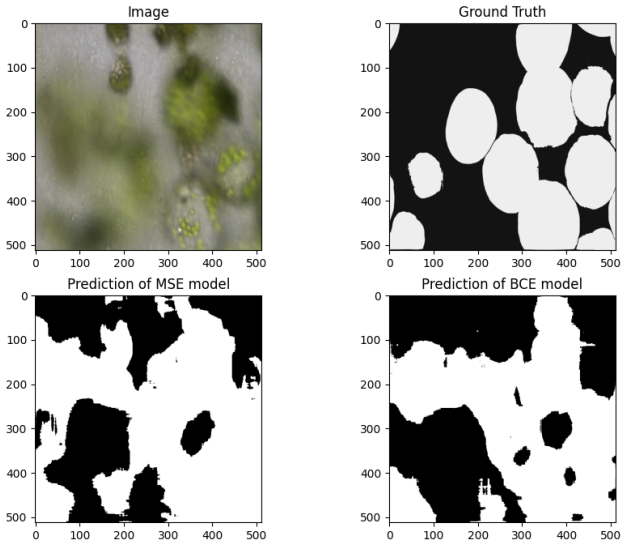


Fig. 13. Z-level 12

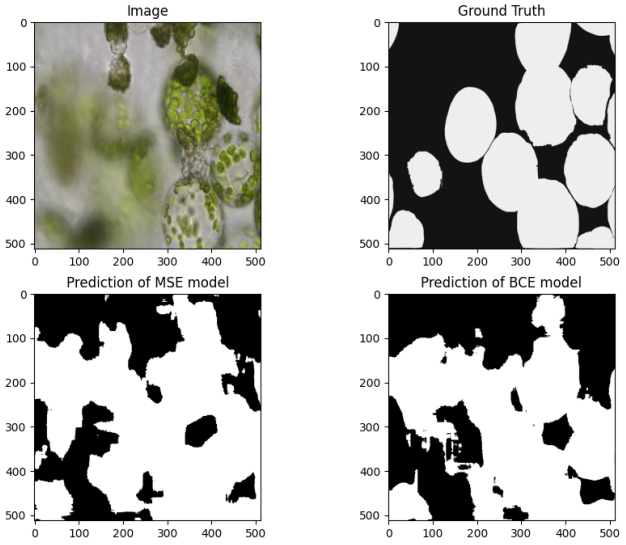


Fig. 14. Z-level 16

[12] Feixiao Long. 2020. Microscopy cell nuclei segmentation with enhanced U-Net. *BMC Bioinformatics* 21, 1 (Jan. 2020), 8. https://doi.org/10.1186/s12859-019-3332-1
[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. http://arxiv.org/abs/1411.4038 arXiv:1411.4038 [cs].
[14] Zuodong Niu and Handong Li. 2019. Research and analysis of threshold segmentation algorithms in image processing. *Journal of Physics: Conference Series* 1237 (June 2019), 022122. https://doi.org/10.1088/1742-6596/1237/2/022122
[15] Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. http://arxiv.org/abs/1511.08458 arXiv:1511.08458 [cs].
[16] Fabien Pierre, Mathieu Amendola, Clémence Bigeard, Timothé Ruel, and Pierre-Frédéric Villard. 2021. Segmentation with Active Contours. *Image Processing On Line* 11 (May 2021), 120–141. https://doi.org/10.5201/ipol.2021.298
[17] Luca Rettenberger, Friedrich Münke, Roman Bruch, and Markus Reischl. 2023. Mask R-CNN Outperforms U-Net in Instance Segmentation for Overlapping Cells. *Current Directions in Biomedical Engineering* 9 (Sept. 2023), 335–338. https://doi.org/10.1515/cdbme-2023-1084
[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. http://arxiv.org/abs/1505.04597 arXiv:1505.04597 [cs].
[19] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556 arXiv:1409.1556 [cs].
[20] Nicholas Sofroniew, Talley Lambert, Grzegorz Bokota, Juan Nunez-Iglesias, Peter Sobolewski, Andrew Sweet, Lorenzo Gaifas, Kira Evans, Alister Burt, Draga Doncila Pop, Kevin Yamauchi, Melissa Weber Mendonça, Genevieve Buckley, Wouter-Michiel Vierdag, Loic Royer, Ahmet Can Solak, Kyle I. S. Harrington, Jannis Ahlers, Daniel Althviz Moré, Oren Amsalem, Ashley Anderson, Andrew Annex, Peter Boone, Jordão Bragantini, Matthias Bussonnier, Clément Caporal, Jan Eglinger, Andreas Eisenbarth, Jeremy Freeman, Christoph Gohlke, Kabilar Gunalan, Hagai Har-Gil, Mark Harfouche, Volker Hilsenstein, Katherine Hutchings, Jessy Lauer, Gregor Lichtner, Ziyang Liu, Lucy Liu, Alan Lowe, Luca Marconato, Sean Martin, Abigail McGovern, Lukasz Migas, Nadalyn Miller, Hector Muñoz, Jan-Hendrik Müller, Christopher Nauroth-Kreß, David Palecek, Constantin Pape, Eric Perlman, Kim Pevey, Gonzalo Peña-Castellanos, Andrea Pierré, David Pinto, Jaime Rodríguez-Guerra, David Ross, Craig T. Russell, James Ryan, Gabriel Selzer, MB Smith, Paul Smith, Konstantin Sofiiuk, Johannes Soltwedel, David Stansby, Jules Vanaret, Pam Wadhwa, Martin Weigert, Jonas Windhager, and Philip Winston. 2024. napari: a multi-dimensional image viewer for Python. https://doi.org/10.5281/zenodo.12556492
[21] Tingxi Wen, Binbin Tong, Yu Liu, Ting Pan, Yu Du, Yuping Chen, and Shanshan Zhang. 2022. Review of research on the instance segmentation of cell images. *Computer Methods and Programs in Biomedicine* 227 (Dec. 2022), 107211. https://doi.org/10.1016/j.cmpb.2022.107211

A APPENDIX

A.1 Disclosures

During the preparation of this work, the author used Writefull and Gemini AI in order to correct the grammar of the text, format the text in LaTEX and find abstracts and information about specific academic papers as well as bug fixing for the code. After using this tool/service, the author reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

A.2 Prediction across z-levels

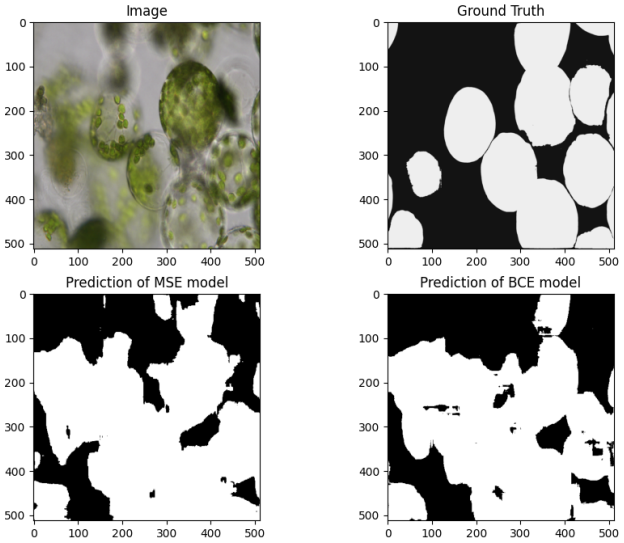


Fig. 15. Z-level 20