Deep Learning-Based Particle Detection and Instance Segmentation for Microscopy Images

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des

Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

Tim Scherr, M.Sc.

Tag der mündlichen Prüfung:	26. September 2023
Hauptreferent:	apl. Prof. Dr.-Ing. Ralf Mikut
Korreferent:	Prof. Dr. Jan G. Korvink

Zusammenfassung

Bildgebende mikroskopische Verfahren ermöglichen Forschern, Einblicke in komplexe, bisher unverstandene Prozesse zu gewinnen. Um den Forschern den Weg zu neuen Erkenntnissen zu erleichtern, sind hoch-automatisierte, vielseitige, genaue, benutzerfreundliche und zuverlässige Methoden zur Partikeldetektion und Instanzsegmentierung erforderlich. Diese Methoden sollten insbesondere für unterschiedliche Bildgebungsbedingungen und Anwendungen geeignet sein, ohne dass Expertenwissen für Anpassungen erforderlich ist. Daher werden in dieser Arbeit eine neue auf Deep Learning basierende Methode zur Partikeldetektion und zwei auf Deep Learning basierende Methoden zur Instanzsegmentierung vorgestellt. Der Partikeldetektionsansatz verwendet einen von der Partikelgröße abhängigen Hochskalierungs-Schritt und ein U-Net Netzwerk für die semantische Segmentierung von Partikelmarkern. Nach der Validierung der Hochskalierung mit synthetisch erzeugten Daten wird die Partikeldetektionssoftware BeadNet vorgestellt. Die Ergebnisse auf einem Datensatz mit fluoreszierenden Latex-Kügelchen zeigen, dass BeadNet Partikel genauer als traditionelle Methoden detektieren kann. Die beiden neuen Instanzsegmentierungsmethoden verwenden ein U-Net Netzwerk mit zwei Decodern und werden für vier Objektarten und drei Mikroskopie-Bildgebungsverfahren evaluiert. Für die Evaluierung werden ein einzelner nicht balancierter Trainingsdatensatz und ein einzelner Satz von Postprocessing-Parametern verwendet. Danach wird die bessere Methode in der Cell Tracking Challenge weiter validiert, wobei mehrere Top-3-Platzierungen und für sechs Datensätze eine mit einem menschlichen Experten vergleichbare Leistung erreicht werden. Außerdem wird die neue Instanzsegmentierungssoftware microbeSEG vorgestellt. microbeSEG verwendet, analog zu BeadNet, OMERO für die Datenverwaltung und bietet Funktionen für die Erstellung von Trainingsdaten, das Trainieren von Modellen, die Modellevaluation und die Modellanwendung. Die qualitativen Anwendungen von BeadNet und microbeSEG zeigen, dass beide Tools eine genaue Auswertung vieler verschiedener Mikroskopie-Bilddaten ermöglichen. Abschließend gibt diese Dissertation einen Ausblick auf den Bedarf an weiteren Richtlinien für Bildanalyse-Wettbewerbe und Methodenvergleiche für eine zielgerichtete zukünftige Methodenentwicklung.

Abstract

Microscopy imaging enables researchers to gain insight into complex processes not understood so far. Reducing the time to insight for researchers requires highly automated, versatile, accurate, easy-to-use, and reliable particle detection and instance segmentation methods. In particular, such methods should perform well for different imaging conditions and applications without requiring expert knowledge for domain adaptation. Therefore, this thesis presents a new deep learning-based particle detection method and two deep learning-based instance segmentation methods. The particle detection approach uses a particle size-dependent upsampling pre-processing and a U-Net for the semantic segmentation of particle markers. After validating the upsampling pre-processing with synthetically created data, the corresponding particle detection tool BeadNet is introduced. Furthermore, results on a real-world fluorescent bead data set show that BeadNet can outperform traditional particle detection methods. The two novel instance segmentation methods use a double-decoder U-Net and are evaluated for four object types and three microscopy imaging techniques. A single unbalanced training data set and a single post-processing parameter set are used for this evaluation. The superior method is further validated in the Cell Tracking Challenge, reaching multiple top-3 rankings and human performance on six data sets. Furthermore, this thesis presents the new instance segmentation tool microbeSEG. Similar to BeadNet, microbeSEG uses OMERO for data management and provides training data creation, model training, model evaluation, and model application functionalities. The qualitative applications of BeadNet and microbeSEG show that these tools enable the accurate processing of many different microscopy image data. Finally, this thesis provides an outlook on the need for more guidelines for image analysis competitions and method comparisons for future method development.

Acknowledgments

This work would not have been possible in this form without the financial, intellectual, and interpersonal support of many people. First, a big thank you to my direct supervisor, apl. Prof. Dr.-Ing. Ralf Mikut for his guidance, interesting discussions (not only about work), engagement, and the possibility to work on exciting topics. For the latter, I also want to thank the head of the Institute for Automation and Applied Informatics, Prof. Dr.-Ing. Veit Hagenmeyer.

Prof. Dr. Jan G. Korvink, thank you for reviewing my dissertation. In addition, I am grateful for the feedback of my thesis advisory committee, namely Prof. Dr. G. Ulrich Nienhaus, Prof. Dr. Uwe Strähle, and apl. Prof. Dr.-Ing. Ralf Mikut.

I appreciate the funding from the BioInterfaces International Graduate School (BIF-IGS), the Helmholtz programs BioInterfaces in Technology and Medicine (BIFTM) and Natural, Artificial and Cognitive Information Processing (NACIP), and the Helmholtz Imaging Platform (HIP) in the project SATOMI.

I want to express my gratitude to all my project partners. In particular, I want to thank Dr. Karolin Streule and Prof. Dr. Véronique Orian-Rousseau for the pleasant cooperation that led to the development of BeadNet. Furthermore, I am happy that I had the possibility to part of the SATOMI team with Prof. Dr. Dietrich Kohlheyer, Dr. Katharina Nöh, Dr. Hanno Scharr, apl. Prof. Dr.-Ing. Ralf Mikut, Bastian Wollenhaupt, Johannes Seiffarth, and Oliver Neumann. It was a joy to work with you on the SATOMI project. I also want to thank the Cell Tracking Challenge organizers for running this great challenge for so many years, and Dr. Sahana Sheshachala for the cooperation on her segregation study. I am incredibly grateful for all the valuable contributions of my past Bachelor's and Master's students Philipp Hallgarten, Hongze Li, Shiye Xia, and Adelina Prokhorova. I also want to thank Verena Heusser for her work as a student assistant.

Before starting, I could not imagine meeting so many kind and down-to-earth colleagues in my working group. First, I want to thank "the old ones" that nicely welcomed me: Andy, Moritz, Benni, Simon W., Nicole, Ángel, Ralf, Christian, and Rennrad-Markus. Of course, I would also like to thank Marian, who started with me. Andy, my tea partner Moritz, Vojtech, Ines, and Sophie, it was nice and always fun to share my office with you. This was also the case when our external doctoral students Patricia, Simon B., and Richi were at the institute and found a place in my office. A special thanks to Katharina, who participated with me several times in the Cell Tracking Challenge. Furthermore, I want to thank *Mate* Jan, my Fleischkäsepartner Friedrich, Stefan, Ninja Warrior Kaleb, Matthias, my protégé Luca D., Luca R., Lisa, Lorenz, Nathalie, the publication machine Marcel, my successor as CIG CEO Oli, karting champion Roman, André, Mark, Lukas, Steff, Yanke, and Karl. I hope all of you enjoyed working with me as I have enjoyed working with you.

Many thanks to all my colleagues at the Institute for Automation and Applied Informatics beyond my working group. In particular, I want to thank Claudia Greceanu, Bernadette Lehmann, Stefan Vollmanshauser, Andreas Hofmann, and the scientific-technical infrastructure team. Furthermore, I thank Hannes, Dennis, Ina, and Firaz from the RWTH Aachen University for the interesting monthly deep learning meetings with us.

Finally, I want to thank my family and friends. Now that this part of my life has ended, I hope I have more time for you.

Contents

Zusammenfassung III

Abstract V

Acknowledgments VII

1 Introduction 1
- 1.1 Image Acquisition and Digital Image Processing . . . 2
  - 1.1.1 Digital Image Representation and Image Formation . . . 2
  - 1.1.2 Microscopy . . . 3
  - 1.1.3 Neighborhood Operations . . . 3
  - 1.1.4 Deep Learning . . . 4
- 1.2 Particle Detection . . . 8
  - 1.2.1 Traditional Methods . . . 9
  - 1.2.2 Deep Learning Methods . . . 9
  - 1.2.3 Benchmark and Training Data Sets . . . 10
  - 1.2.4 Available Software Solutions . . . 10
- 1.3 Instance Segmentation . . . 10
  - 1.3.1 Traditional Methods . . . 10
  - 1.3.2 Deep Learning Methods . . . 11
  - 1.3.3 Benchmark and Training Data Sets . . . 14
  - 1.3.4 Available Software Solutions . . . 15
- 1.4 Evaluation Metrics . . . 15
  - 1.4.1 Precision, Recall and F-Score . . . 15
  - 1.4.2 Normalized Acyclic Oriented Graph Matching Measure for Detection . . . 16
  - 1.4.3 Jaccard Similarity Index and Aggregated Jaccard Index . . . 17
- 1.5 Open Questions . . . 17
- 1.6 Objectives and Thesis Outline . . . 18

2 Particle Detection 21
- 2.1 U-Net-Based Semantic Segmentation of Particle Markers . . . 21
  - 2.1.1 Upsampling Pre-processing for Improved Particle Separation . . . 21
  - 2.1.2 Marker Representation . . . 22
  - 2.1.3 CNN Architecture . . . 23
  - 2.1.4 Training Process . . . 25
  - 2.1.5 Inference and Post-Processing . . . 26
- 2.2 Validation . . . 26
  - 2.2.1 Compared Upsampling Strategies and Marker Representations . . . 27
  - 2.2.2 Training and Test Data . . . 27
  - 2.2.3 Results . . . 29

2.3 Software: BeadNet . . . 33
2.3.1 OMERO Data Management . . . 35
2.3.2 Training Data Creation . . . 36
2.3.3 Model Training and Evaluation . . . 36
2.3.4 Inference and Result Export . . . 37
2.3.5 Implementation, Installation, and Dependencies . . . 37
2.3.6 Workflow Evaluation . . . 38
2.4 Applications . . . 38
2.4.1 Counting of Cells and Cell Nuclei . . . 40
2.4.2 Nanoparticle Detection . . . 40
2.4.3 mRNA Localization . . . 40
2.4.4 Bead Detection . . . 40
2.5 Discussion . . . 43

3 Instance Segmentation 45
3.1 Double-Decoder U-Net-Based Instance Segmentation . . . 45
3.1.1 Robust Encoding of Neighbor Information . . . 45
3.1.2 CNN Architecture . . . 49
3.1.3 Training Process . . . 50
3.1.4 Inference and Post-Processing . . . 52
3.1.5 Code Availability . . . 54
3.2 Validation . . . 54
3.2.1 Training Data Set and Test Data Sets . . . 54
3.2.2 Compared Methods . . . 56
3.2.3 Results . . . 59
3.3 Software: microbeSEG . . . 76
3.3.1 OMERO Data Management . . . 77
3.3.2 Training Data Creation . . . 77
3.3.3 Model Training and Evaluation . . . 78
3.3.4 Inference, Result Analysis, and Result Export . . . 80
3.3.5 Implementation, Installation, and Dependencies . . . 80
3.3.6 Workflow Evaluation . . . 80
3.3.7 Key Feature Comparison . . . 83
3.4 Applications . . . 83
3.4.1 Cell and Cell Nucleus Segmentation . . . 85
3.4.2 Nanoparticle Characterization . . . 86
3.4.3 Fiber Detection . . . 86
3.4.4 Cell Tracking Challenge . . . 88
3.5 Discussion . . . 92

4 Conclusion and Outlook 97
4.1 Conclusion . . . 97
4.2 Outlook . . . 98

A Statistical Significance 101

B microbeSEG Overview 109

Abbreviations **111**

List of Own Publications **113**

References **115**

1 Introduction

State-of-the-art microscopy imaging techniques and microfluidic lab-on-chip systems with highly precise environmental control enable researchers from life sciences and engineering to gain insight into ever more complex processes that are not understood so far [1]–[7]. Objects need to be individually detected or segmented to extract quantities like object counts or size distributions from the acquired images. Due to the large amount of data and objects, manual analysis is inefficient and often infeasible. In addition, low-resolution and low-contrast objects can be challenging to detect, even for human experts. Thus, highly automated, versatile, accurate, and reliable object detection and instance segmentation methods are required to reduce the time to insight. However, virtually error-free analysis of acquired microscopy images remains a challenge. Contrast and signal-to-noise ratio can be low due to lighting and technical limitations, such as avoiding phototoxic stress in biotechnological applications or tradeoffs between frame rate, exposure time, and spatial resolution. Furthermore, adjacent objects can have unclear boundaries, and domain gaps exist between and within experimental data sets.

In recent years, supervised deep learning methods have outperformed traditional methods in many image processing tasks [8]–[12]. For training such supervised methods, annotated data sets are required. While many annotated data sets exist for tasks like driver assistance and autonomous driving [13]–[15] or object detection in everyday scenes [8], [16], the diversity of microscopy imaging and sample preparation techniques and task intrinsic variations make it challenging to find appropriate training data sets, e.g., for different cell genotypes and phenotypes with different morphologies and behavior in industrial biotechnological processes. Consequently, there is a demand for newly annotated task- and domain-specific training data. However, the annotation of training data is time-consuming. Thus, deep learning methods are needed that do not require extensive training data sets and can learn to distinguish adjacent objects from only a few adjacent examples. In addition, these methods should be able to deal with various image domains, such as fluorescence and phase contrast microscopy images.

The key advantages of deep learning-based methods over traditional image processing methods are that they are more versatile and do not require expert knowledge to adapt to a particular application. In most cases, a researcher only needs to create new training data for adaptations to new imaging or experimental settings. In contrast, adapting sophisticated task-specific traditional image processing pipelines requires expert knowledge of the underlying method and, in some cases, code adaptations. Thus, even for experts, it can be more efficient to create new training data and train a deep learning-based method, given that a comprehensive tool covering data management, training data creation, model training, and model application exists. So far, there is a lack of such complete tools for object detection and instance segmentation of microscopy images. However, the research on deep learning and its applications is progressing rapidly with many parallel developments.

This thesis aimed to leverage supervised deep learning for efficient particle detection and instance segmentation in microscopy images. In this context, efficient means (i) that small training data sets are sufficient to train a deep learning model with reasonable quality, i.e., less than one hour

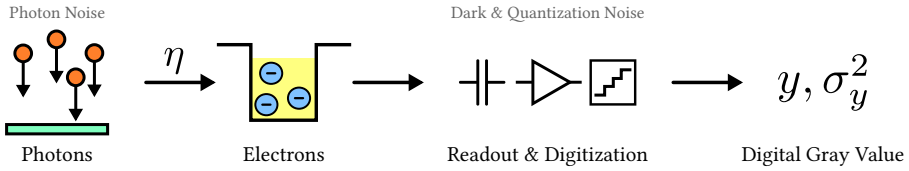


Figure 1.1: Simple Image Sensor Model. Poisson-distributed photons hit the pixel area during the exposure time and release electrons (quantum efficiency η). The charges are stored in a potential well and, during readout, converted by a charge amplifier into a voltage. Finally, the signal is digitized. In the linear signal model, the variances σ^2 of all noise sources add up linearly [18]. Illustration inspired by [18].

of annotation time should be needed, (ii) that the deep learning method is versatile and works for several domains, i.e., a single trained model should be able to accurately process multiple object types and shapes and data acquired with different microscopy techniques, and (iii) that the method needs almost no hyperparameter or post-processing fine-tuning, which enables to build easy-to-use tools. The developed methods were successfully applied to particle detection and cell segmentation. Moreover, the particle detection method has been validated on a newly created bead data set, whereas the instance segmentation method has been validated in the Cell Segmentation Benchmark (CSB) and the Cell Tracking Benchmark (CTB) of the Cell Tracking Challenge (CTC). In addition, both methods are integrated into new tools for efficient microscopy image analysis. The comprehensive workflow of the instance segmentation tool has been examined on microbe data. Though biomedical and biotechnological microscopy data have mostly been used for method development in this thesis as these data have extrinsic and intrinsic variations and are challenging to process, the presented methods and tools are not limited to this research area and can also be applied, for instance, to material science data.

This chapter presents the theoretical background and related work, i.e., image acquisition and digital image processing fundamentals, particle detection methods, and instance segmentation methods. Furthermore, it is referred to existing software solutions and the open research questions on which this thesis is based are formulated.

1.1 Image Acquisition and Digital Image Processing

1.1.1 Digital Image Representation and Image Formation

During image formation, the 3D object space is projected onto a 2D image plane – technically, in visual systems, the 3D world is reconstructed from 2D images. An image represents the spatial distribution of the irradiance at this plane [17]. Digital images are stored as matrices, and an element of such a matrix is called a pixel for 2D images. Figure 1.1 shows a simple model of an image sensor. Light, or more exact photons, hits the pixel area of an image sensor and releases several electrons (photoelectric effect). The charges are stored and converted into a voltage that is digitized. In this process, the quantization maps the measured irradiance to a finite number of discrete gray values. Digitization also means sampling the image at selected positions on a discrete grid. Thus, the gray value of a pixel is the integral over the area of a pixel. The sampling theorem states that sampling loses no information if no wavenumber is larger than the Nyquist wavenumber [17]. There are several noise sources in the image formation process, e.g., Poisson distributed photon noise, dark noise, or quantization noise. Usually, the signal-to-noise ratio (SNR) increases with higher image intensities.

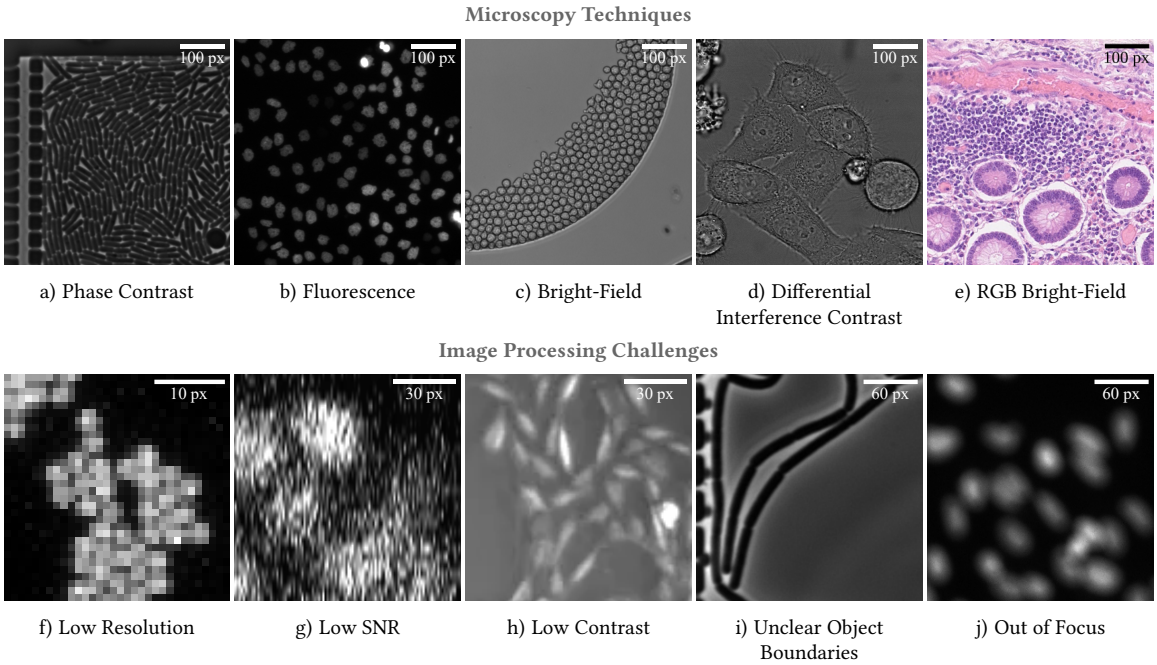


Figure 1.2: Overview of Microscopy Image Techniques and Image Processing Challenges. Data sources: a – Forschungszentrum Jülich GmbH, IBG-1; b, c, d, g, h – Cell Tracking Challenge [27], [28], e – Lizard data set [29], f – BeadNet [30], i – microbeSEG data set [31], j – BBBC006v1 [32].

1.1.2 Microscopy

Microscopy deals with imaging objects and structures typically below the resolution limit of the human eye. The theoretical resolution limit of a microscope depends on the wavelength of the radiation used. Special microscopy imaging techniques [19]–[22], fluorescent tags [23], [24], or staining methods [25], [26] need to be applied to improve the contrast, e.g., for the imaging of colorless and transparent specimens and structures or to highlight structures of interest. Figure 1.2 shows exemplary images of biological cell data acquired with phase contrast microscopy (PhC, Figure 1.2a), fluorescence microscopy (Fluo, Figure 1.2b), bright-field microscopy (BF, Figure 1.2c), differential interference contrast microscopy (DIC, Figure 1.2d), and multi-channel bright-field microscopy (1.2e).

Technical limitations and tradeoffs, e.g., between spatial resolution, temporal resolution (imaging speed), exposure (SNR), and phototoxicity in live cell imaging [21], [33], [34], have a huge impact on the image acquisition and image quality and complicate automated image processing of acquired images. Low object resolution (Figure 1.2f), low SNR (Figure 1.2g), low contrast (Figure 1.2h), adjacent objects with unclear boundaries (Figure 1.2i), and focus drift (Figure 1.2j) are, therefore, common challenges in detection and segmentation tasks involving microscopy images. In addition, the diversity of the imaging, staining, and fluorescent tagging techniques and the variety of object shapes, textures, and sizes complicate image processing further.

1.1.3 Neighborhood Operations

Analyzing the gray values of an image in a small neighborhood can be useful for tasks like object recognition [17]. For instance, gray value changes are usually small within a uniform object and large at the object-background transition. Neighborhood operations combine pixels locally and

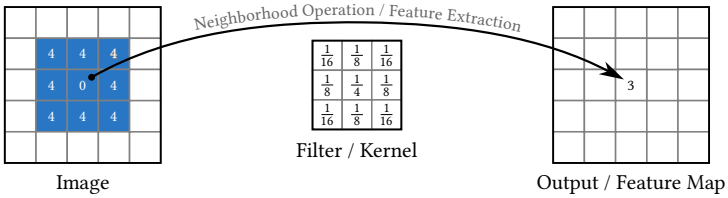


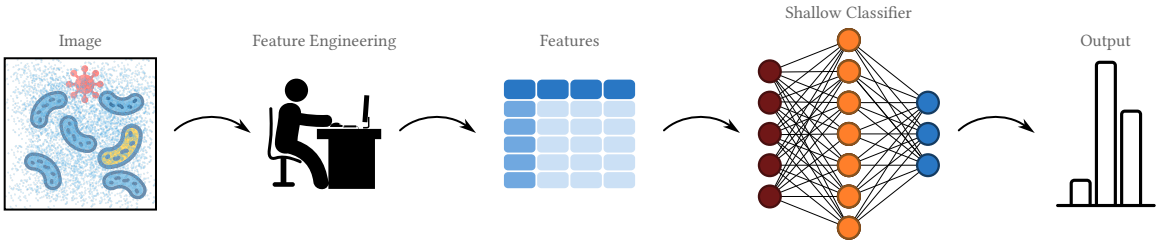
Figure 1.3: Neighborhood Operation with a Smoothing Filter. An illustrative way to apply a filter is to place it on top of the image, multiply the overlapping values, and sum up the products. The feature map can then be calculated by moving the filter over the image (mathematically a convolution with a filter/kernel). The shown smoothing filter is a binomial filter [17]. Pixels contributing to the indicated output pixel are highlighted in blue.

extract certain features of interest from an image or filter information (see Figure 1.3). The result is called a feature image or feature map, and the operators are also called filters. Images need to be extended periodically, padded with constant values, or extrapolated for using a filter at the image boundaries if the feature map size should be equal to the image size. Among others, there are smoothing filters, e.g., Gaussian filters to suppress Gaussian noise, rank value filters, e.g., median filters for filtering outliers, and filters for computing gradient estimates. For higher-order filtering, the filters are applied several times. Multiple filters usually need to be applied subsequently for extracting high-level features like objects in an image, e.g., Gaussian filters, edge detection filters, and Gabor filters for texture analysis.

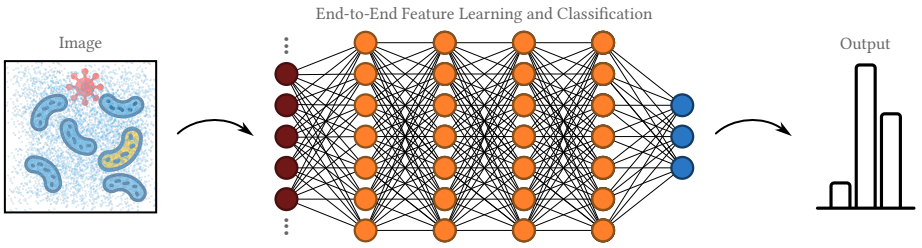
1.1.4 Deep Learning

In traditional image processing and computer vision, hand-crafted feature engineering approaches dominated. Back then, tasks such as image classification usually required a processing step called feature extraction – a complex and application-specific task since it is necessary to define which features are important in each given image for each class [35]. The extracted features and the corresponding ground truth class labels were then used to train a classifier, e.g., shallow neural networks [36], support vector machines [37], or random forests [38]. In contrast, deep neural networks with more than one hidden layer between input and output learn the feature extraction and the classification end-to-end directly from the images and the class labels (see Figure 1.4). This deep learning approach can result in a higher accuracy due to the use of more complex and significant features but generally requires a higher computational effort and more training data, i.e., pairs of images and ground truths like class labels for image classification. Besides a better accuracy than traditional methods in many image processing tasks, e.g., image classification and object detection [8], [39], semantic segmentation [40], [41], instance segmentation [42], [43], surface defect detection [12], [44], and image restoration [11], [45], a further advantage to traditional methods is that less domain-specific expertise is required since no hand-crafted feature definitions are used.

Even though trained artificial neural networks have been around since the 1960s [46], deep learning in computer vision gained significant attention mainly with the success of the deep neural network AlexNet [47] in the ImageNet Large Scale Visual Recognition Challenge for image classification and object detection [8]. Crucial to the (continued) success of deep learning are advances made in hardware technologies and high-performance computing, a better understanding of the training process, deep neural network architectures, and the availability of (annotated) data [48]–[51]. Deep learning overviews are provided in [46], [52]–[54], and a comprehensive introduction can be found in [55].



a) Traditional Computer Vision Workflow for Image Classification



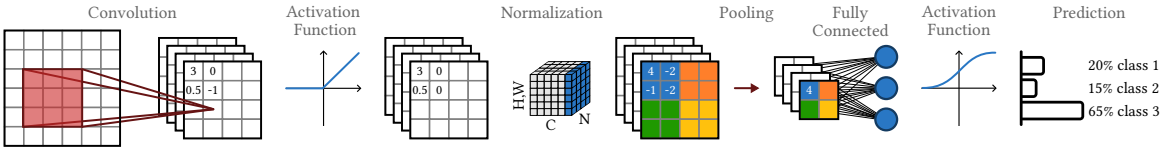
b) Deep Learning Workflow for Image Classification

Figure 1.4: Traditional and Deep Learning Workflow for Image Classification. Traditional computer vision typically requires a feature engineering step before training a classifier, such as shallow neural networks or support vector machines (a). The features can, for instance, rely on the image intensity distribution or texture. Exemplary output classes for a biomedical application are healthy, cancer, and infectious disease, while defect classes are used in industrial surface inspection processes. Deep neural networks are trained end-to-end and do not rely on manual feature engineering (b). This end-to-end learning allows the utilization of more complex features but requires a higher computational effort. Note: deep neural networks with only fully connected layers are usually not used since they are difficult to train. Illustration inspired by [35]. ● – input neuron (one for each feature/pixel), ●/● – hidden/output neuron with learnable weights.

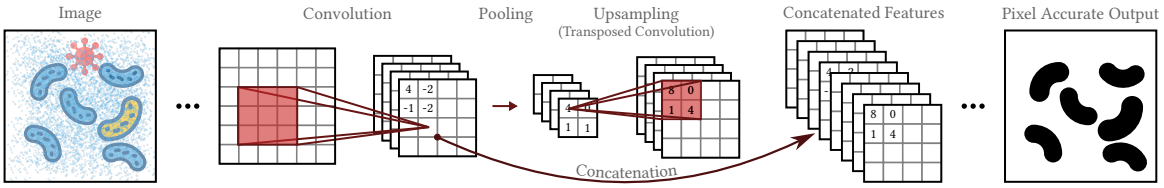
Convolutional Neural Networks

Figure 1.5a shows common components of a convolutional neural network (CNN) for image classification: (i) convolutional layers with learnable kernels for feature extraction, (ii) activation functions to scale features and to prevent linearity enabling the learning of non-linear functions, (iii) normalization layers to stabilize the training, (iv) pooling layers for dimensionality reduction, and (v) fully connected layers, which are mainly used as output layers performing high-level reasoning [49]. Usually, a CNN consists of multiple blocks of convolutional layers, activation functions, normalization layers, and pooling layers. As a result, features are aggregated at different scales, enabling the encoding of more abstract features in the deeper layers [49]. In addition, the pooling reduces the number of connections between convolutional layers.

A convolutional layer can be interpreted as a collection of learned neighborhood operations for feature extraction. Therefore, multiple learnable kernels are used within a layer. New feature maps are obtained by convolving the outputs from the previous layer with the kernels of the current layer. Hence, each kernel goes over the whole channel dimension. Thus, speaking of a 1×1 convolution means a 2D convolution with a $1 \times 1 \times C$ kernel, where C denotes the number of channels (feature maps). An advantage to fully connected layers, which connect all neurons in the previous layer to every single neuron of the current layer, is that each neuron of a feature map is only connected to a region of neighboring neurons in the previous layer. In addition, weights are shared. Together with the dimensionality reduction in pooling layers, this leads to a significant reduction of parameters. For instance, 124 million of the 138 million parameters of the CNN VGG-16 [56] are located in the 3 output fully connected layers, while 14.7 million weights are in the 16 convolutional layers [57]. The reduced model complexity makes CNNs easier to train than fully connected networks.



a) Common Image Classification CNN Components with Intermediate Results and Prediction



b) Upsampling and Concatenation Enable Pixel Accurate Segmentation Outputs

Figure 1.5: Common CNN Components. CNNs use convolutional layers that rely on learning neighborhood operations (see Figure 1.3). Usually, multiple convolutional and pooling layers are used for subsequent feature extraction and dimensionality reduction. Activation functions are applied to scale features and to prevent linearity, while normalization layers help to stabilize the training. In contrast to image classification (a), pixel-wise predictions are needed for segmentation. Therefore, upsampling layers can be used. The concatenation of upsampled feature maps and further convolutional layers allows the combining of features from different scales without losing pixel accuracy (b). Note: shown convolution inputs are padded; batch normalization and maximum pooling are shown in (a); no activation functions and normalization layer are shown in (b). – feature map, – learnable kernel, H/W/C/N – height/width/channel/batch dimension, – normalization layer, – pooling, – output neuron with learnable weights.

When designing a CNN architecture, not only the number and the arrangement of the single layers need to be specified but also some layer-specific parameters: kernel size, number of kernels, stride, and input padding for each convolutional layer, activation functions for each layer, e.g., the rectified linear unit activation function (ReLU), the normalization layer type like batch normalization [58], and the pooling layer type, e.g., maximum pooling or learnable 2×2 convolutions with stride two. A summary of CNN components is provided in [49]. Furthermore, pixel-wise predictions are needed for image segmentation. One way to restore the initial spatial resolution of the input image after the subsequent feature extraction and dimensionality reduction is to use upsampling layers. Common choices are max unpooling [59], traditional upsampling methods, or learnable transposed convolutions [49]. Often, further convolutional layers are applied after upsampling layers for feature refinement. Some segmentation architectures use skip connections that concatenate corresponding feature maps to retain the positional information lost during downsampling [60] (see Figure 1.5b).

The receptive field of a CNN is defined as the input region which can contribute to an output or output pixel and depends on the number and type of convolutional, pooling, and unpooling layers. Usually, the objects to segment should be smaller than the receptive field so that the CNN can learn appropriate features.

U-Net Architecture

The U-Net architecture is an encoder-decoder CNN architecture for image segmentation and consists of convolutional, pooling, and upsampling layers [61]. In addition, skip connections with feature map concatenation are used to retain positional information. Figure 1.6 shows the U-Net architecture, a common CNN architecture for analyzing microscopy images [62], [63]. Due to its success, many U-Net derivates have been developed, e.g., UNet++, which combines U-Nets of varying depths into one unified architecture and uses redesigned nested and dense skip connections [64], UNet 3+ that uses full-scale skip connections and deep supervision [65], the recurrent residual CNN R2U-Net [66],

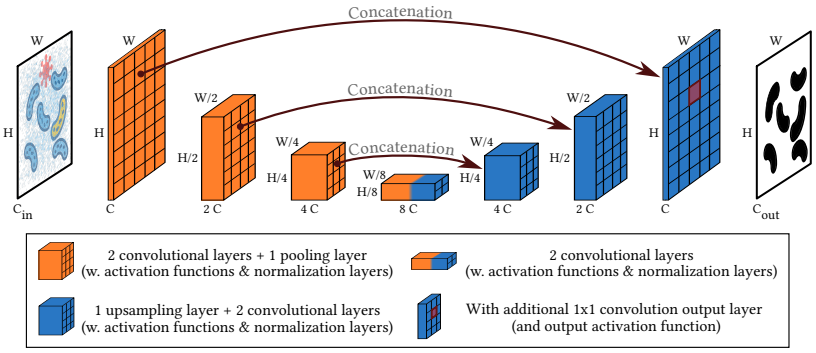


Figure 1.6: U-Net Architecture. The U-Net architecture is an encoder-decoder CNN with upsampling layers and feature map concatenation. Note: concatenated are the outputs of the last convolutional layer in a block and of the corresponding transposed convolutional layer. H/W/C – height/width/channel dimension.

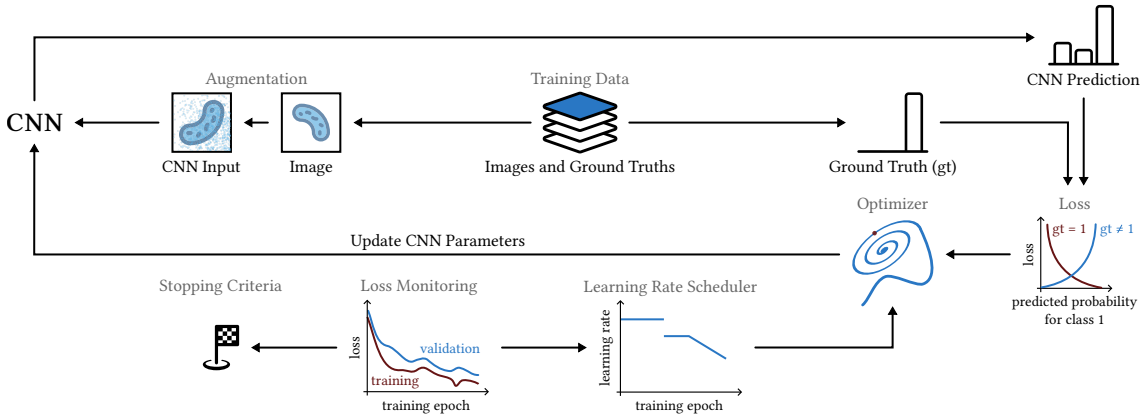


Figure 1.7: Training Process Components. Besides the CNN architecture design and training data quality and quantity, the training process plays an important role in achieving high-quality results. Thus, augmentations (shown: rotation, scaling, noise), the loss function, the optimizer, learning rate schedulers, batch size, and stopping criteria need to be selected and adjusted carefully. Note: for segmentation, the ground truths may also need to be transformed, e.g., for scaling augmentations.

two stacked U-Nets called DoubleU-Net [67], and more [63]. However, standard U-Net architectures can still produce state-of-the-art results [41].

Training Process

Figure 1.7 provides an overview of the training process components and their interactions. First, training data are needed, i.e., images and corresponding ground truths. For image classification, the ground truths are class labels. For image segmentation, the ground truths are segmentation masks. Often, the ground truths are manually or semi-automatically annotated, which means that the ground truth may not be error-free since intra-annotator and inter-annotator variability exist, which may affect the training [68], [69]. The training data are split into a training subset for model parameter update estimation and a validation subset for an unbiased evaluation of the model.

Data augmentation is applied to enhance the training data set and to increase the generalization performance of a trained CNN model [70]. Image augmentation techniques include photometric and contrast transformations, blurring, noise, and geometric transformations, such as rotation, scaling, and perspective transformations. Augmentation in the feature space is also possible.

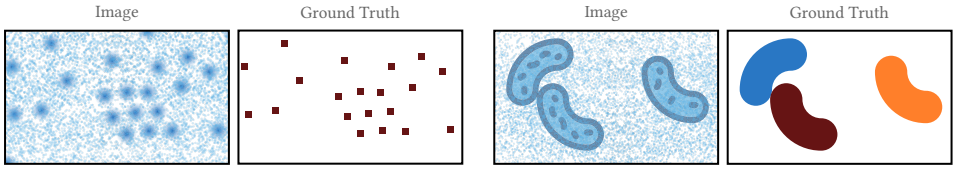


Figure 1.8: Particle Detection and Instance Segmentation. For particle detection, estimating a localization marker representing a single particle is sufficient (a). In contrast, instance segmentation provides full shape and positional information (b, instances are color-coded). Particle detection has the advantage that ground truths can be annotated quicker, while instance segmentation can deliver more information for evaluation. Thus, the choice of particle detection or instance segmentation is application-specific. Especially if uniform, poor-resolved particles of known size need to be counted, particle detection can be more efficient than instance segmentation.

For segmentation, some augmentations, e.g., rotation, require a similar transformation of the corresponding ground truth. Albumentations is a Python library that implements a variety of image transform operations [71].

After augmentation, a batch of augmented images is fed into the network, which yields the predictions. A suitable loss function is needed to measure the deviation of the prediction to the ground truth. This loss function needs to be selected carefully and depends on the task and the data representation. A survey of loss functions is provided in [72] and [73]. The loss function must be differentiable for the optimization process based on error backpropagation [74] and gradient descent [75]. This optimization process is challenging since the loss landscape can be non-convex [76]. Common optimizers are Adam [77], SGD [75], and Lookahead [78]. Hyperparameters such as the learning rate and momentum need to be set and adjusted for successful training [79].

Training and validation loss monitoring is another crucial part of the training process. Diverging training and validation losses can indicate overfitting and poor generalization ability. Furthermore, too large learning rates may result in validation loss plateaus and a need for decreasing the learning rate. Sub-optimal local minima can be escaped by increasing the learning rate, e.g., using cyclical learning rates [80]. Thus, learning rate schedulers that adapt the learning rate based on the validation loss improvement are helpful. Finally, stopping criteria for finishing the training process need to be defined. Simple criteria are fixing the number of training epochs or stopping training after a certain amount of epochs without validation loss improvements.

Inference

Inference is the process of applying a trained model and making predictions on previously unseen data. Techniques like test-time augmentation and model ensembling combine multiple predictions and can increase the accuracy [81] or quantify uncertainty [82].

1.2 Particle Detection

Particle detection is the task of localizing every single particle in an image. Exact shape information is not necessarily provided (see Figure 1.8). According to [83], a particle may be anything from a single molecule to a macromolecular complex, organelle, virus, or macrosphere. Particle detection is, for instance, needed for counting ligand-coupled beads that can be used to study bacterial invasion processes [30], [84], for studying nanoparticle ordering effects [85], or as a step for particle tracking [83]. In this thesis, particle detection refers to the detection of any object of interest in

a microscopy image with a marker that should be located near a particle's center. Typically, the particles are not well-resolved, and challenges are low SNR, small particle size (see Figure 1.2f), and the lack of prominent shape and texture characteristics [86]. Sometimes, particle detection is referred to as spot detection [87] or particle localization [88].

1.2.1 Traditional Methods

Traditional particle detection methods range from thresholding and local maxima detection methods to band-pass filtering, kernel density estimation, model fitting, and feature point detection methods [83], [87], [89]. A simple way is Otsu thresholding [90] with subsequent centroid extraction, e.g., the solution of Winter and Cohen for the Particle Tracking Challenge [83]. Circular particles can be detected with the Hough transform [91], [92]. Olivo-Marin (2002) extracts spots using a wavelet transform and filtering of wavelet coefficients [93]. Sbalzarini and Koumoutsakos (2005) detect local maxima by selecting pixels with constant values before and after grayscale dilation [94]. Byun et al. (2006) apply template matching with an inverted Laplacian of Gaussian blob model [95]. Mueller et al. (2013) fit Gaussians to fluorescent spots [96]. In [97], local maxima are detected after Laplacian of Gaussian filtering, while in [98] top-hat filtering is used. Wilson et al. (2016) use multiscale Haar-like features to construct a particle probability image and soft thresholding [99].

1.2.2 Deep Learning Methods

Deep learning methods for particle detection usually rely on pixel-wise classification, e.g., into the classes background and particle center. However, the direct regression of centroid coordinates is another possible approach. Newby et al. (2018) use a CNN with four layers trained with a cross-entropy loss for pixel-wise classification with the two classes background and particle center region [88]. The particle center region corresponds to a 3×3 px marker at the particle center. The used CNN is based on [100] and utilizes a single bilinear upsampling output layer to restore the original image resolution. Ito et al. (2018) use a CNN with four layers and no pooling and upsampling layers [101]. For training, a negative log-likelihood loss is used to predict the classes background and particle, making this method more of a semantic particle segmentation approach than a pure detection approach. The size of the result is reduced since no padding is applied in the convolutional layers. In [86], the hourglass-shaped CNN DetNet is proposed. The architecture of DetNet uses residual blocks [102] and has a pooling and an upsampling path like the U-Net but no skip connections. Bilinear upsampling is applied for upsampling, and the soft Dice loss is used to train DetNet to distinguish background and particle centers.

In contrast to the methods mentioned above, DeepTrack predicts a particle's coordinates and radial distance to the image center [60], [103]. DeepTrack consists of three convolutional layers with subsequent pooling layers and two fully connected layers. A drawback is that only one particle in an image – the most central particle – is detected. Thus, images showing multiple particles need to be divided into overlapping crops processed separately. DeepTrack is trained to return a large radial distance if no particle is present, and the mean absolute error is used as the loss function. The authors claim that millions of particles are needed for training.

Proposal-based object detection methods that predict bounding boxes, e.g., faster R-CNN [104], can also be used for particle detection. However, bounding box annotation is more time-consuming than marker annotation. In addition, proposal-based methods may fail for densely distributed particles, which is often the case for microscopy images since bounding boxes of adjacent instances

can be suppressed by the non-maximum suppression operation [105], [106]. Detecting small objects may be an issue as well [107].

1.2.3 Benchmark and Training Data Sets

Benchmark data sets with ground truths are needed for objective method comparison and can also be used to train deep learning-based particle detection methods. Therefore, the Particle Tracking Challenge provides 48 simulated data sets with different particle densities and SNRs [83]. The Yeast Image Toolkit benchmark features another 10 data sets [108]. Furthermore, the data set BBBC054 from the Broad Bioimage Bioimage Benchmark Collection showing microscopy images of different cell phenotypes can be used [32]. A conversion of the instance segmentation masks into particle markers enables using the instance segmentation benchmark data sets in Section 1.3 for particle detection. All mentioned data sets are freely available.

1.2.4 Available Software Solutions

Particle detection tools based on traditional methods are the Fiji plugin SpotCaliper [109], [110], which uses the wavelet-based method from [93] for the detection of circular particles, the Icy plugin Spot Detector [111], [112], which is also based on [93], ComDet based on Gaussian and Mexican hat filtering and thresholding [113], the Fiji plugin Particle Tracker [114] based on [94], and FISH-quant for smFISH image analysis [96], [115]. DeepTrack 2.0 is a deep learning-based tool for particle detection and tracking [60].

1.3 Instance Segmentation

As shown in Figure 1.8b, instance segmentation delivers shape and positional information of every object in an image. Microscopy image instance segmentation applications include, among others, biomedical and biotechnological cell, cell nuclei, and gland instance segmentation [116], [117], segmentation of metal powder satellites for material characterization [118], and the estimation of particle size distributions in nanomaterial development [119]. Common challenges, especially for the separation of adjacent objects, are low SNR (see Figure 1.2g), low contrast (see Figure 1.2h), and unclear object boundaries (see Figure 1.2i). The combination of instance segmentation and object classification is called panoptic segmentation [120].

1.3.1 Traditional Methods

In [116], traditional cell segmentation methods are categorized into intensity thresholding, feature detection, morphological filtering, region accumulation, and deformable model fitting approaches. In [121], threshold-based, edge-based and region-based methods are distinguished, while in [28] the categories thresholding, energy minimization, and region growing are used. However, most methods are composed of several steps and approaches, like the majority of reviewed methods in [116].

For example, Chen et al. (2006) proposed an intensity thresholding- and region accumulation-based segmentation method [122]. Simple Otsu thresholding [90] is applied for global image thresholding. For distinguishing adjacent objects, a watershed transform is used [123]. However, since watershed techniques can be prone to oversegmentation [116], [124], a further merging post-processing using a priori object size information is required. Al-Kofahi et al. (2010) combine

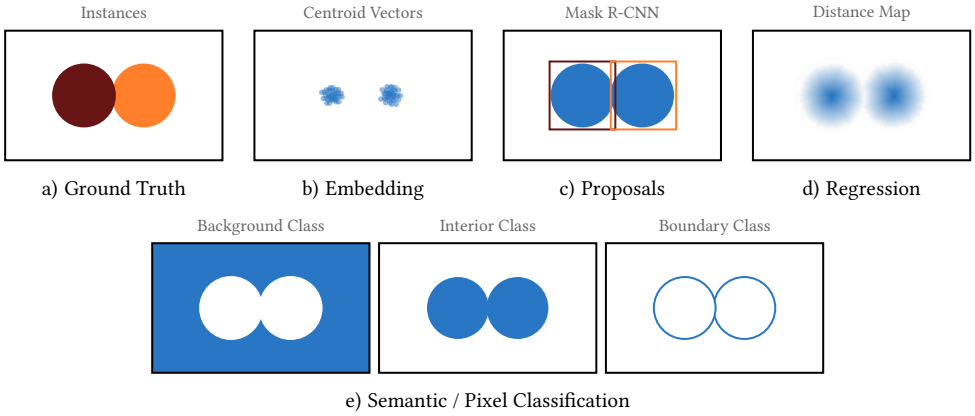


Figure 1.9: Categorization of Deep Learning Instance Segmentation Methods. The ground truth (a) needs to be converted into a data representation that can be learned from a deep learning model. Embedding methods learn an embedding that yields similar values for pixels of the same instance while pixels of different objects have distant values, e.g. vectors to the corresponding object center of a pixel (b). Instances are recovered by clustering. Proposal methods rely on the prediction of object proposals, e.g., bounding boxes (c). The regression of a continuous object representation like distance maps is another possibility (d). Semantic methods classify each pixel into classes, enabling the recovery of single instances (e). Combinations of the single categories are possible. Illustration inspired by [62].

graph-cuts-based binarization, seed point detection by multiscale Laplacian of Gaussian filtering with distance-map-based adaptive scale selection, automatic graph coloring, and α-expansion [125]. Stegmaier et al. (2014) propose the algorithm TWANG that also uses a multiscale Laplacian of Gaussian blob detector [126]. The extracted seeds are used to transform the image into a representation that can be handled by thresholding. Therefore, the weighted dot product of seed normal vectors and smoothed gradients is used. More recent traditional methods are the Hessian blob algorithm [127] and the robust optical flow algorithm [128]. A review and comprehensive comparison of segmentation methods for label-free contrast microscopy provides [129].

1.3.2 Deep Learning Methods

This thesis categorizes deep learning methods for instance segmentation into the categories embedding method (EM), proposal-based method (PM), regression method (RM), and semantic method (SM). Combinations of the single categories are possible. Figure 1.9 illustrates the categories, and Table 1.1 provides a categorization and short method descriptions of 48 instance segmentation methods selected by methodological novelty, impact, and segmentation competition success. Mainly methods applied to microscopy data are considered, and no works on CNN architecture improvements or a focus on loss function adaptations are included to keep this section concise.

Semantic Methods

The most intuitive method category is the semantic or pixel classification method category. Semantic methods classify each pixel of an input image, e.g., into the classes background, object interior, and object boundary (see Figure 1.9e). The post-processing obtains the instances from the semantic per-pixel classification and typically consists of thresholding strategies, maximum a posteriori estimation, or a watershed transform [130]. Therefore, the classes need to be designed in a way that robustly allows the separation of adjacent instances. Ronneberger et al. (2015) use only two classes – foreground and background – but introduce small separations between adjacent objects [61].

In contrast, Peña et al. (2020) utilize the four classes background, object, border between adjacent objects, and gaps between adjacent objects [131].

Embedding Methods

The goal of embedding methods is to learn a representation or embedding where pixels of the same instance have similar values while pixels of different instances have distant values (see Figure 1.9b). Thus, pixel embeddings from the same instance define a cluster, and the instances can be recovered with clustering post-processing. Brabandere et al. (2017) use an n-dimensional feature space and a discriminative loss function that pushes cluster centers away from each other and pulls embeddings toward their corresponding cluster center [132], [133]. Neven et al. (2019) use centroid vector embeddings that point to corresponding instance centroids [134]–[136]. Instead of forcing the CNN to predict the centroid for each pixel of an instance exactly, the proposed hinge loss variant forces pixels to lay within a specific margin around the centroid. This margin is learnable, allowing smaller margins for smaller objects and larger margins for larger objects.

Proposal-Based Methods

A characteristic of proposal-based methods is that multiple regions of interest, e.g., bounding box proposals, are determined, which are often refined or further processed to get the object shape (see Figure 1.9c). Typically, an object proposal redundancy reduction step like non-maximum suppression is required. Many proposal-based instance segmentation methods strongly relate to object detection methods like Faster R-CNN [104] and YOLO [137], which predict bounding box proposals, confidence scores, and class probabilities. Mask R-CNN expands Faster R-CNN for instance segmentation by adding a segmentation branch that outputs a binary mask for each bounding box [42]. Schmidt et al. (2018) use star-convex polygon proposals, which provide a better shape representation than bounding boxes and do not need shape refinement [138].

Regression Methods

In contrast to semantic methods with a discrete object representation into classes, regression methods learn a continuous object representation like topological maps in which an object forms a single smooth topological basin (see Figure 1.9d). In [139], distance map regression is combined with watershed post-processing. Stringer et al. (2021) use heat-diffusion-based topological maps and gradient tracking post-processing in their method called Cellpose [140].

Table 1.1: Overview of Deep Learning Methods with a Focus on Microscopy Image Instance Segmentation.

Method/Authors[†]	Categories	Description
Ronneberger et al. (2015) [61]	SM	Original U-Net; 2-class foreground-background SM with small separations between adjacent objects for instance segmentation; data: cells (DIC, PhC).
Van Valen et al. (2016) [141]	SM	Patch-based 3-class interior-boundary-background SM; data: cells (PhC).
Akram et al. (2016) [142]	PM	Faster R-CNN [104] with thresholding in the bounding boxes, data: cells (Fluo).
Akram et al. (2016) [143]	PM, SM	Faster R-CNN [104] and second CNN for foreground-background prediction within predicted bounding boxes, data: cells (Fluo, PhC) and cell nuclei (Hist).
Chen et al. (2017) [144]	SM	Separate CNN branches for foreground and boundary prediction, data: nuclei and glands (Hist).

Continued on next page

Table 1.1: Overview of Deep Learning Methods with a Focus on Microscopy Image Instance Segmentation. (Continued)

Method/Authors[†]	Categories	Description
Xu et al. (2017) [145]	SM, PM	Separate CNNs for foreground, border, and object proposal prediction with a subsequent fusion CNN, data: glands (Hist).
Kumar et al. (2017) [146]	SM	Patch-based 3-class interior-boundary-background SM, data: cell nuclei (Hist).
Seferbekov (2018) [147]	SM	3-class object-border-background CNN for object separation, 2-class foreground-border CNN for object shape, and model ensembling, data: cell nuclei (Fluo, BF), winner of the 2018 Data Science Bowl [43].
Kulikov et al. (2018) [148]	SM	Prediction of a fixed number of colors to distinguish instances (ground truth coloring is part of the loss), data: leaves and bacterial cells.
Naylor et al. (2018) [139]	RM	Distance map regression, data: cell nuclei (Hist).
Guerrero-Peña et al. (2018) [149]	SM	3-class object-border-background SM with weighted cross-entropy loss using weight maps for class imbalance and cell geometry, data: cells.
Schmidt et al. (2018) [138]	PM, RM	Star-convex polygon proposals (need no shape refinement like bounding boxes) and distance map regression (object probability), data: cell nuclei (Fluo, BF).
Scherr et al. (2018) [150]	*SM*	*This thesis (adapted borders, see Subsection 3.1.1).*
Johnson (2019) [151]	PM	Mask R-CNN [42] for microscopy images, data: cell nuclei (Fluo, BF).
Wang et al. (2019) [152]	RM, PM	CNN for distance map regression and subsequent faster R-CNN [104] object detection on the distance maps, data: cells (Fluo, DIC, PhC).
J. Li et al. (2019) [153]	RM, SM	Separate object interior and center predictions, and regression of vectors pointing towards the center of an instance, data: cell nuclei (Hist).
X. Li et al. (2019) [154]	SM, RM	Two-decoder U-Net for boundary prediction & distance map regression with subsequent feature fusion for the final segmentation, data: cell nuclei (Hist).
Arbelle & Raviv (2019) [155]	SM	2-class foreground-background SM like Ronneberger et al. (2015) [61] with a convolutional long short-term memory U-Net, data: cells (DIC, Fluo, PhC).
Caicedo et al. (2019) [156]	SM	Comparison of patch-based [141] and U-Net-based 3-class interior-boundary-background SMs, data: cells (Fluo).
Chen et al. (2019) [105]	EM, RM	EM with cosine distance loss pushing adjacent objects into orthogonal spaces and a branch for distance map regression, data: cells (Fluo) and leaves.
Graham et al. (2019) [157]	SM, RM	Separate decoder branches for horizontal & vertical gradient map regression, and background-foreground & class prediction, data: cell nuclei (Hist).
Lux & Matula (2020) [158]	SM	Binary foreground and binary markers prediction, data: cells (DIC, Fluo, PhC).
Hollandi et al. (2020) [159]	PM	Mask R-CNN [42] pipeline with image style transfer, data: cells and cell nuclei (BF, Fluo, Hist).
Peña et al. (2020) [131]	SM	4-class object-border-gap-background SM with J regularization loss, data: cells (DIC, Fluo), cell nuclei (Fluo), and plant meristems (Fluo).
Wang et al. (2020) [160]	SM, RM	Approach of Graham et al. (2019) [157] with a bending loss penalizing contour points with large curvatures, data: cell nuclei (Hist).
Kulikov & Lempitsky (2020) [161]	EM	Sine waves-based EM, data: leaves, cells (DIC), and worms (BF, Fluo).
Tokuoka et al. (2020) [162]	SM	Two CNNs for foreground-background & marker prediction, data: cells (Fluo).
Zaki et al. (2020) [163]	RM	Distance map and blurred object boundary regression, data: cell nuclei (Fluo).
Dietler et al. (2020) [164]	SM	2-class foreground-background SM with object borders set to background (similar to Ronneberger et al. (2015) [61]), data: cells (BF, PhC).
Jang et al. (2020) [165]	PM	Mask R-CNN [42] adaptation for partially occluded objects, data: cells.
Isensee et al. (2020) [41], [166]	SM	3-class object-border-background SM with a self-configuring U-Net, data: cells (Fluo).
Scherr et al. (2020) [167], [168]	*RM*	*This thesis (object and neighbor distance maps, see Subsection 3.1.1).*
Stringer et al. (2020) [140]	SM, RM	Heat-diffusion-based vector flow field regression and foreground-background map prediction, data: cells and cell nuclei (Fluo, BF).
Buchholz et al. (2021) [169]	SM	Joint denoising and 3-class object-border-background SM, data: cell nuclei (Fluo).
Rumberger et al. (2021) [170]	EM	Probabilistic EM with vectors pulled to instance centers, data: worms (BF, Fluo).

Continued on next page

Table 1.1: Overview of Deep Learning Methods with a Focus on Microscopy Image Instance Segmentation. (Continued)

Method/Authors[†]	Categories	Description
Eschweiler et al. (2021) [171]	PM, RM	Spherical harmonics coefficients proposals & distance map regression (based on Schmidt et al. (2018) [138]), data: meristems and cell nuclei (Fluo).
Mandal & Uhlmann (2021) [172]	PM, RM	Spline curve control point proposals & distance map regression (based on Schmidt et al. (2018) [138]), data: cell nuclei (Fluo, BF).
Walter et al. (2021) [173]	PM, RM	Extension of Schmidt et al. (2018) [138] for overlapping objects using overlap probabilities, data: cytoplasm.
Lalit et al. (2021) [135], [174]	EM	EM that predicts center offset vectors, uncertainty vectors, and seediness scores (adapts [134]), data: cells and cell nuclei (BF, Fluo, PhC).
Hirling & Horvath (2021) [175]	PM, RM	Fourier coefficients proposals & distance map regression (based on Schmidt et al. (2018) [138]), data: cell nuclei (BF, Fluo, Hist).
Arbelle et al. (2022) [176]	SM	3-class interior-boundary-background SM with additional binary marker prediction (based on Arbelle & Raviv (2019) [155]), data: cells (BF, DIC, Fluo, PhC).
He et al. (2022) [177]	RM, SM	3-class interior-boundary-background prediction, centripetal direction map regression, and center point prediction, data: cell nuclei (Hist).
Jia et al. (2022) [178]	PM	YOLACT [179] for microscopy images, data: cell nuclei (Hist).
Zhang et al. (2022) [180]	RM	Distance map and enhanced boundary map regression (adapts Scherr et al. (2020) [167]), data: bacterial cells (Fluo).
Wagner & Rohr (2022) [181]	RM, SM	Centroid heatmaps regression and semantic ellipse height and width predictions with image pseudocoloring and self-attention, data: cells (Fluo, PhC).
Löffler & Mikut (2022) [136]	EM	EM for combined segmentation and tracking (based on Lalit et al. (2021) [135] and [134]), data: cells (BF, DIC, Fluo, PhC).
Cutler et al. (2022) [182]	SM, RM	Morphology independent adaptation of Stringer et al. (2021) [140] with Eikonal-equation-based vector flow field regression, data: bacterial cells (PhC).

[†] *Sorted by date of publication (stated are the publication years).*
Categories: EM – embedding method, PM – proposal-based method, RM – regression method, SM – semantic method.
Imaging: BF – bright-field microscopy, DIC – differential interference contrast microscopy, Fluo – fluorescence microscopy, Hist – histology images (RGB BF histology slide images), PhC – phase contrast microscopy.

1.3.3 Benchmark and Training Data Sets

Similar to particle detection, benchmark data sets are needed for objective comparison and training of instance segmentation methods. The Cell Tracking Challenge provides public data sets for 20 different scenarios [27], [28]. For the reference annotations, human-made gold truth, in which not necessarily each cell is annotated, and possibly erroneous computer-generated silver truth are distinguished. In addition, the challenge is open for submissions that are monthly evaluated and ranked for each scenario. Therefore, data with non-public gold truth reference annotations are used.

Further, the nucleus instance segmentation data sets CoNSeP [157], Kumar [40], [146], PanNuke [183], [184], MoNuSAC [185], Lizard [29], [186], and CryoNuSeg [187], as well as the glands instance segmentation data sets GlaS [117], and CRAG [188] can be used for training and benchmarking and are briefly described in [189]. The Broad Bioimage Benchmark Collection provides about 13 instance segmentation data sets, including the data from the Kaggle Data Science Bowl 2018 [32], [43]. EVICAN is a data set with about 26 000 annotated cells in 4600 partially annotated and 98 fully annotated BF and PhC images [190]. The Cellpose data set consists of 608 fully annotated images with over 70 000 cells [140], and the Omnipose data set of 794 images with over 47 000 cells [182]. Furthermore, LIVECells provides 1.6 million annotated cells in 5239 PhC images [191]. In regions where cell boundaries are not readily visible, no individual cells are annotated in LIVECell. All mentioned data sets are freely available.

1.3.4 Available Software Solutions

Multiple tools have been proposed, especially for the instance segmentation of biomedical microscopy images. The tools range from rather simple traditional image processing tools like the Otsu thresholding-based tool ChipSeg [192] to more complex multi-functional tool collections like CellProfiler [193], [194] and ilastik [195]. Some deep learning tools focus on leveraging the reuse of trained models, e.g., the Fiji plugin DeepImageJ for the reuse of deep learning models in life sciences applications [196], [197]. Other deep learning tools, such as Cellpose, also provide annotation, pre-labeling, training, and inference functionalities [140], [198]. The software deepflash2 provides multi-expert ground truth estimation, ensemble training, evaluation, prediction, and quality assurance pipelines and is based on the segmentation method of Cellpose [199]. YeaZ is a deep learning tool using a semantic method [164]. Cell-ACDC provides traditional and the deep learning segmentation methods YeaZ and Cellpose, and also cell tracking [200]. Further tools are StarDist as napari plugin [138], [201], AutoCellSeg [202], BacStalk [203], fastER [204], LABKIT [205], Misic [206], Orbit [207], TrackMate 7 [208], YeastSpotter [209], and the Jupyter notebook collection for Google Colab ZeroCostDL4Mic [210].

1.4 Evaluation Metrics

Performance measures are needed for an objective method comparison on benchmark data sets. However, different performance metrics focus on different error sources and penalize errors differently. False negatives can, for instance, be penalized stronger than false positives for applications that require subsequent tracking since false negatives are usually more complex to resolve than false positives. In [211], Maier-Hein et al. discuss pitfalls in metric selection and application and provide recommendations for image analysis validation, i.e., for image-level classification, semantic segmentation, object detection, and instance segmentation. In the following, the evaluation metrics used in this thesis are introduced.

1.4.1 Precision, Recall and F-Score

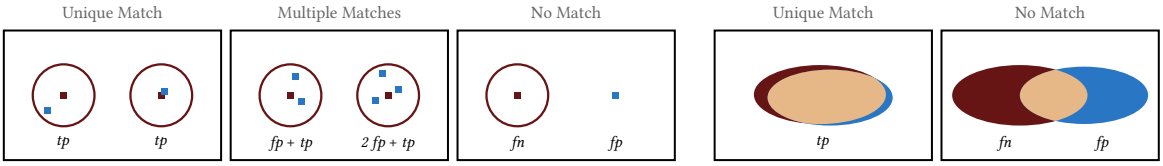
Precision, recall, and F-score are commonly used metrics for the evaluation of object and particle detection methods. The average precision for n images

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{tp_i}{tp_i + fp_i} \tag{1.1}$$

evaluates how many detections are relevant, and the recall

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{tp_i}{tp_i + fn_i} \tag{1.2}$$

which fraction of the objects have been detected. Therefore, the true positives tp, the false positives fp, and the false negatives fn need to be determined in each image i. For particle detection, this can be done by defining an area in which a single predicted marker must be located to be counted as true positive (see Figure 1.10a). Markers too much in such a ground truth area and markers in the predicted background are counted as false positives. An area without predicted marker is counted as a false negative. For instance segmentation tasks, the overlap between a ground truth instance



a) Particle Detection b) Instance Segmentation

Figure 1.10: Prediction - Ground Truth Matching. For particle detection and instance segmentation metric calculation, predictions need to be matched with the ground truth instances. For detection tasks, this can be done by defining an area in which a single predicted object marker needs to be located (a). For instance segmentation tasks, an overlap measure like the Jaccard similarity index can be used to match ground truth and predicted instances (b). Note: multiple matches are for instance segmentation usually not possible since a specific minimum overlap criterion is often applied to avoid this. For further instance segmentation error classes, refer to Figure 1.11. ■ / ■ – ground truth marker / predicted marker, ○ – ground truth area, ● / ● / ● – ground truth mask / predicted mask / overlap area.

and a predicted instance is often used to define true positives, false positives, and false negatives. For instance, an intersection over union or Jaccard similarity index threshold of 0.5 can be used, like in Figure 1.10b.

Finally, the average F-score or F-measure is defined as the harmonic mean of the average precision and average recall:

$$F_1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}} \,. \tag{1.3}$$

1.4.2 Normalized Acyclic Oriented Graph Matching Measure for Detection

The normalized acyclic oriented graph matching measure for detection DET assesses and evaluates how difficult it is to transform an acyclic oriented graph with nodes representing predicted objects into the ground truth graph. Numerically, it is defined as

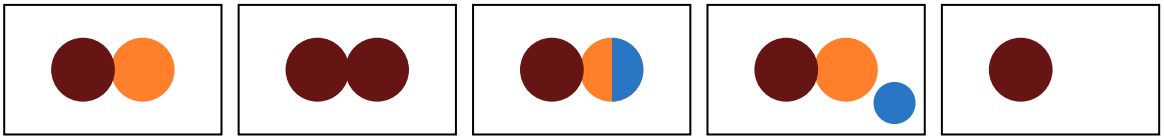
$$\text{DET} = 1 - \frac{\min\left(\text{AOGM-D}, \text{AOGM-D}_0\right)}{\text{AOGM-D}_0} \,, \tag{1.4}$$

where AOGM-D is the cost of transforming a set of nodes into the set of ground truth nodes, and AOGM-D_0 is the cost of creating the set of ground truth nodes from scratch, i.e., AOGM-D_0 is AOGM-D for empty detection results [212]. The costs include resolving false negatives, merged objects, and false positives. The costs are weighted with weights w_i representing the manual correction effort, i.e., false negative and merge costs are weighted more than false positive costs: $w_{\text{fn}} = 10, w_{\text{merge}} = 5, w_{\text{fp}} = 1$. The minimum operator in Eq. 1.4 prevents negative values in case creating the reference set from scratch is cheaper than transforming the predicted nodes. DET ranges from 0 to 1, with higher values indicating a better detection performance.

The DET measure is used in the Cell Tracking Challenge to evaluate the detection quality of the submitted instance segmentation methods. A predicted object and a ground truth object do match if the condition

$$\left|R_i \cap P_j\right| > 0.5 \left|R_i\right| \tag{1.5}$$

is fulfilled, where R_i denotes the set of pixels belonging to the reference object i and P_j the set of pixels of the predicted object j. A reference object can only be assigned to a single predicted object.



a) Ground Truth b) Merge c) Split d) False Positive e) False Negative

Figure 1.11: Instance Segmentation Object-Level Error Types. Besides pixel-level errors like inaccurate shapes, also object-level errors can occur. Adjacent objects can be merged into one object resulting in a merge (b), an object can be split into several parts (c), objects can be added (d), and objects can be missing (e).

1.4.3 Jaccard Similarity Index and Aggregated Jaccard Index

While the DET measure evaluates object-level errors (see Figure 1.11), the Jaccard similarity index can be used to evaluate pixel-level errors, i.e., how well a ground truth object and a predicted object match. The Jaccard similarity index is defined by:

$$J\left(R_i, P_j\right) = \frac{\left|R_i \cap P_j\right|}{\left|R_i \cup P_j\right|} . \tag{1.6}$$

The segmentation accuracy measure SEG $\in [0, 1]$ used in the Cell Tracking Challenge is the mean of the Jaccard similarity indices J of all reference objects. The condition Eq. 1.5 for matching must be fulfilled for the Jaccard index calculation. Otherwise, an empty set is used for the predicted set R_i resulting in a Jaccard similarity index of 0. An SEG score of 1 indicates a perfect match.

The aggregated Jaccard index AJI is computed with an aggregated intersection cardinality numerator and an aggregated union cardinality denominator that considers all ground truth and segmented objects [146]. Therefore, the pixels of false negatives and false positives are added to the denominator. Thus, this single metric can penalize both object-level and pixel-level errors, i.e., false negatives, false positives, under-segmentation, and over-segmentation. The computation of the aggregated Jaccard index is described in detail in [146]. In this thesis, the slightly modified aggregated Jaccard index AJI+ is used, which prevents overpenalization by using a one-to-one mapping instead of a one-to-many mapping for the predicted objects [157], [213]. The AJI and the AJI+ score range from 0 to 1, with 1 indicating perfect instance segmentation.

1.5 Open Questions

At the start of this dissertation, there were several open issues in the domain of deep learning-based particle detection and instance segmentation for microscopy images that needed to be addressed:

- Many microscopy image segmentation tasks are for humans rather simple but time-consuming. In contrast, designing a robust and versatile traditional segmentation method for these tasks is challenging. This discrepancy may be transferred to the data representation in deep learning: it is unclear if, for humans easy to interpret, semantic methods are easier to learn and more robust than more complex regression methods. Especially when using small training data sets, regression methods may be superior since they offer more post-processing options, e.g., for adjusting the object size.

- Instance segmentation requires the distinction of adjacent objects. A potential issue is that only a few adjacent objects are in the training data, but the application demands a good

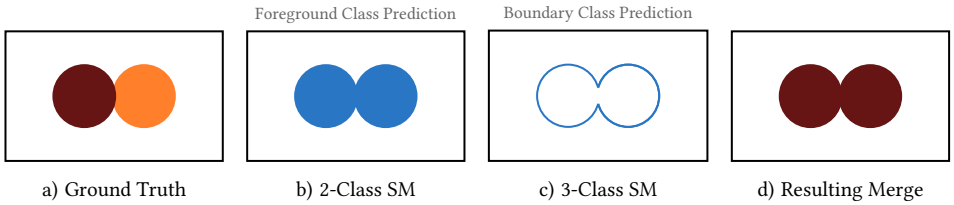


Figure 1.12: Merge Error Sources of Semantic Methods. Semantic methods for instance segmentation that use a boundary or object contour class or simply differentiate between background and foreground are often unreliable since a few miss-classified pixels can result in merges. 2-class methods may not predict the small background separations introduced between adjacent objects (b). 3-class methods with a boundary class may predict most of the boundary correctly but may fail in between adjacent objects.

distinction for an error-free analysis. Figure 1.12 shows potential merge errors of simple semantic methods. Thus, a robust encoding of neighbor information is required to help the CNN learn the separation of adjacent objects from only a few examples.

- Researchers and engineers often work with multiple imaging setups and have various subjects to investigate, e.g., bioengineers with different phenotypes of bacteria. For efficient and accurate analysis of the acquired data, a method is required that works well for multiple imaging and object modalities without the need for major tuning of the training process or large training data sets. Such a method would leverage deep learning for researchers and engineers working with multiple imaging setups.
- Very low-resolution particles are even for humans challenging to detect. However, annotated training data are required for training deep learning methods. Furthermore, particle detection methods can have a resolution limit. For instance, adjacent 3×3 px particles cannot be separated with the 3×3 px markers used in [88]. These issues raise the question of whether it is possible to overcome the resolution limit and facilitate the annotation process.
- Though more and more public benchmark data sets are available, microscopy setups, sample preparation techniques, and the samples themselves are very diverse. This diversity often results in a need for application-specific annotated data to boost performance. However, so far, most tools lack a data management system and a built-in easy-to-use training data creation and model training pipeline, including crop creation, manual annotation, pre-labeling, and model evaluation (see Figure 1.13). This lack hinders an efficient microscopy image analysis.
- A major concern about deep learning is that extensive and time-consuming training data annotation is required and that hyperparameters must be tuned for successful training and application. An open question is if it is possible to get reasonable results in a short time, e.g., in less than one hour – including annotation, training, and inference – starting from scratch without any training data.

1.6 Objectives and Thesis Outline

Based on the previously introduced issues, the main objectives of this thesis are:

- development of a semantic method and a regression method with a robust neighbor information encoding for instance segmentation – specifically, the methods should generally not

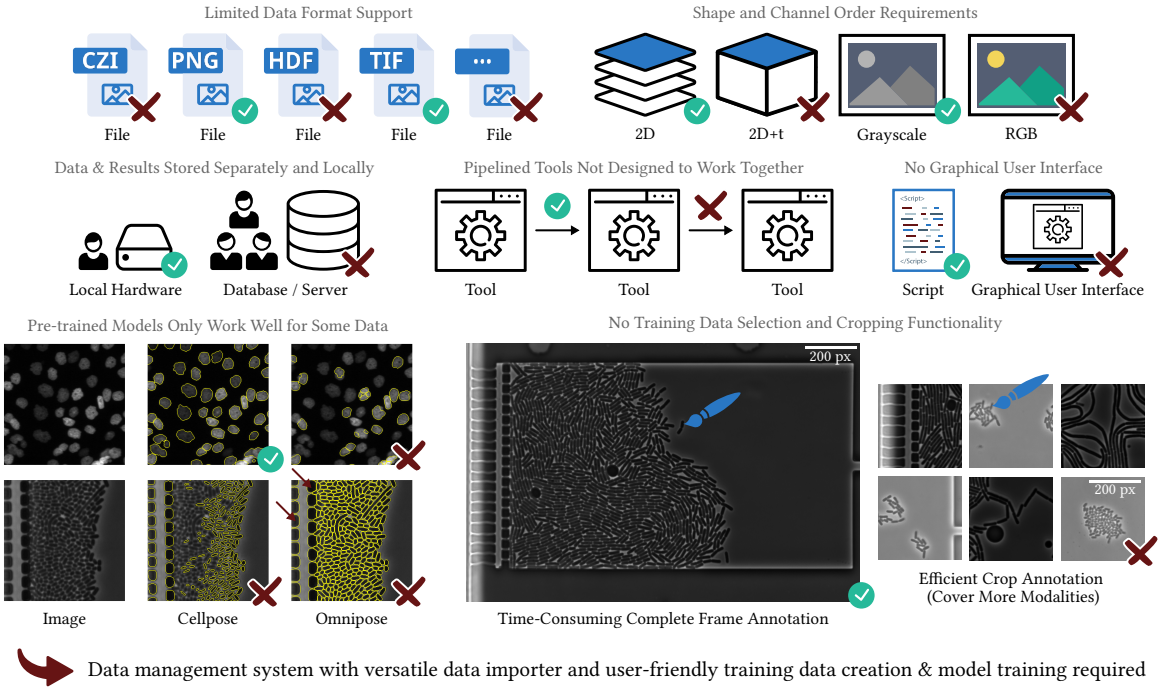


Figure 1.13: Typical Barriers when Using Segmentation Software. The lack of a data management system with a versatile data importer results in additional file format and image shape conversion steps. In addition, multiple potentially incompatible tools often need to be combined to cover the whole workflow, from training data creation to the application of trained models. Again, conversion steps can be needed to enable compatibility. Furthermore, it is for many applications (not yet) possible to do without their own annotated training data. For an efficient annotation, interactive cropping functionalities are needed, especially for dense-growing organisms. Note: Fluo/PhC images inverted for Omnipose/Cellpose [140], [182]. Illustration modified from [214].

need large training data sets and be able to learn to distinguish adjacent objects also from only a few samples in the training data set,

- validation of the microscopy image instance segmentation methods for various objects and various imaging setups, i.e., bright field microscopy, phase contrast microscopy, and fluorescence microscopy,
- development and validation of a particle detection method for low-resolution microscopy images – including a pre-processing facilitating the manual data annotation and the separation of adjacent objects,
- incorporation of the developed methods into new, efficient, and user-friendly open-source tools, and
- the proof that reasonable results can be obtained within one hour with the tools – without using any available annotated data sets.

The remainder of this thesis is as follows: Chapter 2 presents a U-Net-based particle detection method with object size related pre-processing for low-resolution microscopy images. A manually annotated real-world latex bead fluorescence microscopy data set is used for validation, and a synthetic data set is artificially degraded in several steps to analyze the resolution limit of the object detection method. Furthermore, the open-source object detection tool BeadNet is described, and

application cases are given. In Chapter 3, two new neighbor information encodings are introduced, and the corresponding two instance segmentation methods utilizing a U-Net with two decoder paths are described. The instance segmentation methods are compared with other deep learning methods on four Cell Tracking Challenge data sets using a single model for the two bright-field and the two fluorescence microscopy data sets with different object shapes, sizes, and textures. Furthermore, the superior method is validated with Cell Tracking Challenge submissions for the 5th Cell Tracking Challenge edition as part of the ISBI 2020 and the 6th edition as part of the ISBI 2021. Once a single model is used for all data sets and once specialized models are used for the individual data sets. In addition, the open-source instance segmentation software microbeSEG is presented, and the efficient workflow is validated. Finally, Chapter 4 summarizes the contributions of this thesis and offers an outlook for potential future research.

2 Particle Detection

Particle detection is the task of localizing every single particle in an image. In particular, poorly-resolved particles are difficult to detect in microscopy images. This issue even complicates the training data and ground truth annotation for humans. In contrast to instance segmentation, particle detection provides no information about the size and shape of the localized particles. However, an advantage is the faster training data creation since only particle markers need to be annotated and no object contours need to be drawn. So far, deep learning-based semantic segmentation approaches rely on non-robust single-pixel particle center predictions or cannot deal with small particles if larger particle markers are used. Therefore, this chapter focuses on developing and validating a new semantic segmentation approach for detecting small particles that are not well-resolved and lack prominent shape and texture characteristics. Furthermore, application cases of the newly developed software BeadNet are presented. BeadNet facilitates applying the new method using a user-friendly but comprehensive training data creation, model training, and model application workflow, and the OMERO platform for data management.

2.1 U-Net-Based Semantic Segmentation of Particle Markers

Figure 2.1 provides an overview of the semantic, U-Net-based particle detection method. The biquadratic upsampling pre-processing enables the U-Net to work at reasonable scales, which is required for the robust detection of poorly resolved particles. Trained to predict a semantic segmentation with the two classes particle marker and background, the U-Net can be used for particle detection. In contrast to [101], which uses no pooling and upsampling layers, the use of three learnable pooling and three learnable upsampling layers should result in a broader range of particle sizes for which the method works. The skip connections retain positional information, which is important for precise particle localization. Furthermore, the upsampling pre-processing and the marker representation must be designed to avoid merging particle markers. The used marker representations, the upsampling pre-processing, the U-Net architecture, the training process, and the inference are described in this section.

2.1.1 Upsampling Pre-processing for Improved Particle Separation

The upsampling pre-processing has two goals: (i) improving the annotation accuracy and speed during the training data creation and (ii) enabling a robust detection of small, poorly resolved particles. Figure 2.2 shows that upsampling visually facilitates detecting particles and distinguishing adjacent particles for small particle sizes. In addition, the particle center is better defined after the upsampling. Thus, annotating upsampled images is probably faster and more accurate than annotating them at their original scale. The feature extraction of the U-Net may also benefit from the, for the human eye, improved visual discrimination of the single particles. In this thesis, biquadratic spline interpolation is used for upsampling. However, using a higher-order interpolation or bilinear interpolation may work as well.

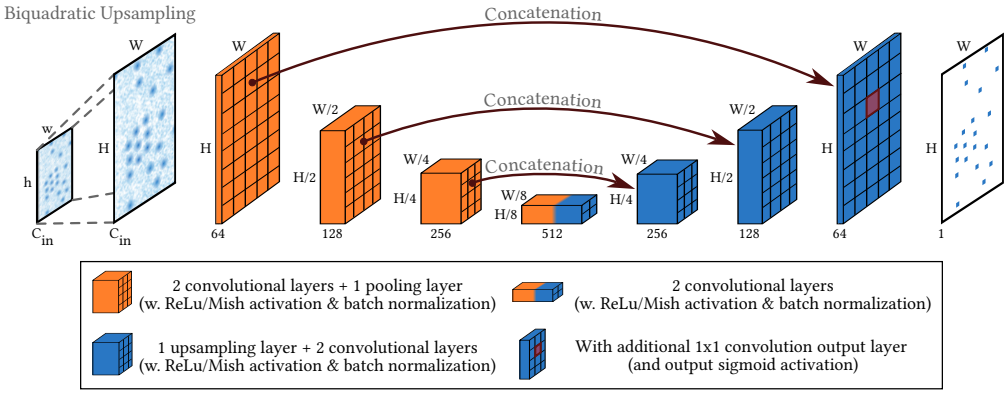
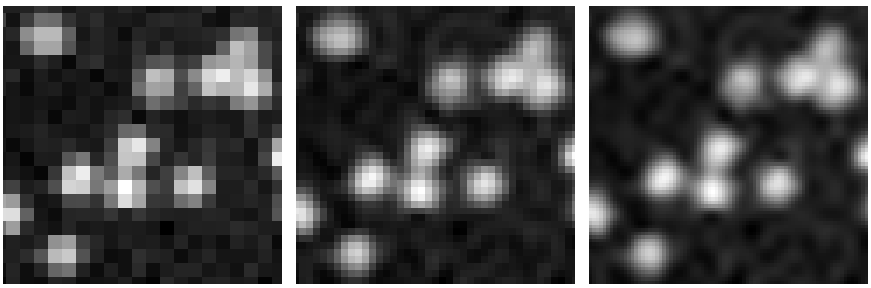


Figure 2.1: Overview of the Particle Detection Method Based on the Semantic Segmentation of Particle Markers. The object size dependent biquadratic upsampling enables the U-Net to work at reasonable scales. The U-Net has three down- and upsampling blocks and is trained to predict particle markers. Note: concatenated are the outputs of the last convolutional layer in a block and of the corresponding transposed convolutional layer. H/W/C – height/width/channel dimension, h/w – initial height/width before upsampling.

The particle size defines an upper limit for the maximum usable marker size since markers of adjacent particles will merge when they reach the particle size. This limit is an issue for poorly resolved particles, e.g., for particles with a diameter of 3 px where no 3×3 markers can be used, as shown in the following. However, smaller markers may not be robust to single-pixel misclassifications. The upsampling pre-processing allows using larger marker sizes for small particles and contributes to a robust detection of small, poorly resolved particles.

2.1.2 Marker Representation

As described before, the upsampling pre-processing and the marker representation must be designed to avoid merging particle markers, i.e., particle markers of adjacent particles should not share an edge. Otherwise, adjacent particles cannot be distinguished with simple connected-component labeling. If this marker representation requirement is fulfilled, the particle detection can be treated as a semantic segmentation task with the two classes background and particle marker, and a simple connected-component labeling post-processing is sufficient. In addition, the marker representation should prevent false positives due to splitting particles into multiple parts.



a) Without Upsampling b) 2x Upsampling c) 4x Upsampling

Figure 2.2: Biquadratic Upsampling Pre-processing. The upsampling helps the human eye to distinguish single, poorly resolved particles like the shown particles with a diameter d from 3 px to 4 px. The upsampling, therefore, facilitates the creation of training data and may enable more accurate predictions. Data source: BBBC004 [32].

1 x 1 Markers

The most obvious and straightforward way is to use 1×1 markers consisting of a single pixel in a particle's center (see Figure 2.3). A drawback of 1×1 markers is that a spatial distance of a single pixel between the ground truth marker and the predicted marker is highly penalized if no specialized loss function is used. However, standard loss functions like the cross-entropy loss compare single pixels and no neighborhoods. Thus, the CNN is enforced during training to predict exactly the ground truth marker to avoid a large loss value, although predicting a neighboring pixel may also be sufficient for the application. In addition, real-world human-annotated training data are imperfect. For instance, depending on the image quality and particle size, it may be very difficult to hit each particle's center during annotation constantly, and it may be that neighboring pixels are annotated instead. This annotation variability may result in a demand for large training data sets or an unstable training process of the 1×1 markers, which would need to reproduce the variability exactly. Furthermore, the class imbalance between the background class and the particle marker class may also negatively influence the training process of the CNN.

Dilated 1 x 1 Markers

The morphological dilation with a cross-shaped kernel generates dilated 1×1 markers that consist of five pixels (see Figure 2.3). Dilated 1×1 markers reduce the class imbalance compared to 1×1 markers and, therefore, can help facilitate a CNN's training process. In addition, the probability of splitting or missing particles due to single-pixel misclassifications should be minimized due to the spatial extension of the markers. This spatial extension also allows partial overlaps of predicted and ground truth markers, which may be beneficial for training with training data samples with imperfect or inconsistent spatial information due to annotation variability. Instead of penalizing predicting a neighboring pixel of the ground truth particle center as center completely (4-neighborhood), the overlap enables partially correct predictions. This may result in a more stable training process that converges to a better minimum. Figure 2.3c illustrates that markers can merge for small particle sizes, which leads to a need for the upsampling pre-processing.

3 x 3 Markers

3×3 markers, like used in [101], consist of nine pixels and further reduce class imbalance compared to 1×1 and dilated 1×1 markers (see Figure 2.3). In addition, the larger 3×3 markers should be more robust to annotation variability than the dilated 1×1 markers. However, the cross shape of the dilated 1×1 markers allows a better separation of small, diagonally touching particles than the 3×3 markers, which require a larger particle size to work well in that case (compare the three particles at the lower left in Figure 2.3c and Figure 2.3d or in Figure 2.3g and Figure 2.3h). Therefore, the 3×3 markers may only be beneficial for large particles or require a larger upsampling than dilated 1×1 markers making them computationally more expensive.

2.1.3 CNN Architecture

The CNN trained to predict particle markers is a U-Net with batch normalization. Figure 2.1 visualizes the architecture. The inputs of convolutional layers are zero-padded to keep the spatial feature map dimensions constant and to avoid cropping before concatenating corresponding encoder and decoder feature maps. Convolutions with stride two are used for downsampling, and transposed convolutions are used for upsampling. In the first convolutional layer, 64 feature maps are used.

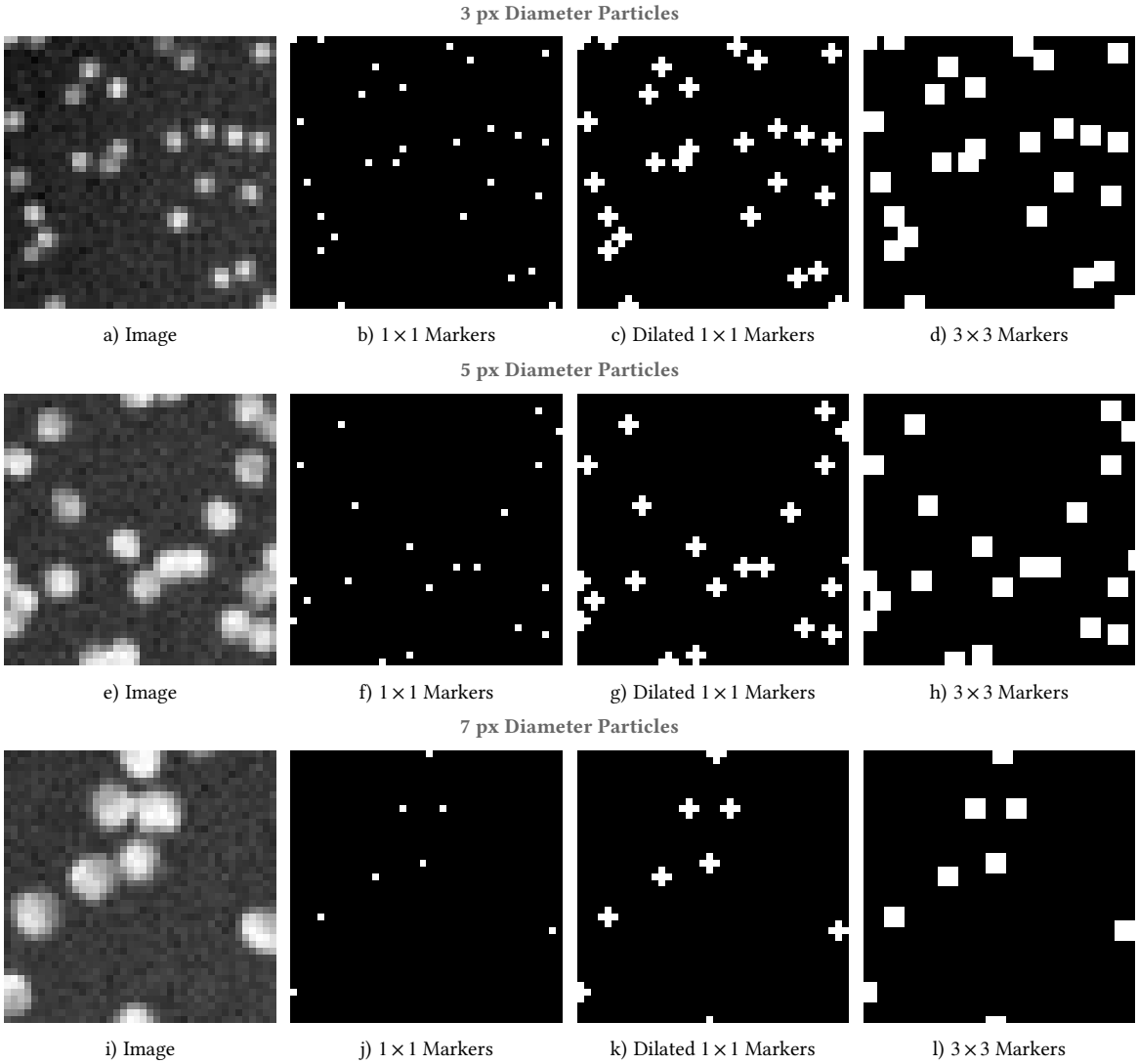


Figure 2.3: Particle Marker Representations. CNNs can be trained to predict particle markers. In doing so, each marker represents a single particle. 1×1 markers consist of a single pixel, commonly a particle's centroid. The morphological dilation with a cross-shaped kernel generates the dilated 1×1 markers. Larger markers like the 3×3 markers can reduce the class imbalance, but the particle size sets an upper limit for the marker size as markers merge for small particles. The background class is visualized in black, and the particle marker class in white in the marker images. Shown are 80×80 px images. Data source: BBBC004 [32].

After downsampling, the number of feature maps is doubled in the following convolutional layer until 512 feature maps are reached. In turn, the number of feature maps is halved in a transposed convolutional layer and in the subsequent convolutional layer after concatenation. The sigmoid activation function is applied after the output convolutional layer and the ReLU activation function elsewhere. The shown U-Net has about 8.5 million trainable parameters.

2.1.4 Training Process

The U-Net is trained with the Adam optimizer in the AMSGrad variant ($\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay). The start learning rate lr_{start} is set to $8 \cdot 10^{-4}$, and a batch size of 4 is used. A training-validation split of 80 %/20 % is used, and trained is on 128×128 px crops extracted from upsampled and min-max normalized images of an annotated data set. The cropping has the advantage that crops extracted from empty regions can be excluded from the training.

Loss Function

The loss function is the weighted sum of binary cross-entropy loss L_{BCE} [215] and Dice loss L_{Dice} [216]:

$$L\left(y, \hat{y}\right) = L_{\text{BCE}}\left(y, \hat{y}\right) + 0.5 L_{\text{Dice}}\left(y, \hat{y}\right), \tag{2.1}$$

with

$$L_{\text{BCE}}\left(y, \hat{y}\right) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \cdot \log \hat{y}_i + \left(1 - y_i\right) \cdot \log\left(1 - \hat{y}_i\right) \right], \tag{2.2}$$

and

$$L_{\text{Dice}}\left(y, \hat{y}\right) = 1 - \frac{2 \sum_{i=1}^{N} y_i \, \hat{y}_i}{\sum_{i=1}^{N} y_i^2 + \sum_{i=1}^{N} \hat{y}_i^2}. \tag{2.3}$$

y is the ground truth marker representation and $\hat{y}$ the marker prediction. Thus, the loss and the optimization process of a model depend on the selected marker representation. The Dice loss part aims to improve the performance for class-imbalanced data and is, therefore, added to the binary cross-entropy loss part.

Learning Rate Scheduler and Stopping Criteria

Models are trained for a maximum number of epochs N_{max} that depends on the number n of training and validation image crops:

$$N_{\text{max}} = \begin{cases} 120 & \text{if } n \geq 1000, \\ 180 & \text{if } 1000 > n \geq 500, \\ 240 & \text{if } 500 > n \geq 250, \\ 300 & \text{if } 250 > n \geq 100, \\ 360 & \text{if } 100 > n \geq 50, \\ 420 & \text{else.} \end{cases} \tag{2.4}$$

The values are set empirically and provide a good trade-off between training time and model performance for many applications. The learning rate is multiplied with a factor γ of 0.25 when the validation loss has not decreased for $N_{\text{patience}} = \frac{1}{20} N_{\text{max}}$ epochs until the minimum learning rate lr_{min} of $3 \cdot 10^{-6}$ is reached. The training process stops after $N_{\text{stop}} = 2N_{\text{patience}} + 5$ epochs without validation loss improvement. This early stopping criterion is often fulfilled before N_{max} epochs have been trained. The model checkpoint with the best validation loss is used for further analysis.

Augmentations

The following training data augmentations are applied independently from each other in the stated order with the stated probability p to improve the generalization of a trained model to unseen data, to avoid overfitting, and to stabilize the training process in case of small training data sets:

- *flipping* ($p = 87.5\,\%$): flip (up-down, left-right), rotation by multiples of 90°, or combination,
- *contrast* ($p = 45\,\%$): histogram equalization, contrast stretching, or gamma adjustment,
- *scaling* ($p = 25\,\%$): scaling with random scale factor $s \in [0.85, 1.15]$,
- *rotation* ($p = 25\,\%$): rotation by the random angle $\alpha \in [-45°, 45°]$,
- *blurring* ($p = 30\,\%$): Gaussian blur with random $\sigma \in [1, 2]$,
- *noise* ($p = 30\,\%$): additive Gaussian noise with random $\sigma \in \left[0.01\, I_{\text{max}}, 0.05\, I_{\text{max}}\right]$.

I_{max} is the maximum intensity in a training image. Label-preserving augmentations like blurring modify only the training image, not the marker image. Label-changing augmentations require a transformation of the marker images as well, e.g., rotation and scaling. Therefore, nearest neighbor interpolation is used since higher-order interpolation can result in values between 0 (background) and 1 (marker), requiring some further processing steps.

2.1.5 Inference and Post-Processing

Min-max normalized and upsampled images are fed to the U-Net for inference. In the post-processing, the raw U-Net predictions are binarized using a threshold of 0.5. Then, connected components are identified in the generated binary maps, and the centroid is extracted for each identified region. Finally, the centroid coordinates are scaled to the initial image resolution with subpixel accuracy.

2.2 Validation

This section evaluates the upsampling strategies and marker representations for various particle sizes on synthetic data sets. All experiments have been performed on a system with Intel Core i9-9990K CPU, 64 GiB RAM, and an NVIDIA TITAN RTX GPU with 24 GiB VRAM. The particle detection method is implemented in Python, and PyTorch is used as deep learning framework (see Section 2.3). A border correction is applied to all results to minimize the influence of partially visible particles at image borders on the results. Therefore, particle predictions within a 6 px image border are filtered in this section. Later in the applications Section 2.4, the new particle detection approach is compared with traditional methods on a real-world fluorescent latex bead data set.

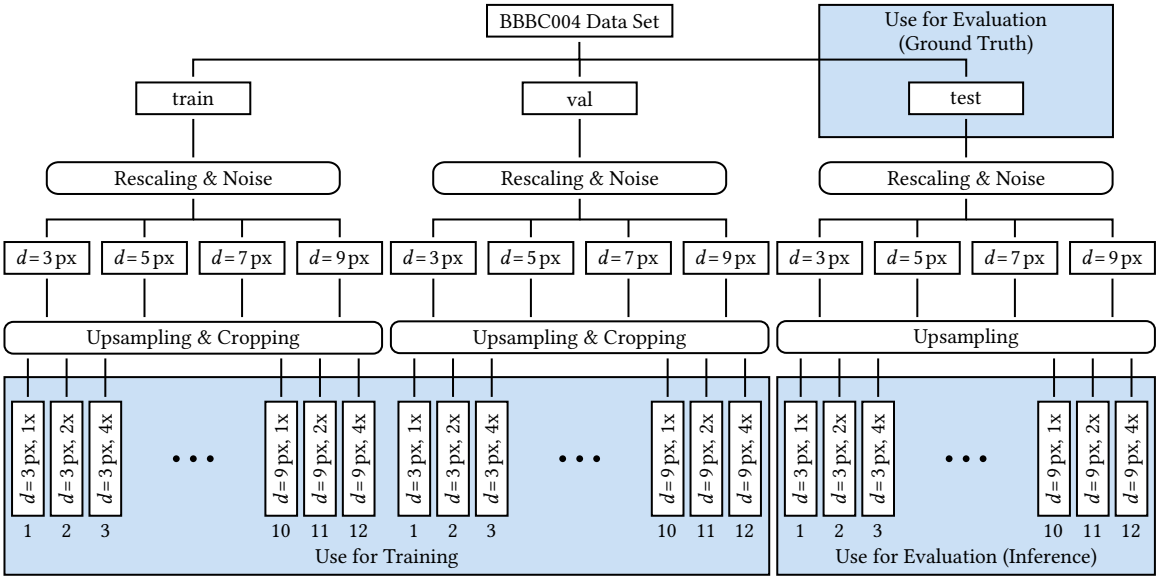


Figure 2.4: Creation of the Training and Test Data Sets for the Particle Detection Method Validation. The BBBC004 data are first assigned to a training (train), a validation (val), or a test subset. Then, the data are rescaled to match a specific particle diameter d, and noise is added to the images to compensate for the noise-filtering effects of the rescaling. Finally, the data are upsampled with the three upsampling strategies (1x, 2x, 4x), and cropped in the case of training and validation subsets. This procedure ensures that the same particles are assigned to the same subset of the twelve created data sets.

2.2.1 Compared Upsampling Strategies and Marker Representations

The optimum upsampling and marker representation depends on the particle size. Thus, no upsampling (1x), 2x upsampling, and 4x upsampling are compared on synthetic data sets with different particle sizes. In addition, the marker representations 1×1, dilated 1×1, and 3×3 are compared for the three upsampling strategies for the different particle sizes.

2.2.2 Training and Test Data

Figure 2.4 shows the creation of synthetic data sets with different particle diameters and upsampling pre-processing. Therefore, the synthetic data set BBBC004 from the Broad Bioimage Benchmark Collection [32] has been rescaled such that the stated particle size has been obtained to create these data sets. After the rescaling, additive Gaussian noise was added. Finally, the data sets have been upsampled depending on the upsampling strategy, and crops of size 128×128 px have been extracted. Figure 2.5 visualizes exemplary images of the twelve newly created training data sets with different particle sizes and upsampling strategies.

Due to the cropping, all generated data sets consist of 128×128 px images. The assignment to the training and validation subset is the same for all twelve data sets to minimize the influence of different training-validation splits on the evaluation. In detail, the normalized images were first assigned to the training or validation subset. Then, the images were rescaled, and crops were generated for the different settings. Due to the different upsampling strategies, the file sizes differ, and the training data set sizes range from $n = 40$ ($d = 3$ px, 1x) to $n = 5049$ ($d = 9$ px, 4x) crops[1].

[1]The 4x upsampling results in 16 times more crops, and the larger diameter in 9 times more crops. This results in 144 times more crops for the largest data set compared to the smallest data set, minus crops without particle.

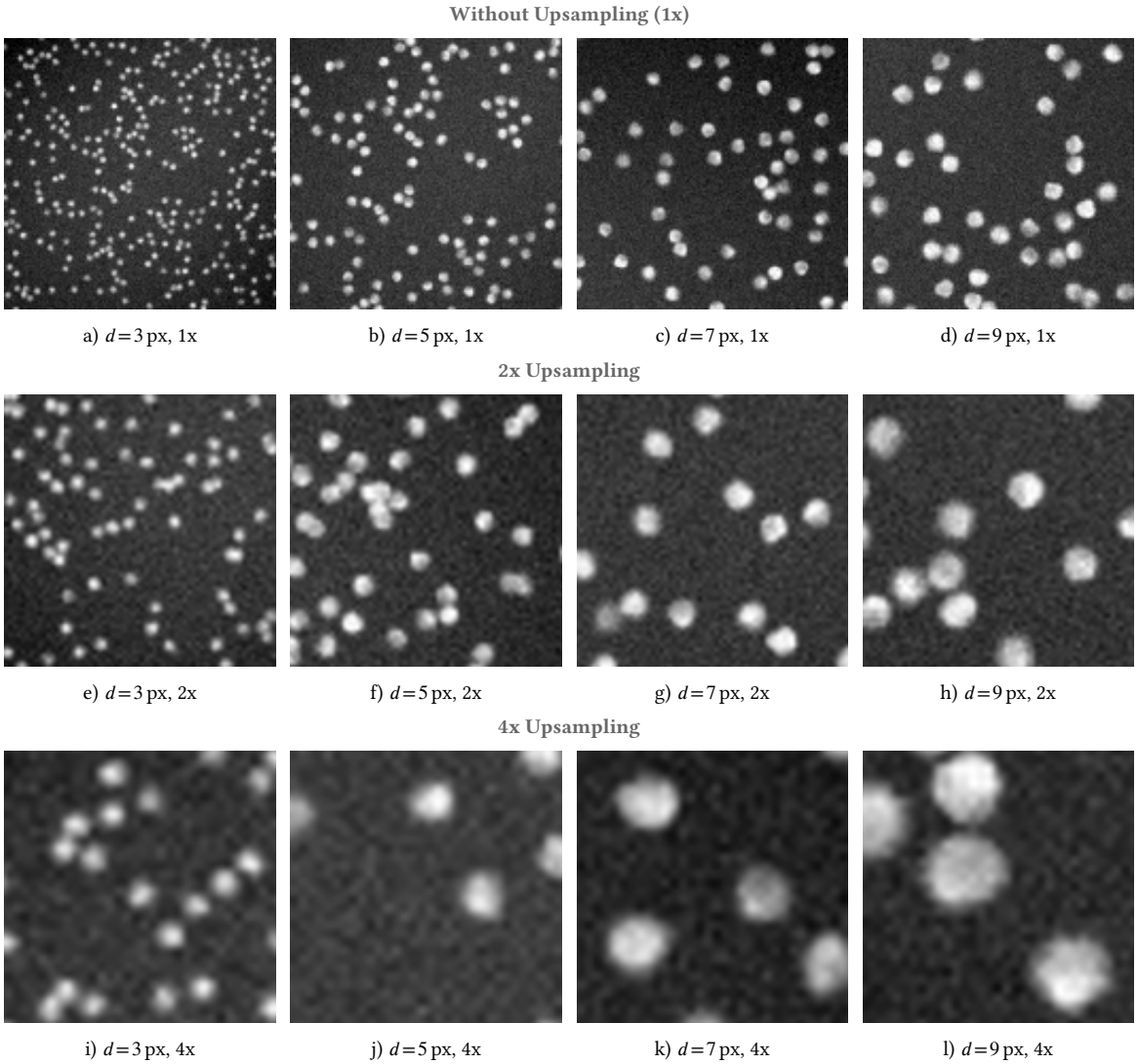


Figure 2.5: Exemplary Images of the Created Training Data Sets. Stated are the particle diameters d before the upsampling. Each training data set consists of 128×128 px crops. The shown data have been created from the BBBC004 data set [32], as illustrated in Figure 2.4.

However, the number of particles is nearly constant in the single data sets (about 11 000 particles). Depending on the image size, only some particles at image borders can be lost during cropping.

Models are trained on the training-validation split of a created training data set. Separate test images with a higher probability of adjacent objects than in the training data are used to evaluate the trained models. The test subset for each particle size and upsampling strategy contains identical 5463 particles at different scales.

Since the BBBC004 data set does not provide instance segmentation ground truth to get particle size information and to extract the particle centroids needed, the high-resolution data have been segmented with microbeSEG (see Section 3.3) using a pre-trained model [217]. The microbeSEG results have been manually inspected and corrected with the image annotation tool ObiWan-Microbi [218].

2.2.3 Results

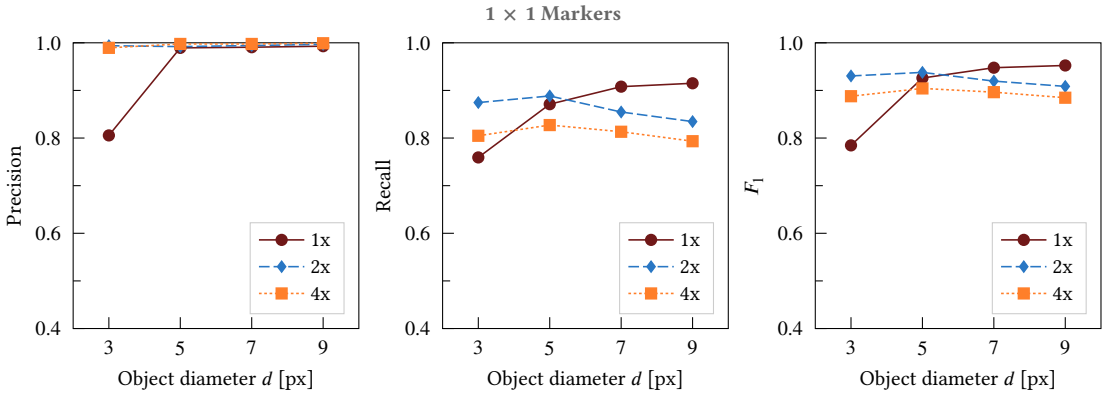
Three 1×1 marker models, three dilated 1×1 marker models, and three 3×3 marker models have been trained on each of the twelve training data sets for the three marker representations. Thus, in total, 108 models have been trained for this validation study. For a fair comparison, all predicted centroid positions have been rescaled to the original BBBC004 data set resolution, and a search radius $r_S = 7$ px is used to specify the ground truth areas, which define true positives, false positives, and false negatives (see error definition in Figure 1.10a in Chapter 1). The particles have a diameter of about 22 px at the original resolution. The rather large ground truth area avoids disadvantaging methods without upsampling for small particle sizes when a particle's center is difficult to define. However, since r_S is still smaller than the particle size at the original resolution, no predictions at the very outer area of a particle are counted as true positives.

Precision, Recall, and F-Score

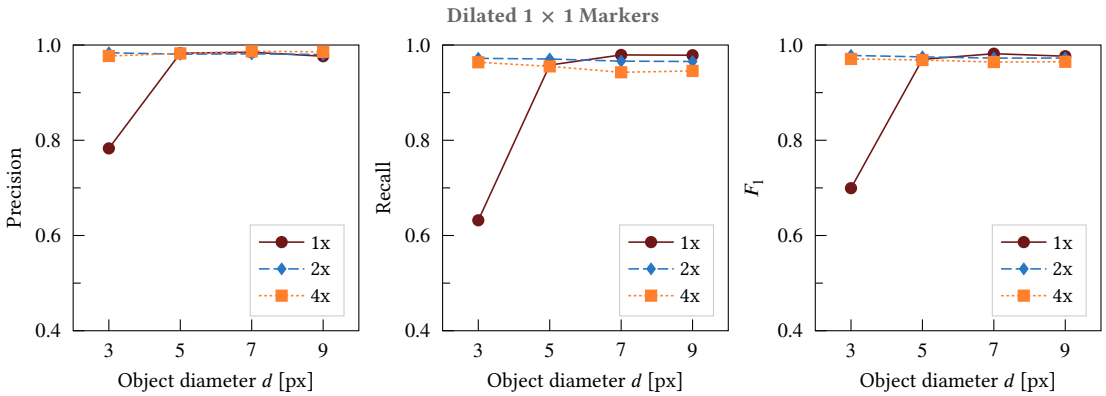
Figure 2.6 shows the median average precision, median recall, and median F_1 score of the three trained models per setting. The results for the 1×1 markers reveal a nearly constant precision except for the score for the 3 px diameter particles without upsampling (see Figure 2.6a). The recall shows that 4x upsampling is beneficial for the 3 px diameter particles. Furthermore, 2x upsampling is beneficial for 3 px, and 5 px diameter particles and constantly outperforms the 4x upsampling. In the other cases, no upsampling (1x) provides the best results. The decline of the recall scores for the 2x and 4x upsampling indicates that particles may get too large for this method, e.g., the detection of the 9 px particles with 4x upsampling corresponds to the detection of 36 px particles on the native scale, which can be difficult with single pixel markers. The F_1 scores are mainly limited by the recalls. Thus, false negatives do mainly limit the 1×1 marker results.

The precision scores of the dilated 1×1 markers show similar behavior to those of the 1×1 markers, but the performance drop for 3 px particles without upsampling is larger (see Figure 2.6b). This larger drop could be related to the higher probability of merging adjacent particles since the particle markers are larger. The recall scores reveal an urgent need for upsampling of the small 3 px diameter particles, while no upsampling provides the best recalls for the 7 px and 9 px diameter particles. Compared to the 1×1 markers, the performance drop due to the upsampling is smaller for these larger particles. The F_1 scores further illustrate the need for upsampling for the 3 px diameter particles. The scores for the other data sets are similar.

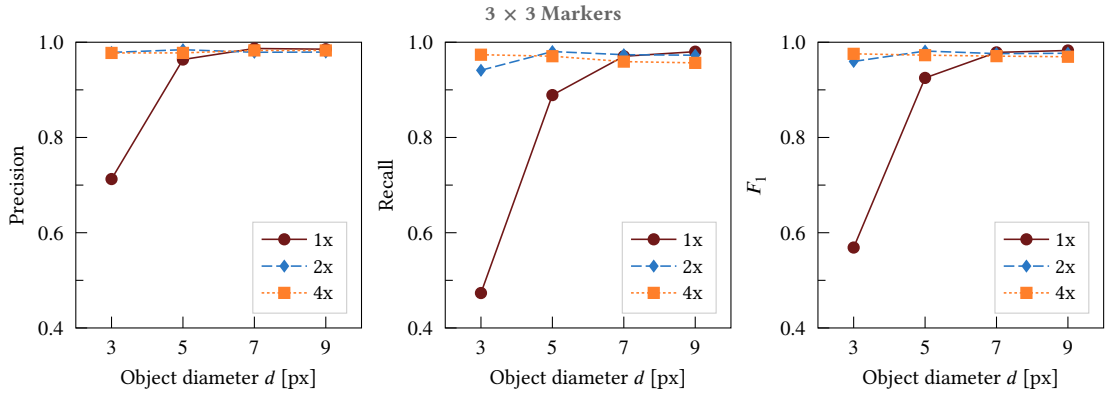
The 3×3 markers require a larger minimum particle size to avoid the merging of adjacent particles. Thus, precision, recall, and F_1 score drop without upsampling for the 3 px and 5 px



a) Precision (left), Recall (center), and F-Score (right) for Particle Detection with 1 × 1 Markers

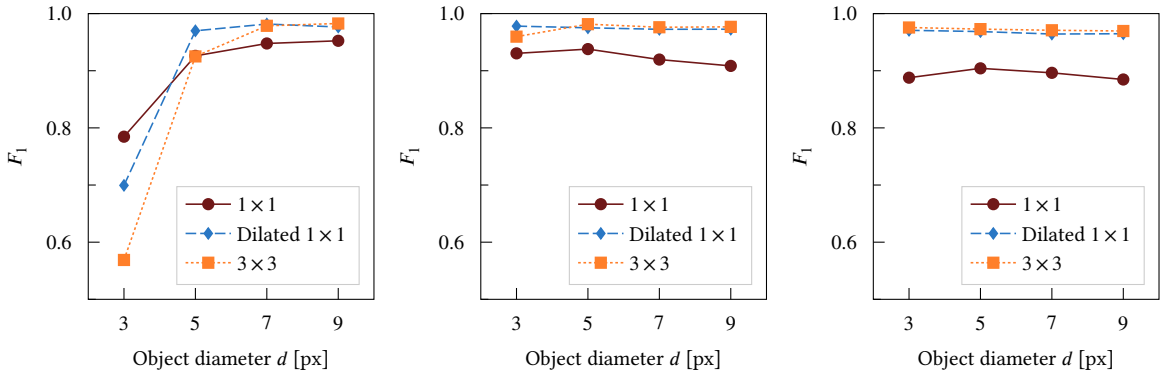


b) Precision (left), Recall (center), and F-Score (right) for Particle Detection with Dilated 1 × 1 Markers



c) Precision (left), Recall (center), and F-Score (right) for Particle Detection with 3 × 3 Markers

Figure 2.6: Particle Detection Results for Different Upsampling Strategies. The results for the 1 × 1 markers (a), the dilated 1 × 1 markers (b), and the 3 × 3 markers (c) show that an upsampling pre-processing is required for small particle sizes, especially for the larger 3 × 3 markers. Each value represents the median of three trained models. Stated are the particle diameters d before the upsampling. All settings have been evaluated on the high-resolution ground truth with a search radius r_s of 7 px.



a) Without Upsampling (1x) b) 2x Upsampling c) 4x Upsampling

Figure 2.7: Particle Marker Representation Comparison. The use of dilated 1×1 and 3×3 markers requires a particle diameter d larger than 3 px and 5 px, respectively. Therefore, upsampling can be applied. The 3×3 markers are slightly better than the dilated 1×1 markers when working on a reasonable particle diameter. In contrast, the 1×1 cannot produce high-quality results. Each shown value represents the median of three trained models. Stated are the particle diameters d before the upsampling. All settings have been evaluated on the high-resolution ground truth with a search radius r_s of 7 px.

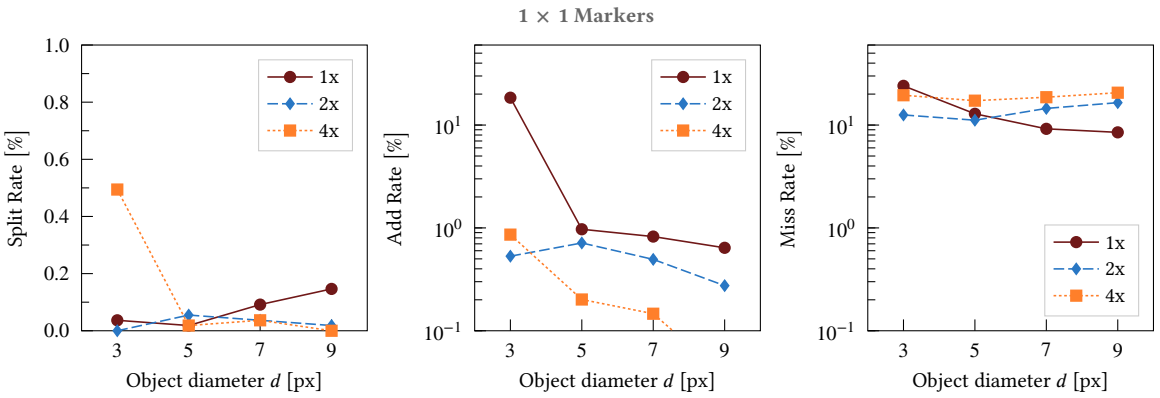
diameter particles (see Figure 2.6c). In addition, the 2x upsampling is not sufficient to reach a high recall for the small 3 px diameter particles. So, the 3×3 markers require a 4x upsampling for 3 px diameter particles and at least a 2x upsampling for 5 px diameter particles. For large particles, the 4x upsampling provides no benefits for all marker representations. Interestingly, as could be assumed, there is no actual performance drop when particles become large compared to the marker size, like for the 2x upsampling recall scores of the 1×1 markers.

The arrangement of the results in Figure 2.7 allows a more straightforward comparison of the marker representations for the three upsampling strategies. The 1×1 markers are only the best choice for 3 px diameter particles when no upsampling is used. However, for such small particles, an upsampling step is required anyway to get good results. With upsampling, the dilated 1×1 markers and the 3×3 markers outperform the 1×1 markers. Thereby, the performance of the 3×3 markers is slightly higher than the performance of the dilated 1×1 markers. Only for the small 3 px diameter particles and 2x upsampling, the dilated 1×1 markers can outperform the 3×3 markers, which do require a larger particle size to prevent particles from merging.

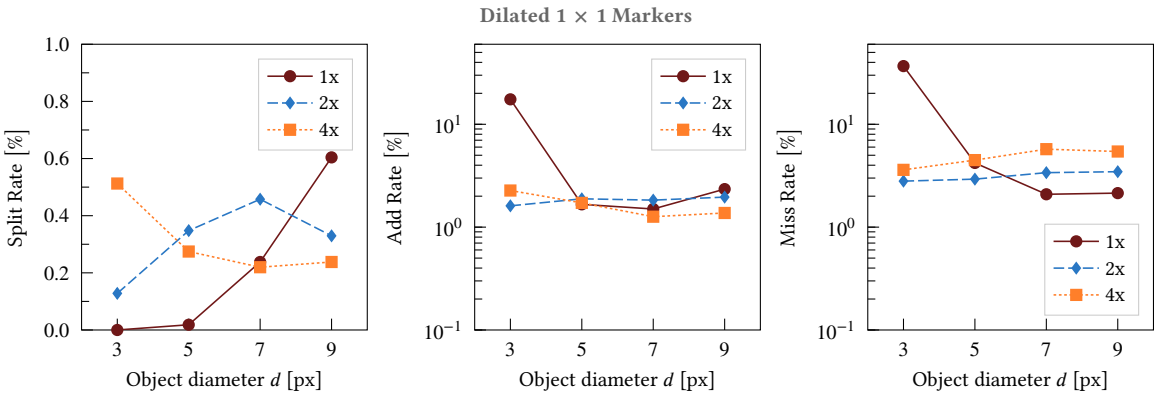
Summarized, 1×1 markers provide the highest precision scores but suffer from a relatively low recall score. Furthermore, upsampling is urgently needed for detecting objects with a diameter smaller than 5 px. The 3×3 markers also need an upsampling for objects with a diameter of 5 px. However, besides for the 1×1 markers, upsampling of large particles shows no decreased performance.

Split Rate, Add Rate, and Miss Rate

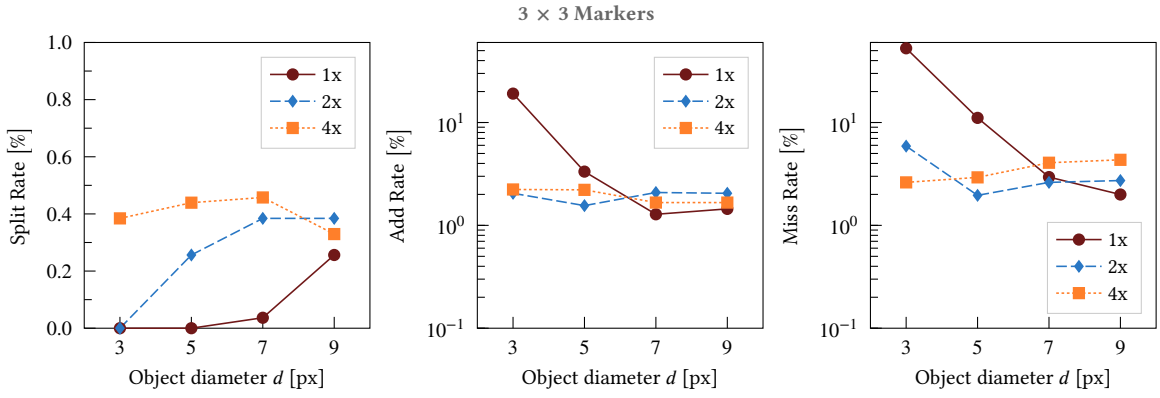
For further analysis of the error sources, the split rate, the add rate, and the miss rate are analyzed. The split rate is the amount of split particles (multiple predictions within a ground truth area) divided by the number of ground truth particles and can be interpreted as a probability to split particles. Correspondingly, the add rate and miss rate are defined using added particles (predicted particles outside the ground truth areas) and missing particles (ground truth areas without any predicted particle). Thus, the split, add, and miss rates are closely related to precision and recall but



a) Split Rate (left), Add Rate (center), and Miss Rate (right) for Particle Detection with 1 × 1 Markers



b) Split Rate (left), Add Rate (center), and Miss Rate (right) for Particle Detection with Dilated 1 × 1 Markers



c) Split Rate (left), Add Rate (center), and Miss Rate (right) for Particle Detection with 3 × 3 Markers

Figure 2.8: Split, Add, and Miss Rates for Different Upsampling Strategies. Each value represents the median of three trained models. Stated are the particle diameters d before the upsampling. All settings have been evaluated on the high-resolution ground truth with a search radius r_s of 7 px. Note: the split rate plots have a different y-axis scaling.

may give some further insights. Merge rates cannot be calculated since there is no way to define a merge when predicting only particle markers. Furthermore, it should be considered that merging two or more particles can result in a false positive since the center of the merged particles can be outside the ground truth areas of the single particles. Multiple misses also occur in this case since there are no predicted particles in the ground truth areas.

Figure 2.8 shows the median split, merge and add rates for the single evaluated settings. Surprisingly, the 1×1 markers have lower split rates than the dilated 1×1 markers and the 3×3 markers. A possible reason could be that the trained models have a bias to predict only easy-to-detect particles without neighboring particles. This would also explain the higher miss rates. Overall, the split rates are low for all settings and the precision scores are, therefore, mainly limited due to added particles. Furthermore, there is no clear trend visible in the split rates, besides a high rate for 4x upsampling.

The add rates without upsampling are above 10 % for 3 px diameter particles for all three methods. This is likely due to merging, as described above. The add rates decrease with increasing particle size for the 1×1 markers. Thus, the small marker size seems not to come at the cost of high add rates for large particles. Applying upsampling, the add rates for the dilated 1×1 markers and the 3×3 markers are around 2 %.

The miss rates reveal that the 1×1 markers are not competitive and suffer from many not or not precisely enough detected particles. Without upsampling, the use of dilated 1×1 marker leads to about 40 % of the 3 px diameter particles being missed. In addition, for the 3×3 markers more than 50 % of the 3 px diameter particles, and about 10 % of the 5 px diameter particles are not detected without upsampling. For the 3 px diameter particles, even 2x upsampling is not sufficient. These high miss rates should again result from merging markers in the post-processing due to a too large marker size or a too small particle resolution.

Qualitative Localization Inspection

The qualitative particle detection results in Figure 2.9, which shows a small test image region with predictions, give further insight into the error sources. All errors in this test region are due to false negatives. This behavior supports the low recall scores and high miss rates of the quantitative analysis. The particle localization improves with increasing particle size. Comparing the dilated 1×1 markers without and with 2x upsampling shows that the upsampling also increases the localization.

Interestingly, the different models seem to have the same tendency when not predicting the exact ground truth centroid position. Maybe, the models surpassed, in those cases, the ground truth annotation accuracy. Overall, the dilated 1×1 markers and the 3×3 markers provide high-quality particle detection results when coupled with an adequate upsampling pre-processing step for small particle sizes.

2.3 Software: BeadNet

Accurate traditional particle detection methods and tools are highly specialized to specific imaging conditions and particle sizes and must be adapted when acquisition settings or experimental parameters change. Unfortunately, these adaptations require expert knowledge of the underlying method. Furthermore, complex detection tasks like low-resolution fluorescent bead detection require a deep learning approach. However, training and applying such an approach involves a pipeline consisting of training data creation, data handling and loading, model training, model

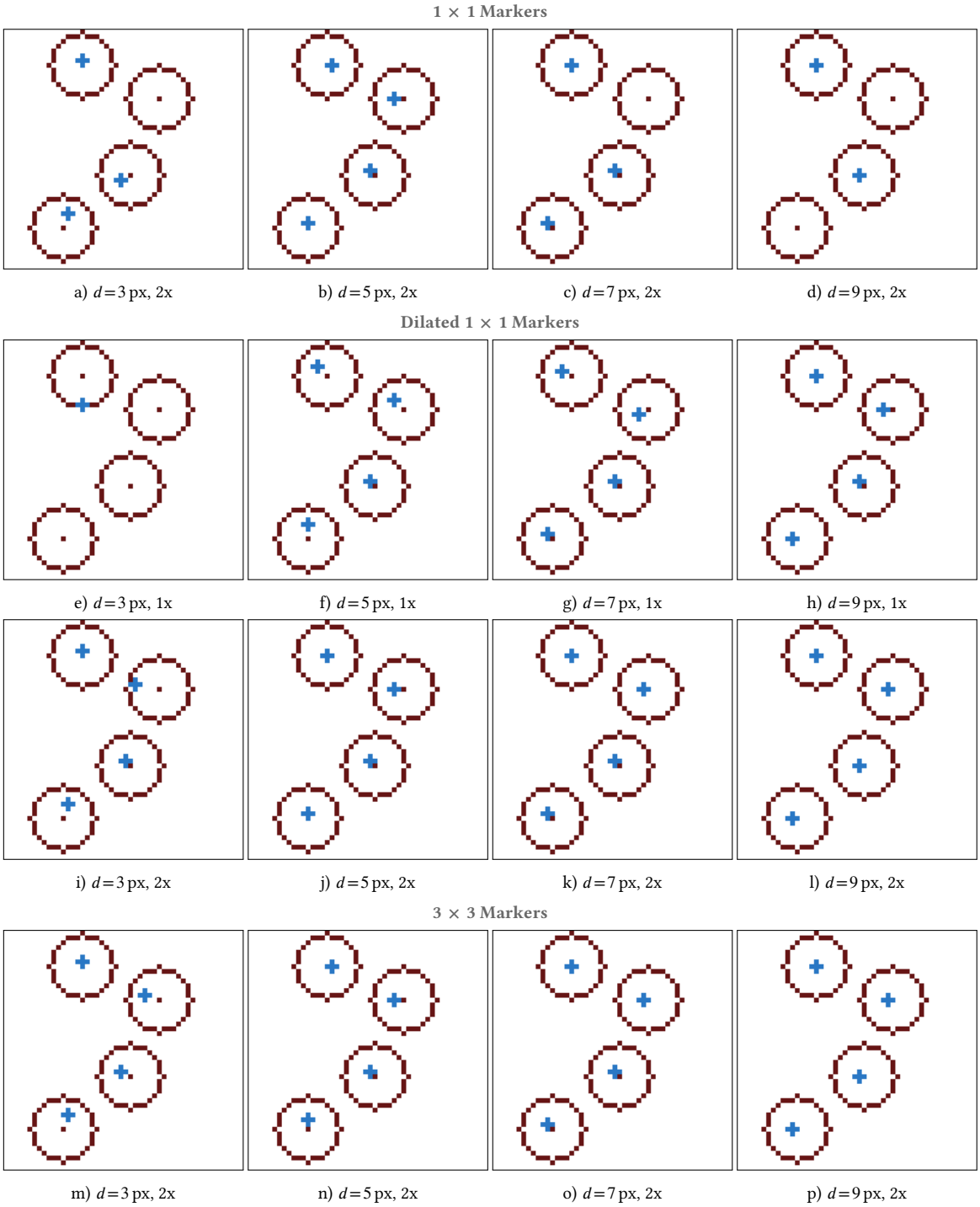


Figure 2.9: Qualitative Particle Detection Results. Shown is a small, high-resolution test image region with ground truth markers and areas (ground truth search radius $r_s = 7$ px), and to the initial resolution upsampled predictions (median models). The main error source of 1×1 markers is false negatives. The particle localization of the dilated 1×1 and 3×3 markers is similar (2x upsampling). The comparison of no upsampling (1x) and 2x upsampling for the dilated 1×1 markers shows that the upsampling enables a more accurate prediction of the particle centroid positions. Stated are the particle diameters d before the upsampling. ■ – ground truth centroid, + – predicted centroid, ○ – ground truth area.

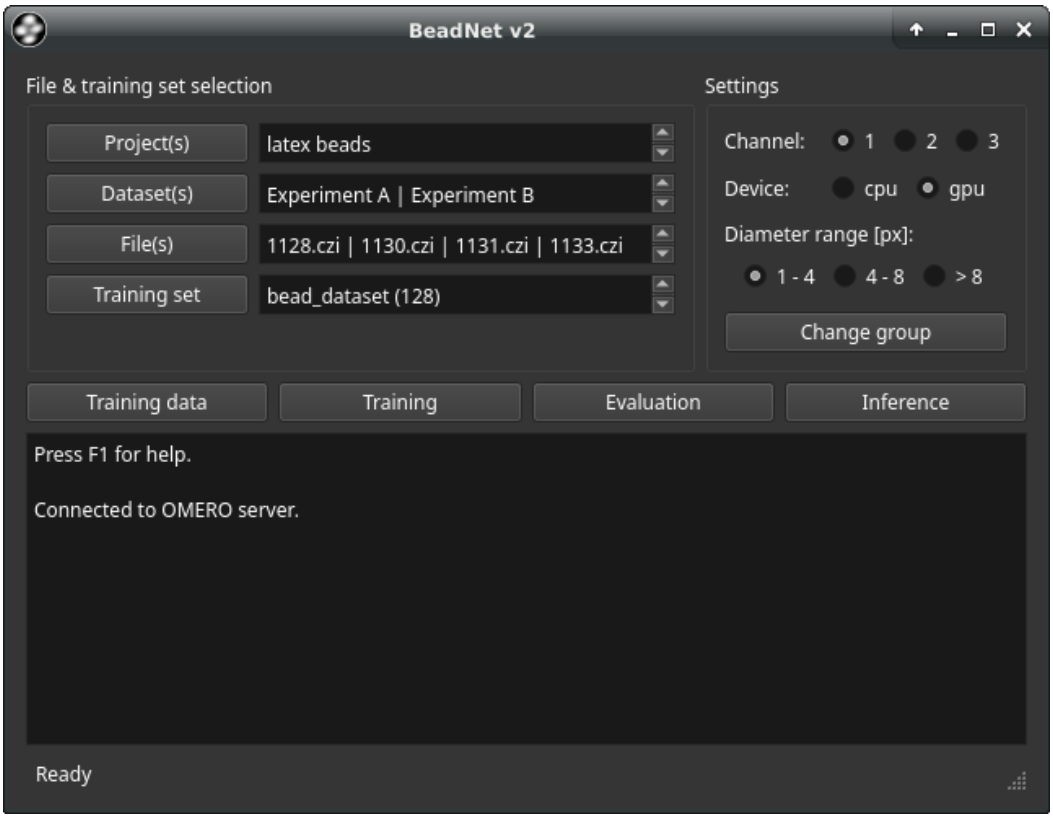


Figure 2.10: Graphical User Interface of BeadNet. The minimalist design with the pre-defined parameters and the simple workflow of BeadNet enable using deep learning for particle detection without expert knowledge. A manual with videos is available at https://github.com/TimScherr/BeadNet-v2.

evaluation, and finally, the application of trained models to experimental data. Integrating all these needed steps in an easy-to-use toolbox is desirable from a user perspective. In particular, the data management, the training data creation, and the training of the deep learning models need to be user-friendly and time efficient.

BeadNet is a new deep learning-based particle detection tool with OMERO data management. Users only need to select a particle diameter range, which sets the applied upsampling strategy, and one of the implemented particle marker representations. The other parameters are pre-defined and work well for many applications. Figure 2.10 shows the graphical user interface of BeadNet. All particle detection experiments in this thesis have been performed with BeadNet.

2.3.1 OMERO Data Management

OMERO is an open-source software platform from the Open Microscopy Environment for accessing and using a wide range of microscopy data [219]. Over 150 image formats can be imported with the OMERO.insight desktop client. Imported data are organized into projects and data sets. After the import, the data can be processed with BeadNet – without any data format conversion steps or programming. Images, BeadNet training data, and BeadNet results can easily be accessed and viewed in the browser with the OMERO.web client.

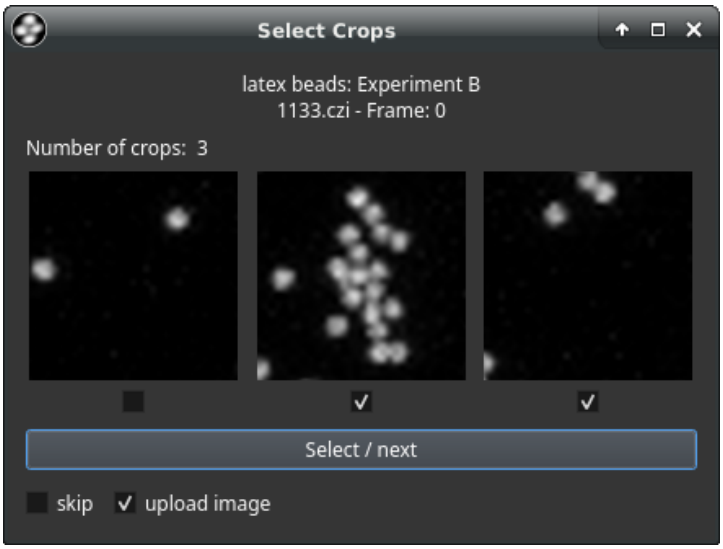


Figure 2.11: BeadNet Crop Selection Interface. Automatically proposed crops can be selected and uploaded to OMERO. The crop proposals are extracted randomly from different non-overlapping image regions (the left crop originates from the left image region, and the right crop from the right).

2.3.2 Training Data Creation

So far, own, application- and domain-specific training data are the best choice for many applications. Thus, an easy but comprehensive training data creation workflow is required without the time-consuming need for acquiring large training data sets. In the BeadNet workflow, training data sets are managed as OMERO data sets, and new training data sets can be added using the graphical user interface. A crop size must be selected when adding a new training data set. This crop size selection is required to avoid the time-consuming annotation of all, possibly densely packed, particles in an image. Annotating smaller crops with fewer particles is more efficient since more image and particle features can be covered in the same annotation time, resulting in a more diverse training data set. Therefore, BeadNet offers a crop selection interface (see Figure 2.11).

Up to three – depending on the image and crop size – crop proposals of the selected files can be viewed and added to the training data set. The applied upsampling depends on the chosen particle diameter range: 4x upsampling for 1 px to 4 px, 2x upsampling for 5 px to 8 px, and no upsampling for particles larger than 8 px. The crops are randomly extracted from non-overlapping image regions and are automatically assigned to a training, a validation, or a test subset. Selected crops are uploaded to OMERO and can be viewed and annotated with the OMERO.web client. Point regions of interest allow joint storage of images and annotations. Furthermore, annotated data sets can be imported, e.g., publicly available training data sets.

2.3.3 Model Training and Evaluation

Particle Detection Marker Representations

Five marker representations are implemented in BeadNet: (i) 1×1 markers, (ii) dilated 1×1 markers, (iii) 2×2 markers, (iv) dilated 2×2 markers, and (v) 3×3 markers. All methods use the U-Net shown in Figure 2.1 with 8.5 million trainable parameters. The network size is automatically reduced to a minimum of 0.5 million trainable parameters if not enough memory is available.

Training

The marker representation, the batch size, the optimizer, and how many models are trained need to be specified for training a model on a selected annotated training data set. Adam and Ranger [220] are available as optimizers, each with pre-defined settings, i.e., start learning rate, minimum learning rate, and learning rate scheduler (see Table 2.1). Adam is coupled with the ReLU activation function, while the Mish activation function [221] is used for the training with Ranger. Furthermore, users can reduce the batch size if memory availability is limited. The augmentations described in Section 2.1 are applied during training. The optimization criterion is the $2:1$ weighted sum of the cross-entropy loss and Dice loss (see Eq. 2.1). After training, the model checkpoint with the best validation loss is saved locally and available in BeadNet.

Evaluation

Trained models can be evaluated on the test subset, which is automatically split during the training data creation. Therefore, a radius needs to be selected, which defines the ground truth areas needed to count true positives, false positives, and false negatives. Then, the number of true positives, false positives, splits, and false negatives is estimated, and the evaluation measures precision, recall, and F_1 score are calculated for each image. Those scores are saved in a separate CSV file for each model. The average precision, recall, and F_1 score and the split, add, and miss rates on the whole test subset are stored in another CSV file. In addition, the ground truth areas overlaid with the predicted particle centroids are saved for visual inspection.

2.3.4 Inference and Result Export

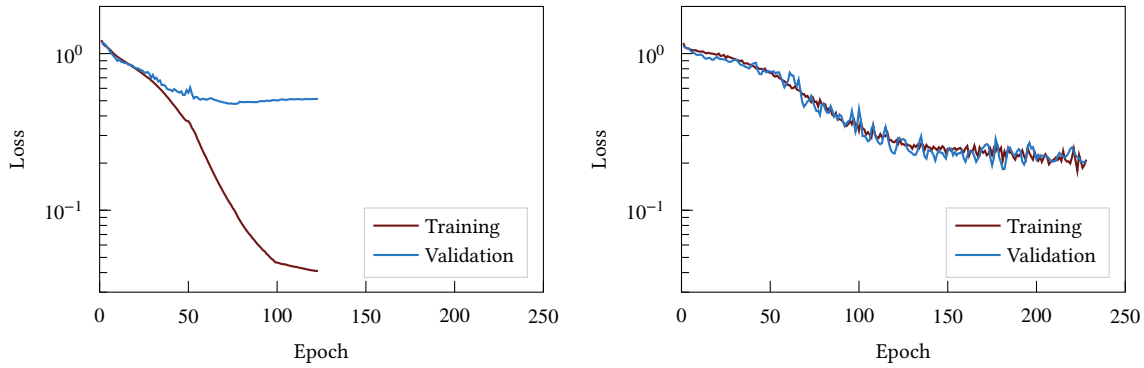
The best F_1 score model is selected automatically for inference, but the user can also select the model manually. The particle detection results are attached as point regions of interest and in a CSV file to the corresponding OMERO image and can be viewed with the OMERO.web client. The result export includes the original image (.tif), the upsampled image (.tif), particle centroid masks (.tif), upsampled image overlaid with predicted centroids (.tif), and the result CSV file.

2.3.5 Implementation, Installation, and Dependencies

BeadNet is implemented in Python and uses PyTorch as deep learning framework. The graphical user interface is built with PyQt. The software, the source code, and a manually annotated bead data set are available under the MIT license at https://github.com/TimScherr/BeadNet-v2. A detailed step-by-step guide enables easy installation and usage. BeadNet requires an OMERO server. A demo server account can be requested from the OME team in Dundee: http://qa.openmicroscopy.org.uk/registry/demo_account/.

Table 2.1: BeadNet Training Parameters. The learning rate lr is multiplied with γ when the validation loss has not decreased for N_{patience} epochs until the minimum learning rate lr_{min} is reached. The training process stops after N_{max} epochs or N_{stop} epochs without validation loss improvement. The model checkpoint with the best validation loss is used for further analysis.

Optimizer	lr_{start}	lr_{min}	γ	N_{max}	N_{patience}	N_{stop}
Adam	$8 \cdot 10^{-4}$	$3 \cdot 10^{-6}$	0.25	see Eq. 2.4	$\frac{1}{20} N_{\text{max}}$	$2N_{\text{patience}} + 5$
Ranger	$6 \cdot 10^{-3}$	$4.5 \cdot 10^{-4}$	0.25	see Eq. 2.4	$\frac{1}{10} N_{\text{max}}$	$2N_{\text{patience}} + 5$



a) Without data augmentation

b) With data augmentation

Figure 2.12: Exemplary Training and Validation Loss Curves. The data augmentation prevents overfitting, i.e., from the divergence of training and validation loss. Shown are the training and validation losses of the median models with and without augmentation trained on a data set annotated in 15 minutes. Both models reached the early stopping criterion.

2.3.6 Workflow Evaluation

The key features of BeadNet are the use of the data management OMERO with its versatile data importer and the optimized workflow with a state-of-the-art particle detection method. An experiment with a limited training data creation time of 15 minutes has been performed to prove that high-quality results can be obtained quickly and that no extensive training data sets are needed. The 3 px diameter particles also used for the upsampling strategies and marker representations study in Subsection 2.2.1 have been selected for this study. A crop size of 128×128 px and 2x upsampling have been set in BeadNet.

During the 15 minutes, 14 images – 8 training, 3 validation, and 3 test images – with 972 particles have been annotated. Three dilated 1×1 markers models have been trained on the training and validation images. The median F_1 score on the test subset from Subsection 2.2.1 is 0.959, slightly lower than the score of 0.978 of the corresponding model trained on about ten times more particles (see Figure 2.6b). Furthermore, three models have been trained without training data augmentation. The median F_1 score without augmentation is 0.870. Figure 2.12 shows the loss curves of the median model trained with augmentation and the median model trained without augmentation. In particular, the miss rate drops from 22 % to 6 %, and the add rate from 7.8 % to 1.7 % with augmentation. Thus, the training data augmentation successfully prevents overfitting on this rather small training data set. The median BeadNet model with augmentation trained 131 s. So, high-quality results can be obtained in less than 20 minutes on the used data set.

2.4 Applications

Figure 2.13 shows exemplary use cases of BeadNet and illustrates the broad applicability to various microscopy imaging techniques and particle types with different sizes. After describing these use cases shortly, the novel particle detection method is applied to a manually annotated, real-world fluorescent latex bead data set. The comparison with traditional image processing methods on this data set gives further insights into the strengths of the method.

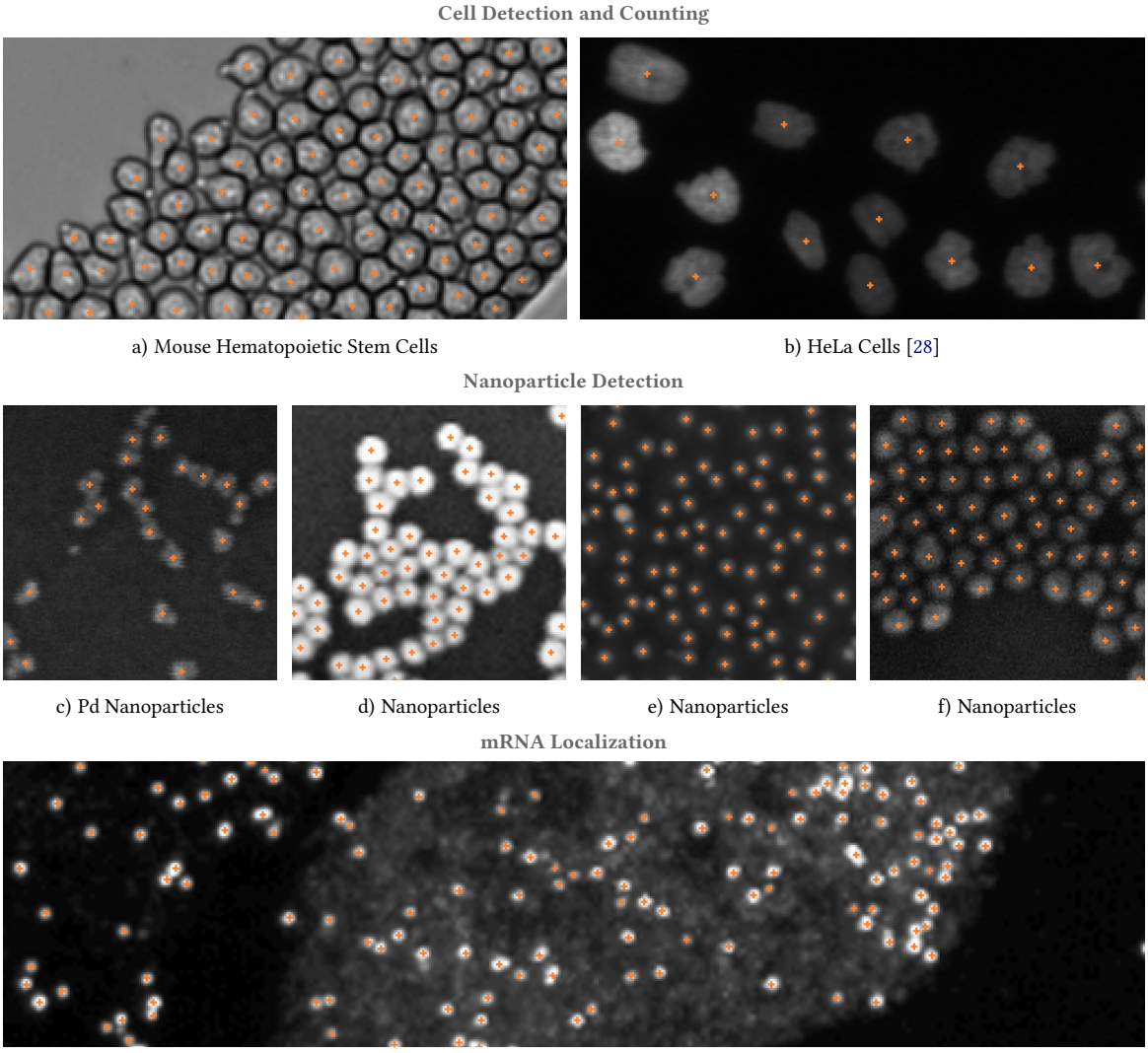


Figure 2.13: BeadNet Applications. Among other applications, BeadNet can be used for the detection and counting of cells (a, b), nanoparticles (c-f), and mRNA molecules (g). Shown results have been produced with BeadNet models that have either been trained on the publicly available BeadNet bead data set (c-f), or on training data similar to the shown applications (a, b, g). Data sources: a, b – Cell Tracking Challenge [27], [28]; c – [85], d-f – NFFA-EUROPE - 100% SEM Dataset [222], [223]; g – FISH-quant [96], [115]. + – predicted centroid position.

2.4.1 Counting of Cells and Cell Nuclei

Particle detection is helpful for biomedical analyses, which require only cell and cell nuclei positions and counts but no shape information, as particle detection training data can be annotated quicker than instance segmentation ground truth. Figure 2.13a shows BeadNet results on bright field images of mouse hematopoietic stem cells, and Figure 2.13b the detection of HeLa cells in fluorescence microscopy images. Therefore, a BeadNet model has been trained for each case. The software works well on both data sets and its application requires no expert knowledge.

2.4.2 Nanoparticle Detection

Nanoparticles are, for instance, used in the organic synthesis of nanosensor arrays [224]. Understanding the structural arrangements of metal nanoparticles on the surface of catalytic materials is the key to further optimization [85]. Figure 2.13c to Figure 2.13f show BeadNet detection results of scanning electron microscopy images of nanoparticles. No application-specific nanoparticle training data have been used to produce the shown results. Applying a BeadNet model that has been trained on the publicly available BeadNet bead data set reaches sufficient accuracy. Thus, for some application no time needs to be spent to annotate data. However, more likely is to create some application-specific training data and add it to the bead data set to train a more robust model.

2.4.3 mRNA Localization

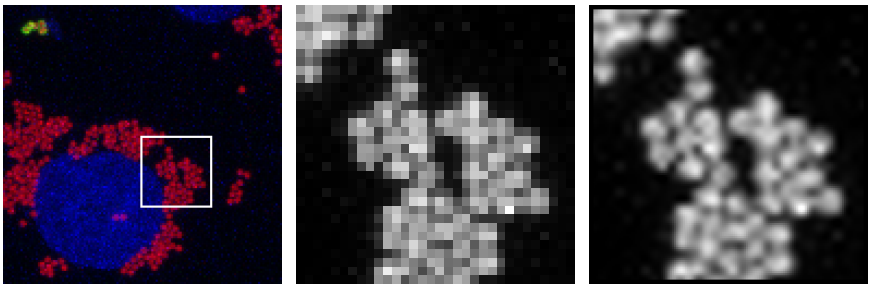
As the final exemplary qualitative use case shows Figure 2.13g the localization of mRNA molecules visualized by single molecule fluorescence *in situ* hybridization. So far, the localization of RNA is not fully understood and requires a fully automated and robust image analysis [115]. After creating some training data, BeadNet can also be used for this kind of data and reduce the time to insight for RNA localization.

2.4.4 Bead Detection

Ligand-coupled beads are used in *in vitro* experiments to mimic bacterial invasion processes [84], [225], [226]. Figure 2.14 shows red fluorescent latex beads of 1 mm size that were chemically coupled with a bacterial surface ligand to investigate their internalization into human cells [30]. The cells were fixed without permeabilization, and external beads can be distinguished from internalized beads using a ligand-specific antibody coupled to a green fluorophore. The evaluation of such experiments requires a reliable bead detection and counting.

Data Set

The with the BeadNet source code available bead data set has been annotated together with a life science expert and consists of 60 training, 15 validation, and 25 test images with 2587 beads [30]. The bead diameters range from about 2 px to 4 px. Acquired microscopy images have been upsampled based on these particle sizes (4x) resulting in particle diameters between 8 px and 16 px. A 2x upsampling would have also been sufficient for obtaining accurate results like Figure 2.6 shows for the synthetic data, but the 4x upsampling facilitates the annotation process as shown before (see Figure 2.2) and has no drawbacks besides a higher computation time. In addition, some compared methods benefit from the larger upsampling. The upsampled images have a size of 128×128 px and Figure 2.14c shows an exemplary upsampled test image.



a) Maximum Intensity Projection

b) Test image (1x)

c) Test image (4x)

Figure 2.14: Ligand-Coupled Beads. Red fluorescent latex beads are chemically coupled with a bacterial surface ligand to study their internalization into human cells. An additional green fluorophore is used to mark beads outside of cells. Cell nuclei are blue fluorescent. A closer look shows the low resolution of the beads (b). A 4x upsampling has been applied for the training and test data annotation (c). Taken from [30].

Compared Methods

The deep learning-based particle detection by the semantic segmentation of particle markers is compared with the traditional methods (i) Hough transform [91], [92], (ii) Otsu thresholding [90] with a distance transform post-processing, and (iii) TWANG [126] on the bead data set. All methods are applied to the upsampled bead images and have been tuned manually for the best possible bead detection results. The implementations and used parameters are stated in the following.

The Hough transform is a traditional method that can be used to find approximately circular objects in images. This thesis uses MATLAB's Hough transform-based spherical object detector implementation `imfindcircles`. The adjusted parameters of `imfindcircles` are radius range: 5 px - 9 px, sensitivity: 0.9, method: TwoStage, and edge threshold: 0.1. The accuracy of `imfindcircles` is limited when the particle radius is less than or equal to 5 px. Thus, the Hough transform requires the use of the 4x upsampled images.

Otsu's method is a straightforward and easy-to-use method without parameters. However, the threshold obtained by Otsu's thresholding method is enlarged for the bead data to reduce the merging of adjacent beads. Therefore, the threshold is multiplied with a manually adjusted factor of 1.4. In addition, a Euclidean distance transform post-processing with bead centroid extraction is applied to prevent further merging of beads.

TWANG is a segmentation method developed for 2D and 3D cell nuclei segmentation but the Laplacian-of-Gaussian-based seed detection can also be used for bead detection. A to the 4x upsampled bead images adjusted XPIWIT pipeline with median filter pre-processing is used for the method comparison [227]: $\sigma_{\text{min}} = 2$, $\sigma_{\text{max}} = 3$, $\sigma_{\text{step}} = 1$, NeighborhoodRadius: 3, StdDevMultiplicator: 1, AllowMaximumPlateaus: True, FuseSeedPoints: True.

Results

Table 2.2 shows the quantitative results for the 25 test images containing 670 beads. As mentioned before, a border correction has been applied to minimize the influence of only partially visible beads. The particle diameter ranges from 8 px to 16 px in the upsampled images. A radius of 3 px is used to define the ground truth areas required for counting false positives or added particles, false negatives or missed particles, and splits. Here, splits are the number of additional particle detections within a ground truth area.

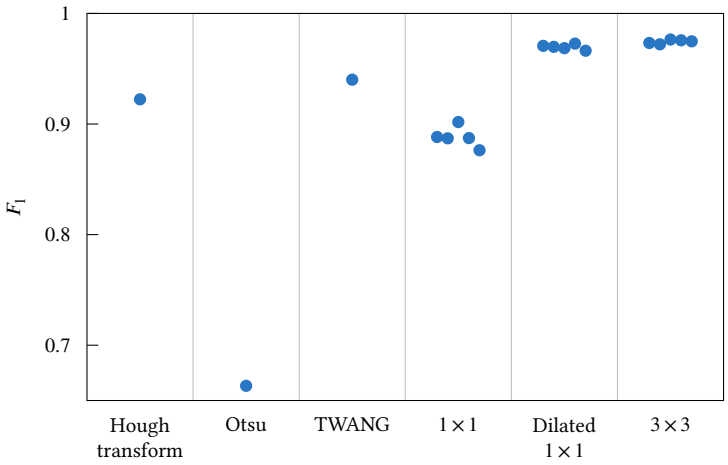


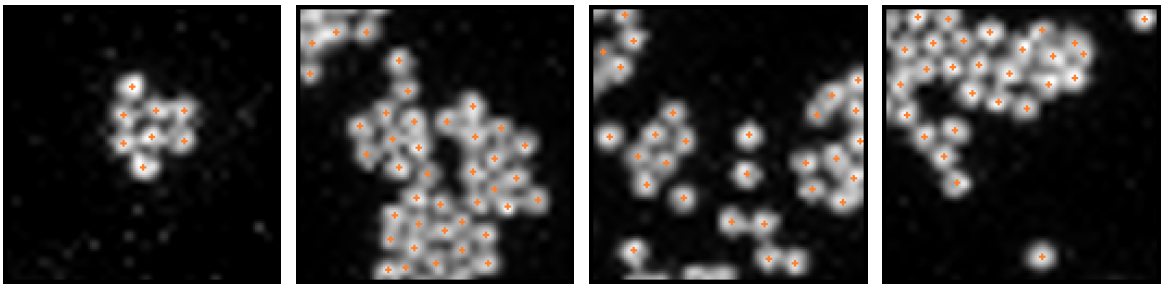
Figure 2.15: Bead Data Set Results with Single Deep Learning Model Initializations. The training process of the dilated 1×1 and the 3×3 markers is very reliable (five trained models per marker representation, the other methods are deterministic). Thus, both methods constantly outperform the other methods on the bead data test subset and are close to the perfect F-score of 1.

TWANG is the traditional method with the highest recall and F_1 score. However, about 4 % of the detections are still outside of a ground truth area, and about 8 % of the beads are not detected or not precisely enough and, therefore, do not lie within a ground truth area. The Hough transform provides a better precision but misses about 13 % of the particles. For both methods, splits are no issue. The simple Otsu method is not competitive on the bead data set and merges many beads resulting in a high miss rate of 44 %.

The dilated 1×1 and the 3×3 markers are superior to the 1×1 markers, which supports the results of the former validation in Subsection 2.2.1. Figure 2.15 demonstrates that each trained dilated 1×1 and 3×3 markers model outperforms all traditional methods. Furthermore, no evaluation measure shows an advantage of the traditional methods. The 3×3 markers produce slightly better results than the dilated 1×1 markers, and the small differences between the single models indicate a robust and reliable training process for both methods. The qualitative bead detection results in Figure 2.16 confirm a good detection and localization of the beads. Using 1×1 markers results in a high miss rate of 18 %. In addition, the differences between single models are pretty high. This variance may be due to the missing robustness to annotation errors or inconsistencies between the annotators and, therefore, an unstable training process. The small marker size also results in splits, contrary to the former validation results.

Table 2.2: Bead Data Set Detection Results. All methods have been evaluated with a ground truth radius of 3 px on the 25 test images containing 670 beads. The median scores out of five trained models are shown for the deep learning-based marker prediction methods. The other methods are deterministic.

Method	Precision	Recall	F_1	Split Rate	Add Rate	Miss Rate
Hough transform	0.964	0.884	0.922	0 %	3.43 %	13.13 %
Otsu	0.769	0.583	0.663	2.09 %	18.36 %	44.18 %
TWANG	0.956	0.925	0.940	0 %	4.33 %	8.06 %
1×1	0.956	0.828	0.887	2.39 %	3.28 %	18.36 %
Dilated 1×1	0.974	0.963	0.970	0.15 %	2.24 %	3.43 %
3×3	0.971	0.978	0.975	0 %	2.69 %	2.09 %



a) Bead Test Image b) Bead Test Image c) Bead Test Image d) Bead Test Image

Figure 2.16: Qualitative Bead Detection Results. The results of the median F_1 score 3×3 marker model for four test images of the bead data set are shown. + – predicted centroid position.

2.5 Discussion

Accurately detecting poorly resolved particles is a challenging task, even for humans. Without an upsampling pre-processing, no good results can be obtained for small particles using a semantic segmentation-based deep learning approach. However, the best upsampling depends on the particle size and the used marker size. 2x upsampling is suitable for all tested particle sizes when using dilated 1×1 markers. The 3×3 markers require a 4x upsampling for particles with a diameter of 3 px. For particles larger than 9 px, upsampling may negatively influence the detection accuracy, as a decline with growing particle sizes can already be seen for the 1×1 markers. Since the upsampling increases the image dimensions and, therefore, the memory demand and computation time, no upsampling is recommended for particles larger than 8 px. Overall, the dilated 1×1 markers and the 3×3 markers provide high-quality particle detection results when coupled with an upsampling step for small particle sizes. Furthermore, the upsampling improves not only the detection quality of poorly resolved particles but also the localization within the ground truth areas. Remarkably, 1×1 markers suffer on the synthetic data sets mainly from missing particles and not from splitting particles, as could be assumed. In fact, splitting particles is a negligible error source for all marker representations on the synthetic data. On the other hand, the split rate for the 1×1 markers is 2.39 % on the real-world bead data set, while the dilated 1×1 markers and 3×3 show almost no splits. This discrepancy between the synthetic and the real-world data is difficult to interpret since the add and miss rates, and the F-scores are similar.

Furthermore, the bead data set results show that the dilated 1×1 markers and the 3×3 markers outperform traditional methods like a tuned TWANG pipeline. In addition, the results of the single models show only a small variance. The high miss rate from the 1×1 markers may indicate absent robustness to annotation variability. For instance, a prediction 1 px apart from the annotated center results in a maximum penalization for this marker representation. Such inconsistencies occur commonly in real-world training data sets, especially when annotated by more than one annotator, each with a different decision boundary. However, using 1×1 markers results also results in a high miss rate for the synthetic data. The synthetic data set consists of about four times more particles than the bead data set, and the particle centroids have been extracted from manually corrected instance segmentations. Thus, there should be less annotation variability. Nevertheless, the performance difference between single trained 1×1 marker models is high compared to dilated 1×1 markers or 3×3 markers. This result indicates an unstable training process. Due to the single-pixel marker size, the reason may be a mixture of missing robustness to annotation variability, a high class imbalance, and no possibility of partially correct predictions during training. All these

reasons may hinder the optimization process and, therefore, lead to false negatives, which would explain the high miss rates.

A simple way to adapt a deep learning-based particle detection approach to other domains and applications is to annotate data and train new models. No expert knowledge of the underlying method is required for this adaptation, in contrast to many traditional image processing methods. Therefore, the new particle detection tool BeadNet enables easy-to-use, high-quality particle detection. Using OMERO for data management allows importing of over 150 image formats. After import, the formats are standardized. Thus, no explicit data format conversion steps into a supported format are needed. The pre-parameterized BeadNet workflow works well for many applications, and all particle detection results in this thesis have been produced with BeadNet. The workflow evaluation shows that accurate detection results are possible within 20 minutes, including training data creation and model training. A significant advantage of deep learning-based particle detection compared to instance segmentation is that particle detection ground truth can be annotated with one click per particle, enabling annotating more than one particle per second as in the workflow evaluation. Furthermore, the evaluation demonstrates that not millions of training data samples are needed, as stated in [103]. Finally, the qualitative applications of cell counting, nanoparticle detection, and mRNA localization indicate that BeadNet can detect different particle types and sizes acquired with different microscopy techniques.

3 Instance Segmentation

The goal of instance segmentation is to locate each object in an image and to determine the shapes of those objects. In the instance segmentation process, adjacent objects must be distinguished, an error-prone task, especially for densely packed objects and unclear object boundaries. Since manual data analysis is time-consuming, methods are needed that can robustly detect every single object. A key to successfully applying supervised deep learning methods in the absence of large training data sets is learning data representations that robustly encode instance information. Thus far, an issue of many deep learning approaches is that no or only adjacent objects provide neighbor information in the training process. Therefore, this chapter presents and validates novel data representations for encoding neighbor information, i.e., adapted border maps and neighbor distance maps. These representations aim to solve the challenging problem of distinguishing adjacent objects without large training data sets. Furthermore, application cases of the newly developed instance segmentation software microbeSEG are presented. This chapter builds upon [150], [167], [168].

3.1 Double-Decoder U-Net-Based Instance Segmentation

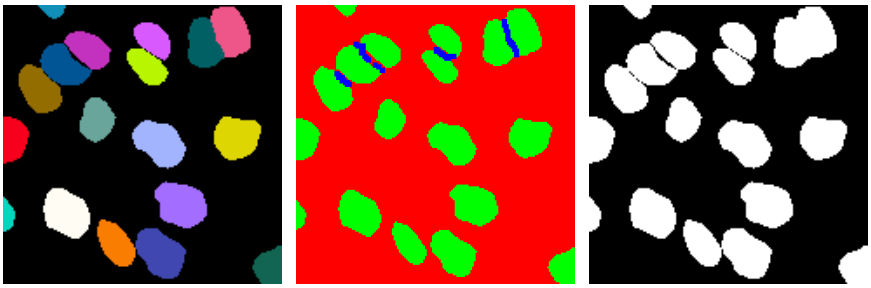
In this thesis, the CNN predictions of a neighbor information data representation are combined with the predictions of another object information data representation in a watershed-based post-processing. Therefore, a U-Net with two decoder paths, one for each representation, is trained. The used neighbor information encodings and data representations, the double-decoder U-Net architecture, the training process, and the inference are described in this section.

3.1.1 Robust Encoding of Neighbor Information

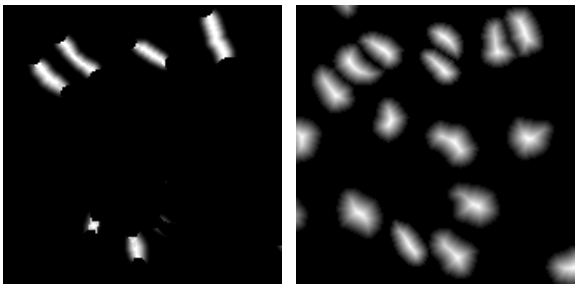
As mentioned before, instance segmentation requires the distinction of adjacent objects. However, sometimes only a few adjacent objects are in the training data, e.g., if mainly the first frames of a growing microbial colony are annotated to save annotation time or if most objects do not touch. This lack of many adjacent objects and their under-representation complicate learning the separation of objects. Nevertheless, a virtually error-free segmentation is also in these cases required and often cannot be resolved by simple data augmentation techniques alone. In Chapter 1, a potential merge error of simple semantic methods that do not utilize a neighbor information encoding, e.g., the prediction of the three classes background, object interior, and object boundary, has been shown (Figure 1.12). For instance, this behavior of misclassifying pixels between adjacent objects can be seen in [150]. Thus, a robust encoding of neighbor information is required to help the CNN learn the separation of adjacent objects.

Adapted Border Representation

A simple adaptation to solve the problem of missing boundary pixels in the interface of adjacent objects when using a semantic method with three or more classes is predicting just the interface of



a) Ground Truth b) Adapted Border c) Binary



d) Neighbor Distance e) Object Distance

Figure 3.1: Data Representations for Instance Segmentation. The multi-channel adapted border label (b) consists of dilated borders (blue), eroded object interiors (green), and background (red). Since the improved object separability means a loss in shape information, the adapted border representation is combined with a binary background-foreground representation (c). The neighbor distance representation utilizes normalized inverse distance to another object (d). The neighbor distance maps are combined with normalized object distance maps (e). The ground truth is from the Cell Tracking Challenge data set Fluo-N2DL-HeLa [27], [28].

them. However, these single-pixel thick borders are not very robust, and single-pixel misclassifications can break the post-processing [150]. The novel adapted border representation overcomes this problem with two further adaptations: (i) morphological dilation and closing of the border class with a 3×3 kernel and (ii) morphological erosion of the object interior class with a 3×3 kernel. Due to the dilation, the borders are thicker and longer. The erosion of the object interior further improves the separation of adjacent objects, as Figure 3.1b shows. In addition, the class imbalance is reduced compared to simple borders. Thus, a slightly improved training process is expected, besides the enhanced robustness to imperfect border predictions.

Since the eroded object interior class does not reflect the initial object shape, the three-class adapted border representation is combined with a two-class foreground-background representation (see Figure 3.1c). The resulting method, referred to as the adapted border method in the following, belongs to the semantic method category.

Neighbor Distance Map Representation

The adapted border method aims to minimize merging adjacent objects due to missing border pixels. However, learning high-quality border predictions depends on the availability of adjacent objects in the training data set. For instance, objects that do not touch but are separated by two background pixels do not contribute border information in the training process. In the neighbor distance map representation, each pixel of an object represents an inverse normalized distance to the nearest pixel of the nearest surrounding object (Figure 3.1d). Thus, also non-touching objects can contribute to learning this neighbor information encoding in the training process.

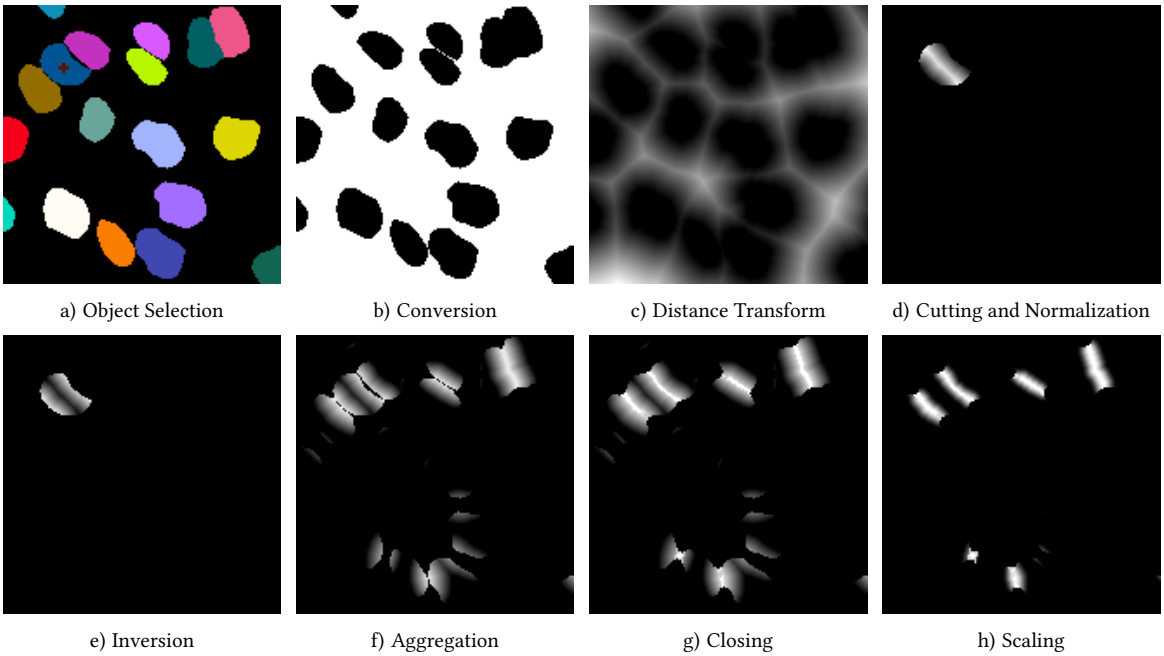


Figure 3.2: Neighbor Distance Map Creation Steps. A selected object (a, indicated with a red cross) and the background are converted to foreground while the other objects are converted to background (b). Then, the Euclidean distance transform is calculated (c), cut to the object shape, and normalized (d). After the inversion (e), these steps are repeated for each object (f). Finally, a closing (g) and a scaling step (h) are applied. Note: Calculating the distance transform (c) only for a small region around the selected object is more efficient.

Figure 3.2 shows the generation of the neighbor distance maps. A background-foreground conversion step is applied independently for each object in an image (Figure 3.2b), and the Euclidean distance transform is calculated (Figure 3.2c). Then, the distance transform is cut to the shape of the selected object and normalized (Figure 3.2d). For the normalization, the cut distance transform is divided by the minimum of its maximum value and the maximum value of its corresponding object distance (Figure 3.1e). The minimum operation has technical reasons and is needed to avoid artifacts for elongated objects. After normalization, the normalized cut region is inverted (Figure 3.2e) and the construction steps are repeated for each object. Finally, gaps are closed in a bottom-hat-transform-based closing step with a disk-shaped kernel with a radius of 3 px (Figure 3.2g). The bottom-hat-transform is only considered in regions with neighbor information. Finally, a scaling step refines the neighbor distance maps, which range from 0 to 1 (Figure 3.2h).

The neighbor distance maps encode only neighbor information but no shape and instance information. Thus, they are combined with normalized object distance maps (see Figure 3.1e). These maps are generated by independently computing the Euclidean distance transform for each object, treating other objects as background. So, each pixel of an object represents the distance to the nearest pixel not belonging to this object. Object distance maps alone can also be used for instance segmentation, e.g., the DIST method proposed in [139] using non-normalized distances.

However, combining object and neighbor distance maps and normalizing them into the range $[0, 1]$ should have several advantages: (i) the normalization allows using a simple threshold-based post-processing instead of a local maxima-based post-processing, (ii) subtracting the neighbor distance maps from the object distance maps improves the separation of object instances, and (iii) the neighbor information can be used to avoid the wrong splitting of, for instance, dumbbell-shaped

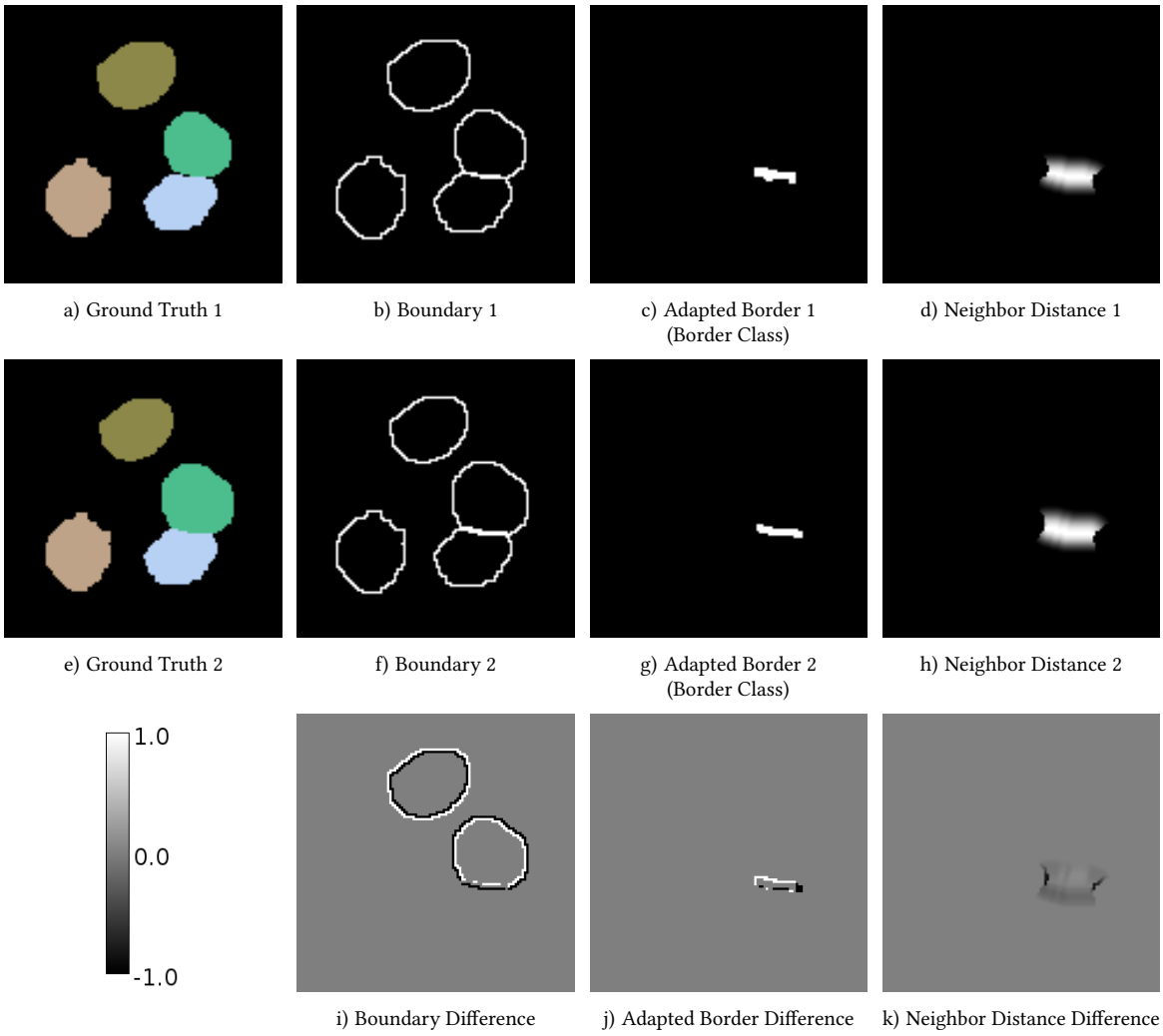


Figure 3.3: Robustness of Data Representations to Annotation Variations. Even small changes in the ground truth, simulated with morphological operations, result in different object boundaries (i). However, the adapted borders have some overlap, and the neighbor distances show a smooth change, which should be beneficial during training.

objects occurring when using a local maxima-based or threshold-based post-processing. This novel combination of object and neighbor distance maps, referred to as the distance method in the following, can be interpreted as a regression version of a semantic method with the three classes background, object interior, and object border.

Robustness to Annotation Variability

CNNs for instance segmentation are commonly trained on data with human-made annotations. Depending on the goal and resources, either multiple annotators annotate the same objects and combine the annotations, multiple annotators annotate different objects, or a single annotator annotates the data. However, unclear decision boundaries of a single annotator and different decision boundaries between annotators lead to variability in the annotations, e.g., in the shape or size of annotated objects. Thus, a neighbor encoding should be robust to single-pixel mispredictions and variability in the training data.

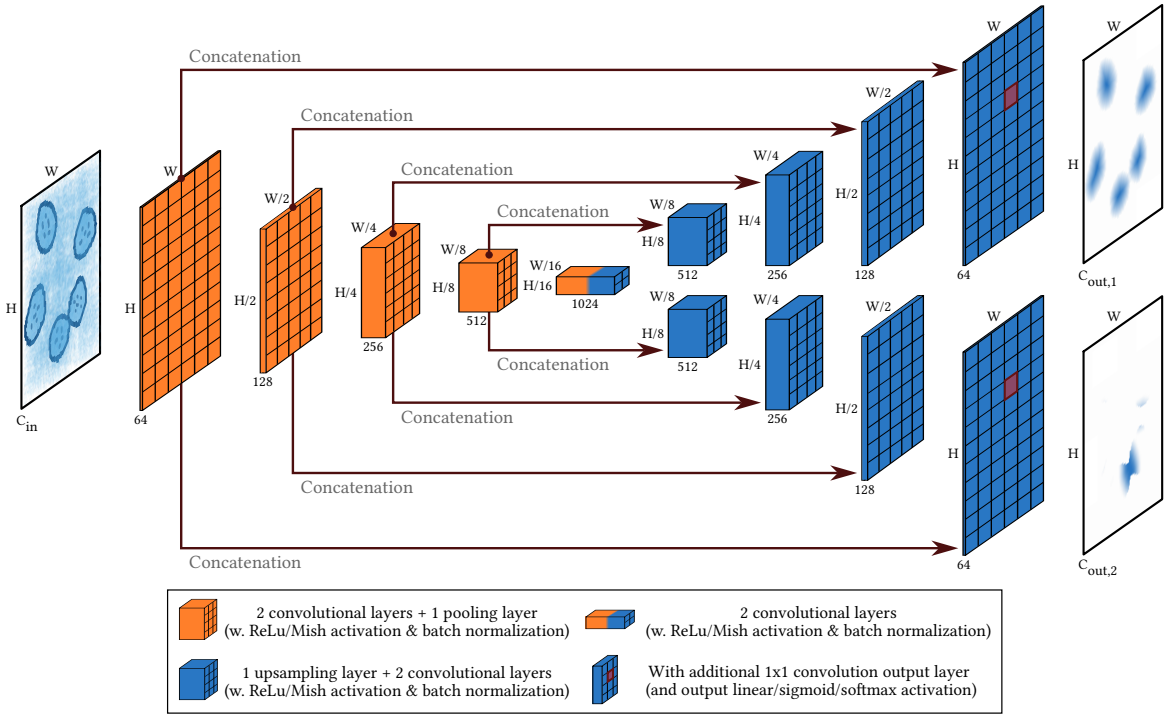


Figure 3.4: Double-Decoder U-Net. The U-Net has two decoder paths, one for each output of the method (adapted border or distance). Note: concatenated are the outputs of the last convolutional layer in a block and of the corresponding transposed convolutional layer. H/W/C – height/width/channel dimension. Shown are illustrations of object and neighbor distance map predictions.

Figure 3.3 shows data representations for two ground truths. The second ground truth has been generated from the first ground truth by applying a morphological dilation to one object and a morphological erosion to another. These modifications result in objects that differ only in single pixels. The boundary difference image shows that the boundaries for both ground truth images have almost no overlap for the two modified objects. Thus, a CNN must exactly reproduce an annotator's annotation to minimize the loss function, a nearly impossible scenario due to the annotation variability.

This missing robustness is probably why the single-pixel borders are often not learned well, as loss functions typically compare single pixels and highly penalize such slight differences complicating the convergence of a trained model to a sufficient minimum. In contrast, the adapted border representation shows some overlap for both ground truth images and the neighbor distance representation a smooth change. Thus, the negative influence of annotation variability on the training process should be reduced compared to boundary or single-pixel thick border representations.

3.1.2 CNN Architecture

The CNN for the adapted border method and the CNN for the distance method need to predict two data representations, i.e., adapted border and binary for the adapted border method, and object distance and neighbor distance for the distance method. Therefore, the CNN is a double-decoder U-Net. Using two paths enables each path to focus on features related to its output. In contrast, the feature extraction in the shared encoder benefits from back-propagated information from both decoder branches. Figure 3.4 visualizes the architecture.

The shown U-Net has about 46 million trainable parameters when predicting two single-channel data representations ($C_{\text{out},1} = 1$, $C_{\text{out},2} = 1$). The number of parameters slightly increases for multi-channel data representations as the parameters of the last 1×1 convolutional layer depend on the output size. The inputs of convolutional layers are zero-padded to keep the spatial feature map dimensions constant and to avoid cropping before concatenating corresponding encoder and decoder feature maps. Convolutions with stride two are used for a learnable downsampling, and transposed convolutions are used for a learnable upsampling. In addition, batch normalization layers are used. In the first convolutional layer, 64 feature maps are used. After downsampling, the number of feature maps is doubled in the subsequent convolutional layer until 1024 feature maps are reached. In turn, the number of feature maps is halved in a transposed convolutional layer and in the subsequent convolutional layer after concatenation.

The ReLU or the Mish activation function is applied to the output of convolutional layers within the CNN. In the output layer of each decoder, the linear activation is applied for the object distance map representation ($C_{\text{out},1} = 1$) and the neighbor distance map representation ($C_{\text{out},2} = 1$), the sigmoid activation function for the binary representation $C_{\text{out},1} = 1$), and the softmax activation function for the three-class adapted borders ($C_{\text{out},2} = 3$).

3.1.3 Training Process

The double-decoder U-Net is trained either with the Adam optimizer in the AMSGrad variant ($\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay) or with the Ranger optimizer ($\alpha = 0.5$, $k = 6$, $\beta_1 = 0.95$, $\beta_2 = 0.999$, no weight decay, with gradient centralization). The Adam optimizer is coupled with the ReLU activation function, while the Mish activation function is used for training with the Ranger optimizer.

For all experiments in this thesis, the batch size is set to 8, resulting in an effective batch size of 4 on each of the two used GPUs. Thus, a batch size of 4 should be appropriate when using a single GPU. A training-validation split of 80 %/20 % is used if not stated otherwise. The double-decoder U-Net is trained on 320×320 px crops extracted from min-max normalized images of an annotated data set. The use of crops and the effective batch size of 4 allow training also on consumer GPUs with 11 GiB VRAM. However, the crop size should be increased for objects larger than 100 px to make multiple complete objects visible in an image crop.

Loss Functions

The loss function to train the double-decoder U-Net for the distance method is the sum of the losses for the object distance map prediction $\hat{y}_1$ and the neighbor distance map prediction $\hat{y}_2$:

$$L_{\text{distance}}\left(y_1, y_2, \hat{y}_1, \hat{y}_2\right) = L_{\text{SL1}}\left(y_1, \hat{y}_1\right) + L_{\text{SL1}}\left(y_2, \hat{y}_2\right) \tag{3.1}$$

with

$$L_{\text{SL1}}\left(y, \hat{y}\right) = \begin{cases} \displaystyle\sum_{i=1}^{N} \frac{\left(y_i - \hat{y}_i\right)^2}{2} & \text{if } \left|y_i - \hat{y}_i\right| < 1 \text{ ,} \\ \displaystyle\sum_{i=1}^{N} \left|y_i - \hat{y}_i\right| - 0.5 & \text{else .} \end{cases} \tag{3.2}$$

y_1 and y_2 are the ground truth data representations. The smooth L1 loss L_{SL1} is less sensitive to outliers than the mean squared error loss [228] and should stabilize the training process.

Table 3.1: Instance Segmentation Training Parameters. The learning rate lr is multiplied with γ when the validation loss has not decreased for N_{patience} epochs until the minimum learning rate lr_{min} is reached. The training process stops after N_{max} epochs or N_{stop} epochs without validation loss improvement. The model checkpoint with the best validation loss is used for further analysis.

Optimizer	lr_{start}	lr_{min}	γ	N_{max}	N_{patience}	N_{stop}
Adam	$8 \cdot 10^{-4}$	$3 \cdot 10^{-6}$	0.25	see Eq. 3.7	$\frac{1}{20} N_{\text{max}}$	$2N_{\text{patience}} + 5$
Ranger	$6 \cdot 10^{-3}$	$4.5 \cdot 10^{-4}$	0.25	see Eq. 3.7	$\frac{1}{10} N_{\text{max}}$	$2N_{\text{patience}} + 5$

The loss function for the adapted border method is a weighted sum of the losses for the binary representation prediction $\hat{y}_1$ and the three-class adapted border representation prediction $\hat{y}_2$, which consists of the single channels $\hat{y}_{2,c}$:

$$\begin{aligned} L_{\text{AB}}\left(y_1, y_2, \hat{y}_1, \hat{y}_2\right) = \alpha_1 L_{\text{BCE}}\left(y_1, \hat{y}_1\right) + \alpha_1 L_{\text{Dice}}\left(y_1, \hat{y}_1\right) \\ + \alpha_2 L_{\text{CE}}\left(y_2, \hat{y}_2\right) + \alpha_2 \sum_{c=2}^{3} \frac{c-1}{2} L_{\text{Dice}}\left(y_{2,c}, \hat{y}_{2,c}\right) . \end{aligned} \tag{3.3}$$

y_1 and y_2 are the ground truth data representations and the single loss parts for an image with N pixels are the binary cross-entropy loss [215]

$$L_{\text{BCE}}\left(y, \hat{y}\right) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \cdot \log \hat{y}_i + \left(1 - y_i\right) \cdot \log\left(1 - \hat{y}_i\right) \right] , \tag{3.4}$$

the Dice loss [216]

$$L_{\text{Dice}}\left(y, \hat{y}\right) = 1 - \frac{2 \sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i^2 + \sum_{i=1}^{N} \hat{y}_i^2} , \tag{3.5}$$

the cross-entropy loss

$$L_{\text{CE}}\left(y, \hat{y}\right) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(\hat{y}_{i,c^*}\right)}{\sum_{c=1}^{3} \exp\left(\hat{y}_{i,c}\right)} , \tag{3.6}$$

and the Dice losses for the single channels $y_{2,c}$ of y_2. In Eq. 3.6, c^* is the channel of the ground truth class of pixel i and $\hat{y}_{i,c*}$ is the predicted probability of the ground truth class of pixel i. For instance, if pixel i belongs to the border class in the ground truth, then $\hat{y}_{i,c*}$ is the predicted border class probability.

The Dice loss parts aim to improve the performance on class-imbalanced data. The intention of the higher weighting of the border class (c=3) compared to the interior class (c=2) is a focus on learning a better separation of objects. The loss term weights α_1 and α_2 enable scaling the losses for y_1 and y_2 to a comparable size. This scaling avoids minimizing only the dominant loss part. In this thesis, $\alpha_1 = 1$ and $\alpha_2 = 0.7$ are used.

Learning Rate Scheduler and Stopping Criteria

Table 3.1 shows the training parameters, i.e., start learning rate lr_{start}, minimum learning rate lr_{min}, decay factor γ, maximum number of epochs N_{max}, patience N_{patience}, and early stopping criterion N_{stop}. As for the particle detection, models are trained for a maximum number of epochs N_{max}

that depends on the number n of training and validation image crops and is used for defining the patience and the early stopping criterion:

$$N_{\max} = \begin{cases} 200 & \text{if } n \geq 1000, \\ 240 & \text{if } 1000 > n \geq 500, \\ 320 & \text{if } 500 > n \geq 200, \\ 400 & \text{if } 200 > n \geq 100, \\ 480 & \text{if } 100 > n \geq 50, \\ 560 & \text{else.} \end{cases} \tag{3.7}$$

Again, the values are set empirically and provide a good trade-off between training time and model performance for many applications. After training, the model checkpoint with the best validation loss is used for further analysis.

Augmentations

The following training data augmentations are applied independently from each other in the stated order with the stated probability p to improve the generalization of a trained model to unseen data, to avoid overfitting, and to stabilize the training process:

- *flipping* ($p = 87.5\,\%$): flip (up-down, left-right), rotation by multiples of 90°, or combination,
- *contrast* ($p = 45\,\%$): histogram equalization, contrast stretching, or gamma adjustment,
- *scaling* ($p = 25\,\%$): scaling with random scale factor $s \in [0.85, 1.15]$,
- *rotation* ($p = 25\,\%$): rotation by the random angle $\alpha \in [-45°, 45°]$,
- *blurring* ($p = 30\,\%$): Gaussian blur with random $\sigma \in [1, 2]$,
- *noise* ($p = 30\,\%$): additive Gaussian noise with random $\sigma \in \left[0.01\, I_{\max}, 0.05\, I_{\max}\right]$.

$I_{\max}$ is the maximum intensity in a training image. Label-preserving augmentations like blurring modify only the training image and not the label image, e.g., of the adapted border representation. Label-changing augmentations require a transformation of the label images as well, e.g., rotation and scaling. Therefore, nearest neighbor interpolation is used for interpolating pixels in the transformed label image of the adapted border method and bi-linear interpolation for the distance method.

3.1.4 Inference and Post-Processing

Min-max normalized images are fed to the double-decoder U-Net for inference. The post-processing relies for both methods on a seed extraction step and a watershed segmentation step.

Adapted Border Method

The first step in the adapted border method post-processing is thresholding the binary representation prediction y_1 with a threshold of 0.5[1]:

$$\hat{y}_{\text{mask}} = \hat{y}_1 > 0.5\ . \tag{3.8}$$

[1]For better readability, some operators like the greater/threshold operator >, the squaring operator ^{2}, or the tan operator are applied element-wise, and from images subtracted values are matrices of the same size.

A threshold of 0.5 is sufficient due to the use of the sigmoid activation function pushing the predictions to 0 or 1. The resulting mask $\hat{y}_{\text{mask}}$ defines the object area in the processed image. Then, the object interior channel $\hat{y}_{2,2}$ of the adapted border representation prediction $\hat{y}_2$ is multiplied by the inverse of the adapted border channel $\hat{y}_{2,3}$. This multiplication yields an improved separation of objects compared to a simple subtraction of $\hat{y}_{2,3}$ from $\hat{y}_{2,2}$. Connected-component labeling (CCL) of the thresholded multiplication result yields the seeds

$$\hat{y}_{\text{seed}} = \text{CCL}\left(\left(\hat{y}_{2,2} \cdot \left(1 - \hat{y}_{2,3}\right)\right) > 0.5\right) . \tag{3.9}$$

Again, 0.5 is a suitable threshold since the predictions are probabilities in the range $[0, 1]$ due to using the softmax activation function. Finally, seeds smaller than 5 px are filtered, and a seed-based watershed is applied to fill the not yet labeled pixels in y_{mask}.

Distance Method

In the distance method post-processing, the object distance map prediction $\hat{y}_1$ is slightly smoothed with a Gaussian kernel g with standard deviation $\sigma = 0.5$, and then a threshold t_1 is applied:

$$\hat{y}_{\text{mask}} = \left(g * \hat{y}_1\right) > t_1 . \tag{3.10}$$

Similar to the adapted border method post-processing, the resulting mask $\hat{y}_{\text{mask}}$ defines the object area in the processed image. The threshold $t_1 \geq 0$ is a hyperparameter and affects the segmented object size. Theoretically, $t_1 = 0$ without smoothing yields the perfect object size. However, a threshold $t_1 = 0.08$ with slight smoothing yields good object sizes and avoids adding, due to imperfect predictions, segmented background noise to the objects. For seed extraction, the neighbor distance map prediction $\hat{y}_2$ is rescaled to damp low values and subtracted from the smoothed object distance map prediction $\hat{y}_1$. Thresholding with a threshold t_2 and connected-component labeling yields the seeds

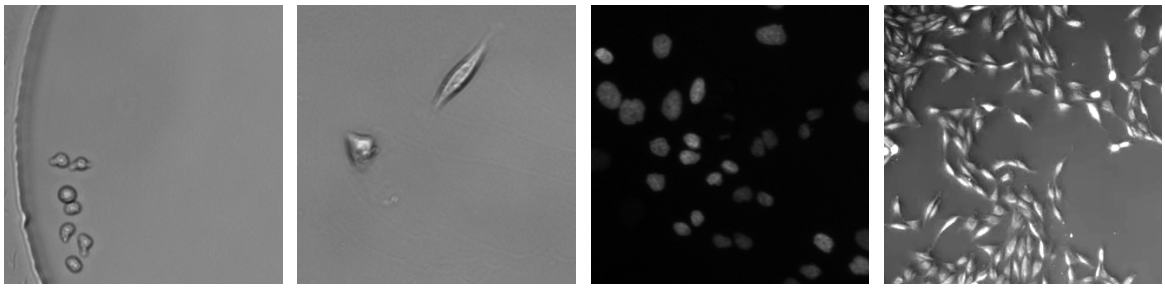
$$\hat{y}_{\text{seed}} = \text{CCL}\left(\left(g * \hat{y}_1 - \tan \hat{y}_2^2\right) > t_2\right) . \tag{3.11}$$

The threshold t_2 is a hyperparameter whose tuning can affect (i) the splitting and merging of objects and (ii) the detection of objects not clearly recognized by the CNN, often resulting in lower object distance map values. However, a threshold $t_2 = 0.5$ is sufficient in many applications and used in the segmentation method validation in this thesis if not stated otherwise. Finally, seeds smaller than 5 px are filtered, and a seed-based watershed is applied to fill the not yet labeled pixels in $\hat{y}_{\text{mask}}$.

Merging-Post-Processing (Distance Method)

Subtracting the neighbor distance map predictions from the object distance map predictions in Eq. 3.11 aims to improve the separation of adjacent objects and, therefore, to reduce merging adjacent objects. However, combining object and neighbor distance map predictions can also reduce over-segmentation, i.e., the splitting of objects. For instance, dumbbell-shaped objects can be over-segmented in a threshold- or local-maxima-based post-processing when using object distance maps alone. This over-segmentation results from extracting multiple, unconnected seeds as the object shape forms multiple basins, with their connection below the threshold t_2.

However, the neighbor distance prediction $\hat{y}_2$ enables detecting such splits by checking the sum of predicted neighbor distances at the interface between segmented adjacent objects. If this sum



a) BF-C2DL-HSC b) BF-C2DL-MuSC c) Fluo-N2DL-HeLa d) PhC-C2DL-PSC

Figure 3.5: Exemplary Training Data Crops. The mouse hematopoietic stem cells (a) and the mouse muscle stem cells (b) grow in a hydrogel microwell, which can be partially seen on the left side in (a). A particular feature of the muscle stem cells is that they can elongate. The HeLa cells show different intensities (c), and the pancreatic stem cells have partially unclear object boundaries (d). The shown data are crops from the corresponding Cell Tracking Challenge data sets [27], [28].

is smaller than a threshold t_3, the objects are merged in the optional merging post-processing. In this thesis, objects are detected as wrongly split due to shape if the sum divided by the number of border pixels is smaller than the empiric threshold $t_3 = 0.075$. The merging post-processing is only applied if stated explicitly.

3.1.5 Code Availability

Python implementations of the adapted border and the distance method are available at https://github.com/TimScherr/DL_based_instance_segmentation_for_microscopy_images. A user-friendly open-source instance segmentation tool is presented later in Section 3.3.

3.2 Validation

The adapted border method and the distance method are compared with state-of-the-art deep learning methods on two bright-field microscopy data sets, one fluorescence microscopy data set, and one phase contrast microscopy data set. All experiments have been performed on a system with Intel Core i9-9990K CPU, 64 GiB RAM, and two NVIDIA TITAN RTX GPUs with 24 GiB VRAM. The deep learning methods are implemented in Python with PyTorch. A field of interest correction is applied to all results. Therefore, segmented objects not reaching into the field of interest are filtered. The field of interest correction considers a similar procedure applied during the annotation process of Cell Tracking Challenge data sets.

3.2.1 Training Data Set and Test Data Sets

The compared methods are trained on a single training data set consisting of data from the Cell Tracking Challenge data sets BF-C2DL-HSC, BF-C2DL-MuSC, Fluo-N2DL-HeLa, and PhC-C2DL-PSC. The data sets BF-C2DL-HSC and BF-C2DL-MuSC were acquired with bright-field microscopy and show mouse hematopoietic stem cells and mouse muscle stem cells, respectively, that proliferate in hydrogel microwells. The fluorescence microscopy data set Fluo-N2DL-HeLa shows HeLa cells, and the phase contrast microscopy data set PhC-C2DL-PSC pancreatic stem cells on a polystyrene substrate. All data sets are 2D+t data sets, which means that the microscopy images were acquired

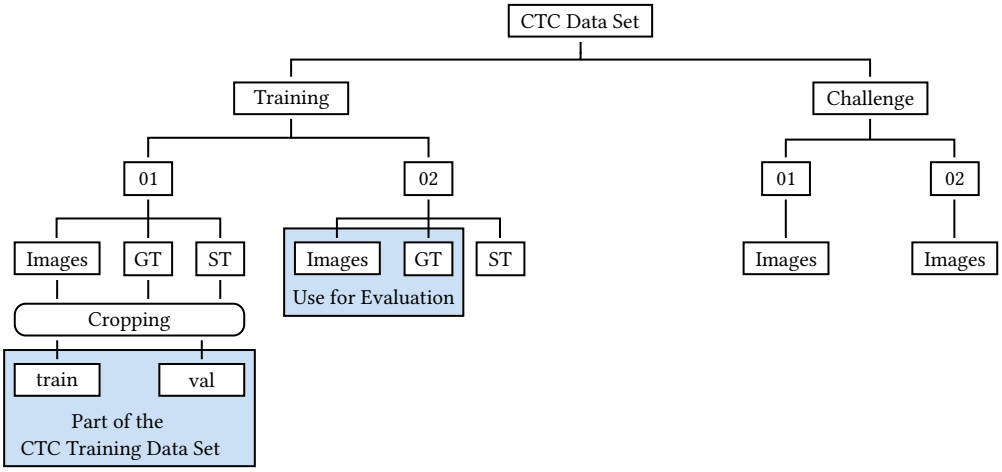


Figure 3.6: Cell Tracking Challenge Data Structure. The Cell Tracking Challenge provides two training and two challenge data sets for each cell type included in the challenge. The ground truths of the challenge data sets are kept secret and are used to evaluate challenge submissions. The two training data sets consist of images, human-made gold truth annotations (GT), and computer-generated silver truth annotations (ST). In this thesis, cropped images, GTs, and STs of the first training data set of four cell types are added to a single data set called the CTC training data set used for training. In contrast, images and GTs of the second set are used for evaluating a method's performance for each cell type separately.

over time. Thus, the object amount and density vary over time due to cell division events. Figure 3.5 shows exemplary 320×320 px crops.

This evaluation aims to find the best method without using models specifically trained for each object morphology and imaging technique. Therefore, a single training data set is extracted from the four mentioned Cell Tracking Challenge data sets. Each data set comes with two subsets with public annotations and two with non-public annotations. The provided reference annotations consist of human-made gold truths, in which not necessarily each cell is annotated, and possibly erroneous computer-generated silver truths. Figure 3.6 gives an overview of the data structure of a Cell Tracking Challenge data set. To evaluate the generalization ability of a trained model, the single training data set, which will be referred to as the CTC training data set, consists of crops taken from the first subsets with public annotations of the four data sets BF-C2DL-HSC, BF-C2DL-MuSC, Fluo-N2DL-HeLa, and PhC-C2DL-PSC. The second subsets with public annotations are used as test data sets, as highlighted in Figure 3.6. Since the STs are potentially erroneous, only GTs are used for the evaluation. In contrast to the training, where a single training data set is used, the segmentation quality is evaluated separately for each of the four cell types.

Table 3.2: CTC Training Data Set Composition. The CTC training data set consists of four cell types acquired with three different microscopy imaging techniques. Thus, methods need to deal with different object shapes, textures, boundaries, intensities, and densities. In addition, the number of objects per cell type in the CTC training data set is not balanced. All crops have a size of 320×320 px.

Cell Type	Crops	Objects	Objects Per Crop	Object-Image-Area-Ratio
BF-C2DL-HSC	90	423	1 to 11	1.45 %
BF-C2DL-MuSC	160	335	1 to 5	1.65 %
Fluo-N2DL-HeLa	90	1594	7 to 39	7.49 %
PhC-C2DL-PSC	90	4619	13 to 191	6.17 %
Total	430	6971	1 to 191	3.78 %

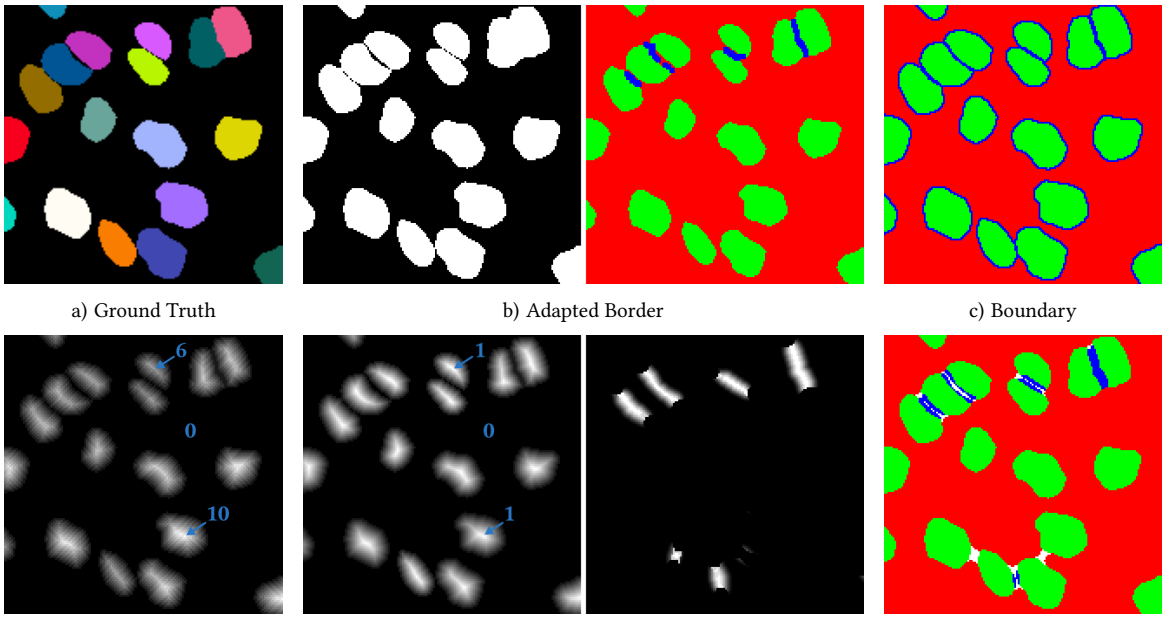


Figure 3.7: Data Representations of the Compared Instance Segmentation Methods. Multi-channel outputs are color-coded with the border/boundary/touching class channel in blue, the object channel in green, the background channel in red, and the gap class channel in white. For the object distance transforms, the values for two object maxima and the background are stated in blue to emphasize the normalization differences. The adapted border method and the distance method predict two outputs y_1 and y_2.

Table 3.2 shows the number of crops and objects per cell type in the CTC training data set and Table 3.3 provides information about the four test data sets. A single, trained model must cope with different microscopy techniques, object densities, and cell morphologies, i.e., shape, size, color, and texture. In addition, the total number of objects per cell type ranges from 335 for BF-C2DL-MuSC to 4619 PhC-C2DL-PSC in the CTC training data set. This imbalance in cell amount is another difficulty in training a well-generalizing model.

3.2.2 Compared Methods

The adapted border method and the distance method are compared on the four test data sets with (i) a three-class semantic method predicting object boundaries referred to as the boundary method, e.g., [41], (ii) the distance map regression method DIST [139], and (iii) a 4-class semantic method referred to as J_4 method [131]. All methods are trained on the CTC training data set introduced above. In contrast to the adapted border method and the distance method, which both predict two

Table 3.3: Properties of the Single Test Data Sets. Each model trained on the CTC training data set is evaluated on each test data set separately. The object count includes multiple appearances of the same cell in subsequent frames.

Test Data Set	Frames	Frame Size	Objects	Objects Per Frame
BF-C2DL-HSC	1764	1010 × 1010 px	64 503	2 to 159
BF-C2DL-MuSC	1376	1036 × 1070 px	7295	1 to 22
Fluo-N2DL-HeLa	92	700 × 1100 px	25 425	125 to 364
PhC-C2DL-PSC	300	576 × 720 px	56 804	65 to 498

data representations, those three methods predict a single data representation y_1. Therefore, a single-decoder version of the U-Net shown in Figure 3.4 is used.

Using the same training process and similar CNN architectures enables comparing the different data representations without possible influences of other architectural design choices. This procedure is appropriate since this thesis focuses on the data representation itself. Furthermore, the compared methods did not propose any architectural improvements and also used U-Net architectures in their original publication.

The post-processing of all compared methods is based on a seed extraction step, which differs mainly in the creation of y_{mask} and y_{seed}, and a watershed transformation. Seeds with an area smaller than 5 px are filtered for all methods but the DIST method, which relies on a local-maximum seed detection step. The methods with their post-processing and the used loss functions are described in the following.

Boundary Method

The prediction of object boundaries to split adjacent instances is a common and simple method, e.g., [141], [144], [146], and [41] utilize boundary predictions. In this thesis, a single-decoder U-Net is trained to predict the output y_1 with the three channels background, object, and object boundary (see Figure 3.7c). Therefore, object boundaries are extracted for each ground truth instance separately with morphological dilation operations.

The softmax activation function is applied in the output layer, and the loss function is the weighted sum of the cross-entropy loss and the Dice losses for the object interior channel $y_{1,2}$ and the object boundary channel $y_{1,3}$:

$$L_{\text{boundary}}\left(y_1, \hat{y}_1\right) = L_{\text{CE}}\left(y_1, \hat{y}_1\right) + \sum_{c=2}^{3} \frac{c-1}{1} L_{\text{Dice}}\left(y_{1,c}, \hat{y}_{1,c}\right) . \tag{3.12}$$

For the watershed-based post-processing, the mask y_{mask} is created by finding the pixels with the object class as the most likely prediction:

$$y_{\text{mask},i} = \begin{cases} 1 & \text{if } \underset{c\in\{1,2,3\}}{\arg\max}\, y_{1,c,i} = 2, \quad (\textit{object class}) \\ 0 & \text{else.} \end{cases} \tag{3.13}$$

In this equation, $y_{\text{mask},i}$ represents the value of y_{mask} at pixel i. The resulting mask is one at pixels where the object channel has the highest probability and zero elsewhere. The boundary class is not considered in Eq. 3.13 since the boundaries were extracted with dilation operations and, therefore, do not belong to an object. The seeds required for the watershed result from connected-component labeling:

$$y_{\text{seed}} = \text{CCL}\left(\left(y_{1,2} \cdot \left(1 - y_{1,3}\right)\right) > 0.5\right) . \tag{3.14}$$

DIST Method

The DIST method proposed in [139] predicts a single-channel object distance map data representation y_1. In contrast to the distance method, which uses normalized Euclidean distances, non-normalized Chebyshev distances are used (see Figure 3.7d). This missing normalization requires a local-maxima-based seed extraction. However, similar to the distance method, the output

activation function of the single-decoder U-Net is the linear activation and the smooth L1 loss is used for training:

$$L_{\text{DIST}}\left(y_1, \hat{y}_1\right) = L_{\text{SL1}}\left(y_1, \hat{y}_1\right) . \tag{3.15}$$

The seed extraction in the post-processing is based on local maxima detection. Let $\mathscr{M}$ be the set of all detected local maxima in y_1 and let $\mathscr{P}_{m,m'}$ be the set of all paths γ from a local maximum $m \in \mathscr{M}$ to a higher local maximum $m' \in \mathscr{M}$ with $y_{1,m'} > y_{1,m}$. Then, m represents an object seed if along all paths γ the decrease in y_1 is at least p_1, which means that two local maxima m and m' only define two seeds if the minimum of the valley between them is at least p_1 smaller than m. Defining the subset of seeds

$$\mathscr{M}_{\text{seed}} = \left\{ m \in \mathscr{M} \;\middle|\; \left(\min_{\gamma \in \mathscr{P}_{m,m'}} \max_i \left(y_{1,m} - y_{1,i}\right) \right) > p_1 \;\forall\, m' \in \mathscr{M},\; y_{1,m'} > y_{1,m} \right\}, \tag{3.16}$$

with i being a pixel of a discrete path γ, the mask and seed creation step can be formulated as

$$y_{\text{mask}} = y_1 > 0.5\,, \tag{3.17}$$

$$y_{\text{seed},i} = \begin{cases} i & \text{if } i \in \mathscr{M}_{\text{seed}}, \\ 0 & \text{else.} \end{cases} \tag{3.18}$$

The seed values can also be numbered in ascending order starting from one, but a unique value for a seed is sufficient.

The highest AJI scores are obtained in [139] using $p_1 = 0$. However, the authors suggest higher values of p_1 to aggregate local maxima to form a single object on noisy images and when making predictions on a different domain than trained on. In this thesis, $p_1 = 1$ is used since this was the best value on the data sets Fluo-N2DL-HeLa and BF-C2DL-HSC. In addition, a value of 2 led to a drastic performance drop on the test data set of PhC-C2DL-PSC (under-segmentation), where $p_1 = 0$ yielded the best results, while $p_1 = 0$ led to a drastic performance drop on the test data set BF-C2DL-MuSC (over-segmentation).

J_4 Method

The last method in the comparison is the J_4 method that applies a J regularization loss to tackle the class imbalance problem when predicting the single output y_1 with the four channels background, object, touching, and gap, as shown in Figure 3.7f [131]. In contrast to the touching class that belongs to an object, the gap class is located in the background of the ground truth. As for the other multi-class predictions, the softmax activation function is applied in the last layer. The loss function is the weighted sum of the cross-entropy loss and the J-regularization loss:

$$L_{J_4}\left(y_1, \hat{y}_1\right) = L_{\text{CE}}\left(y_1, \hat{y}_1\right) + \omega\, L_J\left(y_1, \hat{y}_1\right) . \tag{3.19}$$

For the derivation and definition of the J regularization loss L_J, please refer to [131]. The authors state that the cross-entropy loss should drive the initial optimization process. Thus, this thesis uses the weight $\omega = 0.006$, resulting in a comparable size of both loss terms after roughly a third of the training process. The different weighting to [131] may be due to different training data sets and implementations, like using PyTorch's cross-entropy loss that is numerically more stable than the in [131] stated cross-entropy formula.

Table 3.4: Encoded Neighbor Information of the Compared Methods for the CTC Training Data Set. Stated is the relative amount of pixels with information about neighboring objects, i.e., object border pixels or pixels with a neighbor distance greater than zero. For the J_4 method, ratios of the touching and the gap class are provided separately. Using simple, not dilated, and not robust borders would result in a total relative neighbor information ratio of 0.32 ‰. Note: adjacent objects can also be distinguished without neighbor information; therefore, the amount of neighbor information may not be related to the instance segmentation performance of a method. However, more information should be beneficial in the training process and in the post-processing.

Cell Type	Adapted Border	Boundary	DIST	Distance	J_4
BF-C2DL-HSC	0.47 ‰	0	0	2.98 ‰	0.55 ‰ / 0.72 ‰
BF-C2DL-MuSC	0.11 ‰	0	0	0.53 ‰	0.15 ‰ / 0.11 ‰
Fluo-N2DL-HeLa	0.55 ‰	0	0	5.14 ‰	0.74 ‰ / 1.10 ‰
PhC-C2DL-PSC	2.48 ‰	0	0	11.01 ‰	2.81 ‰ / 8.30 ‰
Total	0.77 ‰	0	0	4.20 ‰	0.92 ‰ / 2.16 ‰

In the post-processing, it needs to be considered that the gap class (channel 4) is, per definition, in the background and does not belong to an object, while the touching class (channel 3) belongs to objects:

$$y_{\text{mask},i} = \begin{cases} 1 & \text{if } \underset{c\in\{1,2,3,4\}}{\arg\max}\ y_{1,c,i} = 2, \qquad (\textit{object class}) \\ 1 & \text{if } \underset{c\in\{1,2,3,4\}}{\arg\max}\ y_{1,c,i} = 3, \qquad (\textit{touching class}) \\ 0 & \text{else,} \end{cases} \tag{3.20}$$

$$y_{\text{seed}} = \text{CCL}\left(\left(y_{1,2} \cdot \left(1 - y_{1,3}\right) \cdot \left(1 - y_{1,4}\right)\right) > 0.5\right) . \tag{3.21}$$

The resulting mask y_{mask} is one at pixels where the object channel or the touching channel has the highest probability and zero elsewhere.

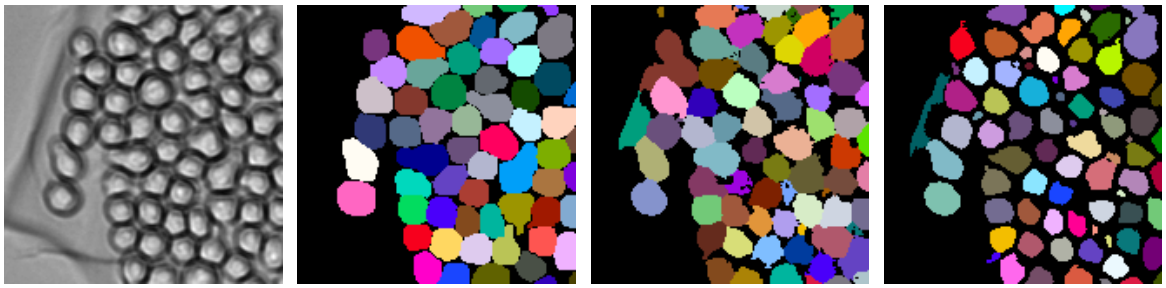
Encoded Neighbor Information of the Compared Methods

Table 3.4 shows the encoded neighbor information of the compared methods for the CTC training data set. In total, 0.77 ‰ of the pixels belong to the border class of the adapted border method. Thus, using of adapted borders reduced the class imbalance compared to single-pixel thick borders, for which only 0.32 ‰ of the pixels belong to the border class. The boundary method and the DIST method do not use explicit information about neighboring objects. The distance method has the highest amount of pixels contributing to the learning of neighboring objects (4.20 ‰). Furthermore, the J_4 method utilizes about 3.08 ‰ of the total pixels, but the information is split into the two classes touching and gap.

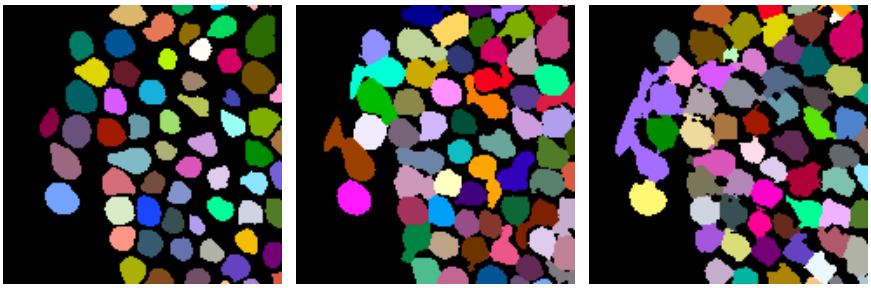
Looking at the single cell types in the CTC training data set reveals differences due to different object amounts and densities for the single cell types. For instance, the distance method has by far the largest amount of neighbor information for the HeLa cells, which are often close to each other but still separated by a few pixels, while it has a similar amount of information as the J_4 method for the PSC cells, which grow very dense.

3.2.3 Results

Eleven models per method were trained on the CTC training data set with the Adam optimizer and eleven with the Ranger optimizer. The DET measure, the SEG measure, and the mean OP_{CSB} of



a) Image b) Ground Truth c) Adapted Border d) Boundary



e) DIST f) Distance g) J_4

Figure 3.8: Qualitative Segmentation Results for BF-C2DL-HSC (Frame 1748). The results of a method's best model trained on the CTC training data set are shown (selected from the 11 Adam and the 11 Ranger models of a method). An issue for this data set is the segmentation of the hydrogel well, in which the cells are growing, visible on the left in the crop. Furthermore, the HSC cells in the training data set are less densely packed than in this late frame from the test data set. A 160×160 px crop is shown for visualization reasons.

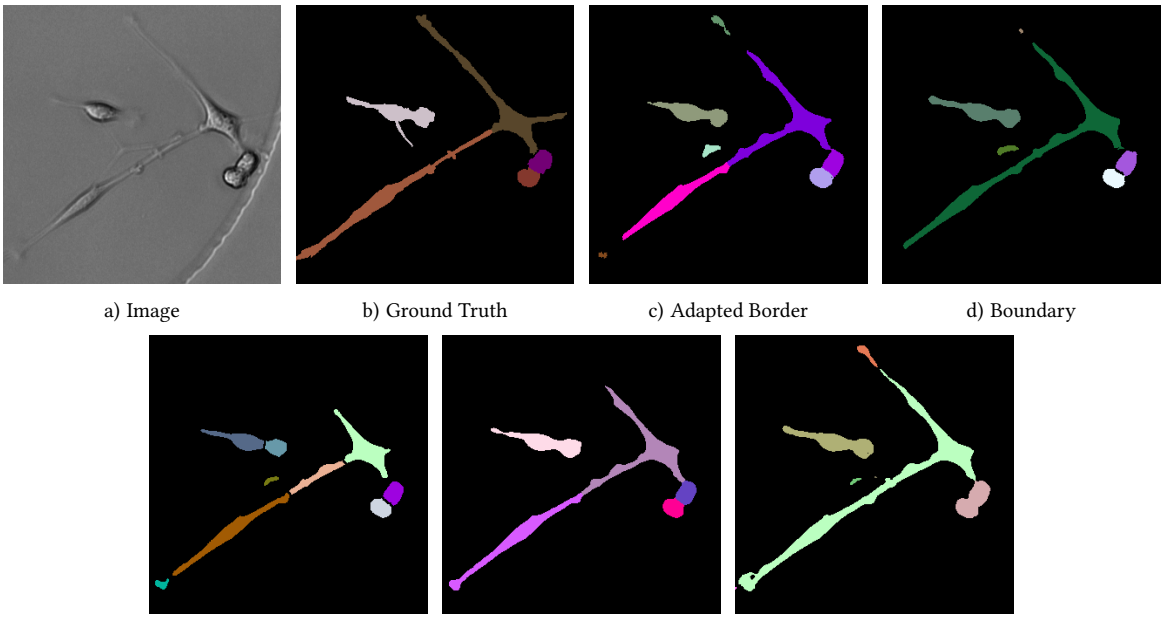
those two measures are used as evaluation metrics. The latter measure is the overall performance measure in the Cell Segmentation Benchmark of the Cell Tracking Challenge.

Qualitative Analysis of the Segmentation Masks

Before analyzing the evaluation metrics, the qualitative analysis of some segmentation results of the best models for each of the four evaluated cell types offers some impressions about the strengths and weaknesses of the compared methods.

Figure 3.8 reveals that all compared semantic methods suffer from tiny holes in the segmentation masks and small split object parts for late frames of the BF-C2DL-HSC data set. However, this behavior may be due to the missing of such densely packed hematopoietic cells in the training data set. The holes could be closed in an additional post-processing step. In addition, the boundary method shows over-segmentation. Segmenting parts of the hydrogel well is an issue for all methods except the DIST method. Although the threshold for the DIST method of 0.5 should theoretically provide the perfect cell size, the cells are segmented too small, indicating that the model has not learned a slope of 1 at the transition from background to cell. The boundary method also suffers from too small segmented cells since the boundary class does, per definition, not belong to the cell area, a drawback for such densely packed cells.

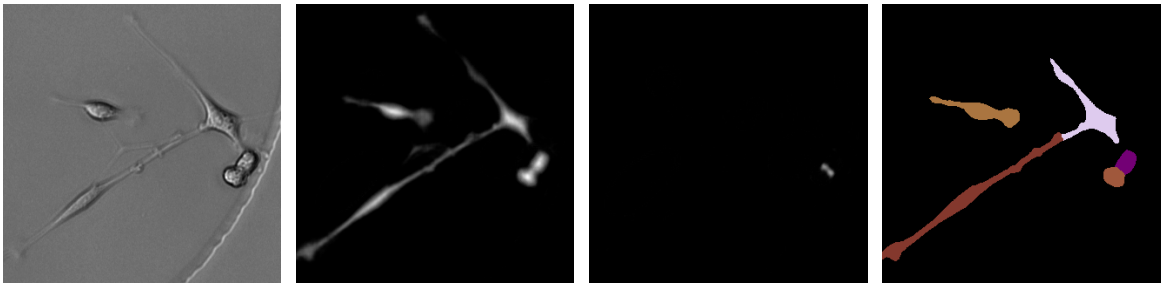
Figure 3.9 shows the challenge of separating adjacent cells in the BF-C2DL-MuSC data set with its muscle stem cells that can elongate. The boundaries between objects are unclear, and without the use of time information, the single cells are difficult to annotate. However, thin branches of the cells are not consistently annotated. Due to the difficult shape and the unclear object boundaries, over-



a) Image b) Ground Truth c) Adapted Border d) Boundary

e) DIST f) Distance g) J_4

Figure 3.9: Qualitative Segmentation Results for BF-C2DL-MuSC (Frame 1106). The results of a method's best model trained on the CTC training data set are shown (selected from the 11 Adam and the 11 Ranger models of a method). Especially the segmentation of the elongated mouse muscle stem cell state and unclear boundaries between adjacent cells are challenges for this data set. A 320×320 px crop is shown for visualization reasons.



a) Image b) Object Distance Prediction c) Neighbor Distance Prediction d) Prediction

Figure 3.10: Improved Object Separation using Neighbor Distance Predictions. Some distance method models fail to learn to separate objects with unclear boundaries in the object distance map. The two cells on the right merge in the post-processing with a threshold t_2 of 0.5 without the neighbor distance prediction. However, using the neighbor distance prediction prevents this erroneous merging, although it has also not been learned perfectly for this difficult case. The ground truth is shown in Figure 3.9.

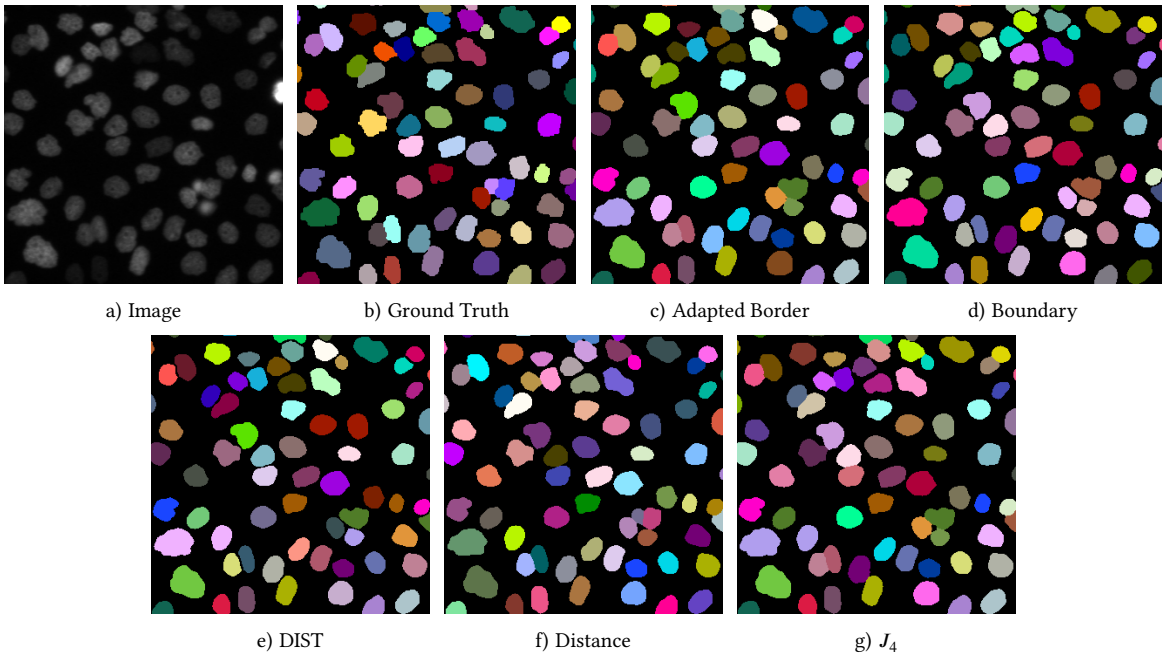


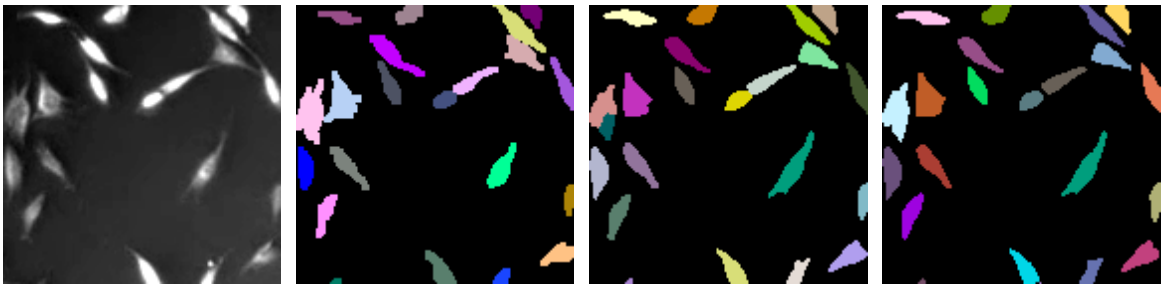
Figure 3.11: Qualitative Segmentation Results for Fluo-N2DL-HeLa (Frame 79). The results of a method's best model trained on the CTC training data set are shown (selected from the 11 Adam and the 11 Ranger models of a method). All compared methods deliver a good segmentation and differ only in the segmentation of a single cell. The brightness differences of the single cells seem not to be an issue and require no further pre-processing step. A 320×320 px crop is shown for visualization reasons.

and under-segmentation occurs. However, the distance method provides a reasonable segmentation quality for the shown crop of a test data set image. Furthermore, Figure 3.10 shows the benefits of the neighbor distance information for separating objects with unclear object boundaries. Some distance method models did not learn to separate the two cells on the right in the object distance prediction. The neighbor distance prediction prevented the merging of these two cells.

All methods segment the HeLa cells in Figure 3.11 well. A reason may be that the number of HeLa cells in the CTC training data set is larger than that of the mouse hematopoietic and mouse muscle stem cells. In addition, the HeLa cells' texture, size, and shape may be easier to learn for the compared methods. The large brightness differences of the single cells are no issue in the shown image crop, and, therefore, no specific pre-processing is required for the Fluo-N2DL-HeLa data set.

The pancreatic stem cells acquired with phase contrast microscopy provide the largest cell number in the CTC training data set. However, low contrast, a small object width, and high object density in late frames are challenging in the PhC-C2DL-PSC data set (see also Figure 3.5). The detection scores should be mainly limited by the segmentation errors in late frames with a high object density, as Figure 3.12 shows a reasonable mid-frame segmentation for all methods. Predicting the exact cell size is difficult because of the low contrast, which is probably why the SEG score and the SEG score inter-annotator agreement are the lowest of the four used Cell Tracking Challenge data sets for PhC-C2DL-PSC as Table 3.5 shows.

Summarized, the best model of the distance method provides a good segmentation for all shown image crops. The best models of the semantic methods have some problems segmenting the late BF-C2DL-HSC frames, and the best DIST method model needs to be more accurate for the challenging elongated cells of the BF-C2DL-MuSC data set. The SEG score inter-annotator agreement for those



a) Image b) Ground Truth c) Adapted Border d) Boundary

e) DIST f) Distance g) J_4

Figure 3.12: Qualitative Segmentation Results for PhC-C2DL-PSC (Frame 182). The results of a method's best model trained on the CTC training data set are shown (selected from the 11 Adam and the 11 Ranger models of a method). All compared methods yield a good segmentation. However, the low contrast makes estimating the exact cell shape challenging, especially for late frames with densely packed cells. A 140×140 px crop is shown for visualization reasons.

elongated cells are probably lower than the stated agreement in Table 3.5 since the agreement for the roundish cells should be similar to the BF-C2DL-HSC data set.

Detection Quality

The detection quality measure DET evaluates object-level errors such as added or split objects, not segmented missing objects, and erroneously merged objects. These error sources are weighted and combined into a single score, as described in Chapter 1.

Figure 3.13 shows DET measure boxplots for the single data sets, methods, and optimizers. The results for training with the Adam optimizer and training with the Ranger optimizer are stated separately. First of all, it is noticeable that the results for the individual data sets are very different.

Table 3.5: Inter-Annotator Agreement for the Four Used Cell Tracking Challenge Data Sets. The Cell Tracking Challenge gold truth masks are annotated independently by three experts. The inter-annotator agreement is stated as the mean and standard deviation of the DET and SEG scores of the three manual annotations and, therefore, provides an upper limit for the DET and SEG scores. The inter-annotator agreement for all not-simulated Cell Tracking Challenge data sets can be found below the Cell Segmentation Benchmark leaderboard on the Cell Tracking Challenge website: http://celltrackingchallenge.net/latest-csb-results/.

Data Set	DET	SEG
BF-C2DL-HSC	0.996 ± 0.005	0.892 ± 0.036
BF-C2DL-MuSC	0.994 ± 0.003	0.843 ± 0.026
Fluo-N2DL-HeLa	0.987 ± 0.002	0.904 ± 0.035
PhC-C2DL-PSC	0.983 ± 0.010	0.788 ± 0.044

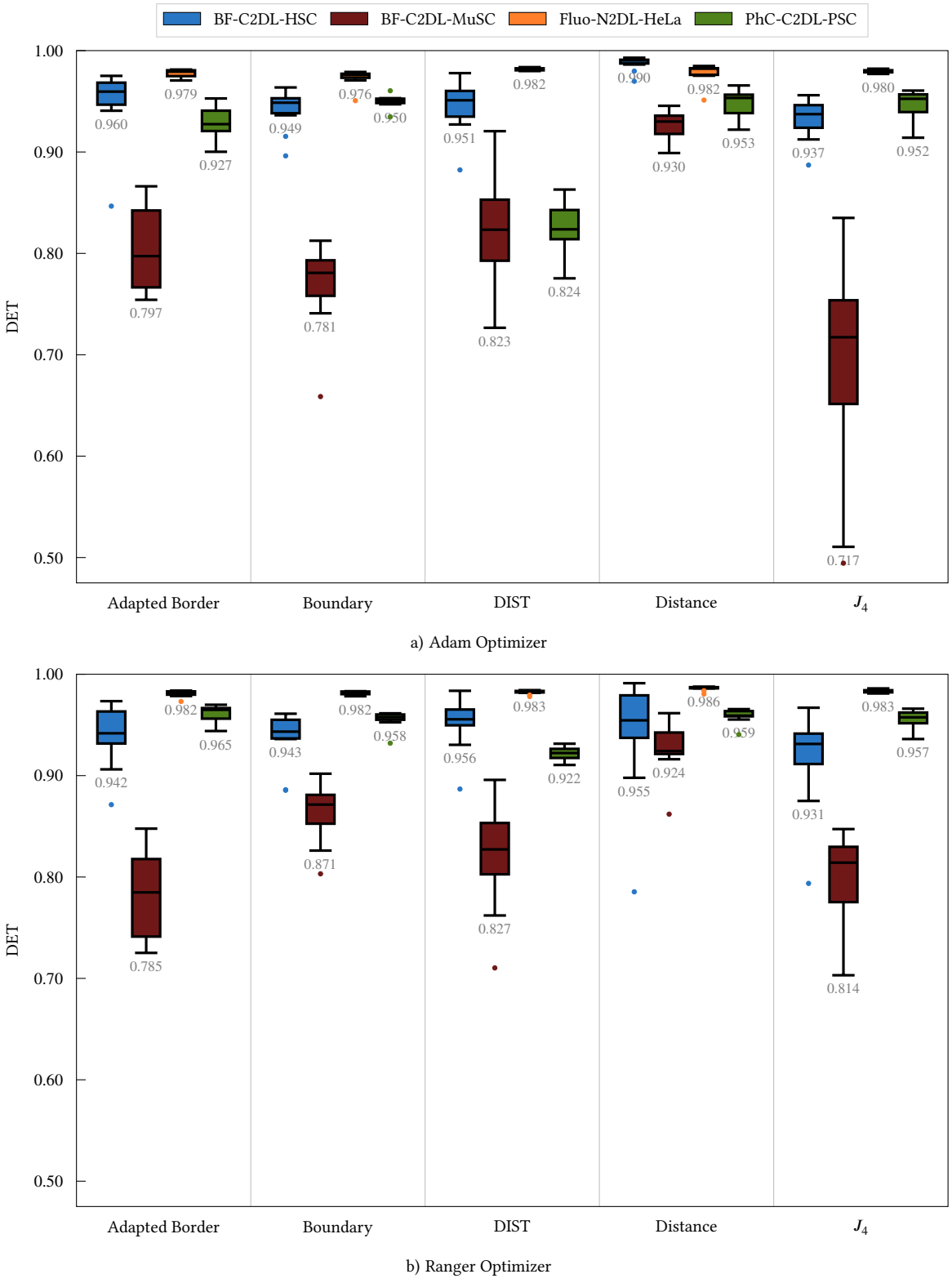


Figure 3.13: DET Measures for the Four Test Data Sets. 11 models have been trained with the Adam optimizer and 11 with the Ranger optimizer for each method. All models are evaluated on the four test data sets separately. The distance method provides overall the highest DET scores. The median values are shown below the lower whiskers.

The Fluo-N2DL-HeLa data set is overall the data set with the highest scores, while the BF-C2DL-MuSC data set has the lowest scores due to its challenging cell morphology. Furthermore, the Ranger optimizer provides an improved median DET score in fourteen cases but a decreased median score in six cases compared to the Adam optimizer results. Especially, the detection quality of the distance method drops for the BF-C2DL-HSC data set when using the Ranger optimizer. However, besides this interesting drop, the optimizer choice only slightly affects the method rankings for the four cell types.

For the BF-C2DL-HSC data set, a distance method model trained with the Adam optimizer provides the highest score. In addition, the median score of the Adam distance models is the highest and the variance the smallest of all compared methods on this data set. The adapted border method is the second best method when trained with the Adam optimizer, but the boundary method, the DIST method, and the J_4 method are close behind. Using the Ranger optimizer results in higher variance and decreased scores for the distance, the adapted border, and the J_4 method, and all methods produce quite similar results in that case.

The mouse muscle stem cells of the BF-C2DL-MuSC data set can occur as small roundish objects and in an elongated state that is prone to over- and under-segmentation. Thus, the scores are quite low and the differences between the single trained models of a method are high. The distance method is the only method that can constantly produce high-quality results for this challenging cell type. The J_4 method can break down when trained with the Adam optimizer on the CTC training data set. The object-level error sources are studied later to gain insight into this behavior.

In contrast, the differences between the compared methods are very small for the Fluo-N2DL-HeLa data set. All methods reach a very good segmentation quality for this cell type. However, the Ranger optimizer can produce slightly better results for all compared methods than the Adam optimizer. The distance method provides the highest median DET score again.

The comparatively low values of the DIST method, especially when using the Adam optimizer, stand out in the PhC-C2DL-PSC results. This is partially due to the choice of $p_1 = 1$ in the post-processing resulting in under-segmentation for this particular data set. As mentioned in the description of the DIST method, $p_1 = 0$ (each local maximum is used as seed) yielded the best results for PhC-C2DL-PSC but resulted in over-segmentation for the other data sets. Thus, unlike the distance method, the DIST method cannot segment all four cell types well when using a single parameter set. However, switching from the Adam optimizer to the Ranger optimizer increases the median score for the DIST method, indicating that the fixed parameter set for all cell types is not the only issue of this method. The adapted border method trained with the Ranger optimizer provides the highest median DET score for the PhC-C2DL-PSC data set. However, the other methods reach similar scores except for the DIST method.

Table 3.6: Cases with Significant DET Score Improvements. The number of cases of a method being significantly better than the baseline method is stated. A method can reach eight significant improvements in each comparison (one for each data set and optimizer). None of the methods is significantly better than the proposed distance method. See Appendix A for the statistical significance tests.

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1	2	0	1
Boundary	2	-	3	0	2
DIST	1	2	-	0	3
Distance	5	4	6	-	4
J_4	2	2	2	0	-

Overall, using the novel distance method yields the most consistent detection quality and the best median DET score for three of the four data sets. The DIST method suffers from its local maximum post-processing due to the use of non-normalized Chebyshev distances. Furthermore, the adapted border method, the boundary method, and the J_4 method produce pretty similar results. Table 3.6 shows that no method is significantly better than the distance method. However, the distance method is at least in four cases better than the other method for each compared method. See Appendix A for more information about the statistical significance testing using almost stochastic order (ASO).

Object-Level Error Sources

The single error sources contributing to the DET measure are stated separately in Figure 3.14, Figure 3.15, and Figure 3.16 to get more insight into the strengths and weaknesses of the single method. The miss rate is the number of false negatives divided by the total cell number in a data set. The combined add and split rate is the number of false positives due to adding or splitting objects divided by the total cell number in a data set. Finally, the merge rate is the number of erroneously merged objects divided by the total cell number in a data set.

The miss rates in Figure 3.14 reveal that the DET performance drop of the distance method for BF-C2DL-HSC when using the Ranger optimizer is due to missing objects, i.e., the miss rate increases from virtually no missing objects to about 10 % in the worst case. Interestingly, the miss rate for BF-C2DL-MuSC does not increase, although the small roundish state of the mouse muscle stem cells looks quite similar to the mouse hematopoietic stem cells. This finding might indicate that the Ranger models do not generalize well to the late frames with high object density. Besides that, high miss rates are mainly an issue for the PhC-C2DL-PSC data set. Especially the DIST method does not generalize well to the late frames with high object density and misses many cells. Thus, the DET scores were not only limited by the for all data sets fixed post-processing parameter p_1. However, the small width of the cells in the late frames is an issue for all compared methods.

Figure 3.15 shows the due to technical metric calculation reasons combined split and add rates for the four test data sets and reveals that false positives are still a problem for all methods and data sets. The problem of many splits and added objects may be underestimated in the DET measure that weights erroneous merges five times more and false negatives ten times more. However, it needs to be considered that some splits may be counted as multiple false positives if no part fulfills the matching condition in Eq. 1.5. In addition, the high add and split rates for the data sets BF-C2DL-HSC and BF-C2DL-MuSC are due to the hydrogel well, in which the cells proliferate, as Figure 3.17 shows. It may be that parts of this hydrogel well are recognized as mouse muscle stem cells in their elongated state. Since the two test data sets consist of 1764 and 1376 frames, respectively, several thousand false positives can easily occur. These false positives can probably be filtered in a post-processing step. However, the generalization to unknown or in the training data under-represented structures without structure-specific false positive filtering is of interest as well. The distance method adds the least false positives but still a significant number.

The merge rates in Figure 3.16 show that the goal of the adapted border method and the distance method, to reduce the erroneous merging of objects, has partially been reached. The adapted order method is the semantic method with the lowest merge rates in this comparison. The distance method can compete with the DIST method without a local maximum post-processing, which is prone to over-segmentation and needs a for each cell type tuned post-processing. However, the low resolution and contrast and the high object density in the PhC-C2DL-PSC data set still result in merges that need to be resolved.

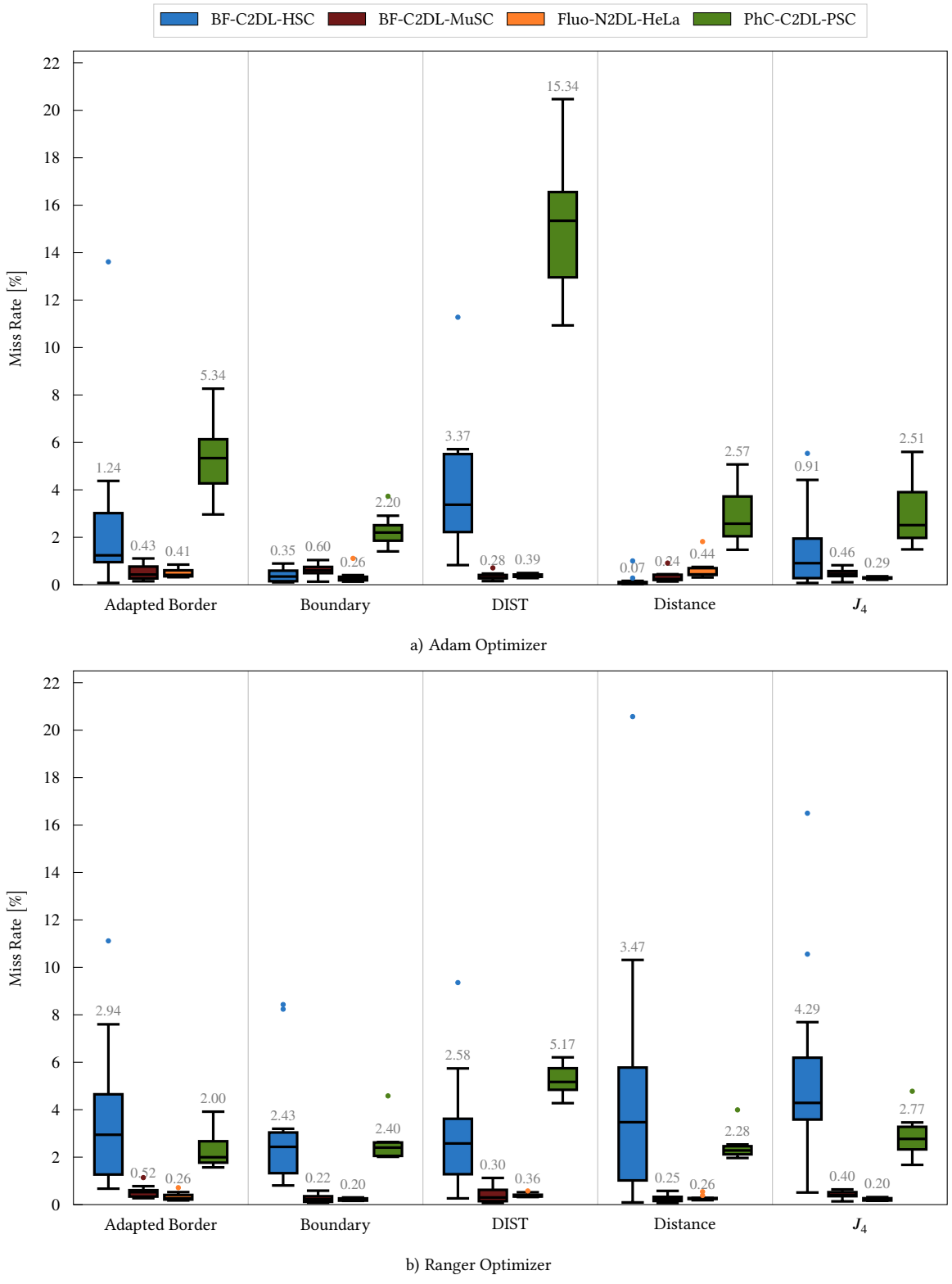


Figure 3.14: Miss Rates for the Four Test Data Sets. 11 models have been trained with the Adam optimizer and 11 with the Ranger optimizer for each method. All models are evaluated on the four test data sets separately. The median values are shown above the upper whiskers.

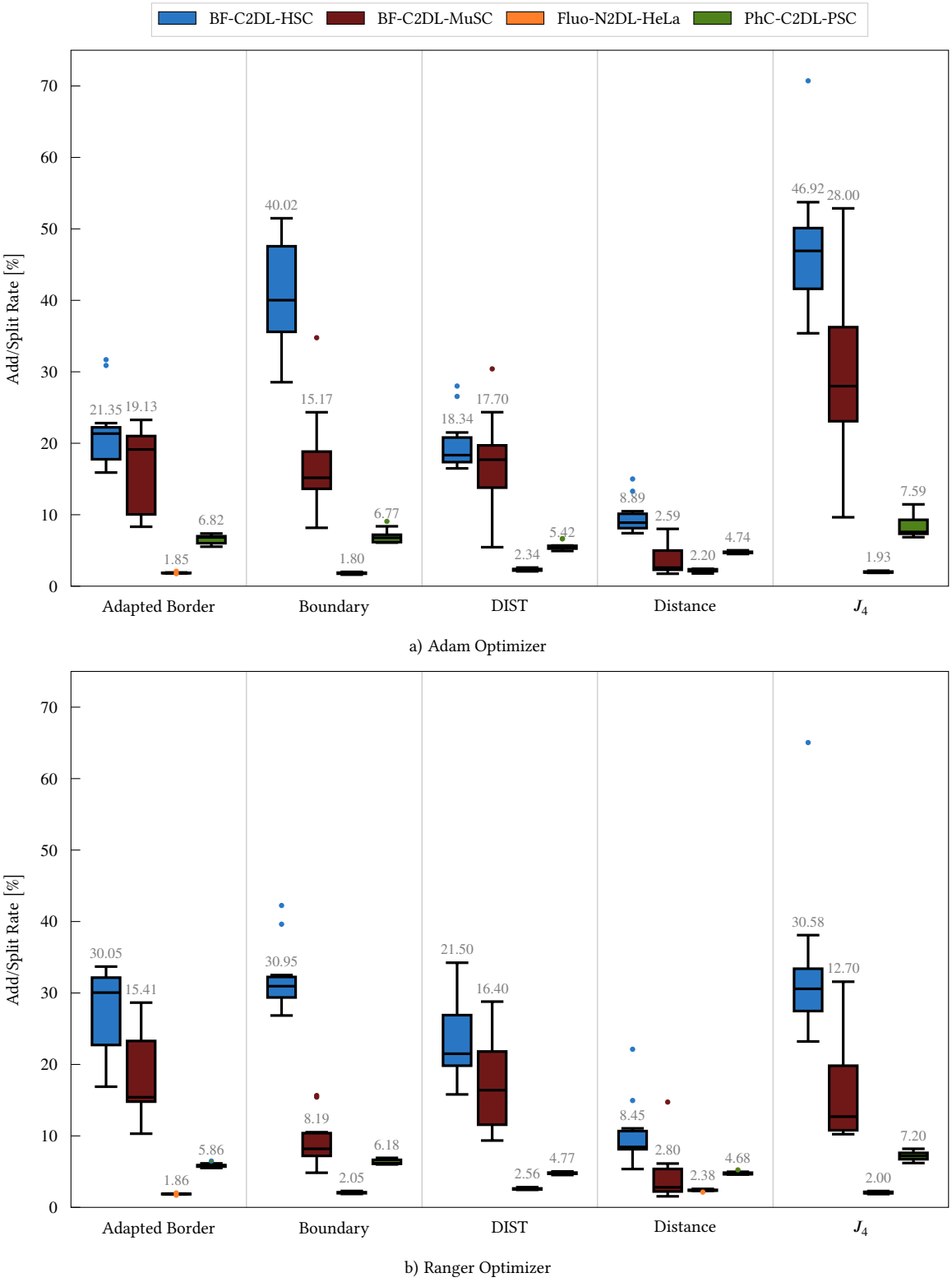


Figure 3.15: Combined Add and Split Rates for the Four Test Data Sets. 11 models have been trained with the Adam optimizer and 11 with the Ranger optimizer for each method. All models are evaluated on the four test data sets separately. The median values are shown above the upper whiskers.

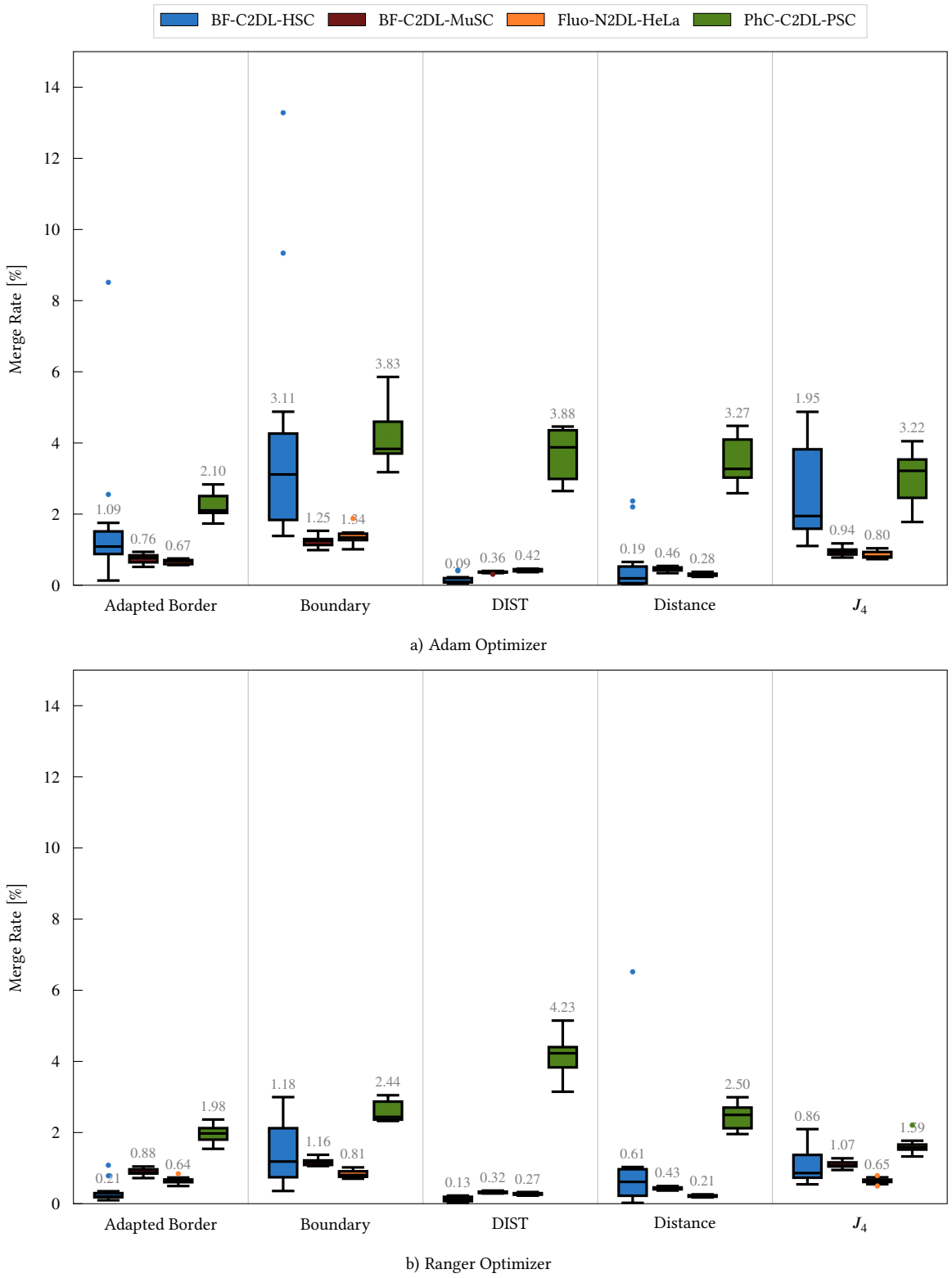


Figure 3.16: Merge Rates for the Four Test Data Sets. 11 models have been trained with the Adam optimizer and 11 with the Ranger optimizer for each method. All models are evaluated on the four test data sets separately. The median values are shown above the upper whiskers.

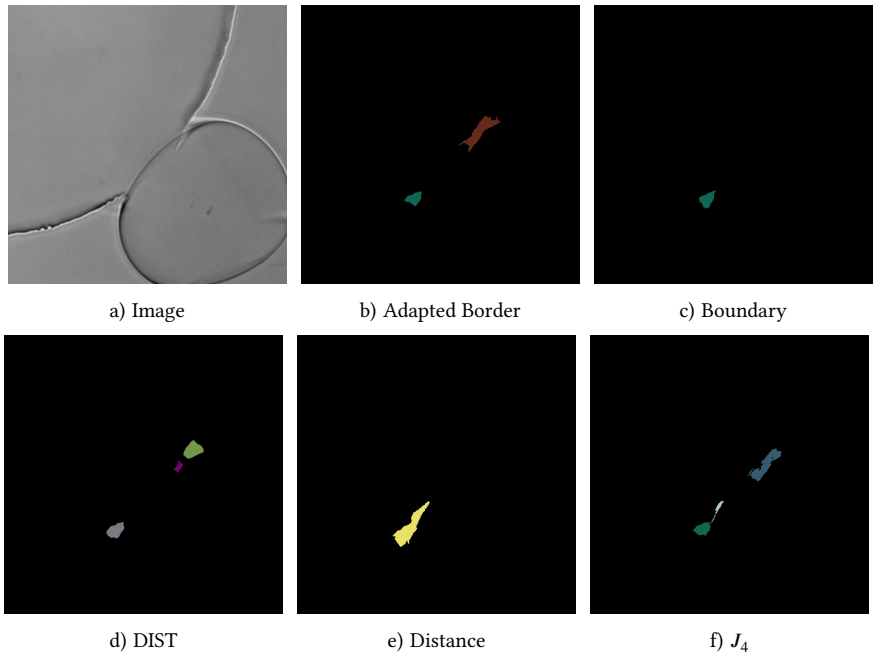


Figure 3.17: BF-C2DL-HSC and BF-C2DL-MuSC False Positive Error Source. The mouse hematopoietic stem cells and muscle stem cells proliferate inside hydrogel microwells. All methods produce false positives at the boundaries of those microwells that need to be filtered. This generalization errors could probably also be reduced when training on whole frames. However, the object-to-background-ratio compared to using crops is smaller as empty crops are filtered. Shown is a 460×460 px crop.

Overall, the object-level error sources are data set specific, but at least for three of the four data sets, the miss rate and the merge rate are low, especially when using the distance method. False positives are the largest error source but are weighted the least in the DET metric since they can easier be corrected as false negatives or merges. Thus, a larger training data set or a filtering post-processing is needed.

Merging Post-Processing

The merging post-processing of the distance method aims to reduce the erroneous splitting of objects into multiple parts. Figure 3.18 shows an exemplary case where the merging post-processing reduces this splitting of objects. Multiple seeds are extracted in the threshold-based post-processing for two of the cells due to the cell shape forming two basins in the predicted object distance map. Since there is no neighbor distance prediction, the split objects are identified as wrongly split and merged in the post-processing.

Table 3.7: Changes of the Median DET Score, Miss Rate, Add/Split Rate, and Merge Rate when Applying the Merging Post-processing on the BF-C2DL-MuSC Data Set. The changes of the median scores from Figure 3.19 are reported. In contrast to the DET measure, a negative change means an improvement for the miss, add/split, and merge rates. The DET score decreases due to different weightings of the error sources, although the merging post-processing corrects more merges than it introduces splits.

Optimizer	Δ DET	Δ Miss Rate [%]	Δ Add/Split Rate [%]	Δ Merge Rate [%]
Adam	−0.015	0	−1.298	0.736
Ranger	−0.002	0	−1.734	0.521

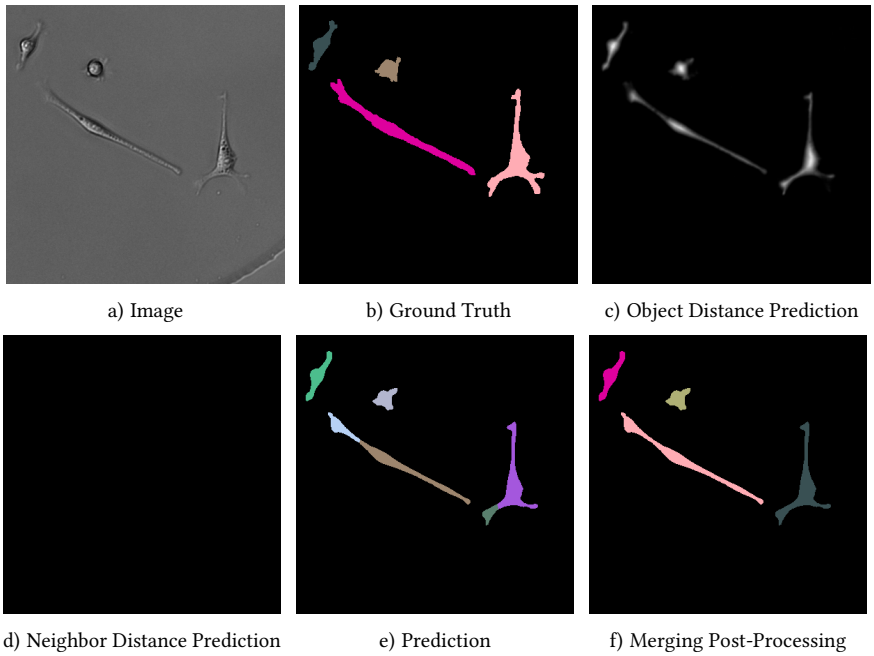


Figure 3.18: Reducing Splits with the Merging Post-Processing. Some object shapes in the BF-C2DL-MuSC data set result in multiple seeds per object when applying the threshold-based seed extraction step in the post-processing. An additional merging post-processing step can fix this over-segmentation by using the information provided by the neighbor distance prediction. In the shown case, no neighbor distance has been predicted, implying that the cells were erroneously split due to their shapes. Thus the single cell parts are merged. Shown is a 400×400 px crop from the BF-C2DL-MuSC data set.

Figure 3.19 shows that the merging post-processing effectively reduces the add and split rate for the mouse muscle stem cells. However, a drawback is that the merge rate increases. The miss rate remains constant, as expected. Thus, a split decrease of about 1.3 % is offset by a merge increase of about 0.7 % for the Adam optimizer (see Table 3.7). Since merges are weighted stronger than splits, the DET metric decreases slightly. For the Ranger optimizer, a 1.7 % decrease in splits faces an 0.5 % increase in merges resulting in a slightly lower DET score as well.

Looking back at Figure 3.9 reveals that adjacent elongated mouse muscle stem cells are difficult to detect and sometimes just correctly recognized as two cells due to luck. The model never really learned to distinguish the two elongated cells; therefore, there is also no neighbor distance prediction. An additional upper cell size constraint can help in these cases to prevent the merging in the merging post-processing. Finally, adjustments such as size constraints, more validation studies, and appropriate data sets are needed to improve and validate the merging post-processing.

So, the influence of the merging post-processing on the DET score is small. However, the total number of errors decreases since more splits are resolved than wrong merges are induced. The evaluation of the merging post-processing is, therefore, ambiguous. Appropriate test data sets and an improved query not only based on the neighbor distance prediction are required. Furthermore, hyperparameter tuning of the threshold t_3 may also improve the merging post-processing, but significant improvements are unlikely.

SEG Measure

The Jaccard similarity index measure SEG evaluates pixel-level errors, i.e., how well the sizes and shapes of a ground truth object and a predicted object match. Therefore, it is an excellent

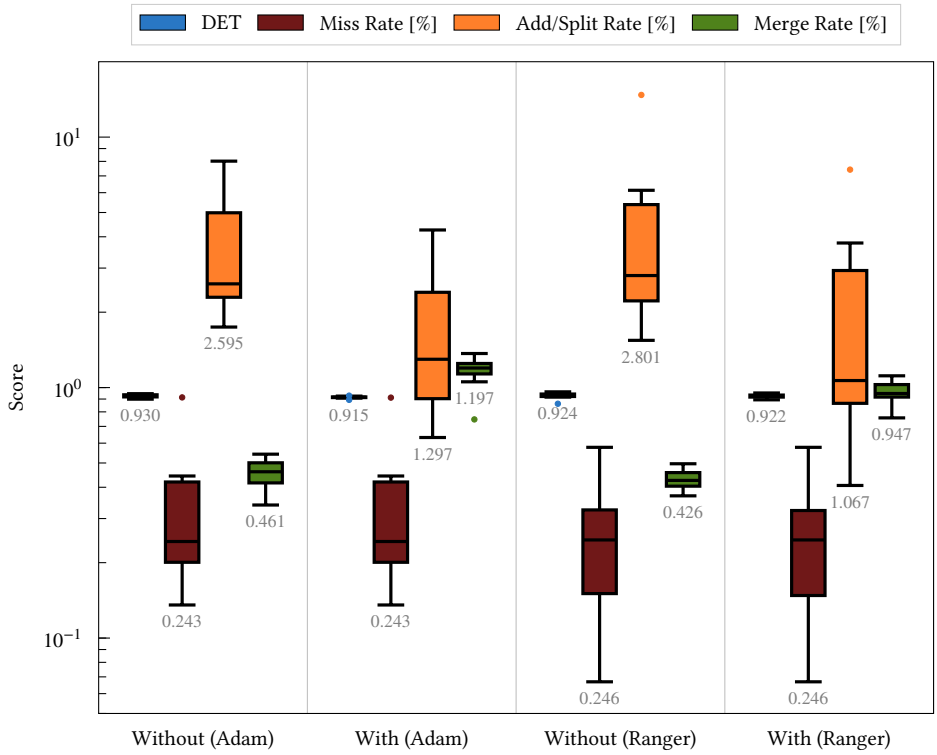
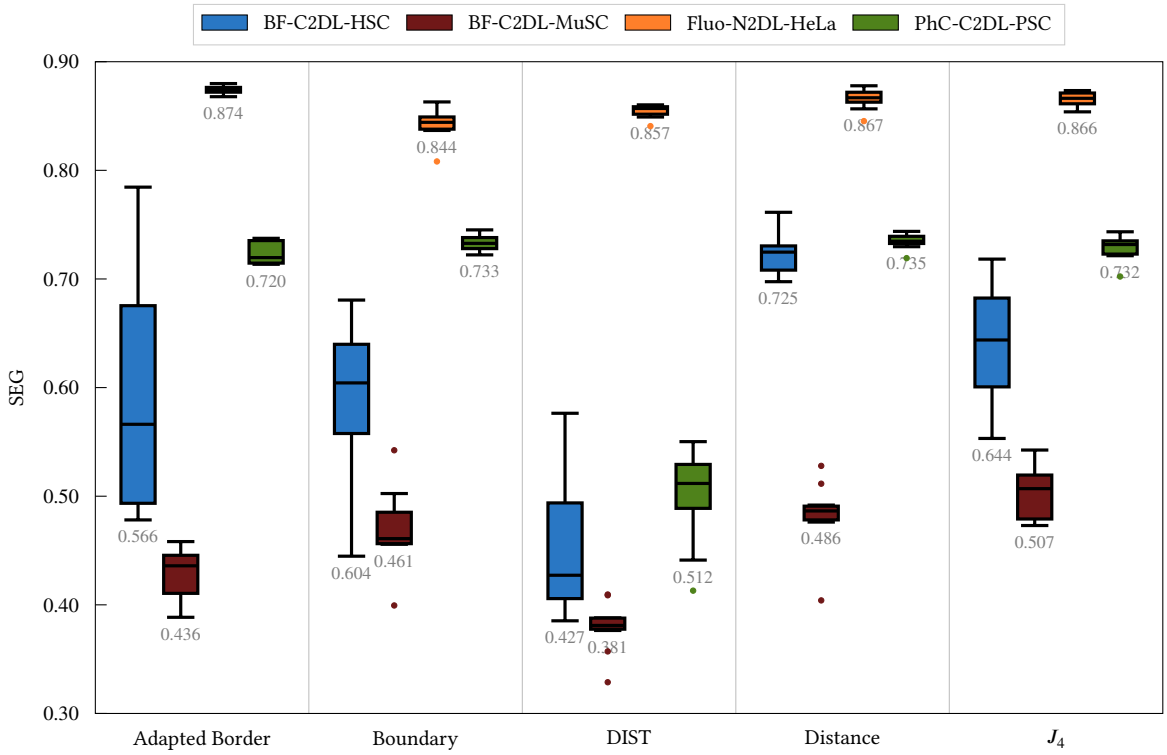


Figure 3.19: Evaluation of the Merging Post-Processing on the BF-C2DL-MuSC Test Data Set. The merging post-processing can reduce the split rates for the BF-C2DL-MuSC data set by merging adjacent objects without neighbor distance predictions. Those objects are probably split due to their specific shape, which leads to the extraction of multiple seeds. However, in some cases, objects are wrongly merged. Although more wrong splits are resolved than wrong merges are induced, the DET measure slightly drops since merges are penalized stronger. The median values are shown below the lower whiskers.

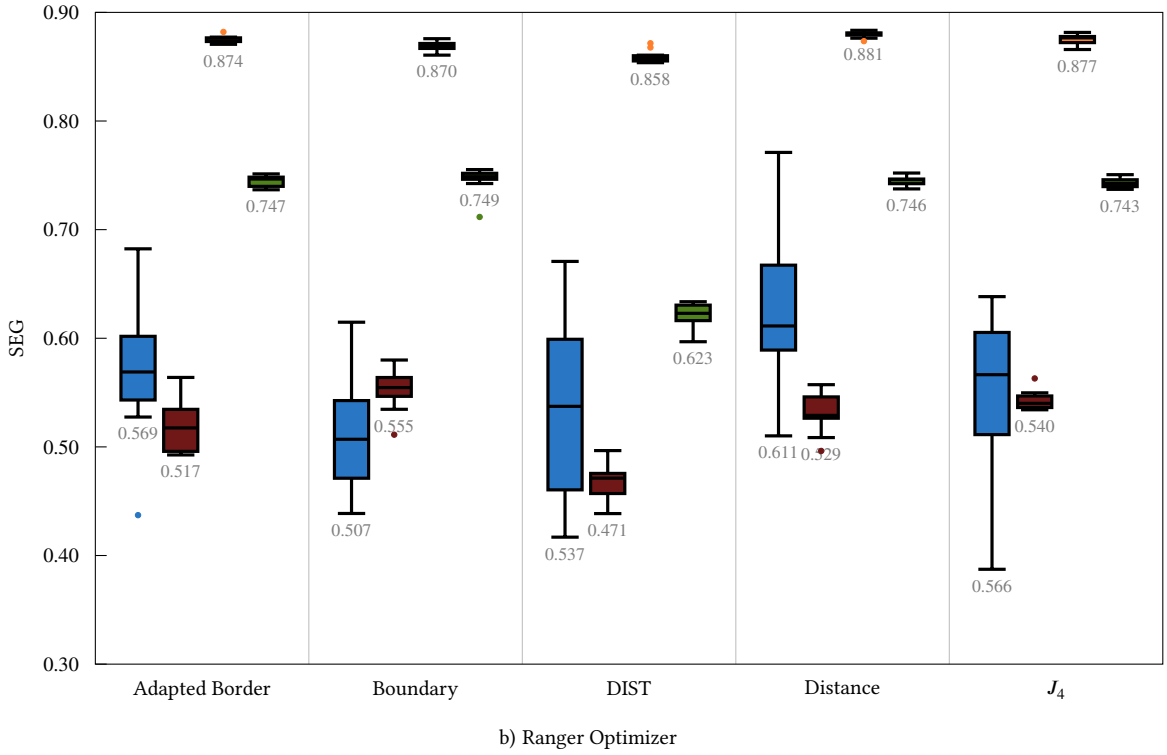
complement to the detection quality measure DET. Figure 3.20 compares the SEG scores of the compared method for the four test data sets. Obviously, the scores differ heavily for the single cell types. The differences have multiple reasons.

First of all, the detection quality influences how well predicted and ground truth objects match. For instance, a split object cannot agree well with the corresponding ground truth object, and each missing object contributes to the SEG score with a Jaccard similarity index of 0. Second, the Cell Tracking Challenge provides much fewer segmentation ground truths than detection ground truths (1340 vs. 154 027 for the four test data sets). Thus, false negatives of the annotated ground truth cells can significantly influence the SEG estimation, even when the miss rates are negligible overall. Furthermore, the four test data sets have different imaging conditions, contrasts, and cell morphologies. At last, reaching a high Jaccard similarity index for small objects like the hematopoietic and the pancreatic stem cells is more difficult than for the larger HeLa cells. This is because a single-pixel difference between two predictions has a larger impact on the Jaccard similarity index for small structures [211].

The Ranger optimizer provides an improved median SEG score in sixteen cases and only a decreased median score in three cases compared to the Adam optimizer results. Especially for the BF-C2DL-MuSC data set, the scores rise, which should only partially be due to an improved detection quality. In contrast, the segmentation performance drop for BF-C2DL-HSC when using



a) Adam Optimizer



b) Ranger Optimizer

Figure 3.20: SEG Measures for the Four Test Data Sets. 11 models have been trained with the Adam optimizer and 11 with the Ranger optimizer for each method. All models are evaluated on the four test data sets separately. The median values are shown below the lower whiskers. The SEG measure is influenced by object level-errors like splits, merges, and false negatives, by pixel-level errors, and by the object size itself (smaller objects need to be segmented more accurate).

Table 3.8: Cases with Significant SEG Score Improvements. The number of cases of a method being significantly better than the baseline method is stated. A method can reach eight significant improvements in each comparison (one for each data set and optimizer). See Appendix A for the statistical significance tests.

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	3	7	1	1
Boundary	3	-	6	1	0
DIST	0	1	-	0	0
Distance	5	4	8	-	3
J_4	2	3	7	0	-

the distance method and the Ranger optimizer is due to the detection quality drop caused by an increased number of missing cells.

The high variance in the SEG measures for the mouse hematopoietic stem cells cannot be explained by the issue of segmenting parts of the hydrogel well since adding objects does not affect the SEG measure. Possible reasons are the holes in the segmentation masks some methods produce, a high merge rate for the boundary and J_4 method, and a high false negative rate, for the DIST method especially. The distance method and the Ranger optimizer provide the most stable and accurate results.

Splitting the cells in the elongated state mainly limits the SEG scores for the BF-C2DL-MuSC data set. In addition, some cell parts with shallow contrast are sometimes missing. Interestingly, using the J_4 or the boundary method results in a higher median SEG score than the distance method, although the distance method produces much better DET scores. Therefore, the sizes and shapes of correctly segmented cells are less accurate when using the distance method than the other two methods. A possible reason may be the separate distance normalization for each object in the object distance data representation. Thus, using a single object-size independent threshold t_1 in the post-processing could have resulted in segmenting the larger elongated cells too small.

For the data sets Fluo-N2DL-HeLa and PhC-C2DL-PSC, the results of the single methods are close together. Only the DIST method cannot keep up with the other methods for segmenting pancreatic stem cells. The consistency of the methods and the single models may come from the fact that those two cell types provide about 89 % of the cells in the CTC training data set. The higher miss and merge rates for the PhC-C2DL-PSC data set explain the lower scores compared to the Fluo-N2DL-HeLa data set.

Table 3.8 shows that only two times a method is significantly better than the distance method, while the distance method can outperform each other method. See Appendix A for more information about the statistical significance testing.

Overall Performance

After evaluating the detection quality and how well segmented and ground truth masks match, the overall performance measure $\mathrm{OP}_{\mathrm{CSB}}$ enables to evaluate and rank the methods with a single evaluation measure that takes pixel-level and object-level errors into account. The $\mathrm{OP}_{\mathrm{CSB}}$ results are shown in Figure 3.21. As the overall performance measure $\mathrm{OP}_{\mathrm{CSB}}$ is the mean of the DET and the SEG measure, the former explanations for performance drops also apply here.

The Fluo-N2DL-HeLa data set can be segmented well and the differences between the single methods are small for this data set. However, for the other data sets, method improvements, sophisticated post-processing approaches including, for instance, false positive filtering, or larger

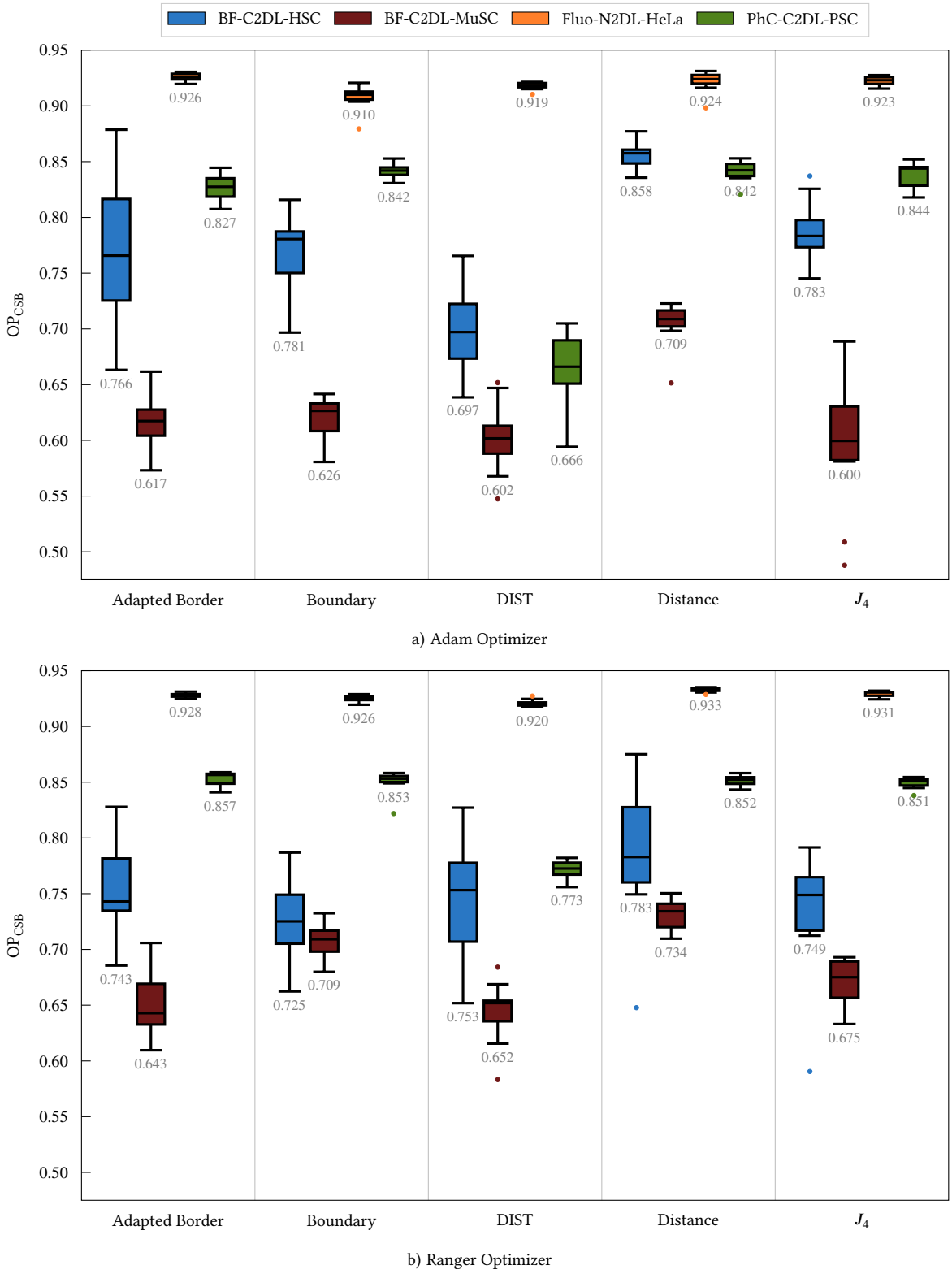


Figure 3.21: Overall Performance Measures for the Four Test Data Sets for the Four Test Data Sets. 11 models have been trained with the Adam optimizer and 11 with the Ranger optimizer for each method. All models are evaluated on the four test data sets separately. The median values are shown below the lower whiskers. The overall performance measure OP_{CSB} is the mean of the DET measure and the SEG measure.

Table 3.9: OP_{CSB} Rankings. Each method's higher median score (Adam, Ranger) is compared for a single data set from Figure 3.21 are compared for the rankings. The new distance method provides the best OP_{CSB} for three of the four data sets score, and the new adapted border method provides the best score for the fourth data set.

Method	BF-C2DL-HSC	BF-C2DL-MuSC	Fluo-N2DL-HeLa	PhC-C2DL-PSC
Adapted border method	4th	5th	3rd	1st
Boundary method	3rd	2nd	4th	2nd
DIST method	5th	4th	5th	5th
Distance method	1st	1st	1st	3rd
J_4 method	2nd	3rd	2nd	4th

Table 3.10: Cases with Significant OP_{CSB} Score Improvements. The number of cases of a method being significantly better than the baseline method is stated. A method can reach eight significant improvements in each comparison (one for each data set and optimizer). None of the methods is significantly better than the proposed distance method. See Appendix A for the statistical significance tests.

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	2	5	0	1
Boundary	2	-	5	0	1
DIST	0	1	-	0	0
Distance	5	6	6	-	4
J_4	1	2	6	0	-

training data sets are needed for a near error-free performance. This finding could be expected for the two mouse stem cell types since the number of those cells in the CTC training data set is small.

Finally, the higher of the two median scores (Adam, Ranger) of a method for a single data set are compared to create a method ranking. This comparison yields the rankings in Table 3.9. The new distance method provides the best OP_{CSB} for three of the four data sets score, and the new adapted border method provides the best score for the fourth data set. Table 3.10 shows that no method is significantly better than the distance method. However, the distance method can outperform each other method. See Appendix A for more information about the statistical significance testing and the results for the single data sets.

3.3 Software: microbeSEG

With the rise of deep learning methods for segmentation also came a need to make them accessible in user-friendly tools that do not require expert programming or deep learning knowledge. For instance, in [62], it is envisioned that replacing existing analysis pipelines with more accurate deep learning counterparts would greatly aid researchers who conduct live-cell imaging experiments. Moreover, a survey of 704 National Science Foundation principal investigators identified the current and future data analysis needs (i) sufficient data storage, (ii) updated analysis software, (iii) training on data management and metadata, (iv) support for bioinformatics and analysis, and (v) training on basic computing and scripting [229].

microbeSEG is a new deep learning-based tool with a graphical user interface for the instance segmentation of microscopy images. Similar to BeadNet, OMERO is used for data management. Users need basically only to select a segmentation method and a crop size for training data creation. The other parameters are pre-defined and work well for many applications. The microbeSEG pipeline

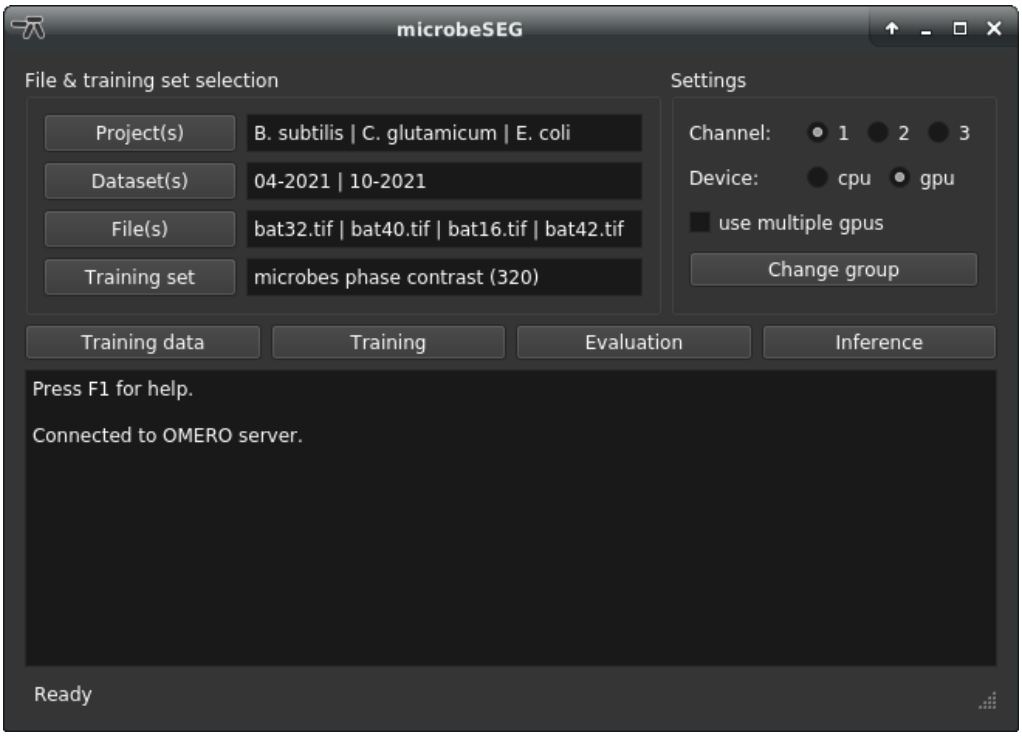


Figure 3.22: Graphical User Interface of microbeSEG. The minimalist design with the pre-defined parameters and the workflow including pre-labeling with subsequent correction enable using deep learning for instance segmentation without expert knowledge. A manual with videos is available at https://github.com/hip-satomi/microbeSEG.

consists of training data creation, data handling and loading, model training, model evaluation, and finally, the application of trained models to experimental data. Figure 3.22 shows the graphical user interface of microbeSEG, which was developed within the Helmholtz Imaging Platform project SATOMI. Figure B.1 provides an overview of microbeSEG and its interaction with OMERO and the annotation tool ObiWan-Microbi [218].

3.3.1 OMERO Data Management

As introduced in Section 2.3, OMERO is an open-source software platform from the Open Microscopy Environment for accessing and using a wide range of microscopy data [219]. Over 150 image formats can be imported with the OMERO.insight desktop client, and imported data are organized into projects and data sets. After the import, the data can be processed with microbeSEG – without any data format conversion steps or programming. Images, microbeSEG training data, and microbeSEG results can easily be accessed and viewed in the browser with the OMERO.web client.

3.3.2 Training Data Creation

As own, application- and domain-specific training data are still the best choice for many applications, microbeSEG offers an easy but comprehensive training data creation workflow that includes time-efficient pre-labeling. New training data sets can be added with the graphical user interface and are managed as OMERO data sets. In doing so, a crop size must be selected. The annotation of crops avoids the time-consuming annotation of all, possibly densely packed, objects in an image, a

task that is especially time-consuming for the generation of instance segmentation annotations. Annotating smaller crops with fewer particles is more efficient since more image and object features can be covered in the same annotation time, resulting in a more diverse training data set. Therefore, microbeSEG offers an interactive crop selection interface (see Figure 3.23).

Similar to BeadNet, up to three – depending on the image and crop size – crop proposals of the selected images can be viewed and added to the training data set. The crop proposals are randomly extracted from non-overlapping image regions. Selected crops are automatically assigned to a training, a validation, or a test subset and uploaded to OMERO. If trained microbeSEG models are already available, pre-labeling enables identifying areas with segmentation errors. Adding such areas facilitates the training of well-generalizing models. In addition, this approach reduces the annotation time for correctly or partially segmented cells. Furthermore, annotated data sets can be imported, e.g., publicly available training data sets.

The selected training data crops can be annotated with ObiWan-Microbi, an open-source microservice platform for annotating object instances in the cloud [218]. This Angular- and Ionic-based web app connects to OMERO and has been developed jointly with microbeSEG. In addition, to the microbeSEG pre-labeling, Omnipose [182] and Cellpose [140] can be applied. The annotations can also be viewed with the OMERO.web client.

3.3.3 Model Training and Evaluation

Segmentation Methods

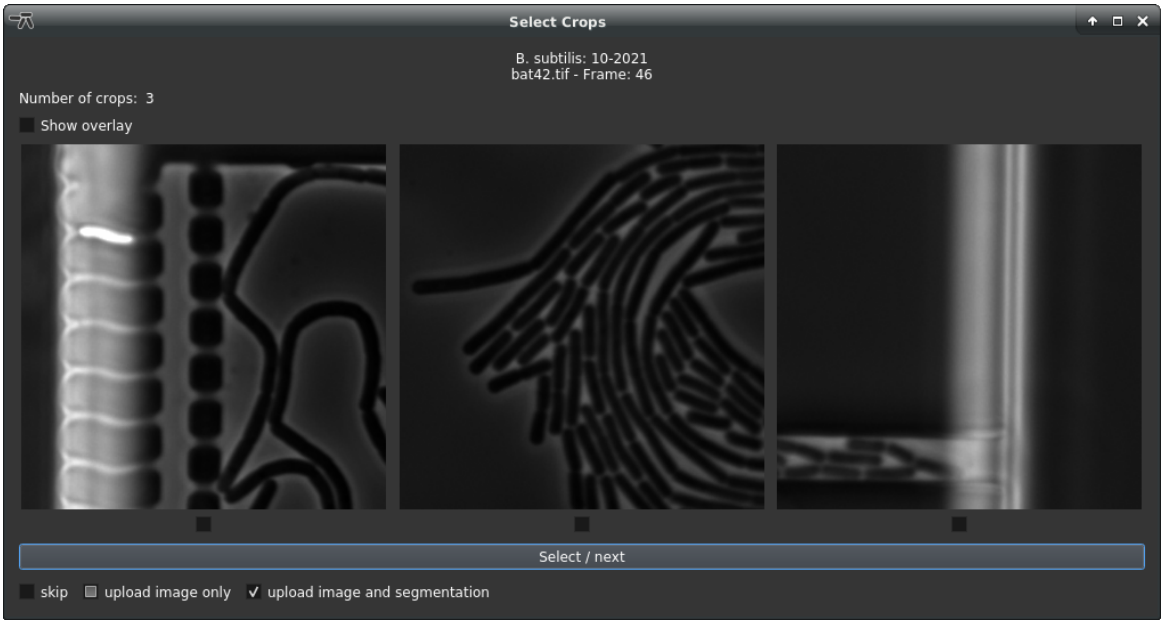
Two deep learning-based instance segmentation methods are implemented in microbeSEG: (i) the simple boundary method, which serves as an easy-to-interpret baseline, and (ii) the distance method. The used single-decoder U-Net has about 34 million parameters, and the double-decoder U-Net has 46 million. If not enough memory is available, the network sizes are automatically reduced to a minimum of 2 or 3 million parameters.

Training

The segmentation method, the batch size, the optimizer, and how many models are trained need to be specified for training a model on a selected annotated microbeSEG training data set. Adam and Ranger are available as optimizer, each with pre-defined settings (see Table 3.1). In addition, trained Ranger models are fine-tuned for 10 % N_{max} epochs with cosine annealing [230]. As before, Adam is coupled with the ReLU activation function, while the Mish activation function [221] is used for the training with Ranger. The augmentations described in Subsection 3.1.3 are applied during training. The optimization criteria are the Dice loss and cross-entropy loss based L_{boundary} loss (see Eq. 3.12) for the boundary method and the smooth L1 loss based L_{distance} loss (see Eq. 3.1) for the distance method. After training, the model checkpoint with the best validation loss is saved locally and available in microbeSEG.

Evaluation

A test subset is automatically split during the training data creation (25 % of the selected crops). Trained models can be evaluated on this test subset. During the evaluation, appropriate parameters $t_1 \in \{0.05, 0.075, 0.10, 0.125\}$ and $t_2 \in \{0.35, 0.45\}$ are set for the post-processing of the distance method. As the evaluation metric, the AJI+ score is used. The AJI+ scores and their mean and standard deviation over the test images are saved in CSV files for each evaluated model.



a) Without Overlay

b) With Overlay

Figure 3.23: microbeSEG Training Data Crop Creation Interface. Automatically proposed crops can be selected and uploaded to OMERO (a). The crop proposals are extracted randomly from different non-overlapping image regions (the left crop originates from the left image region, and the right crop from the right). For the pre-labeling, it is possible to upload only the image or the image with its prediction (b). The image-only upload is helpful when the pre-label predictions require too many manual corrections.

3.3.4 Inference, Result Analysis, and Result Export

The best-evaluated model with its parameter set is selected automatically for inference. A manual selection of a model is also possible. If a distance method model is selected that has not been evaluated, default thresholds t_1 and t_2 are applied in the post-processing. However, the default thresholds may result in (slightly) too large or too small segmented objects. The segmentation results of processed OMERO images can be stored locally or attached to the OMERO image as polygon regions of interest, enabling joint storage of images and results. The results on OMERO can be viewed with the OMERO.web client and, if required, be corrected with ObiWan-Microbi. Furthermore, the object count for each frame, the mean object area, the mean minor axis length, the mean major axis length, and the total object area can be determined. The result export includes the original image (.tif), intensity-coded instance segmentation masks (.tif), an object outlines image (.tif), the original image overlaid with the object outlines (.tif), and an analysis results CSV file.

3.3.5 Implementation, Installation, and Dependencies

microbeSEG is implemented in Python and uses PyTorch as deep learning framework. The graphical user interface is built with PyQt. The software, the source code, and a detailed step-by-step guide for easy installation and usage are available under the MIT license at https://github.com/hip-satomi/microbeSEG. The microbeSEG data set used in the microbeSEG workflow evaluation and pre-trained models are available on Zenodo [31], [217]. The models were trained on a data set consisting of 826 crops of size 320×320 px extracted from the Omnipose data set, the mentioned microbeSEG data set, and the Cell Tracking Challenge data sets BF-C2DL-HSC, BF-C2DL-MuSC, Fluo-N2DL-HeLa, and PhC-C2DL-PSC. An OMERO server can be set up during the installation of ObiWan-Microbi, which is available at https://github.com/hip-satomi/ObiWan-Microbi.

3.3.6 Workflow Evaluation

Two key features of BeadNet and microbeSEG are the use of the data management system OMERO with its versatile data importer and the workflow covering training data creation, model training, model evaluation, and the application of trained models. In Section 2.3, the BeadNet workflow has been evaluated in an experiment with a limited training data creation time of 15 minutes. A result was that high-quality results could be obtained quickly and that no extensive training data sets were needed. This section applies a similar procedure: a microbeSEG user has 30 minutes to create training data. Then, five distance method and five boundary method models are trained. After that, additional 15 minutes are spent on training data creation with pre-labeling. The training data creation time includes crop selection, annotation, and pre-label corrections. Higher training data creation times compared to the BeadNet evaluation are used since the whole contour of an object needs to be drawn to generate instance segmentation ground truths. In contrast, particle detection ground truth can be created with a single click.

Data

B. subtilis and *E. coli* data from the Forschungszentrum Jülich are used for the evaluation. The 2D+t microbe data were acquired with a fully automated time-lapse phase contrast microscope setup, and cultivation took place inside a special microfluidic cultivation device [231]. The test data set consists of 12 *B. subtilis* image crops showing in total 721 cells and 12 *E. coli* image crops with 1168 cells. The crop size is 320×320 px. Three experts annotated the test images. The experts annotated

Table 3.11: microbeSEG Accuracy After 30 Minutes and 45 Minutes of Training Data Creation. Five boundary method and five distance method models are trained on the training data sets created in 30 minutes and 45 minutes, respectively. Each trained model is evaluated on the 24 test images. The median AJI+ scores for the 12 *B. subtilis* images, the 12 *E. coli* test images, and the total test data set are reported. The times include crop selection, annotation, and pre-label corrections. The median training time τ_{train} on a system with one NVIDIA TITAN RTX GPU is stated additionally (batch size: 4). Modified from [214].

Time	N_{crops}	N_{cells}	Method	τ_{train}	AJI+$_{\text{B. subtilis}}$	AJI+$_{\text{E. coli}}$	AJI+$_{\text{total}}$
30 min†	13[a]	309	boundary (Adam)	229 s	0.484	0.653	0.568
			distance (Ranger)	822 s	0.585	0.729	0.657
30 min† + 15 min‡	22[b]	732	boundary (Adam)	276 s	0.505	0.687	0.596
			distance (Ranger)	1060 s	0.614	0.731	0.673

[a] *8 training, 3 validation, and 2 test crops (for post-processing parameter adjustment).*
[b] *13 training, 5 validation, and 4 test crops (for post-processing parameter adjustment).*
† *Same training data set.*
‡ *With pre-labeling using a distance method model trained on the data set annotated in the first 30 minutes.*

different crops but cross-checked their annotations. The microbeSEG user selected and annotated crops of size 320×320 px from different experimental data than the experts used.

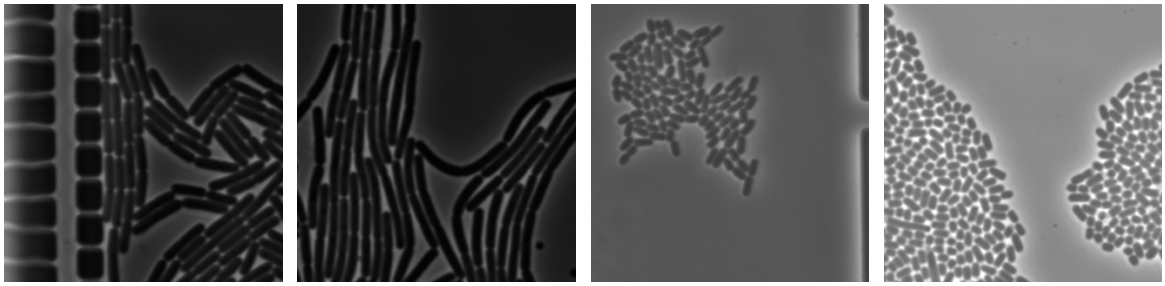
Results

Table 3.11 shows the median AJI+ scores of the boundary method and the distance method after 30 minutes and 45 minutes of training data creation time. The distance method parameters t_1 and t_2 have been automatically adjusted on from the user annotated crops. The distance method provides consistently better results than the boundary method, even when using less training data creation time for the distance method.

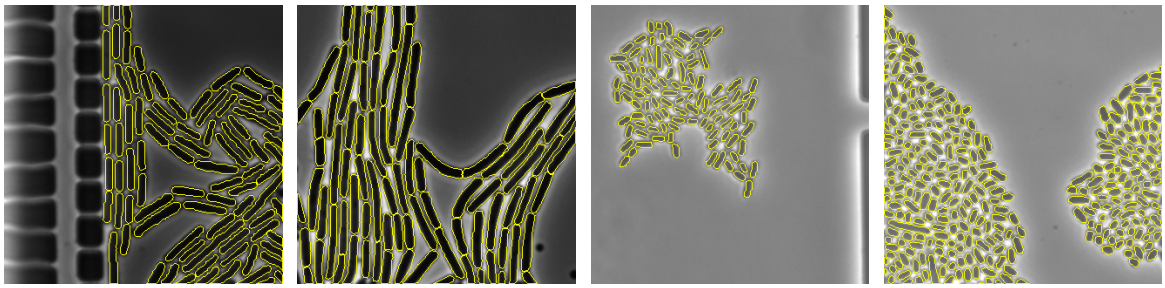
Further, the results show that once a suitable model has been trained, switching from completely manual annotation to correcting pre-labeled image crops results in more objects annotated per time, i.e., 2.8 times more cells per minute, and increased segmentation quality for both methods and each microbe type. Figure 3.24 helps to interpret the differences between the scores for the *E. coli* and the *B. subtilis* test images. The *E. coli* scores for the distance method are mainly limited by pixel-level errors, i.e., the test data set annotators and the microbeSEG user had a different decision boundary for the *E. coli* object size. In contrast, for the *B. subtilis* cells, the scores are limited by object-level errors. This behavior is mainly due to the different decision boundaries of the annotators about when cell mitosis has finished. However, both microbe types' scores should rise significantly when the same persons annotate training and test data sets. Figure 3.25 shows that the neighbor distance is especially helpful in cases with low contrast and unclear object boundaries, but it also shows the difficulty of segmenting *B. subtilis* cells.

The larger training times of the distance method compared to the boundary method are due to the use of the Ranger optimizer with other early stopping conditions (see Table 3.1), the additional cosine annealing, and the larger CNN with two decoders. However, in further experiments, reducing the model parameters could reduce the training times to about 2/3 or 3/4 of the reported times for both methods without significant segmentation performance loss.

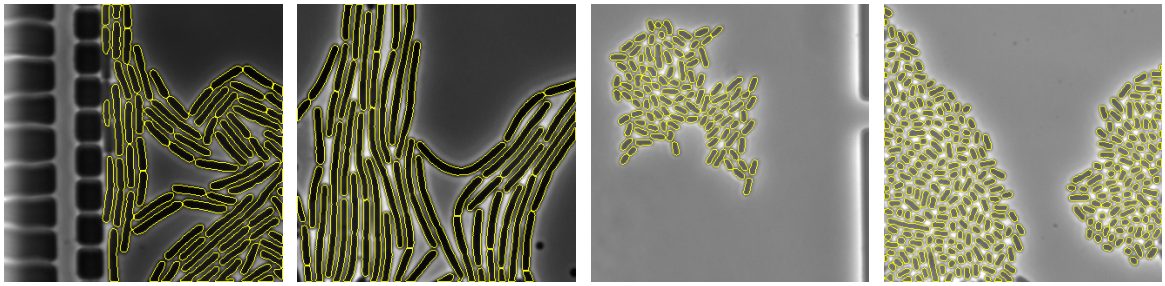
In summary, in about one hour, high-quality results are possible with microbeSEG, including training data creation and training time. Once a broad and diverse collection of annotations is available, either with manual annotation or the import functionality, pre-labeling can be applied immediately. Segmenting structures of the cultivation device is no problem as such structures are also in the created training data set.



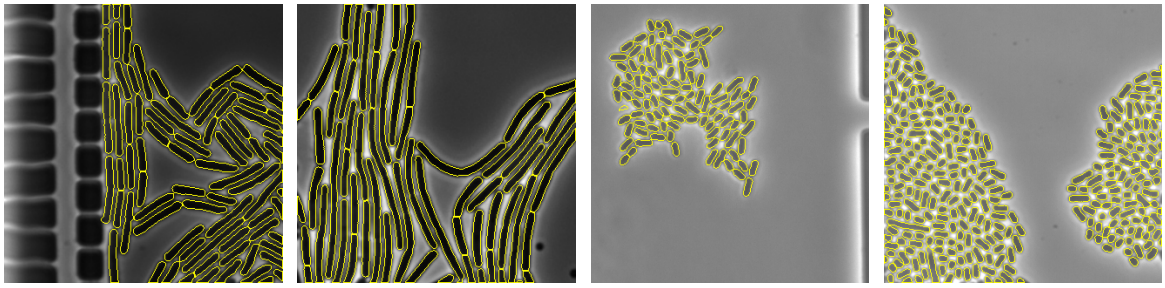
a) *B. subtilis* and *E. coli* Test Images (320 × 320 px)



b) Ground Truths



d) microbeSEG Distance Method Prediction (30 min)



d) microbeSEG Distance Method Prediction (45 min)

Figure 3.24: microbeSEG Predictions for *B. subtilis* and *E. coli* Images. Shown are the predictions of the median distance method models reported in Table 3.11. The two left images are exemplary *B. subtilis* test images, and the right two are exemplary *E. coli* test images. Instances are not color-coded, but the object contours are drawn for better readability. Rearranged from [214].

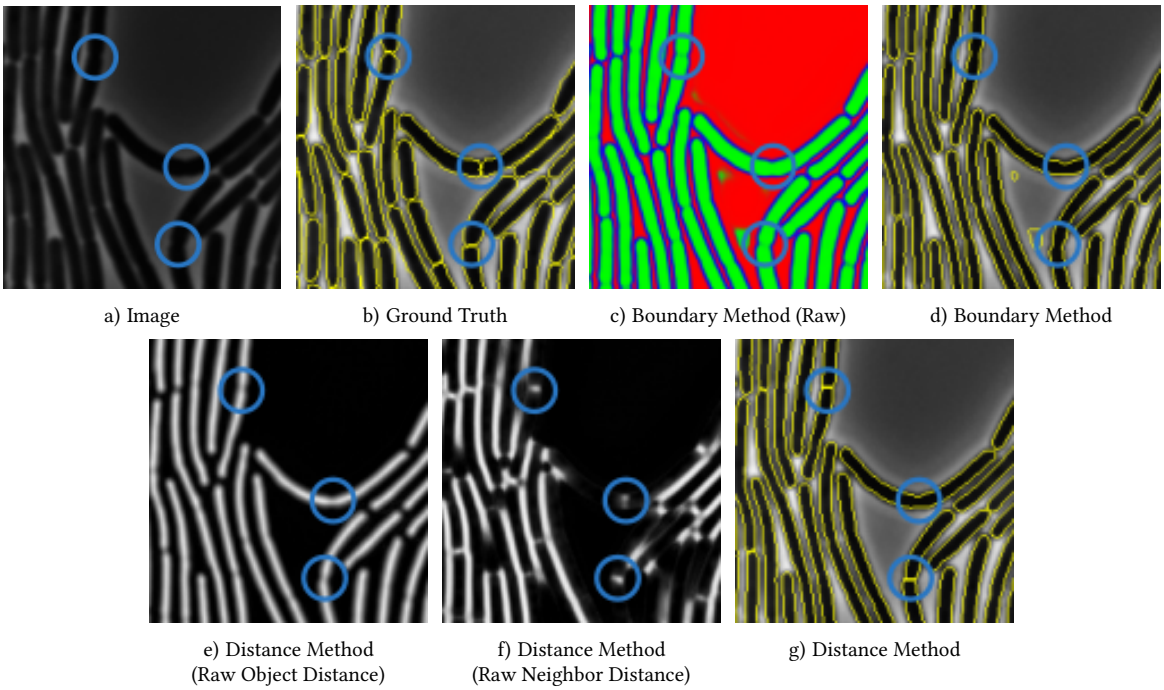


Figure 3.25: Unclear Object Boundaries for *B. subtilis*. Shown are the predictions of the median distance and boundary method model reported in Table 3.11 (45 min). The neighbor distance prediction prevents merging in two of the three highlighted cases. However, the low contrast results in an unclear decision boundary about when *B. subtilis* cells have divided into two cells. The image is a closer view of the second *B. subtilis* test image from Figure 3.24 (second image from the left). Modified from [214].

3.3.7 Key Feature Comparison

Table 3.12 provides an overview of crucial features a modern and easy-to-use microscopy image instance segmentation software should have. Data management systems that can store, visualize, and share data, metadata, and results facilitate using segmentation tools. When tools support only a limited number of data formats, additional data conversion steps are required to use the tool. For deep learning software, a workflow with training data crop creation, annotation, pre-labeling, and training is beneficial. However, so far, most deep learning software with a graphical user interface focuses on applying pre-trained models, and model training is not directly possible and requires programming expertise. Further conversion steps are required if segmentation and annotation tools are not designed to work together. Utilizing the jointly developed annotation tool ObiWan-Microbi, microbeSEG is the only tool that covers all key features.

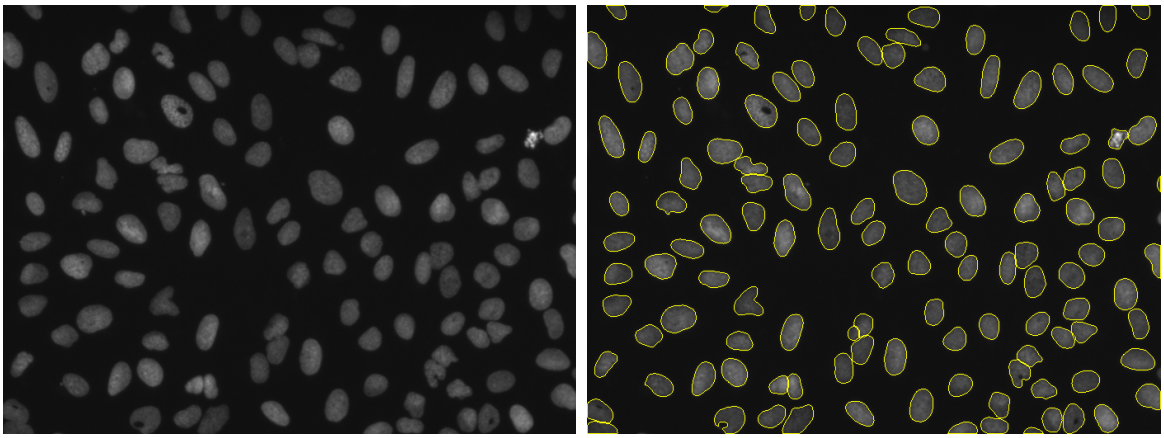
3.4 Applications

This section shows first qualitatively exemplary use cases of the distance method and the broad applicability of microbeSEG to various microscopy imaging techniques and objects. Afterward, Cell Tracking Challenge results are presented, which give further insight into the competitiveness of the method.

Table 3.12: Key Feature Comparison of Instance Segmentation Software for Microscopy Images. A good segmentation software should not require expert knowledge or programming expertise. However, traditional segmentation methods may require expert knowledge for parametrization and are not state-of-the-art anymore. Thus, deep learning tools with a versatile data management system, support of many data formats, training data creation with cropping, annotation, and pre-labeling, model training, and result correction are required. Considered are only tools with a graphical user interface. Support of the stated data formats does not necessarily mean that each image can be processed without conversion steps if no data management system with metadata support is used, e.g., the channel dimension can be the first or the last dimension for .tif files, and the method may have requirements on the channel dimension position. –: feature not fulfilled/supported, ⊘: feature only fulfilled/supported with restrictions, ✔: feature fulfilled/supported. Modified from [214].

Software	Data Management	Data Formats	Deep Learning	Cropping	Annotation	Pre-labeling	Training	Result Correction
AutoCellSeg [202]	–	.jpg, .png, .tif, .bmp	–	–	–	–	–	✔
BacStalk [203]	–	.jpg, .tif	–	–	–	–	–	–[a]
Cellpose/Omnipose [140], [182], [198]	–	.gif, .jpg, .png, .tif	✔	–	✔	✔	✔	✔
ChipSeg [192]	–	.tif	–	–	–	–	–	–
DeepImageJ [196] (Fiji plugin [110])	–	> 150[b]	✔	–	–	–	–	–
microbeSEG	OMERO	> 150[b]	✔	✔	✔[g]	✔	✔	✔[g]
Misic [206] (napari plugin [201])	–	> 150[b]	✔	–	⊘[c]	–	–	–
Orbit [207]	OMERO	> 150[b]	⊘[d]	–	✔	–	⊘[e]	–
StarDist [138] (napari plugin [201])	–	> 150[b]	✔	–	⊘[c]	–	⊘[f]	–
YeaZ [164]	–	25	✔	–	–	–	–	✔
YeastSpotter [209]	–	N.A.	✔	–	–	–	–	–

[a] *Only deletion of objects, no shape refinement or correction of false negatives, merges and split objects.*
[b] *OME Bio-Formats support (napari needs the napari-aicsimageio plugin).*
[c] *In principle possible with napari (plugins) but no training functionality with the graphical user interface.*
[d] *With copy and paste of a segmentation script into the script editor.*
[e] *Only with a Python script outside of Orbit.*
[f] *Not within the graphical user interface, but in principle within the napari IPython console.*
[g] *With the associated tool ObiWan-Microbi.*



a) Image b) Image-Segmentation Overlay

Figure 3.26: Segmentation of U2OS Cells with HeLa Cell Training Data. The microbeSEG distance method model has been trained on HeLa cells from the Cell Tracking Challenge. Data source: BBBC039 [32]. Taken from [214].

3.4.1 Cell and Cell Nucleus Segmentation

Segmentation of U2OS Cell Nuclei with HeLa Cell Training Data

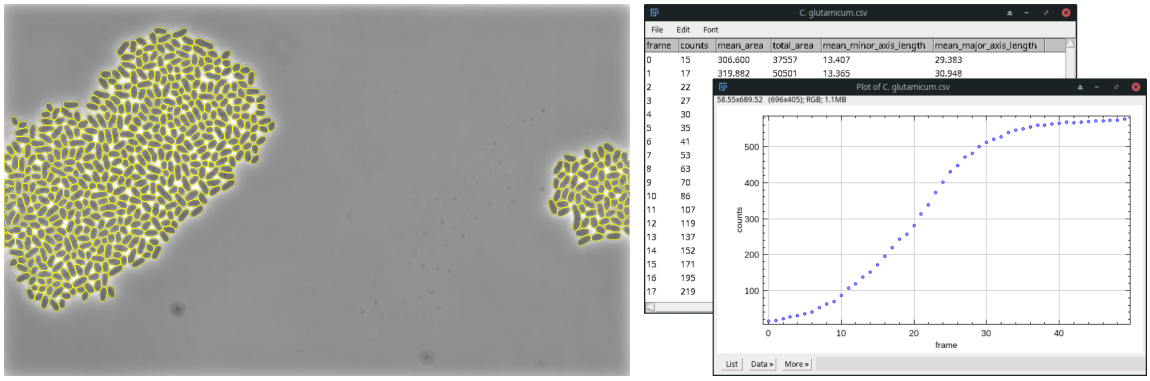
The method validation in Section 3.2 showed that the distance method provides high-quality cell and cell nucleus instance segmentation results. Furthermore, multiple cell types have been segmented well with a single model. Figure 3.26 illustrates that cells or cell nuclei similar to those in the training data can also be segmented without re-training. Thus, the provided microbeSEG models [217], trained on larger and more diverse training data sets, should be a good starting point for many applications.

Growth Analysis of Microbial Cultures

Cell growth is important for the characterization and optimization of microbial cultures. Interesting growth parameters are, for instance, the number of cells over time or size distributions. Acquired 2D+t data can be segmented and analyzed with microbeSEG. Figure 3.27 shows the results for an in a microfluidic cultivation device growing *E. coli* colony.

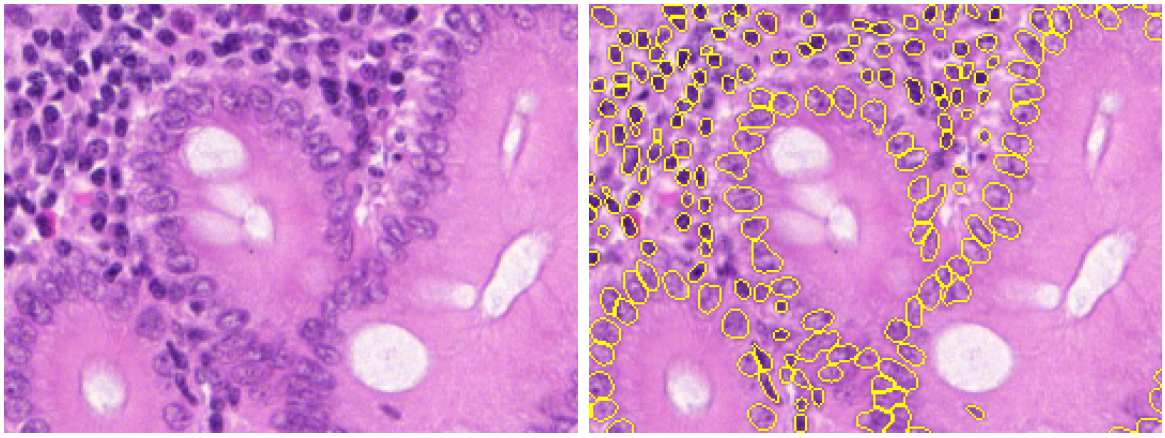
Segmentation and Counting of Colon Nuclei

The segmentation, classification, and quantification of cell nuclei in histology images enable extracting of interpretable features useful for explainable computational pathology [186]. microbeSEG can provide the cell nucleus segmentation, as Figure 3.28 shows. The classification of segmented nuclei can be done with a separate classification network. Another approach is to adapt the distance method for combined cell nucleus segmentation and classification as the single-branch CNN ciscNet does [232]. ciscNet builds upon the distance method and training process but without the neighbor distances due to the focus on classification and ranked ninth in the CoNIC: Colon Nuclei Identification and Counting Challenge 2022, beating the former baseline HoverNet [157], [233]. In addition, the fifth place was reached in the post-challenge analysis that compared methods trained with the same training data split and without ensembling techniques [233].



a) Image-Segmentation Overlay (Frame 46) b) Analysis Results Opened in Fiji [110]

Figure 3.27: Cell Growth of an *E. coli* Colony. The microbeSEG distance method model has been trained on the microbeSEG data set [31]. Taken from [214].



a) Image b) Image-Segmentation Overlay

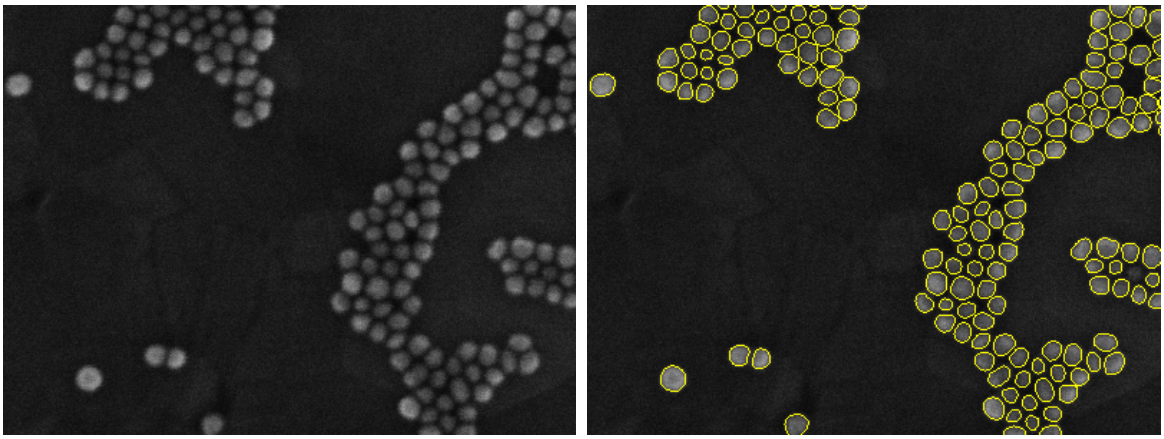
Figure 3.28: Segmentation of Colon Nuclei. The microbeSEG distance method model has been trained on a subset of the Lizard data set and applied to a hold-out Lizard data set image [29].

3.4.2 Nanoparticle Characterization

Chapter 2 shows that nanoparticles can be localized and counted with BeadNet. microbeSEG can also deliver size and shape information at the cost of higher annotation time. However, some nanoparticles can also be segmented without application-specific training data using a model trained on a large and diverse data set, as Figure 3.29 shows. Generally, it is more likely that such a model can be used for pre-labeling to create a sufficient nanoparticle segmentation training data set in a reasonable time.

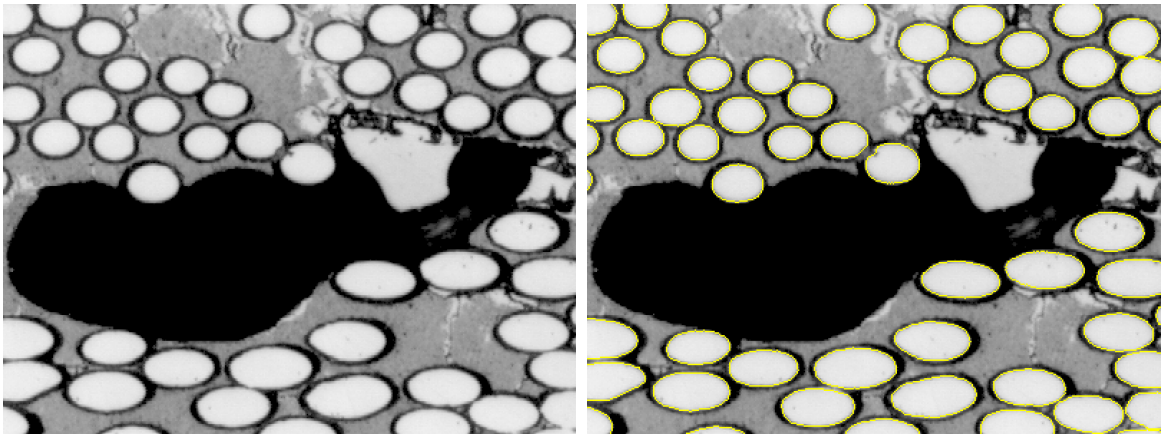
3.4.3 Fiber Detection

Material scientists need an accurate characterization of fiber microstructures to analyze the physical properties of continuous fiber-reinforced composite materials [235]. Acquired cross-sections of a 3D sample can be processed slice-by-slice, and the detections need to be combined to reconstruct the 3D object. Figure 3.30 shows the segmentation of a cross-section. microbeSEG can robustly detect the fibers in the shown cross-section.



a) Image b) Image-Segmentation Overlay

Figure 3.29: Nanoparticle Segmentation. One of the provided on cell data pre-trained microbeSEG models [217] has been applied to segment nanoparticles in scanning electron microscopy images. Only a single dark nanoparticle is missing, a generalization problem that can be solved by creating application-specific training data. Data source: NFFA-EUROPE - 100% SEM Dataset [222], [223].



a) Image b) Image-Segmentation Overlay

Figure 3.30: Fiber Detection. microbeSEG can also segment elliptical fibers in materials science data after creating some training data (12 crops of size 256×256 px). Data source: [234].

3.4.4 Cell Tracking Challenge

The Cell Tracking Challenge is an ongoing initiative to promote the development and objective evaluation of cell segmentation and tracking algorithms [27], [28]. Besides the possibility of submitting results for evaluation at any time, with monthly evaluation and ranking, six Cell Tracking Challenge editions were hosted by the IEEE International Symposium on Biomedical Imaging (ISBI). The challenge consists of a Cell Segmentation Benchmark and a Cell Tracking Benchmark, which both share the same data sets. In 2019, computer-generated segmentation silver truths constructed using former submissions have been added for nine data sets (see [236] for the silver truth generation). Further silver truth annotations have been created for the sixth challenge edition at the ISBI 2021. Currently, thirteen data sets contain silver truths in addition to the human-made gold truths.

5th Challenge Edition at the IEEE ISBI 2020

The 5th Cell Tracking Challenge edition at the ISBI 2020 was the first edition with the availability of silver truth annotations. The distance method provided the segmentation for the submission of the team KIT-Sch-GE (1), which has later been renamed from the challenge organizers to KIT-GE (2). A subsequent graph-based tracking has been applied for cell tracking [167]. The segmentation goal of the team was to train a single distance method model to segment thirteen different cell types.

Therefore, the training data set consisted of 997 crops of size 256×256 px extracted from BF-C2DL-HSC gold truths, BF-C2DL-MuSC gold truths, Fluo-N2DH-GOWT1 gold and silver truths, Fluo-N2DH-SIM+ gold truths, Fluo-N2DL-HeLa silver truths, Fluo-N3DH-SIM+ gold truth slices, Fluo-N3DH-CE gold truth slices, Fluo-C3DL-MDA231 gold truth slices, and some slices from the non Cell Tracking Challenge 3D data sets, i.e., synthetic HL60 cells [237], synthetic cells from [238], and *Drosophila melanogaster* cells [43]. In contrast to Section 3.2, both provided subsets of a Cell Tracking Challenge data set were used for training. The distance method used was a preliminary version of the method described in Section 3.1. In particular, the training process parameters, the augmentation sequence and parameters, the normalization, closing, and scaling steps in the neighbor distance map creation, and the amount of blur and the seed extraction step in the post-processing differ. The influences of the training process adaptations and neighbor distance refinements changes are summarized in [168]. 3D data sets have been processed slice-by-slice, and the post-processing has been straightforwardly adapted to 3D. The method description, the parameter configuration for the single data sets, and the with the Adam optimizer trained model together with an executable are available at http://celltrackingchallenge.net/participants/KIT-GE/.

Table 3.13 shows the Cell Segmentation Benchmark results. A total of seven top-3 rankings were achieved for the DET measure, seven top-3 rankings for the SEG measure, and eight top-3 rankings for the OP_{CSB} measure. The results show the competitiveness of the distance method, although the preliminary training process used may have been a bottleneck for some data sets, as indicated in [168]. Furthermore, the distance method provides reasonable segmentation results for data sets without cell type specific training data in the training data set. The DET score for the data set Fluo-N2DL-HeLa of 0.992 is higher than the reported inter-annotator agreement of 0.987 ± 0.002 from the three annotators. Therefore, a segmentation quality comparable to that of a human expert has been reached for this data set. The drop in the detection ranking for the simulated data set Fluo-N2DH-SIM+ could be due to using only 35 of the 215 gold truth frames available for the training data generation. This design decision was made to focus on non-simulated data sets. However, a drawback was that over-segmentation occurred during cell divisions for this

Table 3.13: Cell Segmentation Benchmark Results (5th Cell Tracking Challenge Edition at the ISBI 2020). Shown are the results of a single distance method model applied to various data sets from the Cell Tracking Challenge. The corresponding Cell Segmentation Benchmark leaderboard is available at: http://celltrackingchallenge.net/files/leaderboards/CSB/2020-04-03.png (team KIT-Sch-GE). State of results: April 3rd, 2020.

Data Set	DET	SEG	OP_{CSB}	Ranking DET	Ranking SEG	Ranking OP_{CSB}
BF-C2DL-HSC	0.974	0.750	0.862	2nd	2nd	2nd
BF-C2DL-MuSC	0.977	0.702	0.839	2nd	1st	1st
Fluo-C3DH-H157†	0.949	0.789	0.869	4th	5th	4th
Fluo-C3DL-MDA231	0.851	0.616	0.733	5th	3rd	3rd
Fluo-N2DH-GOWT1	0.950	0.828	0.889	7th	16th	14th
Fluo-N2DL-HeLa	0.992	0.895	0.944	3rd	6th	3rd
Fluo-N3DH-CE	0.930	0.729	0.830	2nd	1st	1st
Fluo-N3DH-CHO	0.945	0.871	0.908	2nd	7th	3rd
Fluo-N3DL-DRO	0.761	0.562	0.661	4th	2nd	4th
Fluo-N3DL-TRIC†	0.961	0.821	0.891	2nd	1st	2nd
Fluo-N3DL-TRIF†	0.926	0.601	0.763	3rd	3rd	3rd
Fluo-N2DH-SIM+	0.949	0.800	0.875	12th	5th	9th
Fluo-N3DH-SIM+	0.937	0.668	0.802	4th	4th	4th

† *No data of that cell type was used to train the segmentation model.*

particular data set, which probably could have been avoided using more training data. The distance method segmentation also builds a good basis for the subsequent tracking, as nine top-3 rankings have been achieved in the overall performance measure of the Cell Tracking Benchmark.

6th Challenge Edition at the IEEE ISBI 2021 – Primary Track

The primary track of the 6th Cell Tracking Challenge edition at the ISBI 2021 focused on studying the generalization ability of segmentation methods. Therefore, thirteen data sets have been selected by the organizers: BF-C2DL-HSC, BF-C2DL-MuSC, DIC-C2DH-HeLa, Fluo-C2DL-MSC, Fluo-C3DH-A549, Fluo-C3DH-H157, Fluo-C3DL-MDA231, Fluo-N2DH-GOWT1, Fluo-N2DL-HeLa, Fluo-N3DH-CE, Fluo-N3DH-CHO, PhC-C2DH-U373, and PhC-C2DL-PSC. Only teams with highly competitive methods, i.e., an OP_{CSB} score higher than the third-ranked method minus the standard deviation within the individual annotators for at least one of the thirteen included data sets, were allowed to compete in the primary track.

In the primary track, qualified teams needed to submit six results per data set using different configurations:

1. inference with a model trained with gold truths of a particular data set (13 models),
2. inference with a model trained with silver truths of a particular data set (13 models),
3. inference with a model trained with a mixture of gold and silver truths of a particular data set (13 models),
4. inference with a model trained with gold truths of all 13 data sets (1 model for all data sets),
5. inference with a model trained with silver truths of all 13 data sets (1 model for all data sets),
6. inference with a model trained with a mixture of gold and silver truths of all 13 data sets (1 model for all data sets).

In total, 78 results needed to be submitted, and 42 models were needed to produce the results.

Table 3.14: Primary Track Results of the 6th Cell Tracking Challenge Edition at the ISBI 2021. The distance method (team KIT-Sch-GE (2) / KIT-GE (3)) reached the first rank for each evaluated configuration. $\overline{\text{OP}}_{\text{CSB}}$ is the mean OP_{CSB} score over all 78 submitted results. $\overline{\text{OP}}^{\star}_{\text{CSB}}$ is the mean score for the 13 data sets using the best configuration per data set, e.g., the training on silver truths. The results have been presented at the 2021 ISBI Cell Tracking Challenge workshop [239].

Team[†]	Category	$\overline{\text{OP}}_{\text{CSB}}$	$\overline{\text{OP}}^{\star}_{\text{CSB}}$	Average Ranking
KIT-Sch-GE (2) / KIT-GE (3)	RM	0.842	0.877	1.00
PURD-US	SM	0.783	0.843	2.00
MU-Ra-US / MU-US (4)	SM	0.578	0.807	3.67

[†] *Two team names are stated for teams that have been renamed by the challenge organizers in 2022.*

In addition, the post-processing parameter configurations were fixed for each model, resulting in 42 parameter configurations. The training data set creation needed to be fully automated and reproducible, e.g., no manual crop selection was allowed.

This time participated as team KIT-Sch-GE (2), which has later been renamed from the challenge organizers to KIT-GE (3), the best mean overall performance score $\overline{\text{OP}}_{\text{CSB}}$ over all 78 results, and the best mean overall performance score $\overline{\text{OP}}^{\star}_{\text{CSB}}$ over the 13 results using the best configuration for a data set have been reached (see Table 3.14). The results may indicate a better generalization ability of regression methods than semantic methods. However, different training strategies and parameters, training data selection, and CNN architectures are other reasons for the differences. The primary track has been reopened after the ISBI 2021 giving participants additional two months for method development and adjustments. The final results are published in [236]. In contrast to many other teams, the team KIT-Sch-GE (2) decided not to improve the submission further using information about the results of the single configurations presented at the ISBI Cell Tracking Challenge workshop but to focus on making the distance method accessible with microbeSEG. The detailed results for each configuration in Table 3.15 show that, for instance, excluding training data from the rather exotic data set Fluo-C2DL-MSC in configuration 4 may have been a wrong choice. Still, a close second place behind the team CALT-US, which improved to a $\overline{\text{OP}}_{\text{CSB}}$ score of 0.849 and a $\overline{\text{OP}}^{\star}_{\text{CSB}}$ score of 0.878, was reached. Overall, specialized models (configurations 1, 2, and 3) yield slightly better results than non-specialized models (configurations 4, 5, and 6), and configuration 3, using specialized models and a mixture of gold and silver truth, provides the best results. However, the differences are small and more trained models are needed for further analysis.

A crucial element of the training pipeline was to extract 320×320 px crops from the possibly sparsely annotated segmentation gold truths and compare them with the fully annotated detection gold truth markers. With this approach, it was possible to use only image regions with almost all cells annotated for the training. No quality checks have been applied for the potentially erroneous silver truth, but 3D masks were closed with morphological operations before extracting 2D crops. In all configurations using silver truth, the number of silver truth crops per data set has been limited to 280 to reduce the training time. For the mixture of gold and silver truths, only silver truths from frames without gold truth have been used, and the relative amount of silver truth has been further limited to focus on human-made gold truth. For each configuration, one model has been trained with the Ranger optimizer and Mish activation function and additional cosine annealing, and one model with the Adam optimizer and the ReLU activation function. For the first configuration, two models have been trained with each optimizer. The models with the highest OP_{CSB} scores on the publicly available data have been selected for submission except for the second and fifth configuration, where the models with the highest SEG scores calculated with silver truths have

Table 3.15: Detailed Primary Track Results of the Distance Method. Shown are the OP_{CSB} results for each primary track configuration [236]. The best configuration for a data set is highlighted in bold.

Data Set	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5	Config. 6
BF-C2DL-HSC	**0.916**	0.896	0.908	0.901	0.881	0.904
BF-C2DL-MuSC	0.848	**0.875**	0.872	0.850	0.854	0.843
DIC-C2DL-HeLa	0.864	**0.865**	0.853	0.845	0.861	0.824
Fluo-C2DL-MSC	0.565	**0.689**	0.678	0.049	0.641	0.558
Fluo-C3DH-A549	0.920	0.907	**0.925**	0.866	0.694	0.802
Fluo-C3DH-H157	0.911	0.920	**0.926**	0.848	0.916	0.865
Fluo-C3DL-MDA231	0.814	0.779	0.800	0.795	0.784	**0.822**
Fluo-N2DH-GOWT1	0.900	0.872	0.881	**0.921**	0.887	0.896
Fluo-N2DL-HeLa	0.933	**0.939**	0.938	0.933	0.936	0.928
Fluo-N3DH-CE	0.799	0.789	0.799	**0.809**	0.800	0.800
Fluo-N3DH-CHO	0.925	0.848	0.871	**0.926**	0.895	0.915
PhC-C2DH-U373	0.924	**0.927**	**0.927**	0.903	0.909	0.903
PhC-C2DL-PSC	0.827	**0.865**	0.853	0.620	0.859	0.850
Mean score	0.857	0.859	**0.865**	0.790	0.840	0.839

been selected. Thus, for the second and fifth configuration, the gold truth has neither been used for training nor for model selection.

6th Challenge Edition at the IEEE ISBI 2021 – Secondary Track

Contrary to the primary track, the secondary track was open for submissions without qualification criteria and training data configuration constraints. For the KIT-Sch-GE (2) secondary track submission, the segmentations using the third configuration from the primary track have been coupled with an improved graph-based tracking with only a few manually tunable parameters and automated segmentation error correction [240]. Thus, instead of using a single generalizing model as in 2020, per data set specialized models have been used. The method description, the parameter configuration for the third configuration, and the submitted models, together with an executable, are available at http://celltrackingchallenge.net/participants/KIT-GE/.

Table 3.16 shows the secondary track results. Eight top-3 rankings were achieved for the DET measure, five top-3 rankings for the SEG measure, and five top-3 rankings for the OP_{CSB} measure. Among them, two first OP_{CSB} rankings could be reached for a bright-field data set, one for a fluorescence data set, and one for a phase contrast data set. Comparing the scores to the submission with a single model a year before reveals some improvements, like for BF-C2DL-HSC and BF-C2DL-MuSC, but also some performance decreases. The improvements may be due to the availability of silver truths for more data sets and the training process and method refinements of the newer submission. The performance loss may be due to a less well adjusted threshold t_1 that affects the cell size and, therefore, the SEG metric, and due to a simplified post-processing. In addition, it may be that the training of a single model for the segmentation of multiple models may be beneficial for the segmentation of some cell types, e.g., challenging cell types with less good silver truths.

This time, human detection performance could be reached for five data sets, i.e., BF-C2DL-HSC (distance method DET score: 0.991, inter-annotator agreement DET score: 0.996 ± 0.005), DIC-C2DL-HeLa (0.921, 0.965 ± 0.044), Fluo-C3DH-A549 (1.000, 1.000), Fluo-N2DL-HeLa (0.994, 0.987 ± 0.002), and PhC-C2DL-PSC (0.975, 0.983 ± 0.010). The data set Fluo-C3DH-A549 shows only a single, large 3D cell, which explains the perfect annotator agreement and the perfect DET

Table 3.16: Cell Segmentation Benchmark Results (Secondary Track of the 6th Cell Tracking Challenge Edition at the ISBI 2021). Shown are the results of per data set specialized distance method models. The segmentation models were the configuration 3 models from Table 3.15, and have been coupled with a graph-based tracking [240]. Thus, the scores can differ slightly. The corresponding Cell Segmentation Benchmark leaderboard is available at: http://celltrackingchallenge.net/files/leaderboards/CSB/2021-04-13.png (team KIT-Sch-GE (2)). State of results: April 13th, 2021.

Data Set	DET	SEG	OP_{CSB}	Ranking DET	Ranking SEG	Ranking OP_{CSB}
BF-C2DL-HSC	0.991	0.818	0.905	2nd	1st	1st
BF-C2DL-MuSC	0.979	0.777	0.878	3rd	1st	1st
DIC-C2DL-HeLa	0.921	0.778	0.850	15th	15th	14th
Fluo-C2DL-MSC	0.754	0.617	0.686	7th	6th	6th
Fluo-C3DH-A549	1.000	0.849	0.925	1st	4th	4th
Fluo-C3DH-H157	0.982	0.878	0.930	2nd	3th	2nd
Fluo-C3DL-MDA231	0.904	0.710	0.807	2nd	1st	1st
Fluo-N2DH-GOWT1	0.939	0.850	0.895	13th	22th	20th
Fluo-N2DL-HeLa	0.994	0.883	0.938	1st	11th	11th
Fluo-N3DH-CE	0.935	0.642	0.788	2nd	5th	5th
Fluo-N3DH-CHO	0.909	0.833	0.871	8th	12th	8th
PhC-C2DH-U373	0.978	0.876	0.927	13th	16th	15th
PhC-C2DL-PSC	0.975	0.743	0.859	1st	1st	1st

score many challenge participants reach on this data set. The most challenging 2D data set is Fluo-C2DL-MSC with its large irregular-shaped, often over-segmented cells. In addition to the many top-3 rankings in the Cell Segmentation Benchmark, eight top-3 rankings have been achieved in the overall performance measure of the Cell Tracking Benchmark.

Merging Post-Processing and Fluo-C2DL-Huh7 Submission

The merging post-processing reduced the number of wrongly split mouse muscle stem cells, as shown in Figure 3.19. However, the number of wrongly merged cells also increased. Due to the higher penalization of wrong merges than of false positives in the calculation of the DET metric, the OP_{CSB} score dropped slightly. To further validate the merging post-processing, the KIT-Sch-GE (2) BF-C2DL-MuSC submission has been updated using the same segmentation model and parameters as before but with the merging post-processing. Again, the performance measures dropped slightly, as Table 3.17 shows. However, since the submission was handled as an update, the rankings stayed the same and did not compete with the former KIT-Sch-GE (2) scores. When the merging post-processing is applied, a visual inspection of the segmentation results shows a temporally more stable segmentation.

Furthermore, results for the newly added data set Fluo-C2DL-Huh7 have been submitted. The submission achieved a first rank in each segmentation and tracking metric, reaching human detection quality on this data set (DET score: 0.968, inter-annotator agreement DET score: 0.947 ± 0.036).

3.5 Discussion

Instance segmentation of microscopy images has various challenges. The size and shapes of objects can vary vastly, from small roundish objects to large and elongated objects with complex structures like thin branches. Furthermore, image characteristics like SNR, contrast, and texture depend on

Table 3.17: Cell Segmentation Benchmark Results (October 22nd, 2021). The BF-C2DL-MuSC submission is an update using the same model and parameters as for the ISBI 2021 submission but with the merging post-processing. The data set Fluo-C2DL-Huh7 has been newly added to the Cell Tracking Challenge. The corresponding Cell Segmentation Benchmark leaderboard is available at: http://celltrackingchallenge.net/files/leaderboards/CSB/2021-10-22.png (team KIT-Sch-GE (2)). State of results: October 22nd, 2021.

Data Set	DET	SEG	OP_{CSB}	Ranking DET	Ranking SEG	Ranking OP_{CSB}
BF-C2DL-MuSC	0.978	0.774	0.876	3rd	1st	1st
Fluo-C2DL-Huh7	0.968	0.791	0.879	1st	1st	1st

the object type, the microscopy technique used, and the imaging settings. However, the method evaluation shows that the distance method produces high-quality results for the four evaluated object types acquired with three different microscopy techniques without using per cell type or microscopy technique specialized models. In addition, the pre-processing and post-processing were the same for all data. The distance method could especially outperform the regression method DIST and the semantic methods adapted border, boundary, and J_4 in terms of the overall performance metric OP_{CSB} for the brightfield microscopy data sets BF-C2DL-HSC and BF-C2DL-MuSC. For the other two data sets, the compared methods, besides the DIST method for the PhC-C2DL-PSC data set, produce similar results, and the differences between single models are small. These results indicate a better generalization of the distance method, as only 423 mouse hematopoietic stem cells and 335 mouse muscle stem cells are in the CTC training data set. In contrast, the training data set includes 1594 HeLa cells and 4619 pancreatic stem cells. Thus, it may be that the compared methods' results for BF-C2DL-HSC and BF-C2DL-MuSC converge for larger numbers of mouse cells in the training data set. Furthermore, the method selection may not be crucial for large training data sets as long as no method with a bottleneck is chosen.

An advantage of the distance method not used in the evaluation is tuning the post-processing thresholds t_1 and t_2. The threshold t_1 adjusts the object size, and the threshold t_2 the seed extraction. The semantic methods do not allow adjustment of the object size since the used sigmoid or softmax activation functions prevent fine-tuning. However, this thesis used fixed post-processing parameters for a fair comparison. In principle, the distance method's t_1 parameter could be adjusted per object using the size information of a first segmentation with a fixed threshold. This adjustment would also overcome the distance method's disadvantage that the per-object normalization of the object distance map results in different slopes at the background to object transition, which results in larger objects requiring a smaller threshold than smaller objects. However, this normalization, combined with neighbor distance maps, is the main advantage compared to the DIST method. The DIST method requires a local maxima post-processing due to not normalizing the object distance maps. This post-processing is prone to over-segmentation and under-segmentation depending on the data set, and, therefore, the DIST method was not competitive in the evaluation.

The goal of the distance and the adapted border methods to avoid merging of adjacent objects has been reached. Merging is only a relevant error source for the low contrast, thin, densely-packed pancreatic stem cells. Even the densely-packed mouse hematopoietic stem cells in the last frames can be detected well, although relatively few of those cells are in the CTC training data set. However, the detailed detection error analysis revealed that false positives form the largest number of all detection errors. Since false positives are the least weighted error source in the DET measure calculation, relatively high false positive rates do not decrease the DET score much if the number of true positives is high. For instance, an add/split rate of about 20 % still results in a DET score of about 0.960 (adapted border method on BF-C2DL-HSC). Thus, other weightings may

be needed, but the best weighting depends on the use case. Of course, some false positive error sources, like segmenting parts of the hydrogel well or other background structures, can be filtered in the post-processing, but at best, almost no filtering is required. Further, the weighting used in the Cell Tracking Challenge resulted in a slight DET score decrease when using the merging post-processing of the distance method, although the total number of errors could be decreased. After this proof-of-concept of the merging post-processing, the detection of split cells needs to be improved with some further constraints to avoid increasing the merge rate, which, combined with the higher weighting, offsets the decrease in false positives.

The SEG score analysis does not offer many new insights since, for some data sets, the detection quality limited the SEG scores. For similar detection quality, like for the Fluo-N2DL-HeLa data set, the SEG scores of the methods were close. An interesting exception is the BF-C2DL-MuSC data set. The distance method provides, by far, the best detection, but the J_4 method has the highest SEG score. Therefore, the distance method must have some drawbacks in correctly predicting the mouse muscle stem cell size or shape. Maybe the small thin branches are sometimes missing. Another problem is that SEG scores for different data sets are difficult to compare since higher Jaccard indices and, therefore, higher SEG scores are easier to obtain for larger objects since single-pixel errors have less influence in this case. This behavior should explain the higher SEG scores for the data set Fluo-N2DL-HeLa compared to BF-C2DL-HSC with its smaller cells.

A big issue when comparing deep learning methods is that each segmentation approach may have its own best set of training hyperparameters, e.g., optimizer or learning rate. A grid search is very time-consuming and often infeasible. Therefore, this thesis compared the results of two optimizers with their own training parameters. In addition, using learning rate schedulers should have minimized the influence of the start learning rate. The results of the Ranger optimizer and the Adam optimizer in the validation section are consistent in many cases. Thus, it is unlikely that the method ranking is due to the training process design. Overall, the Ranger optimizer produces slightly better results but can also drop the performance, e.g., for the mouse hematopoietic stem cells and the distance method. The reasons have yet to be discovered. Therefore, using the robust Adam optimizer as the default optimizer is a good choice. The Ranger optimizer is a powerful alternative, but not for all object types.

Finally, the distance method has been further validated with multiple submissions to the Cell Tracking Challenge. The corresponding leaderboards and results prove that the distance method is a state-of-the-art method that handles many imaging modalities and cell morphologies well. Furthermore, the primary track results reveal that this approach can produce reasonable segmentations when using gold truth annotations or silver truth annotations and a single model for multiple cell types or per cell type specialized models. The primary track results could probably be improved by not limiting the number of silver truths in the training data sets, training more models, and testing more post-processing parameter sets. However, these decisions were made to reduce the model training and evaluation time since 42 models needed to be selected and submitted within the competition deadline. Further, the distance method reaches human detection performance on six Cell Tracking Challenge data sets. Still, vast and irregular cells are a problem. Such cells may be challenging to segment due to the limited receptive field of the used U-Net, which hinders learning distance maps for those cells well. In addition, the merging post-processing needs to be improved for those data sets.

The new instance segmentation tool microbeSEG uses the distance method as the default method and requires no user interactions for post-processing adjustments since the normalized object distances and neighbor distances enable using a single post-processing parameter set for segmenting

many object shapes. Similar to BeadNet, microbeSEG covers the whole pipeline from training data creation, training, evaluation, and inference. The jointly developed annotation tool ObiWan-Microbi is used for the training data annotation. Both tools work together seamlessly and use OMERO for data management. The workflow analysis shows that the microbeSEG workflow allows for producing good segmentation results in a reasonable time, i.e., in about one hour with model training and without using any formerly annotated data. The distance method outperforms the boundary method when both are trained with these relatively small training data sets. The boundary method may need larger training data sets to perform well due to its missing robustness to training data variability. Furthermore, the results indicate that switching from manual annotations to pre-label corrections is a good strategy once suitable models are available. In particular, more cells can be annotated per time when correcting mainly object-level errors, i.e., merges, splits, and added or missing cells. The AJI+ scores for *B. subtilis* are primarily limited by different decision boundaries of the microbeSEG user and the test data set annotators in this experiment, especially for the cell division events of the challenging filamentous *B. subtilis* cells and for the *E. coli* size. This shows another issue when comparing deep learning methods: the methods can learn to reproduce the decision boundary of a single training data set annotator. Thus, it can happen for small training data sets that the training data set annotator is evaluated and not the methods themselves. Best, more than one person and the same persons annotate the training and the test data for method comparisons.

Once a large and diverse annotated data set is established, only significant changes in the experimental setup or specific cases where the segmentation still fails require further annotation. The microbeSEG crop selection with pre-labeling allows adding exactly those new training data where segmentation errors occur. A drawback of using crops is that many proposed crops show no cells for large images, a small selected crop size, and low object density. This behavior can, for instance, occur in a 2048×2048 px image with less than ten cells and a small crop size of 128×128 px. In such a case, a larger crop size needs to be selected. Another limitation of microbeSEG is that the segmentation method does not support the segmentation of overlapping objects. In summary, microbeSEG covers many features and functionalities needed for efficient instance segmentation, and the qualitative results show that microbeSEG can be applied to many different applications.

4 Conclusion and Outlook

4.1 Conclusion

Reducing the time to insight for researchers working with microscopy image data requires highly automated, versatile, accurate, easy-to-use, and reliable particle detection and instance segmentation methods. In particular, such methods should perform well for different imaging conditions, microscopy techniques, object shapes, and applications without requiring expert knowledge for domain adaptation or large training data sets. Thus, this thesis focused on developing new deep learning approaches and tools to leverage supervised deep learning for efficient particle detection and instance segmentation in microscopy images.

Chapter 2 presented a new particle detection approach using a particle size-dependent upsampling pre-processing and a U-Net for the semantic segmentation of particle markers. After validating the upsampling pre-processing with synthetically created data, the corresponding particle detection tool BeadNet has been introduced. Furthermore, the BeadNet particle detection has been compared with traditional particle detection methods on a real-world fluorescent bead data set. Chapter 3 presented two novel instance segmentation methods using a double-decoder U-Net, i.e., the adapted border method and the distance method. The methods have been validated for four different cell types and three microscopy imaging techniques. In addition, the superior distance method proved its potential in the Cell Tracking Challenge. Therefore, the novel instance segmentation tool microbeSEG uses the distance method. Summarized, the main contributions of this thesis are:

- A new particle detection method consisting of an upsampling pre-processing and the semantic segmentation of particle markers with a U-Net. The method has been compared with traditional methods on a real-world fluorescent latex bead data set and outperforms those on the bead data.
- The evaluation of the upsampling pre-processing for low-resolution particle detection on synthetically created data sets with different particle sizes. This evaluation compared three marker representations and showed that 1×1 markers are inferior to dilated 1×1 and 3×3 markers and that upsampling pre-processing is needed for particles with a diameter smaller than 7 px.
- The development of the open-source particle detection tool BeadNet, which uses OMERO for data management. BeadNet offers training data creation, model training, model evaluation, and model application. Furthermore, BeadNet comes with the bead data set.
- The validation of the BeadNet workflow showing that good detection results can be obtained on the synthetically created data within half an hour on a system with an NVIDIA TITAN RTX GPU. The time includes the manual annotation and the model training.
- Two novel instance segmentation methods, i.e., the adapted border method and the distance method. The adapted border method is a semantic method and predicts a two-class foreground-

background data representation and a three-class representation with the classes eroded object interior, dilated borders, and background. The distance method is a regression method and predicts normalized object distance maps and neighbor distance maps. Both methods have been compared with state-of-the-art methods on four different data sets. The distance method provided the most consistent results on those data sets. Furthermore, it could outperform the other methods on data sets with object types under-represented in the training data set.

- A detailed detection error analysis for the compared instance segmentation methods. The analysis indicated that the DET measure might underrate false positives and that the segmentation of background structures not included or under-represented in the training data is a problem without filtering post-processing.
- A proof-of-concept of the merging post-processing that can overcome the over-segmentation problem for specific object shapes. However, the reduction of splits can increase merge errors.
- Multiple top-3 rankings in the Cell Segmentation Benchmark of the Cell Tracking Challenge with the distance method reaching human detection accuracy on six data sets, i.e., one phase contrast microscopy, three fluorescence microscopy, one bright-field microscopy, and one differential interference contrast microscopy data set. Furthermore, good results could be reached when using a single model to process multiple cell types or specialized models.
- The development of the open-source instance segmentation tool microbeSEG that uses OMERO for data management and covers training data creation with pre-labeling, model training, model evaluation, and model application. Furthermore, microbeSEG models trained on cell data are provided to reduce manual annotation time for life science researchers.
- The validation of the microbeSEG workflow showing that good segmentation results can be obtained for microbe data in about an hour on a system with two NVIDIA TITAN RTX GPUs. The time includes the manual annotation and the model training. In addition, it was shown how much pre-labeling increases the number of cells annotated in a specific time.

When to use particle detection or instance segmentation depends on the information to be extracted from the images. The workflow analyses of BeadNet and microbeSEG showed that particle detection training data could be much faster annotated than segmentation training data. Thus, particle detection can be more efficient than instance segmentation if the size and shape of the objects are *a priori* known or not required for subsequent analysis. However, instance segmentation features more information for subsequent analysis or tasks like object classification and object tracking and, therefore, is more powerful overall.

In summary, this dissertation disproves the often-raised concern about deep learning that extensive and time-consuming annotation of training data is required for successful particle detection and instance segmentation. BeadNet and microbeSEG cover many features and functionalities needed for efficient particle detection and instance segmentation. Good results can be obtained within reasonable times, even without using any available annotated data sets or pre-labeling right from the beginning. The qualitative applications of BeadNet and microbeSEG show that these tools enable the processing of many different microscopy image data.

4.2 Outlook

A possible future direction for the development of BeadNet and microbeSEG is to study how much the U-Net size can be reduced without losing accuracy. Reducing the CNN parameters results in less

memory demand, enables faster training and inference, and thus improves the user experience. In addition, pruning and mixed precision training are interesting techniques but need to be evaluated for particle detection and instance segmentation. Furthermore, annotator variability inspection [69], [241], weakly supervised learning [242], [243], and self-supervised learning [244] strategies are particularly interesting. Such strategies can further accelerate and improve the annotation process.

An idea to improve the distance method is to clip instead of normalize object distance map values. Clipping would result in slopes independent from the object size, while simple threshold-based post-processing can still be applied. In addition, CNNs have a limited receptive field. Thus, it may be easier to learn the clipped value than the distance to the background for objects larger than the receptive field. Furthermore, other transformations than a distance transform may be easier to learn for a CNN.

Of course, a native 3D adaptation of the particle and instance segmentation methods is required. However, annotating data in 3D is much more time-consuming and challenging than in 2D. Thus, novel concepts are needed for an efficient 3D workflow. Such a workflow may need to include automated synthetic training data generation, e.g., with conditional generative adversarial networks [245], denoising diffusion probabilistic models [246], or hybrid approaches combining conventional simulation methods with deep learning [247]. However, so far, those approaches need expert knowledge for training and application, and it is challenging to integrate them into easy-to-use tools.

A big issue when comparing deep learning approaches is that each data representation may have its best architecture, optimizer, learning rate, augmentation set, loss, and its best initialization, among others. Furthermore, the best settings and parameters can be application- and data-set-specific. A grid search is, in many cases, infeasible due to the training times. Thus, it is challenging to draw general conclusions from method comparisons. Image processing competitions avoid this problem by letting the participants tune their solutions. However, this approach makes it impossible to conclude if a method won the competition due to a superior data representation, a good training process, the architecture, some custom augmentations that close domain gaps to the test data, or the use of external data, which is sometimes allowed. For instance, the post-challenge analysis of the CoNIC Challenge revealed that the training-validation split and ensembling techniques partially have more influence than the method selection [233]. Thus, the guidelines for competitions in [248] need, first of all, to be applied and be extended for deep learning methods. The method and tool development may slow down without such guidelines.

One idea to address the result analysis issue for deep learning methods is that competitions should have multiple separate tasks with specified settings like:

- a fixed training-validation split and augmentation set and no ensembling,
- a fixed architecture to compare data representations,
- a fixed data representation to compare architectures,
- using a small or imbalanced training data subset to study the data amount dependency,
- no use of annotated data to compare unsupervised methods,
- a completely free task to indicate the maximum possible segmentation performance.

Of course, the effort of challenge organizers, data annotators, and participants will increase when considering all these recommendations. However, organizers may select only a subset of appropriate tasks when they have a clear goal for their competition. Finally, more meta-analyses, like in [249], are needed.

A Statistical Significance

Deep neural networks display non-convex loss surfaces, and their performance highly depends on hyperparameters and other stochastic factors, making comparisons between methods difficult. Furthermore, the assumption of the t-test that the test data set scores are drawn from normal distributions may not be valid, and tests like the Mann-Whitney U test may rely on an unrealistic stochastic order [250]. Thus, this thesis uses almost stochastic order (ASO) for testing statistical significance [250]–[252]. ASO is a statistical significance test with minimal assumptions on the distribution from which the performance scores are drawn and quantifies the gap between two distributions and the extent to which stochastic order is being violated [250]. The null hypothesis is rejected with a confidence $1 - \alpha$ when the minimal distance ϵ_{min} is smaller than a threshold τ. In [251], ASO is, amongst others, compared with the Student's t-test, the Wilcoxon signed-rank test, and the Mann-Whitney U test, and following [251], this thesis uses the threshold $\tau = 0.2$ instead of $\tau = 0.5$. Table A.1 to Table A.24 report the statistical significance of the evaluation results from Section 3.2.

Table A.1: Significance of the DET Measure Results for BF-C2DL-HSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-HSC DET results with the Adam optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	1.0	0.988	0.755
Boundary	1.0	-	1.0	0.993	0.517
DIST	1.0	0.881	-	0.992	0.487
Distance	**0.055**	**0.018**	**0.029**	-	**0.0**
J_4	1.0	1.0	1.0	1.0	-

Table A.2: Significance of the DET Measure Results for BF-C2DL-HSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-HSC DET results with the Ranger optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.838	1.0	0.819	0.322
Boundary	1.0	-	1.0	0.869	0.405
DIST	0.538	0.470	-	0.648	**0.185**
Distance	1.0	1.0	1.0	-	0.559
J_4	1.0	1.0	1.0	1.0	-

Table A.3: Significance of the DET Measure Results for BF-C2DL-MuSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-MuSC DET results with the Adam optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	**0.148**	1.0	0.996	**0.004**
Boundary	1.0	-	1.0	0.993	**0.129**
DIST	0.557	**0.078**	-	0.9928	**0.001**
Distance	**0.010**	**0.016**	**0.018**	-	**0.0**
J_4	1.0	1.0	1.0	1.0	-

Table A.4: Significance of the DET Measure Results for BF-C2DL-MuSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-MuSC DET results with the Ranger optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	1.0	0.999	1.0
Boundary	**0.008**	-	**0.112**	1.0	**0.002**
DIST	0.232	1.0	-	0.997	0.499
Distance	**0.007**	**0.011**	**0.016**	-	**0.0**
J_4	0.634	1.0	1.0	1.0	-

Table A.5: Significance of the DET Measure Results for Fluo-N2DL-HeLa (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the Fluo-N2DL-HeLa DET results with the Adam optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.282	0.996	0.856	1.0
Boundary	1.0	-	0.983	1.0	0.981
DIST	**0.031**	**0.059**	-	0.267	**0.017**
Distance	1.0	0.563	1.0	-	1.0
J_4	0.207	**0.068**	1.0	0.693	-

Table A.6: Significance of the DET Measure Results for Fluo-N2DL-HeLa (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the Fluo-N2DL-HeLa DET results with the Ranger optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	1.0	1.0	1.0
Boundary	0.699	-	1.0	1.0	1.0
DIST	0.439	0.537	-	1.0	1.0
Distance	**0.024**	**0.007**	**0.042**	-	**0.091**
J_4	**0.114**	**0.068**	0.371	1.0	-

Table A.7: Significance of the DET Measure Results for PhC-C2DL-PSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the PhC-C2DL-PSC DET results with the Adam optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.998	**0.0**	1.0	1.0
Boundary	**0.021**	-	**0.0**	0.712	0.528
DIST	1.0	1.0	-	0.995	0.996
Distance	**0.025**	1.0	**0.010**	-	0.685
J_4	**0.092**	1.0	**0.009**	1.0	-

Table A.8: Significance of the DET Measure Results for PhC-C2DL-PSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the PhC-C2DL-PSC DET results with the Ranger optimizer shown in Figure 3.13. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.370	**0.0**	0.872	0.367
Boundary	1.0	-	**0.0**	1.0	1.0
DIST	1.0	1.0	-	1.0	1.0
Distance	1.0	0.577	**0.006**	-	0.512
J_4	1.0	1.0	**0.006**	1.0	-

Table A.9: Significance of the SEG Measure Results for BF-C2DL-HSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-HSC SEG results with the Adam optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.881	**0.007**	1.0	1.0
Boundary	1.0	-	**0.001**	0.99	1.0
DIST	1.0	1.0	-	0.995	0.998
Distance	**0.033**	**0.023**	**0.008**	-	**0.001**
J_4	0.462	0.215	**0.006**	1.0	-

Table A.10: Significance of the SEG Measure Results for BF-C2DL-HSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-HSC SEG results with the Ranger optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	**0.189**	0.354	1.0	0.610
Boundary	1.0	-	1.0	1.0	1.0
DIST	1.0	0.626	-	1.0	1.0
Distance	**0.182**	**0.006**	**0.047**	-	**0.080**
J_4	1.0	0.457	0.662	1.0	-

Table A.11: Significance of the SEG Measure Results for BF-C2DL-MuSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-MuSC SEG results with the Adam optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	**0.001**	1.0	0.997
Boundary	**0.063**	-	**0.0**	1.0	1.0
DIST	1.0	1.0	-	1.0	0.995
Distance	**0.021**	0.571	**0.007**	-	1.0
J_4	**0.009**	**0.153**	**0.009**	0.244	-

Table A.12: Significance of the SEG Measure Results for BF-C2DL-MuSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-MuSC SEG results with the Ranger optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	**0.0**	1.0	1.0
Boundary	**0.026**	-	**0.0**	**0.138**	0.545
DIST	1.0	1.0	-	0.999	0.994
Distance	0.336	1.0	**0.007**	-	1.0
J_4	**0.085**	1.0	**0.010**	0.301	-

Table A.13: Significance of the SEG Measure Results for Fluo-N2DL-HeLa (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the Fluo-N2DL-HeLa SEG results with the Adam optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	**0.0**	**0.0**	**0.027**	**0.003**
Boundary	1.0	-	1.0	0.998	0.994
DIST	1.0	**0.065**	-	1.0	1.0
Distance	1.0	**0.023**	**0.060**	-	1.0
J_4	1.0	**0.029**	**0.044**	0.994	-

Table A.14: Significance of the SEG Measure Results for Fluo-N2DL-HeLa (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the Fluo-N2DL-HeLa SEG results with the Ranger optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	**0.014**	**0.0**	1.0	0.928
Boundary	1.0	-	**0.004**	0.997	1.0
DIST	1.0	1.0	-	0.997	0.999
Distance	**0.037**	**0.011**	**0.006**	-	**0.029**
J_4	1.0	**0.085**	**0.005**	1.0	-

Table A.15: Significance of the SEG Measure Results for PhC-C2DL-PSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the PhC-C2DL-PSC SEG results with the Adam optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	**0.0**	1.0	1.0
Boundary	**0.127**	-	**0.0**	1.0	0.389
DIST	1.0	1.0	-	0.993	0.994
Distance	**0.032**	0.701	**0.010**	-	0.237
J_4	0.634	1.0	**0.010**	1.0	-

Table A.16: Significance of the SEG Measure Results for PhC-C2DL-PSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the PhC-C2DL-PSC SEG results with the Ranger optimizer shown in Figure 3.20. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.899	**0.0**	1.0	0.575
Boundary	1.0	-	**0.0**	1.0	1.0
DIST	1.0	1.0	-	0.994	0.994
Distance	0.935	0.912	**0.008**	-	0.487
J_4	1.0	0.985	**0.008**	1.0	-

Table A.17: Significance of the OP_{CSB} Measure Results for BF-C2DL-HSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-HSC OP_{CSB} results with the Adam optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.789	**0.018**	0.994	1.0
Boundary	1.0	-	**0.001**	0.993	1.0
DIST	1.0	1.0	-	0.994	0.996
Distance	**0.025**	**0.016**	**0.009**	-	**0.0**
J_4	0.660	0.320	**0.011**	1.0	-

Table A.18: Significance of the OP_{CSB} Measure Results for BF-C2DL-HSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-HSC OP_{CSB} results with the Ranger optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{\text{min}} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.206	0.517	1.0	0.409
Boundary	1.0	-	1.0	1.0	1.0
DIST	1.0	0.493	-	1.0	0.679
Distance	0.544	**0.161**	0.283	-	0.249
J_4	1.0	1.0	1.0	1.0	-

Table A.19: Significance of the OP_{CSB} Measure Results for BF-C2DL-MuSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-MuSC OP_{CSB} results with the Adam optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{min} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	0.386	1.0	0.452
Boundary	1.0	-	0.285	1.0	0.419
DIST	1.0	1.0	-	0.999	0.761
Distance	**0.008**	**0.008**	**0.009**	-	**0.0**
J_4	1.0	1.0	1.0	1.0	-

Table A.20: Significance of the OP_{CSB} Measure Results for BF-C2DL-MuSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the BF-C2DL-MuSC OP_{CSB} results with the Ranger optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{min} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.997	0.615	0.995	1.0
Boundary	**0.011**	-	**0.0**	1.0	**0.0**
DIST	1.0	1.0	-	0.993	1.0
Distance	**0.010**	**0.023**	**0.014**	-	**0.0**
J_4	0.286	1.0	**0.112**	1.0	-

Table A.21: Significance of the OP_{CSB} Measure Results for Fluo-N2DL-HeLa (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the Fluo-N2DL-HeLa OP_{CSB} results with the Adam optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{min} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	**0.0**	**0.0**	0.303	**0.155**
Boundary	1.0	-	0.990	1.0	0.991
DIST	1.0	**0.052**	-	1.0	1.0
Distance	1.0	**0.074**	0.745	-	1.0
J_4	1.0	**0.040**	**0.058**	0.786	-

Table A.22: Significance of the OP_{CSB} Measure Results for Fluo-N2DL-HeLa (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the Fluo-N2DL-HeLa OP_{CSB} results with the Ranger optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{min} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	**0.077**	**0.0**	1.0	1.0
Boundary	1.0	-	**0.025**	0.997	1.0
DIST	1.0	1.0	-	0.997	0.999
Distance	**0.006**	**0.012**	**0.006**	-	**0.027**
J_4	0.418	**0.034**	**0.006**	1.0	-

Table A.23: Significance of the OP_{CSB} Measure Results for PhC-C2DL-PSC (Adam). Stated are the minimal distances ϵ_{min} of the ASO test for the PhC-C2DL-PSC OP_{CSB} results with the Adam optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{min} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	1.0	**0.0**	1.0	1.0
Boundary	**0.024**	-	**0.0**	0.916	0.382
DIST	1.0	1.0	-	0.994	0.994
Distance	**0.023**	1.0	**0.010**	-	0.485
J_4	**0.143**	1.0	**0.010**	1.0	-

Table A.24: Significance of the OP_{CSB} Measure Results for PhC-C2DL-PSC (Ranger). Stated are the minimal distances ϵ_{min} of the ASO test for the PhC-C2DL-PSC OP_{CSB} results with the Ranger optimizer shown in Figure 3.21. Bold distance scores ϵ_{min} indicate that a method is significantly better than the compared baseline method ($\epsilon_{min} < 0.2$, $\alpha = 0.05$).

Method \ Baseline	Adapted Border	Boundary	DIST	Distance	J_4
Adapted Border	-	0.682	**0.0**	0.941	0.388
Boundary	1.0	-	**0.0**	1.0	1.0
DIST	1.0	1.0	-	0.995	0.996
Distance	1.0	0.765	**0.007**	-	0.372
J_4	1.0	0.978	**0.007**	1.0	-

microbeSEG Overview

B

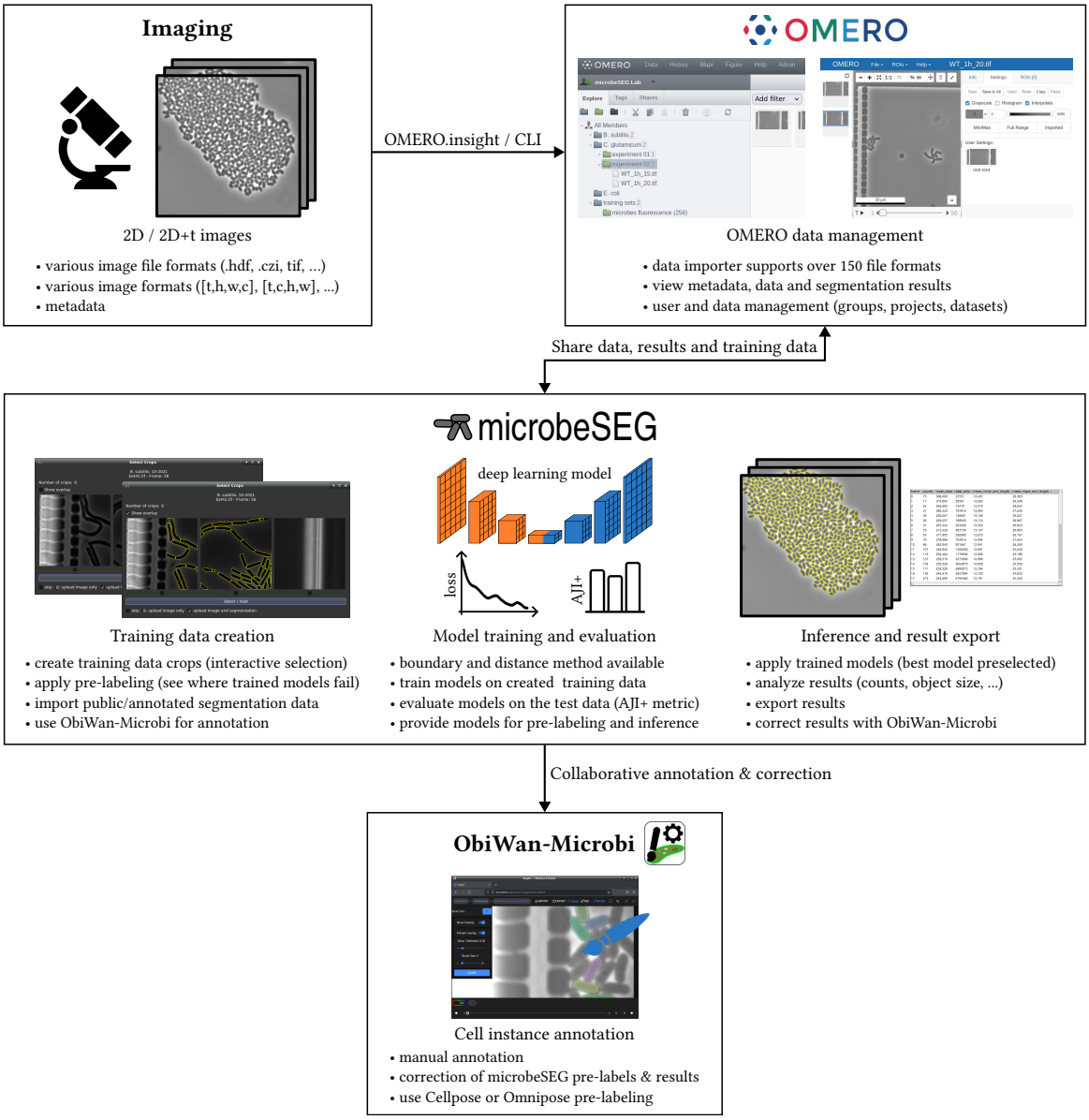


Figure B.1: microbeSEG Overview. See https://github.com/hip-satomi/microbeSEG for the microbeSEG manual with illustrations and videos. Modified from [214].

Abbreviations

Abbreviation	Description
AJI	Aggregated Jaccard index.
AJI+	AJI version that prevents overpenalization.
BF	Bright-field microscopy.
C	Channel or feature map number / channel dimension.
CCL	Connected-component labeling.
CNN	Convolutional neural network.
CSB	Cell Segmentation Benchmark of the Cell Tracking Challenge.
CTB	Cell Tracking Benchmark of the Cell Tracking Challenge.
CTC	Cell Tracking Challenge.
DET	Normalized acyclic oriented graph matching measure for detection.
DIC	Differential interference contrast microscopy.
EM	Embedding method (pixels of the same object should have similar values).
Fluo	Fluorescence microscopy.
H	Height / height dimension of an image or feature map.
N	Batch size / batch dimension.
OP_{CSB}	Overall performance measure of the CSB.
PhC	Phase contrast microscopy.
PM	Proposal-based method, e.g., prediction of bounding boxes or polygon distances.
RM	Regression method, e.g., prediction of distance maps.
R-CNN	Region-based CNN [42], [104], [253].
SEG	Segmentation accuracy measure based on the Jaccard similarity index.
SM	Semantic method, e.g., prediction of foreground, background, and boundaries.
SNR	Signal-to-noise ratio.
W	Width / width dimension of an image or feature map.

List of Own Publications

This list shows all my journal and conference articles, preprints, data sets, and trained deep learning models produced and published while working on this thesis. The items are ordered by their appearance in this thesis.

[30] T. Scherr, K. Streule, A. Bartschat, *et al.*, "BeadNet: Deep learning-based bead detection and counting in low-resolution microscopy images", *Bioinformatics*, vol. 36, no. 17, pp. 4668–4670, 2020. DOI: 10.1093/bioinformatics/btaa594.

[31] T. Scherr, J. Seiffarth, B. Wollenhaupt, *et al.*, "microbeSEG dataset", version 1.0, *Zenodo*, 2022. DOI: 10.5281/zenodo.6497715.

[69] M. P. Schilling, T. Scherr, F. R. Münke, *et al.*, "Automated annotator variability inspection for biomedical image segmentation", *IEEE Access*, vol. 10, pp. 2753–2765, 2022. DOI: 10.1109/ACCESS.2022.3140378.

[150] T. Scherr, A. Bartschat, M. Reischl, J. Stegmaier, and R. Mikut, "Best practices in deep learning-based segmentation of microscopy images", in *Proceedings 28. Workshop Computational Intelligence*, Dortmund, Germany: KIT Scientific Publishing, 2018, pp. 175–195. DOI: 10.5445/IR/1000087734.

[167] T. Scherr, K. Löffler, M. Böhland, and R. Mikut, "Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy", *PLOS ONE*, vol. 15, no. 12, art. e0243219, 2020. DOI: 10.1371/journal.pone.0243219.

[168] T. Scherr, K. Löffler, O. Neumann, and R. Mikut, "On improving an already competitive segmentation algorithm for the Cell Tracking Challenge - lessons learned", *bioRxiv*, 2021, preprint, version v1. DOI: 10.1101/2021.06.26.450019.

[214] T. Scherr, J. Seiffarth, B. Wollenhaupt, *et al.*, "microbeSEG: A deep learning software tool with OMERO data management for efficient and accurate cell segmentation", *PLOS ONE*, vol. 17, no. 11, art. e0277601, 2022. DOI: 10.1371/journal.pone.0277601.

[217] T. Scherr, J. Seiffarth, B. Wollenhaupt, *et al.*, "microbeSEG models", version 1.0, *Zenodo*, 2022. DOI: 10.5281/zenodo.7221152.

[218] J. Seiffarth, T. Scherr, B. Wollenhaupt, *et al.*, "ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud", *bioRxiv*, 2022, preprint, version v1. DOI: 10.1101/2022.08.01.502297.

[232] M. Böhland, O. Neumann, M. P. Schilling, *et al.*, "Ciscnet - a single-branch cell nucleus instance segmentation and classification network", in *2022 IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*, Kolkata, India: IEEE, 2022, pp. 1–5. DOI: 10.1109/ISBIC56247.2022.9854734.

[233] S. Graham, Q. D. Vu, M. Jahanifar, *et al.*, "CoNIC challenge: Pushing the frontiers of nuclear detection, segmentation, classification and counting", *arXiv*, 2023, preprint, version v2. DOI: 10.48550/ARXIV.2303.06274.

[236] M. Maška, V. Ulman, P. Delgado-Rodriguez, *et al.*, "The Cell Tracking Challenge: 10 years of objective benchmarking", *Nature Methods*, 2023. DOI: 10.1038/s41592-023-01879-y.

[240] K. Löffler, T. Scherr, and R. Mikut, "A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction", *PLOS ONE*, vol. 16, no. 9, art. e0249257, 2021. DOI: 10.1371/journal.pone.0249257.

[241] M. P. Schilling, N. Ahuja, L. Rettenberger, T. Scherr, and M. Reischl, "Impact of annotation noise on histopathology nucleus segmentation", *Current Directions in Biomedical Engineering*, vol. 8, no. 2, pp. 197–200, 2022. DOI: 10.1515/cdbme-2022-1051.

[249] M. Eisenmann, A. Reinke, V. Weru, *et al.*, "Biomedical image analysis competitions: The state of current participation practice", *arXiv*, 2022, preprint, version v1. DOI: 10.48550/arXiv.2212.08568.

[254] A. Bartschat, T. Unger, T. Scherr, J. Stegmaier, R. Mikut, and M. Reischl, "Robustness of deep learning architectures with respect to training data variation", in *Proceedings 28. Workshop Computational Intelligence*, Dortmund, Germany: KIT Scientific Publishing, 2018, pp. 129–138. DOI: 10.5445/IR/1000087724.

[255] A. Bartschat, S. Allgeier, T. Scherr, *et al.*, "Fuzzy tissue detection for real-time focal control in corneal confocal microscopy", *at - Automatisierungstechnik*, vol. 67, no. 10, pp. 879–888, 2019. DOI: 10.1515/auto-2019-0034.

[256] A. Bartschat, S. Allgeier, S. Bohn, *et al.*, "Digitale Bildverarbeitung und tiefe neuronale Netze in der Augenheilkunde - aktuelle Trends", *Klinische Monatsblätter für Augenheilkunde*, vol. 236, no. 12, pp. 1399–1406, 2019. DOI: 10.1055/a-1008-9400.

[257] M. Böhland, T. Scherr, A. Bartschat, R. Mikut, and M. Reischl, "Influence of synthetic label image object properties on GAN supported segmentation pipelines", in *Proceedings 29. Workshop Computational Intelligence*, Dortmund, Germany: KIT Scientific Publishing, 2019, pp. 289–309. DOI: 10.5445/IR/1000100253.

[258] M. Takamiya, J. Stegmaier, A. Y. Kobitski, *et al.*, "Pax6 organizes the anterior eye segment by guiding two distinct neural crest waves", *PLOS Genetics*, vol. 16, no. 6, art. e1008774, 2020. DOI: 10.1371/journal.pgen.1008774.

[259] S. Sheshachala, M. Grösche, T. Scherr, *et al.*, "Segregation of dispersed silica nanoparticles in microfluidic water-in-oil droplets: A kinetic study", *ChemPhysChem*, vol. 21, no. 10, pp. 1070–1078, 2020. DOI: 10.1002/cphc.201901151.

[260] M. Böhland, L. Tharun, T. Scherr, *et al.*, "Machine learning methods for automated classification of tumors with papillary thyroid carcinoma-like nuclei: A quantitative analysis", *PLOS ONE*, vol. 16, no. 9, art. e0257635, 2021. DOI: 10.1371/journal.pone.0257635.

[261] M. P. Schilling, O. Neumann, T. Scherr, *et al.*, "A computational workflow for interdisciplinary deep learning projects utilizing bwHPC infrastructure", in *Proceedings of the 7th bwHPC Symposium*, Ulm, Germany: Ulm University, Ulm, Germany, 2022, pp. 69–74. DOI: 10.18725/OPARU-46069.

References

[1] A. Grünberger, W. Wiechert, and D. Kohlheyer, "Single-cell microfluidics: Opportunity for bioprocess development", *Current Opinion in Biotechnology*, vol. 29, pp. 15–23, 2014. DOI: 10.1016/j.copbio.2014.02.008.

[2] J. Hemmerich, S. Noack, W. Wiechert, and M. Oldiges, "Microbioreactor systems for accelerated bioprocess development", *Biotechnology Journal*, vol. 13, no. 4, art. 1700141, 2018. DOI: 10.1002/biot.201700141.

[3] T.-L. Liu, S. Upadhyayula, D. E. Milkie, *et al.*, "Observing the cell in its native state: Imaging subcellular dynamics in multicellular organisms", *Science*, vol. 360, no. 6386, art. eaaq1392, 2018. DOI: 10.1126/science.aaq1392.

[4] C. A. Casacio, L. S. Madsen, A. Terrasson, *et al.*, "Quantum-enhanced nonlinear microscopy", *Nature*, vol. 594, no. 7862, pp. 201–206, 2021. DOI: 10.1038/s41586-021-03528-w.

[5] J. Na, G. Kim, S.-H. Kang, S.-J. Kim, and S. Lee, "Deep learning-based discriminative refocusing of scanning electron microscopy images for materials science", *Acta Materialia*, vol. 214, art. 116987, 2021. DOI: 10.1016/j.actamat.2021.116987.

[6] S. R. Spurgeon, C. Ophus, L. Jones, *et al.*, "Towards data-driven next-generation transmission electron microscopy", *Nature Materials*, vol. 20, no. 3, pp. 274–279, 2021. DOI: 10.1038/s41563-020-00833-z.

[7] B. Shen, S. Liu, Y. Li, *et al.*, "Deep learning autofluorescence-harmonic microscopy", *Light: Science & Applications*, vol. 11, no. 1, art. 76, 2022. DOI: 10.1038/s41377-022-00768-x.

[8] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet large scale visual recognition challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[9] G. Litjens, T. Kooi, B. E. Bejnordi, *et al.*, "A survey on deep learning in medical image analysis", *Medical Image Analysis*, vol. 42, pp. 60–88, 2017. DOI: 10.1016/j.media.2017.07.005.

[10] F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, "Deep learning in microscopy image analysis: A survey", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4550–4568, 2018. DOI: 10.1109/TNNLS.2017.2766168.

[11] C. Belthangady and L. A. Royer, "Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction", *Nature Methods*, vol. 16, no. 12, pp. 1215–1225, 2019. DOI: 10.1038/s41592-019-0458-z.

[12] P. M. Bhatt, R. K. Malhan, P. Rajendran, *et al.*, "Image-based surface defect detection using deep learning: A review", *Journal of Computing and Information Science in Engineering*, vol. 21, no. 4, art. 040801, 2021. DOI: 10.1115/1.4049535.

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset", *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013. DOI: 10.1177/0278364913491297.

[14] M. Cordts, M. Omran, S. Ramos, *et al.*, "The Cityscapes dataset for semantic urban scene understanding", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350.

[15] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "NuScenes: A multimodal dataset for autonomous driving", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, 2020, pp. 11 618–11 628. DOI: 10.1109/CVPR42600.2020.01164.

[16] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: Common objects in context", in *Computer Vision – ECCV 2014*, Zurich, Switzerland: Springer, Cham, 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

[17] B. Jähne, *Digital Image Processing*, 6th ed. Springer, Berlin, Heidelberg, 2005. DOI: 10.1007/3-540-27563-0.

[18] EMVA. "EMVA Standard 1288, Standard for characterization of image sensors and cameras." version 4.0 Linear. (Jun. 2021), [Online]. Available: https://www.emva.org/wp-content/uploads/EMVA1288Linear_4.0Release.pdf (visited on 07/19/2022).

[19] M. Pluta, *Advanced light microscopy*. Elsevier, Amsterdam, 1989, vol. 2: Specialized methods, ISBN: 0444989188.

[20] P. J. Goodhew, F. J. Humphreys, and R. Beanland, *Electron microscopy and analysis*, 3rd ed. CRC Press, London, 2000. DOI: 10.1201/9781482289343.

[21] D. J. Stephens and V. J. Allan, "Light microscopy techniques for live cell imaging", *Science*, vol. 300, no. 5616, pp. 82–86, 2003. DOI: 10.1126/science.1082160.

[22] D. B. Murphy and M. W. Davidson, *Fundamentals of Light Microscopy and Electronic Imaging*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2012. DOI: 10.1002/9781118382905.

[23] H. Sahoo, "Fluorescent labeling techniques in biomolecules: A flashback", *RSC Advances*, vol. 2, no. 18, pp. 7017–7029, 2012. DOI: 10.1039/C2RA20389H.

[24] C. P. Toseland, "Fluorescent labeling and modification of proteins", *Journal of Chemical Biology*, vol. 6, no. 3, pp. 85–95, 2013. DOI: 10.1007/s12154-013-0094-5.

[25] D. P. Penney, J. M. Powers, M. Frank, C. Willis, and C. Churukian, "Analysis and testing of biological stains– the biological stain commission procedures", *Biotechnic & Histochemistry*, vol. 77, no. 5-6, pp. 237–275, 2002. DOI: 10.1080/bih.77.5-6.237.275.

[26] T. J. Beveridge, J. R. Lawrence, and R. G. E. Murray, "Sampling and staining for light microscopy", in *Methods for General and Molecular Microbiology*. ASM Press, Washington, DC, USA, 2007, ch. 2, pp. 19–33. DOI: 10.1128/9781555817497.ch2.

[27] M. Maška, V. Ulman, D. Svoboda, *et al.*, "A benchmark for comparison of cell tracking algorithms", *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, 2014. DOI: 10.1093/bioinformatics/btu080.

[28] V. Ulman, M. Maška, K. E. G. Magnusson, *et al.*, "An objective comparison of cell-tracking algorithms", *Nature Methods*, vol. 14, pp. 1141–1152, 2017. DOI: 10.1038/nmeth.4473.

[29] S. Graham, M. Jahanifar, A. Azam, *et al.*, "Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification", in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, Canada: IEEE, 2021, pp. 684–693. DOI: 10.1109/ICCVW54120.2021.00082.

[32] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation", *Nature Methods*, vol. 9, no. 7, pp. 637–637, 2012. DOI: 10.1038/nmeth.2083.

[33] Z. Liu, L. D. Lavis, and E. Betzig, "Imaging live-cell dynamics and structure at the single-molecule level", *Molecular Cell*, vol. 58, no. 4, pp. 644–659, 2015. DOI: 10.1016/j.molcel.2015.02.033.

[34] P. P. Laissue, R. A. Alghamdi, P. Tomancak, E. G. Reynaud, and H. Shroff, "Assessing phototoxicity in live fluorescence imaging", *Nature Methods*, vol. 14, no. 7, pp. 657–661, 2017. DOI: 10.1038/nmeth.4344.

[35] N. O'Mahony, S. Campbell, A. Carvalho, *et al.*, "Deep learning vs. traditional computer vision", in *Advances in Computer Vision*, vol. 943, Las Vegas, NV, USA: Springer, Cham, 2020, pp. 128–144. DOI: 10.1007/978-3-030-17795-9_10.

[36] F. Murtagh, "Multilayer perceptrons for classification and regression", *Neurocomputing*, vol. 2, no. 5, pp. 183–197, 1991. DOI: 10.1016/0925-2312(91)90023-5.

[37] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification", *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999. DOI: 10.1109/72.788646.

[38] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns", in *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil: IEEE, 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4409066.

[39] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection", *Neurocomputing*, vol. 396, pp. 39–64, 2020. DOI: 10.1016/j.neucom.2020.01.085.

[40] N. Kumar, R. Verma, D. Anand, *et al.*, "A multi-organ nucleus segmentation challenge", *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1380–1391, 2020. DOI: 10.1109/TMI.2019.2947628.

[41] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation", *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021. DOI: 10.1038/s41592-020-01008-z.

[42] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN", in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy: IEEE, 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[43] J. C. Caicedo, A. Goodman, K. W. Karhohs, *et al.*, "Nucleus segmentation across imaging experiments: The 2018 data science bowl", *Nature Methods*, vol. 16, pp. 1247–1253, 2019. DOI: 10.1038/s41592-019-0612-7.

[44] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection", *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020. DOI: 10.1007/s10845-019-01476-x.

[45] M. Weigert, U. Schmidt, T. Boothe, *et al.*, "Content-aware image restoration: Pushing the limits of fluorescence microscopy", *Nature Methods*, vol. 15, pp. 1090–1097, 2018. DOI: 10.1038/s41592-018-0216-7.

[46] J. Schmidhuber, "Deep learning in neural networks: An overview", *Neural Networks*, vol. 61, pp. 85–117, 2015. DOI: 10.1016/j.neunet.2014.09.003.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. DOI: 10.1145/3065386.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy: JMLR, 2010, pp. 249–256.

[49] J. Gu, Z. Wang, J. Kuen, *et al.*, "Recent advances in convolutional neural networks", *Pattern Recognition*, vol. 77, pp. 354–377, 2018. DOI: 10.1016/j.patcog.2017.10.013.

[50] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures", *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019. DOI: 10.1109/ACCESS.2019.2912200.

[51] L. Alzubaidi, J. Zhang, A. J. Humaidi, *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions", *Journal of Big Data*, vol. 8, no. 1, art. 53, 2021. DOI: 10.1186/s40537-021-00444-8.

[52] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539.

[53] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review", *Neurocomputing*, vol. 187, pp. 27–48, 2016. DOI: 10.1016/j.neucom.2015.09.116.

[54] E. Meijering, "A bird's-eye view of deep learning in bioimage analysis", *Computational and Structural Biotechnology Journal*, vol. 18, pp. 2312–2325, 2020. DOI: 10.1016/j.csbj.2020.08.003.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv*, 2015, preprint, version v6. DOI: 10.48550/ARXIV.1409.1556.

[57] M. Z. Alom, T. M. Taha, C. Yakopcic, *et al.*, "The history began from AlexNet: A comprehensive survey on deep learning approaches", *arXiv*, 2018, preprint, version v2. DOI: 10.48550/ARXIV.1803.01164.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, Lille, France: JMLR, 2015, pp. 448–456.

[59] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017. DOI: 10.1109/TPAMI.2016.2644615.

[60] B. Midtvedt, S. Helgadottir, A. Argun, J. Pineda, D. Midtvedt, and G. Volpe, "Quantitative digital microscopy with deep learning", *Applied Physics Reviews*, vol. 8, no. 1, art. 011310, 2021. DOI: 10.1063/5.0034891.

[61] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation", in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, Munich, Germany: Springer, Cham, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[62] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, "Deep learning for cellular image analysis", *Nature Methods*, vol. 16, no. 12, pp. 1233–1246, 2019. DOI: 10.1038/s41592-019-0403-1.

[63] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: A review of theory and applications", *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021. DOI: 10.1109/ACCESS.2021.3086020.

[64] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation", *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1856–1867, 2020. DOI: 10.1109/TMI.2019.2959609.

[65] H. Huang, L. Lin, R. Tong, *et al.*, "UNet 3+: A full-scale connected Unet for medical image segmentation", in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain: IEEE, 2020, pp. 1055–1059. DOI: 10.1109/ICASSP40776.2020.9053405.

[66] M. Z. Alom, C. Yakopcic, T. M. Taha, and V. K. Asari, "Nuclei segmentation with recurrent residual convolutional neural networks based U-Net (R2U-Net)", in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, Dayton, OH, USA: IEEE, 2018, pp. 228–233. DOI: 10.1109/NAECON.2018.8556686.

[67] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "DoubleU-Net: A deep convolutional neural network for medical image segmentation", in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, Rochester, MN, USA: IEEE, 2020, pp. 558–564. DOI: 10.1109/CBMS49503.2020.00111.

[68] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis", *Medical Image Analysis*, vol. 65, art. 101759, 2020. DOI: 10.1016/j.media.2020.101759.

[70] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of Big Data*, vol. 6, no. 1, art. 60, 2019. DOI: 10.1186/s40537-019-0197-0.

[71] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations", *Information*, vol. 11, no. 2, art. 125, 2020. DOI: 10.3390/info11020125.

[72] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A comprehensive survey of loss functions in machine learning", *Annals of Data Science*, vol. 9, no. 2, pp. 187–212, 2022. DOI: 10.1007/s40745-020-00253-5.

[73] S. Jadon, "A survey of loss functions for semantic segmentation", in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Via del Mar, Chile: IEEE, 2020, pp. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.

[74] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: 10.1038/323533a0.

[75] L. Bottou, "Stochastic gradient descent tricks", in *Neural Networks: Tricks of the Trade*, Springer, Berlin, Heidelberg, 2012, pp. 421–436. DOI: 10.1007/978-3-642-35289-8_25.

[76] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets", in *Advances in Neural Information Processing Systems*, vol. 31, Montreal, Canada: Curran Associates, Inc., 2018.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv*, 2015, preprint, version v9. DOI: 10.48550/arXiv.1412.6980.

[78] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: K steps forward, 1 step back", in *Advances in Neural Information Processing Systems*, vol. 32, Vancouver, Canada: Curran Associates, Inc., 2019.

[79] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning", in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA: JMLR, 2013, pp. 1139–1147.

[80] L. N. Smith, "Cyclical learning rates for training neural networks", in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA: IEEE, 2017, pp. 464–472. DOI: 10.1109/WACV.2017.58.

[81] N. Moshkov, B. Mathe, A. Kertesz-Farkas, R. Hollandi, and P. Horvath, "Test-time augmentation for deep learning-based cell segmentation on microscopy images", *Scientific Reports*, vol. 10, art. 5068, 2020. DOI: 10.1038/s41598-020-61808-3.

[82] M. Abdar, F. Pourpanah, S. Hussain, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges", *Information Fusion*, vol. 76, pp. 243–297, 2021. DOI: 10.1016/j.inffus.2021.05.008.

[83] N. Chenouard, I. Smal, F. De Chaumont, *et al.*, "Objective comparison of particle tracking methods", *Nature Methods*, vol. 11, no. 3, pp. 281–289, 2014. DOI: 10.1038/nmeth.2808.

[84] L. Braun, H. Ohayon, and P. Cossart, "The InlB protein of Listeria monocytogenes is sufficient to promote entry into mammalian cells", *Molecular Microbiology*, vol. 27, no. 5, pp. 1077–1087, 1998. DOI: 10.1046/j.1365-2958.1998.00750.x.

[85] D. A. Boiko, E. O. Pentsak, V. A. Cherepanova, and V. P. Ananikov, "Electron microscopy dataset for the recognition of nanoscale ordering effects and location of nanoparticles", *Scientific Data*, vol. 7, no. 1, art. 101, 2020. DOI: 10.1038/s41597-020-0439-1.

[86] T. Wollmann, C. Ritter, J. N. Dohrke, J.-Y. Lee, R. Bartenschlager, and K. Rohr, "Detnet: Deep neural network for particle detection in fluorescence microscopy images", in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy: IEEE, 2019, pp. 517–520. DOI: 10.1109/ISBI.2019.8759234.

[87] I. Smal, M. Loog, W. Niessen, and E. Meijering, "Quantitative comparison of spot detection methods in fluorescence microscopy", *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 282–301, 2010. DOI: 10.1109/TMI.2009.2025127.

[88] J. M. Newby, A. M. Schaefer, P. T. Lee, M. G. Forest, and S. K. Lai, "Convolutional neural networks automate detection for tracking of submicron-scale particles in 2d and 3d", *Proceedings of the National Academy of Sciences*, vol. 115, no. 36, pp. 9026–9031, 2018. DOI: 10.1073/pnas.1804420115.

[89] P. Ruusuvuori, T. Äijö, S. Chowdhury, *et al.*, "Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images", *BMC Bioinformatics*, vol. 11, art. 248, 2010. DOI: 10.1186/1471-2105-11-248.

[90] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, 1979. DOI: 10.1109/TSMC.1979.4310076.

[91] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of Hough transform methods for circle finding", *Image and Vision Computing*, vol. 8, pp. 71–77, 1990. DOI: 10.1016/0262-8856(90)90059-E.

[92] M. Smereka and I. Dulęba, "Circular object detection using a modified hough transform", *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 1, pp. 85–91, 2008. DOI: 10.2478/v10006-008-0008-9.

[93] J.-C. Olivo-Marin, "Extraction of spots in biological images using multiscale products", *Pattern Recognition*, vol. 35, no. 9, pp. 1989–1996, 2002. DOI: 10.1016/S0031-3203(01)00127-3.

[94] I. F. Sbalzarini and P. Koumoutsakos, "Feature point tracking and trajectory analysis for video imaging in cell biology", *Journal of Structural Biology*, vol. 151, no. 2, pp. 182–195, 2005. DOI: 10.1016/j.jsb.2005.06.002.

[95] J. Byun, M. R. Verardo, B. Sumengen, G. P. Lewis, B. S. Manjunath, and S. K. Fisher, "Automated tool for the detection of cell nuclei in digital microscopic images: Application to retinal images", *Molecular Vision*, vol. 12, no. 105-07, pp. 949–960, 2006.

[96] F. Mueller, A. Senecal, K. Tantale, *et al.*, "FISH-quant: Automatic counting of transcripts in 3D FISH images", *Nature Methods*, vol. 10, pp. 277–278, 2013. DOI: 10.1038/nmeth.2406.

[97] N. Tsanov, A. Samacoits, R. Chouaib, *et al.*, "smiFISH and FISH-quant – a flexible single RNA detection approach with super-resolution capability", *Nucleic Acids Research*, vol. 44, no. 22, art. e165, 2016. DOI: 10.1093/nar/gkw784.

[98] L. C. Stapel, B. Lombardot, C. Broaddus, *et al.*, "Automated detection and quantification of single RNAs at cellular resolution in zebrafish embryos", *Development*, vol. 143, no. 3, pp. 540–546, 2016. DOI: 10.1242/dev.128918.

[99] R. S. Wilson, L. Yang, A. Dun, *et al.*, "Automated single particle detection and tracking for large microscopy datasets", *Royal Society Open Science*, vol. 3, no. 5, art. 160225, 2016. DOI: 10.1098/rsos.160225.

[100] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[101] E. Ito, T. Sato, D. Sano, E. Utagawa, and T. Kato, "Virus particle detection by convolutional neural network in transmission electron microscopy images", *Food and Environmental Virology*, vol. 10, no. 2, pp. 201–208, 2018. DOI: 10.1007/s12560-018-9335-7.

[102] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[103] S. Helgadottir, A. Argun, and G. Volpe, "Digital video microscopy enhanced by deep learning", *Optica*, vol. 6, no. 4, pp. 506–513, Apr. 2019. DOI: 10.1364/OPTICA.6.000506.

[104] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", in *Advances in Neural Information Processing Systems*, vol. 28, Montreal, Canada: Curran Associates, Inc., 2015, pp. 91–99.

[105] L. Chen, M. Strauch, and D. Merhof, "Instance segmentation of biomedical images with an object-aware embedding learned with local constraints", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, Shenzhen, China: Springer, Cham, 2019, pp. 451–459. DOI: 10.1007/978-3-030-32239-7_50.

[106] W. Huang, S. Deng, C. Chen, X. Fu, and Z. Xiong, "Learning to model pixel-embedded affinity for homogeneous instance segmentation", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, virtual conference: AAAI Press, Palo Alto, California USA, 2022, pp. 1007–1015. DOI: 10.1609/aaai.v36i1.19984.

[107] H. Huang, X. Tang, F. Wen, and X. Jin, "Small object detection method with shallow feature fusion network for chip surface defect detection", *Scientific Reports*, vol. 12, no. 1, art. 3914, 2022. DOI: 10.1038/s41598-022-07654-x.

[108] C. Versari, S. Stoma, K. Batmanov, *et al.*, "Long-term tracking of budding yeast cells in brightfield microscopy: Cellstar and the evaluation platform", *Journal of The Royal Society Interface*, vol. 14, no. 127, art. 20160705, 2017. DOI: 10.1098/rsif.2016.0705.

[109] Z. Püspöki, D. Sage, J. P. Ward, and M. Unser, "SpotCaliper: fast wavelet-based spot detection with accurate size estimation", *Bioinformatics*, vol. 32, no. 8, pp. 1278–1280, 2015. DOI: 10.1093/bioinformatics/btv728.

[110] J. Schindelin, I. Arganda-Carreras, E. Frise, *et al.*, "Fiji: An open-source platform for biological-image analysis", *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012. DOI: 10.1038/nmeth.2019.

[111] F. de Chaumont and S. Dallongeville. "Spot detector". (2011), [Online]. Available: https://icy.bioimageanalysis.org/plugin/spot-detector/ (visited on 09/07/2022).

[112] F. de Chaumont, S. Dallongeville, N. Chenouard, *et al.*, "Icy: An open bioimage informatics platform for extended reproducible research", *Nature Methods*, vol. 9, no. 7, pp. 690–696, 2012. DOI: 10.1038/nmeth.2075.

[113] E. Katrukha, "ekatrukha/comdet: Comdet 0.5.5", version 0.5.5, *Zenodo*, 2022. DOI: 10.5281/zenodo.6546038.

[114] MOSAIC Group. "MosaicSuite documentation". (2021), [Online]. Available: https://sbalzarini-lab.org/MosaicSuiteDoc/particleTracker.html (visited on 09/07/2022).

[115] A. Imbert, W. Ouyang, A. Safieddine, *et al.*, "FISH-quant v2: A scalable and modular tool for smFISH image analysis", *RNA*, vol. 28, no. 6, pp. 786–795, 2022. DOI: 10.1261/rna.079073.121.

[116] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]", *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, 2012. DOI: 10.1109/MSP.2012.2204190.

[117] K. Sirinukunwattana, J. P. W. Pluim, H. Chen, *et al.*, "Gland segmentation in colon histology images: The glas challenge contest", *Medical Image Analysis*, vol. 35, pp. 489–502, 2017. DOI: 10.1016/j.media.2016.08.008.

[118] R. Cohn, I. Anderson, T. Prost, J. Tiarks, E. White, and E. Holm, "Instance segmentation for direct measurements of satellites in metal powders and automated microstructural characterization from image data", *JOM*, vol. 73, no. 7, pp. 2159–2172, 2021. DOI: 10.1007/s11837-021-04713-y.

[119] P. Monchot, L. Coquelin, K. Guerroudj, *et al.*, "Deep learning based instance segmentation of titanium dioxide particles in the form of agglomerates in scanning electron microscopy", *Nanomaterials*, vol. 11, no. 4, art. 968, 2021. DOI: 10.3390/nano11040968.

[120] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, 2019, pp. 9396–9405. DOI: 10.1109/CVPR.2019.00963.

[121] F. Kulwa, C. Li, X. Zhao, *et al.*, "A state-of-the-art survey for microorganism image segmentation methods and future potential", *IEEE Access*, vol. 7, pp. 100 243–100 269, 2019. DOI: 10.1109/ACCESS.2019.2930111.

[122] X. Chen, X. Zhou, and S. T. C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy", *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 4, pp. 762–766, 2006. DOI: 10.1109/TBME.2006.870201.

[123] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms", *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, 1993. DOI: 10.1109/83.217222.

[124] P. S. Umesh Adiga and B. B. Chaudhuri, "An efficient method based on watershed and rule-based merging for segmentation of 3-d histo-pathological images", *Pattern Recognition*, vol. 34, no. 7, pp. 1449–1458, 2001. DOI: 10.1016/S0031-3203(00)00076-5.

[125] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010. DOI: 10.1109/TBME.2009.2035102.

[126] J. Stegmaier, J. C. Otte, A. Kobitski, *et al.*, "Fast segmentation of stained nuclei in terabyte-scale, time resolved 3d microscopy image stacks", *PLOS ONE*, vol. 9, no. 2, art. e90036, 2014. DOI: 10.1371/journal.pone.0090036.

[127] B. P. Marsh, N. Chada, R. R. Sanganna Gari, K. P. Sigdel, and G. M. King, "The Hessian blob algorithm: Precise particle detection in atomic force microscopy imagery", *Scientific Reports*, vol. 8, no. 1, art. 978, 2018. DOI: 10.1038/s41598-018-19379-x.

[128] M. C. Robitaille, J. M. Byers, J. A. Christodoulides, and M. P. Raphael, "Robust optical flow algorithm for general single cell segmentation", *PLOS ONE*, vol. 17, no. 1, art. e0261763, 2022. DOI: 10.1371/journal.pone.0261763.

[129] T. Vicar, J. Balvan, J. Jaros, *et al.*, "Cell segmentation methods for label-free contrast microscopy: Review and comprehensive comparison", *BMC Bioinformatics*, vol. 20, no. 1, art. 360, 2019. DOI: 10.1186/s12859-019-2880-8.

[130] F. A. Guerrero-Peña, P. D. M. Fernandez, T. I. Ren, and A. Cunha, "A weakly supervised method for instance segmentation of biological cells", in *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*, Shenzhen, China: Springer, Cham, 2019, pp. 216–224. DOI: 10.1007/978-3-030-33391-1_25.

[131] F. A. Guerrero Peña, P. D. Marrero Fernandez, P. T. Tarr, T. I. Ren, E. M. Meyerowitz, and A. Cunha, "J regularization improves imbalanced multiclass segmentation", in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Iowa City, IA, USA: IEEE, 2020, pp. 1–5. DOI: 10.1109/ISBI45749.2020.9098550.

[132] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function", *arXiv*, 2017, preprint, version v1. DOI: 10.48550/ARXIV.1708.02551.

[133] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation for autonomous driving", in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA: IEEE, 2017, pp. 478–480. DOI: 10.1109/CVPRW.2017.66.

[134] D. Neven, B. D. Brabandere, M. Proesmans, and L. Van Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, 2019, pp. 8829–8837. DOI: 10.1109/CVPR.2019.00904.

[135] M. Lalit, P. Tomancak, and F. Jug, "Embedding-based instance segmentation in microscopy", in *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*, vol. 143, Lübeck, Germany: JMLR, 2021, pp. 399–415.

[136] K. Löffler and R. Mikut, "EmbedTrack – simultaneous cell segmentation and tracking through learning offsets and clustering bandwidths", *IEEE Access*, vol. 10, pp. 77 147–77 157, 2022. DOI: 10.1109/ACCESS.2022.3192880.

[137] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[138] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, "Cell detection with star-convex polygons", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Granada, Spain: Springer, Cham, 2018, pp. 265–273. DOI: 10.1007/978-3-030-00934-2_30.

[139] P. Naylor, M. Laé, F. Reyal, and T. Walter, "Segmentation of nuclei in histopathology images by deep regression of the distance map", *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 448–459, 2019. DOI: 10.1109/TMI.2018.2865709.

[140] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: A generalist algorithm for cellular segmentation", *Nature Methods*, vol. 18, pp. 100–106, 2021. DOI: 10.1038/s41592-020-01018-x.

[141] D. A. Van Valen, T. Kudo, K. M. Lane, *et al.*, "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments", *PLOS Computational Biology*, vol. 12, no. 11, art. e1005177, 2016. DOI: 10.1371/journal.pcbi.1005177.

[142] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä, "Cell proposal network for microscopy image analysis", in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA: IEEE, 2016, pp. 3199–3203. DOI: 10.1109/ICIP.2016.7532950.

[143] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä, "Cell segmentation proposal network for microscopy image analysis", in *Deep Learning and Data Labeling for Medical Applications*, Athens, Greece: Springer, Cham, 2016, pp. 21–29. DOI: 10.1007/978-3-319-46976-8_3.

[144] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "DCAN: Deep contour-aware networks for object instance segmentation from histology images", *Medical Image Analysis*, vol. 36, pp. 135–146, 2017. DOI: 10.1016/j.media.2016.11.004.

[145] Y. Xu, Y. Li, Y. Wang, *et al.*, "Gland instance segmentation using deep multichannel neural networks", *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2901–2912, 2017. DOI: 10.1109/TBME.2017.2686418.

[146] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology", *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1550–1560, 2017. DOI: 10.1109/TMI.2017.2677499.

[147] S. Seferbekov. "[ods.ai] topcoders, 1st place solution". Winner of the 2018 Data Science Bowl. (2018), [Online]. Available: https://www.kaggle.com/competitions/data-science-bowl-2018/discussion/54741 (visited on 06/30/2022).

[148] V. Kulikov, V. Yurchenko, and V. Lempitsky, "Instance segmentation by deep coloring", *arXiv*, 2018, preprint, version v1. DOI: 10.48550/ARXIV.1807.10007.

[149] F. A. Guerrero-Peña, P. D. Marrero Fernandez, T. Ing Ren, M. Yui, E. Rothenberg, and A. Cunha, "Multiclass weighted loss for instance segmentation of cluttered cells", in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece: IEEE, 2018, pp. 2451–2455. DOI: 10.1109/ICIP.2018.8451187.

[151] J. W. Johnson, "Automatic nucleus segmentation with Mask-RCNN", in *Advances in Computer Vision*, Las Vegas, NV, USA: Springer, Cham, 2020, pp. 399–407. DOI: 10.1007/978-3-030-17798-0_32.

[152] W. Wang, D. A. Taft, Y.-J. Chen, *et al.*, "Learn to segment single cells with deep distance estimator and deep cell detector", *Computers in Biology and Medicine*, vol. 108, pp. 133–141, 2019. DOI: 10.1016/j.compbiomed.2019.04.006.

[153] J. Li, Z. Hu, and S. Yang, "Accurate nuclear segmentation with center vector encoding", in *Information Processing in Medical Imaging*, Hong Kong, China: Springer, Cham, 2019, pp. 394–404. DOI: 10.1007/978-3-030-20351-1_30.

[154] X. Li, Y. Wang, Q. Tang, Z. Fan, and J. Yu, "Dual U-Net for the segmentation of overlapping glioma nuclei", *IEEE Access*, vol. 7, pp. 84 040–84 052, 2019. DOI: 10.1109/ACCESS.2019.2924744.

[155] A. Arbelle and T. R. Raviv, "Microscopy cell segmentation via convolutional LSTM networks", in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy: IEEE, 2019, pp. 1008–1012. DOI: 10.1109/ISBI.2019.8759447.

[156] J. C. Caicedo, J. Roth, A. Goodman, *et al.*, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images", *Cytometry A*, vol. 95, no. 9, pp. 952–965, 2019. DOI: 10.1002/cyto.a.23863.

[157] S. Graham, Q. D. Vu, S. E. A. Raza, *et al.*, "Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images", *Medical Image Analysis*, vol. 58, art. 101563, 2019. DOI: 10.1016/j.media.2019.101563.

[158] F. Lux and P. Matula, "Cell segmentation by combining marker-controlled watershed and deep learning", *arXiv*, 2020, preprint, version v1. DOI: 10.48550/ARXIV.2004.01607.

[159] R. Hollandi, A. Szkalisity, T. Toth, *et al.*, "NucleAIzer: A parameter-free deep learning framework for nucleus segmentation using image style transfer", *Cell Systems*, vol. 10, no. 5, pp. 453–458, 2020. DOI: 10.1016/j.cels.2020.04.003.

[160] H. Wang, M. Xian, and A. Vakanski, "Bending loss regularized network for nuclei segmentation in histopathology images", in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Iowa City, IA, USA: IEEE, 2020, pp. 1–5. DOI: 10.1109/ISBI45749.2020.9098611.

[161] V. Kulikov and V. Lempitsky, "Instance segmentation of biological images using harmonic embeddings", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, 2020, pp. 3842–3850. DOI: 10.1109/CVPR42600.2020.00390.

[162] Y. Tokuoka, T. G. Yamada, D. Mashiko, *et al.*, "3d convolutional neural networks-based segmentation to acquire quantitative criteria of the nucleus during mouse embryogenesis", *npj Systems Biology and Applications*, vol. 6, no. 1, art. 32, 2020. DOI: 10.1038/s41540-020-00152-8.

[163] G. Zaki, P. R. Gudla, K. Lee, *et al.*, "A deep learning pipeline for nucleus segmentation", *Cytometry Part A*, vol. 97, no. 12, pp. 1248–1264, 2020. DOI: 10.1002/cyto.a.24257.

[164] N. Dietler, M. Minder, V. Gligorovski, *et al.*, "A convolutional neural network segments yeast microscopy images with high accuracy", *Nature Communications*, vol. 11, no. 1, art. 5723, 2020. DOI: 10.1038/s41467-020-19557-4.

[165] W.-D. Jang, D. Wei, X. Zhang, *et al.*, "Learning vector quantized shape code for amodal blastomere instance segmentation", *arXiv*, 2020, preprint, version v1. DOI: 10.48550/ARXIV.2012.00985.

[166] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein. "DKFZ-GE". Cell Tracking Challenge algorithm description. (2020), [Online]. Available: https://public.celltrackingchallenge.net/participants/DKFZ-GE.pdf (visited on 07/19/2022).

[169] T.-O. Buchholz, M. Prakash, D. Schmidt, A. Krull, and F. Jug, "DENOISEG: Joint denoising and segmentation", in *Computer Vision – ECCV 2020 Workshops*, Glasgow, UK: Springer, Cham, 2020, pp. 324–337. DOI: 10.1007/978-3-030-66415-2_21.

[170] J. L. Rumberger, L. Mais, and D. Kainmueller, "Probabilistic deep learning for instance segmentation", in *Computer Vision – ECCV 2020 Workshops*, Glasgow, UK: Springer, Cham, 2020, pp. 445–457. DOI: 10.1007/978-3-030-66415-2_29.

[171] D. Eschweiler, M. Rethwisch, S. Koppers, and J. Stegmaier, "Spherical harmonics for shape-constrained 3d cell segmentation", in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, Nice, France: IEEE, 2021, pp. 792–796. DOI: 10.1109/ISBI48211.2021.9433983.

[172] S. Mandal and V. Uhlmann, "Splinedist: Automated cell segmentation with spline curves", in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, Nice, France: IEEE, 2021, pp. 1082–1086. DOI: 10.1109/ISBI48211.2021.9433928.

[173] F. C. Walter, S. Damrich, and F. A. Hamprecht, "Multistar: Instance segmentation of overlapping objects with star-convex polygons", in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, Nice, France: IEEE, 2021, pp. 295–298. DOI: 10.1109/ISBI48211.2021.9433769.

[174] M. Lalit, P. Tomancak, and F. Jug, "EmbedSeg: Embedding-based instance segmentation for biomedical microscopy data", *Medical Image Analysis*, vol. 81, art. 102523, 2022. DOI: 10.1016/j.media.2022.102523.

[175] D. Hirling and P. Horvath, "Fully automatic cell segmentation with fourier descriptors", *bioRxiv*, 2021, preprint, version v1. DOI: 10.1101/2021.12.17.472408.

[176] A. Arbelle, S. Cohen, and T. R. Raviv, "Dual-task ConvLSTM-UNet for instance segmentation of weakly annotated microscopy videos", *IEEE Transactions on Medical Imaging*, vol. 41, no. 8, pp. 1948–1960, 2022. DOI: 10.1109/TMI.2022.3152927.

[177] H. He, Z. Huang, Y. Ding, *et al.*, "CDNet: Centripetal direction network for nuclear instance segmentation", in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, Canada: IEEE, 2021, pp. 4006–4015. DOI: 10.1109/ICCV48922.2021.00399.

[178] Y. Jia, C. Lu, X. Li, *et al.*, "Nuclei instance segmentation and classification in histopathological images using a DT-Yolact", in *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, London, United Kingdom: IEEE, 2021, pp. 414–420. DOI: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00072.

[179] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation", in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, 2019, pp. 9156–9165. DOI: 10.1109/ICCV.2019.00925.

[180] J. Zhang, Y. Wang, E. D. Donarski, *et al.*, "BCM3D 2.0: Accurate segmentation of single bacterial cells in dense biofilms using computationally generated intermediate image representations", *npj Biofilms and Microbiomes*, vol. 8, no. 1, art. 99, 2022. DOI: 10.1038/s41522-022-00362-4.

[181] R. Wagner and K. Rohr, "CellCentroidFormer: Combining self-attention and convolution for cell detection", in *Medical Image Understanding and Analysis*, Cambridge, UK: Springer, Cham, 2022, pp. 212–222. DOI: 10.1007/978-3-031-12053-4_16.

[182] K. J. Cutler, C. Stringer, T. W. Lo, *et al.*, "Omnipose: A high-precision morphology-independent solution for bacterial cell segmentation", *Nature Methods*, vol. 19, no. 11, pp. 1438–1448, 2022. DOI: 10.1038/s41592-022-01639-4.

[183] J. Gamper, N. Alemi Koohbanani, K. Benet, A. Khuram, and N. Rajpoot, "PanNuke: An open pan-cancer histology dataset for nuclei instance segmentation and classification", in *Digital Pathology*, Warwick, UK: Springer, Cham, 2019, pp. 11–19. DOI: 10.1007/978-3-030-23937-4.

[184] J. Gamper, N. A. Koohbanani, K. Benes, *et al.*, "PanNuke dataset extension, insights and baselines", *arXiv*, 2020, preprint, version v7. DOI: 10.48550/ARXIV.2003.10778.

[185] R. Verma, N. Kumar, A. Patil, *et al.*, "MoNuSAC2020: A multi-organ nuclei segmentation and classification challenge", *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3413–3423, 2021. DOI: 10.1109/TMI.2021.3085712.

[186] S. Graham, M. Jahanifar, Q. D. Vu, *et al.*, "CoNIC: Colon nuclei identification and counting challenge 2022", *arXiv*, 2021, preprint, version v1. DOI: 10.48550/arXiv.2111.14485.

[187] A. Mahbod, G. Schaefer, B. Bancher, *et al.*, "CryoNuSeg: A dataset for nuclei instance segmentation of cryosectioned H&E-stained histological images", *Computers in Biology and Medicine*, vol. 132, art. 104349, 2021. DOI: 10.1016/j.compbiomed.2021.104349.

[188] S. Graham, H. Chen, J. Gamper, *et al.*, "MILD-Net: Minimal information loss dilated network for gland instance segmentation in colon histology images", *Medical Image Analysis*, vol. 52, pp. 199–211, 2019. DOI: 10.1016/j.media.2018.12.001.

[189] E. S. Nasir, A. Perviaz, and M. M. Fraz, "Nuclei & glands instance segmentation in histology images: A narrative review", *arXiv*, 2022, preprint, version v1. DOI: 10.48550/ARXIV.2208.12460.

[190] M. Schwendy, R. E. Unger, and S. H. Parekh, "EVICAN—a balanced dataset for algorithm development in cell and nucleus segmentation", *Bioinformatics*, vol. 36, no. 12, pp. 3863–3870, 2020. DOI: 10.1093/bioinformatics/btaa225.

[191] C. Edlund, T. R. Jackson, N. Khalid, *et al.*, "LIVECell—a large-scale dataset for label-free live cell segmentation", *Nature Methods*, vol. 18, no. 9, pp. 1038–1045, 2021. DOI: 10.1038/s41592-021-01249-6.

[192] I. de Cesare, C. G. Zamora-Chimal, L. Postiglione, *et al.*, "ChipSeg: An automatic tool to segment bacterial and mammalian cells cultured in microfluidic devices", *ACS Omega*, vol. 6, no. 4, pp. 2473–2476, 2021. DOI: 10.1021/acsomega.0c03906.

[193] C. McQuin, A. Goodman, V. Chernyshev, *et al.*, "CellProfiler 3.0: Next-generation image processing for biology", *PLOS Biology*, vol. 16, no. 7, art. e2005970, 2018. DOI: 10.1371/journal.pbio.2005970.

[194] D. R. Stirling, M. J. Swain-Bowden, A. M. Lucas, A. E. Carpenter, B. A. Cimini, and A. Goodman, "CellProfiler 4: Improvements in speed, utility and usability", *BMC Bioinformatics*, vol. 22, no. 1, art. 433, 2021. DOI: 10.1186/s12859-021-04344-9.

[195] S. Berg, D. Kutra, T. Kroeger, *et al.*, "Ilastik: Interactive machine learning for (bio)image analysis", *Nature Methods*, vol. 16, no. 12, pp. 1226–1232, 2019. DOI: 10.1038/s41592-019-0582-9.

[196] E. Gómez-de-Mariscal, C. García-López-de-Haro, W. Ouyang, *et al.*, "DeepImageJ: A user-friendly environment to run deep learning models in ImageJ", *Nature Methods*, vol. 18, pp. 1192–1195, 2021. DOI: 10.1038/s41592-021-01262-9.

[197] W. Ouyang, F. Beuttenmueller, E. Gómez-de-Mariscal, *et al.*, "Bioimage model zoo: A community-driven resource for accessible deep learning in bioimage analysis", *bioRxiv*, 2022, preprint, version v1. DOI: 10.1101/2022.06.07.495102.

[198] M. Pachitariu and C. Stringer, "Cellpose 2.0: How to train your own model", *Nature Methods*, vol. 19, no. 12, pp. 1634–1641, 2022. DOI: 10.1038/s41592-022-01663-4.

[199] M. Griebel, D. Segebarth, N. Stein, *et al.*, "Deep-learning in the bioimaging wild: Handling ambiguous data with deepflash2", *arXiv*, 2021, preprint, version v1. DOI: 10.48550/ARXIV.2111.06693.

[200] F. Padovani, B. Mairhörmann, P. Falter-Braun, J. Lengefeld, and K. M. Schmoller, "Segmentation, tracking and cell cycle analysis of live-cell imaging data with Cell-ACDC", *BMC Biology*, vol. 20, no. 1, art. 174, 2022. DOI: 10.1186/s12915-022-01372-6.

[201] N. Sofroniew, T. Lambert, K. Evans, *et al.*, "Napari: A multi-dimensional image viewer for python", version v0.4.16, *Zenodo*, 2022. DOI: 10.5281/zenodo.6598542.

[202] A. u. M. Khan, A. Torelli, I. Wolf, and N. Gretz, "AutoCellSeg: Robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques", *Scientific Reports*, vol. 8, no. 1, art. 7302, 2018. DOI: 10.1038/s41598-018-24916-9.

[203] R. Hartmann, M. C. F. van Teeseling, M. Thanbichler, and K. Drescher, "BacStalk: A comprehensive and interactive image analysis software tool for bacterial cell biology", *Molecular Microbiology*, vol. 114, no. 1, pp. 140–150, 2020. DOI: 10.1111/mmi.14501.

[204] O. Hilsenbeck, M. Schwarzfischer, D. Loeffler, *et al.*, "fastER: A user-friendly tool for ultrafast and robust cell segmentation in large-scale microscopy", *Bioinformatics*, vol. 33, no. 13, pp. 2020–2028, 2017. DOI: 10.1093/bioinformatics/btx107.

[205] M. Arzt, J. Deschamps, C. Schmied, *et al.*, "LABKIT: Labeling and segmentation toolkit for big image data", *Frontiers in Computer Science*, vol. 4, art. 777728, 2022. DOI: 10.3389/fcomp.2022.777728.

[206] S. Panigrahi, D. Murat, A. Le Gall, *et al.*, "Misic, a general deep learning-based method for the high-throughput cell segmentation of complex bacterial communities", *eLife*, vol. 10, e65151, 2021. DOI: 10.7554/eLife.65151.

[207] M. Stritt, A. K. Stalder, and E. Vezzali, "Orbit image analysis: An open-source whole slide image analysis tool", *PLOS Computational Biology*, vol. 16, no. 2, art. e1007313, 2020. DOI: 10.1371/journal.pcbi.1007313.

[208] D. Ershov, M.-S. Phan, J. W. Pylvänäinen, *et al.*, "TrackMate 7: Integrating state-of-the-art segmentation algorithms into tracking pipelines", *Nature Methods*, vol. 19, no. 7, pp. 829–832, 2022. DOI: 10.1038/s41592-022-01507-1.

[209] A. X. Lu, T. Zarin, I. S. Hsu, and A. M. Moses, "YeastSpotter: Accurate and parameter-free web segmentation for microscopy images of yeast cells", *Bioinformatics*, vol. 35, no. 21, pp. 4525–4527, 2019. DOI: 10.1093/bioinformatics/btz402.

[210] L. von Chamier, R. F. Laine, J. Jukkala, *et al.*, "Democratising deep learning for microscopy with ZeroCostDL4Mic", *Nature Communications*, vol. 12, no. 1, art. 2276, 2021. DOI: 10.1038/s41467-021-22518-0.

[211] L. Maier-Hein, A. Reinke, E. Christodoulou, *et al.*, "Metrics reloaded: Pitfalls and recommendations for image analysis validation", *arXiv*, 2023, preprint, version v5. DOI: 10.48550/ARXIV.2206.01653.

[212] P. Matula, M. Maška, D. V. Sorokin, P. Matula, C. Ortiz-de-Solórzano, and M. Kozubek, "Cell tracking accuracy measurement based on comparison of acyclic oriented graphs", *PLOS ONE*, vol. 10, no. 12, art. e0144959, 2015. DOI: 10.1371/journal.pone.0144959.

[213] Q. D. Vu and S. Graham. "HoVer-Net". GitHub repository, commit a0f80c7. (2021), [Online]. Available: https://github.com/vqdang/hover_net (visited on 08/04/2021).

[215] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations", in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Québec City, Canada: Springer, Cham, 2017, pp. 240–248. DOI: 10.1007/978-3-319-67558-9_28.

[216] R. Zhao, B. Qian, X. Zhang, *et al.*, "Rethinking Dice loss for medical image segmentation", in *2020 IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy: IEEE, 2020, pp. 851–860. DOI: 10.1109/ICDM50108.2020.00094.

[219] C. Allan, J.-M. Burel, J. Moore, *et al.*, "OMERO: Flexible, model-driven data management for experimental biology", *Nature Methods*, vol. 9, no. 3, pp. 245–253, 2012. DOI: 10.1038/nmeth.1896.

[220] L. Wright. "Ranger - a synergistic optimizer." GitHub repository, commit: a170e3d. (2020), [Online]. Available: https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer (visited on 12/08/2020).

[221] D. Misra, "Mish: A self regularized non-monotonicactivation function", in *The 31st British Machine Vision Virtual Conference*, virtual conference: British Machine Vision Association, 2020, pp. 1–14.

[222] R. Aversa, M. H. Modarres, S. Cozzini, R. Ciancio, and A. Chiusole, "The first annotated set of scanning electron microscopy images for nanoscience", *Scientific Data*, vol. 5, no. 1, art. 180172, 2018. DOI: 10.1038/sdata.2018.172.

[223] R. Aversa, M. H. Modarres, S. Cozzini, and R. Ciancio, "NFFA-EUROPE - 100% SEM dataset", *B2SHARE*, 2018. DOI: 10.23728/B2SHARE.80DF8606FCDB4B2BAE1656F0DC6DB8BA.

[224] S. Mubeen, T. Zhang, B. Yoo, M. A. Deshusses, and N. V. Myung, "Palladium nanoparticles decorated single-walled carbon nanotube hydrogen sensor", *The Journal of Physical Chemistry C*, vol. 111, no. 17, pp. 6321–6327, 2007. DOI: 10.1021/jp067716m.

[225] K. Hebert, D. Seidman, A. Oki, *et al.*, "Anaplasma marginale outer membrane protein A is an adhesin that recognizes sialylated and fucosylated glycans and functionally depends on an essential binding domain", *Infection and Immunity*, vol. 85, no. 3, 2017. DOI: 10.1128/IAI.00968-16.

[226] C. Jung, A. Matzke, H. H. Niemann, C. Schwerk, T. Tenenbaum, and V. Orian-Rousseau, "Involvement of CD44v6 in InlB-dependent Listeria invasion", *Molecular Microbiology*, vol. 72, no. 5, pp. 1196–1207, 2009. DOI: 10.1111/j.1365-2958.2009.06716.x.

[227] A. Bartschat, E. Hübner, M. Reischl, R. Mikut, and J. Stegmaier, "XPIWIT—an xml pipeline wrapper for the insight toolkit", *Bioinformatics*, vol. 32, pp. 315–317, 2015. DOI: 10.1093/bioinformatics/btv559.

[228] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, 2018, pp. 2235–2245. DOI: 10.1109/CVPR.2018.00238.

[229] L. Barone, J. Williams, and D. Micklos, "Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators", *PLOS Computational Biology*, vol. 13, no. 10, art. e1005858, 2017. DOI: 10.1371/journal.pcbi.1005755.

[230] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts", *arXiv*, 2016, preprint, version v5. DOI: 10.48550/ARXIV.1608.03983.

[231] E. Kaganovitch, X. Steurer, D. Dogan, C. Probst, W. Wiechert, and D. Kohlheyer, "Microbial single-cell analysis in picoliter-sized batch cultivation chambers", *New Biotechnology*, vol. 47, pp. 50–59, 2018. DOI: 10.1016/j.nbt.2018.01.009.

[234] H. Yu, Y. Zhou, J. Simmons, *et al.*, "Groupwise tracking of crowded similar-appearance targets from low-continuity image sequences", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 952–960. DOI: 10.1109/CVPR.2016.109.

[235] Y. Zhou, H. Yu, J. Simmons, C. P. Przybyla, and S. Wang, "Large-scale fiber tracking through sparsely sampled image sequences of composite materials", *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4931–4942, 2016. DOI: 10.1109/TIP.2016.2598640.

[237] D. Svoboda, M. Kozubek, and S. Stejskal, "Generation of digital phantoms of cell nuclei and simulation of image formation in 3d image cytometry", *Cytometry A*, vol. 75A, no. 6, pp. 494–509, 2009. DOI: 10.1002/cyto.a.20714.

[238] J. Stegmaier, "New methods to improve large-scale microscopy image analysis with prior knowledge and uncertainty", Ph.D. dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2017. DOI: 10.5445/KSP/1000060221.

[239] M. Kozubek, M. Maška, and C. Ortiz-de-Solorzano, *Cell Tracking Challenge*, Cell Tracking Challenge Workshop at the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), 2021.

[242] A. Chamanzar and Y. Nie, “Weakly supervised multi-task learning for cell detection and segmentation”, in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Iowa City, IA, USA: IEEE, 2020, pp. 513–516. DOI: 10.1109/ISBI45749.2020.9098518.

[243] T. Zhao and Z. Yin, “Weakly supervised cell segmentation by point annotation”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 10, pp. 2736–2747, 2021. DOI: 10.1109/TMI.2020.3046292.

[244] L. Rettenberger, M. Schilling, S. Elser, M. Böhland, and M. Reischl, “Self-supervised learning for annotation efficient biomedical image segmentation”, *IEEE Transactions on Biomedical Engineering*, pp. 1–11, 2023. DOI: 10.1109/TBME.2023.3252889.

[245] D. Eschweiler, M. Rethwisch, M. Jarchow, S. Koppers, and J. Stegmaier, “3d fluorescence microscopy data synthesis for segmentation and benchmarking”, *PLOS ONE*, vol. 16, no. 12, art. e0260509, 2021. DOI: 10.1371/journal.pone.0260509.

[246] D. Eschweiler and J. Stegmaier, “Denoising diffusion probabilistic models for generation of realistic fully-annotated microscopy image data sets”, *arXiv*, 2023, preprint, version v1. DOI: 10.48550/arXiv.2301.10227.

[247] R. Bruch, F. Keller, M. Böhland, *et al.*, “Synthesis of large scale 3d microscopic images of 3d cell cultures for training and benchmarking”, *PLOS ONE*, vol. 18, no. 3, art. e0283828, 2023. DOI: 10.1371/journal.pone.0283828.

[248] L. Maier-Hein, M. Eisenmann, A. Reinke, *et al.*, “Why rankings of biomedical image analysis competitions should be interpreted with care”, *Nature Communications*, vol. 9, no. 1, art. 5217, 2018. DOI: 10.1038/s41467-018-07619-7.

[250] R. Dror, S. Shlomov, and R. Reichart, “Deep dominance - how to properly compare deep neural models”, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 2773–2785. DOI: 10.18653/v1/P19-1266.

[251] D. Ulmer, C. Hardmeier, and J. Frellsen, “deep-significance - easy and meaningful statistical significance testing in the age of neural networks”, *arXiv*, 2022, preprint, version v1. DOI: 10.48550/arXiv.2204.06815.

[252] E. del Barrio, J. A. Cuesta-Albertos, and C. Matrán, “An optimal transportation approach for assessing almost stochastic order”, in *The Mathematics of the Uncertain*, Springer, Cham, 2018, pp. 33–44. DOI: 10.1007/978-3-319-73848-2_3.

[253] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA: IEEE, 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81.