UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Accurate Whole Cell Instance Segmentation from Brightfield Images

Author:
Yaroslav PRYTULA

Supervisor:
Dr. Dmytro FISHMAN

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences and Information Technologies
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Yaroslav PRYTULA, declare that this thesis titled, "Accurate Whole Cell Instance Segmentation from Brightfield Images" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Science is not only a disciple of reason but also one of romance and passion."

Stephen Hawking

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Accurate Whole Cell Instance Segmentation from Brightfield Images

by Yaroslav PRYTULA

Abstract

Semantic and instance segmentations have revolutionized biomedical image analysis, playing a crucial role in numerous biological applications. The development of accurate segmentation pipelines has enabled fast and reliable image analysis. Previous state-of-the-art methods in cellular biology rely on accurate cell segmentation without preserving knowledge of overlapping instances. In this work, we first show that extending the model by introducing multiple decoupled decoders for multi-task learning greatly helps in scene understanding and results in high-fidelity segmentations. Furthermore, we identify cases of overlap occurrence and construct probability maps based on cell spatial proximity. Additionally, to overcome the lack of annotated samples, we introduce a way to synthesize brightfield images and show that applying overlap-aware weight maps directly to the loss function guides the model to attend to regions of occluded cells, thus improving segmentation performance. We then propose an approach to extend our model to perform instance segmentation. Compared to previous state-of-the-art approaches, we utilize a conceptually novel method of learning instance activation maps that highlight informative regions for different cells for global awareness. Without bells and whistles, we combine multi-task learning with overlap awareness for instance segmentation, and show that our approach achieves state-of-the-art results.

Acknowledgements

I would like to express my appreciation and gratitude to Dmytro Fishman, who supervised and guided me throughout the whole project. I extend my thanks to the Biomedical Computer Vision Lab at the University of Tartu for their invaluable support and opportunities for engaging discussions and talks.

I am deeply grateful to the Machine Learning Laboratory at the Ukrainian Catholic University for providing access to expand my knowledge, learn more, and grow in the field of studies. Big thank you to Oles Dobosevych, Maria Dobko, Yurii Yelisieiev, Ostap Viniavskyi, Markiian Novosad, Ruslan Partsey, and Oleh Smolkin for their assistance, motivation, and friendly discussions. I am also grateful to all the tutors, including Stepan Fedynyak, Rostyslav Hryniv, Oleksii Molchanovskyi, Oleg Farenyuk, and others, for their unwavering support and openness to discussions.

Furthermore, I wish to express my immense gratitude to my family and friends who have supported and motivated me throughout my academic journey, encouraging me to strive for greater achievements.

Contents

List of Figures

List of Tables

List of Abbreviations

FCN	**F**ully **C**onvolutional **N**etwork
GAN	**G**enerative **A**dversarial **N**etwork
cGAN	**c**onditional **G**enerative **A**dversarial **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
FID	**F**réchet **I**nception **D**istance
IAM	**I**nstance **A**ctivation **M**aps
SE	**S**queeze (and) **E**xcitation
mAP	**m**ean **A**verage **P**recision
BF	**B**rightfield
PC	**P**hase-**C**ontrast

Dedicated to the Armed Forces of Ukraine and all brave people defending Ukraine for providing safety and security to enable education, science, and business to move forward.

Chapter 1

Introduction

Studying biological systems at the cellular and whole sample levels is critical to advancing our understanding of complex biological processes. Cell-level studies focusing on individual cells can reveal quantitative details about numerous cellular properties such as shape and position and space, signaling pathways, RNA and protein expressions [5] [4]. In contrast, whole-sample studies, which examine the collective behavior of cells within a tissue or organism, can provide insights into higher-level processes such as tissue development. Combining these approaches allows researchers to gain a more comprehensive understanding of biological systems and to develop effective treatments for diseases such as cancer, Alzheimer's, and cardiovascular disease [47]. In recent years, studies of cancer at the cellular level have led to the development of targeted therapies that kill cancer cells leaving healthy cells intact [16] [43] [57]. Meanwhile, whole-sample studies have revealed important insights into the behavior of tumors in their native environments, paving the way for new approaches to cancer treatment. Ultimately, advances in cell-level and whole-sample studies are critical to developing personalized medicine and improving the health and well-being of individuals worldwide.

As we delve into the exciting field of biomedical imaging, the increasing availability of large datasets and the growing power of computer hardware have paved the way for the rise of deep learning models for image segmentation. They are a crucial step in analyzing medical images, enabling researchers to identify and quantify different structures and features within the images. Convolutional neural networks (CNNs) have shown to be particularly effective for segmentation tasks due to their ability to learn complex and hierarchical representations of images.

Applying deep learning models to biomedical imaging has yielded impressive results, achieving state-of-the-art performance on benchmark datasets and outperforming traditional machine learning algorithms and human experts in many cases.

The rise of deep learning models for image segmentation in biomedical imaging represents a notable development in the field, with the potential to significantly enhance our understanding of disease processes and contribute to developing more effective treatments. As we continue to explore this exciting field, deep learning models for image segmentation will likely play an increasingly important role in advancing our knowledge of the human body and its complex systems.

The use of deep learning models for segmentation has emerged as an important and vibrant area of research. In tissue analysis, cell segmentation refers to the process of identifying and separating individual cells within an image, which is a crucial step in analyzing microscopic samples and studying the properties and behavior of cells. Quantitative cell biology requires measurements of different cellular properties, such as the position and shape of individual cells. To achieve this, one must first segment the image volume into cell bodies based on cytoplasmic markers.

Semantic segmentation plays a crucial role in cell segmentation as it involves labeling every pixel in an image with a corresponding class, such as the nucleus, cytoplasm, or background. This approach allows researchers to identify and quantify different regions of interest within the image, providing insights into the structure and function of cells in one sample. Nonetheless, despite its effectiveness, this type of segmentation faces a challenge when it comes to distinguishing between multiple distinctive objects belonging to the same class. Thus, making it impossible to tell them apart without additional postprocessing.

While semantic segmentation is one approach to cell segmentation, more is needed to study cell interactions on the instance level. In contrast, instance segmentation involves labeling every pixel with a class and distinguishing between individual instances of that class. In the context of cell segmentation, instance segmentation is particularly important for studying cell behavior and interactions on a microscopic level. This paradigm is essential because cells can exhibit different properties and behaviors within the same tissue or organism. Analyzing them at the object level can reveal important insights into biological processes.

Using deep learning models for cell segmentation is crucial as they allow researchers to analyze microscopic samples at the level of individual cells, providing insights into complex cellular processes that are not easily achievable through classical approaches.

Nonetheless, instance segmentation is challenging when working with images of different modalities. In the realm of biomedical imaging, samples are often very heterogeneous, each presenting unique challenges and opportunities for analysis. One of the factors that can vary significantly between samples is the object count of structures present in the image. Additionally, samples may have varying levels of cell proximity or the degree to which cells cluster together in the image. It is usual in biological images to encounter multiple overlapping instances; the cells can often lose contact inhibition and therefore stack up on top of each other, making it difficult to distinguish between them.

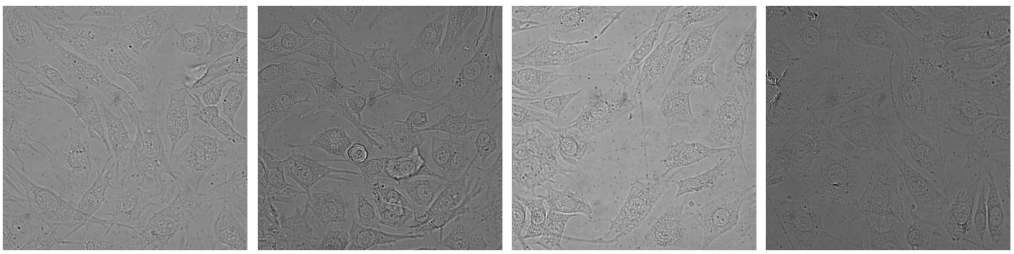


FIGURE 1.1: Examples of brightfield modality focal planes.

Due to the ease of acquisition and versatility, brightfield microscopy has become a valuable tool for cell imaging [62]. Brightfield domain is a more straightforward type of cell imaging [44]. The process involves shining a beam of light through a sample and observing the resulting image. Brightfield imaging can be done with standard laboratory equipment, such as a light microscope, and does not require specialized sample preparation or labeling. Additionally, brightfield imaging can capture living cell images in real time, allowing for dynamic observations of cellular processes. This ease of acquisition and versatility makes brightfield imaging a valuable tool for cell imaging and analysis.

Other types of microscopy, such as phase-contrast, confocal, or electron microscopy, may require more specialized equipment and training and more advanced image processing and analysis techniques.

Brightfield microscopy is widely used in biological research and medical diagnostics [2] [17] [44] [51], as it allows for the visualization of cells and tissues at high resolution without requiring complex equipment or techniques. Despite its popularity, the segmentation of brightfield images has received relatively little attention compared to other imaging modalities, such as fluorescence microscopy. This is mainly because of the inherent complexity of the input samples, which often contain a vast amount of noise and variability, making it challenging to identify and segment individual cells accurately.

Chapter 2

Related works

Semantic segmentation is a vital task in the field of computer vision and serves as a foundation for many downstream applications. Unlike image classification, which produces an image-level prediction, semantic segmentation produces a dense per-pixel category prediction. Accurately segmenting cell structures in images is one of the most encouraging applications of semantic segmentation. However, cell segmentation can be challenging in low contrast and high object density cases and may require sophisticated image processing pipelines.

2.1 Classical Approaches

As we dwell on the history of cell segmentation methods, the classical methods for processing images were the to-go approaches in the beginning. They all acquired preprocessing and image manipulation techniques to perform segmentation further with various hand-crafted algorithms. One of the most widely used techniques is thresholding, which involves binarizing an image based on threshold values. [71] suggests applying a straightforward Otsu intensity thresholding [45] with edge detection. Acquiring more complex cell structures requires a more extensive processing pipeline. [32] proposes to use stacked morphological operations traversing from Gaussian filtering [13] and Canny edge detection [8] and ending with hole filling and region closing postprocessing steps. Nevertheless, many classical approaches of such category are sensitive to natural noise. It becomes extensively hard to renormalize or preprocess images to fit the gaps of those edge problems. Moreover, it is a burden for the algorithm to differentiate foreground and background pixels in an environment with low contrast or uneven illumination.

While classical methods for cell segmentation have been extensively studied, their performance is often limited by the complexity of biological images. In recent years, convolution-based deep-learning methods have emerged as a promising choice for biomedical segmentation, achieving state-of-the-art results on various benchmarks as they can learn compound feature representations from data. Nevertheless, deep learning-based methods often require vast amounts of annotated data to achieve high performance, which can be time-consuming and expensive. Therefore, developing effective methods for segmentation that can operate with limited annotated data is an active area of research in computer vision.

2.2 Semantic Segmentation Using U-Net

While Fully Convolutional Network (FCN) [41] architecture was one of the first widely used neural network architectures for semantic segmentation, it only relied

on an encoder network to downsample the input image and a decoder network to upsample the feature map to obtain the segmentation map.

On the other hand, U-Net [50] is a symmetric encoder-decoder approach, where the encoder and decoder simultaneously learn to capture the image's context and spatial features. The U-Net model comprises encoder-decoder "U-shaped" architecture. It downsamples the convolutional maps several times via a contracting pathway which captures the contextual information, and upsamples in a mirror-symmetric fashion using expanding pathway that facilitates accurate localization. The encoder network consists of several convolutional layers that reduce the input image resolution and extract features from the input image. The decoder network then takes the output from the encoder and produces a segmentation map by upsampling and concatenating feature maps from the encoder. One of the novel features of the U-Net model is its use of skip connections between the encoder and decoder. These skip connections enable the model to recover fine-grained details in the segmentation map that may vanish during the down-sampling process.

The strategy thoroughly relies on solid data augmentation to alleviate and help the model generalize on a small dataset. The U-Net model has been shown to be very effective in handling complex object representations in biomedical image segmentation tasks.

2.3 Instance Segmentation

As a step towards further segmentation acquisition, instance segmentation has surfaced as a powerful technique in computer vision. In recent years, the instance segmentation paradigm has become one of the crucial problems in machine learning. Unlike semantic segmentation, where each pixel is assigned to a unique class type, instance segmentation distinguishes between different instances of the same class. With the rise of deep learning and the availability of large annotated datasets, instance segmentation has become increasingly accurate and widely used in various fields, including medical image analysis, autonomous driving, video analysis, and robotics. With its relevancy and usefulness, many subsequent approaches have been proposed to increase the performance of segmentation algorithms in many domains.

2.3.1 MaskRCNN

Proposal-based methods have emerged as a popular choice for instance segmentation in natural images, where Mask R-CNN [23] is considered a base approach. Mask R-CNN is a baseline representative method that extends Faster R-CNN [48] by adding a mask prediction branch for a robust end-to-end instance segmentation.

2.3.2 YOLOv5

YOLOv5 [31] was introduced in May 2020 as a better successor to the previous v4 generation architecture. It uses a two-stage detector. The backbone of YOLOv5 proposes a Cross Stage Partial Network (CSPNet) [61] and a Spatial Pyramid Pooling network (SPP) [24], which enable dynamic input sizes and robustness against object deformations. The CSPNet strategy enables partitioning the base layer's feature map into two parts and then merging them through a cross-stage hierarchy. In parallel, the SPP block is used to enlarge the receptive field and segregate the most relevant context features. It aggregates the information received from the inputs

and returns a fixed-length output. Thus it has the advantage of significantly increasing the receptive field and segregating the most relevant context features. The head of YOLOv5 uses a feature pyramid network idea via Path Aggregation Network (PANet) [40] for instance segmentation.

2.3.3 YOLOv8

Following the older YOLOv5 version, the new and better YOLOv8 model was introduced recently as a state-of-the-art for segmentation and object detection. As it presents new features and improvements, the model brings gains in COCO [39] Mean Average Precision (mAP) scores compared to its ancestor variants. The recent YOLOv8 is starting to revert to the residual blocks with new modified types of convolutions. The old main building C2f block (CSP Bottleneck) [61] was replaced with a modified C3 version. Following the C2f version, all the outputs from the 3x3 convolutions (Bottlenecks) were concatenated. In contrast, the new C3 block exploits only the final output. The Bottleneck itself adopted 3x3 convolutions replacing the old 1x1.

Regarding the training process, the YOLO family exploits a unique and robust data augmentation scheme. The image transformation comprises a Mosaic Augmentation [21], which is a beneficial part of the model. This approach involves stitching four images together, pushing the model to learn objects in new locations, with different levels of occlusion, and in different environments.

2.3.4 CellPose

CellPose [53] is a recent and promising addition to cell segmentation methods. It offers a general approach by generating topological maps using a simulated diffusion process. The authors train a U-Net [50] architecture neural network to predict the horizontal and vertical gradients of the topological maps, as well as a binary map of cell pixel predictions. The predicted gradients are used to construct a vector field for a prediction, thus, assigning the direction of every foreground pixel to some local sink point, defined as mask center-of-mass. Through gradient tracking [37], all pixels belonging to a given cell can be routed to its center. Thus, the algorithm recovers individual cells by grouping pixels that converge to the same point. Further, the cell shapes are refined by removing redundant pixels that were predicted by the neural network to be outside of cells.

Recently, many approaches have employed single-stage detectors, particularly anchor-free detectors. Unlike traditional bounding-box-based detectors, these approaches represent objects by center pixels and perform segmentation using the center features. This method is favorable as it does not rely on the accuracy of the initial proposals, unlike approaches that segment instances using object detections. The quality of the initial proposals limits such methods and cannot recover from errors made by the initial detector.

Chapter 3

Problem Formulation

Microscopy segmentation from microscopy images presents a significant challenge. The brightfield modality is complex, and cell specimens appear with other problems, such as overlapping instances and highly dense samples.

We focus on solving the problem with overlapping cells. We aim to train the model to produce distinct representations of overlapping regions. In natural conditions, overlapping cells are common. Thus, locating them can be a crucial step toward a more accurate instance segmentation of cells.

Segmentation and object detection have been the two most widespread problems in computer vision over the past many years. Converting semantic segmentation results into instance segmentation is crucial for biomedical studies focusing on individual objects. Proposal-based methods such as Mask R-CNN are popular ways to perform instance segmentation of natural images. Complications arise from differences between the domain of natural and microscopic images. While objects in natural images are typically either vertically or horizontally aligned, objects in microscopy generally acquire complex, non-convex shapes which are randomly oriented. Additionally, these models usually require a pre-trained backbone network and have difficulties segmenting such uniquely shaped objects. Hence, methods that employ axis-aligned bounding boxes perform poorly [18] [34].

In this work, we primarily focus on developing a method for robust cell segmentation on brightfield images. To support our idea of strong instance segmentation, we additionally attend to solving the problem of occluded cells by first explicitly segmenting overlap regions between them.

We start with a U-Net-based model and extrapolate more morphology-based information to enhance the performance of the segmentation network. We scale our model by introducing decoupled decoder branches for more auxiliary outputs. The new multiheaded methodology enables us to construct more accurate and reliable predictions.

Primarily, we use our hand-labeled dataset, described in Section 4.1, to solve the problem of amodal perception. With such a small dataset, we show that handling overlapping regions between the cells is hard. Thus, we additionally exploit the use of a generative adversarial network (GAN) model for generating high-fidelity brightfield planes by incorporating conditional phase-contrast information. After, we perform the domain translation to create a large-scale dataset from our minimally annotated data while focusing on overlap criteria. In parallel, we further suggest an intuitive loss function to guide our model into outputting better overlap region estimates by incorporating information about the shape of the cell and its proximity to the neighboring cell structures.

We additionally put forward an approach to enhance the U-Net-based model to leverage instance-level predictions by comprising a sparse set of instance activation maps that serve as an object representation. We later discuss the ideas of guiding the instance segmentation process by incorporating additional information on overlapping cells, thus creating a better cell reconstruction with conditional knowledge about the cell intersection parts.

In the context of this study, the following contributions of this work are:

- Constructing a small-scale instance segmentation dataset by hand labeling all cell bodies in multiple samples.
- Enhancing the U-Net-based model by incorporating multiple decoupled auxiliary branches for multi-task learning.
- Solving the problem of detecting overlaps between cells in a small dataset by introducing a loss penalty term.
- Creating a method for high-fidelity brightfield image generation and augmentation.
- Exploiting instance segmentation using a U-Net-based model via learning object-wise instance activation maps.

As a part of our research project, we have also received positive approbation from the scientific community at the Institute of Computer Science at the University of Tartu.

Chapter 4

Method

To approach the whole cell segmentation problem in the brightfield domain, we start with semantic segmentation. We modify the baseline model to support decoupled scene-informative predictions (cytoplasm, nuclei, border, background). Next, we adopt the watershed transform [49] for cell separation based on predicted cytoplasm and nucleus. Nonetheless, we obtain overlap-agnostic instance predictions, leading to many incorrect pixel assignments when cells overlap. To further target the problem, we start by predicting the overlap regions. Mainly we focus on leveraging our small dataset with weak overlap annotations by generating a new synthetic dataset of brightfield images using phase contrast information. Later we describe the proposed method to boost the overlap localization significantly. With this in mind, we propose a box-free modified U-Net model to perform efficient instance segmentation.

4.1 Labeling the Dataset

We start by introducing a new dataset produced by the PerkinElmer company. The dataset comprises 11,808 images of 6 different cell lines of brightfield modality. Each image is acquired at x63 magnification level and constitutes a pair of lower and higher brightfield planes. Along with the brightfield images, we had access to the same amount of corresponding phase-contrast and fluorescent images. All the nuclei in the images were semi-automatically pre-segmented using a proprietary classical algorithm. To access accurate instance-level cell annotations, we manually labeled a portion of the dataset. We utilize the LabelStudio software [58] to label every foreground instance separately, as shown in Figure 4.1. Mimicking the approach used in the original U-Net paper, we strive to annotate a rather small portion of the original data to show that our approach can produce high-fidelity results with even a small number of images.

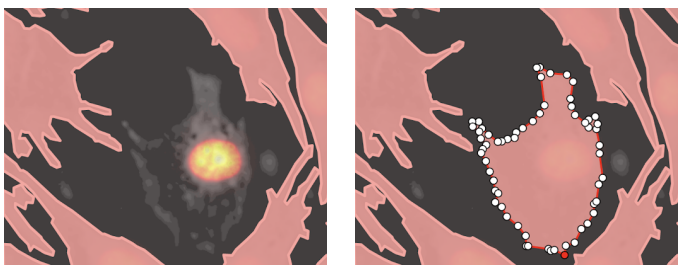


FIGURE 4.1: LabelStudio mask annotations. Every mask describes a polygon with multiple connected node points surrounding a single cell instance.

Naturally, the phase-contrast images have a much higher signal-to-noise ratio, thus, they become the basis for data labeling. We set up our annotation pipeline by expanding a preprocessing strategy for more visual context solely for annotation purposes. First, we throw away samples with a high density of stacked and occluded cells, as they tend to be increasingly hard to annotate. We then utilize an adaptive histogram equalization technique for image level correction. This help to make cells more visible by increasing the brightness and contrast of a sample. Additionally, we attend to the fluorescence data to provide a more feasible visualization of cell instances. We superimpose both phase and fluorescence channels and apply additional coloring to the fluorescence. The resulting visuals (Figure 4.2) provide additional insight into the position and distribution of nuclei within the cells. We find this to be a crucial step in a robust cell annotation pipeline.

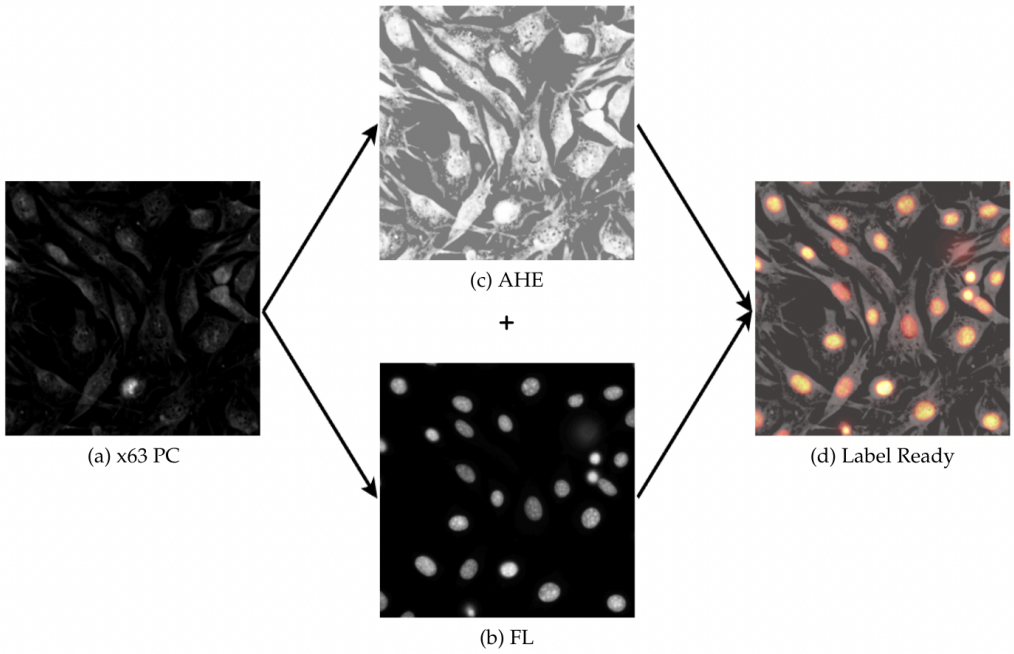


FIGURE 4.2: Process of data preparation for more precise labeling. The labeling pipeline takes a phase-contrast (a) image of x63 magnification level as input. We utilize Adaptive Histogram Equalization (AHE) and Fluorescent images (FL) to introduce more certainty of cytoplasm location.

With the proposed annotation approach, we acquire a set of 28 labeled bright-field images of size 1080x1080 consisting of individually labeled cell instances. Later on, we mainly exploit this dataset for various experiments and attend to it as "exhaustive".

4.2 Semantic Segmentation

Since the original U-Net [50] model was trained on a similar-sized dataset, we leverage this type of network and enhance it by incorporating the Squeeze-and-Excitation blocks (SEU-Net) as a baseline for our segmentation experiments. We have found from empirical evaluations that the U-Net model, when enhanced with Squeeze-and-Excitation (SE) blocks [26], exhibits promising potential for semantic

segmentation of cells in the brightfield domain. This model has shown a state-of-the-art performance on datasets that the research group from the University of Tartu has worked on, surpassing the traditional U-Net and other variations such as Pyramid Pooling (PPUnet) [3].

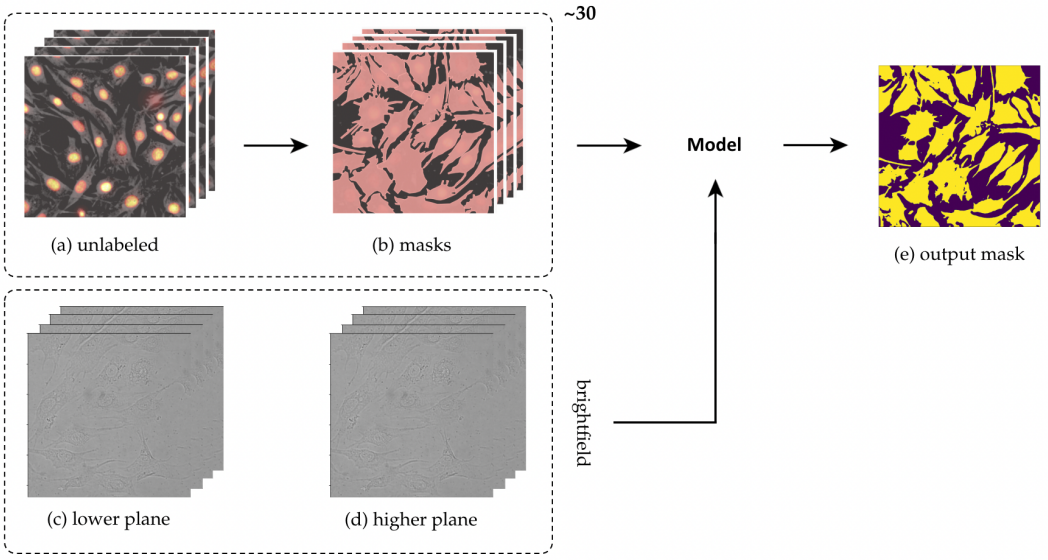


FIGURE 4.3: Dataset and Training Details. We supervise model training with hand-labeled data from an "exhaustive" dataset. We annotate 28 brightfield images and use lower and higher brightfield focal planes for training.

4.2.1 Squeeze-and-Excitation Blocks

SEU-Net is a convolutional neural network architecture that combines the advantages of the famous U-Net architecture with the Squeeze-and-Excitation (SE) blocks. The SE blocks are added to the encoding and decoding pathways of the U-Net to improve the model's feature representation by fusing the spatial and channel information.

The Squeeze-and-Excite block is a common method used in neural networks to improve the quality of feature representations. The SE block performs channel-wise calibration of the output. It generates a weighting vector that is used to scale the output of the convolutional block. The purpose of this weighting vector is to give greater importance to more significant features in the input.

The SE block consists of the squeeze and excitation operations. The squeeze operation reduces the spatial dimensions of the input feature map and aggregates the channel information. This is done by applying global average pooling to the input features, resulting in a single vector output representing the channel information.

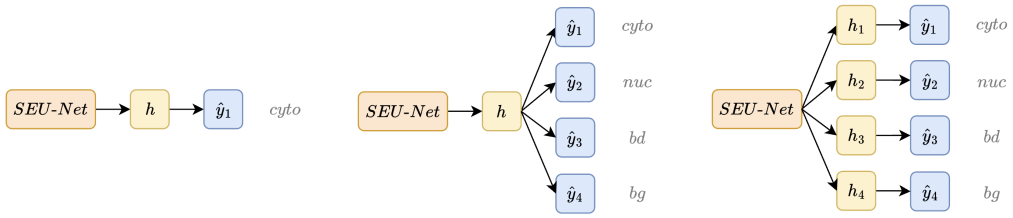
The excitation operation is a non-linear transformation that takes the squeezed vector as input and generates a weighting vector reflecting each channel's significance in the input feature map. This weighting vector is then used to rescale the original feature map, resulting in an output feature map with enhanced feature representation.

4.2.2 Adding more morphology-based information

Following the aforementioned segmentation approach of utilizing the Squeeze-and-Excitation enhanced U-Net model, we hypothesize about improving the score by providing the model with more descriptive features about the sample. To elevate the problem of adding more contextual representation and reducing prediction errors, we follow the idea from the original U-Net paper [50]. In addition to only having one class, we employ border and background classes to reduce the number of corresponding errors. We add a nuclei class to improve the models' perception of cells further. Our experiments which we will discuss later showed that understanding the relationship between the nuclei and the corresponding cytoplasm matter has elevated the overall performance.

4.2.3 Decoupled Decoders

To enable the model multi-task understanding, we employ the multiheaded architecture, extending the decoder with three more auxiliary branches for nuclei, border, and background predictions. We reason that such modification allows the model to simultaneously learn representations for the cell interior, the relation of one-to-one cell-nuclei assignment, cell border, and background, which can help improve the overall segmentation performance.



(A) Baseline (B) Multi-channel (C) Multi-headed

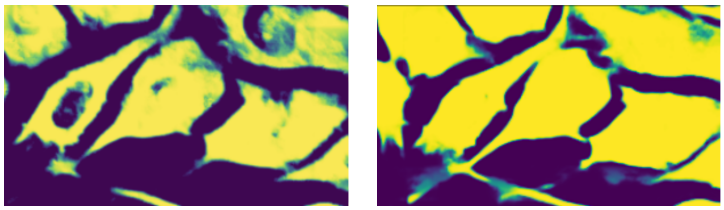
FIGURE 4.4: This figure illustrates different model variations used in our study for cell segmentation. We start with the baseline approach (a) of predicting the cytoplasm channel. Comparably, the multi-channel approach (b) predicts more classes, including cell nuclei, border, and background, for a broader scene understanding. Moreover, we adopt a multi-headed approach (c) that simultaneously learns representations for the cell interior, one-to-one cell-nuclei assignment, cell border, and background.

Each auxiliary branch has its decoder head responsible for predicting the respective class. By having multiple decoder heads, the model can learn to extract features specific to each class, allowing for more accurate predictions.

Furthermore, the auxiliary branches allow us to leverage additional supervision signals during training. Adding multiple decoder heads is essential as it enables the model to learn more complex features related to each object class. By predicting the nuclei, border, and background classes, we can provide the model with additional cues to help it learn more discriminative representations for each class. Such a paradigm is imperative in cases where the boundary between the cell and the background is ambiguous or the cell shapes are highly irregular.

One of the primary objectives of the 4-headed output approach was to address the inconsistency in predicted cytoplasm structure. Upon observing the results of the baseline model, we noticed slight perturbations in the reconstructed mask near

the nuclei regions and the cytoplasm prediction chunks tending to be separate from their origin cell, Figure 4.5.



(A) Baseline results (1-head) (B) 4-head results

FIGURE 4.5: Closing the gaps. Additional morphological image information allows us to punish the model for not predicting the nuclei and for classifying pixels as background simultaneously, thus improving the overall performance. The first image displays the segmentation results of the 1-headed model trained on a cytoplasm channel only. The 4-headed version, utilizing all four channels, visually performs much better.

Targeting per-class optimization allowed us to penalize the model more in the above-mentioned scenarios. We use a weighted linear combination of binary cross-entropy losses [68].

$$\mathcal{L}_{BCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{4.1}$$

With each head focusing on a different aspect of the segmentation task, we combine the losses from each head to generate a final loss.

$$\mathcal{L}_{total} = \lambda_{cyto} \cdot \mathcal{L}_{cyto} + \lambda_{nuc} \cdot \mathcal{L}_{nuc} + \lambda_{bd} \cdot \mathcal{L}_{bd} + \lambda_{bg} \cdot \mathcal{L}_{bg} \tag{4.2}$$

Where λ_{cyto}, λ_{nuc}, λ_{bd}, and λ_{bg} are the weighting coefficients for the cytoplasm, nuclei, cell border, and background losses, respectively. These coefficients represent the relative importance of each component in the final loss function.

4.3 Segmenting Overlapping Cells

With the aforementioned introduction of auxiliary outputs in Section 4.2.3, we have observed that the model is prone to errors in the regions where two or more cells occlude. Both semantic and instance segmentation approaches actively depend on accurate scene understanding. Thus, the ability to precisely segment overlapping foreground instances is a crucial part of either segmentation method.

We try to alleviate the problem of weakly labeled data for accurate segmenting overlaps. Overlapping regions are defined as the intersection of multiple cells. We propose to use an approach that leverages the spatial proximity of cells to identify overlapping regions. We hypothesize that nearby cells are more likely to form overlapping instances, and this information can be used to guide the segmentation model toward regions where overlaps are more likely to occur.

For every cell instance mask m_i of mask M, we compute the distance transform from its border to create a unique map g_i, resulting in a batch of cell-border-based distance maps $D = \{d_0, d_1, \ldots, d_n\}$. Each of the corresponding distance fields d_i accounts for the contribution of m_ith cell to every other pixel in one sample. We combine them using a max-reduction process to generate a single probability map

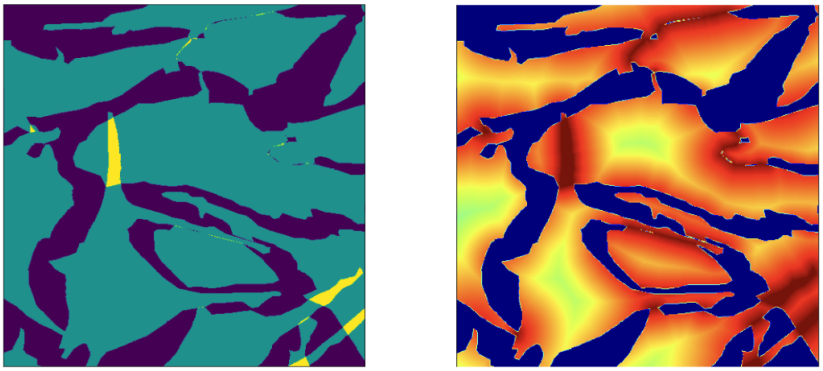


FIGURE 4.6: The left image shows the annotation mask for a 512x512 brightfield sample. The right image displays the final constructed overlap probability map, which incorporates the spatial proximity of cell instances.

P. As an additional reweighting step, we apply log-scaling to refine the probability map further to create a more steep descent toward local centers. This ensures that the probability values are much higher in regions where cells are closer. The probability values of P corresponding to the background region of the mask sample M are set to 0.1.

We exploit the Euclidean distance transform to get the distance maps:

$$d_i(x_i, x_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.3}$$

Followed by an element-wise max-reduction:

$$P = \max_{i=0}^{n} d_i \tag{4.4}$$

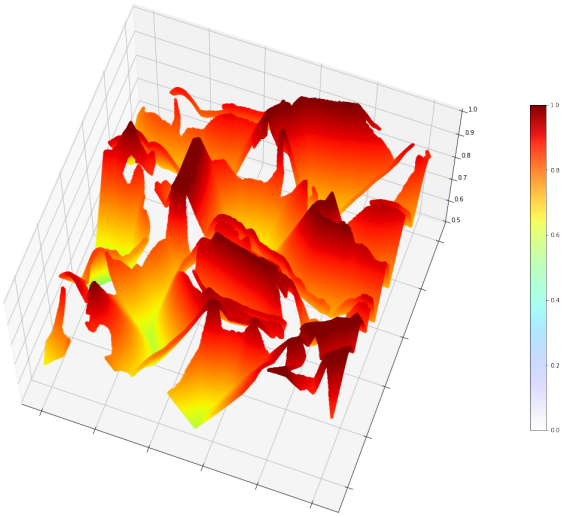


FIGURE 4.7: Overlap Region Probability Map.

While training, the resulting probability map P adjusts the loss by reweighting

the prediction probabilities. Similar to [7], we recompute the loss by multiplying it by $1 - P$. As a result, we degrade values where the confidence of the overlap region is high and continuously punish the model more for predicting high values in regions of low confidence. The model incurs a heavier penalty for predicting overlaps in regions where cells are located far apart and encourage such prediction in places where overlapping instances are more likely to occur. Given the prediction $\hat{y}$, we imply the information about the knowledge of overlap likelihood P to the loss to attend the model to localize and understand overlappings better. To emphasize the fewer overlapping areas, we apply the binary focal loss [38] rather than traditional cross-entropy loss [68] to account for assigning a higher weight to hard-to-classify examples. Mathematically, the reweighted focal loss can be expressed in the following way:

$$\mathcal{L}_{overlap}(y', \hat{y}) = -\alpha(1-\hat{y})^{\gamma} y' \log(\hat{y}) - (1-\alpha)\hat{y}^{\gamma}(1-y')\log(1-\hat{y}), \tag{4.5}$$

where $y' = y(1-P)$

4.4 Synthetic Dataset

As an additional contribution, considering our small annotated portion of the dataset and the low quality of annotated overlaps, in addition to probability maps, we explored different ideas for expanding our training dataset.

One approach would exploit the possibility of using an unlabeled subset of brightfield images and constructing corresponding pseudo labels, which would serve as training data for the later stages. Such dataset construction would not be ideal. Firstly, we would still lack information about the overlap locations as the final output of the model is a binary mask. Next, with such an approach, we lose track of instance information in a larger context of a single sample.

In light of these problems, we propose constructing new synthetic images while preserving annotations on an instance level. We are motivated by prior work of copy-pasting instances between the dataset samples [19]. In addition, the work has shown that such a type of augmentation allows the model to generalize faster and learn better representations. Furthermore, we extend this approach and adopt it to our original cellular datasets.

Since cells are somewhat transparent and function as lenses, copying and pasting them in the brightfield domain cannot produce high-fidelity images. We need to reconstruct the overlapping regions to have upper and lower cells visible enough to identify such regions as overlapping. We have seen from local experiments that the model only segments "fake overlaps" and completely ignores the real ones when we try to copy-paste cells in the brightfield domain.

To solve the problem, we initially set a prior condition over the phase images. We have made this assumption based on the prior empirical evidence gathered by our collaborators from PerkinElmer, who have developed a phase-contrast generating model that we have also used in this work. The properties of phase-contrasts allow for approaching the problem more straightforwardly. Thus, we assume that overlaps are achievable in phase contrast by adding pixel intensities. Meaning that superimposed cells impose approximately additive intensity contribution of individual objects.

We adopt a GAN-based approach to learn and model a non-linear transformation for translation from the phase-contrast domain to the brightfield, along with a custom cellular sample construction module.

4.4.1 Copy-Paste Module

Prior works such as MixUp [70], Mosaic [21], and CutMix [69] are efficient as they combine multiple images on the crop level during training. These techniques have contributed significantly towards achieving robust training. The YOLO family [31] uses multiple combinations of such augmentation types in their pipelines and achieves remarkable results. Nevertheless, such image transformations are not instance-aware. The Copy-Paste procedure [19], on the other hand, is similar to the CutMix and MixUp. The augmentation copies instances between different samples while preserving only annotated parts which makes it an object-aware augmentation. Unlike Contextual Copy-Paste [14], we do not benefit from using context-aware instance pasting due to the random nature of microscope images. Rather, we exploit a random set of transformations to the copying instance before pasting it in. As a further attribution to our method, we extend the usage of the Copy-Paste augmentation to create a new synthetic dataset by first constructing new samples.

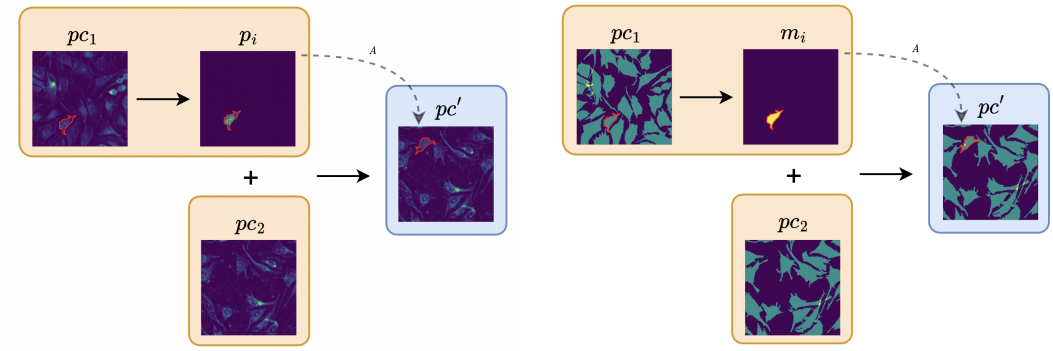


FIGURE 4.8: Process of a single Copy-Paste iteration of a cell between different images along with their annotations. With a selected cell pc_1, we choose a random cell instance p_i with its corresponding mask m_i. For each selected cell, we apply an augmentation A to mask and phase crop. The selected cell gets added to a new image along with the annotations.

Given a set of annotated phase-contrast images P that is not a part of the test set and a corresponding set of masks M of shape (N_i, H, W), where N_i is the total number of annotated individual cells present in the ith sample, we randomly select a group $G_m = \{m_0, m_1, \ldots, m_n\} \in \mathbb{R}^{N \times H \times W}$ of N non-overlapping cells instance masks from the set M. Consecutively, we crop N phase-contrast instances from P corresponding to the selected masks of group G_m : $G_p = \{p_0, p_1, \ldots, p_n\} \in \mathbb{R}^{N \times H \times W}$.

For each selected cell p_i and its annotation m_i from the groups G_m, G_p, we apply a combination of both linear and nonlinear/not affine transformations. To enrich the distribution set, we mainly focus on utilizing resize, rotate, and shift augmentations along with elastic deformations, which incorporate random pixel displacement, modeled using a vector field. The random set of transforms is applied directory to phase crop p_i and mask m_i.

We formulate novel data samples by initializing empty fields P' and M' of shapes $(1, H, W)$, (N, H, W), respectively. Next, we subsequently insert the transformed annotations and image crops into P' and M'. Considering the additive characteristics of phase-contrast modality, we add the intensities of the pasted cell crops p_i, resulting in a completely new image P' with corresponding instance-level annotations M'.

We additionally apply Gaussian blur on the edges of p_i with $\sigma = 20$ to get a more natural-looking transition from the cell to the background. Furthermore, we control the cell injection process by exploiting overlap criteria between two superimposed cells: $0 \leq p \leq 0.2$, where p denotes the overlap ratio of a pasted cell with other cells.

4.4.2 pix2pix

As a next step to the aforementioned construction of more new brightfield images, we propose to use a generative adversarial network (GAN) in the conditional setting. Following pix2pix [30], we use a U-Net-based architecture for the generator G and a "PatchGAN" classifier as a discriminator D. The generator and discriminator networks are trained together in an adversarial setting, where the generator tries to produce images that fool the discriminator. In contrast, the discriminator tries to distinguish between the actual and generated samples correctly. Thus providing the generator with more concrete feedback.

Since generative models require a vast amount of data, we cannot reason the mapping from semantic masks as an input to get a good performance. For one reason, our labeled portion of the dataset is relatively minimal to an extent. Instead, we propose to learn the model to perform a mapping from phase to brightfield domain. This way, we do not require any labeled annotations and can exploit the whole unannotated dataset.

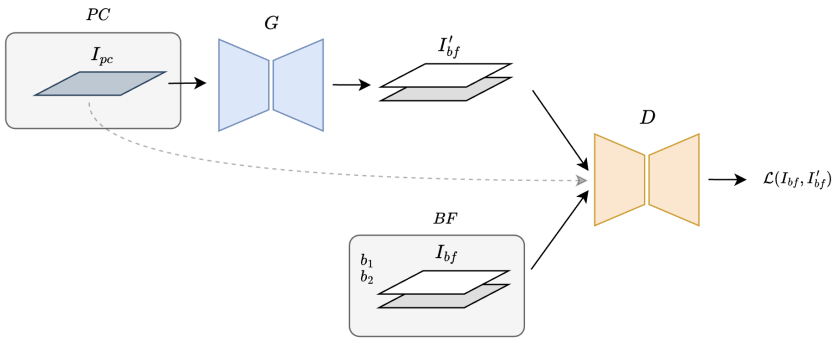


FIGURE 4.9: pix2pix training process overview.

The generator G is U-Net-based architecture and comprises nine stacked convolutional layers with increasing spatial resolution for the encoder. Along with that, the model uses skip connections to preserve high-level features as well as spatial information. Each skip-convolution layer uses 2D convolutions with a kernel size of 4x4, stride of 2x2, and padding of 1x1. Every convolution block is followed by Batch Normalization [29] to ensure features are scaled appropriately to achieve zero mean and unit variance. The original Rectified Linear Unit (ReLU) [1] activation was replaced with the LeakyReLU [65]. LeakyReLU helps to avoid the dead neuron problem during model optimization by introducing a little negative slope in the

negative region of the activation function. The decoder layer consists of 9 upsampling convolutional blocks consisting of transpose 2D convolutions with a kernel size of 4x4, stride of 2x2, and padding of 1x1. These convolutions are used to deconvolve the features by increasing resolution, thus halving the number of feature channels. Every deconvolution block is followed by the same Batch Normalization and a ReLU activation function. At the final layer, a 1x1 deconvolution followed by a Tanh activation function is used to map the resulting 128-dimensional feature vector to 2-channel focal brightfield planes.

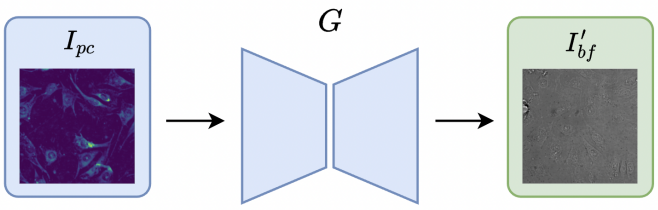


FIGURE 4.10: Process of translating a single phase-contrast image into two brightfield focal planes.

In the training process, we feed the network a 512x512 crop of phase-contrast images. The generator G takes as input image I_{pc} and produces the output of two images representing brightfield focal planes I_{bf}: $I'_{bf} = G(I_{pc})$.

The discriminator is trained in a "PatchGAN" manner. In addition to the output of G, we use conditional information of the initial phase plane I_{pc} by concatenating the channel dimensions and passing the input to the discriminator D. Following the above-mentioned encoder architecture, we use a 5-layer discriminator for classifying "real" and "fake" samples. We encode images by subsequently employing 2D convolutions with a kernel size of 4x4, a stride of 2x2, and padding of 1x1, followed by Batch Normalization and a LeakyReLU at each layer. The output layer 4x4 convolution produces a single probability value identifying whether the predicted sample is "real" or "fake". The output image I'_{bf} conditioned on the phase information gets classified in 70x70 patches rather than the whole image. This method allows discriminator D to make fine-grained decisions about the realism of specific regions of the image for more local information feedback. To achieve this, a 70x70 window is convolved across the entire image. Thus, the output of D is an averaged response from local patches.

We utilize a tiling approach to preserve a full input resolution of 1024x1024. The input image I_{pc} gets split into non-overlapping windows of size 512x512. Generator G takes I_{pc} as an input and outputs the brightfield image I_{bf} with the same cell structure. At the same time, the adversarially trained discriminator D tries to evaluate the quality of the output and distinguish it between "real" and "fake" images.

On inference, we construct a new phase-contrast sample P' using the aforementioned copy-pasting module, described in Section 4.4.1, and feed it to the network. In order to target the blank zero-background inpainting in the new input image P', we utilize 0.5 dropouts in generator layers as a form of noise to get non-deterministic outputs.

We train the conditional model, where generator G tries to fool the adversarial discriminator D and minimize the objective, and D tries to differentiate the actual

data from the generated, thus maximizing the overall objective:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}x, y[\log D(x, y)] + \mathbb{E}x, z[\log(1 - D(x, G(x, z))] \tag{4.6}$$

Since $L1$ loss enables the model to capture low frequencies, it is less prone to output blurry images:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1] \tag{4.7}$$

The final network loss is formulated as a sum of L_{cGAN} and $L1$ losses, we set λ to be 100 as in the original implementation [30].

$$\mathcal{L} = \arg\min_G \arg\max_D \mathcal{L}_{cGAN}(G, D) + \lambda \cdot \mathcal{L}_{L1}(G) \tag{4.8}$$

4.5 Instance Segmentation

Previously, most instance segmentation approaches laboriously relied on object detection and performed mask prediction based on estimated bounding boxes or dense centers [23] [36]. Many of them have been specifically designed for natural objects, which renders them less efficient for biomedical images [53] [12]. Throughout our experiments, we have observed those object detection models notoriously underperformed in scenes with high object density and occlusion of elongated cells. The object detection stage was the primary bottleneck in a whole pipeline due to many ambiguous cell shapes [34].

4.5.1 SparseU-Net

We implemented a detection-free instance segmentation method to prevent all issues related to the poor generalization of object detection pipelines. We target a full segmentation of cells using our U-Net model. Motivated by [11] [63] [64] [73], we enhance the SEU-Net model (Section 4.2) with an instance-aware decoder for sparse prediction outputs. Rather than relying on predicting the region of interest or centers to represent objects, we exploit a similar idea to [11] and elevate instance learning by employing a sparse set of instance activation maps (IAM). Therefore, every object gets represented by its instructive pixel regions. Next, the classification and segmentation are directly performed from the learned instance features.

Many anchor and center-based object detection methods are usually limited by the local contextual information and the receptive field of pixels [67] [10]. Instead, instance activation maps exploit feature aggregation from the entire frame, therefore, achieving broader context awareness and global reasoning.

Compared to the prior methods [23], in scenes where densely packed and occluded objects exhibit complex and unnatural shapes, instance activation maps can serve as a more efficient method of aggregating instance features for segmenting individual instances. Mainly, it can highlight informative object areas and suppresses other redundant pixels. Estimating boxes brings unnecessary background features that are usually inessential, leading to suboptimal segmentation results, considering the task of segmenting overlapping cells [72] [34] [66] [18]. Additionally, the effectiveness of center-based detection methods may become limited as foreground objects acquire varying forms and shapes. Therefore we can not guarantee its robustness in targeting explicit centers of instances.

During training, we push the instance activation maps (IAM) module to attend to informative features. Since IAM output is conditioned on the input image, it exploits arbitrary activation maps for every object. Therefore, we supervise model training through explicit ground truth matching.

4.5.2 Instance Aware Decoder

The decoder consists of two decoupled instance and mask branches. With an instance-aware decoder, the model tries to learn to embed the characteristics of each instance (e.g., intensity, appearance, shape, location, etc.). This is done explicitly by learning kernels produced from instance-aware maps dynamically conditioned on the input. Each of the regressed weight maps produced by the instance branch attributes to the informative object-wise regions. In parallel, the model also learns mask features which are further aggregated with kernels to result in a set of predicted instance masks.

Learning instance activation maps does not exploit direct supervision as we don't have the corresponding ground truth. Instead, the model pushes the instance activation maps module to discover and attend to informative instance-related parts of an image. Learning to comprehend desirable regions belonging to specific foreground objects is enforced with instance matching strategy, which exploits one-to-one ground truth prediction superimposition. This ensures that produced instance activation maps exploit correspondence with ground truth labels once aggregated with mask features.

Namely, the following pair of decoupled pixel-wise mask and instance branches are added to our SEU-Net model to enable cell instance segmentation. As a part of the "vanilla" stage model, we add instance segmentation blocks only to the last layer of the model.

Given the input image, we pass it through the encoder to extract informative features on different levels of the encoder down pass. For the decoder part, we combine upsampled features with the skip connections from earlier layers and exploit instance separation with the resulting concatenated features $X \in \mathbb{R}^{D \times H \times W}$ at the output layer. Both mask and instance branches propagate image features X through a series of convolutional blocks resulting in M and I feature maps, respectively. For our experiments, we adopt four 3×3 kernel convolution blocks for both branches. Then, instance activation maps can be formulated as $A = F(I) \in \mathbb{R}^{N \times H \times W}$, where A is a sparse set of N activation maps that highlight informative regions for every object. F is a simple 3×3 convolution network with sigmoid non-linearity activation. With feature maps X, we obtain instance features by aggregating them with normalized to 1 activation maps $A' : a = A' \cdot X^T$:

$$a = \sum_{k=1}^{HW} A'_{ik} X^T_{kj} \tag{4.9}$$

where $a = \{a_i\}^N \in \mathbb{R}^{N \times D}$ is a group of feature representations for N objects, and A' is a sparse set of normalized to 1 instance activation maps. Therefore, every object gets encoded into a 256-dimensional vector.

With instance features a, the model produces output class vector $c \in \mathbb{R}^{N \times 1}$ and mask kernel $k = \{k_i\}^N \in \mathbb{R}^{N \times D}$ by projecting a to a lower dimension space with two linear layers.

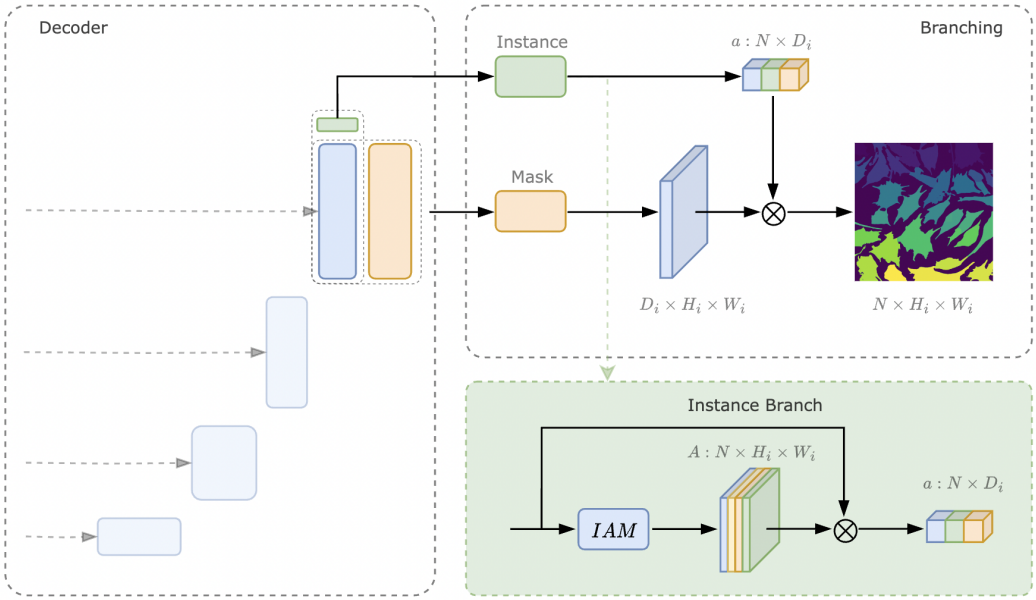


FIGURE 4.11: Proposed SparseSEU-Net model architecture for end-to-end bounding-box free instance segmentation. The decoder at the last layer consists of two mask (orange) and instance (green) branches. In parallel, both branches aim to provide mask and instance features along with mask kernels, respectively. Later, both the mask feature and mask kernel are aggregated and produce sparse instance predictions.

Produced instance-aware mask kernels $\{k_i\}^N$ can be directly aggregated with mask features M to produce final sparse segmentation masks. Specifically, this is done by element-wise multiplication: $m_i = k_i \cdot M$, where M is the mask features, k_i is the corresponding instance kernel which incorporates features related to a specific instance, and m_i is the produced instance mask.

4.5.3 Mask Level Matching

Since the model initially produces N sparse outputs where N is larger than the number of actual instances in an image, it becomes hard to evaluate the prediction with respect to the ground truth. To assess this problem, we utilize a matching strategy to assign predictions to the labeled data and compute losses. We employ the optimal bipartite matching scheme recently proposed in [9] [11], resulting in a set of corresponding $\{prediction, ground\text{-}truth\}$ instance mask pairs. We adopt one-to-one label assignments to eliminate redundant predictions and elevate the ones with the best correspondence. Given a set of M ground-truth masks $G = \{g_0, g_1, \ldots, g_m\}$ and a fixed-size set of N predictions $P = \{p_0, p_1, \ldots, p_n\}$, where $N > M$, we compute losses on the subset of best-matched predictions of P. The one-to-one matching assignment finds a minimum weighted bipartite graph matching $\sigma \in S$ within the sets G and P:

$$\sigma = \arg\min_{\sigma \in S} \sum_{i=1}^{n} C(p_{\sigma(i)}, g_i) \tag{4.10}$$

$$C = C_{cls}^{(1-\lambda)} \cdot C_{mask}^{\lambda} \tag{4.11}$$

where σ is the permutation representing the matching between predicted and ground truth masks that minimizes the sum, S is the set of permutations, and C is a pair-wise matching cost between G and P that is a weighted combination of both classification cost C_{cls} and mask regression cost C_{mask}. We set λ coefficient to 0.8. Each target gets assigned to an object prediction through an optimal assignment problem computed efficiently using the Hungarian algorithm [52]. With the Hungarian approach, we find the optimal match between M ground-truth objects and N predictions given a weighted cost matrix C:

$$\min_{X} \sum_{i=1}^{n} \sum_{j=1}^{n} C(i,j) \cdot X(i,j) \tag{4.12}$$

subject to an assignment X:

$$\begin{aligned} \sum_{i=1}^{N} X(i,j) &= 1 \quad \forall j \\ \sum_{j=1}^{N} X(i,j) &= 1 \quad \forall i \\ X(i,j) &\in \{0,1\} \quad \forall i,j \end{aligned}$$

The cost function C takes into account the class prediction c_i of the output mask and mask similarity score of g_i and p_i. Specifically, we propose a pair-wise dice-based matching score for ground truth and instance prediction masks, where the dice coefficient for g_ith and p_ith masks is defined as:

$$\text{Dice}(p_i, g_i) = \frac{2 \sum_i p_i \cdot g_i}{\sum_i p_i^2 + \sum_i g_i^2} \tag{4.13}$$

We supervise mask prediction with a linear combination of three losses on the positive (matched) predictions. To address the class imbalance between background and foreground in instance masks, we utilize a hybrid mask loss function by combining the Dice Loss [54] and Binary Cross Entropy Loss [68].

$$\mathcal{L}_{Dice}(y, \hat{y}) = 1 - \frac{2 \sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i^2 + \sum_{i=1}^{N} \hat{y}_i^2} \tag{4.14}$$

$$\mathcal{L}_{BCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{4.15}$$

$$\mathcal{L}_{mask} = \lambda_{dice} \cdot \mathcal{L}_{dice} + \lambda_{bce} \cdot \mathcal{L}_{bce} \tag{4.16}$$

Unlike for other losses, for classification, we calculate the loss for all predictions, including the non-matched ones. We use a focal loss [38]:

$$\mathcal{L}_{cls}(p_t) = -\alpha (1 - p_t)^{\gamma} \log(p_t) \tag{4.17}$$

The final training loss is defined in Eq. (4.18) as a combination of mask and classification losses.

$$\mathcal{L} = \lambda_{mask} \cdot \mathcal{L}_{mask} + \lambda_{cls} \cdot \mathcal{L}_{cls} \tag{4.18}$$

Specifically, on inference, we want to rank the predicted masks. We utilize the classification scores to explicitly target every predicted instance's confidence. In addition, we also compute the maskness metrics [63] for every instance: $m = \frac{1}{N} \sum_{i=1}^{N} p$, where p is the predicted probability mask with N pixels. Thus, the combined confidence score s is computed as a contribution of both class confidence c and maskness score m: $s = c \cdot m$

4.5.4 Scaling the Model

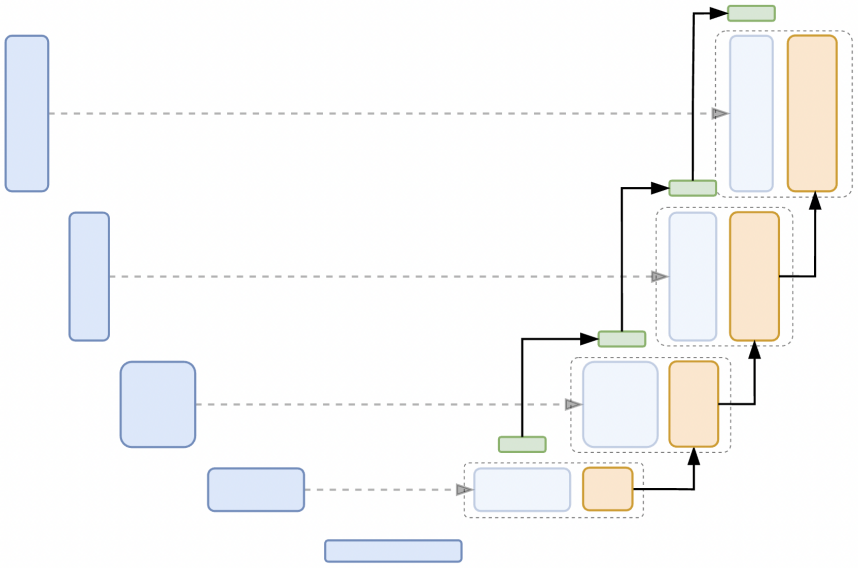


FIGURE 4.12: SparseSEU-Net with multi-level feature aggregation. With the mask and instance branches at the last layer of the "vanilla" model, we extend their usage to every model level.

We further employ the idea and enable the model to learn better semantic and instance representations through a multi-level feature aggregation with the SEU-Net model. Just as in the "vanilla" version of the instance segmentation model where the instance and mask branching takes place only in the output decoder layer (Section 4.5.2), we employ this type of forking on all the decoding layers. The instance branch aims to generate N per-object activation maps with instance features. In parallel, the mask branch is designed to encode instance-aware semantic mask features. With each new level of the decoder branch, we upscale and pass the instance features ${I_i}^{Di \times Hi \times Wi}$ and mask feature maps ${M_i}^{Di \times Hi \times Wi}$ from the layer below.

For every up pass, the mask features, along with the instance features maps, get upsampled by the factor of x2. For each of these explicit passes, we employ concatenation with features from the skip connection to propagate more spatial information until the last level. Thus, the resulting mask and instance feature maps at layer m of the decoder branch are a contribution of a skip connection from the encoder concatenated with the semantic features X_i and upsampled corresponding mask and instance features from the previous layer n. This way, we maintain full feature aggregation from the bottleneck layer to the very top and preserve the branching paradigm.

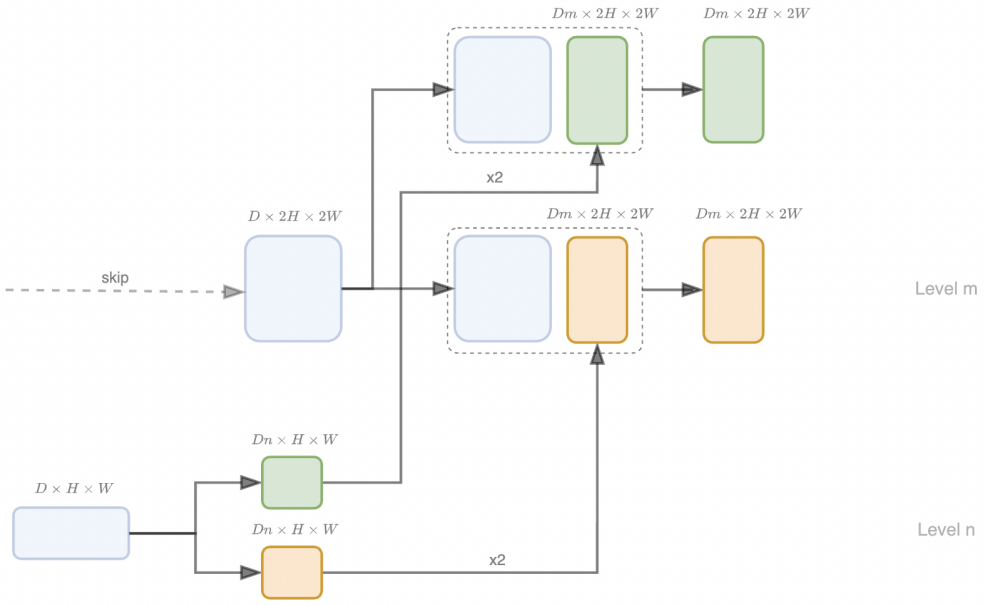


FIGURE 4.13: An instance aware decoder up-pass with skip connection. Both mask (orange) and instance (green) features get propagated from the lower layer of the model to the outer layer via feature aggregation.

Chapter 5

Experiments and Results

We present experimental results mainly on our local manually labeled small-scale "exhaustive" dataset (Section 4.1). The data labels consist of high-fidelity cell segmentations along with the corresponding nuclei and overlap annotations, which allowed us to evaluate the model from different perspectives. To further extend the assessment of model performance, we conduct evaluations on the publicly available LIVECell dataset [15], which consists of expert-validated annotations for phase-contrast modality images. LIVECell is one of the most extensive datasets regarding images and annotated cells, consisting of over 1.6 million cells of diverse morphologies.

5.1 Implementation Details

Our work presents all the results implemented using the Python 3.7 [59] programming language. For training, we used the PyTorch [46] framework, which provides powerful capabilities for neural network building and training. NumPy [22], OpenCV [6], and SciPy [60] were used to handle data manipulation and preprocessing tasks. All of the visual results of our experiments were made using Matplotlib [28], a popular plotting library. These tools formed the backbone of our implementation and allowed us to achieve reliable and reproducible results.

5.2 Training Details

All the experiments were conducted on a single Tesla V100 GPU 32GB, provided by the High-Performance Computing Center of the Institute of Computer Science at the University of Tartu.

To train our model, we have adopted a training scheme published in earlier works [55]. We use AdamW optimizer [42] with an initial learning rate of 1e-5 with a 1e-5 weight decay. Initially, all images were resized to a 1024x1024 size. We adopt random flip, coarse dropout, and easy elastic deformation augmentations when training. While training, we extracted 512x512 random crops from randomly sampled input images and their corresponding masks to accumulate 4 samples per mini-batch.

5.3 Semantic Segmentation

We start our experimentation with the SEU-Net (Sections 4.2, 4.2.2) semantic segmentation model and show how additional morphological features propagate more information and achieve better results. We use our in-house dataset, described in

Section 4.1, to show that the semantic segmentation model can generalize well to unseen data after training on a relatively small subset of images. We split the data into 18 train, 5 validation, and 5 test set images.

We specifically experiment by introducing decoupled decoder branches to target additional semantic features. We extend our model to predicting cytoplasm, nuclei, background, and border. We perform an ablation study to show the importance of additional representations and further assess the overall performance of all the models. The SEU-Net model, in its different variations, was trained from scratch for 1000 epochs. On inference, we sampled sliding windows of size 512x512 with an overlap of 256x256 and averaged them to get a final prediction of size 1024x1024. For a fair comparison, we apply a 0.5 prediction threshold for all the models on all class levels. We visually and quantitatively observe a big jump in performance metrics with an additional nuclei head (2-head model). Similarly, as in Section 4.1 where we observe that overlayed fluorescence data on top of phase image exploits a better visual representation of cell instances. We can also observe a similarly big leap to higher scores when we transition from the 3-headed model, which exploits cytoplasm, cell border, and background prediction, to a 4-headed network that additionally incorporates nuclei information.

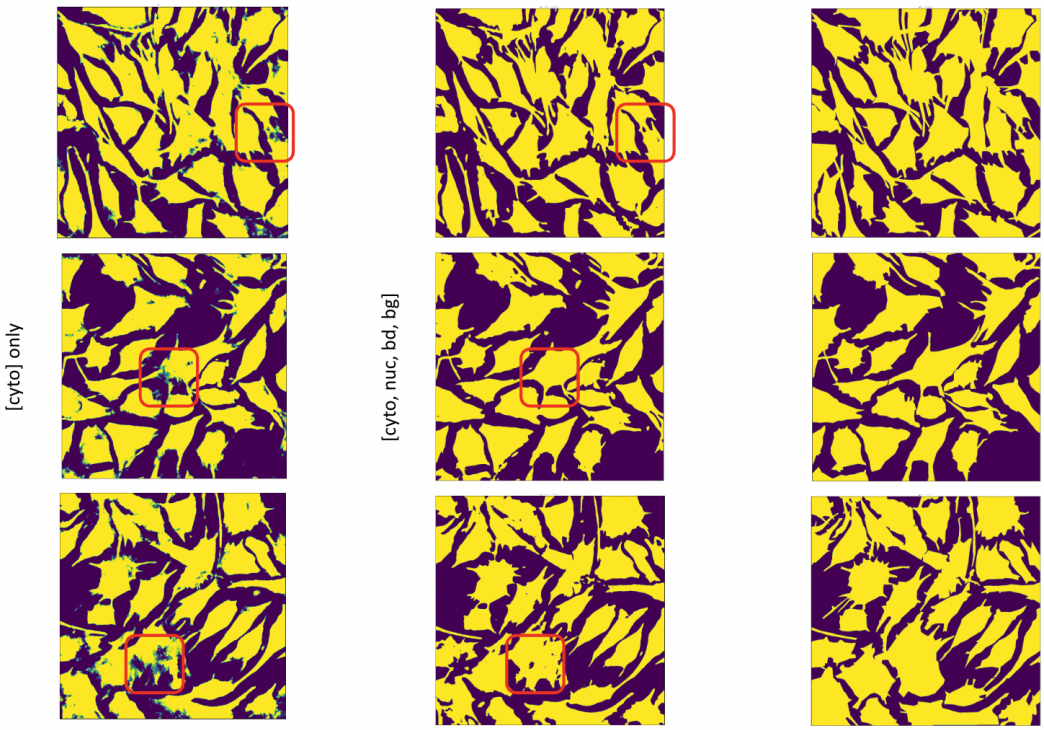


FIGURE 5.1: Visual comparison of the base model trained with cytoplasm channel only and multi-headed model with decoupled decoders. The images show cell segmentation results on sample bright-field images using both model variations. The first column demonstrates the output of the base model, the second column shows the output of the multi-headed model, and the third column shows the ground-truth masks.

Therefore, we reason a boost in segmentation performance as the model attends to understanding the cytoplasm location based on nuclei positioning. Adding more

information about the surrounding scene of a sample brings slightly better performance as the model punishes inaccurate predictions due to the proposed combination of cytoplasm, nuclei, cell border, and background losses. When it comes to evaluating the performance of segmenting the nucleus, we also observe an increase in score numbers. With multiple decoder heads, the model has a direct correspondence in segmenting cytoplasm and nuclei pixels.

Model	Acc.	Prec.	Rec.	F1	IoU
Cyto					
SEU-Net (base)	0.9016	0.9318	0.8600	0.9061	0.8300
SEU-Net (2-head)	0.9214	0.9508	0.9036	0.9266	0.8613
SEU-Net (3-head)	0.9216	**0.9509**	0.9040	0.9268	0.8636
SEU-Net (4-head)	**0.9229**	0.9493	**0.9083**	**0.9282**	**0.8661**

TABLE 5.1: Cytoplasm segmentation performance of the SEU-Net model on the test set of the "exhaustive" dataset.

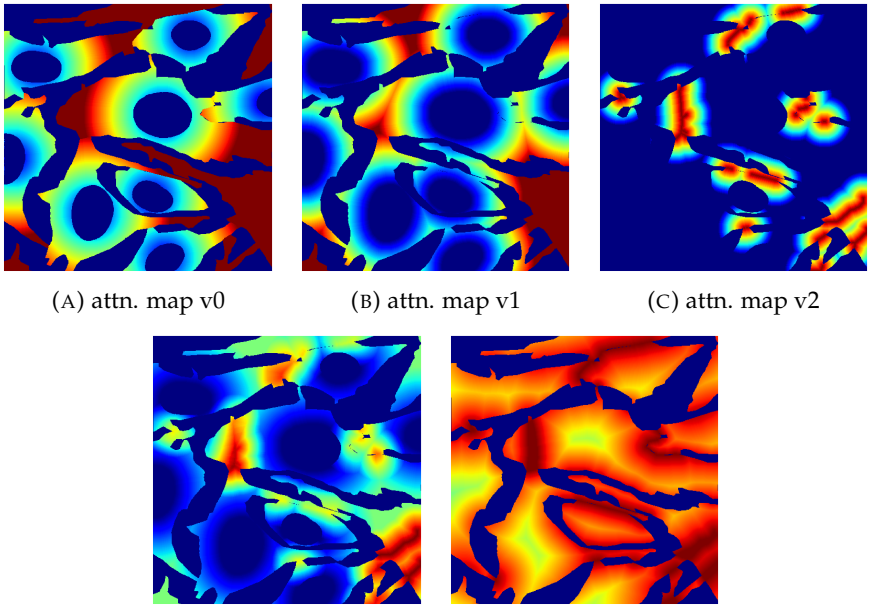
Model	Acc.	Prec.	Rec.	F1	IoU
Nuc					
SEU-Net (base)	0.9796	0.9118	0.8566	0.8834	0.7911
SEU-Net (2-head)	0.9814	0.9113	0.8800	0.8950	0.8101
SEU-Net (3-head)	-	-	-	-	-
SEU-Net (4-head)	**0.9852**	**0.9481**	**0.8847**	**0.9152**	**0.8437**

TABLE 5.2: Nuclei segmentation performance of the SEU-Net model on the test set of the "exhaustive" dataset.

5.4 Segmenting Overlapping Cells

As a target of our study, we also exploit a segmentation over the overlap regions of cells in a single sample. First, we conduct experiments training the SEU-Net model with the same training pipeline. We train models with both cytoplasm and overlap decoders in parallel for segmenting overlaps. On inference, we only use the overlap branch prediction maps and apply a threshold of 0.3 to target the low confidence problem. Given our small dataset with weak overlap labels, we propose to evaluate the performance mainly using F1 and Precision scores. Nonetheless, we include all prediction metrics for a broader comparison. To access the amodal perception information, we utilize probability maps to guide the model. Since all the masks need a corresponding probability map for the model to readjust the loss appropriately, we construct them beforehand.

We proposed and experimented with different variations of weight maps considering scenarios of overlap positioning. In initial experiments, we computed the distances from the nuclei center. We also proposed different approaches for aggregating the probability maps of separate cells, considering neighboring cell influence. Given a cell, we computed its probability map in several ways. In the initial experiments, we used to get the distance map within the bounded space of one cell to indicate the low probability of other cells overlapping the area of the nucleus. We specifically targeted the nuclei location and set the probability value to be very low for the corresponding pixels and the background area (Figure 5.2a). Next, we removed the boundary limit and extended the influence of cells to one another. The computed



(A) attn. map v0 (B) attn. map v1 (C) attn. map v2

(D) attn. map v3 (E) attn. map v4

FIGURE 5.2: Visual overview of different probability maps constructed from given annotations of "exhaustive" dataset. In our experiments, we refer to them as attention maps (attn. maps). All of the above subfigures exploit overlap awareness to some degree.

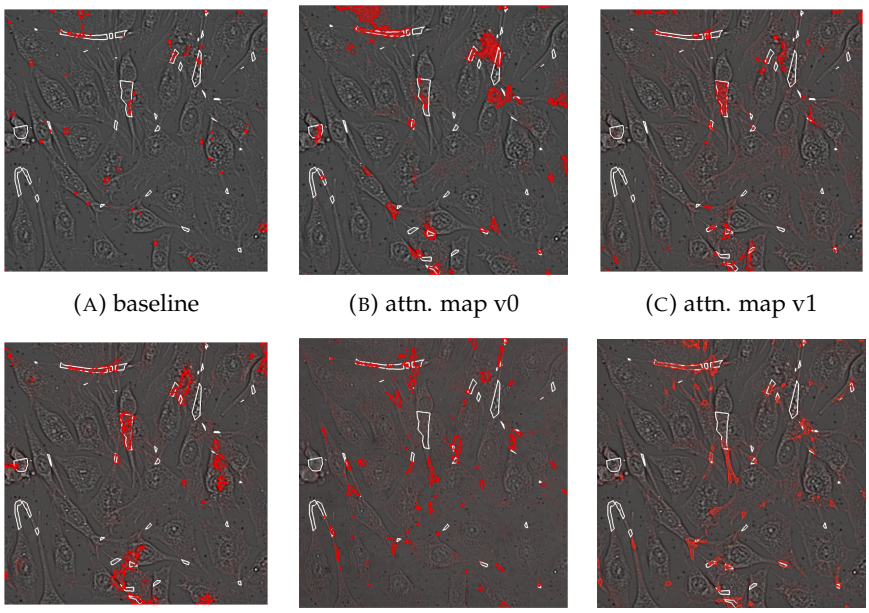
distance maps to every sample point from all nucleus centers were merged using the overall maximum probability value (Figure 5.2b).

Next, we experimented with extending probabilities near the annotated overlap regions to introduce relaxation constraints over the loss function, considering weak overlap annotations (Figure 5.2c). Later, we merge the nuclei-based with the overlap-aware distance map approaches (Figure 5.2d). As the final modification to this approach, we construct a probability map (Figure 5.2e), considering cell borders over cell centers and including overlap awareness, as described in Section 4.3.

Model	**Acc.**	**Prec.**	**Rec.**	**F1**	**IoU**
Overlaps					
SEU-Net	0.9855	**0.3352**	0.0639	0.1073	0.0567
SEU-Net + attn. map (v0)	0.9775	0.1855	0.1914	0.1884	0.1040
SEU-Net + attn. map (v1)	0.9725	0.1549	0.2274	0.1843	0.1015
SEU-Net + attn. map (v2)	**0.9797**	0.2168	0.1858	0.2001	0.1112
SEU-Net + attn. map (v3)	0.9738	0.1865	0.2737	0.2219	0.1248
SEU-Net + attn. map (v4)	0.9760	0.2177	**0.2908**	**0.2490**	**0.1422**

TABLE 5.3: Overlap segmentation performance of the SEU-Net model on the test set of the "exhaustive" dataset. The table shows a comparison of different types of probability maps that were used to train the model.

Table 5.3 shows the influence of model guidance via overlap-aware probability attention maps introduced in the training procedure. Adding overlap knowledge brings significant improvement, specifically of the $F1$ score to 0.25 in overlap segmentation performance. This can be explained by the fact that weight maps exploit lowering the false positives rate on the background region and grouping predictions near the high overlap probability values as the network tends to understand



(A) baseline (B) attn. map v0 (C) attn. map v1

(D) attn. map v2 (E) attn. map v3 (F) attn. map v4

FIGURE 5.3: Visual overlap segmentation performance starting with the model trained solely from given annotations (baseline) and traversing through the results of the model trained with loss recalibration guidance (Section 4.3). Visualized results exploit ground-truth annotations (white) and prediction map activations (red) thresholded to a max value 0.3.

the region localization of overlapping instances. Nonetheless, reweighting probability maps help greatly improve scores on almost all metrics. As an additional factor, we reason great performance based on visual inspection. The output prediction of the model trained solely on overlap annotations exploits many almost random predictions. On the other hand, having the model guided by the initial distance maps, which indicate high probability values of overlap occurrence, allows the model to expoit reasonable predictions. We observe multiple activations in non-annotated regions, which visually exploit intersection regions of overlapping instances.

In the training process, we also experiment with successive ways of attending to overlap regions. The obtained probability map P, described in Section 4.3, is used to adjust the output by recalibrating the loss function. Instead of applying the complement of P directly to the predicted probabilities, following [7], we ensure the importance of overlapping regions by reweighting the entire pixel-wise focal loss [38]:

$$\mathcal{L}'_{overlap}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} P(i) \cdot \mathcal{L}_{overlap}(y_i, \hat{y}_i) \tag{5.1}$$

Nonetheless, we didn't see significant improvements over the proposed method described in Section 4.3

5.5 Synthetic Data

Following the problem with detecting overlaps, we also strive to exhaust our dataset to the fullest and construct synthetic samples following the brightfield distribution. As a goal of ours, we first experiment with a simple cell Copy-Pasting (Section 4.4.1) between different images. Since the brightfield modality is not additive but exploits a more complex way of modeling light scattering, the model started overfitting on the overlaps belonging to newly pasted cells and ignored the real ones. To address this problem, we propose to use a combination of Copy-Paste augmentation and domain translation, described in Section 4.4.2

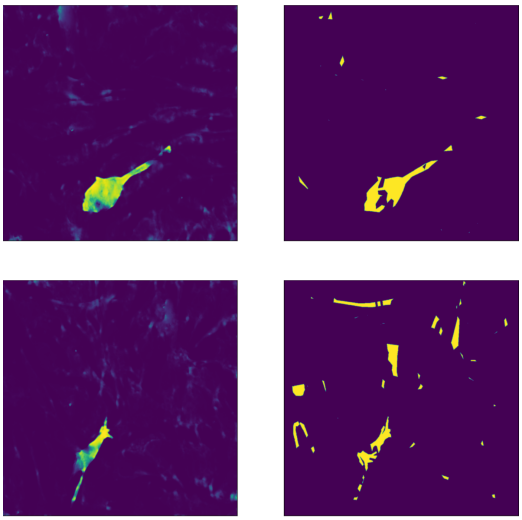


FIGURE 5.4: Overfitting in brightfield Copy-Paste. The left images show the predicted overlaps, and the images on the right depict the annotated overlap regions from combined manual labeling and Copy-Pasting. Model overfits and begins segmenting "fake overlaps" that occur from pasting cells in from different images directly in the brightfield domain.

We exploit the same procedure for training the model on the newly constructed synthetic data as we have for the initial dataset. We also experiment with size variations of the newly constructed dataset and show our results. To explicitly compare synthetic data influence, we compute the metrics on the same test subset from the original data to show that the model could generalize over all segmentation classes with a dataset constructed via image synthesis.

Model	Training Method	Acc.	Prec.	Rec.	F1	IoU
Cyto						
SEU-Net (2-head)	Original	0.9214	**0.9508**	0.9036	0.9266	0.8613
SEU-Net (2-head)	**Synthetic (100)**	**0.9226**	0.9192	0.9418	0.9304	0.8698
SEU-Net (2-head)	**Synthetic (1000)**	0.9221	0.9123	**0.9494**	**0.9305**	**0.8700**

TABLE 5.4: Cytoplasm segmentation performance of the SEU-Net model trained on different datasets, including the original and synthetic data of different sizes on the test set of the "exhaustive" dataset.

We observe an improvement for almost every metric in multiple segmentation tasks, including the segmentation of cytoplasm, nuclei, and overlapping regions in

Model	Training Method	Acc.	Prec.	Rec.	F1	IoU
Nuc						
SEU-Net (2-head)	Original	0.9852	**0.9481**	0.8847	0.9152	0.8437
SEU-Net (2-head)	**Synthetic (100)**	0.9830	0.9031	0.9094	0.9062	0.8285
SEU-Net (2-head)	**Synthetic (1000)**	**0.9865**	0.9318	**0.9177**	**0.9247**	**0.8600**

TABLE 5.5: Nuclei segmentation performance of the SEU-Net model trained on different datasets, including the original and synthetic data of different sizes on the test set of the "exhaustive" dataset.

Model	Training Method	Acc.	Prec.	Rec.	F1	IoU
Overlap						
SEU-Net (3-head)	Original	**0.9855**	**0.3352**	0.0639	0.1073	0.0567
SEU-Net (3-head)	**Synthetic (100)**	0.9620	0.1151	0.2661	0.1607	0.0874
SEU-Net (3-head)	**Synthetic (1000)**	0.9729	0.1955	**0.3150**	**0.2412**	**0.1372**

TABLE 5.6: Overlap segmentation performance of the SEU-Net model trained on different datasets, including the original and synthetic data of different sizes on the test set of the "exhaustive" dataset.

brightfield images. We also indicate a boost in overlaps segmentation performance. With our limited "exhaustive" dataset, the model struggles to identify and grasp the understanding of intersecting cell regions. The problem of initial low-quality segmentation performance can be explained by the fact that the small-scale dataset exploits inaccurate annotations for the overlap areas. Therefore, we find it crucial for the model to have more available data and information to attend to these regions accurately. The proposed approach, on the other hand, exploits nearly perfect labels for overlapping locations. Since our Copy-Paste procedure is controlled to establish a considerable degree of freedom for creating new annotations, the resulting synthetic dataset exploits a massive variety of labeled overlap occurrences. We observe comparable results to those in **??**, which prove the wight maps to be a crucial part of guiding the model training accurately.

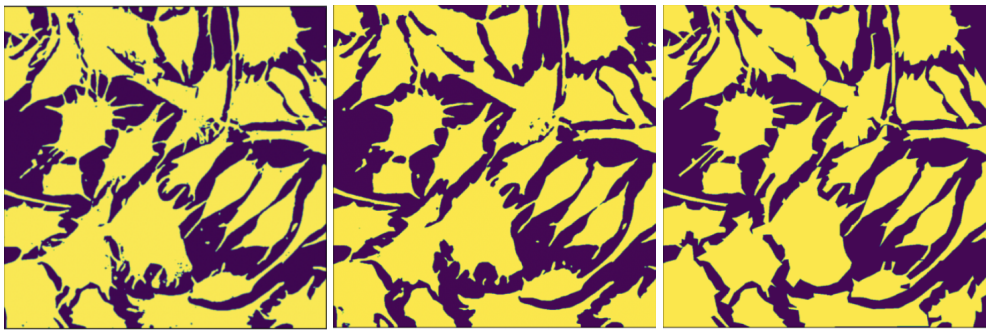


FIGURE 5.5: Prediction on the original "exhaustive" dataset sample from the test group, with models trained on synthetic and original datasets. Noticeably, the model trained on synthetic data performs better semantic map reconstruction. The model pays more attention to small details, some of which were even missed during the annotation process resulting in a more detailed segmentation.

5.6 GANs

Since the brightfield domain is not ideal for the direct construction of new data, we use our phase-contrast domain to guide the model to output realistic samples with a piece of rich feature information. We assume that the additive property of phase image is only an approximation and use it for modeling overlaying cells. We reason assumption based on the prior empirical evidence gathered by our collaborators from PerkinElmer, who have developed a phase-contrast generating model that we have also used in this work. Therefore, we propose to use a conditional generative adversarial network (cGAN) to model the translation between the two domains.

We start by formulating a problem of creating a synthesized sample by merging domain translation and Copy-Pasting (Section 4.4.1). Since directly moving cells from place to place in brightfield images proved challenging, we opted to train a cGAN to perform the transition from synthesized samples of the simpler domain to brightfield focal planes.

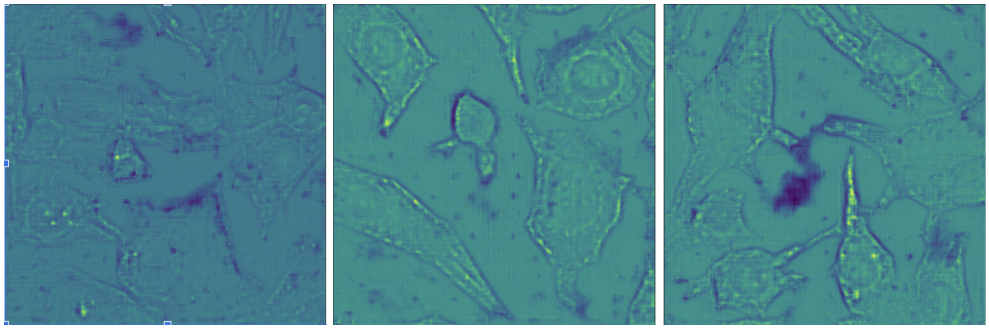


FIGURE 5.6: pc2bf. Visual brightfield generation results from the phase-contrast image using a U-Net-based autoencoder.

First and foremost, we design a simple U-Net autoencoder to manage translation from phase to brightfield domain. For training, we utilize our whole unannotated portion of the dataset to enrich the amount of fed-in data. As input, we pass random 512x512 crops. We test the model training with a combination of Mean-Square Error (MSE) and Binary Cross-Entropy (BCE) losses to balance out the perceptual quality of the output. On inference, we first perform a straightforward single-cell pasting between different annotated phase samples. Then we pass the new image to the model to get the two final brightfield planes. We have observed that an autoencoder model had trouble converging and did not capture intricate patterns to generate highly realistic brightfield samples from phase images. As an additional problem, all the dataset samples already exploit high cell density. Immense object solidity introduced a small degree of freedom when constructing new samples. With such a limitation, newly pasted cells have created highly out-of-the-distribution overlapping visually distinguishable instances.

As a follow-up approach, we were motivated to establish a more flexible and controllable method for constructing new brightfield planes. Pix2pix being a state-of-the-art baseline model, was ideal for this task. With the introduction of the "PatchGAN" discriminator, the model could converge better and achieve plausible visual results. Since this type of discriminator is specifically designed to work on local patches, it was able to provide necessary feedback to the generator. As a result,

the model managed to grasp high-frequency data along with low-frequencies supervised by the L1 loss. We utilize all the available dataset images for the model training. To ensure the model's convergence ability, we introduced Fréchet Inception Distance (FID) [25] score metric and used it along with the L1 loss output. The FID metric is used to evaluate the quality and diversity of generated images. To compute the similarity score of the generated to the actual data, we extracted a 2048-dimensional feature representations vector for each minibatch using a pretrained Inception-v3 network [56]. The final score was computed as a Euclidean distance between the mean and covariance matrices of the real and generated features. For training purposes, we follow an approach from [20]. We use minibatch SGD and apply the AdamW solver [42], with an initial learning rate of 2e-4, and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. We apply dropout with a 0.5 rate on inference to output non-deterministic results.

In the second stage of synthesizing data, the trained pix2pix model maps phase images to brightfield focal planes. We construct new input data, following the mentioned earlier Copy-Paste approach in Section 4.4.1. After, the new sample is fed to the model. As a result, we get the final brightfield modality image along with the constructed annotations. We have observed that the trained model does not struggle with generating background noise in brightfield samples from spaces resulting from pasting instances onto a new blank canvas (Section 4.4.1).
Visual results for this methodology proved to be indistinguishable from the real data by inspection.

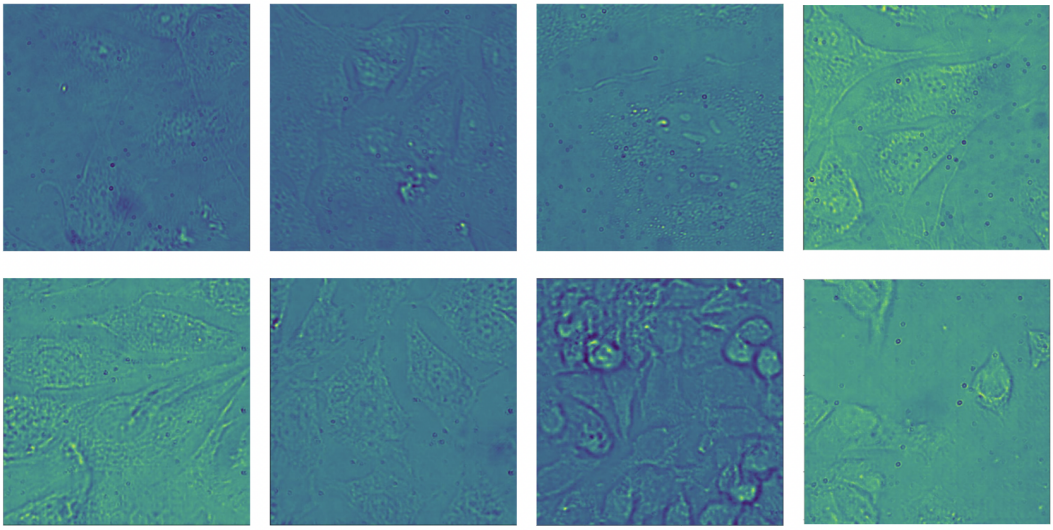


FIGURE 5.7: pc2bf. Visual brightfield generation results from the phase-contrast image using pix2pix. To compare the robustness of brightfield generation, we provide real data of four images on the top row and generated focal planes on the bottom.

In the earlier stages of our experiments, we only incorporated the semantic data as an input to the pix2pix model as a part of *mask2img* translation. We have observed that the model struggled to produce plausible brightfield images. The problem of generating high-fidelity images from labels can be resonated with the fact that the generative models require enormous amounts of data. Therefore, our small annotated subset is insufficient for this task in given conditions.

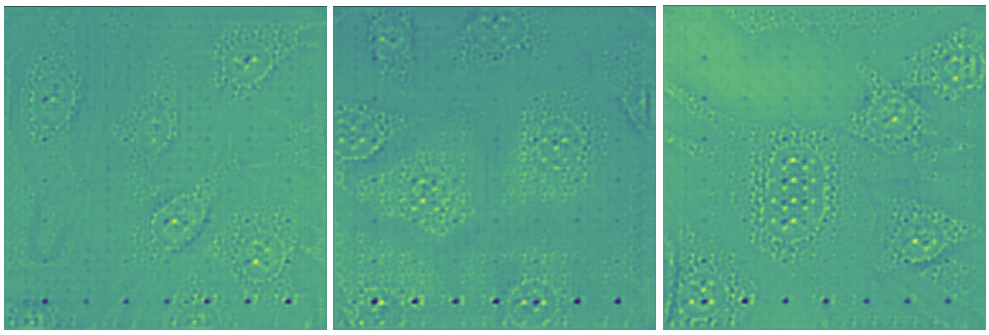


FIGURE 5.8: mask2bf. Visual brightfield generation results from the segmentation mask using pix2pix.

5.7 Instance Segmentation

For our main results of instance segmentation, we report COCO [39] mask AP scores on the test subsets for all datasets. We also propose to compare the performance with the state-of-the-art models in both natural and cellular domains. For this, we use Mask-RCNN [23], YOLOv5 [31], YOLOv8, and CellPose [53] models, as they have been widely used in many cases of cell segmentation.

We divide our local "exhaustive" dataset (Section 4.1) using the same split strategy resulting in 18 train, 5 validation, and 5 test images. For fairness, we train the models for 2000 epochs with a batch size 8 on the same dataset partitions. Instead of sampling random crops, we resize the input images to 512x512 to ensure the same evaluation for all the approaches. All the models used for comparison were initialized with pretrained weights on respective datasets. Thus, we might expect slight variations in the results.

During inference, we maintained the same threshold coefficients for the non-maximum suppression overlap and confidence threshold for object detection-based networks and used the same 0.5 mask predictions threshold for all the trained models.

Considering our approach, we train the SparseSEU-Net model from scratch for 2000 epochs on resized 512x512 images. Besides the model's sparse N outputs, we had to sample corresponding ground truth masks. This limited our memory usage on a single GPU machine. Resulting in 100 output masks with a batch size of 2, the training process occupied 28 Gb of storage. To resolve the problem of memory overhead, we acquired gradient accumulation of 4 batches into the training pipeline to store the gradients for several forward passes and run backpropagation. Regardless, we indeed expect a much higher performance with an increased number of GPUs.

We start by training the model with a small initial learning rate of 1e-5 with 1e-5 weight decay. We later change the learning rate and decay values to 5e-5 and 0.005, respectively, following [11]. We adopt N=100 sparse outputs for each image.

Additionally, we apply the watershed transform to our SEU-Net semantic segmentation model from Section 4.2 with four decoupled outputting branches for multiple modalities. Table 5.7 shows our model achieves a comparable result to the state-of-the-art models, reaching 75.8 mAP. We also observed that our model could capture smaller objects compared to every other model. Nonetheless, we assume that the model has yet to perform better once with pretrained weights. This experiment also delegates the importance of accurately segmenting overlapping cells.

Directly learning distinct instances instead of depending on manual postprocessing of semantic segmentation enables the achieving of more accurate results. Since the watershed technique cannot recover whole cells due to a large number of object occlusions, it has proven to be prone to errors of assigning pixels to wrong instances, even with a highly detailed segmentation performance.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN*	22.2	60.7	10.8	0.0	6.1	26.2
YOLOv5*	33.7	79.0	25.9	0.0	20.5	38.8
YOLOv8*	**39.8**	**79.5**	**34.8**	0.0	21.8	45.0
CellPose*	32.2	68.7	27.7	0.0	10.2	37.2
SEU-Net + wtshd	27.6	43.2	31.1	0.5	17.9	**57.6**
SparseSEU-Net	39.6	75.8	34.1	**12.7**	**42.8**	39.5

TABLE 5.7: Cell instance segmentation mask AP (%) performance on the test set of the "exhaustive" dataset. We also compare instance segmentation results with the initial SEU-Net model with applied watershed transform [49]. The * denotes model initialization with pretrained weights before training.

For the LIVECell data, we utilize the existing split train, validation, and test subsets. Considering the heavy burden and high computational cost associated with training on a large dataset, we choose to limit our training data to only 2% of the available samples for all models. Since now our model is limited to outputting 100 sparse masks, we had to make sure to include image crops with no more than 90 instances. We subsequently utilize the same training procedure as for the local dataset. We train all the available models for 400 epochs and report instance segmentation results on a complete set of test images. We observe a similar performance in segmentation compared to every other model (Table 5.8). Nonetheless, we witness a similar behavior of segmentation performance of objects of different sizes as in with the "exhaustive" dataset. The model outperforms all the competitors, exhibiting prominent performance in smaller object segmentation by achieving AP_S of 26.5 % beating the second-best result of 8.9 AP. The network also achieves comparable results of AP_M and AP_L.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN*	14.4	32.5	10.0	4.4	16.1	20.8
YOLOv5*	15.9	42.8	9.1	0.7	16.5	24.7
YOLOv8*	**26.5**	**53.1**	**24.6**	2.5	**23.5**	37.5
CellPose*	24.8	48.1	24.2	8.9	23.0	**44.6**
SparseSEU-Net	21.8	44.7	22.3	**26.5**	18.2	39.8

TABLE 5.8: Cell instance segmentation mask AP (%) performance on the test set of the LIVECell dataset. The * denotes model initialization with pretrained weights before training.

5.7.1 Adding Overlap Awareness

Empirically we have observed that our U-Net network, enhanced with Squeeze-and-Excitation (SE) blocks and decoupled auxiliary decoders (Section 4.2.2), was able to produce a viable overlap segmentation approach. To leverage accurate instance reconstruction, we also enhanced our SparseSEU-Net model similarly. We add a side decoder branch to learn overlap representations in parallel with instance

semantic representations from the initial model version. Following the [33] approach, we hypothesize a performance boost with feature aggregation. Therefore, we conduct experiments with

We formulate feature merging in the following way. Given input features I, we decode them with overlap and instance-aware decoder branches, resulting in X_o and X_i feature maps, respectively. Since overlap areas are instance-agnostic, we can directly combine overlap features with grouped instance features. Before forking into mask and instance branches (Section 4.5.2) on the last decoder layer, we aggregate feature maps from parallel decoders by conditioning the instance-aware features on overlap features. Thus, the resulting dimension is increased by a factor of x2. We idealize that additional information about the overlapping regions might help produce more accurate instance predictions.

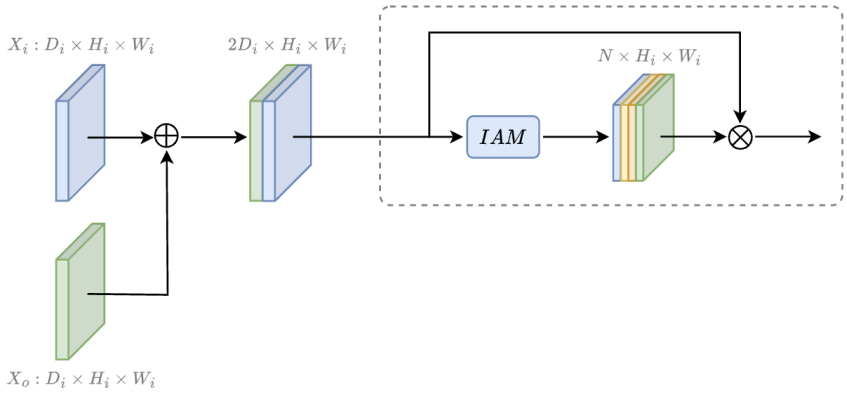


FIGURE 5.9: Adding overlap awareness to the decoder layer. The model employs two parallel decoupled decoders to learn overlap and instance representations. The IAM module outputs instance activation maps. The information about the overlaps is passed to the IAM by adding conditional overlap features X_o. $\bigoplus$ denotes the concatenation operation.

We perform an ablation study by sequentially feeding more information to the "vanilla" version of the SparseSEU-Net model. We first propose to validate the segmentation performance by introducing the auxiliary overlap-aware head and aggregating both cell and overlap features for reconstruction. Then, we enhance the training procedure by attending to more probable overlap regions to directly optimize overlap segmentation with the wight maps proposed in Section 4.3. Overall, we have not seen any performance gain in any score metrics. We assume that adding conditional overlap information affects the general training of the model. Instead of leveraging the information about the common region of multiple overlapping cells to separate objects into multiple instances, the model tries to merge them. As a result, we observe a performance drop in cases of segmenting especially large objects that overlap more often than small-sized cell instances shown in Table 5.9.

5.7.2 Multi-Level Feature Aggregation

As the next step towards a more accurate segmentation, we exploit scaling the overall architecture of the model. We apply the same forking into mask and instance branches, as described in Section 4.5.2 and propagate the feature information from the bottleneck level of the decoder to the very top. Table 5.10 shows that with

overlap head	attn. mask	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
		39.6	75.8	34.1	**12.7**	**42.8**	**39.5**
✓		38.0	74.7	35.2	7.6	42.5	23.6
✓	✓	38.8	**76.9**	**35.4**	9.2	42.7	29.1

TABLE 5.9: Ablation study of cell instance segmentation mask AP (%) performance of the SparseSEU-Net. We consecutively increase the model's overlap awareness by first introducing the overlap decoder branch. We then enhance the training with the weight maps (attn. maps) for better overlap supervision.

multi-feature aggregation, the model was able to converge much faster and get high validation results in the relatively early stages of training. Besides, by passing information from lower to layers, the model achieved the best performance of $mAP_{50:95}$ and mAP_{75} scores surpassing all previous approaches. We find that the multi-level SparseSEU-Net model could segment smaller- and average-sized cells more accurately, surpassing the "vanilla" version by a large margin.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
single-level	39.6	**75.8**	34.1	12.7	42.8	**39.5**
multi-level	**44.3**	74.6	**46.8**	**16.0**	**48.3**	36.0

TABLE 5.10: Cell instance segmentation mask AP (%) performance of the "vanilla" version of the model vs. multi-level feature aggregation SparseSEU-Net.

5.7.3 IoU Aware Objectness

In addition to the classification outputs, similar to implementation [11] [27] [35], we additionally experiment with model training by adding the objectness scores to explicitly target the confidence of every predicted instance. Specifically, on inference, we want to rank the predicted masks. To ensure the model understands the predictions it is producing and gives more reasoning about the instance shapes, we estimate the IoU between each predicted mask and the ground truth object it covers. The IoU prediction head is trained with a mean-square-error loss between the IoU prediction, and the IoU predicted with the ground truth masks.

$$\mathcal{L}_{obj}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.2}$$

The final training loss is defined is then defined in Eq. (5.3) as a combination of mask, IoU-aware objectness, and classification losses.

$$\mathcal{L} = \lambda_{mask} \cdot \mathcal{L}_{mask} + \lambda_{obj} \cdot \mathcal{L}_{obj} + \lambda_{cls} \cdot \mathcal{L}_{cls} \tag{5.3}$$

Nonetheless, we observe a huge performance drop as the model struggles to generalize on the "exhaustive" data. Resulting in a near 0.25 mAP score. We reason for this problem by the fact the model also struggles to produce unique instances. Meaning that the outcome predictions result in many duplicates and similar outputs, thus creating a contradictive training process between classification and objectness optimization.

5.7.4 Error Analysis

We perform error analysis in two iterations to get a better quantitative understanding of SparseSEU-Net for instance mask predictions. First, we propose to replace the sparse predictions with the corresponding ground truth labels. Specifically, we compute the dice scores with ground truth masks for each predicted binary mask and replace the predictions with the best-matched ground truth (w/ gt mask). The model achieves substantially good results. Results in Table 5.11 suggest that there is still a big potential for improving segmentation branches for learning a better representation of semantic features for a more accurate instance reconstruction.

Then, we set up the same prediction-ground truth matching scheme (gt matched) and only keep predicted masks that resolve in high dice scores with their corresponding ground truth annotations. We observe an expected huge leap and increase of mAP_{50} scores up to 83.1 %. This way of the evaluation show that, indeed, the mask confidence scoring described in Section 4.5.3 is not ideal; thus, there is a big room for improving the mask ranking in order to identify the ones with correct high confidence.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
baseline	44.3	74.6	46.8	16.0	48.3	36.0
w/ gt mask	97.8	97.9	97.8	95.4	98.2	100.0
gt matched	47.8	83.1	48.9	20.9	51.2	40.1

TABLE 5.11: Comparing analysis of cell instance segmentation mask AP (%) performance. (w/ gt mask) indicates the ideal performance when we substitute the best prediction with the ground-truth data. (gt matched) metrics show the performance of the best-matched predictions with the ground-truth data.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this work, we primarily have shown that multi-task learning enables enriching the performance of the semantic segmentation model. We specifically introduced decoupled auxiliary decoder branches to give the model more context information. We have underlined the importance of segmenting the occlusion regions of overlapping cells and identified key factors to achieve better performance by introducing overlap-aware probability maps. The experiments demonstrate that proposed weight maps exploit a significant boost in overlap segmentation performance. Additionally, we designed a way to generate annotated synthetic brightfield images. We establish a newly constituted large-scale dataset and show that segmentation performance increases once trained on the synthesized images. As one of the main contributions to our work, we enhance our SEU-Net model with instance segmentation blocks. By enabling multi-level feature aggregation, we show that the model is able to achieve state-of-the-art results.

6.2 Future Work

At our current research stage, we considered a few possibilities for improving the instance segmentation pipeline. As an additional attribution, we wanted to ensure that the predicted spares masks exploit uniqueness, which means that a single prediction mask shape is not repeated. Till this point, we have observed multiple duplicate masks, some of which were effectively disregarded by the confidence of the initial predictions. Nonetheless, there are ideas to target this problem. Additionally, we plan to measure the performance of our approach on other downstream tasks, including segmenting cells in different modalities. We hope that our work in the future can be a key to a more robust cell instance segmentation.

Bibliography

[1] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE].

[2] Mohammed A S Ali et al. "ArtSeg—Artifact segmentation and removal in brightfield cell microscopy images without manual pixel-level annotations". In: *Scientific Reports* 12.1 (July 2022), p. 11404.

[3] Mohammed A S Ali et al. "Evaluating Very Deep Convolutional Neural Networks for Nucleus Segmentation from Brightfield Cell Microscopy Images". en. In: *SLAS Discov* 26.9 (June 2021), pp. 1125–1137.

[4] Mikael Björklund et al. "Identification of pathways regulating cell size and cell-cycle progression by RNAi". In: *Nature* 439.7079 (2006), pp. 1009–1013. ISSN: 1476-4687. DOI: 10.1038/nature04469. URL: https://doi.org/10.1038/nature04469.

[5] Michael Boutros, Florian Heigwer, and Christina Laufer. "Microscopy-Based High-Content Screening". In: *Cell* 163.6 (2015), pp. 1314–1325. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2015.11.007. URL: https://www.sciencedirect.com/science/article/pii/S0092867415014877.

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] Francesco Caliva et al. *Distance Map Loss Penalty Term for Semantic Segmentation*. 2019. arXiv: 1908.03679 [eess.IV].

[8] John Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

[9] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].

[10] Qiang Chen et al. *You Only Look One-level Feature*. 2021. arXiv: 2103.09460 [cs.CV].

[11] Tianheng Cheng et al. *Sparse Instance Activation for Real-Time Instance Segmentation*. 2022. arXiv: 2203.12827 [cs.CV].

[12] Junwei Deng et al. "CellSegNet: an adaptive multi-resolution hybrid network for cell segmentation". In: *Medical Imaging 2022: Digital and Computational Pathology*. Ed. by John E. Tomaszewski, Aaron D. Ward, and Richard M. Levenson. Vol. 12039. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Apr. 2022, 1203914, p. 1203914. DOI: 10.1117/12.2605439.

[13] Johannes P.F. D'Haeyer. "Gaussian filtering of images: A regularization approach". In: *Signal Processing* 18.2 (1989), pp. 169–181. ISSN: 0165-1684. DOI: https://doi.org/10.1016/0165-1684(89)90048-0. URL: https://www.sciencedirect.com/science/article/pii/0165168489900480.

[14] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. *Modeling Visual Context is Key to Augmenting Object Detection Datasets*. 2018. arXiv: 1807.07428 [cs.CV].

[15] Christoffer Edlund et al. "LIVECell—A large-scale dataset for label-free live cell segmentation". In: *Nature Methods* 18.9 (Sept. 2021), pp. 1038–1045.

[16] Shannon E. Elf and Jing Chen. "Targeting glucose metabolism in patients with cancer". In: *Cancer* 120.6 (2014), pp. 774–780. DOI: https://doi.org/10.1002/cncr.28501. eprint: https://acsjournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cncr.28501. URL: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/cncr.28501.

[17] Dmytro Fishman et al. "Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescently labelled samples". en. In: *J Microsc* 284.1 (June 2021), pp. 12–24.

[18] Patrick Follmann and Rebecca König. *Oriented Boxes for Accurate Instance Segmentation*. 2020. arXiv: 1911.07732 [cs.CV].

[19] Golnaz Ghiasi et al. *Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation*. 2021. arXiv: 2012.07177 [cs.CV].

[20] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[21] Wang Hao and Song Zhili. "Improved Mosaic: Algorithms for more Complex Images". In: *Journal of Physics: Conference Series* 1684 (Nov. 2020), p. 012094. DOI: 10.1088/1742-6596/1684/1/012094.

[22] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585 (2020), 357–362. DOI: 10.1038/s41586-020-2649-2.

[23] Kaiming He et al. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].

[24] Kaiming He et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 346–361. DOI: 10.1007/978-3-319-10578-9_23. URL: https://doi.org/10.1007%2F978-3-319-10578-9_23.

[25] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG].

[26] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2019. arXiv: 1709.01507 [cs.CV].

[27] Zhaojin Huang et al. *Mask Scoring R-CNN*. 2019. arXiv: 1903.00241 [cs.CV].

[28] John D Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in science & engineering* 9.3 (2007), pp. 90–95.

[29] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167 [cs.LG].

[30] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV].

[31] Glenn Jocher et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Version v7.0. Nov. 2022. DOI: 10.5281/zenodo.7347926. URL: https://doi.org/10.5281/zenodo.7347926.

[32] Cefa Karabağ et al. "Semantic segmentation of HeLa cells: An objective comparison between one traditional algorithm and four deep-learning architectures". en. In: *PLoS One* 15.10 (Oct. 2020), e0230605.

[33] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. "Deep Occlusion-Aware Instance Segmentation with Overlapping BiLayers". In: *CVPR*. 2021.

[34] Alexander Kirillov et al. *InstanceCut: from Edges to Instances with MultiCut*. 2016. arXiv: 1611.08272 [cs.CV].

[35] Alexander Kirillov et al. *Segment Anything*. 2023. arXiv: 2304.02643 [cs.CV].

[36] Youngwan Lee and Jongyoul Park. *CenterMask : Real-Time Anchor-Free Instance Segmentation*. 2020. arXiv: 1911.06667 [cs.CV].

[37] G Li et al. "Segmentation of touching cell nuclei using gradient flow tracking". en. In: *J Microsc* 231.Pt 1 (July 2008), pp. 47–58.

[38] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. 2018. arXiv: 1708.02002 [cs.CV].

[39] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312.

[40] Shu Liu et al. "Path Aggregation Network for Instance Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8759–8768. DOI: 10.1109/CVPR.2018.00913.

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV].

[42] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[43] Brian J. Morrison, John C. Morris, and Jason C. Steel. "Lung cancer-initiating cells: a novel target for cancer therapy". In: *Targeted Oncology* 8.3 (2013), pp. 159–172. ISSN: 1776-260X. DOI: 10.1007/s11523-012-0247-4. URL: https://doi.org/10.1007/s11523-012-0247-4.

[44] Larry E. Morrison et al. "Brightfield multiplex immunohistochemistry with multispectral imaging". In: *Laboratory Investigation* 100.8 (2020), pp. 1124–1136. ISSN: 0023-6837. DOI: https://doi.org/10.1038/s41374-020-0429-0. URL: https://www.sciencedirect.com/science/article/pii/S0023683722003798.

[45] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66. DOI: 10.1109/TSMC.1979.4310076.

[46] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[47] Ondrej Pös et al. "Circulating cell-free nucleic acids: characteristics and applications". In: *European Journal of Human Genetics* 26.7 (July 2018), pp. 937–945.

[48] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV].

[49] Jos B. T. M. Roerdink and Arnold Meijster. "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies". In: *Fundam. Informaticae* 41 (2000), pp. 187–228.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[51] Danny Salem et al. "YeastNet: Deep-Learning-Enabled Accurate Segmentation of Budding Yeast Cells in Bright-Field Microscopy". In: *Applied Sciences* 11.6 (2021). ISSN: 2076-3417. DOI: 10.3390/app11062692. URL: https://www.mdpi.com/2076-3417/11/6/2692.

[52] Russell Stewart and Mykhaylo Andriluka. *End-to-end people detection in crowded scenes*. 2015. arXiv: 1506.04878 [cs.CV].

[53] Carsen Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". In: *Nature Methods* 18.1 (Jan. 2021), pp. 100–106.

[54] Carole H Sudre et al. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations". en. In: *Deep Learn Med Image Anal Multimodal Learn Clin Decis Support (2017)* 2017 (Sept. 2017), pp. 240–248.

[55] Peize Sun et al. *What Makes for End-to-End Object Detection?* 2021. arXiv: 2012.05780 [cs.CV].

[56] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[57] Yan Tie et al. "Immunosuppressive cells in cancer: mechanisms and potential therapeutic targets". en. In: *J Hematol Oncol* 15.1 (May 2022), p. 61.

[58] Maxim Tkachenko et al. *Label Studio: Data labeling software*. Open source software available from https://github.com/heartexlabs/label-studio. 2020-2022. URL: https://github.com/heartexlabs/label-studio.

[59] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[60] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[61] Chien-Yao Wang et al. *CSPNet: A New Backbone that can Enhance Learning Capability of CNN*. 2019. arXiv: 1911.11929 [cs.CV].

[62] Gufeng Wang and Ning Fang. "Detecting and tracking nonfluorescent nanoparticle probes in live cells". en. In: *Methods Enzymol* 504 (2012), pp. 83–108.

[63] Xinlong Wang et al. *SOLO: Segmenting Objects by Locations*. 2020. arXiv: 1912.04488 [cs.CV].

[64] Xinlong Wang et al. *SOLOv2: Dynamic and Fast Instance Segmentation*. 2020. arXiv: 2003.10152 [cs.CV].

[65] Bing Xu et al. *Empirical Evaluation of Rectified Activations in Convolutional Network*. 2015. arXiv: 1505.00853 [cs.LG].

[66] Xinpeng Yang et al. "Ship Instance Segmentation Based on Rotated Bounding Boxes for SAR Images". In: *Remote Sensing* 15.5 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15051324. URL: https://www.mdpi.com/2072-4292/15/5/1324.

[67] Ze Yang et al. *Dense RepPoints: Representing Visual Objects with Dense Point Sets*. 2020. arXiv: 1912.11473 [cs.CV].

[68] Ma Yi-de, Liu Qing, and Qian Zhi-bai. "Automated image segmentation using improved PCNN model based on cross-entropy". In: *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.* 2004, pp. 743–746. DOI: 10.1109/ISIMP.2004.1434171.

[69] Sangdoo Yun et al. *CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features*. 2019. arXiv: 1905.04899 [cs.CV].

[70] Hongyi Zhang et al. *mixup: Beyond Empirical Risk Minimization*. 2018. arXiv: 1710.09412 [cs.LG].

[71] Lei Zhang, Qiuguang Wang, and Jiping Qi. "Research Based on Fuzzy Algorithm of Cancer Cells in Pleural Fluid Microscopic Images Recognition". In: *2006 International Conference on Intelligent Information Hiding and Multimedia*. 2006, pp. 211–214. DOI: 10.1109/IIH-MSP.2006.264982.

[72] Longhao Zhang and Huihua Yang. "Adaptive attention augmentor for weakly supervised object localization". In: *Neurocomputing* 454 (2021), pp. 474–482. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.05.024. URL: https://www.sciencedirect.com/science/article/pii/S092523122100761X.

[73] Bolei Zhou et al. *Learning Deep Features for Discriminative Localization*. 2015. arXiv: 1512.04150 [cs.CV].