Universiteit Leiden

Master Computer Science

Segmenting and Analysing 3D images of FUCCI-reporter induced breast cancer nuclei spheroids with deep learning models

Name:	Leonard Tiling
Student ID:	s3660699
Date:	20/08/2024
Specialisation:	Bioinformatic
1st supervisor:	Dr. Lu Cao
2nd supervisor:	Dr. S.E. Le Dévédec

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract Effective and accurate analysis of 3D microscopy images is crucial for advancing cancer drug research. This study aims to establish a foundation for anti-cancer drug screening by developing a segmentation pipeline for cell cycle state analysis of 3D spheroid nuclei cluster images. Leveraging open-source deep learning tools, we introduce novel processing strategies to facilitate cell cycle state analysis based on FUCCI reporter sensors. We developed, refined, and compared U-Net, Cellpose and StarDist models for 3D nuclei segmentation in order to establish the fundamental framework for subsequent cell cycle state analysis. To address the biomedical imaging challenges of low image quality, limited 3D data and highly demanding 3D annotation, we present a pipeline and segmentation model that optimize downstream analysis. Experimental validation confirmed that our approach achieved a segmentation performance of 77% with minimal manual annotation and wielding inferior 3D image quality as well as efficient open-source models. These results form the basis for accurate 3D segmentation and facilitate cancer drug discovery by identifying critical factors for scientific work with FUCCI nucleus segmentation and analysis.

Contents

1 Introduction

Breast cancer presents a lethal threat to women worldwide, stated as the leading cause of death by cancer on a global scale [1]. The aggressive and rapid proliferation is a characteristic behavior of tumor cells and is therefore targeted in treatment research. With the help of modern microscopes, 3D imaging of cancer spheroids is becoming more and more popular. Detailed 3D structures preserve more information and give valuable insight to composition and structure, superior to 2D imaging [2, 3]. Hence, researchers target the distinct tumor proliferation behavior in drug research, mastering even cancer cells that are resistant against chemotherapy [4]. To observe the cell cycle state of quiescent like breast cancer cells, the Fluorescence Ubiquitination Cell-Cycle Indicator (FUCCI) imaging system is utilized [5]. The FUCCI reporter exploits gene expression to signal the exact cell cycle state of each nucleus, allowing to observe the phases of nuclei in a spheroid in real time.

The overall goal of this research is to utilize 3D microscopy imaging to monitor cell cycle behavior to indicate the effectiveness of drugs on the breast cancer cells.

Accurate and reliable nuclei segmentation is crucial for downstream cell cycle analysis. While nuclei segmentation is well-established in 2D, it becomes considerably more challenging in 3D due to factors such as low signal-to-noise ratio, poor contrast, and dense nuclear clustering. Traditional methods, such as watershed algorithm and thresholding, often fail in accuracy and reliability in 3D image segmentation. As a result, researchers have encountered a bottleneck in downstream analysis rather than in image acquisition [6]. Recent advancements in deep learning (DL) have shown great success in addressing these challenges. DL models are known as favorable tool for processing high-throughput microscopy data and are widely used for analysing large cell image data. Convolutional neural networks (CNNs) have excelled at extracting and recognizing patterns in images, making them a popular choice for biomedical image analysis. Applying CNN models accelerates an accurate cancer drug screening workflow. The biomedical community employed the use of DL for 3D segmentation successfully in histopathology, cancer research, and tissue analysis [7, 8].

In this study we explore a novel approach to cancer drug screening by utilizing 3D DL models - Cellpose, StarDist, and U-Net – to accurately segment noisy and challenging 3D nuclei images with the least possible manual correction.

1.1 Biological Background

Fighting and understanding cancer is one of the major goals in medicine research. This study investigates the potential of 3D spheroid breast cancer models as a more physiologically relevant platform for drug discovery and safety assessment. By identifying and tracking the proliferation dynamics of malignant tumor cells, this research supports the ongoing efforts to identify and characterize potential anticancer agents. Using novel spheroid cultures is shown to be beneficial towards 2D cultures, based on almost in vivo characteristics of the cultivated cells [3]. Segmentation methods are crucial to identify, differentiate and quantifying the FUCCI emitted fluorescent signals for downstream cell cycle analysis. As component of high content screening, segmentation in combination with the analysis is employed to asses cancer drug efficacy. Cell cycle analysis reveals which drugs lead to cell cycle arrest or cell death. The accurate and precise segmentation especially in densely clustered nuclei is crucial and demonstrates the bottleneck of this research pipeline [9]. Utilizing FUCCI reporter promotes the process of identifying concrete cell proliferation states in sample cells while improving conventional cell-cycle markers [5]. Using fluorescent confocal microscopy imaging technology, a life and in vivo observation of proliferation activity of our cancer cell samples at the same time is achievable and will be used thorough this work [10].

The adoption of 3D microscopy images has gained prominence in segmentation tasks, as they provide valuable spatial information that aids in the differentiation of clustered cells. Furthermore, there is a high availability for certain image types that are already implemented in medical pathology, such as brain MRI or CT images of various organs [9]. Nevertheless, 3D image availability and variation is still a represents a central challenge, due to most diverse microscopy images and imaging techniques. Greenwald et. al. solved the shortage of images by constructing a diverse and distinct tissue dataset for 2D images [11] . To the best of our knowledge, comparable approaches are lacking in the 3D case.

1.1.1 Microscopy Imaging Parameters

To evaluate the efficacy of novel cancer drugs, 3D microscopy images are acquired. These images provide spatial information essential for identifying and tracking individual cells within spheroids. The quality of these images depends on various factors, including sample preparation, staining, and microscopy settings.

High-resolution 3D images are desirable but can be computationally demanding. For that reason image quality with data efficiency is crucial. Pinhole size, objective lens, and pixel spacing influence image resolution. Smaller pinholes and higher numerical aperture (NA) lenses improve resolution and contrast. Microscopes with oil immersion offer higher numerical aperture (NA), leading to better resolution and light-gathering ability. Air objectives provide lower NA, resulting in lower resolution. Previous studies have explored optimal imaging conditions and revealed the trade off between high image resolution and vast data generation for 3D cell analysis. Le et al. and Wagner et al. initially employed large images $(2000, 1000, 500)$ pixels with a fine z-step size $(1.0\mu m)$ in their CellSeg segmentation model. To improve computational efficiency, subsequent versions of the model utilized smaller images $(512, 512, 30)$ pixels [12, 13]. Weigert et al. demonstrated that a 63x oil immersion objective and sampling at $0.116\mu m$ and $0.122\mu m$ per pixel can produce high-quality images for training DL segmentation models [14]. This configuration yielded exceptionally high-resolution images, enabling detailed analysis of their model organisms. With 28 and 6 images respectively, they acquired sufficient data to train a deep learning segmentation model. The image dimensions were $(1157, 140, 140)$ and $(512, 512, 34)$ pixels. Notably, previous state-of-the-art 3D segmentation models employed smaller pinhole sizes of 89 µm $(0.6 - 1.0$ airy units) and coarser pixel spacing of $0.2\mu m$ or $0.126\mu m$, $0.126\mu m$ for the XY axes and $0.122 - 1.0\mu m$ for the Z axis [15, 12, 14, 16].

1.1.2 FUCCI Reporter

Observing the dynamic behavior of living cells provides insights into growth processes. Cell proliferation is a primary indicator of tumor growth [17]. The lifespan of a cell is divided in four states, *G1*, *S*, *G2* and *M*. *G1, S* and *G2* represent the interphase of a cell, M describes the mitotic cell division phase. During the *G1* phase, the cell reproduces its organelles and mainly gains physiological stability and largeness. Subsequent, in the synthesizing *S* phase the cell produces a comprehensive copy of its DNA, stored in the cell nucleus. Lastly, the cell increasingly grows in size and number of organelles to fully prepare for mitosis in the *M* phase [18]. The complex procedure of mitosis can take between 9h for fast proliferating cells and up to 24h in human cells. The stages of Mitosis are prophase, metaphase, anaphase and telophase and describe how the DNA condensates, reorganizes, separates and finally divides in two filial cells [18]. Exactly this intricate process is where the FUCCI reporter makes use of the countless proteins that assists and guide throughout the phases and stages.

Developed in 2008, Sakaue-Sawano et al. were able to genetically design and implement fluorescent protein that label the cell cycle phases [17]. The authors identified made use two protein named Geminin and Cdt1, that oscillate in expression level based on the cell cycle phase, that the cell is currently in. In this study, Green Fluorescent protein (GFP), Cyanine3 (Cy3) and Hoechst as nuclear counterstain were employed.

Genetically modifying the unique indicator proteins made it possible that every nuclei of each cultivated cell emits either a red signal (Cy3) for G1 phase and a green signal (GFP) for *S, G2* and *M* phase. The transmission from *G1* to *S* will produce an yellow or orange like fluorescent signal, provoked by the overlaying expression of the red and green protein labels [17]. Hence, accurate representation of the spatiotemporal cell cycle patterns, as the effect of drugs on the nucleus and its mitosis activity, can be discovered. Providing this powerful method to visualize and analyse the cell cycle behavior of any target cells, secured the process for various subsequent research projects that applied and improved the FUCCI method with different fluorescent protein markers and target cells in cancer and stem cells [19, 10].

The concept of observing cancer cells in real time was discussed from Shuya Yano et al. [4]. His work documented the success of identifying the cell cycle state of chemo-resistant cancer cells, that were then monitored while induced with drug that arrest the cells in a phase, that was most vulnerable to existing chemotherapeutic treatments. Likewise, this work will utilize the FUCCI system to monitor the proliferation behavior of breast cancer cells to identify drugs that arrest the cell cycle state when induced. The green fluorescent protein (GFP), the cyanine3 dye (Cy3) as a red G1 signal and the Hoechst stain were selected as stains. The Hoechst staining binds to DNA molecules and is excited by UV-light, emitting a blue fluorescent signal and is therefore used to identify the nuclei of a cell [20]. Hoechst is excited by a wavelength of $409,8$, GPF by $489,0$ and Cy3 by $562,6$ nanometer.

1.2 Technological Background

The rise of DL led to the application of CNN's in computer vision tasks, enabling fast processing of high-resolution, high-throughput microscopy images [21]. Nuclei and cell segmentation remain fundamental to biomedical research, supporting diagnosis, treatment, and disease understanding [13]. Developing DL methods for 3D image segmentation is an active research area with potential for significant biomedical impact [22].

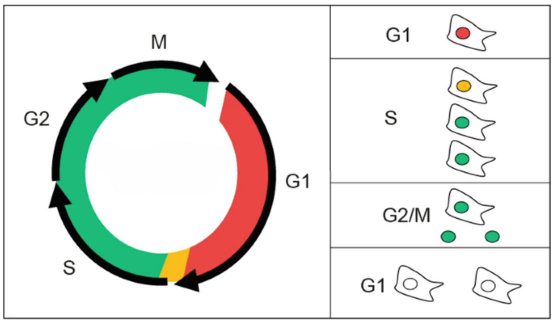

Figure 1: **Visualization of the FUCCI system.** The genetically modified fluorescent proteins are transcribed during cell cycle. The red signal Cy3 is emitted during G1 phase. During the S, G2 and M phase, only green, fluorescent signals from GFP are emitted. In the early S phase, overlapping red and green signals are associated with a yellow color. Lastly, the change from M to G1 is indicated by an absence of both markers and is covered by a nuclei stain such as Hoechst. Image taken from[10]

Traditional 3D segmentation is known to be labor intensive and requires extensive human expertise. Trained CNN's offer a crucial advantage by automatically recognizing and extracting relevant features for computer vision tasks without expert knowledge [22]. These networks are extensively trained on large amount of annotated data, using supervised learning. Once trained, the network can generalize and apply the learned "knowledge" to new, unseen data, making the effort of further annotations and human expert knowledge obsolete. Recent advancements in DL have been applied to the medical and biomedical fields, including prediction of stem cell state prediction and osteogenic differentiation feature measurement [23, 24].

1.2.1 Deep Learning Segmentation Models

The work of Caicedo et. al. emphasizes the design of this study to implement DL into the field of analyzing microscopy images. The authors work with nuclei images and compare DL approaches to classical approaches, demonstrating that DL can improve accuracy in nuclei segmentation and aids minimizing the segmentation errors [25]. Biomedical image data varies widely, including CT, sonar, tissue microscope, and fluorescent images. Generalization becomes challenging due to the unique characteristics, quality, and interpretation requirements of each image type. Despite these challenges, DL has shown superior efficiency and accuracy in complex segmentation and classification tasks compared to manual methods like image thresholding, especially in terms of generalization [26, 7]. The superiority of DL becomes particularly evident when working with large datasets, such as those generated by 3D microscopy [7]. This makes DL a promising approach for addressing the challenges associated with analyzing diverse biomedical image data.

Several models were developed in recent years, often based and inspired by the influential U-Net architecture, designed for fine-grained image segmentation in 2015 [27]. Based on the field of application, researchers developed models specifically trained on certain data, such as the PlantSeg model is only designed for plant tissue segmentation [28]. While U-Net employs pixel-wise class prediction based on loss functions, other methods utilize more conceptual image information. StarDist predicts distances to mimic polygon shaped cell representation and was one of the first model to generalize from different cell types while maintaining segmentation accuracy [29]. In 2021 Cellpose was developed with the aim of generalization across various tissues, shapes and cell sizes, employing multiple models for gradient based predictions [30]. However, most models were trained on 2D data, limiting the exploitation of spatial information. The foundation for 3D models was laid in 2016 by implementing 3D convolutional layers and therefore extending the input shape of the U-Net model to 3D [31]. More complex models such as StarDist and Cellpose addressed the increased complexity of 3D space by simplifying predictions and stitching multiple axial 2D predictions to an 3D image [14, 30] .

1.2.2 Scarceness of Training Images

The development of robust 3D segmentation models in biomedical image analysis has been hindered by the lack of large, annotated 3D datasets and the computational complexity associated with 3D processing. Although 3D imaging is becoming more accessible, 3D images exhibit diverse structures, resolutions, qualities and types. Most important, annotating 3D data is considerably infeasible and more challenging in time and expertise [32]. Hence, annotating and manual labeling of 3D image data is one of the biggest challenges in the field of nuclei segmentation. Researchers have proposed various strategies to address these limitations, including pipelines and toolkits for streamlining the workflow, leveraging pre-trained 2D models for few-shot training on 3D data, and combining automated training with manual refinement in a human-in-the-loop approach [15, 14, 32]. The challenge of 3D annotation is demonstrated by Thiyagarajan et al. who try to reduce the human time to produce annotations significantly by training a DL model to extend sparsely annotated 2D data into dense 3D annotations [32].

1.2.3 Class Imbalance and Validation metrics

Medical images introduce a lot of their own challenges, such as bad noisy to image relation, underexposure or densely clustered cells. A significant challenge in medical pixel-wise segmentation lies in the class imbalance of rare observations, such as tumors in pathology, specific organs in whole CT scans, or cell detection[33, 34]. As a result, the imbalance poses challenges for traditional metrics like accuracy, which will provide misleading indicators of segmentation quality of segmentation algorithms. For instance, in nuclei microscopy datasets the predominance of pixel are classified as background class. Hence, in majority and easy to classify the overall image accuracy will invariably appear high. The same issue applies to precision and recall when averaged over all classes and reported as a single value. To address this limitation, in this study segmentation quality assessment is consistently reported using class-specific accuracy, recall, and precision, alongside visual inspection examples.

1.2.4 Loss Functions

To effectively train DL models, carefully selected loss functions are essential for guiding the learning process and minimizing the discrepancy between predicted and ground-truth segmentations. While differentiable loss functions are essential for training convolutional neural networks, class imbalance and segmentation metrics like accuracy and precision are not suitable due to their non-differentiability. The assessment of segmentation masks typically involves pixel-by-pixel comparison with ground truth masks, resulting in binary (True/False) evaluations. However, the use of class probabilities during training and binary masks during validation introduces different metrics for each phase.

Addressing class imbalance in multiclass 3D images is a significant challenge in this study. The small fraction of target classes can lead to instability in training pixelwise and semantic segmentation systems [34, 33]. Various strategies have been proposed to mitigate this issue, including adapting the training set with sufficient examples for each class and reducing variance through pixel standardization and normalization [35]. Given the limitations of generating and annotating large-scale 3D microscopy datasets, finding and adapting a suitable loss function that addresses class imbalance is a more sophisticated approach, as demonstrated in previous medical image segmentation studies [27, 36, 33, 37].

While metrics like Intersection over Union (IoU) and Dice Score can help mitigate over and under-classification, reporting them for the entire image may introduce bias towards the overrepresented class. Similarly, loss functions that incorporate the discrepancy between predictions and ground truth on a pixel-by-pixel basis may be biased as well, hindering the learning process for underrepresented classes. To address this, specialized loss functions are necessary, as suggested in previous research [37, 38, 39, 36]. Several authors have proposed adapted versions of existing loss functions or metrics to compensate for class imbalance in various ways [33, 40, 41, 42].

1.2.5 Human in the Loop

Human-in-the-loop (HITL) strategies offer a potential solution to the challenges posed by the tedious nature of 3D image annotation, especially in the field of deep learning-based nuclei segmentation [32]. The strategy was introduced back in 2018, initially only for machine learning and has since gained popularity in various fields. The survey by Xingjiao Wu et al. shows how the new nature of interaction between DL models and humans is being leveraged [43, 44]. It shows that embedding human knowledge into machine learning processes can significantly improve the performance and efficiency of models [44]. The key idea is the interplay of iterative human correction of DL model results, which leads to progressively improved results when used again to train the models. In turn, subsequent results require fewer corrections and speed up the learning process while minimizing human effort. This work provides an interface for the HITL strategy to be used in future work to optimize model performance while reducing reliance on laborious manual annotation.

1.3 Research Objectives

Accurate 3D nuclear segmentation remains challenging due to data limitations and complexities. Acquisition and annotation of 3D image data is a laborious process prone to inter- and intra-observer variability [32]. Furthermore, the pronounced class imbalance found in microscopy nuclei images intensifies the difficulty of training models on limited annotated data. The crucial success of training a DL model relies on a suitable ground truth dataset, containing a representative amount of annotated training examples for every class, which is often unavailable. The absence poses a particular challenge, which this work attempts to overcome. Image quality also impacts model performance, with noisy and low-resolution nuclei, as found in spheroids, posing additional challenges. Computational resources are another critical factor, as 3D image processing is computationally demanding and requires high-performance computing (HPC) infrastructure like ALICE [45]. Despite these challenges, models trained directly on 3D volumes have the potential for superior performance, especially when dealing with complex 3D structures.

This study addresses these limitations and challenges by developing a reliable 3D nuclear segmentation model tailored to specific conditions of FUCCI-labeled nuclei spheroids. The architecture is designed to be lightweight and class balances are taken into account by testing and evaluating dedicated loss functions. To reduce the consequences of poor image quality, we employ carefully selected image pre- and post-processing techniques that have been proven to significantly improve performance [16]. We propose to optimize the U-Net architecture and parameter configurations while minimizing manual annotation effort and utilizing minimal amount of low-quality training data. To evaluate the effectiveness of training from scratch versus utilizing a pre-trained model, we compare the U-Net implementation to a pre-trained state of the art model (StarDist). This comparative analysis provides insights into the trade-offs between model complexity and data specificity.

For downstream applications and cell cycle analysis, using FUCCI labeling provides valuable insights into drug efficacy. Laying the foundations to be able to track individual cells over time can enhance the understanding of cellular behavior and response to treatments. Incorporating temporal information into the segmentation process, through methods such as recurrent networks or transformers, holds the potential to significantly improve the accuracy of these downstream applications.

On the whole, this work aims to facilitate downstream applications and future work in drug discovery and development.

1.4 Structure of the Thesis

The presented study is structured as follows. Section 1 provides a brief overview of the research goal, highlighting existing research gaps and the motivation behind the study. Key concepts and terminology are introduced, placing the research within the broader research context. Subsequently, Section 2 outlines the methodological foundation, introducing the main models employed, data and image acquisition, image processing techniques, as well as explanation of metrics and validation techniques. A detailed description of the manual 3D image annotation process follows. The experimental results are presented in Section 3, supported by visual representations. Building upon these findings, Section 4 offers a comprehensive discussion that places the results within the theoretical framework and states study limitations. Potential steps for future research are explored, linked to the limitations and key findings. Finally, the paper concludes by summarizing key contributions and stating the data accessibility.

2 Methods

2.1 Workflow

As expected, problems emerged during the work on this project. Several strategies where tried out, changed and adapted. Figure 2 illustrates the proposed workflow, outlining the sequential steps and methodologies employed in this study. The diagram also incorporates a HITL intersection for future work to enhance the outcomes of the study. The workflow is divided into three sub-pipelines: **A, B** and **C**, corresponding to distinct work areas. **A** represents image acquisition and preprocessing, **B** integrates manual labeling steps and **C** depicts model training and its interaction with other components. To align the workflow with the thesis structure,

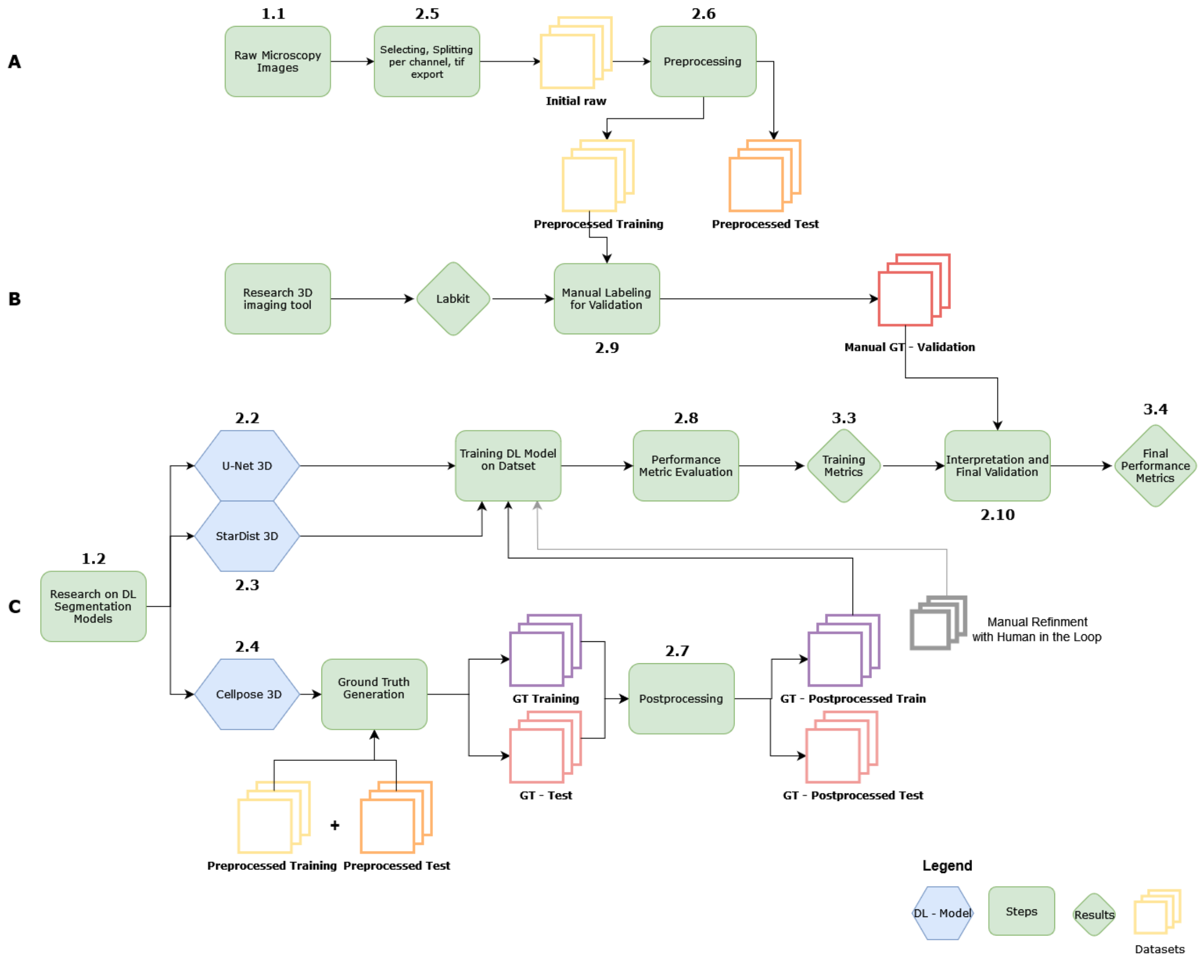

Figure 2: **Workflow overview.** This figure illustrates the pipeline of the thesis. **A** represents image acquisition and preprocessing. **B** integrates manual labeling steps. **C** depicts model training and its interaction with other components. The **numbers** in bold above the steps correspond to respective chapter sections of this work. The legend is located in the lower right corner. The grey dataset placeholder indicates the potential intersection human in the loop procedure, which is not implemented in this work.

chapter numbers above steps and result signs indicate where methods are described and results are reported.

The diagram differentiates between DL models (blue diamond), Datasets (stack of unique colored images) and pipeline steps and processing that was conducted in this research (green rectangles). These steps encompass literature review, software and model testing, microscopy image acquisition coordination, and DL model development, deployment, testing, and evaluation. Additionally, the steps include software development, DL model training, manual image review, Jupyter notebook deployment for pre- and post-processing, and other tasks indicated in step titles. The image stacks are color-coded to represent their status and attributes.

2.2 U-Net Model

Ronneberger et al. introduced a novel deep convolutional neural network (CNN) architecture, known as the U-Net, which revolutionized biomedical image segmentation tasks and paved the way for subsequent models [27]. The authors main objective was to address a significant challenge in biomedical imaging which is the shortage of training images and ground truth datasets. Until today, researchers endeavor with the creation of correctly labeled ground truth images due to the laborious and time-consuming manual labeling process, requiring expert domain knowledge [6, 32, 15].

U-Nets architecture is built on a convolutional encoder - decoder principle, compressing the input image while simultaneously doubling the number of features, followed by an expanding phase deploying upconvolution and dividing the feature layers to restore the original input resolution [27]. In detail, the encoder architecture was implemented by employing double convolution blocks utilizing a 3x3 convolution kernel with Rectified Linear Unit (ReLU) [46] activation functions, along with a stride 2 max-pooling layer to reduce image dimensions while doubling the feature dimension. Subsequently, the decoder part was implemented by an upconvolution block with a 2x2 kernel to increase image size while halving the feature channel dimension followed by the same double convolution block utilized in the encoder part. Finally, a 1x1 convolutional layer reduces the feature dimension to the desired output classes of the output segmentation map [27]. The networks architecture is illustrated in Figure 3.

Double convolution blocks with a 3x3 convolution kernel, ReLu, a stride 2 maxpooling layer to decrease the image size while double the feature dimension and finally the upconvolution block with a 2x2 kernel increasing the image size while halving the feature channel dimension in each step, followed by the same double convolution block as in the encoder part [27].

Thus, the ”U” shaped architecture, visible in Figure 3, combines two critical components: learning location information from the encoder section and acquiring contextual information from the decoder section. To impede information loss during downsampling, the authors introduced, which concatenate learned features from the encoder level to corresponding features in the decoder level. This enables U-Net to preserve and enhance spatial information, facilitating the segmentation of finer details compared to fully connected networks [27].

Overall, the achievement of U-Net lies in its ability to achieve high pixel-based classification accuracy using only a small number of annotated training images, reducing training time and computational resources in terms of memory and network complexity [27].

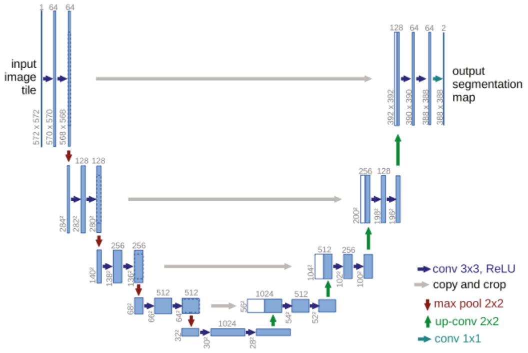

Figure 3: **U-Net architecture.**This image, taken from the original U-Net paper illustrates the typical ”U”-shaped network architecture [27]. Skip connection as grey horizontal line transfer feature information from encoder to decoder part of the network. Building blocks consist of double convolution blocks.

For this work, a custom adaptation of U-Net was implemented with pytorch and tested on a laptop with NVIDIA GPU (2GB) with initially smaller input image size. The implementation was designed to work and handle the specific image type of our dataset. More importantly, it balanced the tradeoff between a lightweight model that required as little GPU memory as possible with the best possible segmentation and feature detection capability. The development of the program also considered the possibility of application by non-data science users. In the further stages of the project, Leiden University kindly provided access to the Academic Leiden Interdisciplinary Cluster Environment (ALICE), which was used for training with larger images and correspondingly high memory requirements of up to 24 GB [45].The architecture of the final model can be seen in Figure X. The online platform wandb was used for the visualization and tracking of ML experiments [47]. Implemented customized functions in combination with wandb facilitated real-time observation of the learning curve, evaluation metrics and actual segmentation predictions online, independent of the server or laptop infrastructure. The advantage of this stand-alone implementation was that we had full control over evaluation methods, intermediate results, parameter settings and optimization strategies compared to using existing packages and tools.

2.3 StarDist Model

In 2018, Schmidt et al. addressed the challenge of segmenting densely packed cell instances with overlapping boundaries in noisy microscopy images. They proposed a method that combines pixel-wise segmentation with instance localization using bounding boxes [29]. Utilizing deep learning architectures to learn complex relationships between image features and cell properties allows the model to adapt to different cell types in challenging conditions. This approach contrasts with the traditional bottom-up approach of semantic segmentation followed by pixel merging, which struggles in such scenarios as shown by Caicedo et al. [25]. The latter approach often requires manual intervention or assigning high weights to boundary classes [25]. Conversely, top-down approaches, such as Mask R-CNN, have shown success in localizing instances using pre-defined bounding boxes [e.g., Mask R-CNN source].

However, the axis-aligned nature of bounding boxes limits their ability to accurately capture complex cell shapes [29]. To address this limitation, Schmidt et al. introduced the StarDist model, which utilizes a more accurate cell shape representation based on star-convex polygons [29]. By assuming a generally circular morphology for the nuclei in the microscopy images, StarDist, in conjunction with non-maximum suppression, achieves superior performance on challenging clustered microscopy datasets compared to conventional methods [29].

The model architecture is based on a lightweight U-Net architecture variation, elevating the low computational resources while maintaining competitive segmentation accuracy. The core functionality of StarDist is based on the prediction of two crucial pieces of information for each pixel, **Object Probability** and **Star-convex Polygon Distances**. The Object probability reflects the likelihood of a pixel belonging to a cell, calculated as the normalized Euclidean distance to the nearest background pixel and provides more information than a binary classification into background and cell. Also, prioritizing polygons associated with pixels closer to the cell center, leads to more accurate segmentations. Second, the model predicts distances to the object boundary along a set of predefined radial directions. The amount of radial directions, describing an nucleus instance is crucial for the shape resolution and can be set as a hyperparameter [29].

Extending U-Net and Cellpose for 3D segmentation presented a greater challenge compared to 2D applications. Weigert et al. [14] successfully addressed this in the year 2020 by focusing on three key factors: 1) developing an efficient way to represent 3D polyhedral shapes, 2) optimizing the implementation for GPUs without significantly increasing computational demands, and 3) incorporating the anisotropic properties of 3D microscopy data, which can pose a significant obstacle for image analysis. Lastly, the authors evaluated their method on two demanding datasets revealing its significant advantages over both traditional and deep learning-based techniques [14].

2.4 Cellpose Model

Stringer et al. [30] addressed the challenge of limited training data in biomedical image segmentation by introducing Cellpose, a novel general-purpose approach. This method prioritizes working across a wide range of microscopy image types while minimizing model training requirements and facilitating user execution without the need for extensive parameter tuning [30]. Cellpose stands out in delivering accurate segmentation results without extensive user customization or retraining, making it a broadly applicable tool. Trained on a highly diverse dataset exceeding $70,000$ segmented objects, the model demonstrates proficiency in handling various user-provided data while maintaining high segmentation accuracy. Initially developed as a 2D prediction model for image analysis, Cellpose has been extended to a 3D variant that leverages its core 2D network for 3D prediction. This is achieved by strategically slicing the 3D volume (X, Y, Z) into a series of 2D sections (XY, XZ, YZ). The trained 2D prediction network is then applied to each section, and the resulting pixel-wise predictions are ultimately combined to generate a 3D segmentation mask. Notably, Cellpose achieves accurate segmentation in 3D applications by employing pre-trained 2D weights, without requiring any dedicated 3D training [30].

The core innovation of Cellpose lies in its unique architectural design. In contrast to conventional methods that directly predict segmentation masks (e.g., U-Net), Cellpose employs a two-stage process. The first stage involves predicting vector flow fields. These flow fields represent the horizontal and vertical intensity gradients within the image, essentially providing information about potential cell locations and their corresponding boundaries. This is achieved through a binary segmentation mask. Subsequently, the model leverages these flow fields in the second

Table 1: **Comparison of model architecture and features.** StarDist and Cellpose extend the U-Net architecture with specialized strategies and refined architectures, including residual layers and distinct output channels (object-, distance and gradient-prediction). These modifications enhance segmentation accuracy but increase model complexity and computational demands compared to the original U-Net. [14, 30].

Feature	**U-Net3D**	**StarDist 3D**	**Cellpose nuclei**
Backbone	U-shaped CNN	ResNet based CNN	U-Net with residual building blocks
Prediction method	pixelwise class prediction	object probability and centroid distances based polyhedrion shape prediction	gradient based shape prediction with binary refinement
trainable params	22.396.739		
Encoder	symmetrical 3D convolution + skip connections	3D convolution + ResNet blocks	3D convolution
Decoder	symmetrical 3D trilinear deconvolution + skip connections	ResNet and non maximum supression	upsampling and feature summation
Output	Segmentation mask of background, nuclei and edge	object probability, StarConvex radial distances	horizontal+vertical gradients as shape, and binary pixelwise prediction (refinement)

stage to generate the final, refined segmentation masks. This two-step approach contributes to the robustness and adaptability of Cellpose, making it a valuable tool for a wide range of cellular image analysis tasks [30].

Unlike U-Net's direct mask prediction, Cellpose utilizes a two-step approach. The first stage involves a convolutional neural network (CNN) specifically designed to predict ”flow” fields within the image. The architecture of this prediction network follows an encoder-decoder principle with residual convolutional building blocks, as described by Stringer et al. [30]. To further enhance performance, Cellpose incorporates several architectural modifications. These modifications include using average pooling only in the smallest convolutional layer and employing feature summation instead of traditional skip connections [30]. The resulting flow fields essentially encode the direction and magnitude of intensity gradients, guiding the model towards accurate object boundary detection.

Furthermore, Cellpose incorporates a user-friendly Graphical User Interface (GUI) that empowers researchers to select models, adjust hyperparameters, and crucially, perform manual correction and refinement of predicted segmentation masks.

2.5 Dataset and Image Acquisition

The dataset used in this work contains 3D images of breast cancer cell spheroids. The spheroids were cultured and imaged in the laboratories of the Leiden Academic Centre for Drug Research (LACDR). The image files have a shape of $(3, 512, 512, 21)$ with (C, X, Y, Z), containing 3 reporter channels (C), X/Y pixels and 21 slices (z), saved in Tagged Image File format (TIF). For downstream preprocessing and as model input, we extracted the three reporter channel

Table 2: Settings of the confocal fluorescence microscope. The mentioned settings were applied for image acquisition. A low pinhole size, a minimum Z-step size and a high magnification are important for a high resolution.

Setting	Value
Magnification	20x
Numerical Aperture	0.75
Refractive Index	1
Pinhole Size	60 μm
Timepoints per Specimen	11
z-slices	21
z-step size	5 μm
x/z pixel size	1.243 μm

individually and converted the 16-bit to 8-bit grayscale images. In that way, the pixel intensity range was reduced from $0-65535$ into the 8-bit grayscale range of $0-255$ using the NIS-Elements Viewer Imaging software [48].

The spheroids are induced with the FUCCI reporter system that indicates the cell cycle state of each cells nucleus, making it possible to report the proliferation activity of that cancer specimen type. With that, the effectiveness of induced experimental cancer treatment can be observed, reported and analyzed in downstream analysis application. Ten different specimen were imaged over seven continuous days with two pictures per well, every 30 minutes. The images were acquired with a NikonC2plus confocal fluorescent microscope to enhance and capture the different emitted fluorescent reporter signals in a sequential point by point manner. Combining fluorescent microscopy and confocal acquisition has the benefit of increased effective resolution, improved signal to noise ration and reduced blurring in the image due to background illumination. Especially in thick specimen, such as spheroids, the consecutive sharp focus point with adjusted laser power should provide the best possible quality for every slice of the sample. Table 2 provides the acquisition details.

2.6 Preprocessing of 3D image data

Microscopy images are known to have low signal to noise ratios and are of low quality due to challenging microscopy acquisition. To improve image quality for downstream segmentation tasks using ML and deep learning DL algorithms, several image preprocessing steps were applied. These steps targeted contrast enhancement, noise reduction, detail preservation and enhancement, as well as oversampling correction. The effectiveness of image preprocessing in computer vision, particularly for segmentation tasks, is well documented in the literature. [49].

Handling the 3D spheroid images, two primary challenges were addressed. Improving poor signal to noise ratio and reducing the overall present background noise. The high background noise hid true nuclei signals, while the fluorescent laser settings caused overly intense (glowing) nuclei, leading to merged clusters and a loss of detailed 3D nuclear shapes. Furthermore, small artefacts in the cytoplasm that caught fluorescence required removal to minimize misinterpretation of the true signal.

The raw microscope data (.nd2 format) was imported into NIS-Viewer software for channel separation i. e. Cy3, GFP and Hoechst [48]. Then, each image was converted to 8-bit grayscale (intensity range: 0-255) as described in the dataset and image acquisition section (2.5). A thresholding step was applied as a preprocessing measure to eliminate background noise. Pixel intensities equal to or greater than 7 were set to zero. The NIS-Viewer software's Look Up Table (LUT) function allowed for channel-specific adjustment of the 32-bit intensity range, enhancing threshold selection through immediate visual feedback and precise background noise reduction. This capability is considered valuable for future analyses with new datasets.

Gamma correction is then applied to reduce noise and enhance details at the same time. This technique adjusts the image's luminance to selectively emphasize darker regions for improved signal details or suppress overexposed artefacts. The chosen gamma value (γ) controls this non-linear transformation, with $\gamma > 1$ darkening bright areas and $\gamma < 1$ brightening dark areas. Importantly, gamma correction does not skew the overall intensity distribution of the histograms image [50].

Subsequently, **Contrast Limited Adaptive Histogram Equalization** (CLAHE) was employed to address challenging lighting variations within the images. CLAHE is an image processing tool that enhances contrast by dividing the image into local regions and then adjusting their contrast histograms individually [51]. Unlike standard Adaptive Histogram Equalization (AHE), CLAHE incorporates a *clip limit value* to prevent excessive contrast amplification. This localized approach, achieved by dividing the image into a grid and applying CLAHE to each grid section, allows for contrast adjustments based on local nuclei characteristics, independent of uneven background illumination [52].

Even though CLAHE tried to prevent the noise amplification, it usually did not succeed entirely, and background noise was amplified. For this reason, the next step was to apply a medium soft filter to compensate for the residual noise. Despite prior processing, some minor overexposure artefacts remained, visible as small, bright dots in the image. To eliminate these smaller than a certain threshold, a **Tophat white filter** was employed. This filter effectively identifies small, isolated bright regions, allowing them to be subtracted from the original image, resulting in a cleaned final image [53].

The parameter of the preprocessing pipeline *threshold, γ, CLAHE-grid size, CLAHE clip value* and *median blur size* were determined through a series of experiments and carefully chosen to optimize image quality across all three channels. These parameters can be further fine-tuned based on specific input image characteristics in future work. The impact of applying individual pre-processing steps and the entire pipeline is presented in the results section. The effect of applying individual and full preprocessing is shown in the results section.

2.7 Postprocessing of 3D ground truth masks data

Cellpose achieved the highest mean average precision on unseen data when compared to StarDist and a U-Net, as demonstrated in the work by Vijayan et al. Using Cellpose to create the initial ground truth dataset was consistent with their findings [15]. To increase the segmentation performance on our specific dataset, preprocessing was applied. Nevertheless, the output of the Cellpose nuclei model was not sufficient to be used as a ground truth dataset for further training. Manual examination revealed distorted, pixelated and grossly segmented cell boundaries. Nuclei masks were not evenly connected across the z-slices, noise was segmented into non-uniform shapes and nuclear shapes were incorrectly segmented into myriads of small unconnected masks.

Manual correction of all segmentation errors was impractical due to time constraints and expert resource limitations. Therefore, to address this and improve the segmentation performance of newly trained DL models, a post-processing step was applied to the segmentation masks. The steps aimed to smooth out falsely segmented, pixelated cell boundaries, merge masks corresponding to clustered nuclei and eliminate small, irrelevant masks. It is important to note that Cellpose segmentation only generated masks for background and cell classes, lacking an edge class. Research by Weigert et al. (2020) shows that incorporating an edge class significantly improves the separation of clustered and touching nuclei instances [14]. By predicting boundaries around nuclei, DL models can effectively distinguish them within the 3D image space. Hence, an edge class was added to the output mask images. The Cellpose segmentation output consisted of 16-bit TIF files with a shape of $(21, 512, 512)$. These files encoded binary segmentation classes for background and nuclei, additionally integrating unique intensity values ($>= 1$) for consistent nuclei instance labeling across all z-slices.

The post-processing pipeline applied the following steps to every Cellpose output image. First, it loaded and converted the image into a NumPy array for efficient processing. A crucial step involved maintaining the original instance labeling while adding edges and smoothing the masks across the entire 3D volume of the image. To achieve this, the pipeline identified the total number of unique masks (represented by intensity values) within the array. It then iterated through each intensity value, corresponding to a specific mask, slice by slice (among the z-axis) using a double loop. If a mask with the corresponding intensity value was absent from a specific slice, subsequent operations were not performed. Otherwise, the area of the mask was compared to a certain**minimum area threshold**. Masks falling below this threshold were removed. Qualifying masks were processed with a 2D convolutional filter with a small kernel to **smooth out their boundaries** and **eliminate unwanted convex artefacts**. Next, the smoothed mask was used to generate its corresponding edges through **dilation and subtraction** operations. Finally, the edge pixels were scaled to the **edge classifier value of** 1 and merged with the smoothed mask. The resulting post-processed ground truth was saved as a TIF file with the same dimensions as the input. This file encoded intensity values for $background = 0$, $edge = 1$ and $nuclei >= 2$. When analysing the images, a class imbalance distribution of approximately 93% background, 1% edge and 6% cell class was found in our data set. Visualizations of this process are presented in the results section.

2.8 Validation Metrics and Loss Functions in Semantic Segmentation

Biomedical segmentation seeks to combine instance and semantic segmentation at once. The StarDist selected for use is able to assign pixel wise labels and distinguish instances from each other with unique identifiers. U-Net on the other hand, focuses on identifying the class label for each pixel in the image only. This work used several validation metrics intended for semantic segmentation to score the classification of neighboring pixels (e.g. nucleus).

2.8.1 Validation Metrics

The success of the prediction in this work is measured by comparing each pixel class prediction to the ground truth pixel label. In the binary case, the class will be either $0 = background$ or $1 = cell$. In the multi class prediction problem, per definition the classes are $0 = background$, $1 = celledge$ and $2 = cell$.

To asses the performance of prediction, the following scores are used in this work, calculated based on the per-class classification results: true positive (**TP**), false positive (**FP**), true negative (**TN**), and false negative (**FN**). This allows for the definition of various metrics for comprehensive evaluation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Accuracy (Eq. 1) measures the overall success rate of the classified class. It scores the proportion of correctly classified pixels out of all predictions made.

Precision (Eq. 2) and recall (Eq. 3) are emphasizing the over and under classification of a model, providing a possibility to penalize FP and FN pixel classification. The refined metric precision is also known as the positive prediction value and describes the proportion of TPs in the subset of all positive predictions made by the model. Additionally, recall is known as the sensitivity of a models classification performance. The importance of this measurement is demonstrated by medical test that have to detect every SARS-COV-2 virus in a sample of one million cells to be reliable.

The **F1-score** or also known as **Dice Score** (Eq. 4) calculates the harmonic mean of precision and recall, creating a balanced perspective of both metrics. Penalizing both FP and FN, the F1 score is especially useful in medial segmentation where both errors are treated as problematic.

As an alternative to the Dice Score, this work used a similarity descriptor developed by Grove Karl Gilbert in 1964 [54]. The **Jaccard index** is used in the field of object recognition and also called **Intersection over Union (IoU)** score, seen in Eq. 5. It measures the overlap of two bounding boxes or sets, i.e. the true and the predicted segmentation mask, and results in 1 for a perfect overlap and 0 for no overlap [38, 41]. The IoU score penalizes FP and FN more strictly than the DiceScore and thus provides a direct measure of segmentation quality.

$$DiceScore/F1Score == Precision\ +\ Recall = \frac{2 * TP}{2 * TP + FP + FN} == \frac{overlapArea}{totalArea} \tag{4}$$

$$JaccardScore = IoU = \frac{TP}{TP + FP + FN} = \frac{Intersection}{AreaofUnion} = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

2.8.2 Loss Functions

The baseline used in this work is the least intuitive but most commonly used loss for pixel-wise segmentation of multiple classes: the cross-entropy loss (**CE**) [55]. We are interested in the pixelwise disparity of the predicted probabilities to the boolean ground truth classes of the complete image. To address the class imbalance, one can assign a weight factor ω_i for each class, to emphasize or impair a class which describes the weighted cross entropy loss (**WCE**). Equation 6 shows the definition, with the **ground truth class** y_n and the **softmax transformed logits outputs** p_n, for $n \in 1, .., N$ pixel input associated to the $i \in 1, ...c$th class, summed up to provide a numerical measure of the models prediction error.

$$WCE\_L = -\sum_{i}^{c} \omega_i y_{i,n} \log(p_{i,n}) \tag{6}$$

With the WCE loss we are able to find class weights, that forces the model to weight feature learning for under represented classes more than for other classes. Since the search for suitable weights is arbitrary, it does not guarantee the best performance for each data set.

WCE does have its limitations in underrepresented hard to classify classes, which this work is dealing with. Hence, we implemented the work of Lin et al. [36], who developed Focal Loss (**FL**) as tuned CE. The authors prove that it outperforms WCE and CE by defining a function that regulates the class weights according to the networks prediction certainty [36]. Equation 7 describes FL as combination of CE with the automatic weight scaling in the γ controlled regularization term $(1 - p_{i,n})$ and introducing α, γ as additional hyperparameter to regulate the power of weight adjustment.

$$FL = -\alpha \sum_{i}^{c} (1 - p_{i,n})^{\gamma} y_{i,n} \log(p_{i,n}) \tag{7}$$

To compare the effectiveness of weight scaling and regularization terms, this study employed the IoU score as a loss function. Equation 8 displays the calculation of the IoU loss for each class c, considering every predicted pixel p and its corresponding ground truth pixel value y. A small value, ϵ, is added as a smoothing term to avoid division by zero. The IoU loss is particularly well-suited for this purpose due to its ability to strictly penalize FP and FN, effectively acting as a regularization term. Compared to the Dice Score, the IoU loss's strict penalization makes it more suitable as a loss function.

To obtain a single overall loss value, the IoU scores for all classes are averaged and then subtracted from 1 (Eq. 8. This ensures that a perfect overlap between the predicted and ground truth segmentations results in a loss of 0, allowing for optimization.

$$IoULoss = 1 - mean\left(\frac{\sum_i^c \omega_i p_{i,n} y_{i,n}}{\sum_i^c \omega_i p_{i,n} + y_{i,n} - (\sum_i^c \omega_i p_{i,n} y_{i,n}) + \epsilon}\right) \tag{8}$$

2.9 Manual Labeling Tools

This study investigated two labeling tools and their effectiveness in improving segmentation quality and effortless accurate labeling.

Cellpose, as a significant contribution to 3D deep learning tools, provided a built-in user interface that claimed to be easy to use. Research confirmed that Cellpose was the most competent tool for creating a labeling dataset from a few ground truth examples and achieving the best generalization. Consequently, the Cellpose GUI was selected to apply the pretrained Cellpose model, manually segment, and correct the output. Special emphasis was placed on the promised user-friendliness for potential integration into biological researchers' workflows.

Cellpose, as a segmentation tool, was primarily designed for use with its built-in models and lacked significant cross-platform features. The GUI allowed for immediate correction of segmentation results, automatically saved manual changes, and highlighted detected nuclei on the original image. The masks, flowcharts, metadata, and original TIF image could be saved in a Cellpose-generated Python format (.npy). Additionally, the masks were saved as grayscale TIF files, with background labeled as 0 and nuclei instances labeled as $1, 2, \ldots, n$. While originally designed for 2D cases, the Cellpose GUI could merge manually segmented nuclei per slice into a 3D volume. However, 3D rendering was not possible for closer inspection. Cellpose assisted with mask drawing, but users had to provide an area for edge detection. Per-pixel labeling was not possible, as the tool was designed for mask shape correction. Erasing single pixels in the mask was also not supported, requiring deletion of the entire produced mask.

Segmenting a 3D image could take up to 0.5 hours without a GPU, significantly hindering efficient ground truth generation. If the model was not well-trained for specific images, the segmentation results were often inaccurate, necessitating full deletion of the mask object to improve accuracy.

Labkit serves as a plugin of the open source image processing package Fiji [56] and was easy to install [57]. It was designed to work seamlessly with large biological image data, has a detailed online documentation, works with GPU and high-performance clusters and is developed for pixel-based segmentation. The initially overwhelming menu and user interface was quickly understood thanks to well documented online resources and a quick learning curve. Compared to Cellpose GUI, Labkit boasts broader data compatibility and applicability across various image analysis tasks. Importantly, Labkit utilizes TIF files for both input and output, ensuring seamless integration with other tools and platforms [57, 56].

To acquire segmentation masks, raw images were loaded into Fiji using the Labkit plugin. The random forest classifier within the segmentation module was employed for pixel-wise segmentation. Various filters, including Gaussian, min/max, Hessian, and eigenvalue structure tensors, were available for customization. Basic filter settings were applied based on image characteristics. To train the classifier, manually drawn pixel masks were provided as examples. Unlike Cellpose GUI, only a general outline of regions was required, simplifying the labeling process. An iterative approach was used, with additional examples incorporated and the classifier retrained to refine segmentation. The classification process was efficient, taking approximately 5 minutes per image, with GPU acceleration possible.

Explicit drawing tools in Labkit enabled precise correction of the exported binary segmentation masks. The NIS 3D viewer was used for 3D visualization to assist in separating overlapping nuclei. The binary segmentation class was exported as a TIF image, and the classifier was saved for future use. The exported TIF file was then loaded into Fiji for instance segmentation using the MorphoLibJ plugin's "Connected Component Labeling" function. A connectivity parameter of 6 was used to convert the binary image into a 16-bit grayscale image containing individual mask instances. To further refine instance segmentation, the original image was opened in Labkit, and the imported instance segmentation masks were overlaid. Manual adjustments were made to achieve optimal segmentation accuracy for each mouse. The final segmentation result was saved as a TIF linked to the original image.

This workflow enabled precisely masking challenging microscopy images of various types with a fair time effort of 15-20min per image and accurate results for downstream DL model training and validation.

2.10 Final Performance Validation

To assess the actual segmentation accuracy and retrieve a metric, a final segmentation score was created and applied to manual labeled evaluation images. The dataset marked in red in Figure 2 suits as test set to be applied as last step in training a DL model. Naturally, this set of images need to have masks that are as accurate as possible. Therefore, Labkit was used to manually draw masks for one 3D images per channel. This manual labeling was conducted rather strict, masking only the most evident nuclei in the images. In that way, it was assured that the compared models are evaluated on their ability to find the explicit signal nuclei as fundamental approach.

First, trained U-Net3D and StarDist models were used to predict segmentation masks for unseen validation images. The resulting output TIF files $(21, 512, 512)$ were standardized to a binary class (background and nuclei). U-Net3D did not produce instance labels, so the connected component labeling function of the cc3d package was used to identify and label 3D components [58]. By looping through the unique labels of each image, mask characteristics such as mask areas, centroids, bounding boxes and cloud-coordinates where saved for downstream analysis.

To evaluate predicted masks, a comparison was made with corresponding ground truth masks from the manual validation set and composed in the Evaluation Score (Eq. 9). Using the ground truth mask coordinates, the corresponding regions in the predicted masks were identified and isolated. A DiceScore was calculated for each predicted mask against its ground truth, and a sub-region DiceScore within a certain tight bounding box of the ground truth mask was computed. A weighted combination of these scores determined the overall segmentation success, as detailed in Equation 9. The separation of clustered nuclei was prioritized with a weight of 1.05 over subregion segmentation (weight of 0.95). Sub-regions were necessary to address the model's oversegmentation tendency, especially in regions with noise or clustered nuclei. The cc3d package's connected component labeling function struggled with noisy or clustered nuclei, often combining them into a single label. By isolating nuclei within sub-regions, false positive segmentations were effectively eliminated, enabling more accurate evaluation.

$$EvaluationScore = \frac{(0.95 * subregionScore + 1.05 * fullScore)}{2} \tag{9}$$

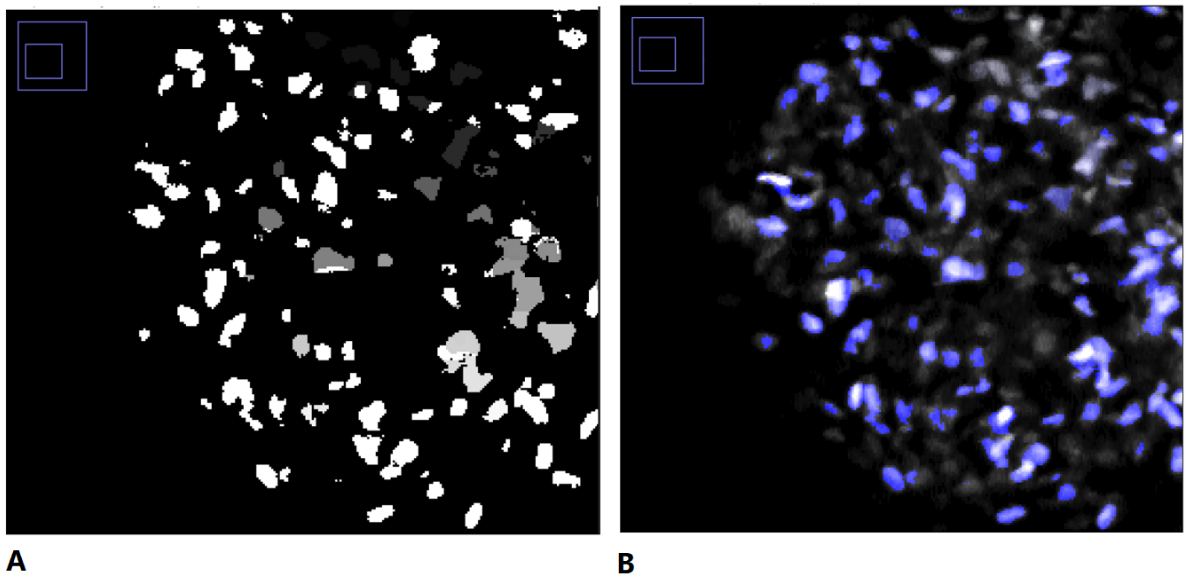

Figure 4: **Raw Cellpose Segmentation Results.** Panel **A** presents the raw segmentation masks, where each unique gray value represents an individual nucleus. Here, poor segmentation and clumping of nuclei are evident. **B** depicts an overlay of the binary segmentation mask (blue) on the original microscopy image. While strong signals are segmented well, noisy regions pose a challenge for Cellpose.

3 Results

3.1 Post-Processing

Due to absence of expert-labeled ground truth data, the initial segmentation masks generated by the Cellpose model demonstrated inaccuracies. To address this limitation, a post-processing notebook was applied to refine the masks, creating more precise ground truth annotations. Accurate nuclei segmentation is crucial for the subsequent DL model training and directly impacts their overall performance.

Cellpose exhibited limitations in accurately segmenting nuclei in our dataset as shown in Figure 4. Segmentation masks instances visualized in distinct grey values for differentiation. Panel A of Figure 4 depicts distorted nuclei shapes with frayed out edge structures, instances split into multiple incorrect masks, and over-segmentation of noise surrounding true nuclei. Each gray value represents a nucleus identified by Cellpose. Figure 4B shows the same binary segmentation mask as an overlay (colorized in blue) over the corresponding microscopy image. It demonstrates, the amount of nuclei missed by the model. In addition, the model was not able to separate single nuclei in densely packed regions. Similarly, the model struggled with noise while correctly segmenting strong signals.

Figure 5 displays the outcome of the post-processing pipeline step. A detailed view of a 2D slice showcases the transformation: irregular Cellpose mask edges (unrealistic nuclei shapes) are smoothed and merged into improved ground truth masks. Notably, instance segmentation is preserved. Each nucleus retains its unique gray value label in 3D space, but masks smaller than

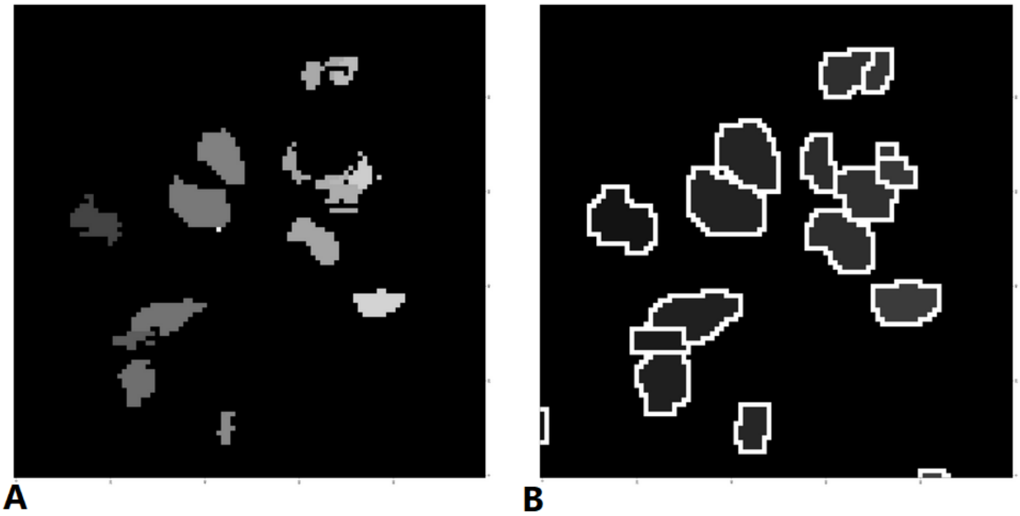

Figure 5: **Post-Processing of Cellpose Output.** Figure **A** shows the raw segmentation output from Cellpose, while **B** demonstrates the effects of the post-processing pipeline step. This process effectively smoothed and merged irregular mask shapes, resulting in a more accurate ground truth mask. The figure provides a detailed view of a slice from the B10_T01_Cyr microscopy image.

20 voxels (in 2D slice) are eliminated. By looping through every intensity value individually, the post-processing pipelines assures that every nucleus gains an edge, which separates clustered instances from each other along the z-dimension. Consequently, new deep learning models can learn a more accurate shape representation from the microscopy images.

To reduce the amount of oversegmentation, division of single nucleus into multiple masks and to align the segmentation mask shape to a more realistic round shape, certain image processing steps were included. Figure 6 demonstrates the effectiveness of the chosen image processing methods, resulting in a more accurate ground truth. Figure 6A highlights the limitations of post-processing without top-hat and dilation steps, while Figure 6B showcases the final result with improved nuclei shapes and the removal of small perturbing instances. With these improved and corrected shapes, they serve as better ground truth segmentation masks for the subsequent DL model training.

Figure 7A illustrates a challenging case of post-processing in a densely clustered nuclei region. The noisy and overlapping nature of these nuclei led Cellpose to oversegment the area, resulting in merged nuclei and unrealistic shapes. Despite the applied post-processing operations the extensive masking of uncertain nuclei boundaries hindered complete correction. As a result, Figure 7B presents a dense, noisy oversegmentation that is unsuitable as a ground truth mask. However, the instance-wise edge creation effectively separates individual masks even in highly cluttered regions, visualized as white boundaries.

3.2 StarDist Training

As introduced in the Methods section, the pretrained StarDist model was employed as a state-of-the-art reference benchmark. Fine-tuning, prediction, and data processing were conducted using

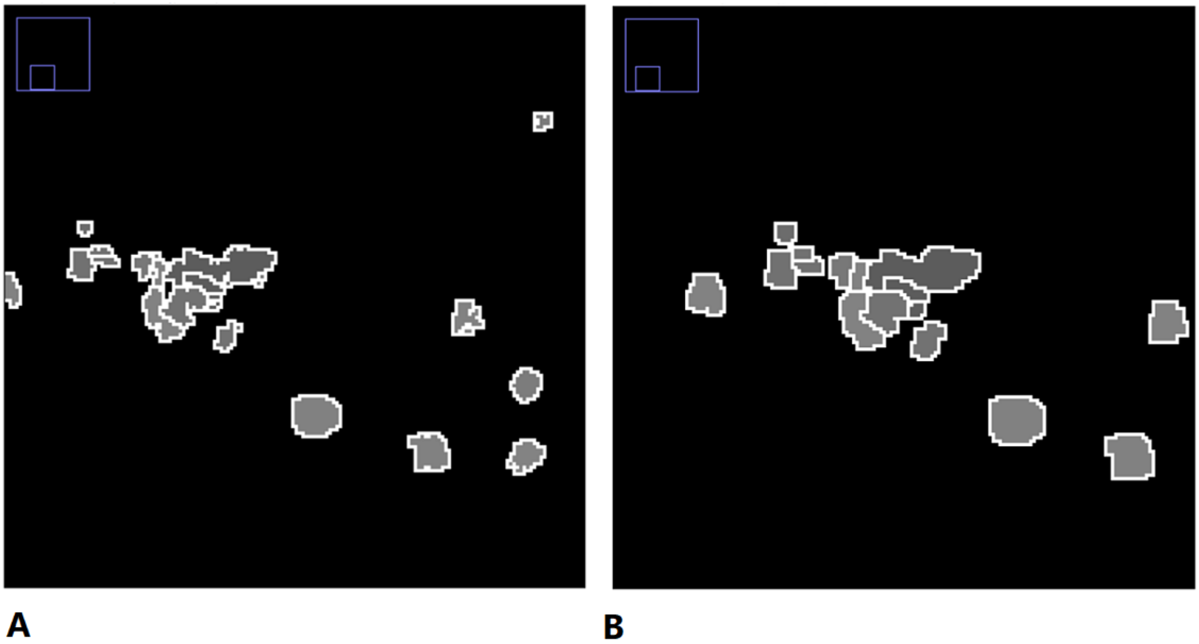

Figure 6: **Effect of image processing methods**. Panel **A** shows the intermediate result of post-processing without top-hat and dilation. Panel **B** displays the final post-processed segmentation mask with improved core shapes and edges.

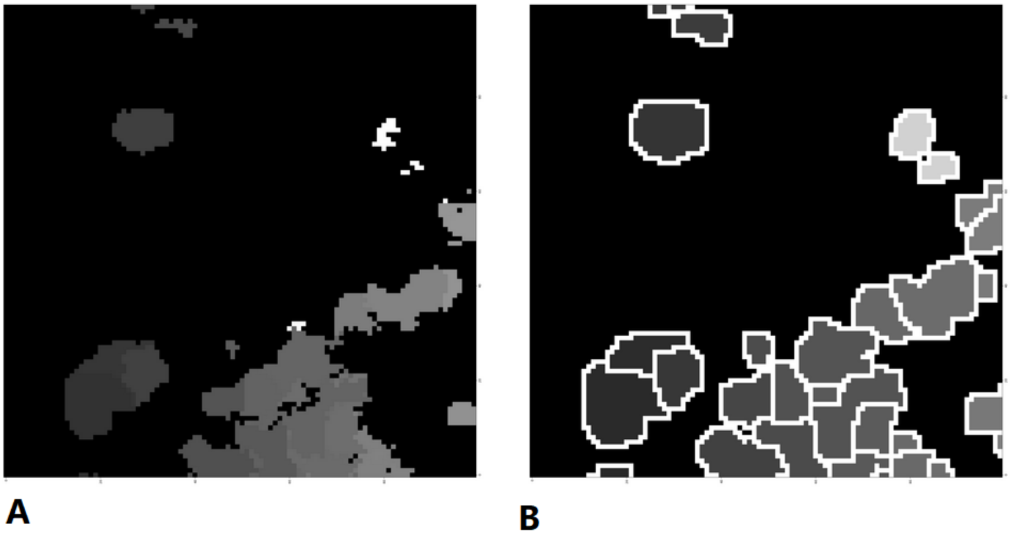

Figure 7: **Post-Processing Dense Cellpose Segmentation Output.** Displayed in **A** is the raw Cellpose output of a challenging region with densely clustered nuclei and noise. The post-processed result in **B** demonstrates the limitations in generating accurate ground truth masks. Despite instance-wise edge creation effectively separating individual masks, oversegmentation and noise remains due to the challenging nature of the image data.

Table 3: Comparison of StarDist training metrics. Various StarDist model configurations, trained on an Nvidia RTX 2080Ti, and the resulting training metrics are shown here. All models included a ResNet backbone except StarDist-U-Net, which was trained with a U-Net architecture. The best results per metric are highlighted in bold.

Wandb-ID	Time	Input	Epochs	n-Rays	T-Loss	MAE	IoU
StarDist-Resnet	2h18min	(12,512,512)	400	96	0.267	1.165	**0.630**
StarDist-U-Net	1h32min	(12,512,512)	400	96	0.272	1.120	0.615
small-patch	**6min**	(4,128,128)	400	96	0.243	0.935	0.6231
small-patch-128Ray	**6min**	(4,128,128)	400	128	0.243	0.933	0.624
small-patch-1k	14min	(4,128,128)	1k	96	**0.211**	**0.796**	0.559

the provided Jupyter notebooks [14]. GPU-accelerated training was performed on the ALICE server, enabling GPU RAM intensive tasks. Table 3 outlines the model configurations and training parametersVariations in training patch size, StarDist backbone architecture, number of rays, and epochs were investigated. Direct comparison of training success with U-Net was ineffective due to the differences in loss function. Furthermore, the absence of wandb style visualization and custom F1-scores limited model assessment. Instead, training performance was evaluated based on epoch loss curves, measuring the distance mean absolute error (MAE), epoch loss and distance IoU metrics.

Table 3 provides an overview of training metrics for all tested configuration. It shows, that increase the n-rays did not benefit the training process. Increasing the number of rays did not improve performance. While small patch sizes significantly reduced training time (6 minutes versus 1.5-2 hours), this advantage did not translate consistently to improved IoU, indicating a potential trade-off between speed and segmentation performance. Extending training epochs to thousand lowered the loss but did not enhance IoU, indicating model overfitting. Despite lower loss values, StarDist ResNet achieved the highest IoU of 0.630 as the most important metric. Based on these findings, StarDist ResNet, U-Net, and small-patch were selected for further analysis and performance evaluation.

Figure 8 presents a comparative analysis of loss curves for the most relevant models, evaluating training success. StarDist-smallPatch exhibited the lowest training loss, indicating superior training convergence compared to StarDistResnet and StarDistU-Net. However, the IoU score revealed the opposite trend, with small-patch demonstrating the poorest performance. The IoU score was calculated on the same validation images as in the U-Net training. ResNet consistently outperformed U-Net across all metrics.

3.3 U-Net Training and Parameter Tuning

This section will present the hyperparameter tuning results of the U-Net-3D implementation. The online tool wandb [47] was utilized to facilitate tracking and visualizing the experiments. This platform assigns unique names to the experimental runs, simplifying differentiation between them. The names are noted down in the results tables as **”Wandb-ID”** such as ”fine Yogurt”, ”golden Pond” or ”comic Firefly”.

Every experiment was executed on the ALICE HPC, using GPU nodes such as Tesla A100 and GeForce RTX 2080i. The default configuration is a model with input shape of $(15, 500, 500)$, lr

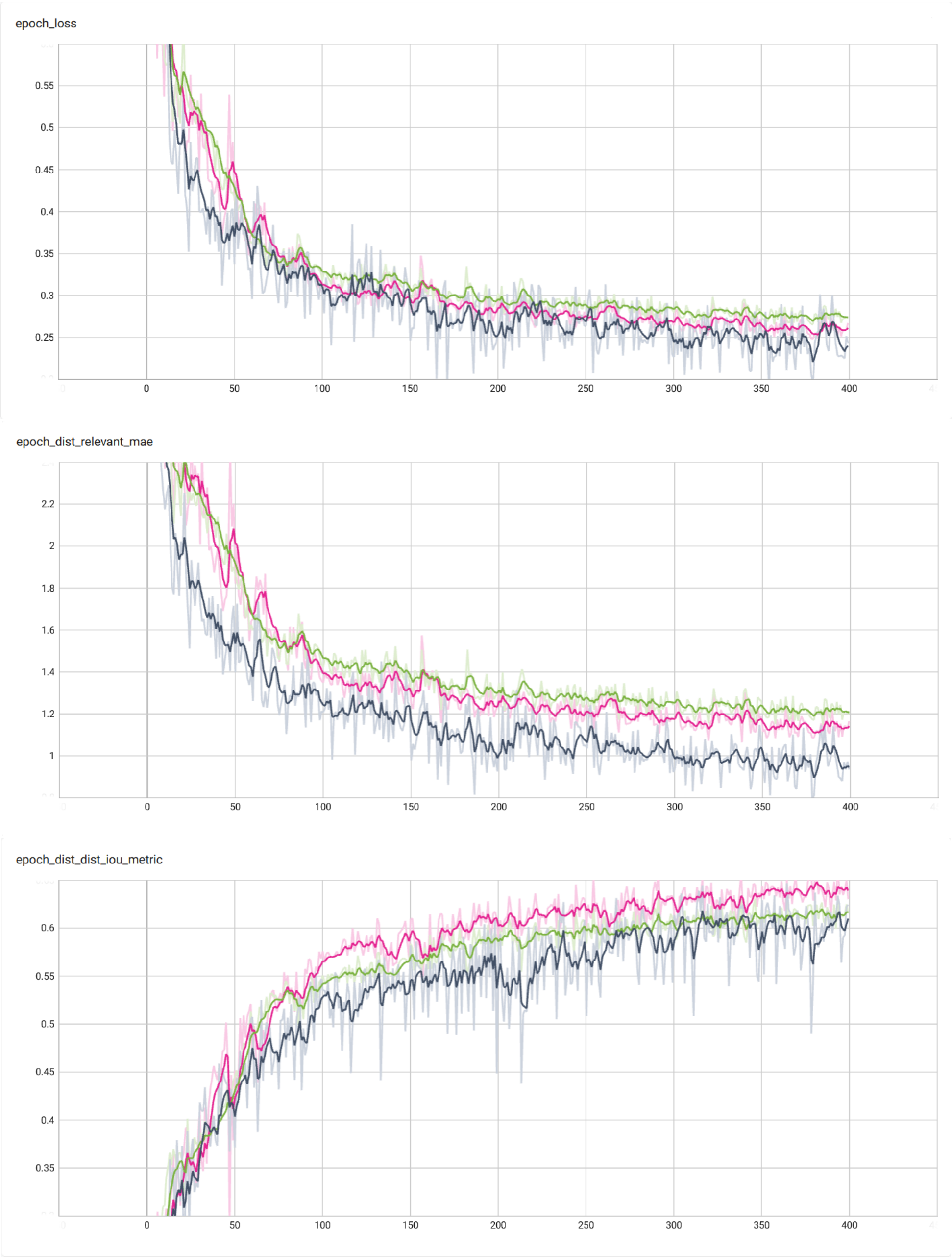

Figure 8: **StarDist training losses.** Visualized here are the training loss curves for three StarDist models: ResNet (pink), U-Net (green), and SmallPatch (black). The y-axis represents the loss value, while the x-axis indicates the number of training epochs. While minimal differences in training curves were observed between U-Net and ResNet, the latter exhibited marginally lower MAE and total training loss. The SmallPatch model consistently outperformed both U-Net and ResNet in terms of loss, although it achieved a lower IoU score compared to the other two models.

Table 4: Comparison of Loss Function and weight combination. Displayed here, are the results of the loss function experiments, the best and second best F1 scores are highlighted.

Wandb-ID	**weigths**	**Loss**	**lr**	**F1 edge/cell**	**time**
avid hill	1.0,2.5,1.5	CrossEntropy	0.001	0.000 / 0.5854	1h 14m
earnest salad	1.0, 1.0, 1.0	FocalLoss	0.001	0.395 / 0.812	1h 17m
lemon feather	1.0, 2.0, 1.5	FocalLoss	0.001	**0.410 / 0.7764**	1h 16m
genial water	1.0, 2.5, 2.5	CrossEntropy	0.001	0.509 / 0.8146	1h 17m
comic firefly	1.0, 3.0, 2.0	CrossEntropy	0.001	**0.529 / 0.815**	1h 15m
unique sea	1.0, 1.0, 1.0	CrossEntropy	0.001	0.359 / 0.814	1h 15m
divine violet	1.0, 1.0, 1.0	IoULoss	0.0001	0.542 / 0.840	1h 18m
skilled meadow	1.0, 1.0, 1.0	IoULoss	0.001	0.521 / 0.840	1h 15m
golden pond	1.0, 3.0, 2.0	IoULoss	0.0001	0.552 / 0.850	1h 20m
fine yogurt	1.0,2.0,1.5	IoULoss	0.0001	**0.555 / 0.853**	1h 17m

$= 0.001$, Adam as optimizer, image normalization, $[64, 128, 256]$ as network feature layers and 2 validation images (GFP, Cy3) cropped to $(2, 200, 200)$ for visual performance comparison in wandb. These configurations are the results of previous testing on smaller GPU and recommendation out of the authors papers [31, 14, 59].

To estimate the performance, not the loss but the segmentation metrics were used. The DiceScore (F1) of edge and cell segmentation was considered as most impact.

3.3.1 Loss Function and Weights

Table 4 presents a comparative analysis of loss functions and weight configurations. Loss function such as Cross Entropy (CE) and Focal Loss (FL) incorporate class weighting with certain chosen factors to address imbalanced datasets. These factors are noted down as [x, y, z] for x=backgound, y=edge and z=cell class. FL, with gamma = 5 exhibited the worst performance in edge and cell-F1. CE demonstrates better results, except "avid hill" which drops mid training and fails to segment anything. The best results are exhibited by the IoU loss, which achieves better performance than CE and FL in any parameter configuration. Reaching (0.555 / 0.853) at max for edge-F1 and cell-F1 respectively and a weight of [1.0,2.0,1.5] it beats the best CE run "comic firefly" with (0.529 / 0.815) and stronger weight emphasize of the edge class [1.0, 3.0, 2.0]. Interestingly, the same weights decrease the performance of IoU, as seen in "golden pound" and a result of (0.552 / 0.850). CE on the other hand benefits from stronger weights for edge and cell. This can be seen in the edge and cell weight of "unique sea" with [1.0, 1.0, 1.0] and "genial water" with [1.0, 2.5, 2.5] and results of (0.359 / 0.814) and (0.509 / 0.8146). Only here, the weighted FL loss "lemon feather" surpasses CE in the edge F1 score (0.410 / 0.7764), but underlays in cell segmentation.

The only difference in comparison between CE and IoU loss was the lr in those experiments. Whereas CE was executed with a lr of 0.001, IoU had a lr of 0.0001. This leads us to the next experiments, CE FL and IoU loss with different lr settings. As described in the next section, IoU with lr of 0.001 still outperforms CE and FL with default configurations as seen in run "skilled meadow" and F1 scores of (0.521 / 0.840).

Table 5: Learning Rate and gamma

Wandb-ID	weigths	loss	lr	F1 edge/cell	time	gamma
smart sea	1.0, 2.0, 1.5	CrossEntropy	0.001	0.498 / 0.797	1h 17m	n.a.
glad vortex	1.0, 2.0, 1.5	CrossEntropy	0.00001	0.376 / 0.814	2h 48m	n.a.
amber glitter	1.0, 3.0, 2.0	CrossEntropy	0.0001	**0.557 / 0.844**	2h 51m	n.a.
hearty voice	1.0, 2.0, 1.5	CrossEntropy	0.005	0.482 / 0.773	2h 49m	n.a.
deep cherry	1.0,2.5,1.5	CrossEntropy	0.0001	0.524 / 0.838	1h 23m	n.a.
skilled sun	1.0, 1.0, 1.0	FocalLoss	0.005	0.301 / 0.811	2h 45m	7
glowing dew	1.0, 1.0, 1.0	FocalLoss	0.001	0.441 / 0.835	2h 50m	7
drawn firebrand	1.0, 1.0, 1.0	FocalLoss	0.0001	**0.459 / 0.841**	2h 48m	7
whole galaxy	1.0, 1.0, 1.0	FocalLoss	0.0001	0.418 / 0.822	1h 20m	8
skilled meadow	1.0, 1.0, 1.0	IoULoss	0.001	0.521 / 0.840	1h 15m	n.a.
fine yogurt	1.0,2.0,1.5	IoULoss	0.0001	**0.555 / 0.853**	1h 17m	n.a.

3.3.2 Learning Rate and Gamma

Table 5 presents the results of taking the best configurations out of the previous run and investigating the behavior of increased and decreased lr. The experiments prove, that the loss functions suffer from greatly diverging values. Here, IoU is most sensitive to and increased lr of 0.001 compared to 0.001, diminishing the edge-F1 score from (0.542 / 0.840) to (0.521 / 0.840) with same weights. Decreasing the learning rate while having strong class weights for CE and increased gamma for FL leads to an increase in edge-F1 and cell-F1. The default lr configuration reaching (0.529 / 0.815) was improved to (0.557 / 0.844) with run "amber glitter". For FL, a higher gamma of 7 in combination with the lr of 0.0001 leads to an improvement from (0.395 / 0.812) to (0.459 / 0.841), not surpassing CE in edge but approaching the cell same accuracy, as demonstrated in run "drawn firebarn". Choosing a higher gamma (8), performance gain stops and decreases again, as seen in run "whole galaxy". The worst performance is visible in run "skilled sun" with the highest lr of 0.005. As expected, the training was unstable and concluded in the worst edge-F1 score of 0.301. Noticeably, the FL in combination with a high gamma of 7 was still able to segment the cells precisely, what the cell-F1 of 0.811 shows. Nevertheless, FL combined best F1 scores of (0.459 / 0.841) was achieved with the decreased lr of 0.0001 same as CE. In contrast, IoU does not favor of stronger weights with lr of 0.001. The results of "golden pond" (0.552 / 0.850) perform slightly worse with weights like CE [1.0, 3.0, 2.0] compared to "fine yogurt". Its weights of [1.0,2.0,1.5] represent the best performance in F1 scores for the whole experiments so far (0.555 / 0.853).

All in all, the lr of 0.0001 benefits all loss functions and together with carefully chosen emphasize on underrepresented classes the models achieved new maximal performance values.

3.3.3 Network Architecture

Crucial part to recognize and learn patterns and features in images is the network architecture. Based on the number of parameters in each layer, the networks ability to learn varies. The following experiments added and minimized the number of convolutional blocks. To be able to train the network without exceeding the GPU resources, the model input size was reduced to 18x350x350 pixel for some runs. Table X displays the results. Reducing the feature parameter

Table 6: Network architecture and Input shape

Wandb-ID	**weigths**	**Layers**	**Loss**	**Input hape**	**lr**	**F1 edge/cell**
rare-sun	1.0, 3.0, 2.0	[64,128]	CE	15,500,500,	0.001	0.519 / 0.797
divine armandillo	1.0, 3.0, 2.0	[128,256]	CE	15,500,500,	0.001	**0.525 / 0.813**
worthy lion	1.0, 3.0, 2.0	[64,128,256,512]	CE	20,500,500	0.001	0.438 / 0.771
upbeat armadia	1.0,2.0,1.5	[64,128,256,512]	IoU	20,500,500	0.0001	0 / 0
distincitve snowflake	1.0,2.0,1.5	[128,256]	IoU	15,500,500,	0.0001	0 / 0
cosmic smoke	1.0,2.0,1.5	[64,128]	IoU	15,500,500,	0.0001	0.489 / 0.838

and double convolutional building blocks to [128,256] results in the best edge and cell F1 scores for CE with (0.525 / 0.813), as seen in “divine armandillo”. By omitting the low-level pattern recognition of the first layer with [64], the model was able to learn the details of intricate edges. Changing the layers to [64,128] results in a cell-F1 with similar results, only the edge-F1 suffers (0.519 / 0.797), as seen in “rare sun”. Nevertheless, the measured GPU utilization exceeds 29GB with the [128,256] double convolution blocks, which is significantly more than the usual $8-9$ GB of the default configuration. Comparing to the baseline, there was not a noticeable increase of performance for CE neither FL, hence the excessive GPU utilization is disproportionate to the marginal increase in performance. Adding an extra feature layer to the model while minimizing the input shape to 18x350x350 decreases the performance for CE and FL from (0.529 / 0.815) and (0.410 / 0.7764) to (0.488 / 0.786) and (0.373 / 0.754) respectively. To assure that the reduced input size had no great effect, the added layer was tested on the same configuration with run “worthy lion”. The results demonstrate that with higher GPU utilization of 20.1 GB the performance of the model still does not exceeds the baseline. Similar behavior was seen for the IoU loss. Adding [512] and changing to [128, 256] is leading to a complete failure of learning edge and cell, the whole image was segmented into the background class. Only “cosmic smoke” was able to learn properly, with [64, 128] configuration. Still, segmentation performance decreased drastically from the best to the worst of (0.489 / 0.838).

In summary, the trade of between GPU utilization, network architecture and image size could not exceed the baseline performances.

3.4 Performance Validation

This section presents the calculated evaluation scores and a visual analysis of the segmentation performance for StarDist3D and U-Net models. A custom-built analysis function assigned unique labels to each nucleus in the three ground truth evaluation images. Utilizing the label-based point coordinates of segmentation masks, 3D representations of predicted and ground truth masks were generated using matplotlib. As described in the methods section, subarray and full image Dice scores from the final performance evaluation were visualized as 3D overlays to highlight FP, TP FN and TN mask predictions.

3.4.1 Evaluation Score Table

The evaluation score introduced in Method section 2.10 was used to assess the successful segmentation performance of each trained model. Table 7 displays the results of models that stood out in the parameter training. The Score was calculated for each channel independently

Table 7: Final evaluation results. The final performance evaluation results of the trained StarDist and U-Net model configuration are presented here. Highlighted are the best averaged results and the highest channel scores of each Stardist (top) and U-Net (bottom).

	channel av.	Cy3	GFP	Hoechst
StarDistResnet	0.776	0.608	**0.896**	**0.826**
StarDistU-Net	**0.778**	**0.652**	0.879	0.804
StarDistSmall	0.663	0.434	0.844	0.712
fineYogurt	0.725	0.826	0.810	0.543
amberGlitter	0.763	**0.869**	0.844	0.576
goldenPond	0.744	0.847	0.827	0.559
divineViolet	0.715	0.826	0.793	0.527
drawnFirebrand	0.692	0.739	0.844	0.494
comicFirefly	**0.771**	0.826	**0.862**	**0.625**
glowing_dew	0.537	0.500	0.758	0.353

to identify strength and weaknesses of the model and image channels. To propose the best performing model the channel scores were averaged. Here, Table 7 presents the StarDist model as best model, beating all U-Net architectures with scores of $0.778, 776$ for StarDistU-net and StarDistResnet respectively. Closely followed by comic firefly with an averaged score of 0.771. The highest per channel scores were produced by amber glitter (0.869) for Cy3, StarDistResnet (0.896) for GFP and again StarDistResnet (0.826) for Hoechst. Especially the result for Hoechst diverges greatly from the strongest U-Net model with 0.635. It shows that Hoechst poses a challenge especially for U-Net models, whereas Cy3 represents the best performance for U-Net models.

Surprisingly, the results do not correlate with the F1 edge and cell segmentation results from the parameter tuning experiments. Fine Yogurt, Amber Glitter and Drawn Firebrand achieved the best F1 values in the experimental comparison, but fell short of these standards in the evaluation. As a result, Comic Firefly impressed with average F1 scores and the best average rating. Only amber glitter achieved the highest evaluation score for Cy3 and carried over the success from the F1 values to the final evaluation score.

3.4.2 Visual examination

To further investigate the results in Table 7, several representative examples are shown. The provided visualization illustrates the performance of a prediction model in relation to a ground truth. The assignment of an image to case 1 or case 2 is labelled in the top left-hand corner of the image. The key elements are:

- **Overlap (Yellow)**: This represents areas where the prediction of the model correctly matches the ground truth.
- **False Positive:** These are regions where the model predicted an object or area that does not exist in the ground truth. This is known as oversegmentation. Shown in **purple for case 1** and **green for case 2**.

- **False Negatives (Purple):** These are areas where the model failed to detect an object or area that is present in the ground truth. This is known as undersegmentation and shown **only in case 2**.

Figure 9 and 10 demonstrate the ability of the model to separate nuclei within challenging clustered regions, emphasizing the distinction between subarray and full image score results. Figure 11 highlights the superior segmentation performance of amber glitter in the Cy3 channel, seen in Table 7. It presents a consistent segmentation along the z-axis. To demonstrate the impact of merged nuclei on Dice score, a nucleus with adjacent neighbors in the z and x-y planes was selected. Figure 12 compares the best-performing model, StarDistResnet, to lower-performing models (amber glitter, comic firefly, glowing dawn) ranked by decreasing full image Dice score, underscoring the significance of accurate edge segmentation.

To evaluate detailed accuracy of the best performing models, a nucleus with a concave shape was analyzed. Figure 14 exhibits segmentation results for comic firefly, amber glitter, fine yogurt, and StarDistResnet. Interestingly, model performance deviated from the overall evaluation scores, with StarDist demonstrating difficulty in segmenting the complex shape while comic firefly captured more details. However, all models failed to accurately represent the nucleus's concave structure.

4 Discussion

The following discussion aims to interpret the findings, explore their potential implications and outline directions for future research. The findings conclude that the U-Net architecture achieves good segmentation performance. Trained from scratch, using only a small number of 3D images that were improved in quality by preprocessing, the U-Net can compete with the state of the art StarDist segmentation performance. Using Cellpose as initial ground truth generation, refining the masks automatically with image processing and finally training a versatile and efficient open source network like U-Net successfully reduces the human annotation effort and the demand for large training datasets. The segmentation performance of StarDist compared to U-Nets suggests that the need for training a model on a specialised image dataset only, does not guarantee better performance than fine tuning pretrained models with the exact same image data.

Nevertheless, the findings do not answer the question weather the fine tuned performance can be exceeded with better image and annotation quality or if it is the utmost. Comparing the distance based segmentation of StarDist with the pixel wise classification confirms U-Nets efficacy and potential in simplicity. The findings of the loss function demonstrate, that the class imbalance in 3D images can only be addressed through specialised loss functions or heavy weight factors that correct for underrepresented class examples. The decrease in segmentation performance when extending the U-net architecture indicates that the gray scale nuclei images do not benefit from large feature representation and efficiency in architecture and computational resource is in favor of this work.

4.0.1 Cellpose and Postprocessing

Cellposes ability to serve as an initial ground truth generator was proven experimental in the research of Vijayan et. al. and is congruent with our findings [15]. The power to generalize with

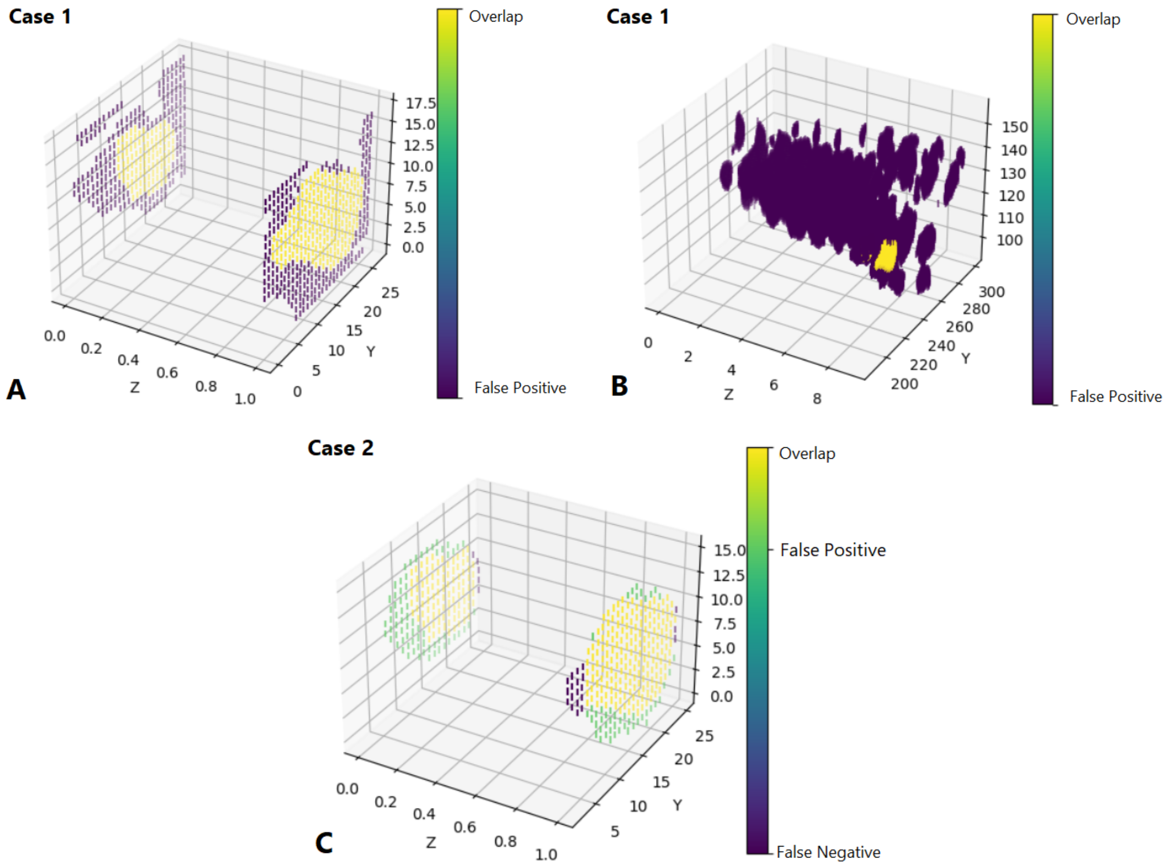

Figure 9: **Segmentation of clustered region.** This Figure visualizes the 3D segmentation mask together with the ground truth segmentation mask. Here, the segmentation of gt label **197** (Hoechst) from model drawn firebrand and model StarDistResnet are put into comparison. In **A** the **subarray** segmentation of drawn firebrand with a DiceScore of **0.616** is shown, in **B** the segmentation of the **full image** DiceScore with **0.044**.**C** exhibits the **subarray** segmentation of StarDistResnet with a score of **0.799**. StarDistResnet was able to correctly separate that nuclei in the clustered region, hence achieving the same score for the full image. In **Case 1**, overlapping regions between ground truth and prediction are highlighted in **yellow**, while **purple** indicates false positive segmentations. In **Case 2**, false positive regions are displayed in **green** and false negative segmentations are colored in **purple**.

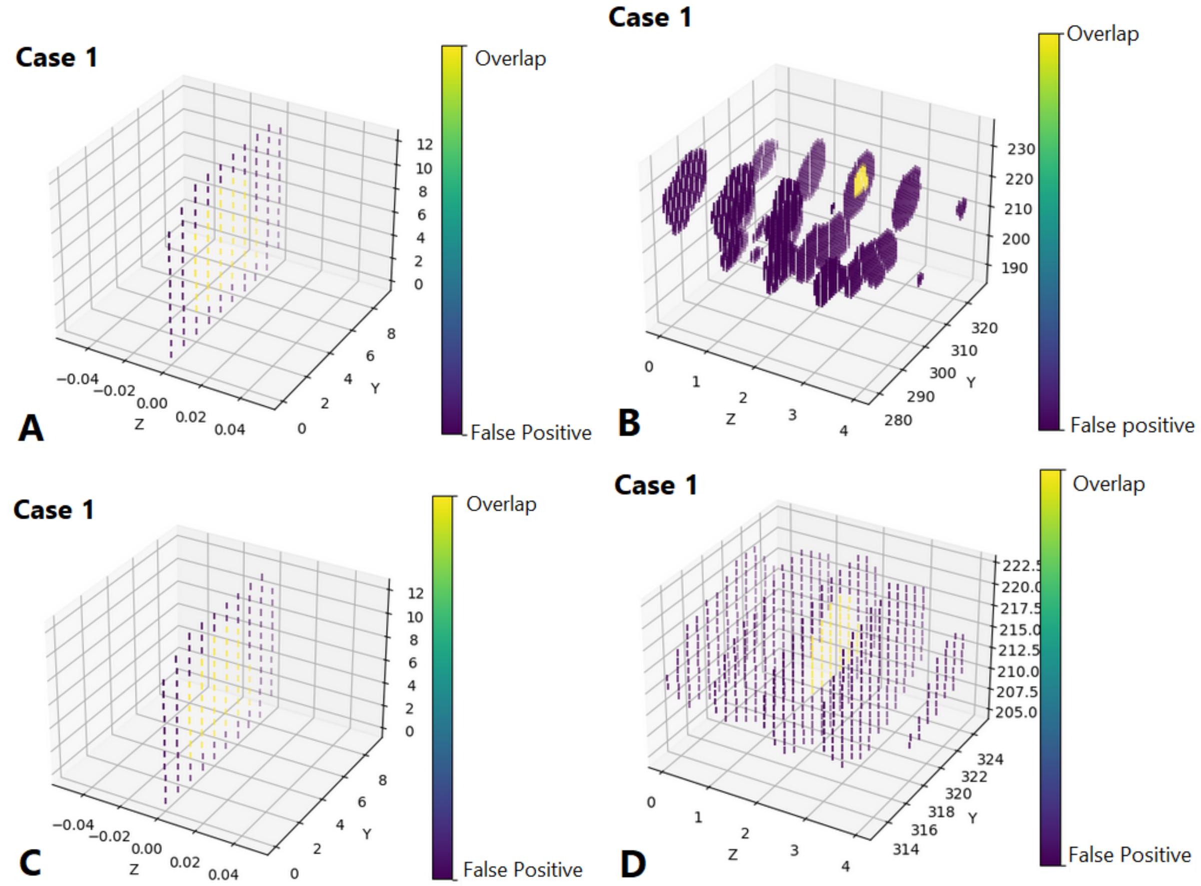

Figure 10: **Clustered nuclei Comparison for U-Net.** This Figure illustrates an example of failed and successful separation for the U-Net model on nuclei label **92** (Hoechst). In **A** drawn firebrand is exhibited with **sub array** DiceScore of 0.529, in **B** its **full image** DiceScore of **0.031**. **C** and **D** display segmentation results of comic firefly with **sub array** score of **0.532** and **full image** DiceScore of **0.131**. Comic firefly achieving the highest F1-edge score, proving a better nuclei separation than drawn firebrand, who achieves the higher F1-cell score. In **Case 1**, overlapping regions between ground truth and prediction are highlighted in **yellow**, while **purple** indicates false positive segmentations.

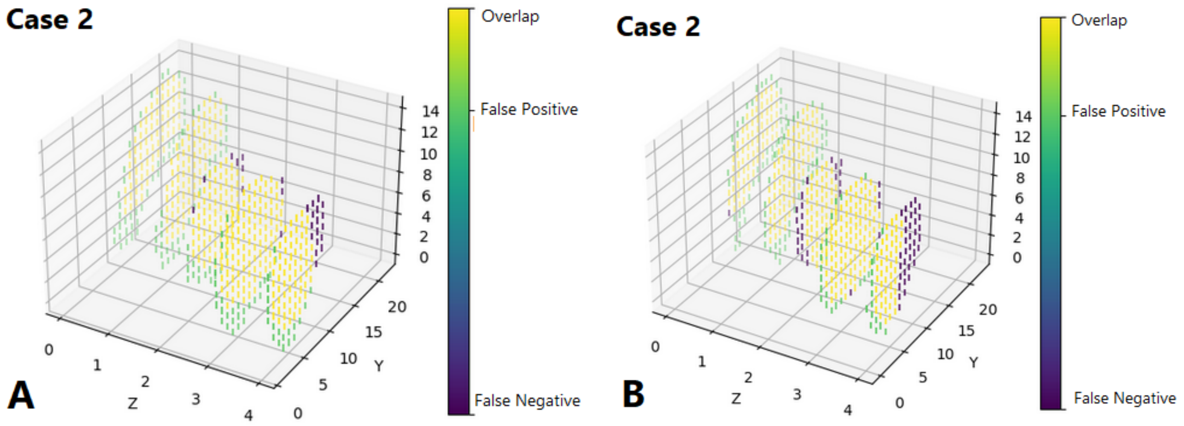

Figure 11: **Highest Cy3 segmentation value ”amber glitter”** This figure demonstrates the best segmentation of the highest Cy3 result (amber glitter shown in **A**) compared to the second worst Cy3 result (drawn firebrand in shown in **B**). Amber glitter achieves a **subarray** dice score of **0.822**, and drawn firebrand in **B** achieves a **subarray** dice score of **0.788**. During training, both models reached a similar cell F1 value, only the edge F1 value is lower for drawn firebrand. In **Case 2**, false positive regions are coloured in **green** and false negative segmentations in **purple**.

unseen data originates most likely from a diverse training set, including a variety of different cell- and microscopy - type images and additionally natural shape patterns such as shells and stones [30] . Even though, no 3D images were used for training the stitching strategy seems to reveal the relations in the 3D dimension, which is promising for all future 3D model application that could be based on existing 2D training sets.

Regardless the practicability, visual inspections as seen in the result section (Figure X) revealed a devastating poor segmentation quality. Cellpose greatly over segmented noise, divided nuclei into numerous tiny parts and failed to separate clustered nuclei. The applied post-processing strategy of the output segmentation masks reduced tiny artifacts, emulated round shapes and separated clustered nuclei by introducing edge classes. Automate this process saved significantly on time and human annotation effort even thought, the masks were still not perfect and definitely lack on refinement. It is save to say, that without the mask correction, neither U-Net nor StarDist would be able to learn from the data. Indisputably, this step offers room for improvement, which will also influence the subsequent performance and downstream analysis. As 3D segmentation is an active field of research, this learning can be usefeull for future model refinement and creation.

4.0.2 StarDist Training

Training the StarDist model followed the objective of sourcing the best performance out of this benchmark model. The model specific parameter with the most theoretical impact are the backbone architecture and the n-rays that were used to predict the polyhedral shape of the cells. For our purpose, 96 rays conferment to be enough to train well, as Table 3 supports. There was no noticeable effect of the backbone architecture on the training performance either, Resnet and U-Net were suitable equally with a slight advantage for the Resnet architecture. The most

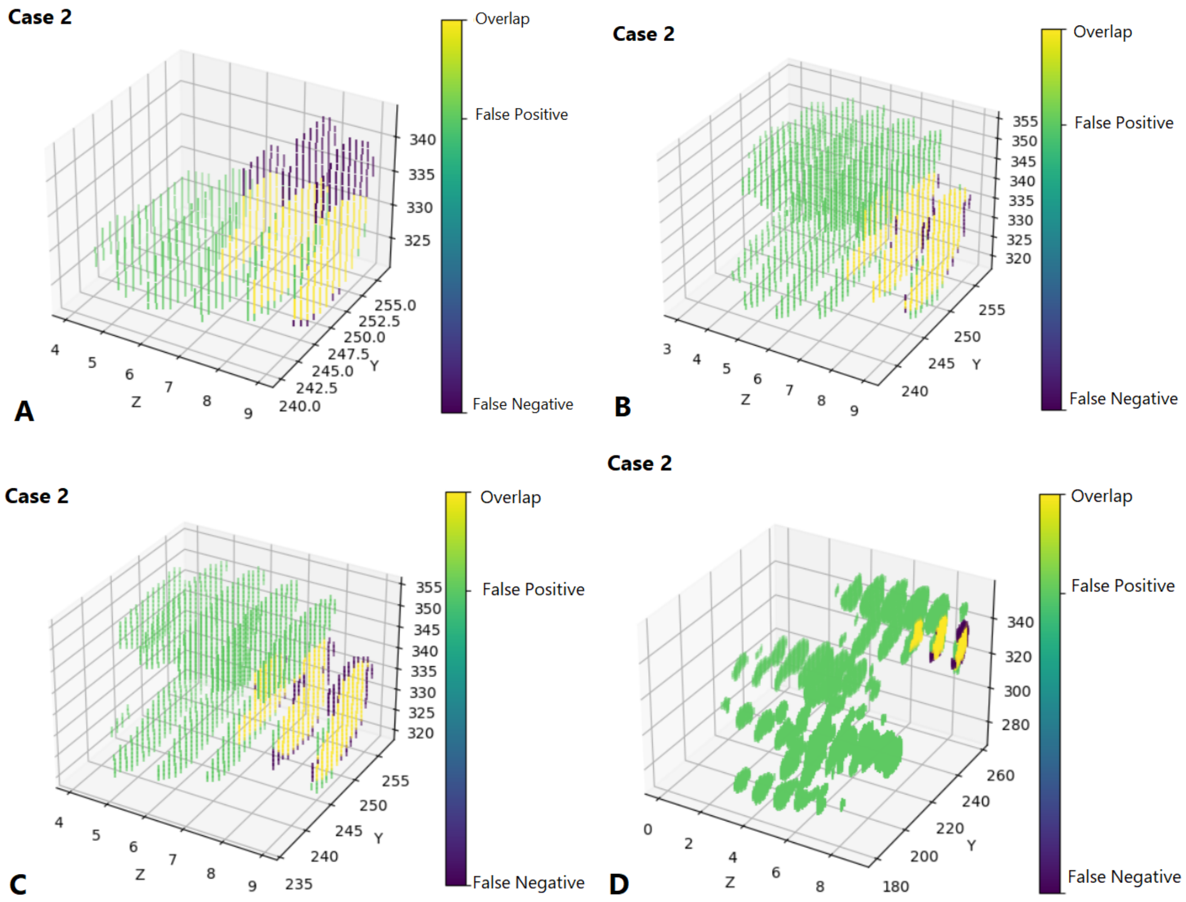

Figure 12: **Visualization of touching borders in GFP.** This figure shows the complete image segmentation of gt nuclei label 18, which touches two other nuclei in 3D space at the sides and bottom. The separation ability of the best to worst model, i.e. from **A** to **D** are demonstrated. In **A**, StarDistResnet achieves the best **full image** DiceScore and thus the best separation with **0.496**. **B** displays amber glitters performance with **0.337**, closely followed by comic firefly **(C)** with **0.322**. The worst segmentation, in line with the evaluation score, was achieved by glowing dawn **(D)** with **0.1052**. Only StarDist **(A)**was able to separate the neighboring cores to z-depth. In **Case 2**, false positive regions are coloured in **green** and false negative segmentations in **purple**.

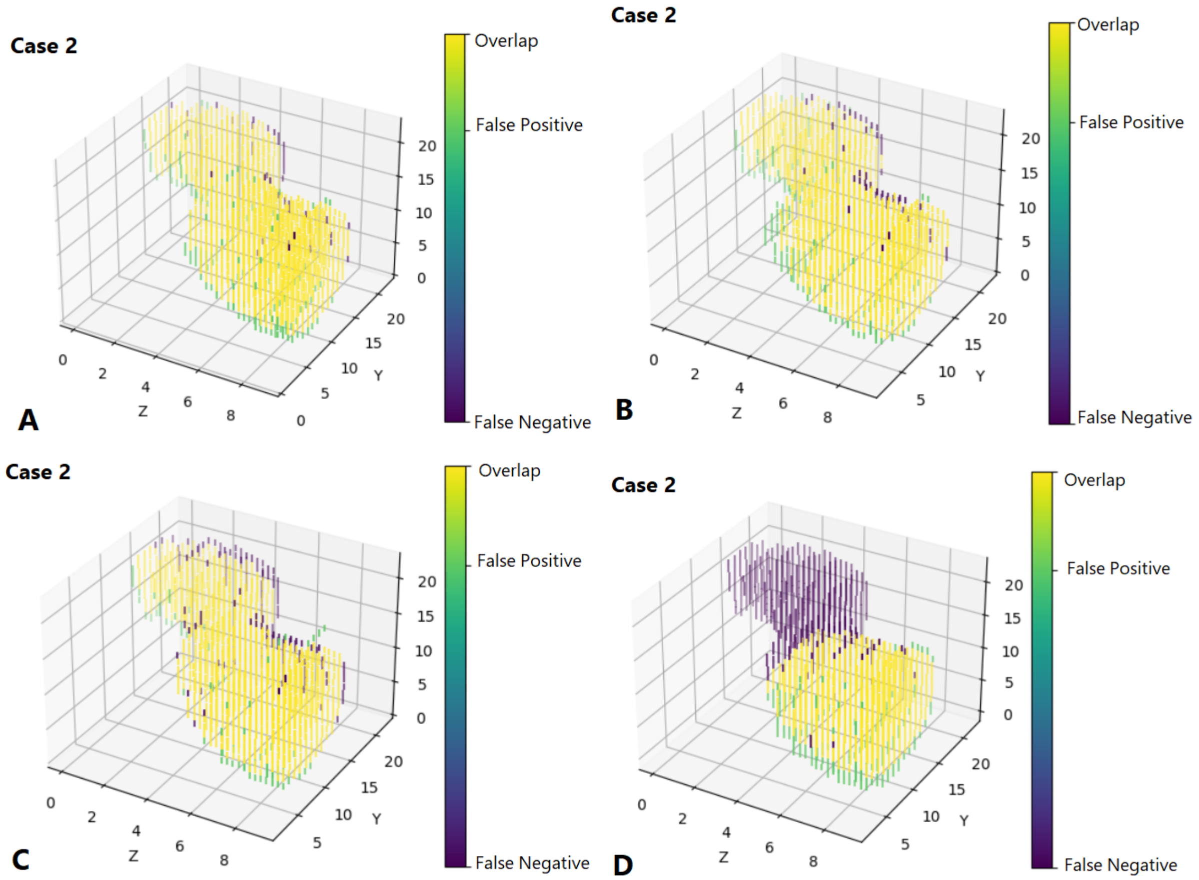

Figure 13: **Segmentation results for a representative Hoechst nucleus (label 47).** This figure presents Dice Scores for **sub-array** segmentation, demonstrating the models' ability to accurately delineate nuclear regions. Fine Yogurt **(A)** achieved the highest sub-array Dice Score of **0.944**, followed by amber glitter **(B** (**0.9332**) and comic firefly **(C)** with **0.9307**. In contrast, StarDist Resnet **(D)** yielded a lower Dice Score of **0.6953**. Notably, this example highlights a challenging segmentation region where StarDist Resnet **(D)** failed to fully encompass the nuclear stain. Interestingly, comic firefly **C**, despite exhibiting the highest overall Hoechst performance, did not surpass the performance of Fine Yogurt **(A)** or amber glitter **(B)** in this specific instance. In **Case 2**, false positive regions are coloured in **green** and false negative segmentations in **purple**.

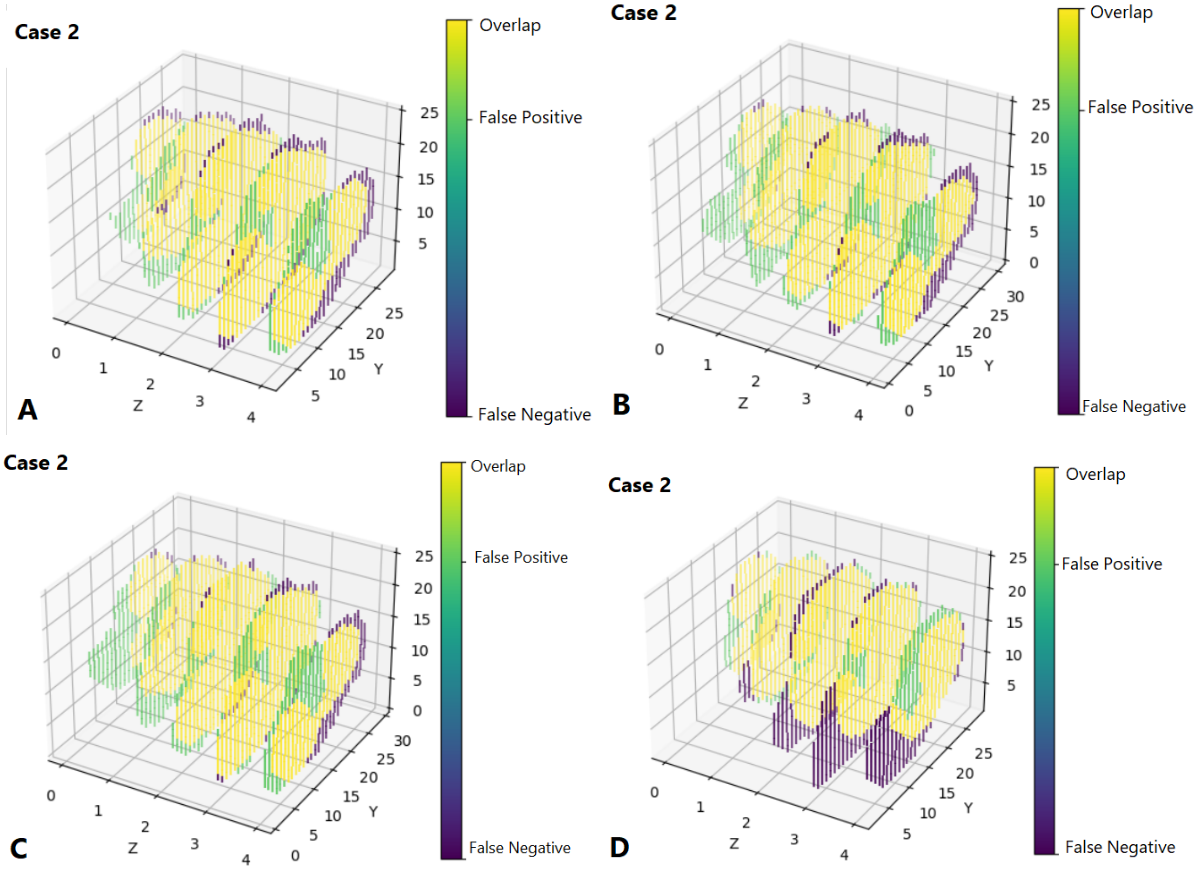

Figure 14: **Subarray segmentation of challenging nuclei shape in GFP.** The specific gt nuclei (label 20 GFP) exhibits a concave structure illustrating a challenge for segmentation. None of the displayed models is able to fully capture the shape. The **sub array** score decreases from **A-D**. Comic firefly **(A)** yielding **0.8495** followed by amber glitter reaching **(B) 0.8377**. Fine yogurt **(C)** on the bottom left achieves a score of **0.8368** followed by StarDistResnet **(D)** yielding the lowest DiceScore of **0.8224**. In alignment with the final performance score, comic Firefly **(A)** beats other U-Net configuration but also the StarDist models. The concave part of the nuclei was entirely filled with FP predictions by all models. In **Case 2**, false positive regions are coloured in **green** and false negative segmentations in **purple**.

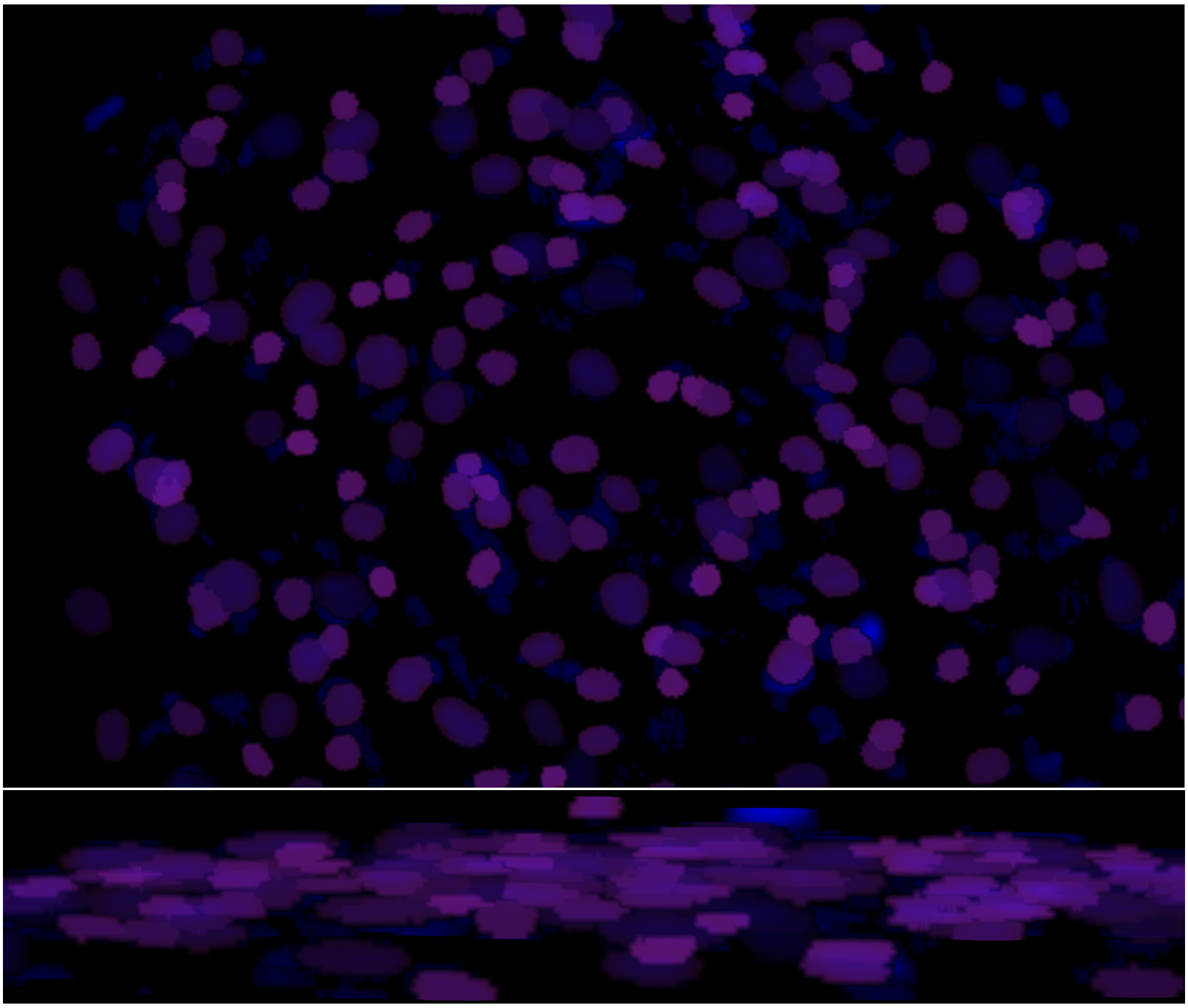

Figure 15: **Complete 3D rendering of best segmentation model.** StarDistResnet segmentation masks are shown in violet and as overlay to the Cy3 channel evaluation image in blue. Above the XY top down view, below that the z-axis view.

significant difference was observed, when minimizing the image input shape to $(4, 128, 128)$, reducing the training time to 6min instead of 1.5-2h. The metrics, indicating the learning success of the model were also improved by this reduction and even more improved when extending the training to one thousand epochs. Nevertheless, the training metrics are not reflecting the actual segmentation performance but the ability to pic up the features out of the training images. For that reason, IoU loss used as most demonstrative segmentation performance indicator. Here, small-patch and small patch-1k were yielding the lowest scores confirming the problem of overfitting for those models. The model was learning patterns only present in the training set but failing to recognize nuclei of unseen images. This is also supported by a unstable learning and IoU curve in Figure 3. Another reason could be the poor quality of the training image dataset. Training very successfully on poor segmentation masks forces the model to learn the poor segmentation as gold standard and can therefore not re adapt when tested against the manual ground truth images. This would also explain the poor final evaluation score of the starDist small model in Table 7, where Resnet and U-Net scored higher while yielding lower training metrics. Another reason could be the that the small input size was not capturing the full complexity towards all dimensions but only scoring high on a tiny section of the image. Literature implies the advantage of StarDist and U-Net handling big input data [14, 29, 31]. These findings suggest training StarDist with low input size on perfect manually labelled ground truth data.

Still, StarDists ability to train significantly faster with comparable performance is a great benefit of this model and confirms its complex learning and approximation algorithms. Also, thanks to the provided notebooks the data handling, model training and prediction was straightforward to apply and execute, assuming a suitable GPU is available.

4.0.3 Final Performance Validation

Investigating the results of Table 7, difference in segmentation performance per channel become obvious. The poor score of approximately 0.5 for the Hoechst channel is due to the cluttered, dispersed nature of the fluorescent signal caused by unsuccessful staining in the experiment. Furthermore, the score involves a crucial bottleneck, declining the *full Score* of every U-Net model. U-Net outputs a binary segmentation mask, edges are predicted to separate clustered nuclei. For accurate segmentation and evaluation, every nuclei needs to be labeled with an identifier, which is called instance segmentation. In the 3D space, this represents an even harder challenge and field of research. For this work, we used the connected component function of the cc3d packages, which determines where a nuclei ends and another starts [58] If only at one point, the edge prediction failed and two nuclei touch each other, the function considered them as one instance. As the Figures 10, 9 demonstrate, this leads to huge merged instances that are punished by the evaluation score (Eq. 9, even though within the subregion, the segmentation was successful. The clustered nature of the Hoechst staining suffers the most out of the three channels under these circumstances. This could have been improved with better segmentation performance of the model or with an additional post-processing of the segmentation mask. This can be seen in Figure 12, where only glowing dawn connects all neighboring nuclei resulting in a significantly lower score. Finally, more accurate and precise ground truth images would mitigate the effects of this bottleneck.

Unexpectedly, the model with the best final evaluation score differs from the best training F1 score. The observed high performing U-Net models (amberGlitter, drawn firebrand, Fine yogurt) do not completely correlate with the evaluation score of Table 7. The ”FineYogurt” run as an example exhibits an average good performance and is beaten by run ”comic Firefly”. This could have the following two reasons. All experiments were trained on the same images, and evaluated on 2 evaluation images, excluded from the training set. Since only GFP and Cy3 were chosen for evaluation, the poor performance on the Hoechst channel of ”fine Yogurt” was unnoticed and revealed in the final evaluation only. Second, this study employed effortful manually annotated images for the final score evaluation, as mentioned in methods. Here, the accuracy of manual segmentation excels the Cellpose generated ground truth images and is therefore challenging different capabilities of the models. To maintain a fair comparison, the images for F1-scores and final evaluation had to differ. Certainly, the score is influenced by the connected component bottleneck as well. Even though the F1-score for edge segmentation were demonstrative, the final scores suggest that the model was not able to sufficiently separate clustered nuclei. The effectiveness of the edge class as separator is evidenced by the poor F1-edge performance and resulting low final score of ”drawn firebrand” and therefore consistent with the literature [14].

The crucial role of the edge class as a segmentation boundary is evident in Figure 10, While Comic Firefly achieved the highest overall evaluation score, Drawn Firebrand excelled in cell F1-score. Visual inspection underscores the importance of the edge class while also highlighting the potential of Focal Loss. Although Focal Loss demonstrated the ability to learn intricate feature representations, it struggled with underrepresented classes such as edges. Figure 11 further emphasizes the nuanced differences in detailed shape segmentation. Again, Focal Loss in Drawn Firebrand showcased potential for capturing fine details, but weighted CE consistently outperformed it. This is supported by Table 7, where only edge F1-scores varied significantly, while both models achieved comparable high cell F1-scores. This suggests the effectiveness of weighted CE in addressing class imbalance when appropriate weights are applied.

The challenges of 3D segmentation in densely packed regions are shown in Figure 12, which presents the best achieved score of only 0.496, reflecting the complexity of 3D segmentation. Despite these difficulties, the models demonstrated the capacity to identify and separate specific nuclei within highly clustered areas.

Figure 14 illustrates the limitations of current methods in accurately capturing non-convex nuclear shapes. Even the top-performing model struggled to delineate the concave region of the selected nucleus. The underperformance of StarDist, previously the best model, can be attributed to its underlying assumption of polyhedral nuclear shapes. Surprisingly, Fine Yogurt with IoU loss achieved comparable results to weighted CE, suggesting alternative loss functions may offer advantages in specific scenarios.

4.0.4 Potential Limitations and Human in the Loop

A comprehensive review of the literature and our experimental findings highlighted the pivotal role of human intervention in achieving accurate 3D segmentation performance. Recent research focused on reducing annotation burden with novel strategies rather than developing advanced network architectures or algorithms [32, 60]. The integration of expert corrected segmentations aimed to promote the learning process of CNN models and create a synergistic relationship between lowered human expertise and accelerated machine learning [59, 15, 14] Unfortunately, our current study was limited in this aspect. To address this, the workflow of this study was structured to incorporate human in the loop refinement as described by Vijayan et al. to maximize learning from image data. [15]. Figure 2 visually outlines the human in the loop component, highlighted by the pink box, which includes iterative steps accessible to human correction. Successive model retraining benefits from the improved ground truth, while human correction effort is eased through reduced segmentation errors and an increasingly capable model. Future work is able to apply this strategy to the existing models of this work to optimize overall performance.

5 Limitations

Despite achieving promising results, the lack of qualitative and quantitative comprehensive ground truth annotations limited the full realization of this study's potential. The search for tools and programs to handle the specific type of 3D data was tedious and time consuming. While specialized tools like Labkit were available, initial efforts focused on less specialized options such as CellPose-GUI. Other comparable software packages, such as napari and its plugins, presented slow learning curves due to complex user interfaces [61]. Managing, structuring and processing large volumes of 3D TIF images was labor-intensive and tougher than working with 2D images. Furthermore, establishing the appropriate infrastructure for computational work was more challenging than expected. DL models for 3D image analysis require substantial GPU resources, often exceeding 8GB and reaching up to 29GB of GPU RAM. Initial limitations in GPU access, restricted to a laptop with a 2GB GPU hindered the model development. GPU constraints will always represent a limiting factor for research processing large 3D image datasets. In addition, the microscopy parameters that determine the quality of image acquisition affect the performance of the model. Optimal image acquisition requires high resolution, minimal signal to noisy ratio, limited over expression to prevent nuclei overlap, and robust staining capable of penetrating deep tissue layers, unlike Hoechst stain. Nevertheless, the field of 3D image segmentation in fluorescent microscopy is rapidly evolving, with new methodologies, models, and publications published regularly, providing novel insights into the ongoing challenges in this challenging research area.

6 Future Work

Further research and refinement are beyond scope of this thesis. Future research should focus on generating manual segmentations as part of a profound ground truth training dataset. Labkit and its machine learning tools will serve as aid in these efforts. Given that, further in-depth evaluations of model performances can be conducted with more expressive results. The human in the loop strategy should be implemented as suggested in this study, to minimize the human effort in segmentation and gaining the most performance out of it. Once a dataset, build of high-quality 3D images, representing the channels and classes in appropriate manner is build, a future direction of research could be the exploitation of novel 3D segmentation models and backbones for U-Net implementation. Finally, future work involves the identification of cell cycle states based on the FUCCI signal of the segmented nuclei. Automating this process will enable the cell cycle analysis, crucial for novel drug screening and efficacy assessment.

Another interesting direction is the integration of time series images in the 3D space. Previous tracking of localized nuclei will not only advance accurate segmentation of clustered nuclei, but also enable more sophisticated cell-cycle analysis in real time.

7 Conclusion

In conclusion, this study successfully curated 3D microscopy images, processed them into a training data set and trained, tuned and applied deep learning nuclei segmentation models. The study was not able to perform a cell cycle analysis on breast cancer spheroids. Nevertheless, it proved its strength in providing a profound foundation based on literature research theoretical, considerations as well as identifying useful tools and techniques. The study demonstrated a successful approach to gaining meaningful 3D nuclei segmentation out of a raw, noisy dataset and integrating as little as possible human annotation effort. Concluding results were decent and exemplify the potential of simple network architectures, competing with pretrained complex models.

Overall, this research provides a robust foundation for precise 3D nuclei segmentation, thereby facilitating downstream in-depth analyses of cell cycle progression and contributing to the advancement of cancer drug discovery.

8 Data Availability

9 Thanks To

References

[1] K. P. Trayes and S. E. Cokenakes, “Breast cancer treatment,” *American family physician*, vol. 104, no. 2, pp. 171–178, 2021.

[2] M. Kapałczyńska, T. Kolenda, W. Przybyła, M. Zajaczkowska, A. Teresiak, V. Filas, M. Ibbs, R. Bliźniak, Ł. Łuczewski, and K. Lamperska, “2d and 3d cell cultures–a comparison of different types of cancer cell cultures,” *Archives of medical science*, vol. 14, no. 4, pp. 910–919, 2018.

[3] C. Jensen and Y. Teng, “Is it time to start transitioning from 2d to 3d cell culture?” *Frontiers in molecular biosciences*, vol. 7, p. 33, 2020.

[4] S. Yano, H. Tazawa, S. Kagawa, T. Fujiwara, and R. M. Hoffman, “FUCCI Real-Time Cell-Cycle Imaging as a Guide for Designing Improved Cancer Therapy: A Review of Innovative Strategies to Target Quiescent Chemo-Resistant Cancer Cells,” *Cancers*, vol. 12, no. 9, p. 2655, 9 2020. [Online]. Available: https://www.mdpi.com/2072-6694/12/9/2655

[5] N. Zielke and B. A. Edgar, “FUCCI sensors: powerful new tools for analysis of cell proliferation,” *Wiley interdisciplinary reviews. Developmental biology*, vol. 4, no. 5, pp. 469–487, 4 2015. [Online]. Available: https://doi.org/10.1002/wdev.189

[6] X. Galindo, T. Barry, P. Guyot, C. Riviere, R. Galland, and F. Levet, “3d nuclei segmentation by combining gan based image synthesis and existing 3d manual annotations,” *bioRxiv*, pp. 2023–12, 2023.

[7] J. Ker, L. Wang, J. Rao, and T. Lim, “Deep learning applications in medical image analysis,” *Ieee Access*, vol. 6, pp. 9375–9389, 2017.

[8] A. Wang, Q. Zhang, Y. Han, S. Megason, S. Hormoz, K. R. Mosaliganti, J. C. Lam, and V. O. Li, “A novel deep learning-based 3d cell segmentation framework for future image-based disease detection,” *Scientific reports*, vol. 12, no. 1, p. 342, 2022.

[9] M. Marzec, A. Piórkowski, and A. Gertych, “Efficient automatic 3D segmentation of cell nuclei for high-content screening,” *BMC Bioinformatics*, vol. 23, no. 1, 5 2022. [Online]. Available: https://doi.org/10.1186/s12859-022-04737-4

[10] L. Fischer and I. Thievessen, “FUCCI Reporter Gene-Based Cell Cycle Analysis,” *Methods in molecular biology*, pp. 371–385, 1 2023. [Online]. Available: https://doi.org/10.1007/978-1-0716-3052-5_24

[11] N. F. Greenwald, G. Miller, E. Moen, A. Kong, A. Kagel, T. Dougherty, C. C. Fullaway, B. J. McIntosh, K. X. Leow, M. S. Schwartz, C. Pavelchek, S. Cui, I. Camplisson, O. Bar-Tal, J. Singh, M. Fong, G. Chaudhry, Z. Abraham, J. Moseley, S. Warshawsky, E. Soon, S. Greenbaum, T. Risom, T. Hollmann, S. C. Bendall, L. Keren, W. Graf, M. Angelo, and D. Van Valen, “Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning,” *Nature Biotechnology*, vol. 40, no. 4, pp. 555–565, 11 2021. [Online]. Available: https://doi.org/10.1038/s41587-021-01094-0

[12] M. Y. Lee, J. S. Bedia, S. S. Bhate, G. L. Barlow, D. Phillips, W. J. Fantl, G. P. Nolan, and C. M. Schürch, “CellSeg: a robust, pre-trained nucleus segmentation and pixel quantification software for highly multiplexed fluorescence images,” *BMC Bioinformatics*, vol. 23, no. 1, 1 2022. [Online]. Available: https://doi.org/10.1186/s12859-022-04570-9

[13] R. Wagner and K. Rohr, "Efficientcellseg: efficient volumetric cell segmentation using context aware pseudocoloring," in *International Conference on Medical Imaging with Deep Learning.* PMLR, 2022, pp. 1311–1321.

[14] M. Weigert, U. Schmidt, R. Haase, K. Sugawara, and G. Myers, "Star-convex polyhedra for 3d object detection and segmentation in microscopy," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 3666–3673.

[15] A. Vijayan, T. A. Mody, Q. Yu, A. Wolny, L. Cerrone, S. Strauss, M. Tsiantis, R. S. Smith, F. A. Hamprecht, A. Kreshuk, and K. Schneitz, "A deep learning-based toolkit for 3D nuclei segmentation and quantitative analysis in cellular and tissue context," *Development*, vol. 151, no. 14, 7 2024. [Online]. Available: https://doi.org/10.1242/dev.202800

[16] M. Marzec, A. Piórkowski, and A. Gertych, "Efficient automatic 3d segmentation of cell nuclei for high-content screening," *BMC bioinformatics*, vol. 23, no. 1, p. 203, 2022.

[17] A. Sakaue-Sawano, H. Kurokawa, T. Morimura, A. Hanyu, H. Hama, H. Osawa, S. Kashiwagi, K. Fukami, T. Miyata, H. Miyoshi, T. Imamura, M. Ogawa, H. Masai, and A. Miyawaki, "Visualizing Spatiotemporal Dynamics of Multicellular Cell-Cycle Progression," *Cell*, vol. 132, no. 3, pp. 487–498, 2 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0092867408000548

[18] D. Nelson and M. Cox, *Lehninger Biochemie*, 4th ed. Springer, 12 2010.

[19] A. M. Singh, R. Trost, B. Boward, and S. Dalton, "Utilizing fucci reporters to understand pluripotent stem cell biology," *Methods*, vol. 101, pp. 4–10, 2016.

[20] T. F. Scientific, "DAPI and Hoechst Nucleic Acid Stains," 05 2024. [Online]. Available: https://www.thermofisher.com/order/catalog/product/de/de/62249

[21] M. M. Usaj, E. B. Styles, A. J. Verster, H. Friesen, C. Boone, and B. J. Andrews, "High-content screening for quantitative cell biology," *Trends in cell biology*, vol. 26, no. 8, pp. 598–611, 2016.

[22] F. Hörst, M. Rempe, L. Heine, C. Seibold, J. Keyl, G. Baldini, S. Ugurel, J. Siveke, B. Grünwald, J. Egger *et al.*, "Cellvit: Vision transformers for precise cell segmentation and classification," *Medical Image Analysis*, vol. 94, p. 103143, 2024.

[23] M. Kim, Y. Namkung, D. Hyun, and S. Hong, "Prediction of stem cell state using cell image-based deep learning," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2300017, 2023.

[24] Y. Lan, N. Huang, Y. Fu, K. Liu, H. Zhang, Y. Li, and S. Yang, "Morphology-Based Deep Learning Approach for Predicting Osteogenic Differentiation," *Frontiers in bioengineering and biotechnology*, vol. 9, 1 2022. [Online]. Available: https://doi.org/10.3389/fbioe.2021.802794

[25] J. C. Caicedo, J. Roth, A. Goodman, T. Becker, K. W. Karhohs, M. Broisin, C. Molnar, C. McQuin, S. Singh, F. J. Theis *et al.*, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images," *Cytometry Part A*, vol. 95, no. 9, pp. 952–965, 2019.

[26] L. Lu, Y. Zheng, G. Carneiro, and L. Yang, "Deep learning and convolutional neural networks for medical image computing," *Advances in computer vision and pattern recognition*, vol. 10, pp. 978–3, 2017.

[27] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 1 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28

[28] A. Wolny, L. Cerrone, A. Vijayan, R. Tofanelli, A. V. Barro, M. Louveaux, C. Wenzl, S. Strauss, D. Wilson-Sánchez, R. Lymbouridou *et al.*, "Accurate and versatile 3d segmentation of plant tissues at cellular resolution," *Elife*, vol. 9, p. e57613, 2020.

[29] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, "Cell detection with star-convex polygons," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11.* Springer, 2018, pp. 265–273.

[30] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *Nature methods*, vol. 18, no. 1, pp. 100–106, 12 2020. [Online]. Available: https://doi.org/10.1038/s41592-020-01018-x

[31] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19.* Springer, 2016, pp. 424–432.

[32] V. V. Thiyagarajan, A. Sheridan, K. M. Harris, and U. Manor, "A deep learning-based strategy for producing dense 3d segmentations from sparsely annotated 2d images," *bioRxiv*, pp. 2024–06, 2024.

[33] M. S. Hossain, J. M. Betts, and A. P. Paplinski, "Dual Focal Loss to address class imbalance in semantic segmentation," *Neurocomputing*, vol. 462, pp. 69–87, 10 2021. [Online]. Available: https://doi.org/10.1016/j.neucom.2021.07.055

[34] Z. Li, K. Kamnitsas, and B. Glocker, "Analyzing overfitting under class imbalance in neural networks for image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 3, pp. 1065–1077, 2021.

[35] M. Lai, "Deep learning for medical image segmentation," *arXiv preprint arXiv:1505.02000*, 2015.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 318–327, 2 2020. [Online]. Available: https://doi.org/10.1109/tpami.2018.2858826

[37] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, 1 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-67558-9_28

[38] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[39] K. Hu, Z. Zhang, X. Niu, Y. Zhang, C. Cao, F. Xiao, and X. Gao, "Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function," *Neurocomputing*, vol. 309, pp. 179–191, 10 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.05.011

[40] S. Iqbal, M. U. Ghani, T. Saba, and A. Rehman, "Brain tumor segmentation in multi-spectral mri using convolutional neural networks (cnn)," *Microscopy research and technique*, vol. 81, no. 4, pp. 419–427, 2018.

[41] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," *arXiv preprint arXiv:1412.1441*, 2014.

[42] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4204–4212, 2019.

[43] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran, "Accelerating human-in-the-loop machine learning: Challenges and opportunities," in *Proceedings of the second workshop on data management for end-to-end machine learning*, 2018, pp. 1–4.

[44] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A survey of human-in-the-loop for machine learning," *Future Generation Computer Systems*, vol. 135, pp. 364–381, 2022.

[45] L. University and LUMC.

[46] A. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[47] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[48] N. Europe, "NIS-Elements — NIS-Elements Viewer," 06 2024. [Online]. Available: https://www.microscope.healthcare.nikon.com/en_EU/products/software/nis-elements/viewer

[49] D. Eschweiler, T. V. Spina, R. C. Choudhury, E. Meyerowitz, A. Cunha, and J. Stegmaier, "Cnn-based preprocessing to optimize watershed-based cell segmentation in 3d confocal microscopy images," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, vol. 1, 2019, pp. 223–227.

[50] OpenCV, "OpenCV: Changing the contrast and brightness of an image!" [Online]. Available: https://docs.opencv.org/4.x/d3/dc1/tutorial_basic_linear_transform.html

[51] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, Inc., 1994, pp. 474–485.

[52] OpenCV, "OpenCV: cv::CLAHE Class Reference." [Online]. Available: https://docs.opencv.org/4.x/d6/db6/classcv_1_1CLAHE.html

[53] opencv, "OpenCV: Morphological Transformations." [Online]. Available: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

[54] A. H. Murphy, "The Finley Affair: A Signal Event in the History of Forecast Verification," *Weather and forecasting*, vol. 11, no. 1, pp. 3–20, 3 1996. [Online]. Available: https://doi.org/10.1175/1520-0434(1996)011⟨0003:tfaase⟩2.0.co;2

[55] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System technical journal*, vol. 27, no. 3, pp. 379–423, 7 1948. [Online]. Available: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[56] M. Arzt, J. Deschamps, C. Schmied, T. Pietzsch, D. Schmidt, P. Tomancak, R. Haase, and F. Jug, “LABKIT: Labeling and Segmentation Toolkit for Big Image Data,” *Frontiers in computer science*, vol. 4, 2 2022. [Online]. Available: https://doi.org/10.3389/fcomp.2022.777728

[57] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: an open-source platform for biological-image analysis,” *Nature methods*, vol. 9, no. 7, pp. 676–682, 6 2012. [Online]. Available: https://doi.org/10.1038/nmeth.2019

[58] W. Silversmith, “cc3d: Connected components on multilabel 3D & 2D images.” Sep. 2021.

[59] M. Pachitariu and C. Stringer, “Cellpose 2.0: how to train your own model,” *Nature methods*, vol. 19, no. 12, pp. 1634–1641, 2022.

[60] J. Bragantini, M. Lange, and L. Royer, “Large-scale multi-hypotheses cell tracking using ultrametric contours maps,” *arXiv preprint arXiv:2308.04526*, 2023.

[61] C.-L. Chiu and N. Clack, “napari: a Python Multi-Dimensional Image Viewer Platform for the Research Community,” *Microscopy and Microanalysis*, vol. 28, no. S1, pp. 1576–1577, 8 2022. [Online]. Available: https://doi.org/10.1017/s1431927622006328