Distribution Agreement

Signature:

_______________________________________ _____________

Hanyi Yu Date

Computational Image Processing and Deep Learning with Multi-Model Biomedical Image Data

By

Hanyi Yu
Doctor of Philosophy

Computer Science and Informatics

Jun Kong, Ph.D.
Advisor

Vaidy Sunderam, Ph.D.
Co-advisor

Imon Banerjee, Ph.D.
Committee Member

John Nickerson, Ph.D.
Committee Member
Accepted:

Kimberly Jacob Arriola, Ph.D., MPH
Dean of the James T. Laney School of Graduate Studies

Date

Computational Image Processing and Deep Learning with Multi-Model Biomedical Image Data

By

Hanyi Yu
B.Eng., Shanghai Jiao Tong University, Shanghai, China, 2014
M.Eng., Shanghai Jiao Tong University, Shanghai, China, 2017

Advisor: Jun Kong, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2022

Abstract

Computational Image Processing and Deep Learning with Multi-Model Biomedical Image Data
By Hanyi Yu

With the rapid advance in medical imaging technology in recent decades, computational image analysis has become a popular research topic in the field of biomedical informatics. Images from various imaging acquisition platforms have been widely used for the early detection, diagnosis, and treatment response assessment in a large number of disease and cancer studies. Although conventional computational methods present higher analysis efficiency and less variability than manual analyses, they require appropriate parameter settings to achieve optimal results. This can be demanding for medical researchers lacking relevant knowledge about computational method development. In the last decade, deep neural networks trained on large-scale labeled datasets have provided a promising and convenient end-to-end solution to biomedical image processing. However, the development of deep-learning tools for biomedical image analysis is often restrained by inadequate data with high-quality annotations in practice. By contrast, a large number of unlabeled biomedical images are generated by daily research and clinical activities. Thus, leveraging unlabeled images with semi-supervised or even unsupervised deep learning approaches has become a significant research direction in biomedical informatics analysis.

My primary doctoral research focuses on the field of medical image processing, utilizing computational methods to facilitate biomedical image analysis with limited supervision. I have explored two ways to achieve this goal: (1) Optimizing the model of existing approaches for specific tasks and (2) Developing semi-supervised/unsupervised deep learning approaches. In my research, I mainly focus on image segmentation and object tracking, two common biomedical image analysis tasks. By experimenting with different types of images (e.g., fluorescence microscopy images and histopathology microscopy images) from various sources (e.g., bacteria, human liver biopsies, and retinal pigment epithelium tissues), my developed methods demonstrate their promising potential to support biomedical image analysis tasks.

Computational Image Processing and Deep Learning with Multi-Model Biomedical Image Data

By

Hanyi Yu
B.Eng., Shanghai Jiao Tong University, Shanghai, China, 2014
M.Eng., Shanghai Jiao Tong University, Shanghai, China, 2017

Advisor: Jun Kong, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2022

Acknowledgments

During my doctoral studies, I have received enormous support from my advisors, collaborators, friends, and family. This dissertation cannot be completed without their help.

I am deeply indebted to my advisor, Dr. Jun Kong, for his insightful guidance, beneficial advice, and precious support throughout my doctoral career. The encouraging and inspiring discussions in your office and online, the meticulous revisions on my paper writing, amongst many other precious supports, make working in your lab an unforgettable and life-changing journey. I also sincerely appreciate my co-advisor, Dr. Vaidy Sunderam, for his considerate assistance and supervision of my doctoral studies.

I would like to express my gratitude to Dr. John Nickerson for supporting my research program. I obtain lots of help from him and my other collaborators, Dr. Yong-Kyu Kim, Dr. Alton B Farris, Ms. Bre Shelton, Dr. Nima Sharifai, Dr. Kun Jiang, Dr. Fusheng Wang, and Dr. George Teodoro, for expertise and data annotations.

I would like to thank my friends, Xiang Li, and Xiaoyuan Guo, for their helpful discussions and encouragement in academics.

Finally, I am extremely grateful to my parents, Shuling Han and Junlin Yu, and my partner, Fan Chen, for their constant understanding, support, and encouragement. They are my motivation to go the extra mile.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent decades, the rapid development of medical imaging technologies makes the computational image analysis important and active in the field of biomedical informatics. Numerous imaging acquisition methods, e.g., computed tomography (CT), magnetic resonance (MR), positron emission tomography (PET), ultrasound, X-ray, fluorescence microscopy, and histopathology microscopy, produce image data of a large number of image modalities widely used for the early detection, diagnosis, and treatment response assessment of diseases [1].

The analyses and interpretations of biomedical images have been mostly conducted by domain experts. Although conventional computational methods present a higher efficiency and robustness than manual analyses, medical researchers need to be knowledgeable with relevant computational expertise before they can set parameters appropriately and achieve optimal results. In the last decade, deep neural networks trained on sufficiently large labeled datasets have provided convenient end-to-end solutions to biomedical image processing. However, the performance of supervised deep-learning methods for biomedical image analysis highly depends on large-scale good-quality annotations. In practice, such annotated data collection is restrained by the limited human annotation bandwidth, various ethical and legal constraints, and large intra- and inter-variability [2]. By contrast, daily research and clinical activities produce a large number of unlabeled biomedical images. Thus, leveraging unlabeled images with semi-supervised and even unsupervised deep learning methods has emerged as a promising research direction in the field of biomedical image analysis.

1.1 Research contributions

This dissertation presents medical image processing research solutions that utilize computational methods to facilitate biomedical image analysis under limited supervision. Two ways have been explored to achieve this goal: (1) Optimizing the architecture of existing models for specific analysis tasks and (2) Developing new semi-/unsupervised deep learning methods. Two common biomedical image analysis tasks are included in this dissertation: image segmentation and object tracking. Validated by different image types (e.g., fluorescence microscopy images and bright field histopathology microscopy images) from various sources (e.g., bacteria, lung cancer spheroids, human liver biopsies, and retinal pigment epithelium tissues), the developed methods demonstrate their promising potential to support biomedical image analysis tasks. In summary, my research work contribution includes the following four aspects.

- **Object motion analysis for time-Lapse image sequences**. Two different models were presented to improve the tracking performance in non-Gaussian conditions. In addition, a new tracking management strategy was designed to accelerate the model updating. With the new mapping step after updating the particle states, the tracking accuracy was improved. The performance of this developed approach was demonstrated with both artificial image sequences and real time-lapse fluorescent image datasets that captured 2D bacteria and 3D lung cancer cells in motion. This work has been published as a book chapter in *Modern Statistical Methods for Health Research* [3]. Besides, this work also supported the bacteria motion analysis in an immunology study [4].

- **Biomedical image segmentation with supervised learning**. A novel UNet-based deep convolutional neural network was developed to automatically segment portal tract regions from high-resolution liver biopsy Whole Slide Images (WSIs). The two cascaded convolution structures in the original UNet design were substituted by a Residual Spatial Attention (RSA) processing block to enhance network performance. Additionally, the output layer of the developed network directly synthesized up-sampling features from multiple image resolutions. By such a Multiple Up-sampling Path (MUP) mechanism, the developed deep learning model reduced the false-negative rate and generated smoother borders. The network was trained with image patches and applied to liver biopsy WSIs. The resulting portal tract fibrotic percentage and average portal tract fibrotic area computed by the developed method presented a strong correlation with the clinical Scheuer fibrosis stage. The performance of this developed approach was both qualitatively and quantitatively compared with that of the widely used methods. To demonstrate the contributions of individual modules, I also conducted ablation experiments and presented ablation study results. The developed method presented superior performance, suggesting its promising potential to assist clinical diagnosis. This work has been published by *Computers in Biology and Medicine* [5].

- **Biomedical image segmentation with semi-supervised learning** The Generative Adversarial Networks (GANs) mechanism was leveraged to enrich the training dataset with a massive amount of unlabeled weak RPE cells and mitigate the model overfitting problem. The resulting deep learning model, namely MultiHeadGAN, was built upon the state-of-the-art image segmentation model UNet, but with a new training strategy simultaneously leveraging a small set of annotated and a large set of unlabeled RPE cells from flatmount microscopy images for morphology feature extraction and RPE structure reconstruction. Additionally, a new shape loss for model training was designed to produce closed cell borders. The method was both qualitatively and quantitatively evaluated and compared with state-of-the-art deep learning approaches. The extensive experimental results demonstrated the superiority of the developed segmentation method, suggesting its potential to facilitate further biomedical research on RPE aging. This work has been published by *Computers in Biology and Medicine* [6]. Besides, this work also supported cell morphological analysis in an ophthalmology study [7].

- **Biomedical image segmentation with self-supervised learning** A self-supervised learning strategy was developed to train a semantic segmentation network with an encoder-decoder architecture. A reconstruction and a pairwise representation loss were employed to make the encoder extract structural information, while a morphology loss was created to have the decoder produce the segmentation map. In addition, a novel image augmentation algorithm (AugCut) was developed to produce multiple views for self-supervised learning and enhance the network training performance. To validate the method efficacy, the developed S^4 method for RPE cell segmentation was compared with other state-of-the-art deep learning approaches. The developed method demonstrated a better performance by both qualitative and quantitative evaluations, suggesting its promising potential to support large-scale cell morphological analyses in RPE aging S^4 investigations.This work is currently under review by *Medical Image Analysis.*

1.2 Paper list

The publications that I substantially contributed to during my Ph.D. study are listed as follows:

- Yu, H., Yoon, S. B., Kauffman, R., Wrammert, J., Marcus, A., and Kong, J. (2021). Non-Gaussian Models for Object Motion Analysis with Time-Lapse Fluorescence Microscopy Images. In *Modern Statistical Methods for Health Research* (pp. 15-41). Springer, Cham. https://doi.org/10.1007/978-3-030-72437-5_2

- Yu, H., Wang, F., Teodoro, G., Nickerson, J., and Kong, J. (2022). MultiHeadGAN: A deep learning method for low contrast retinal pigment epithelium cell segmentation with fluorescent flatmount microscopy images. *Computers in Biology and Medicine*, 146, 105596. https://doi.org/10.1016/j.compbiomed.2022.105596

- Yu, H., Sharifai, S., Jiang, K., Fusheng, W., Teodoro, G., and Kong, J. (2022). Artificial Intelligence based Liver Portal Tract Region Identification and Quantification with Transplant Biopsy Whole-Slide Images. *Computers in Biology and Medicine*, 150, 106089. https://doi.org/10.1016/j.compbiomed.2022.106089

- Yu, H., Wang, F., Teodoro, G., Chen, F., Guo, X., Nickerson, J. and Kong, J. Self-supervised semantic segmentation of retinal pigment epithelium cells in flatmount fluorescent microscopy images. Under Review by *Medical Image Analysis.*

- Guo, X., Yu, H., Rossetti, B., Teodoro, G., Brat, D., and Kong, J. (2018, July). Clumped nuclei segmentation with adjacent point match and local shape-based intensity analysis in fluorescence microscopy images. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 3410-3413). IEEE. https://doi.org/10.1109/EMBC.2018.8512961

- Kauffman, R. C., Adekunle, O., Yu, H., Cho, A., Nyhoff, L. E., Kelly, M., Harris, J. B., Bhuiyan, T. R., Qadri, F., Calderwood S. B., Charles, R. C., Ryan, E. T., Kong, J., and Wrammert, J. (2021). Impact of immunoglobulin isotype and epitope on the functional properties of Vibrio cholerae O-specific polysaccharide-specific monoclonal antibodies. *Mbio*, 12(2), e03679-20. https://doi.org/10.1128/mBio.03679-20

- Kim, Y. K., Yu, H., Summers, V. R., Donaldson, K. J., Ferdous, S., Shelton, D., Zhang, N., Chrenek, M. A., Jiang, Y., Grossniklaus, H. E., Boatright, J. H., Kong, J., and Nickerson, J. M. (2021). Morphometric analysis of retinal pigment epithelial cells from C57BL/6J mice during aging. *Investigative Ophthalmology and Visual Science*, 62(2), 32-32. https://doi.org/10.1167/iovs.62.2.32

- Li, H., Yu, H., Kim, Y. K., Wang, F., Teodoro, G., Jiang, Y., Nickerson, J. M., and Kong, J. (2021). Computational Model-Based Estimation of Mouse Eyeball Structure From Two-Dimensional Flatmount Microscopy Images. *Translational Vision Science and Technology*, 10(4), 25-25. https://doi.org/10.1167/tvst.10.4.25

1.3 Outlines

The rest of the dissertation is organized as follows: Chapter 2 introduces the background knowledge of relevant research fields and significant studies in each field; Chapter 3 presents my developed method for object motion analysis and evaluation results with multiple time-lapse fluorescence image datasets; Chapter 4 presents my developed method for supervised biomedical image segmentation and evaluation results with a liver biopsy image dataset; In Chapter 5 and 6, I respectively present methods utilizing semi-supervised learning and self-supervised learning for biomedical image segmentation and evaluation results with a retinal pigment epithelium fluorescent microscopy image dataset; In Chapter 7, I conclude my research work and discuss future directions.

Chapter 2

Literature Review

This chapter provides a comprehensive review of the research work related to biomedical image processing on image segmentation, object tracking, unsupervised learning, generative adversarial networks, and data augmentation.

2.1 Deep learning

In recent decades, machine learning has become one of the most popular topics in the field of computer science. Traditionally, the performance of machine learning algorithms highly relies on the quality of extracted feature representations from data. Therefore, the feature engineering has been a significant research topic in machine learning for a long time [8]. Comparatively, deep learning algorithms, as a rising branch of machine learning algorithms, perform feature extraction in an automated way. They train multiple layers of artificial neural networks to extract latent representations of inputs for downstream tasks. Deep learning algorithms have achieved a great success in numerous application domains, including computer vision [9], natural language processing [10, 11], audio processing [12], and 3D point cloud processing [13], and reshaped a large number of industries, such as medical diagnosis [14, 15], autonomous driving [16], and traffic monitoring [17]. In some real-world applications, supervised learning models trained with sufficient high-quality data present dominant performance. However, deep learning models often suffer from inadequate labeled training data due to ethical and economic constraints in many fields. In such cases, unsupervised learning, semi-supervised learning, and weakly supervised learning become promising solutions [18, 19, 20].

Self-supervised learning is a collection of unsupervised learning methods that extract training signals from enormous amounts of unlabeled data and build good representations to facilitate downstream tasks. Based on how samples are organized, self-supervised learning approaches can be roughly divided into two broad categories: self-prediction and contrastive learning.

One typical set of self-prediction methods is to reconstruct the input signals using a bottleneck architecture that encodes a high-dimensional input into a latent low-dimensional code [21, 22, 23, 24]. Besides the reconstruction, some methods intentionally drop a part of the sample and predict the missing part of the sample given the existing part [25, 26, 27]. In these studies, some transformation (e.g., rotation, and jigsaw) of one data sample either maintains the original information or follows the desired innate logic, which provides supervision without task-specific labels [28, 29, 30].

Contrastive learning is a group of unsupervised representation learning methods with inputs from different branches encoded as latent representations for the loss calculation. The intent of contrastive learning is to associate representations of related inputs and disassociate representations of unrelated inputs. Multiple studies have suggested that a large number of negative representations can effectively boost the contrastive learning performance [31, 32, 33]. To achieve this goal, some studies store the representations of inputs in previous training epochs as negative representations [31, 32, 34, 35]. In addition to storing negative representations in a memory bank, other studies directly use negative inputs within the same training data batch [33, 36, 37].

Generative Adversarial Networks (GANs) [38] have achieved an impressive success in the field of image generation [39, 40, 41], image representation learning [42, 43], and image translation[44, 45, 46, 47, 48]. In general, a GAN structure consists of a generator and a discriminator. The discriminator is trained to judge if inputs are real or fake, while the generator is trained to produce synthetic outputs similar to real ones. For the supervised learning, the GAN-based method pixel2pixel provides a general framework for a wide range of applications, such as style transfer, object replacement, and background removal [44]. For the unsupervised learning, some studies adopt cycle consistency loss to make generators retain structural information [45, 46, 47]. In addition, CUT is another solution to the unsupervised learning by the contrastive learning in the training process [48].

2.2 Image segmentation

Semantic segmentation and instance segmentation are two typical image segmentation types that are frequently employed to support image analysis in biomedical research [49, 50, 51, 52, 53]. While semantic segmentation algorithms assign a class label to each pixel in an image, the goal of instance segmentation is to detect each object and delineate it with a bounding box or segmentation mask [54].

Early semantic segmentation approaches rely on hand-crafted features and traditional classifiers, including boosting [55], support vector machine [56], and random forest [57]. In the recent decade, deep neural networks have much advanced the development of semantic segmentation. For example, Fully Convolution Network (FCN) can create segmentation maps by replacing fully connected layers in classification neural networks [58, 59, 60] with deconvolution layers [61]. Built upon FCN, DeepLab successfully improves its performance by a multitude of techniques, such as atrous convolution, conditional random field, and spatial pyramid pooling [62]. Additionally, UNet is developed from FCN and adopts a symmetric encoder-decoder structure with skip connections between the encoder and the decoder at each resolution level [63]. Due to this architectural change, it alleviates the information loss problem.

Instance segmentation applications have shown success with the Region Convolutional Neural Network (R-CNN) and its derivatives (Fast R-CNN, Faster R-CNN, and Masked R-CNN). The basic stream of R-CNN is to generate each proposal Region of Interest (RoI), extract features from each RoI with CNN, evaluate each RoI with a classifier, and adjust bounding boxes with a regressor [64]. Fast R-CNN accelerates the processing speed by selecting RoI on feature maps instead of input images [65]. By introducing a Region Proposal Network (RPN) to dynamically generate proposal RoI, Faster R-CNN further improves the efficiency [66]. In addition to the existing branches for classification and bounding box regression in Faster R-CNN, Mask R-CNN adds a branch for predicting a segmentation mask [67].

2.3 Object tracking

Image object tracking is to track the movement of objects in time-lapse imaging data. Traditional approaches used for tracking low-speed small objects usually consist of two stages. In the first stage, objects in each image frame of an image sequence are detected individually. When the objects are sparsely distributed and have high contrast to the background pixels, multi-level threshold methods are effective [68]. Although watershed-based methods are useful for dealing with clumped objects, they frequently suffer from the over-segmentation problem [69]. Methods based on the gradient flow are good solutions when image gradient vectors within objects generally point to their centers [70]. In the second stage, detected objects are modeled and linked to recover motion trajectories by various strategies, such as nearest neighbor

[71], meanshift [72], and dynamic programming [73]. As these approaches only utilize the static object information with dynamic information omitted, their performance declines when objects either are overlapped or move at a high speed.

Algorithms based on particle filtering can produce robust tracking results as they contain and update dynamic information in object states [74]. In addition to the commonly used Gaussian distribution model for biomedical images [75, 76], some studies attempt to use parametric active contour models as an alternative to fit objects with more complex shapes [77, 78, 79]. However, active contour models have a large number of parameters and require an exponentially growing number of particles to cover the state space, resulting in a worse computational performance.

2.4 Data augmentation

Image augmentation artificially creates training images by different processing ways and their combinations. It significantly improves the performance of deep networks. Image augmentation is also a critical component in pairwise learning, as it creates views with the same semantics but in different appearances. In addition to such typical image translations as random rotation, shifts, shear, and flips, Mixup trains a neural network by the convex combinations of pairs of three examples and their labels as augmentation [80]. CutMix improves regional dropout using a new strategy where patches are cut and pasted among training images, and the ground truth labels are also mixed proportionally to the area of the patches [81]. RandConv uses a convolution operation with random parameters to create new samples. Intuitively, the randomized convolutions create an infinite number of new domains with similar global shapes but random local textures [82]. Copy-Paste takes advantage of segmentation labels for instance segmentation and randomly pastes objects onto an image to generate new samples [83]. Background-Augmentation generates saliency masks to separate background and foreground objects and includes multiple background operation methods for contrastive learning frameworks [84].

Chapter 3

Object Motion Analysis for Time-Lapse Image Sequences

In this chapter, I present the developed object motion analyzing method and its applications in multiple biomedical image datasets.

The analysis of fluorescence microscopy images has emerged as an effective avenue for a large spectrum of biological and cancer studies. Thanks to modern fluorescence microscopy technologies, a high throughput time-lapse imaging data can be routinely generated to characterize diverse biomedical objects of interest, including cells, vesicles, proteins, and bacteria among others. As numbers of these objects in most biomedical research are large and varying over time, it is infeasible to manually analyze their motion patterns with sufficient accuracy and efficiency. Therefore, development of efficient, accurate, robust, and automated object tracking methods is of great importance to facilitate biomedical investigations.

Compared with two-stage tracking algorithms [71, 72, 73], particle filtering (PF) based algorithms [75, 85, 76] can produce robust tracking results when objects move at a high motion speed. However, due to the complex object shape and limited microscopic image resolution, such object intensities may not always follow the Gaussian

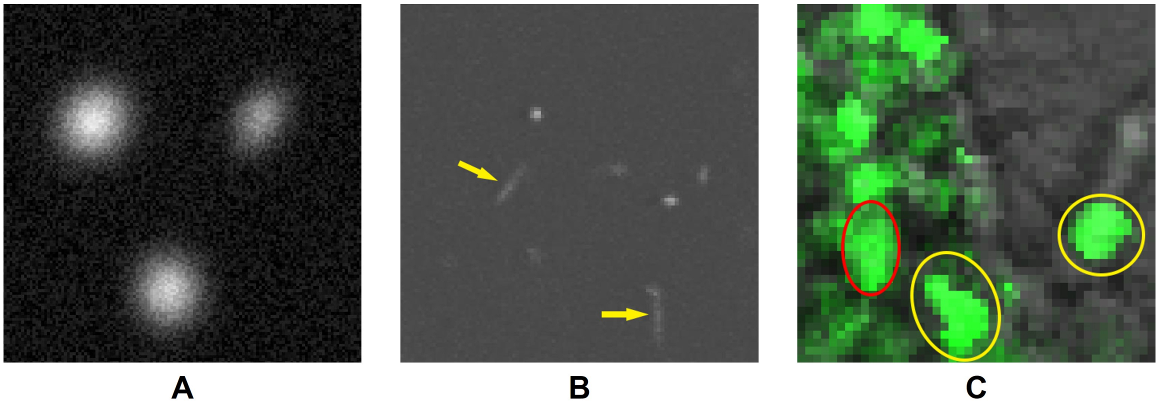

Figure 3.1: Illustration of fluorescent images with (A) object intensity following the Gaussian model, (B) objects with sharp edges (yellow) and shifted center (red), and (C) deformed objects due to motion blur.

distribution. Figure 3.1 illustrates cases when object intensity can and cannot be modeled as a Gaussian distribution. Although some studies [79, 78, 77] overcome such deficiency by leveraging parametric active contours for more precise object state descriptions, they are vulnerable to large shape variations, especially in the 3D space. Additionally, the larger number of parameters necessary for such models inevitably requires an exponentially increasing number of particles (or random guess) to cover the state space, resulting in a worse computational performance.

To address these problems, I generalize the traditional particle filtering approaches in this work. Specifically, two different models are presented to improve the tracking performance in non-Gaussian conditions. In addition, a new tracking management strategy is designed to accelerate the model updating. With this new mapping step after updating the particle states, the tracking accuracy can be improved. Experiments on both artificial and real biomedical time-lapse fluorescence image data for 2D and 3D space demonstrate the robustness and accuracy of the generalized method.

3.1 Method

The tracking approach developed in this work is based on the particle filtering algorithm. In this section, I first briefly recapitulate the particle filtering tracking framework and introduce the object segmentation method used to distinguish objects from background. Next, I present the realization of observation models and dynamics models that are customized for biomedical fluorescent imaging applications. Finally, I explain how the method is extended to multiple objects. As the methods for 2D images can be directly derived from those for 3D images, the description of the developed approach focuses on the 3D case.

3.1.1 Particle tracking framework

Particle filtering algorithm is derived from the Bayesian estimation that infers knowledge about the hidden object state $\boldsymbol{x}_t$ with a sequence of noisy observations $\boldsymbol{z}_{1:t} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t\}$. A recursive formula to estimate the evolution of the hidden state is given in [86]:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t) \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1} \tag{3.1}$$

where $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ is the posterior density function, $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is the state transition model, and $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is the likelihood distribution. The merit of the recursion representation is that is enables real-time processing so that it is not necessary to re-compute previous data if a new observation is generated. With the probability density function $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$, an estimation of the state can be easily computed by such statistical method as expectation and minimum mean squared error (MMSE) estimate.

One problem with such this approach, the optimal solution of Eq 3.1 is only solvable in some rare cases, such as Gaussian or grid-based modeling [87]. For practical applications, particle filtering algorithm is frequently used by a feasible approximation where the desired posterior density function is estimated with N random samples and associated weights $\{\boldsymbol{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \approx \sum_{n=1}^{N} w_t^{(n)} \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^{(n)}) \tag{3.2}$$

These weights are updated and normalized recursively by sequential importance sampling:

$$w_t^n \propto \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^{(n)})p(\boldsymbol{x}_t^{(n)}|\boldsymbol{x}_{t-1}^{(n)})}{q(\boldsymbol{x}_t^{(n)}|\boldsymbol{x}_{t-1}^{(n)}, \boldsymbol{z}_t)} w_{t-1}^{(n)} \tag{3.3}$$

where the importance function $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{z}_t)$ describes the possibility of the distribution of the new state $\boldsymbol{x}_t$ in the state-space. Therefore, the generation of particle $\boldsymbol{x}_t^{(n)}$ follows the importance function.

3.1.2 Object segmentation

Segmentation is an essential step in the tracking analysis, as the initialization and the updating of hidden states $\boldsymbol{x}_t$ require that voxels of each object are accurately assigned with a distinct label. One can either manually choose a threshold or use Otsu algorithm [88] to calculate a data-driven threshold for simple segmentation. Unfortunately, objects in fluorescence microscopy images are often so clumped that it is too challenging to separate them with a single threshold. Therefore, an automated object segmentation method that uses voxel gradient information is applied [70].

The segmentation method is based on the hypothesis that the fluorescent intensity captured by each object of interest declines from its center to its periphery gradually. Thus, the gradient $\boldsymbol{f} = \nabla I = (f_x, f_y, f_z)$ within an object points to the object center. With this property, an image volume can be segmented by assigning the same object label to all voxels pointing to the same object center. However, due to varying image noise, directions of gradient vectors are deteriorated, leading to over-segmentation. In order to obtain biologically meaningful results, the gradient field is regulated by gradient vector flow (GVF) [89], a non-irrotational external force field that does not need any prior knowledge about image edges. The GVF field $\boldsymbol{g} = (u, v, w)$ of a 3D fluorescence image volume $I(x, y, z)$ can computed by solving the following Euler-Lagrange equations:

$$\begin{aligned} \mu\nabla^2 u - (u - I_x)(I_x^2 + I_y^2 + I_z^2) = 0 \\ \mu\nabla^2 v - (v - I_y)(I_x^2 + I_y^2 + I_z^2) = 0 \\ \mu\nabla^2 w - (w - I_z)(I_x^2 + I_y^2 + I_z^2) = 0 \end{aligned} \tag{3.4}$$

where μ is a weight coefficient and ∇^2 is the Laplacian operator. The reason to use GVF algorithm to segment objects in this work is that GVF can be applied directly without training and is resistant to image noise.

After computing the GVF field $\boldsymbol{g}$, voxels are grouped into sub-volumes with distinct object labels by finding paths in the GVF field. Given a voxel $\boldsymbol{r}^{(i)} = (x^{(i)}, y^{(i)}, z^{(i)})^T$, its linked voxel $\boldsymbol{r}^{(i+1)}$ is:

$$\boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)} + S(\boldsymbol{g}(\boldsymbol{r}^{(i)}), \xi) + S(\boldsymbol{g}(\boldsymbol{r}^{(i)}), -\xi) - (1, 1, 1)^T \tag{3.5}$$

where $S(\boldsymbol{g})$ is a vector of step functions:

$$S(\boldsymbol{g}, \xi) = \begin{pmatrix} \varepsilon(u + \xi) \\ \varepsilon(v + \xi) \\ \varepsilon(w + \xi) \end{pmatrix}, \varepsilon(a) = \begin{cases} 1, & a \geq 0 \\ 0, & a < 0 \end{cases}$$

Eq.3.5 suggests that the next voxel to be linked, i.e. $\boldsymbol{r}^{(i+1)}$, can be found by moving forward or backward along the corresponding direction according to the sign of the GVF vector at $\boldsymbol{r}^{(i)}$, i.e. $\boldsymbol{g}(\boldsymbol{r}^{(i)}) = (u(\boldsymbol{r}^{(i)}), v(\boldsymbol{r}^{(i)}), w(\boldsymbol{r}^{(i)}))$, when the absolute value of at least one GVF field component is greater than or equal to ξ, i.e. $|u| \geq \xi$ or $|v| \geq \xi$

or $|w| \geq \xi$. Otherwise, such linking process is terminated. Thus, the parameter ξ controls the speed of linking process. This linking process is repeated until all voxels are connected to some center voxels. Further, I assign the same but unique label to all voxels connected with the same center voxel and consider the space by all voxels sharing the same label a distinct sub-volume. By this approach, background voxels would be linked to some center voxel for each sub-volume. To remove such background voxels, Otsu algorithm [88] is used to compute a global threshold and local thresholds for each sub-volume. Voxels with intensity either lower than the global threshold or the corresponding local threshold would be labeled as zero, i.e. the label for background.

3.1.3 Observation and dynamics models

To apply the particle filtering algorithm to time-lapse fluorescence microscopy images, the observations are time series of gray-scale images of size $A \times B \times C$. Thus, observation $\boldsymbol{z}_t = \{z_t(i,j,k) \mid i \in [1,A], j \in [1,B], k \in [1,C]\}$ is interpreted as the voxel intensity at location (i,j,k) and time t, while the state vector $\boldsymbol{x}_t$ characterizes a vector of status properties of an object of interest at time t. As shown in Eq. 3.3, particle filtering algorithm also requires computation of the likelihood function $p(\boldsymbol{z}_t|\boldsymbol{x}_t^{(n)})$ that assesses the appropriateness of particle $\boldsymbol{x}_t^{(n)}$ for a given observation $\boldsymbol{z}_t$, and the transition prior $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ that describes the state evolution used for particle generation. A common state space vector for fluorescence microscopy image is $\boldsymbol{x}_t = (\boldsymbol{r}_t, \boldsymbol{v}_t, \boldsymbol{s}_t, I_t)$, where $\boldsymbol{r}_t = (x_t, y_t, z_t)$, $\boldsymbol{v}_t = (\dot{x}_t, \dot{y}_t, \dot{z}_t)$, $\boldsymbol{s}_t = (\sigma_{max,t}, \sigma_{min,t}, \sigma_{z,t}, \theta_t)$, and I_t denote the spatial position, velocity, shape, and object intensity, respectively [74, 85]. This model assumes that the contribution of the state $\boldsymbol{x}_t$ to the observation intensity can be well approximated by a Gaussian function, thus:

$$h_t(i,j,k;\boldsymbol{x}_t) = I_t \exp(-\frac{1}{2}\boldsymbol{m}^T\boldsymbol{R}^T\Sigma^{-1}\boldsymbol{R}\boldsymbol{m}) + b_t \tag{3.6}$$

where b_t denotes the estimated background intensity; $\Sigma = \text{diag}(\sigma^2_{max,t}, \sigma^2_{min,t}, \sigma^2_{z,t})$ is the covariance matrix; $\boldsymbol{R} = \boldsymbol{R}(\theta_t)$ is the rotation matrix on the x-y plane, and $\boldsymbol{m}^T = (i - x_t, j - y_t, k - z_t)$. The likelihood function can be defined in multiple ways, including Sum of Absolute Difference (SAD) [74], Normalized Cross Correlation (NCC) [71], or other intensity-based similarity metrics.

However, due to the aggregated effect from diverse factors in the imaging acquisition process, objects of interest in fluorescence microscopy images dot not always fit the Gaussian model that assumes the voxel intensity varies smoothly and reaches the peak at the object center. To address this issue, two models are designed for non-Gaussian cases.

(1) Ellipsoid Model

Figure3.1B illustrates the transition between object foreground and background could be abrupt and brighter voxels could deviate from the object center in fluorescence microscopy image data of real biomedical research. For such cases, a more appropriate ellipsoid model is designed to fit 3D object voxel intensity distribution. By this model, the volumes of objects from gray-scale image $\boldsymbol{z}_t$ are extracted with the segmentation method described in Section 3.1.2. The resulting object volumes are denoted as $G(\boldsymbol{z}_t)$. Next, each object volume $g_t \in G(\boldsymbol{z}_t)$ is fitted by an ellipsoid $\mathcal{E}_t$ in a way such that the overlap between volume g_t and the ellipsoid is maximized. Since most objects are noticed to have a small range along the z direction in a large number of biomedical applications, the elevation angle is ignored. Thus, the ellipsoid $\mathcal{E}_t$ has two axes $\sigma_{max,t}, \sigma_{min,t}$ parallel to the x-y plane, and the third axis $\sigma_{z,t}$ perpendicular to the x-y plane. With the ellipsoid $\mathcal{E}_t$, I define the state vector $\boldsymbol{x}_t = (\boldsymbol{r}_t, \boldsymbol{v}_t, \boldsymbol{s}_t)$ where $\boldsymbol{r}_t = (x_t, y_t, z_t)$, $\boldsymbol{v}_t = (\dot{x}_t, \dot{y}_t, \dot{z}_t)$, and $\boldsymbol{s}_t$ represent the ellipsoid centroid, velocity, and the shape vector, respectively. For shape characterization, $\boldsymbol{s}_t$ includes the half principal axis length $\sigma_{max,t}, \sigma_{min,t}, \sigma_{z,t}$ and the rotation angle θ_t around the z-axis.

The likelihood function computes the degree of overlap between the state vector specified volume and the segmented object volume. With the ellipsoid model, the likelihood function is defined as:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^{(n)}) = \max_{g_t \in G(z_t)} \frac{\left| g_t \cap E\left(\boldsymbol{x}_t^{(n)}\right) \right|}{\left| g_t \cup E\left(\boldsymbol{x}_t^{(n)}\right) \right|} \tag{3.7}$$

where $E(\boldsymbol{x}_t^{(n)}) = \{e(i,j,k;\boldsymbol{x}_t^{(n)})\}$ represents a 3D volume with an ellipsoid mask specified by the state vector $\boldsymbol{x}_t^{(n)}$. Additionally, the voxel value $e(i,j,k;\boldsymbol{x}_t^{(n)})$ can be either 0 or 1 determined by the formula modified from Eq.3.6:

$$e(i,j,k;\boldsymbol{x}_t^{(n)}) = \epsilon(\boldsymbol{m}^T \boldsymbol{R}^T \Sigma^{-1} \boldsymbol{R}\boldsymbol{m} - 1), \tag{3.8}$$

where $\epsilon(\cdot)$ is the unit step function.

Meanwhile, I assume that changes in object motion and shape are independent. Thus, the transition prior can be factorized as:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}), \tag{3.9}$$

where the motion vector $\boldsymbol{y}_t = (x_t, \dot{x}_t, y_t, \dot{y}_t, z_t, \dot{z}_t)$.

Further, the transition prior for the motion vector is given by:

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{P}\boldsymbol{y}_{t-1}, \boldsymbol{q}_1) \tag{3.10}$$

where $\mathcal{N}(\mu, \Sigma)$ is the normal distribution with mean μ and covariance matrix Σ. The process transition matrix $\boldsymbol{P}$ is defined as following:

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{P}' & 0 & 0 \\ 0 & \boldsymbol{P}' & 0 \\ 0 & 0 & \boldsymbol{P}' \end{pmatrix}, \quad \boldsymbol{P}' = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Similarly, the transition prior for the shape vector is given by another normal distribution:

$$p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) = \mathcal{N}(\boldsymbol{s}_{t-1}, \boldsymbol{q}_2) \tag{3.11}$$

In Eq. 3.10 and 3.11, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ represent the noise level for motion and shape, respectively. Note both parameters can be tuned during experiments.

(2) Voxel-Based Model

Although the ellipsoid model can be used to characterize objects with non-Gaussian intensity distribution, it still assumes objects are approximately ellipsoidal by shape. As presented in Figure 3.1C, objects from images of real biomedical studies can be elongated in shape due to motion blur. To enable particle filtering algorithm for such cases, a Voxel-Based (VB) model is designed to accommodate such shape aberrations. Instead of using a shape vector for shape representation, VB model records voxel coordinates of an object associated with a particle based on its state vector. The resulting state vector is defined as $\boldsymbol{x}_t = (\boldsymbol{r}_t, \boldsymbol{v}_t, C_t) = (\boldsymbol{y}_t, C_t)$ where C_t denotes the object voxel coordinate set. Therefore, the likelihood function is given as:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^{(n)}) = \max_{g_t \in G(z_t)} \frac{\left|g_t \cap C_t^{(n)}\right|}{\left|g_t \cup C_t^{(n)}\right|}. \tag{3.12}$$

Note that I only consider the change of motion vector $\boldsymbol{y}_t$, and spatial shift of the coordinate set C_t for the transition prior computation. Thus, spatial coordinate set cab be updated by the following equation:

$$C_t = C_{t-1} + \boldsymbol{r}_t - \boldsymbol{r}_{t-1} \tag{3.13}$$

Therefore, unlike the factorization Eq.3.9 in the ellipsoid model, the transition

Algorithm 1 Multiple object tracking management framework

```
Input: time lapse image volumes $\{\boldsymbol{z}_t\},\ t \in [1, T]$
Output: object state sets $\{\boldsymbol{x}_{t,k} |\ k \in [1, M_t],\ t \in [1, T]\}$
Initialize state set $\{\boldsymbol{x}_{1,k}\}$ with $\boldsymbol{z}_1$
for $t = 2 : T$ do
    Extract states $\{\boldsymbol{x}_{t,k}\}$ from $\boldsymbol{z}_t$ and set their labels to 0
    for $j = 1 : M_{t-1}$ do
        Generate particles $\{\boldsymbol{x}_{t,j}^{(n)}\}$ according to the state $\boldsymbol{x}_{t-1,j}$
        Compute weight $\pi_{t,j}^n$ for each particle $\boldsymbol{x}_{t,j}^{(n)}$
        Normalize weights such that $\sum_{n=1}^N \pi_{t,j}^n = 1$
        Compute the estimated state $\boldsymbol{x}'_{t,j}$
    end for
    Map labels of estimated states $\{\boldsymbol{x}'_{t,j}\}$ to detected states $\{\boldsymbol{x}_{t,k}\}$
end for
```

prior in this case is simplified as:

$$p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) = p(C_t | C_{t-1}) = p(\boldsymbol{y}_t | \boldsymbol{y}_{t-1}) \tag{3.14}$$

As the developed VB model does not take into account shape information, I next present a tracking management strategy developed to replace the shape information update missing in the VB model.

3.1.4 Multiple object tracking management

I have developed an automatic tracking management process for multiple object tracking problems. This strategy includes four steps: initialization, prediction, updating, and mapping. The complete workflow is illustrated in a diagram in Figure 3.2. Note that all steps but initialization, can be executed recurrently, thus leading to a reduced computational complexity. An algorithmic description of the developed approach is presented in Algorithm 1. In addition, I provide step details as follows.

(1) **Initialization**: In this step, global parameters are initialized: particle number N, noise levels $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. A large N in general results in a higher tracking

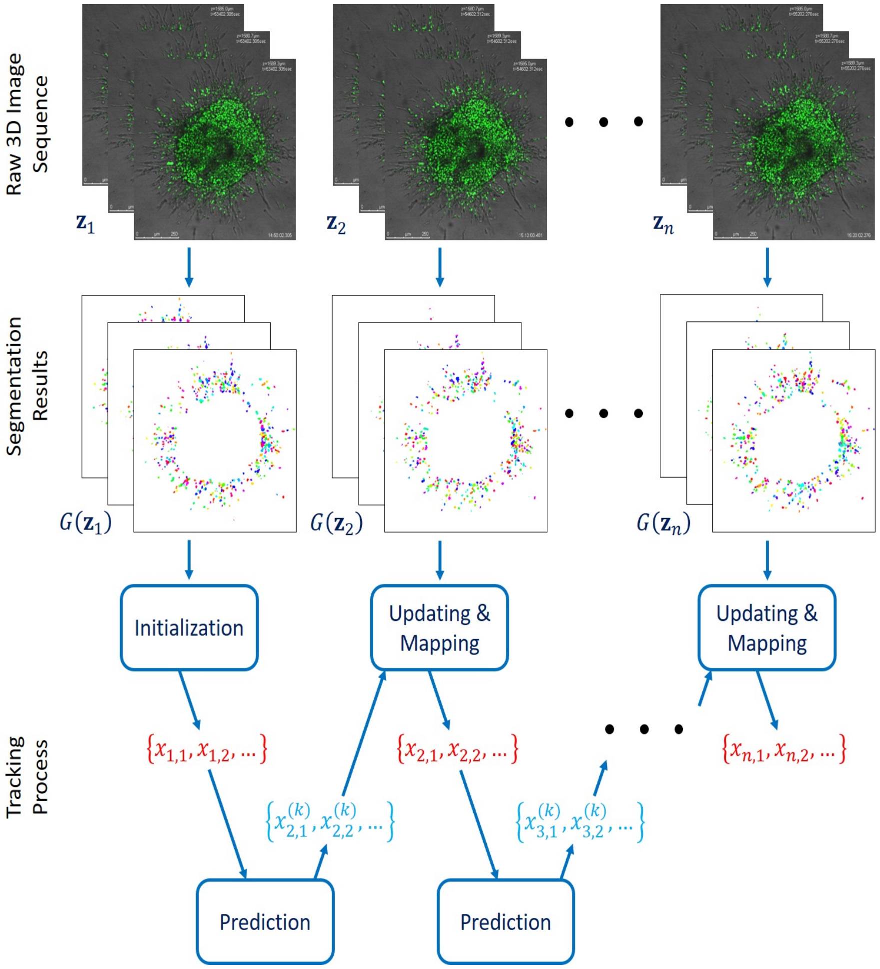

Figure 3.2: Overall schema for multiple object tracking management method. Raw 3D images (first row) are first processed to assign each object of interest a unique label (second row). Labels are further modified by the tracking process (remaining rows) such that each object of interest retains the same unique label in temporal imaging data. For tacking process, the estimated states are represented in red texts, while produced particles are in blue.

precision but at a higher computational time cost. Additionally, a higher noise level helps tracking drastic changes in object states, but decreases the resistance to interference when object density is high. In this step, the time lapse image data $\boldsymbol{z}_t$ is segmented as a temporal volume set $\{G(\boldsymbol{z}_t)\}$ by the gradient-based algorithm presented in Sect. 3.1.2. I denote the number of volume in $G(\boldsymbol{z}_t)$ as $M_t = |G(\boldsymbol{z}_t)|$. Therefore, states $\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,2}, \ldots, \boldsymbol{x}_{1,M_1}$ are extracted from the first image volume when $t = 1$. In addition to the basic information of an object, i.e. location, speed, intensity among others, a state vector $\boldsymbol{x}_{t,k}$ contains an object label for identification of the same object traversing image volumes at different time points.

(2) **Prediction**: In the prediction step, particles $\{\boldsymbol{x}_{t,k}^{(n)} \mid n \in [1, N],\ k \in [1, M_{t-1}]\}$ are generated according to Eq. 3.9, Eq.3.10, and Eq.3.11 when the ellipsoid model is used. When VB model is adopted, Eq. 3.10, Eq.3.13, and Eq.3.14 are used for particle generation instead. As each state has N particles, the total number of particles in each iteration is $N \times M_{t-1}$.

(3) **Updating**: The likelihood of each particle is updated by Eq. 3.7 and Eq. 3.12 when the ellipsoid model and VB model are used respectively. I use the likelihoods as particle weights $\pi_{t,k}^{n}$ and normalize them with $\sum_{n=1}^{N} \pi_{t,k}^{n} = 1$. Therefore, the estimated state is computed by:

$$\boldsymbol{x}'_{t,k} = \sum_{n=1}^{N} \pi_{t,k}^{n} \boldsymbol{x}_{t,k}^{(n)}$$

(4) **Mapping**: Finally, the relationship between estimated state $\boldsymbol{x}'_{t,k}$ and detected state $\boldsymbol{x}_{t,k}$ is characterized by the likelihood function. For each estimated state $\boldsymbol{x}'_{t,j}$, I compute its likelihood with all detected states $\{\boldsymbol{x}_{t,k}\}$ and assign the label of the j-th state in frame $t-1$ to the state with the highest likelihood in frame t. Note when the highest likelihood is less than a specified threshold D, such labeling process does not occur. Objects without any matched estimated state are treated as disappearing objects, while the ones without any matched detected state are treated as newly emerged objects with unused labels assigned.

3.2 Results

To validate and assess the developed tracking method performance, I apply the complete and automated workflow to multiple time-lapse fluorescence microscopy image data sets, including one artificial dataset with known ground truth, as well as real biological image data sets from two time-lapse microscopy studies on bacteria motility

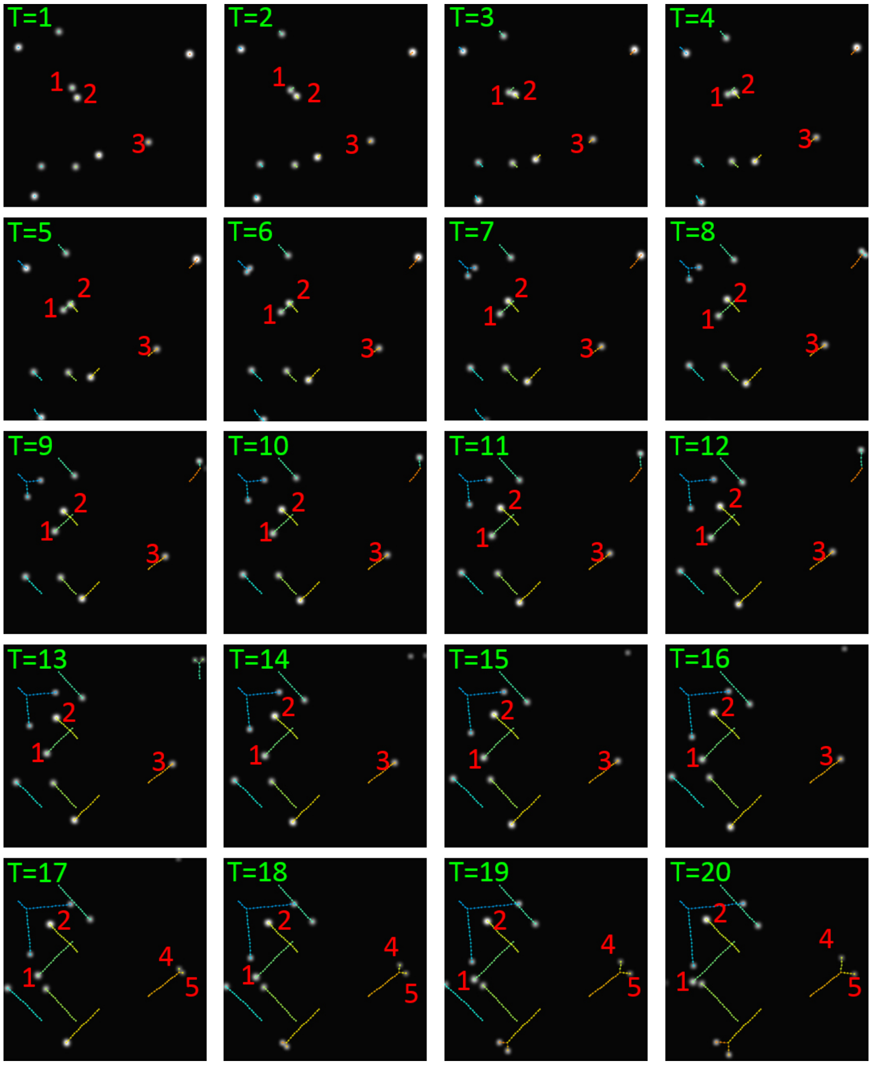

Figure 3.3: Experimental results of artificial data with left to right and top to bottom time order. Trajectories are color coded and overlaid on original images. Note that object 1 and 2 are overlapped in frame 3 and 4. Additionally, object 3 is split into two child objects, i.e. object 4 and 5, from frame 17. In both cases, the developed method can track objects correctly.

and 3D lung cancer spheroid analysis.

3.2.1 Validation with artificial data

My developed tracking approach is first tested and validated with a synthetic 2D image data set with each image of 1000×1000 pixels in size. This data set is generated by artificially initializing object states, updating states with the Gaussian model, and producing individual temporal image frames with evolving object states and noise background. In the data set, 10 objects are pruduced initially with speed subject to a uniform distribution between 8 and 11 pixels per frame. Each object can split into two child objects with probability 0.02 in each frame. Parameters of the approach with the ellipsoid model are fixed with the following values: $N = 200$ for each object, $\boldsymbol{q}_1 = (10, 1, 10, 1)$, and $\boldsymbol{q}_2 = (0.5, 0.5, 0.2)$. As objects in synthetic images are sparse and move relatively slow, almost all objects in all frames are correctly tracked in reference to their ground truth, with the overall tracking accuracy 99.7%. Additionally, the Root Mean Square Error (RMSE), a frequently used metric computed with the ground truth and estimated object positions, is 1.91 ± 0.32 for those correctly tracked objects. Figure 3.3 demonstrates typical tracking results where motion trajectories of objects are illustrated. When objects are split, both object and its parent trajectories are represented by forked chains in one color. Note that in Figure 3.3 both motion crossed and proliferated objects are correctly tracked. For example, object 1 and 2 are partially overlapped in frame 3 and 4. However, the developed method manages to track them after their collision. Additionally, object 3 is an example where one object is proliferated into two child objects, i.e. object 4 and 5, from frame 17. The trajectories of the resulting two child objects are linked to the trajectory of object 3, clearly demonstrating their pedigreed relationship. All result above suggest the effectiveness, robustness, and positioning accuracy of the developed approach in simple cases where objects are sparse and present a good contrast to the background.

3.2.2 Bacteria motility analysis

I further test the developed method with a real time-lapse 2D fluorescence image data set on the motility of the bacterial pathogen *Vibrio cholerae* after treatment with bacteria-specific motility inhibiting monoclonal antibodies. In this experiment design, the inhibition of bacteria average speed is considered as the metric of bacteria vitality.

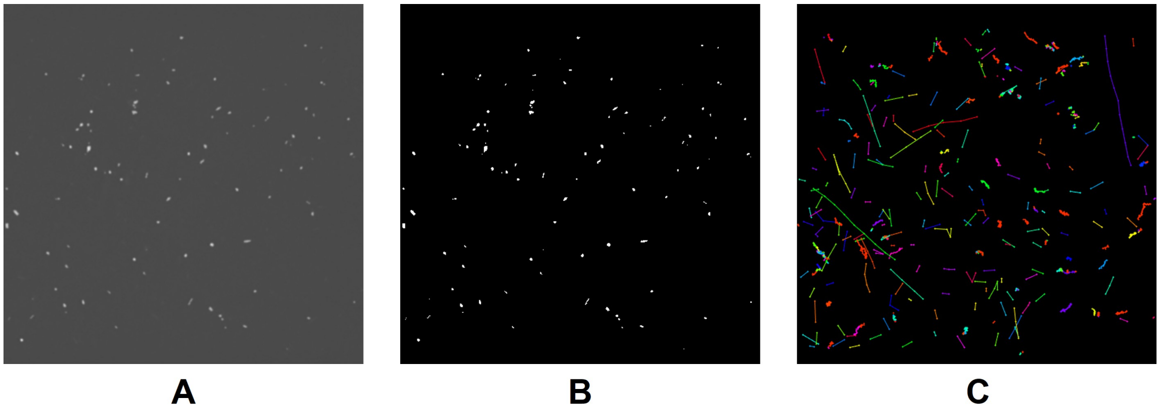

Figure 3.4: (A) A typical raw image frame from a data set for bacteria motility study; (B) Bacteria segmentation results; (C) Tracking result demonstrated with all bacteria trajectories overlaid on the image.

This is inversely proportional to the potential protective efficacy of cholera vaccines that aim, in part, to induce antibodies able to inhibit bacterial motility. With a high-speed confocal microscope, motility of bacteria is observed at 100ms intervals for five seconds, with five minutes post treatment on six types of antibodies in various dose concentrations. The resulting data set consists of 23 image sequences. Each includes 50 temporal image frames of 512×512 pixels in image resolution. Figure 3.4A presents an image frame of a typical image sequence that captures both active bacteria with high speeds and in elongated shape, as well as slow-moving bacteria with vitality significantly reduced by vaccine. I have applied the developed tracking method with the VB model to bacteria image sequences with following empirical parameter setup: $N = 200$ per object, and $q_1 = 2$. Figure 3.4B presents the bacteria segmentation result of Figure 3.4A. In Figure 3.4C, bacteria tracking results are illustrated. Specifically, the motion trajectories of bacteria are plotted in colors. For each bacterium, its trajectory is visualized from the frame of its occurrence to the current example frame shown in Figure 3.4A. Additionally, I present the dynamic tracking results frame by frame in Figure 3.5 where each bacterium tracking result over eight temporal frames is illustrated. In particular, I demonstrate one specific bacterium motion tracking

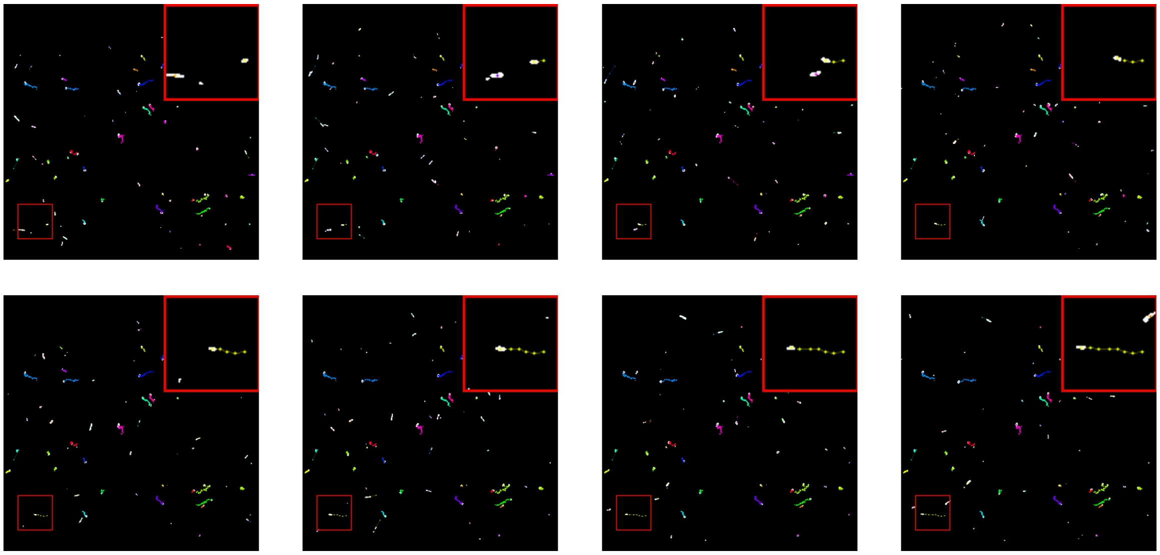

Figure 3.5: Illustration of the tracking result dynamics of a 2D bacteria motility dataset. With the image inset, one typical bacterium from a local region is enlarged for its trajectory demonstration in detail.

trajectory in an inset.

I quantitatively assess the developed track method by comparisons of manually annotated and machine produced bacteria trajectories from a validation set. The validation set includes 230 bacteria randomly selected from random image sequences at each time point. Two metrics, i.e. precision and recall, are used to evaluate the tracking performance: Recall $= \frac{L_1}{L_3}$, Precision $= \frac{L_1}{L_2}$, where L_1, L_2, and L_3 represent the number of frames with correct tracking results, total number of frames tracked by the developed method, and the number of frames by human annotations, respectively.

To assess the performance of the developed method, I apply the classical particle filtering method [75], the particle filtering method improved by data-dependent importance sampling [85], and the developed method with two developed models to bacteria image sequences. The particle numbers for all methods are set to $N = 200$ per object. In the developed models, I empirically set $\boldsymbol{q}_1 = (2, 0.5, 2, 0.5)$ and $\boldsymbol{q}_2 = (0.2, 0.2, 0.2)$. The validation set includes 50 low and 50 high speed bacteria randomly selected from image sequences at each time point. The cutoff value between low and high speed

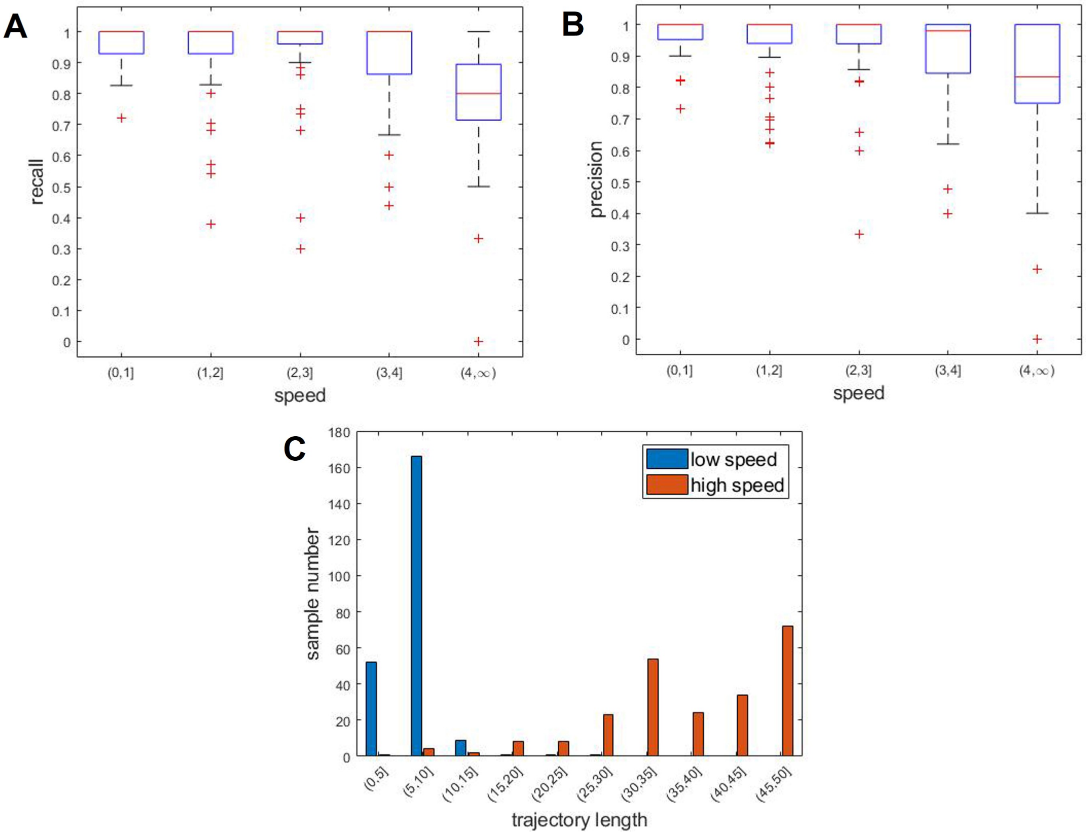

Figure 3.6: (A) Tracking Recall grouped by bacteria motion speed; (B) Tracking Precision grouped by bacteria motion speed; (C) Distribution of bacteria trajectory length for low and high speed bacteria populations with speed threshold of 3 pixels per frame.

populations is three pixels per frame. I compute the trajectory Precision and Recall for bacteria populations with low and high speed and present tracking results from different methods in Table 3.1. For the low speed bacteria population, all methods present good performance with minor difference. However, the developed method, especially with the VB model, is superior to other two state-of-the-art methods for tracking the high speed population.

To better understand behaviors of the developed method, I further analyze tracking Recall and Precision of 460 randomly selected individual bacteria using the developed VB model. As shown in Figure 3.6A and Figure 3.6B, it is noticeable these performance metrics decrease as the bacteria motion speed increases. In the mean-

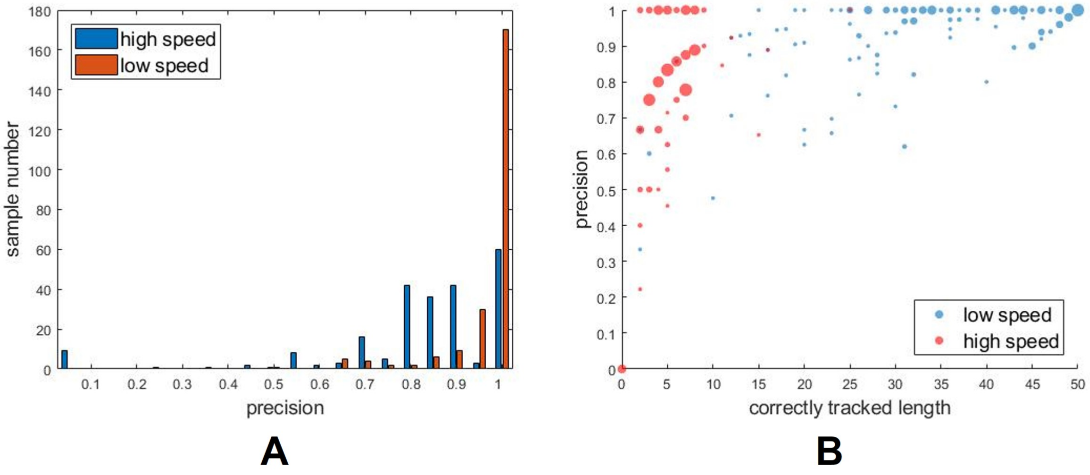

Figure 3.7: Comparison of trajectory Recall for low and high speed bacteria populations in the 2D validation set. (A) Distribution of trajectory Recall; (B) Scatter plot of trajectory Recall. Note the unit of trajectory length is frame number, and size of each dot represents number of bacteria samples.

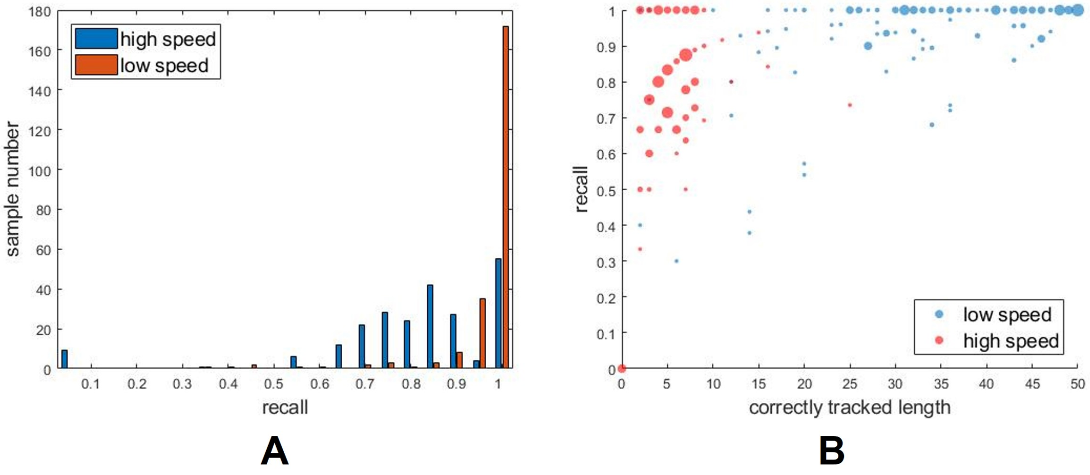

Figure 3.8: Comparison of trajectory Precision for low and high speed bacteria populations in the 2D validation set. (A) Distribution of trajectory Precision; (B) Scatter plot of trajectory Precision. Note the unit of trajectory length is frame number, and size of each dot represents number of bacteria samples.

while, I investigate the trajectory length distribution for high and low speed bacteria with the speed cutoff value of three pixels per frame. Note that in Figure 3.6C most

Table 3.1: Comparison of trajectory Precision and Recall for bacteria of different motion speeds.

Population	Method	Precision		Recall	
		mean	std	mean	std
High speed	Classical [75]	0.6338	0.1415	0.7393	0.1699
	Improved [85]	0.7327	0.1577	0.7410	0.1738
	Ellipsoid model	0.7768	0.1904	0.7586	0.2173
	VB model	0.8032	0.1679	0.7970	0.1698
Low speed	Classical [75]	0.9450	0.1177	0.9647	0.0626
	Improved [85]	0.9544	0.0784	0.9670	0.0596
	Ellipsoid model	0.9591	0.1076	0.9615	0.0961
	VB model	0.9594	0.0575	0.9562	0.0964

high speed bacteria are captured in less than 10 image frames, with an average of 6.6 frames. These analyses explain why the trajectory Recall and Precision of most high speed bacteria tend to be substantially deteriorated even if there is only one erroneously tracked image frame. Additionally, the comparisons between low and high speed bacteria populations by Recall and Precision are presented in Figure 3.7, and Figure 3.8, respectively. Note that Recall and Precision of low speed bacteria are mostly larger than 0.9, while these two metrics of high speed bacteria are mostly larger than 0.66.

3.2.3 Tumor spheroid study

For further method validation, I test the developed method with 3D time-lapse imaging data from an in vitro experiment investigating 3D spheroid invasion in the 4T1 mouse mammary carcinoma cell line. In vitro spheroids are formed by centrifuging 3000 4T1 murine cancer cells in a round bottom, ultra-low attachment 96-well plate (Corning). After 72 hours, compacted spheroids are collected and embedded in 3.0 mg/ml rat tail collagen type-I (Corning) in a μ-Slide 8 well chamber slide (Ibidi). Images are taken every 10 minutes for 16 hours post-embedding using a Leica SP8 confocal microscope at 10X magnification. Time-lapse data show that these cancer cells invade with different behaviors, i.e. either collectively (in chain-like cellular protrusions) or individually. Cancer cells moving in collective chains are termed "in-chain" cells, while the invading cells not in chains are termed "single" cells. Through quantitative analyses, distinct moving patterns for each cancer cell population are supposed to be characterized.

The data set for analysis in this study consists of four longitudinal 3D image sequences acquired at 93 time points. Each image volume at one time point includes 24-32 image planes of 512×512 pixels in resolution. Figure 3.9A presents a x-y slice of a typical RGB-model fluorescent image volume at a time point. For computation convenience, each RGB 3-channel image is transformed to a gray-scale image. As cells in this study are captured by fluorescent signals from the green channel, I first extract the green channel from the original images and use it as the derived gray-scale image. Note that in Figure 3.9B the background surrounding the central spheroid is noisy and has similar intensity values to that of cancer cells, resulting in poor algorithm performance. To address this problem, a feasible mapping formula is used to effectively distinguish cells from background: $I_{gray} = -0.5r + g - 0.5b$, where r, g, and b represent red, green, and blue channel, respectively. The enhanced image is demonstrated in Figure 3.9C.

In this study, I are only interested in tracking and characterizing cells invading out of the central core of the spheroid. Thus, a mask is implemented to remove cells of non-interest within the core of the spheroid before the tracking process, which enormously increases the computation speed. For automatic identification of such mask, the resulting gray-scale image is converted to a binary image with thresholding (Figure 3.9D), followed by morphological algorithms to fill holes, remove outliers (Figure 3.9E) and smooth the resulting mask contour (Figure 3.9F). Figure 3.9G presents the rough cell segmentation result of Figure 3.9C where each color coded block contains a cell of interest and its surrounding background. Figure 3.9H presents

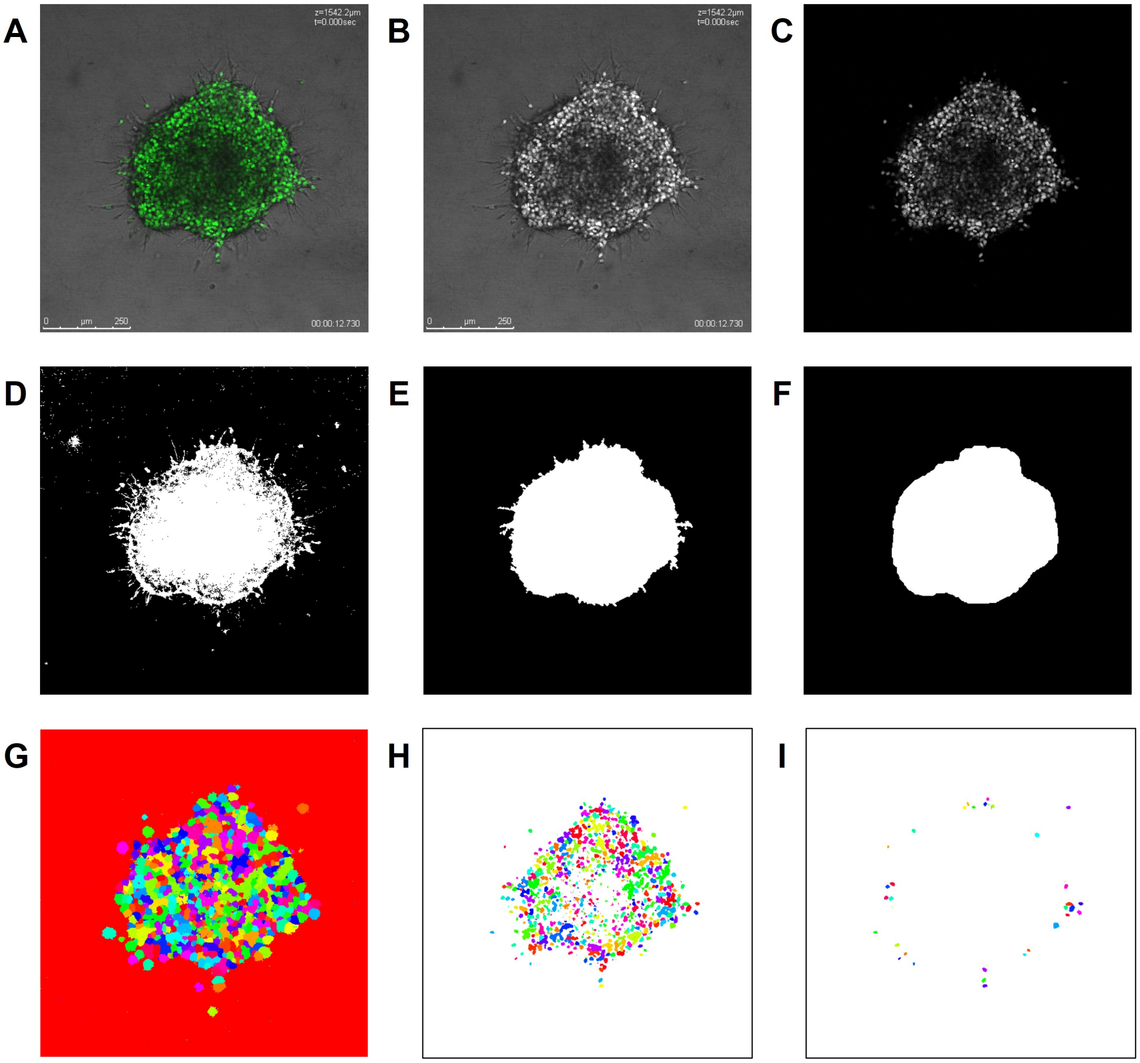

Figure 3.9: 2D x-y cross section views of preprocessing and segmentation results for a 3D tumor spheroid dataset. (A) Original image; (B) Gray-scale green channel; (C) Improved gray-scale image by mapping in the RGB color space; (D) Binary image after thresholding; (E) Binary mask with holes filled and outliers removed; (F) Final mask after mask contour regulation; (G) Rough segmentation result; (H) Refined segmentation result with a global and a local Otsu threshold applied to each block; (H) Final segmentation result with removed mask-specified spheroid.

the refined segmentation result after a global and a local Otsu thresholding are applied to each block. Next, with the spheroid mask illustrated in Figure 3.9F, the volume of the core of the spheroid is removed from tracking analysis. The final segmentation result is illustrated in Figure 3.9I. As this data set captures cancer cells in 3D, I present

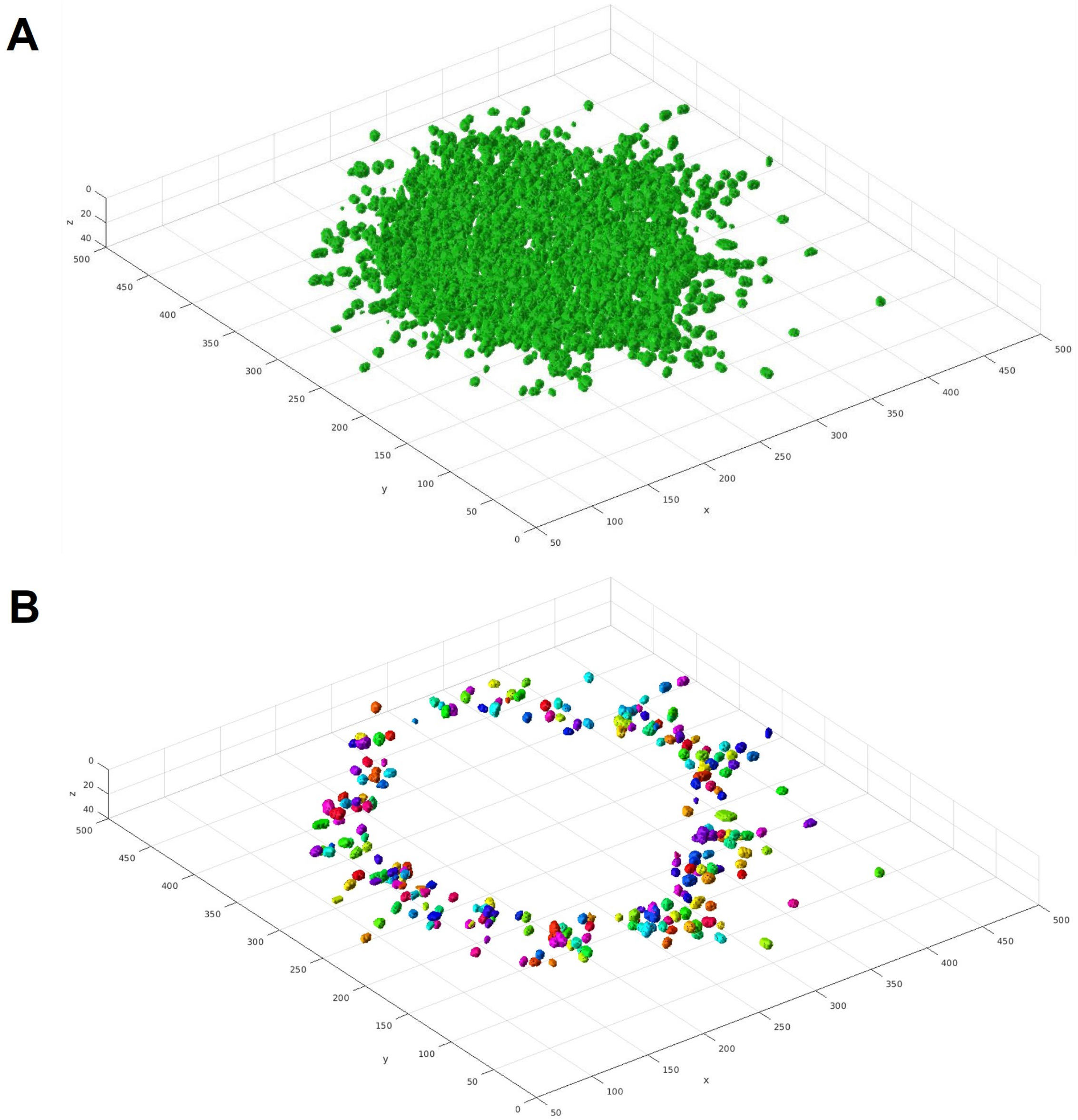

Figure 3.10: (A) 3D view of the preprocessed tumor spheroid image volume; (B) 3D segmentation result of the 3D image volume.

a 3D view of a typical 3D image volume before and after the segmentation analysis in Figure 3.10A and Figure 3.10B, respectively. After segmentation, all cells of interest are uniquely labeled at each time point. Next, the developed tracking method with the ellipsoid model is applied to the time-lapse 3D labeled imaging volumes with the following empirical parameter setup: $N = 500$ per object, $\boldsymbol{q}_1 = (5, 1, 5, 1, 1, 1)$, and

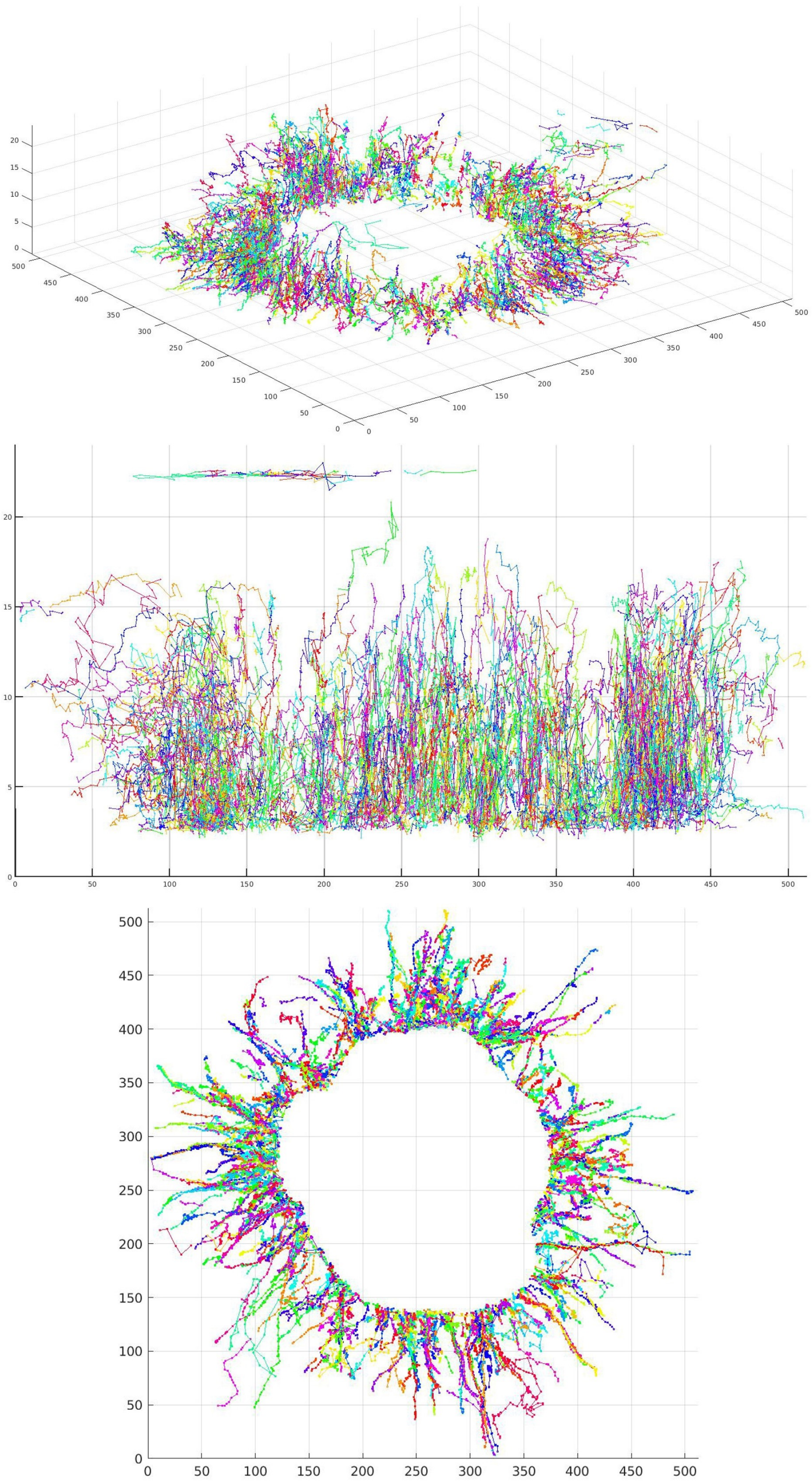

Figure 3.11: Cancer cell tracking results of a 3D tumor spheroid dataset presented in three different views. (Top) 3D view; (Middle) y-z view; (Bottom) x-y view.

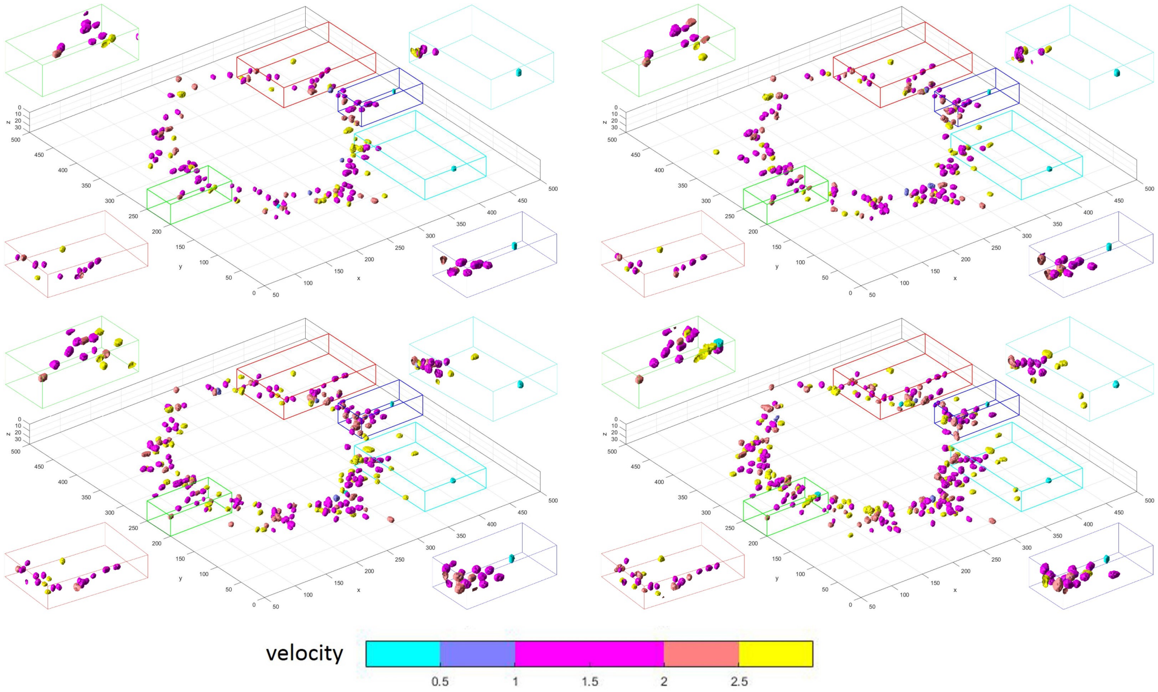

Figure 3.12: 3D view of tracking results with 3D cancer cells color coded by motion speed. The frame interval for visualization is five for enhanced motion effect. The figure insets present enlarged details sub-volumes capturing representative "in-chain" and "single" cells.

$\boldsymbol{q}_2 = (2, 2, 0.5, 0.2)$. Figure 3.11 demonstrates 3D views of the tracking result of a longitudinal image data where all cell trajectories are recovered.

To assess the tracking performance, I generate the validation set with 200 cells of interest by the same strategy as Section 3.2.2. After manually checking trajectories of "in-chain" cells and "single" cells shown in Figure 3.11, Note that, in general, the former population moves radially outward from the spheroid center to peripheral areas at a moderate speed, while the latter group moves either at a near to zero speed or at a relatively high speed with frequently varying directions, leading to zig-zag trajectories. Typical examples of "single" cells, typically with z-axis value $z > 20$, can be clearly observed from the y-z view in Figure 3.11. Additionally, Figure 3.12 visualizes cells of interest with color codes representing motion speed. From this figure, it is salient that spatial invasion patterns of "in-chain" and "single" cells are different. Insets of

Table 3.2: Comparison of trajectory Precision and Recall for cancer cells of different populations.

Population	Method	Precision		Recall	
		mean	std	mean	std
Single	Classical [75]	0.9014	0.1114	0.8973	0.1011
	Improved [85]	0.9190	0.0909	0.9135	0.0923
	Ellipsoid model	0.9342	0.0794	0.9097	0.0969
In-Chain	Classical [75]	0.9364	0.0596	0.9519	0.0347
	Improved [85]	0.9388	0.0573	0.9592	0.0402
	Ellipsoid model	0.9578	0.0451	0.9584	0.0314

each 3D plot enlarge subvolumes that capture representative cell chains color coded in purple for moderate speed. These subvolumes also capture some representative "single" cells in cyan and yellow, suggesting low and high motion speed, respectively. Table 3.2 presents quantitative evaluation of tracking quality with the metrics defined in Section 3.2.2. Note that both Precision and Recall for cells moving at different speed levels are promising, suggesting the efficacy of the developed approach for cell invasion automatic tracking and quantitative motion pattern characterization in cancer research.

3.3 Summary

In this work, I extend the particle filtering approach by developing non-Gaussian models and the corresponding tracking management strategy. With a gradient-based segmentation algorithm, objects in image sequences are extracted and modeled by states. The evolution of these states can be used to recover object motion trajectories and quantitatively characterize object motion behaviors. Experiments on both artificial and real biomedical time-lapse fluorescence image data for 2D and 3D space demonstrate the robustness and accuracy of the generalized approach.

Chapter 4

Biomedical Image Segmentation with Supervised Learning

In this chapter, I present the developed supervised semantic segmentation method and its application in detection of liver portal tract regions and diagnosis of liver fibrosis stage.

Detection of early-stage fibrosis in transplant liver biopsies is important for predicting disease progression and guiding medical management [90]. Known as a strong predictor of liver disease progression and mortality, liver fibrosis can be captured by multiple non-invasive medical imaging techniques, such as computed tomography (CT), magnetic resonance elastography (MRE), and transient elastography (TE) [91]. For accurate liver fibrosis staging, however, the histopathologic examination of liver biopsy samples remains the "gold standard" for liver fibrosis assessment [90]. Although numerous histopathological staging systems have been utilized for liver fibrosis evaluation in current clinical practice, including Knodell, Metavir, Ishak, and Scheuer systems, only manual reviews or semi-quantitative evaluations are conducted by these staging systems, resulting in large inter- and intra-observer variability [92, 93, 94].

To reduce such variations, evaluation methods based on machine learning based algorithms, such as random forests, K-nearest neighbors, and support vector machines, have been developed to provide objective diagnostic tools for liver fibrosis staging [95, 96, 97]. In contrast to these conventional machine learning methods, deep learning has emerged as a powerful tool for diverse biomedical image processing studies due to its great success across different image modalities [98]. Unlike the traditional machine learning methods, deep learning methods require no manual feature engineering and can support multiple imaging modalities for liver fibrosis diagnosis, including CT [99, 100], MRI [91], and ultrasonography images [101, 102]. The resulting image features and other clinical demographic information (e.g., gender and age) can be leveraged for an integrated prediction analysis by multiple fully connected layers attached to the convolutional neural network backbone.

However, few studies have been carried out for deep learning based fibrosis analysis with the "gold standard", i.e., liver biopsy histopathology whole-slide images (WSIs). Although a study used a pre-trained AlexNet [58] to predict the liver fibrosis stage, its input images were acquired from second-harmonic generation microscopy [94]. A modified UNet architecture was also utilized to detect portal tract regions in mouse liver biopsy histopathology WSIs, but no comparison experimental result was given [103]. In the prior work [90], researchers have manually delineated portal tract regions in liver biopsy images and demonstrated that the resulting quantitative portal tract fibrotic percentage and average portal tract area of portal tract regions are correlated with the liver fibrosis stage made by domain experts. However, such results are subject to intra- and inter-observer variability due to the manual annotation process [92]. Therefore, the development of fully automated and accurate segmentation algorithms for liver portal tract regions is an essential step to improve the evaluation consistency.

To address this problem, a Multiple Up-sampling and Spatial Attention guided UNet model (MUSA-UNet) is developed to segment liver portal tract regions in whole-slide images of liver tissue slides. To enhance the segmentation performance, depth-wise separable convolution, the spatial attention mechanism, the residual connection, and multiple up-sampling paths are adopted in the developed model. The network is trained with image patches and applied to liver biopsy WSIs. The segmentation performance evaluation and clinical correlation analysis demonstrate the efficacy of the developed method.

4.1 Method

The overall schema of the developed method is presented in Figure 4.1A. Images in the dataset are scanned with stained liver biopsy sections and utilized for training the developed deep neural network. With network prediction results and human annotations, I quantitatively evaluate the network performance by statistical analyses.

4.1.1 Deep neural network architecture

To make a full use of image information for segmentation, I have developed a Multiple Up-sampling and Spatial Attention guided UNet model (MUSA-UNet) that leverages the UNet architecture as the building block. The UNet architecture is known as a symmetric encoder-decoder framework that can effectively differentiate foreground pixels from the background by learning and incorporating local features from the higher resolution images and global information from the lower resolution images [63]. However, the UNet model demonstrates a noticeably high false-negative rate by the experiments. To enhance model performance, two new mechanisms are designed to specifically address this problem.

(1) I have developed a new Residual Spatial Attention (RSA) block to replace the sequence of two convolution layers in the original UNet for enhanced network performance. The designed RSA block consists of a residual network embedded with one Depth-wise Separable Convolution (DSC) and one Spatial Attention (SA) module.

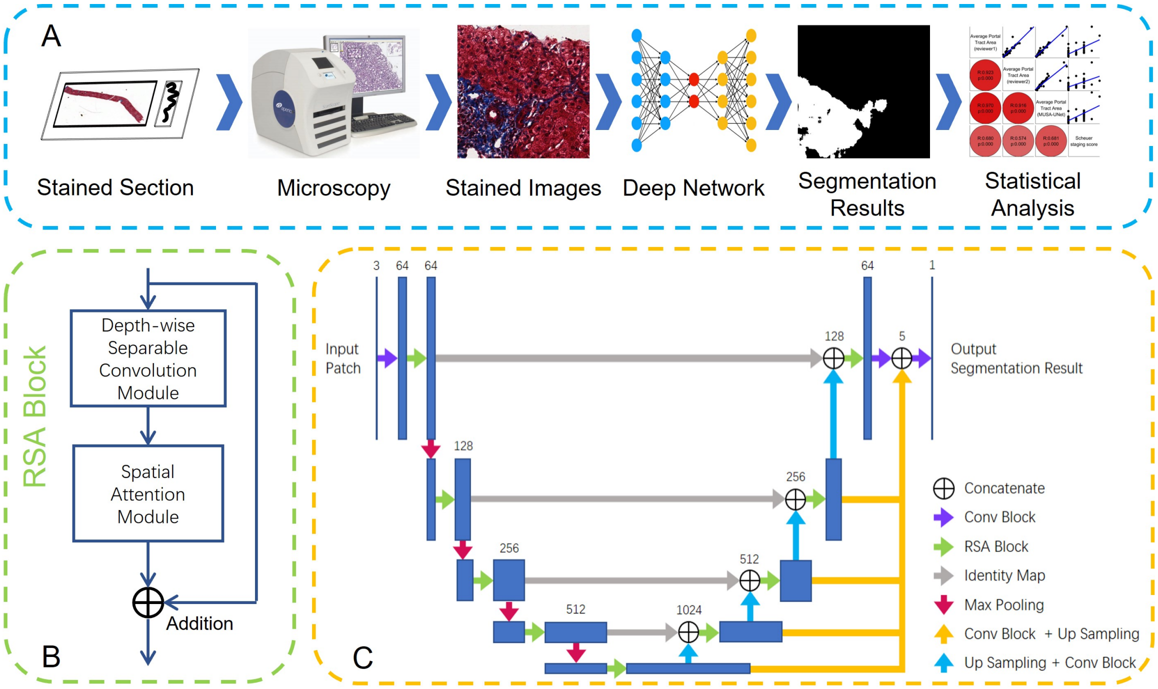

Figure 4.1: Overall schema of the developed model. (A) Tissue sections were fixed, embedded, stained, and scanned for WSI generation. Resulting WSIs with human annotations are provided to the developed MUSA-UNet for portal tract segmentation and statistical analyses; (B) I present the structure of the developed RSA block that substitutes cascaded convolutional layers in the traditional UNet architecture. It primarily consists of one Depth-wise Separable Convolution (DSC) block and one Spatial Attention (SA) module connected by a residual network; (C) The developed deep learning neural network MUSA-UNet for image segmentation concatenates features from all decoders. As there are multiple paths providing lower resolution features from decoders to the output layer, such a Multiple Up-sampling Path (MUP) mechanism alleviates the false negative problem noticeable in the original UNet model in this study.

The RSA block architecture is presented in Figure 4.1B. Specifically, the output of a RSA block can be formulated as follows:

$$RSA\left(x\right) = SA\left(DSC\left(x\right)\right) + x \tag{4.1}$$

where x is the input feature array; $SA(\cdot)$ and $DSC(\cdot)$ are the spatial attention and the DSC module, respectively.

A DSC module has been adopted to divide a regular convolution layer into a depth-wise and a point-wise convolution layer for parameter number regulation [104, 105]. It has been shown that the performance of a DSC module is similar to that of the regular convolution layer in UNet architecture [106]. I replace the regular convolution modules with DSC modules in the RSA model to reduce model parameter number and accelerate training speed.

Additionally, SA modules are adopted to further improve network performance. Both SA and Channel Attention (CA) modules are originally proposed as components of the Convolutional Block Attention Module (CBAM) [107], a lightweight attention method. As the training and testing input image sizes can be different, the CA module barely improves or even degrades the segmentation performance in tests. Therefore, only the SA module is leveraged in the developed model. The output of the SA module can be represented as $SA(x) = M_{SA}(x) \otimes x$, where $\otimes$ denotes element-wise multiplication, and $M_{SA}(x)$ is the 2D spatial attention map. To enable the element-wise multiplication, I broadcast the spatial attention map along the channel dimension to match the tensor size. The spatial attention values are determined by the average- and max-pooled features across channels. Specifically, the average- and max-pooled features are concatenated and convolved in a convolution layer:

$$M_{SA}(x) = \sigma(f^{7\times 7}([AvgPool\,(x)\,; MaxPool\,(x)])) \tag{4.2}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $f^{7\times 7}(\cdot)$ denotes a convolution operation with kernel size of 7×7.

I further use the residual connection to encapsulate the DSC and SA modules for direct information forward-feeding and back-propagation paths in the developed deep network. Originally adopted to improve the image classification [108], residual connection block has shown its promising efficacy for the biomedical image segmentation tasks [51, 109]. Given the original network is denoted as $H(x)$, its residual representation is $H(x) + x$. The residual connection in the developed RSA block can improve the network performance without extra convolution layers.

(2) The second primary method development contribution is that I concatenate features from all decoders at different resolution levels as input to the output layer (i.e., orange arrows in Figure 4.1C. In addition to features at the highest image level, the feature arrays in the lower image resolutions are leveraged in the developed model by convolving with a 3×3 filter for feature dimension reduction. The reduced features are resized to the highest image resolution by the bilinear interpolation before they are concatenated at the output layer. In contrast to FCN utilizing features from encoders [61], the developed model uses features from decoders. This design enables the output layer to make full use of multi-scale features and avoid the false negative problem with only a negligible increase in the parameter number. As there are multiple signal paths that lower resolution features from decoders can follow to reach the output layer in the model architecture, such a Multiple Up-sampling Path (MUP) mechanism is an effective solution to remedy the false negative problem observed in the UNet model in this study.

The architecture of the developed MUSA-UNet model is presented in Figure 4.1C. Specifically, the MUSA-UNet consists of one input layer, four encoder-decoder pairs, and one output module. The encoders gradually decrease the image resolution by max-pooling layers while the decoders increase the image resolution by bilinear interpolation layers. In addition to the primary information encoding and decoding path, there are skip connections between the encoder output and the decoder input at each spatial resolution level. Therefore, there are two information sources provided to each decoder, one from a lower resolution decoder and another from the encoder output at the same resolution level. Note the feature representations from the lower resolution decoder are up-sampled and convolved before they are concatenated with the encoder output from the same resolution level. The outputs from distinct resolution levels are convolved and up-sampled before they are concatenated as the input to the output module.

4.1.2 Model implementation

Due to the overwhelming size of histopathology WSIs and the limited Graphical Processing Unit (GPU) memory size, deep learning models cannot be practically trained or tested on arbitrarily large images to achieve seamless segmentation. Therefore, I divide each WSI into image patches, apply trained models to individual patches, and assemble the patch-wise results.

A straight-forward partitioning strategy is to divide each WSI by a grid pattern. In that way, the segmentation output image can be produced by patch-wise segmentation results in the same spatial order of input image patches. However, the performance of this strategy could be degraded by the image patch border effect. Note that the prediction results of the same region in patches of varying sizes can be inconsistent, especially for those regions near patch borders. As deep learning analyses heavily depend on convolution operations and produce output patches of the same size as the input patches, padding methods for convolutions on pixels close to image borders are required [110]. The prediction results of pixels near patch borders are subject to the padded pixels and, therefore, can deviate from the ground truth.

To mitigate such image border effect, a patch partitioning strategy is adopted to support a seamless semantic segmentation [63]. Its overall schema is presented in Figure 4.2. First, an input WSI is divided by a regular grid pattern. To predict a target image patch in the grid, its region scope is extended before provided to the network MUSA-UNet for image segmentation. In Figure 4.2, the image regions denoted by dotted lines are the target image patches, while those in solid boundaries are extended counterparts. The margin for such an image patch expansion is set in

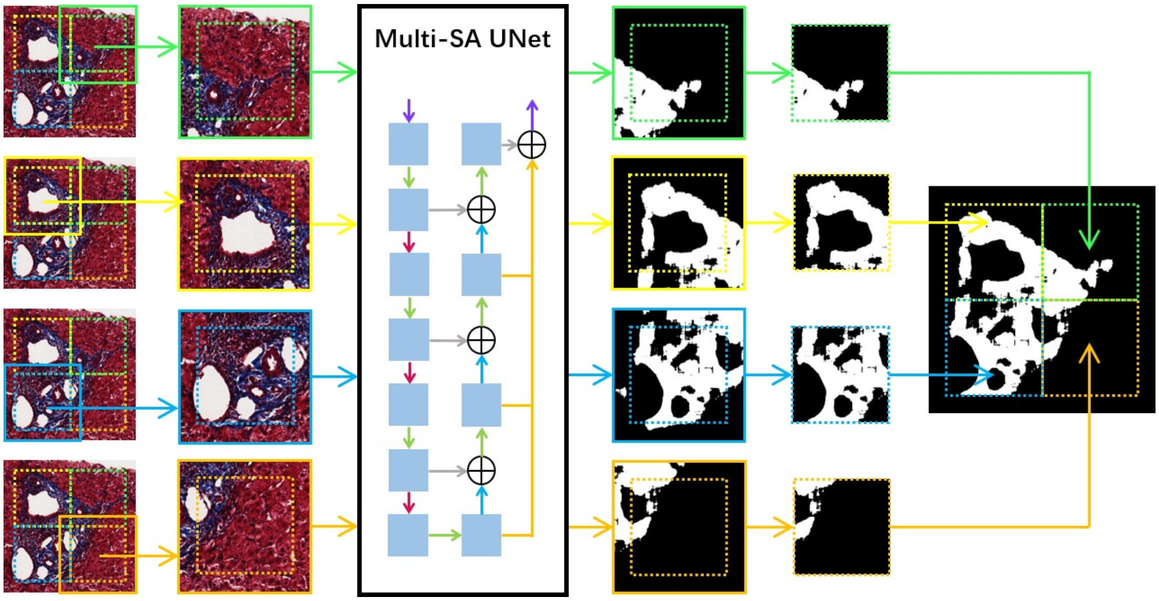

Figure 4.2: The patch partitioning strategy for seamless semantic segmentation in a large-scale image. To predict target patches in dotted lines, these image patches are extended before they are provided to the deep learning network for segmentation. The resulting segmentation output images are cropped back to the original patch size before the segmentation map aggregation.

such a way that prediction results of the original image patches are not influenced by padded pixels. After segmentation analysis by the deep learning network, I retain the segmentation result of the interior regions associated with the original image patch region and assemble such results for the whole-slide segmentation maps by their spatial positions.

In the testing stage, only image patches with enough foreground tissue (i.e., foreground patches) are expanded and provided to the trained network. Those with no significant tissue presence are skipped for the segmentation analysis, and the corresponding pixels in the resulting segmentation map are set to zero. For foreground patch recognition, each image patch is converted from the RGB to HSV color space and count the number of foreground pixels with a saturation value larger than 0.2. Those with more than 1% foreground pixels are considered as foreground patches. To accelerate the testing speed, the image resolution is reduced by 16 times before the foreground detection approach is applied in practice.

Note that the strategy allows parallel computing on multiple GPUs. I implement codes in the Python 3.6 programming language and PyTorch 1.7.1 machine learning framework [111] and run programs on two NVIDIA Tesla K80 GPUs. Balancing the tradeoff between network efficacy and computational efficiency, I design five image resolution levels in the developed model, with 64, 128, 256, 512, and 1024 filters from the highest to the lowest level, respectively. The loss function is the binary cross-entropy that can effectively reflect the pixel-wise difference between label and prediction. The model is trained with the Adam optimization algorithm [112] for 40 epochs. The initial learning rate is set as 0.001 and the learning rate decay is 0.1 per ten epochs. In the testing stage, each image patch has $1,000 \times 1,000$ pixels, with an extended margin width of 140 pixels. Thus, each extended image patch has $1,280 \times 1,280$ pixels by size.

4.1.3 Portal tract guided fibrosis quantification

As reported in the prior study [90], portal tract fibrotic percentage (i.e., portal tract fibrosis%) and average portal tract area derived from portal tract regions are correlated with Scheuer fibrosis staging. In this study, the Aperio ImageScope Positive Pixel Count (PPC) algorithm (Aperio Technologies Inc., Vista, CA) is applied to portal tract regions for quantification of the fibrous component in each portal tract by blue hue in the Masson's Trichrome stain. After the fibrous components from the portal tract regions are measured by the PPC algorithm, the portal tract fibrosis% and the average portal tract area are computed. The portal tract fibrosis% is calculated as the proportion of the total fibrosis area in the total portal tract region area, while the average portal tract area is computed by dividing the total portal tract area by the portal tract region number in a slide. I further investigate the correlation of 1) Scheuer stage scores and average fibrosis areas; and 2) Scheuer stage scores and portal tract percentages (i.e., portal tract%), respectively. The average fibrosis area is computed by dividing the total fibrosis area by the portal tract number in a slide, while the portal tract% is the proportion of the total portal tract area in the total tissue area in a slide. The total tissue area is computed by subtracting the background pixel number from the total pixel number in an image.

4.1.4 Statistical analysis

In this study, statistical analyses are performed with Python 3.6 and MATLAB R2021a (MathWorks Inc., Natick, MA). For the segmentation performance evaluation, precision, recall, F1 score, accuracy, Jaccard index, and Fowlkes–Mallows index are computed. For performance comparisons, the paired sample t-test is used to determine the statistical significance of differences in these metrics. Correlations between fibrosis stage and portal tract measures (including portal tract fibrosis%, average portal tract area, average fibrosis area, and portal tract%) are evaluated by linear regression and Spearman correlation analysis. The paired sample student's t-test is used to determine the statistical significance of the calculated Spearman correlation coefficients. A p-value less than significance level 0.05 is considered significant.

4.2 Results

4.2.1 Training and testing datasets

The dataset for this study includes 53 WSIs of liver tissue biopsies. Two pathologists with GI/Liver pathology expertise (K.J. and A.B.F.) provide portal tract region ground truth for the dataset. The portal tract regions are first annotated by K.J. and then validated and corrected by A.B.F. Biopsies are partitioned into training, validation, and testing dataset. Note all biopsies for the training and validation are mutually exclusive from those for the testing. Of all biopsies, 30 biopsies including 22 men and

8 women are used to generate image patches for model training and validation, with a mean ± standard deviation (S.D.) age of 54.5 ± 6.9 years. We programmatically load manually annotated portal tract contours, calculate their bounding boxes, and divide them into patches of size 512×512 pixels. Additionally, we rotate image patches by 90, 180, and 270 degrees for training data augmentation. In total, we generate 6,012 image patches, with 80% and 20% for training and validation, respectively. The remaining 23 biopsies WSIs are allocated for testing, with 18 men and 5 women with a mean ± S.D. age of 51.8 ± 7.7 years.

4.2.2 Deep learning model validation

Figure 4.3A presents a typical portal tract region segmentation result by the developed MUSA-UNet network. The model detected portal tract region borders are in yellow, while the ground truth portal tract regions are manually delineated and indicated by green borders in Figure 4.3B. Such portal tract regions are automatically identified by binarization of the probability maps from the network in Figure 4.3C. By visual assessments, Note that the predicted region contours are highly concordant with the corresponding ground truth regions, suggesting the effectiveness of the developed model. As detailed in the methods section, each original WSI for testing is divided into a set of patches and process them separately. Due to this partitioning step, portal tract regions close to image patch borders are subject to an image padding effect, resulting in inaccurate segmentation results. Figure 4.4 presents portal tract segmentation results of two typical biopsy image regions divided with and without patch expansion partitioning strategy. Note that inaccurate segmentation results (by yellow arrows) when images are divided directly (blue dashed lines). Due to the border effect, portal tract regions on patch borders tend to be missed by the model. By contrast, the expanded patches by the patch expansion partitioning strategy are indicated by solid blue lines. This strategy substantially eliminates the segmenta-

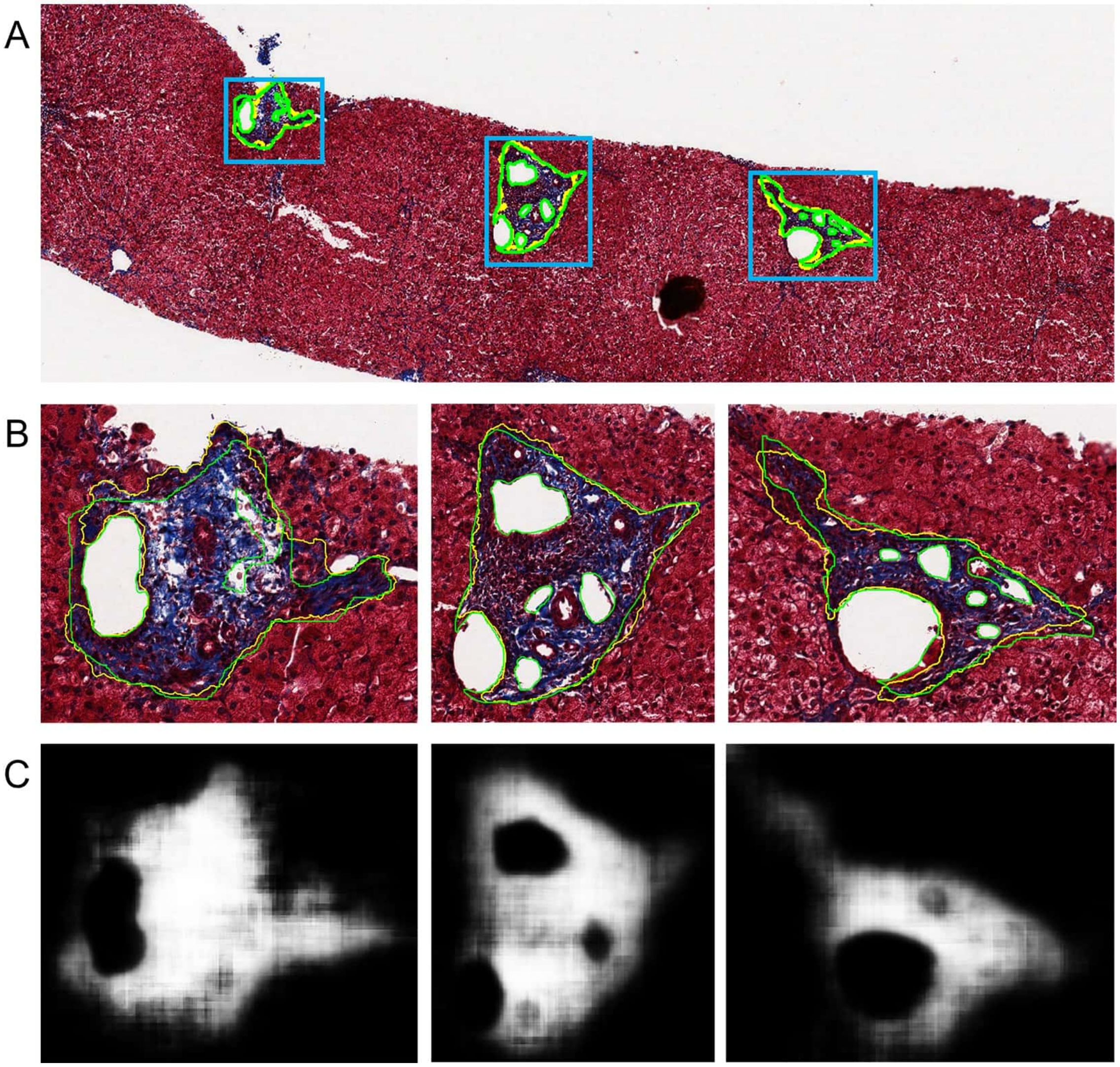

Figure 4.3: typical portal tract segmentation result with a liver biopsy WSI. (A) Manual annotations (i.e., ground truth) and deep learning results of portal tract regions by the MUSA-UNet deep neural network are delineated in green and yellow, respectively; (B) Annotation and segmentation details are presented in close-up views; (C) The model generated prediction probability maps are presented for the same corresponding image regions.

tion errors by adding additional image margins to make the inception fields more informative and consistent.

In addition to qualitative assessments, I next validate the developed model quantitatively. The developed MUSA-UNet model is compared with three widely used approaches, i.e., FCN [61], UNet [63], and DeepLab [62]. FCN and UNet have been

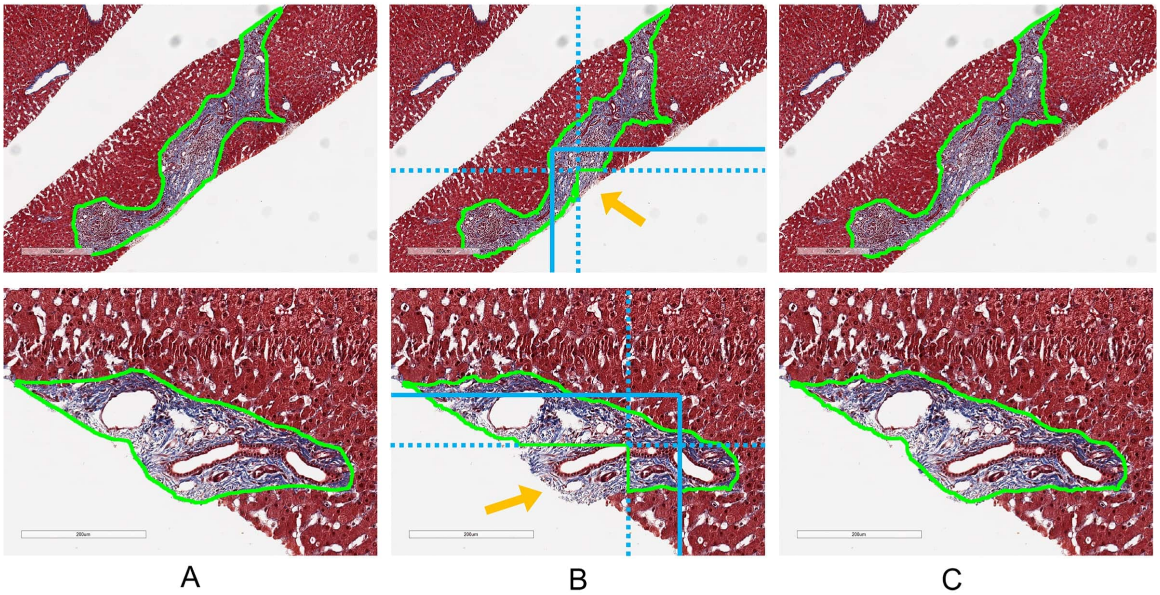

Figure 4.4: Comparison of portal tract segmentation results of two biopsy tissue regions with and without the partitioning strategy. (A) Ground truth portal tract contours are annotated by human experts; (B) Portal tract segmentation results are presented when WSIs are simply divided into non-overlapping patches with their borders in blue dashed lines. The resulting segmentation defect is highlighted by a yellow arrow; (C) Portal tract segmentation results are demonstrated when the patch expansion partitioning strategy is used. The solid blue lines in (B) represent the borders of the expanded patches. With the expansion partitioning strategy, such negative border effects are successfully mitigated.

widely applied to a large number of biomedical image segmentation tasks [98]. The DeepLab model is derived from the FCN model, but with an atrous convolution [62]. This change expands the convolution perception field for enhanced segmentation accuracy without an increase in the parameter number. All the approaches are trained with the same training parameters and dataset as the developed model. I present and compare typical normal tissue segmentation results by these models in Figure 4.5. By visual comparisons, the segmentation results from the MUSA-UNet are more concordant with the ground truth than other methods. Additionally, I present and compare typical abnormal portal tract segmentation results in Figure 4.6. These abnormal portal tract types include portal tracts with (1) lymphoid aggregate, (2) duc-

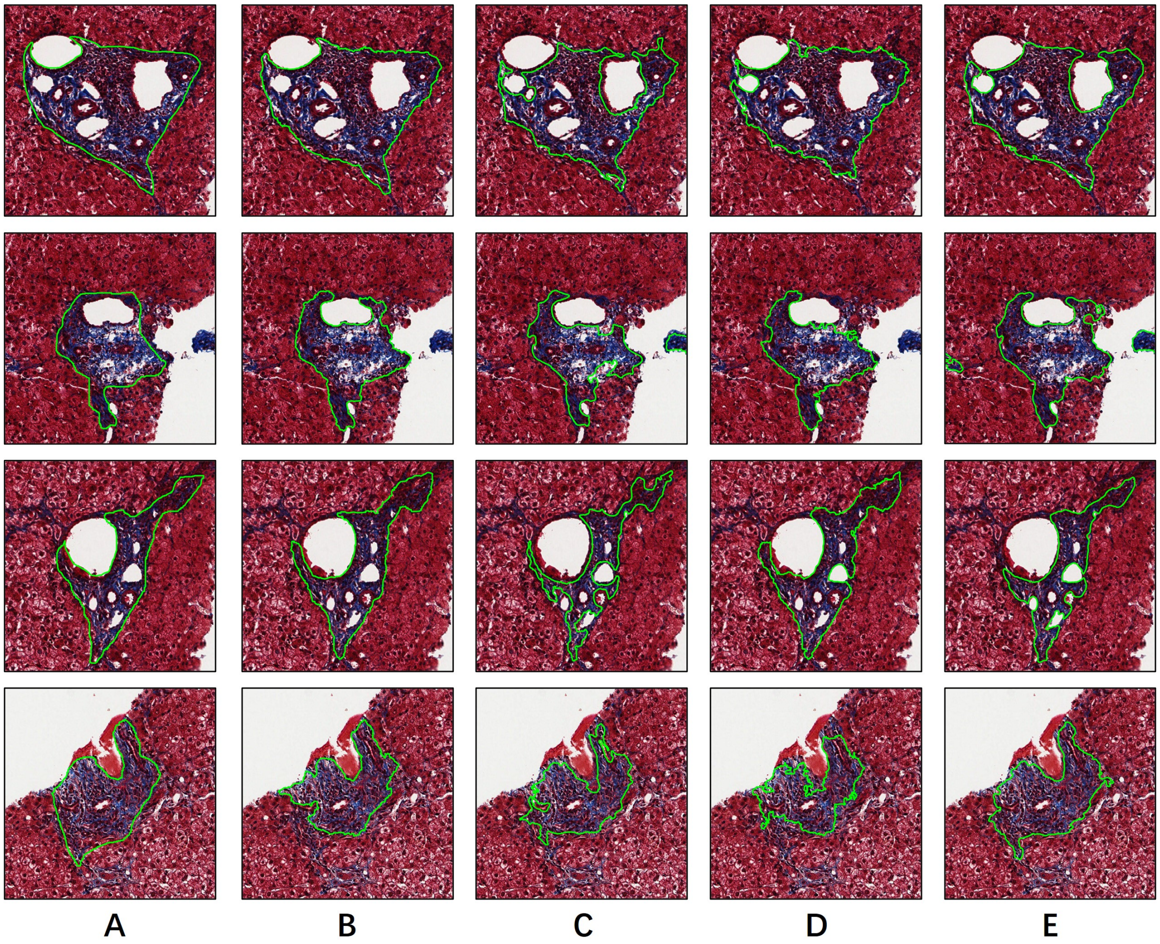

Figure 4.5: Qualitative comparison of deep learning models for normal liver portal tract segmentation. Typical segmentation results of four normal liver tract regions are presented by (A) human annotations (i.e., ground truth), (B) the developed MUSA-UNet model, (C) DeepLab, (D) UNet, and (E) FCN, respectively.

tular proliferation with minimal collagen, (3) edema, mild inflammation, and ductular proliferation, (4) features of acute cellular rejection, including mixed inflammatory infiltrate and ductitis, and (5) portal vein herniation and moderate chronic inflammation. Compared with FCN and DeepLab, the developed MUSA-UNet demonstrates a better generalizability on abnormal portal tract segmentation.

Additionally, I compare segmentation results from different models with the ground truth from human annotations and quantitatively evaluate their performances. Compared to the ground truth, each pixel in the segmentation map is labeled as one of

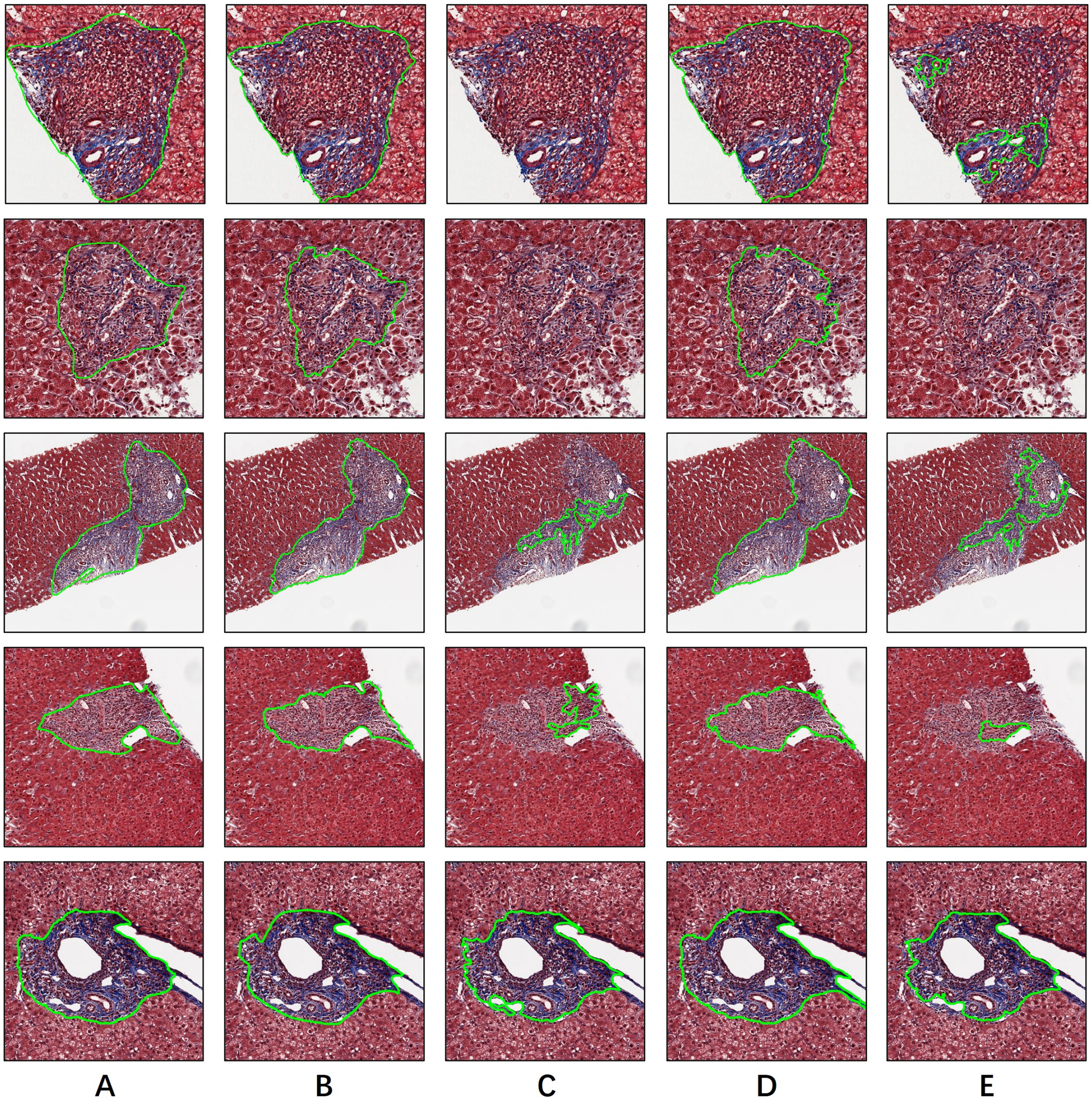

Figure 4.6: Qualitative comparison of deep learning models for abnormal liver portal tract segmentation. Typical segmentation results of multiple abnormal liver tract regions are presented by (A) human annotations (i.e., ground truth), (B) the developed MUSA-UNet model, (C) DeepLab, (D) UNet, and (E) FCN, respectively. From top to bottom, abnormal portal tracts contain (1) lymphoid aggregate, (2) ductular proliferation with minimal collagen, (3) edema, mild inflammation, and ductular proliferation, (4) features of acute cellular rejection, including mixed inflammatory infiltrate and ductitis, and (5) portal vein herniation and moderate chronic inflammation.

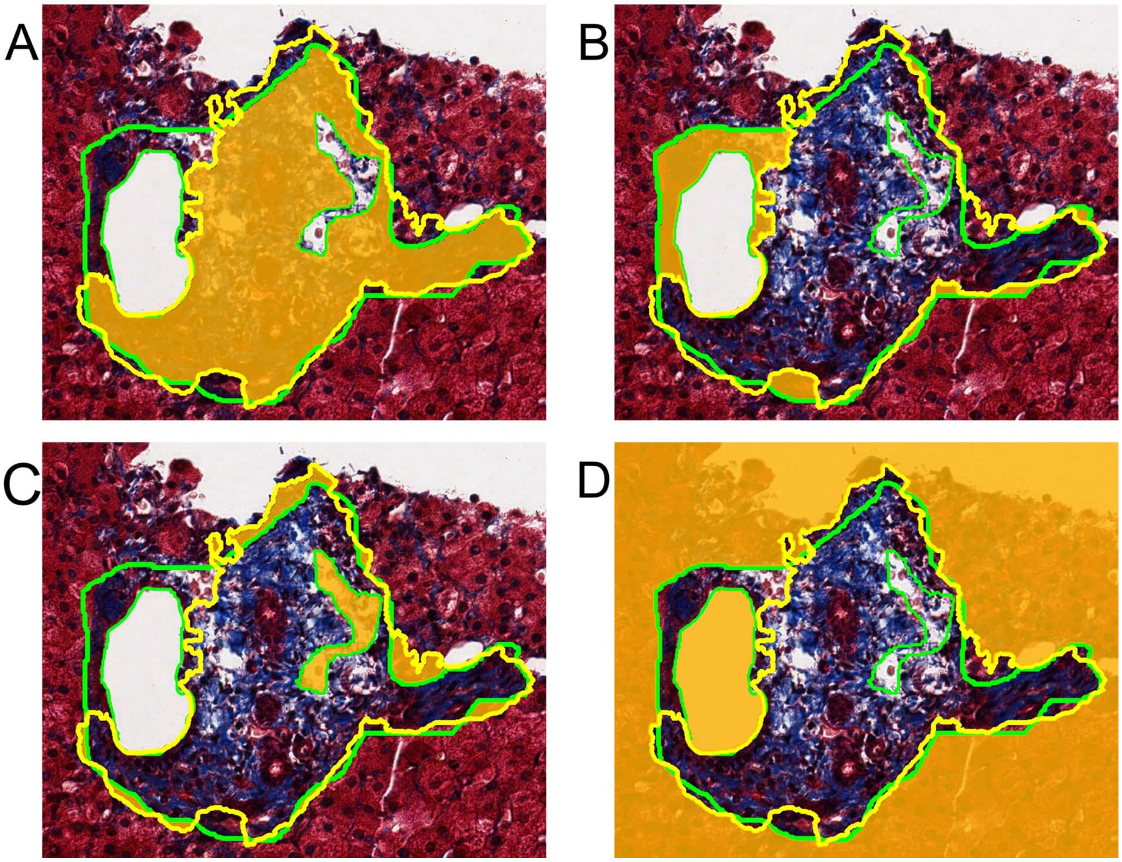

Figure 4.7: Pixel-wise segmentation labels for quantitative evaluation. Ground truth and deep learning segmentation results are represented by green and yellow contours. (A) TP is the class for pixels that are correctly segmented as portal tract; (B) FP is the label for pixels that are falsely recognized as portal tract; (C) FN is class for pixels that are missed as portal tract by mistake; (D) TN is the label for pixels that are correctly recognized as non-portal tract.

the four classes, True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). TP is the class for pixels that are correctly segmented as portal tract; FP is the label for pixels that are falsely recognized as portal tract; FN is the class for pixels that are missed as portal tract by mistake. Finally, TN is the label for pixels that are correctly recognized as non-portal tract. These four classes of pixels are illustrated in Figure 4.7. With these defined classes, I compute pixel-based evaluation metrics (each ranging from 0 to 1), including Precision (P), Recall (R), F1

Table 4.1: Quantitative performance comparison across the developed MUSA-UNet model and other state-of-the-art segmentation models by multiple evaluation metrics (mean ± standard deviation).

Model	Precision	Recall	F1 Score	Accuracy	JI	FMI
UNet	0.943	0.806	0.858	0.866	0.765	0.900
FCN	**0.958**	0.688	0.776	0.798	0.664	0.780
DeepLab	0.942	0.830	0.874	0.870	0.787	0.900
MUSA-UNet	0.940	**0.847**	**0.886**	**0.889**	**0.801**	**0.914**

score (F1), Accuracy (A), Jaccard index (JI), and Fowlkes–Mallows Index (FMI):

$$\begin{aligned} P &= \frac{TP}{TP+FP}, & R &= \frac{TP}{TP+FN}, \\ F1 &= \frac{2 \cdot P \cdot R}{P+R}, & A &= \frac{TP+TN}{TP+FP+FN+TN}, \\ JI &= \frac{TP}{TP+FP+FN}, & FMI &= \sqrt{P \times R}. \end{aligned} \tag{4.3}$$

Table 4.1 presents quantitative evaluation results of all models for comparison with Precision, Recall, F1 score, Accuracy, Jaccard index, and Fowlkes–Mallows Index. Although FCN has the best performance by Precision (0.958), other methods (i.e., UNet, DeepLab, and MUSA-UNet) do not present significantly worse performances by paired sample t-tests with p-value 0.59, 0.30, and 0.29, respectively. By Recall, the MUSA-UNet demonstrates the best performance (0.847) and a statistically significant performance difference compared with UNet, FCN, and DeepLab with p-value 0.007, <0.001, and 0.03, respectively. By F1 score, the MUSA-UNet achieves the best performance (0.886) and presents a statistically significant performance difference compared with UNet, FCN, and DeepLab with p-value 0.01, <0.001, and 0.03, respectively. When assessed by Accuracy, the MUSA-UNet presents the best performance (0.889) and a statistically significant performance difference compared with UNet, FCN, and DeepLab with p-value 0.04, 0.002, and 0.04, respectively. By JI, MUSA-UNet has the best performance (0.801) and presents a statistically significant performance difference compared with UNet, FCN, DeepLab with p-value

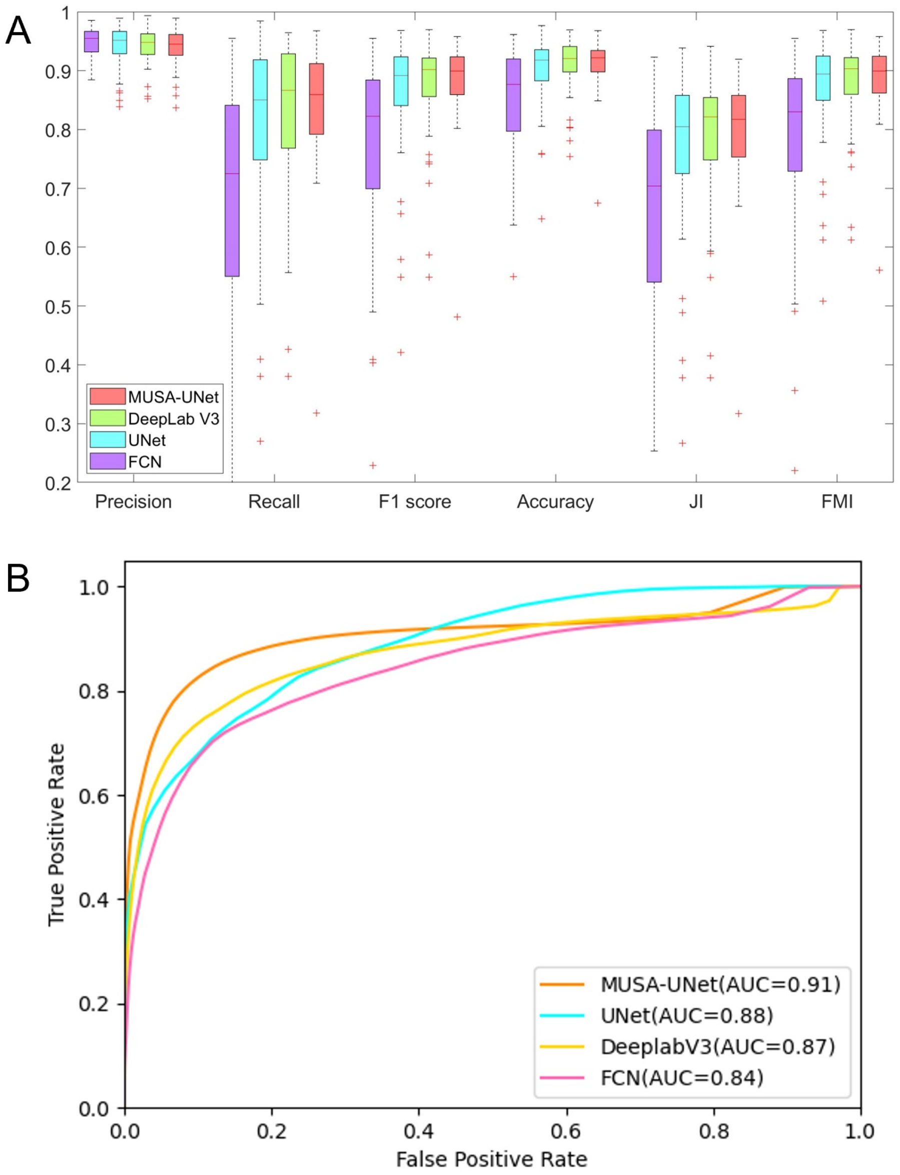

Figure 4.8: Quantitative comparison of deep learning models for liver portal tract segmentation. (A) Paired sample t-tests between the MUSA-UNet and other three widely used models (i.e., FCN, UNet, and DeepLab) suggest a statistically significant performance difference with p-values<0.05 by Recall, F1 score, Accuracy, JI, and FMI; (B) Of all deep learning models for comparison, the developed MUSA-UNet achieves the best AUC with Receiver Operating Characteristic (ROC) curves.

0.01, <0.001, and 0.03, respectively. the MUSA-UNet presents the best performance by FMI (0.914) and a statistically significant performance difference compared with UNet, FCN, and DeepLab with p-value 0.03, <0.001, and 0.05, respectively. I present the evaluation results in Figure 4.8A where evaluation performances of deep learning

Table 4.2: Quantitative model performance comparisons for the ablation study (mean ± standard deviation).

Model	Precision	Recall	F1 Score	Accuracy	JI	FMI
UNet	0.943	0.806	0.858	0.866	0.765	0.900
UNet+DSC+CBAM	0.942	0.757	0.822	0.879	0.716	0.833
UNet+DSC+CA	**0.957**	0.530	0.633	0.860	0.504	0.797
UNet+DSC+SA	0.940	0.846	0.885	0.880	0.799	0.911
UNet+RSA	0.941	0.844	0.884	0.888	0.799	0.914
UNet+MUP	0.942	0.816	0.867	0.866	0.774	0.899
MUSA-UNet (UNet+RSA+MUP)	0.940	**0.847**	**0.886**	**0.889**	**0.801**	**0.914**

models for comparison are demonstrated by all six metrics. Note that MUSA-UNet presents fewer outliers than other methods, implying its strong stability. In Figure 4.8B, I present and compare the Receiver Operating Characteristic (ROC) curves of MUSA-UNet, UNet, DeeplabV3, and FCN models, respectively. Of all these models, the developed MUSA-UNet model achieves the largest Area Under the Curve (i.e., AUC=0.91).

4.2.3 Ablation study

To investigate the contribution of individual modules for portal tract segmentation, I carry out ablation experiments and present the ablation study results in Table 4.2. Small Attention (SmaAt) UNet replaces convolution layers with two cascaded DSC modules and appends CBAM (CA+SA) blocks to DSCs [113]. Noticeably, model UNet+DSC+CBMA (i.e., SmaAt UNet) presents an inferior performance to that of the UNet model for the portal tract segmentation task. To identify the performance degradation reason, I remove either a SA or a CA module from the UNet+DSC+CBMA separately. The experimental results suggest that the DSC+CA dramatically decreases the performance while the DSC+SA improves Recall (0.846), F1 score (0.885), Accuracy (0.880), JI (0.799), and FMI (0.911). I thus design a new RSA block by retaining only one DSC module and encapsulating the DSC and SA

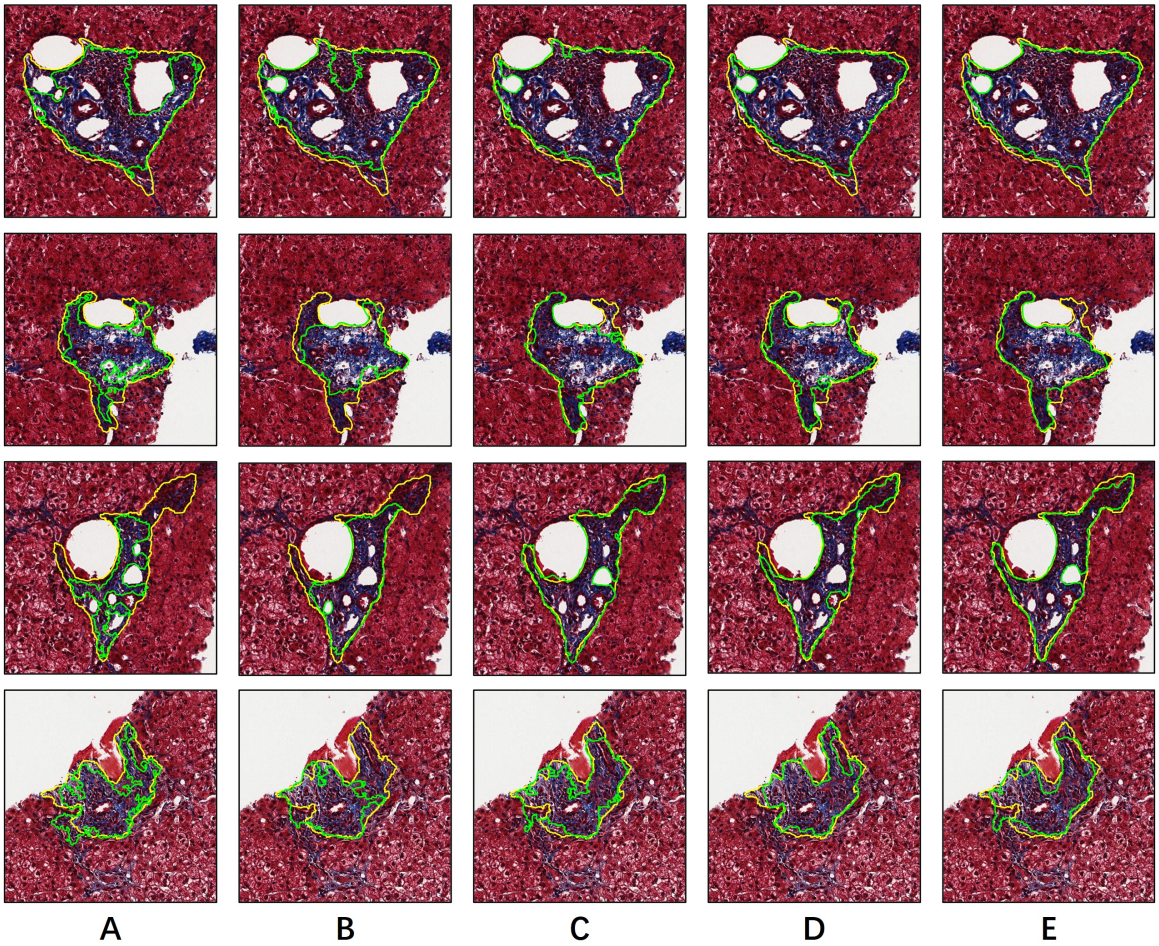

Figure 4.9: Qualitative comparison of ablated models for liver portal tract segmentation. I illustrate and visually compare typical tissue segmentation results of ablated UNet models (i.e., (A) UNet+DSC+CBAM, (B) UNet+DSC+CA, (C) UNet+DSC+SA, (D) UNet+RSA, and (E) UNet+MUP) in green and those of the developed MUSA-UNet model in yellow.

models in a residual connection structure. In addition to an improved processing speed, the designed RSA block achieves 0.844, 0.884, 0.888, 0.799, and 0.914 by Recall, F1 score, Accuracy, JI, and FMI, respectively. The paired sample t-test between DSC+SA and RSA block results in a p-value less than 0.001, suggesting a comparable model performance of the designed RSA block. To prove the effectiveness of the MUP mechanism, I add it to the original UNet and achieve improved performance by Recall (0.816), F1 score (0.867), Accuracy (0.866), JI (0.774) and FMI (0.899).

Table 4.3: Summary statistics for multiple portal tract measures (Mean ± standard error; Range).

Measure	Reviewer 1	Reviewer 2	MUSA-UNet
Portal Tract Fibrosis%	51.06±3.61; 19.60 to 89.50	41.27±3.55; 14.36 to 83.28	42.97±3.55; 15.72 to 84.54
Average Portal Tract Area	56,468±8,628; 8,583 to 215,486	50,349±7,153; 5,373 to 181,709	49,688±8,000; 3,226 to 202,321
Average Fibrosis Area	22,541±3,820; 1,618 to 83,115	21,761±3,748; 1,268 to 80,599	22,406±4,090; 1,207 to 91,188
Portal Tract%	2.02±0.40; 0.25 to 9.25	1.86±0.35; 0.22 to 7.93	1.89±0.38; 0.13 to 8.39

Figure 4.9 presents typical segmentation results of four tissue regions by multiple ablated models for comparison. By visual comparisons, the segmentation results from DSC+CA are the worst as multiple portal tract regions are missing. This visual assessment conclusion agrees with the quantification analysis results. Although the result difference between the UNet+RSA and MUSA-UNet model is visually subtle, MUSA-UNet tends to produce smoother portal tract boundaries due to the new MUP design.

4.2.4 Clinical correlation analysis

I investigate the correlation across measures of the portal tract area, the fibrosis area, and the clinical staging score. In addition to the ground truth established by the primary reviewers (reviewer 1: K.J., A.B.F.), a secondary board-certificated pathologist with GI/Liver pathology fellowship training (reviewer 2: N.S.) annotates portal tracts independently for this correlation analysis. Figure 4.10 demonstrates the multivariate analysis results with the linear regression and Spearman correlation. With manually delineated and MUSA-UNet predicted portal tract regions, I compute multiple measures, including portal tract fibrosis%, average portal tract area, average fibrosis area, and portal tract%. Additionally, I investigate and compare their correlations with the clinical Scheuer staging score (mean ± standard error: 0.85±0.23).

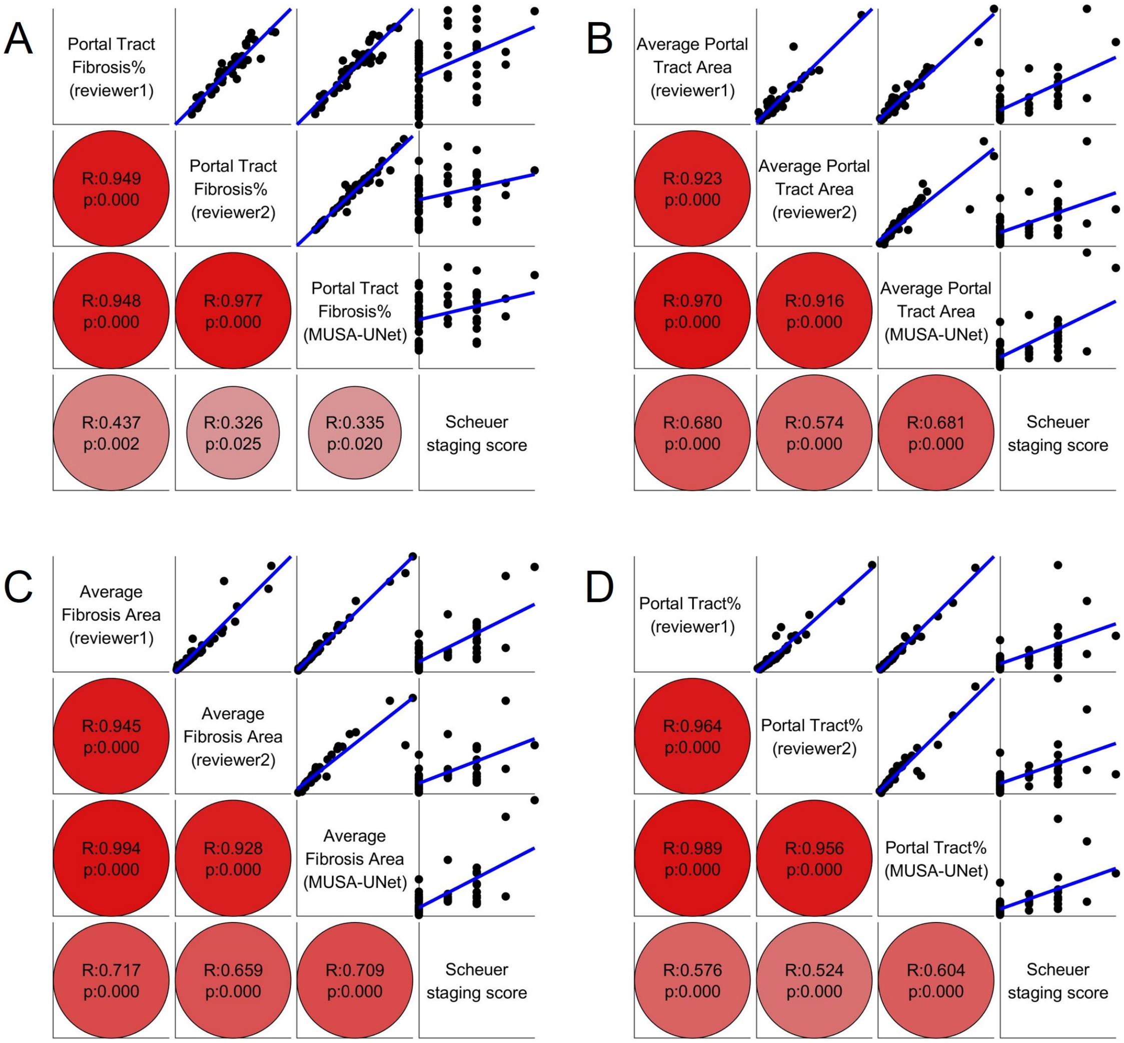

Figure 4.10: Multivariate correlation analysis across portal tract area, fibrosis area, and clinical staging score. The four subplots present the correlation analysis results between Scheuer staging score and (A) portal tract fibrotic percentage, (B) average portal tract area, (C) average fibrosis area, and (D) portal tract percentage, respectively. In each subplot, results from linear regression (top-right) and Spearman correlation (bottom-left) are presented to support the multivariate analysis. For Spearman correlation results, larger correlation coefficients and lower p-values are indicated by darker colors and larger circles.

The summary statistics for these measures are presented in Table 4.3. By Spearman correlation analysis, average portal tract area and portal tract fibrosis% derived from deep learning detected portal tract regions are correlated with clinical Scheuer staging score ($R=0.681$; $p<0.001$ and $R = 0.335$; $p=0.020$, respectively). When the MUSA-

UNet derived measures are replaced with those from portal tract regions annotated by reviewer 1 and reviewer 2, average portal tract areas present comparable correlation relationships with clinical Scheuer staging score (i.e., $R=0.680$, $p<0.001$; and $R = 0.574$, $p<0.001$, respectively). With portal tract regions annotated by reviewer 1 and 2, the portal tract fibrosis% presents similar correlation relationships with clinical Scheuer staging score (i.e., $R=0.437$, $p=0.002$; and $R = 0.326$, $p=0.016$). Such comparable correlation results imply the good concordance between portal tract regions recognized by the developed deep learning model and manual annotators. Figure 4.10A and Figure 4.10B demonstrate a strong correlation between human and deep-learning-derived measures, including portal tract fibrosis% and average portal tract area. Suggested by Figure 4.10C, the correlation between Scheuer staging score and average fibrosis area from deep learning identified portal tract regions is comparable to that between Scheuer staging score and average fibrosis area from human-annotated portal tract regions. By contrast, the correlation between Scheuer staging score and portal tract% from deep learning identified portal tract regions is stronger than that between Scheuer staging score and portal tract% from human-annotated portal tract regions in Figure 4.10D.

Additionally, the differences in the clinical support between the developed model and other methods for comparison are demonstrated. In Figure 4.11, I plot portal tract percentage populations by five Scheuer staging score groups, i.e., stage 0 to 4. Applied to the segmentation results from the developed MUSA-UNet model, the analysis of variance (ANOVA) test suggests a significant difference in population means across staging groups with a p-value 1.44e-4. By contrast, p-values with results from UNet, FCN, and DeepLab are 3.32e-4, 2.92e-2, and 7.70e-2, respectively.

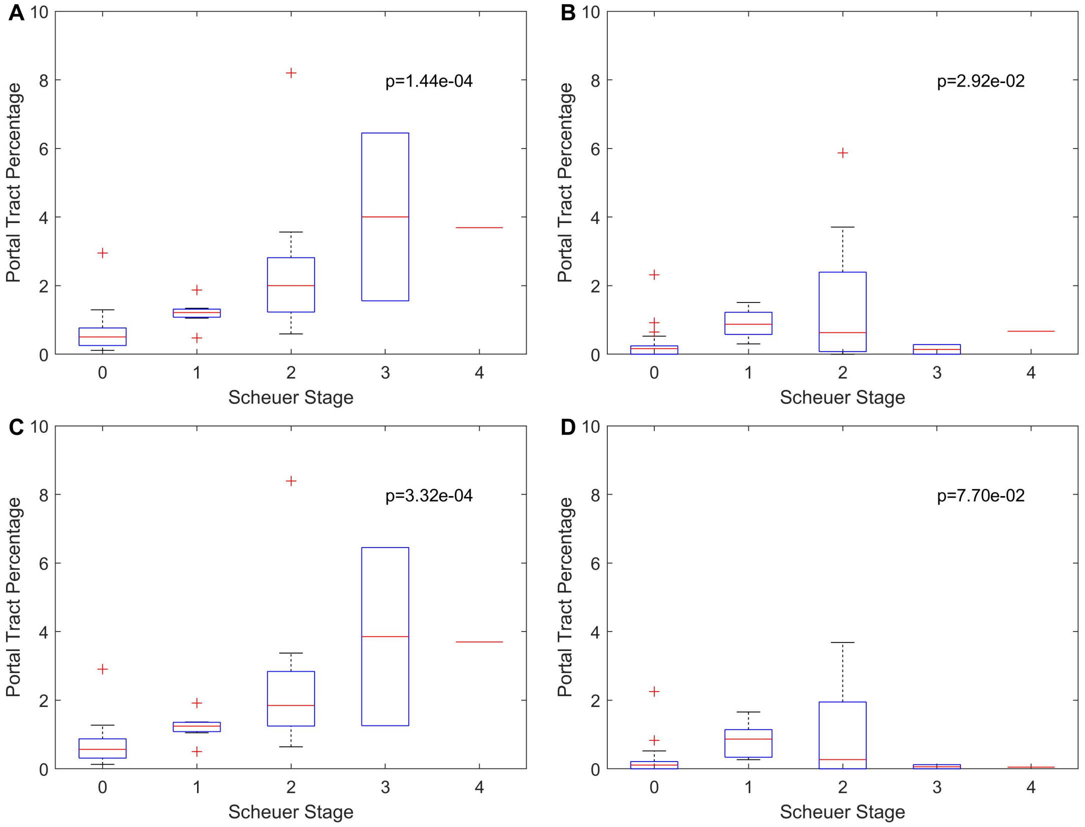

Figure 4.11: Comparison of deep learning models for clinical support. By ANOVA test, the significance of population mean difference across Scheuer staging groups with the portal tract percentage derived from segmentation results of (A) MUSA-UNet, (B) FCN, (C) UNet, (D) DeepLab, is respectively presented.

4.3 Discussion and summary

Leveraging the UNet architecture as a building block, the MUSA-UNet model is developed for liver portal tract region segmentation with liver biopsy WSIs. To reduce the parameter number and accelerate the model processing speed, the regular convolution layers in UNet are replaced with cascaded Depth-wise Separable Convolution (DSC) modules. By experiments, I notice UNet has a limited performance by Recall or JI. To further improve its performance, the attention mechanism is included in the model. Inspired by SmaAt UNet [113], I first append a Convolutional Block Attention Module (CBAM) to the cascaded DSC modules (i.e., UNet+DSC+CBAM), leading

Table 4.4: Model comparison by parameter number and the average processing time cost for a 512 × 512 image patch.

Model	Parameter number	Processing time (ms)
UNet	37,384,833	30.5
FCN	51,938,881	31.1
DeepLab	58,625,857	39.9
MUSA-UNet	9,123,958	23.4

to worse results. To investigate the cause of the model degradation, Channel Attention (CA) and Spatial Attention (SA) modules (i.e., two components in CBAM) are respectively appended to the cascaded DSC modules. The resulting UNet+DSC+CA model presents a degraded segmentation performance, while the UNet+DSC+SA model demonstrates an improved performance. Therefore, I only retain one DSC module, add a SA module, and encapsulate them by a residual connection block to make it more effective for back-propagation. This structure is defined as a Residual Spatial Attention (RSA) block. The resulting model (i.e., UNet+RSA) has fewer parameters, contributing to a better prediction performance and a faster execution speed.

The UNet architecture tends to focus on features derived from the highest image resolution level. By contrast, Fully Convolutional Networks (FCNs) only up-sample the output from the lowest image resolution layer (e.g., the FCN-32s model) [61]. Enlightened by these facts, I address the false-negative segmentation problem commonly seen around portal tract region boundaries by combining features from multiple image resolution levels for the probability map generation. Therefore, the network has a Multiple Up-sampling Path (MUP) mechanism, as there are multiple signal connections between lower resolution features from decoders and the output layer. By experimental results, the concatenated use of features from the top three image resolution levels significantly improves performance. Features from additional lower image resolution levels marginally improve the model performance, but at the cost of the increased model complexity.

In the model design, DSC modules are used to decrease the model parameter number. Table 4.4 presents the parameter number and processing time cost of diverse models for performance comparisons. The processing time cost is calculated by averaging the time cost of 50 image patches of size 512×512. Compared with the developed MUSA-UNet, the original UNet model has the same image resolution level number and the feature number in each level. The FCN model and the DeepLab model are constructed on the base of the ResNet101 backbone [108]. By Table 4.4, the parameter numbers in the models without DSC modules (i.e., UNet, FCN, and DeepLab) are one order of magnitude larger than that of MUSA-UNet. This large difference in model parameter number has an important impact on the resulting processing speed. It takes about two hours for UNet to complete training with a data epoch on the current hardware setup, while the training time cost for the MUSA-UNet model is about 25 minutes. On average, it takes 23.4 ms for MUSA-UNet to predict a 512×512 image patch, promising to support an efficient segmentation analysis for clinical settings.

Chapter 5

Biomedical Image Segmentation with Semi-supervised Learning

In this chapter, I present the developed semi-supervised semantic segmentation method and its application in retinal pigment epithelium (RPE) cell segmentation with flatmount fluorescent microscopy images.

RPE is a pigmented cell layer between the choroid and the neurosensory retina. The main RPE functions are to transport nutrients, maintain the photoreceptor excitability, and secrete immunosuppressive factors [114]. Aging of the RPE can cause a loss or reduction of the indicated functions that affect the function and survival of photoreceptor cells and choroidal cells. Therefore, it may result in the secondary degeneration of photoreceptors and finally lead to irreversible vision loss [115]. Previous studies have suggested that RPE cell morphological features, such as area, perimeter, aspect ratio, polymegathism, and pleomorphism, can be indicators of the cell pathophysiologic status to determine the degree of RPE aging [116, 117, 7].

RPE flatmount images have been widely used to characterize RPE cell morphological features. In the prior work [7], a machine-learning-based ImageJ (National Institutes of Health, Bethesda, MD, USA) plugin known as Trainable Weka Segmen-

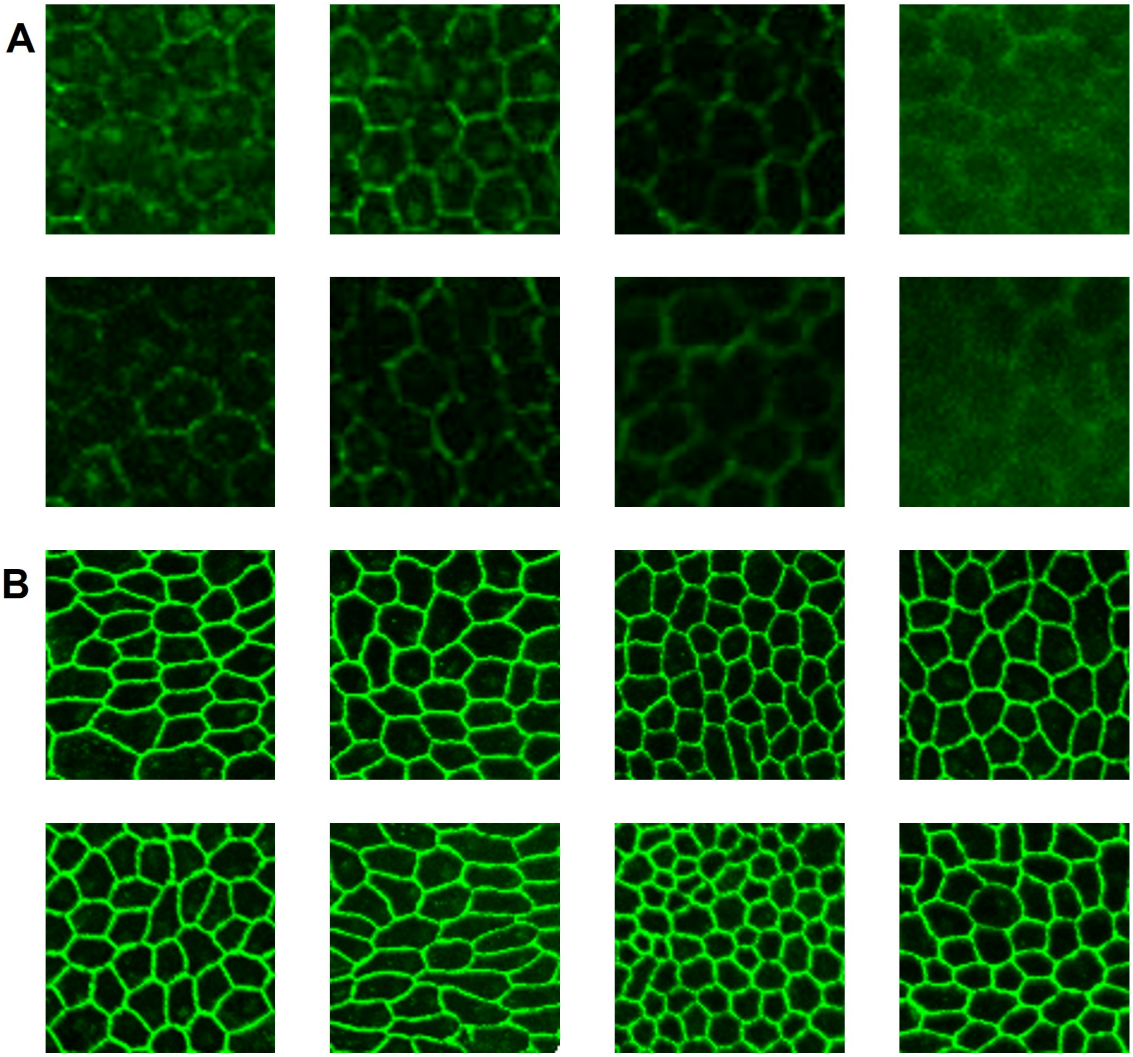

Figure 5.1: Representative examples of RPE flatmount image regions. (A) RPE cells in damaged regions present weak or missing cell borders with a partial or complete cell structure loss. (B) RPE cells in normal regions often have cell borders with a high contrast.

tation [118] was utilized to extract cell borders with a limited success, especially in impaired regions enriched with weak or missing RPE cell borders. Typical examples of damaged regions are given in Figure 5.1. Cell segmentation is a prerequisite step to extract useful parameters such as the average size, shape, orientation, and variations of individual RPE cells. Although computational tool suites are available for cell feature measurement (e.g. CellProfiler [119]), they highly depend on accurate cell segmentation results with RPE tissue sheets. To ensure the accuracy of downstream morphology analyses, manual post-processing steps are, therefore, required to remove these damaged regions from further analyses. This process is not only time-consuming but also significantly reduces the scale of data for analysis, resulting in a weaker study power. More importantly, such an exclusion makes it infeasible to study RPE cell morphology and structures within damaged regions necessary for RPE recovery mechanism and aging investigations. Thus, it is imperative to develop an effective and efficient approach to recover blurred and missing RPE cell borders in large scale flatmount microscopy images.

Unlike traditional machine learning methods, deep neural networks require no manual feature engineering and present an enhanced data learning power to support biomedical research [98, 120]. The basic structure of a deep neural network is composed of layers of computational nodes analogous to brain neurons. For semantic segmentation tasks, a class of deep neural networks consists of an encoder for latent feature extraction from input images and a decoder for mapping the extracted features to desired segmentation results. Deep neural networks have been widely used for segmentation with multiple image modalities, ranging from bright field histopathology image slides [121, 49], CT [50, 122], MRI [51], and immunofluorescence microscopy images [123, 124]. Although there are multiple state-of-the-art deep learning models [124, 61, 63, 62] that can be potentially used to segment RPE cells presenting blurred or missing cell borders, they require a large-scaled annotated training dataset. As the manual annotations on RPE cells in damaged regions are time-consuming, we only have a small set of annotated weak RPE cells insufficient to support the supervised learning strategy by these state-of-the-art deep learning models.

To address this challenge, a semi-supervised deep learning approach, namely MultiHeadGAN, is developed to segment low contrast cells from impaired regions in RPE flatmount images. The developed deep learning model has a multi-head structure that allows model training with only a small scale of human annotated data. To strengthen model learning, I further train the model with RPE cells without ground truth cell borders by generative adversarial networks. Additionally, a new shape loss is designed to guide the network to produce closed cell borders in the segmentation results. Compared with other state-of-the-art deep learning approaches, the developed method demonstrates its superior qualitative and quantitative performance.

5.1 Methods

5.1.1 Deep neural network architecture

For deep learning based image segmentation, deep neural networks often consist of an encoder and a decoder that are trained with a large amount of annotated data. By contrast, a GAN-based image translation mechanism and a semi-supervised learning strategy are used to improve the model performance due to a limited set of data annotations available in this work. In image translation tasks, GANs usually consist of two key components, i.e., a generator and a discriminator. The generator attempts to minimize the adversarial loss and translate inputs to images indistinguishable from real target images by the discriminator. By contrast, the discriminator is trained to maximize the adversarial loss and distinguish the fake images from real ones.

The overall architecture of the developed multi-head deep learning model (MultiHeadGAN) is presented in Figure 5.2. Note MultiHeadGAN makes a full use of both limited data with annotations and a large set of unlabeled images for training. The generator in MultiHeadGAN is derived from the U-Net [63, 44], but extended to multi-heads for contrast enhanced gray-scale and binary segmentation outputs. Different from the U-Net with one encoder and one decoder, the developed generator has one encoder, two decoders, and one feature extractor. For each encoder input s, there are two decoder output images $G1(\boldsymbol{s})$ and $G2(\boldsymbol{s})$ and a feature extractor output

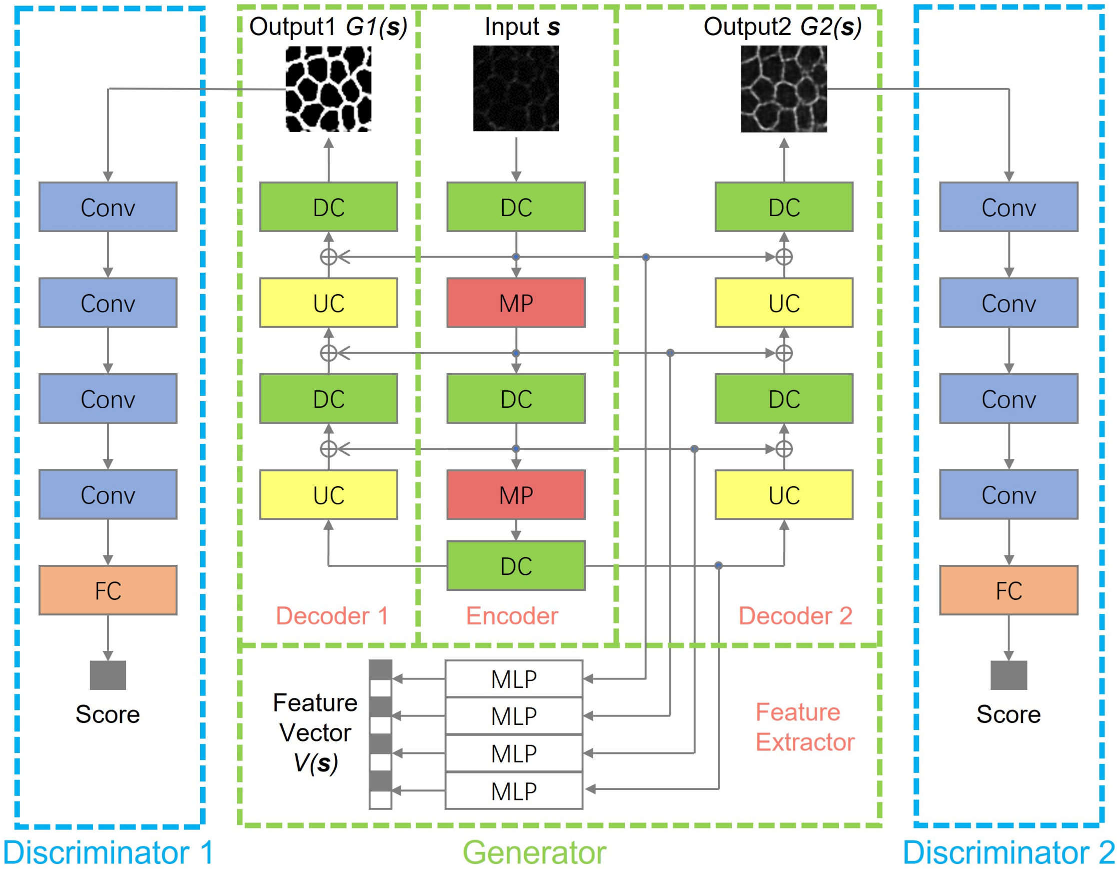

Figure 5.2: Overall schema of the developed multi-head deep learning approach MultiHeadGAN. The developed deep learning generator consists of one encoder, two decoders, and one feature extractor. For each image input, the network produces two output images and one feature vector. Note the generator has four image resolution levels. Not all levels are shown in the schema for conciseness. With such a model design, RPE cell borders in damaged regions within flatmount microscopy images can be effectively detected. *Conv*: Convolution layer; *DC*: Double convolution layers; *FC*: Fully connected layer; *MP*: Max-pooling layer; *MLP*: Multi-layer perceptron; *UC*: Up-sampling + convolution layer.

$V(\boldsymbol{s})$. $G1(\boldsymbol{s})$ from Decoder 1 represents a segmentation map, while $G2(\boldsymbol{s})$ from Decoder 2 is the translated image with enhanced RPE cell borders. The output from the feature extractor $V(\boldsymbol{s})$ is used for the contrastive representation learning in the model training. Although the generator has four resolution levels, not all levels are shown in Figure 5.2 for conciseness. At each image resolution level, the encoder convolves the image with a double convolution layer and next scales down the convolution response by a max-pooling layer. In the decoding analysis, an image representation is up-sampled, interpolated by a bilinear interpolation layer, and convolved with a double convolution layer in turn at each image resolution level. Additionally, the encoder outputs at different image resolution levels are processed by Multi-Layer Perceptron (MLP) modules, with the outcomes concatenated for the image feature vector construction.

To process two image outputs $G1(\boldsymbol{s})$ and $G2(\boldsymbol{s})$ from the generator, I include two corresponding discriminators $D1$ and $D2$. Each discriminator has multiple convolution layers and a fully connected output layer [45]. These discriminators help recognize the difference between generated and true images and thus force the generator to produce high-quality images similar to the true counterparts.

5.1.2 Model implementation

With training batches $\boldsymbol{P} \subseteq \mathcal{P}$, $\boldsymbol{X} \subseteq \mathcal{X}$, and $\boldsymbol{Y} \subseteq \mathcal{Y}$, I would like to achieve two training objectives on image segmentation and translation. 1) For a given image and its ground truth pair $(\boldsymbol{z}, \boldsymbol{w}) \sim \boldsymbol{P}$, the segmentation result $G1(\boldsymbol{z})$ from the generator is supposed to be similar to the segmentation ground truth $\boldsymbol{w}$ and indistinguishable by discriminator $D1$. 2) For RPE cells with weak (i.e., $\boldsymbol{x} \sim \boldsymbol{X}$) and strong borders (i.e., $\boldsymbol{y} \sim \boldsymbol{Y}$), the translated weak image is supposed to be $G2(\boldsymbol{x})$ indistinguishable by discriminator $D2$ and the translated strong image is supposed to keep intact, i.e., $G2(\boldsymbol{y}) \approx \boldsymbol{y}$. To achieve these training goals, the objective function for the GAN training strategy is defined as follows:

$$\mathcal{L}_{total} = (1 - \lambda) \mathcal{L}_s (\boldsymbol{P}) + \lambda \mathcal{L}_u (\boldsymbol{X}, \boldsymbol{Y}) \tag{5.1}$$

where two loss terms are balanced by the relative contribution factor λ.

This weight λ is dynamic and depends on the epoch number t:

$$\lambda(t) = \begin{cases} 1 & t \leq t_1 \\ 1 - \frac{1-c}{t_2-t_1} \cdot (t - t_1) & t_1 < t < t_2 \\ c & t \geq t_2 \end{cases} \tag{5.2}$$

where t_1 and t_2 are transient time cutoff values; The constant c is the weight factor after λ is stablized.

The first loss term $\mathcal{L}_s$ describes the similarity between the output of Decoder 1 and the segmentation ground truth:

$$\begin{aligned} \mathcal{L}_s(\boldsymbol{P}) &= \mathcal{L}_{s-GAN}(\boldsymbol{P}) + \lambda_1 \mathcal{L}_{s-idt}(\boldsymbol{P}) + \lambda_2 \mathcal{L}_{shape}(\boldsymbol{P}) \\ &= \mathbb{E}_{(\boldsymbol{z},\boldsymbol{w})\sim\boldsymbol{P}} \left(\log\left(1 - D1\left(G1\left(\boldsymbol{z}\right)\right)\right) + \log D1\left(\boldsymbol{w}\right)\right) \\ &+ \lambda_1 \mathbb{E}_{(\boldsymbol{z},\boldsymbol{w})\sim\boldsymbol{P}} \|G1(\boldsymbol{z}) - \boldsymbol{w}\|_1 \\ &+ \lambda_2 \mathbb{E}_{(\boldsymbol{z},\boldsymbol{w})\sim\boldsymbol{P}} \|\left(G1(\boldsymbol{z}) - \boldsymbol{w}\right) \cdot \boldsymbol{w}\|_1 \end{aligned} \tag{5.3}$$

In Eq. 5.3 the first two terms are the adversarial loss and the identity loss widely used in supervised GAN approaches [44, 45]. In this work, the exploratory experimental results suggest that the segmented RPE cell borders are often not closed, leading to a significantly different RPE cell topology. This artifact results from the fact that the misclassification of cell border pixels has a small influence on the identity loss that in turn is due to a small proportion of cell border pixels in an entire image. In favor of closed RPE cell contours in the segmentation results, cell border misclassification is penalized more by a shape loss (i.e., the third term in Equation 5.3). In the designed shape loss term, training attention of the developed model is directed to cell border misclassification events by multiplying the ground truth $\boldsymbol{w}$ to the difference between $G1(\boldsymbol{z})$ and $\boldsymbol{w}$. As border and background pixels in the ground truth take value 1 and 0, respectively, such a multiplication results in a focused attention to the misclassification on cell borders.

Similarly, the second loss term $\mathcal{L}_u$ characterizes the quality of gray-scale outputs from the generator:

$$\begin{aligned} \mathcal{L}_u(\boldsymbol{X}, \boldsymbol{Y}) = \mathcal{L}_{u\text{-}GAN}(\boldsymbol{X}, \boldsymbol{Y}) + \lambda_3 \mathcal{L}_{u-idt}(\boldsymbol{Y}) \\ + \lambda_4 \mathcal{L}_{NCE}(\boldsymbol{X}) \end{aligned} \tag{5.4}$$

In Eq. 5.4, the first term $\mathcal{L}_{u\text{-}GAN}(\boldsymbol{X}, \boldsymbol{Y})$ is the adversarial loss for unsupervised GAN learning that takes the following format:

$$\begin{aligned} \mathcal{L}_{u\text{-}GAN}(\boldsymbol{X}, \boldsymbol{Y}) = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}} \log\left(1 - D2\left(G2\left(\boldsymbol{x}\right)\right)\right) \\ + \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{Y}} \log D2(\boldsymbol{y}) \end{aligned} \tag{5.5}$$

The second term $\mathcal{L}_{u-idt}(\boldsymbol{Y})$ in Eq. 5.4 is the identity loss to retain strong images at the generator output in the unsupervised image translation. While I aim to transfer weak to strong images by the generator, I also would like to keep those strong images unchanged during the translation. Therefore, the identity loss $\mathcal{L}_{u-idt}(\boldsymbol{Y})$ is defined by:

$$\mathcal{L}_{u-idt}(\boldsymbol{Y}) = \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{Y}} \|G2(\boldsymbol{y}) - \boldsymbol{y}\|_1 \tag{5.6}$$

The third term $\mathcal{L}_{NCE}(\boldsymbol{X})$ in Equation 5.4 is Noised Contrastive Estimation (NCE) loss [125]. It aims to train an encoder that associates only corresponding inputs [126]. In the unsupervised GAN training, this NCE loss prevents a generator from randomly producing images with high quality in the target domain but irrelevant to inputs [48]. Applying this loss, I aim to achieve a high mutual information between an input $\boldsymbol{x}_i$ and its translation output $G2(\boldsymbol{x}_i)$, and a low mutual information between the input $\boldsymbol{x}_i$ and other translation outputs $G2(\boldsymbol{x}_j)$. Illustrated in Figure 5.2, encoded feature maps are processed with MLP modules for a vector representation $V(\boldsymbol{s})$. Let $\boldsymbol{v}_i = V(\boldsymbol{x}_i)$ and $\hat{\boldsymbol{v}}_i = V(G2(\boldsymbol{x}_i))$, the NCE loss contributed by image $\boldsymbol{x}_i$ is defined with a cross-entropy loss:

$$l_i = -\log \frac{e^{\boldsymbol{v}_i \cdot \hat{\boldsymbol{v}}_i / \tau}}{e^{\boldsymbol{v}_i \cdot \hat{\boldsymbol{v}}_i / \tau} + \sum_{j \neq i}^{N} e^{\boldsymbol{v}_i \cdot \hat{\boldsymbol{v}}_j / \tau}} \tag{5.7}$$

where τ is the scaling factor.

The resulting NCE loss for a training batch $\boldsymbol{X}$ is defined as:

$$\mathcal{L}_{NCE} = \frac{1}{N} \sum_{i=1}^{N} l_i \tag{5.8}$$

I implement the segmentation model with Python 3.8 programming language and PyTorch 1.8.1 deep learning framework [111] and run the segmentation analysis with two NVIDIA Tesla K80 GPUs. Balancing the tradeoff between computational efficiency and deep network efficacy, four image resolution levels are designed in the generator, with 64, 128, 256, and 512 filters from the highest to the lowest level, respectively. Each MLP embedding image features from the Encoder has two neural network layers, with 256 units at each layer. For discriminators, two convolutional neural networks are included, both including the same resolution levels and filter numbers as the generator. Instead of max-pooling layers, image representations in discriminators are down-sampled by convolution layers of stride 2. For training parameters, I have $t_1 = 40$, $t_2 = 70$, $c = 0.7$, $\lambda_1 = \lambda_2 = 0.5$, $\lambda_3 = \lambda_4 = 1$, and the scaling factor $\tau = 0.07$ determined empirically. In each training epoch, 24 image patches from each training set (i.e., the annotated set $\mathcal{P}$, the negative set $\mathcal{X}$, and the positive set $\mathcal{Y}$) are randomly selected and cropped into image regions of size 64 × 64 pixels from these patches as the augmented training batch.

5.1.3 Evaluation metrics

In this work, border and background pixels are set as positive and negative classes, respectively. I derived metrics from the confusion matrix for model evaluations

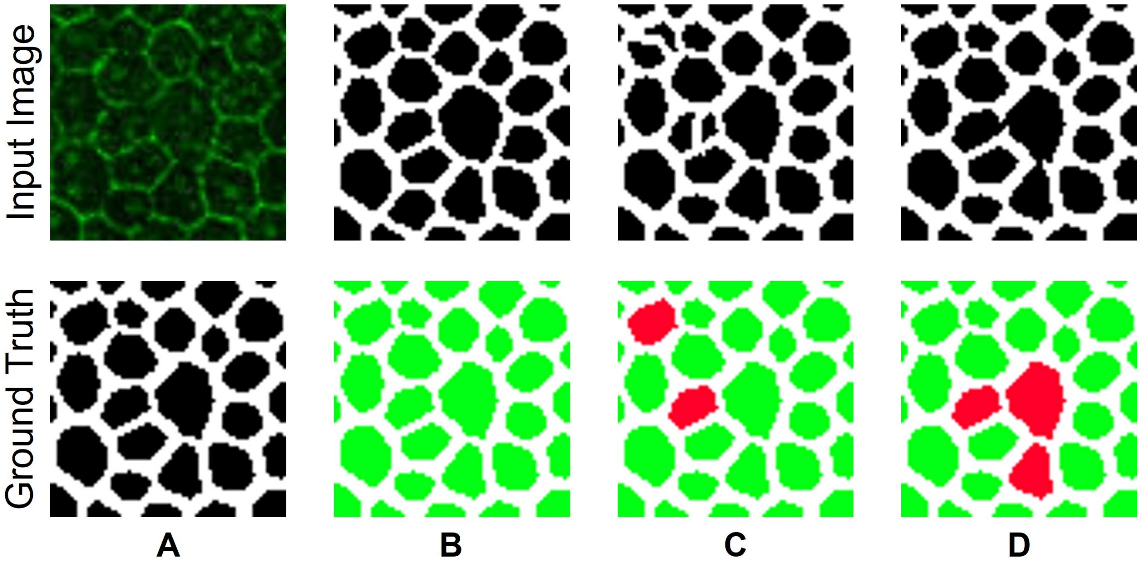

Figure 5.3: Evaluation of segmentation results. (A) A representative image (top) and its ground truth for segmentation (bottom) are presented. The (B) correct, (C) over-, and (D) under-segmentation results are illustrated, respectively. In (B-D), the top images present the segmentation results, while the bottom images highlight regions with erroneous segmentation results (red).

[61, 63, 62]. The confusion matrix results in TP (number of correctly classified border pixels), FP (number of incorrectly classified background pixels), FN (number of incorrectly classified border pixels), and TN (number of correctly classified background pixels). As TN is much larger than the other three in practice, I do not use accuracy ACC=(TP+TN)/(TP+FP+FN+TN)) as a metric. Instead, I adopt precision P=TP/(TP+FP) and recall R=TP/(TP+FN) for model evaluation.

Additionally, I introduce metrics to indicate RPE cell topology. For each cell with ground truth, its Intersection over Union (IoU) is computed with all overlapped cells predicted from models. If a cell has an IoU larger than 0.5, it is marked as a correct hit (CH); otherwise, it is a wrong hit (WH) as presented in Figure 5.3. The Correct Rate (CR) of segmentation is defined as:

$$\mathrm{CR} = \frac{|\mathrm{CH}|}{|\mathrm{CH}| + |\mathrm{WH}|} \tag{5.9}$$

where $|\cdot|$ represents the set size.

To penalize wrong segmentation by the cell size, I also use such information to calculate the Weighted Correct Rate (WCR) of segmentation:

$$\mathrm{WCR} = \frac{\sum_{r \in \mathrm{CH}} |r|}{\sum_{r \in \mathrm{CH}} |r| + \sum_{r \in \mathrm{WH}} |r|} \tag{5.10}$$

5.2 Results

5.2.1 Training and testing datasets

In this study, the mouse RPE flatmount images are selected from our reference database [127, 128]. As these RPE images have high resolutions (around 4,000 × 4,000 pixels each), each image is divided into non-overlapping patches. Both a small set of annotated and a large set of unlabeled RPE cells are included in the training set. The annotated set $\mathcal{P} = \{(z_i, w_i)\}$ includes patch pairs where z_i and w_i are the image patch and manually annotated ground truth, respectively. The large number of unlabeled RPE cells for training come from a negative and a positive set. The negative set $\mathcal{X} = \{x_i\}$ includes image patches of RPE cells with as many weak or missing borders as possible, while the positive set $\mathcal{Y} = \{y_i\}$ includes image patches of only RPE cells with strong borders. As RPE cells with weak or missing borders are often spatially mixed with those presenting strong borders, it becomes challenging to find the positive training set including only RPE cells with strong borders and the negative training set including as many RPE cells with weak or missing borders as possible when the patch size is unduly large. After multiple experiments, the size of each image patch is set to 96 × 96 in pixels. The number of image patches in training set $\mathcal{P}$, $\mathcal{X}$, and $\mathcal{Y}$ is 155, 987, and 653, respectively. For the testing set, I include 34,742 RPE cells with weak or missing cell borders from 200 image patches.

5.2.2 Deep learning model validation

To validate the method performance, I compare the developed method MultiHeadGAN with four state-of-the-art models, including FCN [61], DeepLab [62], UNet [63], and Cellpose [124]. FCN, DeepLab, and UNet have been widely applied to a large number of biomedical image segmentation applications [98]. Cellpose is a pre-trained cell segmentation model built on UNet. It is trained to predict gradient vector fields and produce segmentation results by gradient tracking. For fair comparisons, CycleGAN [45] and CUT [48] are used to enhance the RPE cell border contrast before UNet is used for segmentation. While FCN, DeepLab and UNet are trained with the annotated training set $\mathcal{P}$, CycleGAN and CUT model are trained with the unlabeled training set $\mathcal{X}$ and $\mathcal{Y}$. Typical tissue segmentation results of these models are presented in Figure 5.4. By visual comparisons, the results from FCN, DeepLab and UNet have large under-segmented regions. CycleGAN and CUT can effectively mitigate the degree of under-segmentation, while the developed MultiHeadGAN model achieves the best segmentation results. Although Cellpose can generate separated cell masks, its performance highly depends on the gradient vectors in cells. As not all cells present convergent gradient fields, Cellpose can fail in these cases.

I quantitatively evaluate segmentation results by Correct Rate (CR), Weighted Correct Rate (WCR), Precision (P) and Recall (R). Demonstrated in Table 5.1, the developed MultiHeadGAN achieves the best performance with 85.4% (CR), 88.8% (WCR), 87.3% (Precision) and 80.1% (Recall), respectively. In Figure 5.5, quantitative evaluation results are plotted to present the statistical difference between the developed approach and others. Noticeably, MultiHeadGAN presents fewer outliers and a smaller variation, implying its strong stability. By metrics of WCR, P and R, MultiHeadGAN is significantly better than all other approaches. By CR, it is also significantly better than all other approaches except Cellpose.

Additionally, I test all approaches on a human RPE flatmount image dataset. Each

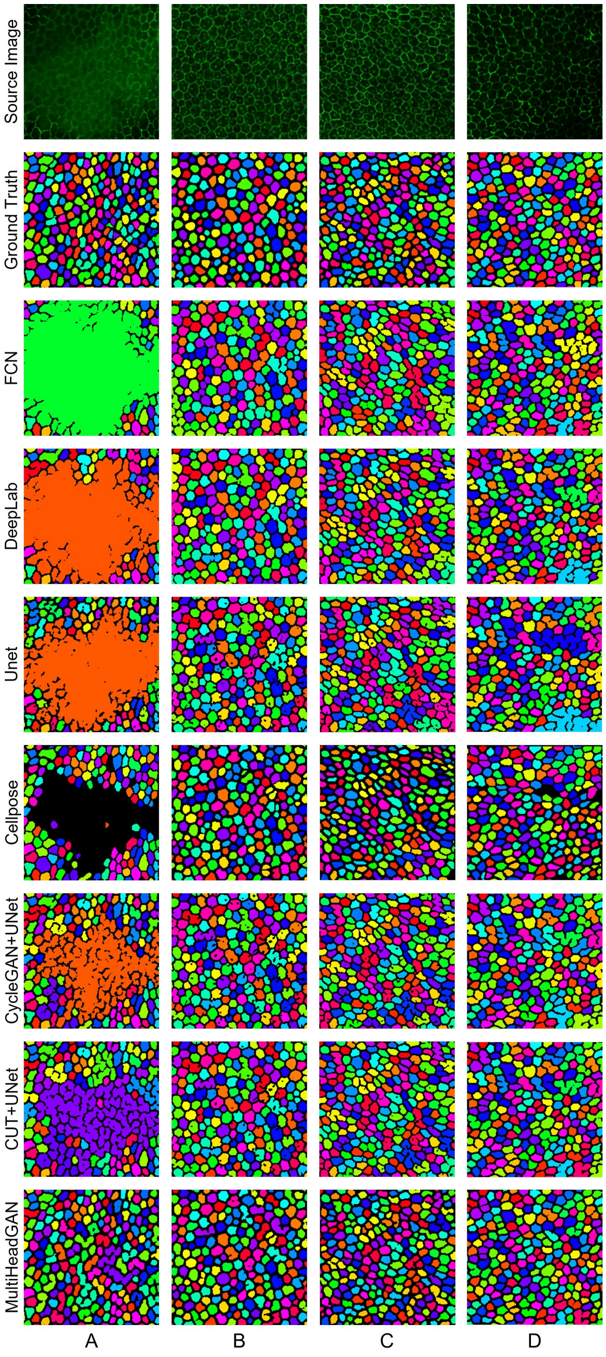

Figure 5.4: Qualitative comparison of deep learning approaches for RPE cell segmentation with flatmount microscopy images. Four typical impaired image regions are shown in columns (A-D) with rows for ground truth and corresponding segmentation results of FCN, DeepLab, UNet, Cellpose, CycleGAN+UNet, CUT+UNet, and the developed MultiHeadGAN, respectively. Column (A) demonstrates the case that the whole region is severely blurred, while columns (B-D) present cases that cell borders are partially missing

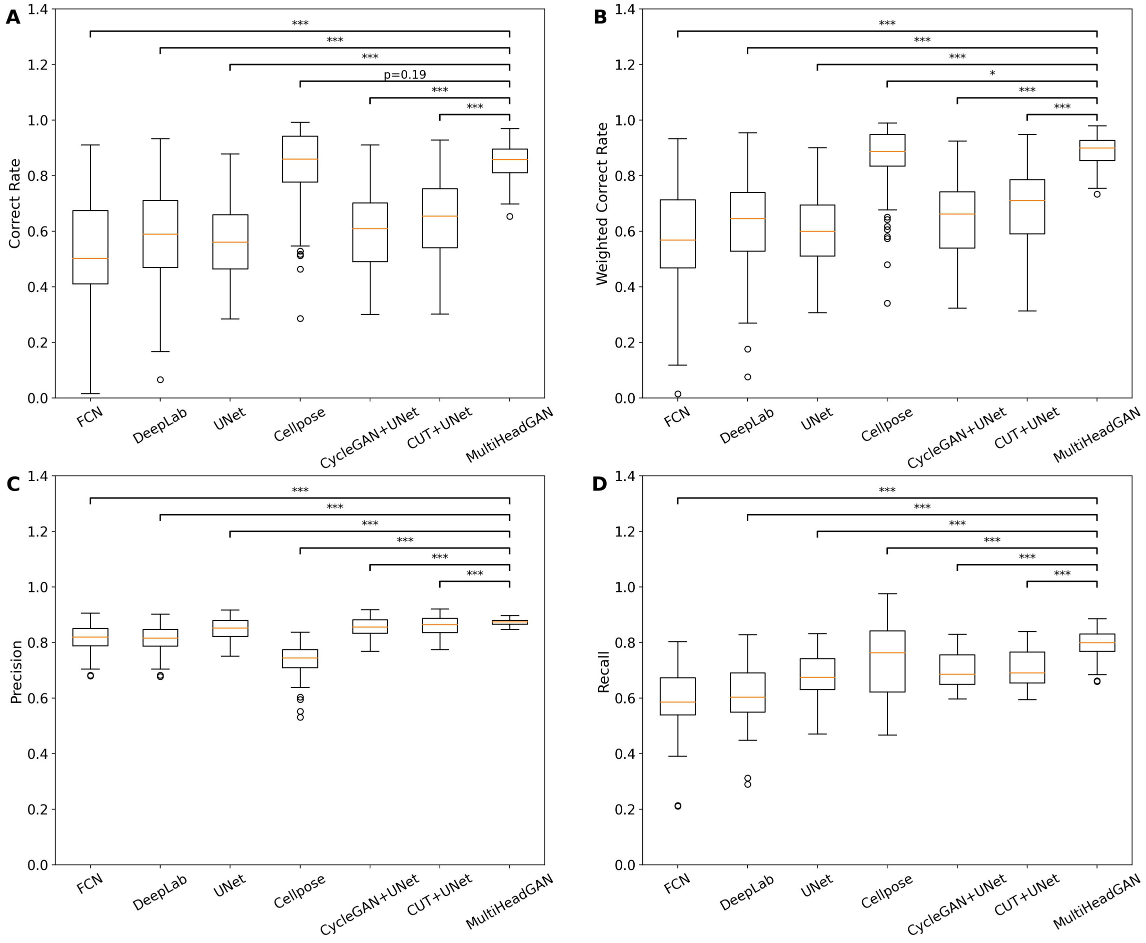

Figure 5.5: Quantitative comparison of deep learning approaches for RPE flatmount image segmentation. RPE cell segmentation performance of deep learning approaches is compared by (A) Correct Rate, (B) Weighted Correct Rate, (C) Precision, and (D) Recall. Paired sample *t*-tests between the developed MultiHeadGAN and other six state-of-the-art approaches suggest a statistically significant performance difference. The notations for *, **, and *** represent a *p*-value less than 0.05, 0.005, and 0.0005, respectively.

image has 1,024×1,024 pixels by size. The training and testing data include 14 and 16 human samples. The human training set is used for transfer learning with FCN, DeepLab, UNet and MultiHeadGAN. The resulting method performances are shown in Table 5.1. As most cell borders in the human dataset are strong, all methods for comparison present no significant difference. The developed approach achieves the best performance by Precision (98.9%). By CR, WCR, and Recall, all methods

Table 5.1: Quantitative performance comparison between the developed MultiHeadGAN and other state-of-the-art deep learning approaches by different evaluation metrics on the mouse and human dataset. *Enh*: Methods for image enhancement; *Seg*: Methods for image segmentation *L*: Training data with ground truth (i.e., $\mathcal{P}$); *UL*: Unlabeled training data (i.e., $\mathcal{X}$, $\mathcal{Y}$); *CR*: Correct Rate; *WCR*: Weighted Correct Rate; *P*: Precision; *R*: Recall. All value units are in percentage (%).Included and excluded training data are checked by “✓” and “✗”, respectively. The absence is represented by “/”.

Method		Data		Mouse				Human			
Enh	*Seg*	L	UL	CR	WCR	P	R	CR	WCR	P	R
/	FCN	✓	✗	52.5	57.2	82.0	59.8	95.3	89.4	96.1	**95.0**
/	DeepLab	✓	✗	58.1	62.9	81.5	61.3	94.9	**89.6**	96.3	94.9
/	UNet	✓	✗	56.1	60.1	85.0	68.5	92.4	85.2	97.9	90.6
/	Cellpose	/	/	83.8	86.7	73.8	73.7	**97.1**	87.9	95.1	91.4
CycleGAN	UNet	✓	✓	59.9	64.2	85.7	70.0	96.7	88.8	**98.9**	92.4
CUT	UNet	✓	✓	64.5	68.8	86.1	70.3	96.9	89.0	**98.9**	93.0
MultiHeadGAN		✓	✓	**85.4**	**88.8**	**87.3**	**80.1**	96.5	88.8	**98.9**	92.7

Table 5.2: Quantitative performance comparison across different training loss combinations by different evaluation metrics on the mouse dataset.*CR*: Correct Rate; *WCR*: Weighted Correct Rate; *P*: Precision; *R*: Recall. All value units are in percentage (%). Included and excluded loss terms are checked by "✓" and "✗", respectively.

Index	Loss Term						CR	WCR	P	R
	$\mathcal{L}_{s-idt}$	$\mathcal{L}_{s-GAN}$	$\mathcal{L}_{u-GAN}$	$\mathcal{L}_{u-idt}$	$\mathcal{L}_{NCE}$	$\mathcal{L}_{shape}$				
1	✓	✗	✗	✗	✗	✗	56.1	60.1	85.0	68.5
2	✓	✓	✗	✗	✗	✗	59.4	63.4	**88.7**	58.7
3	✓	✓	✗	✗	✗	✓	66.4	70.5	87.0	57.4
4	✓	✓	✓	✗	✗	✗	53.4	57.2	86.8	60.4
5	✓	✓	✓	✓	✗	✗	67.7	71.3	86.7	59.4
6	✓	✓	✓	✓	✓	✗	80.9	84.2	86.7	78.9
7	✓	✓	✓	✓	✓	✓	**85.4**	**88.8**	87.3	**80.7**

present similar performances. The difference between the developed approach and the best approach is only 0.6%, 0.6%, and 2.3%, respectively.

5.2.3 Ablation study

To demonstrate the contribution from individual training losses in the developed MultiHeadGAN model, I carry out ablation experiments with seven loss combinations. The resulting performances are presented in Table 5.2. Loss combination 7 is for the

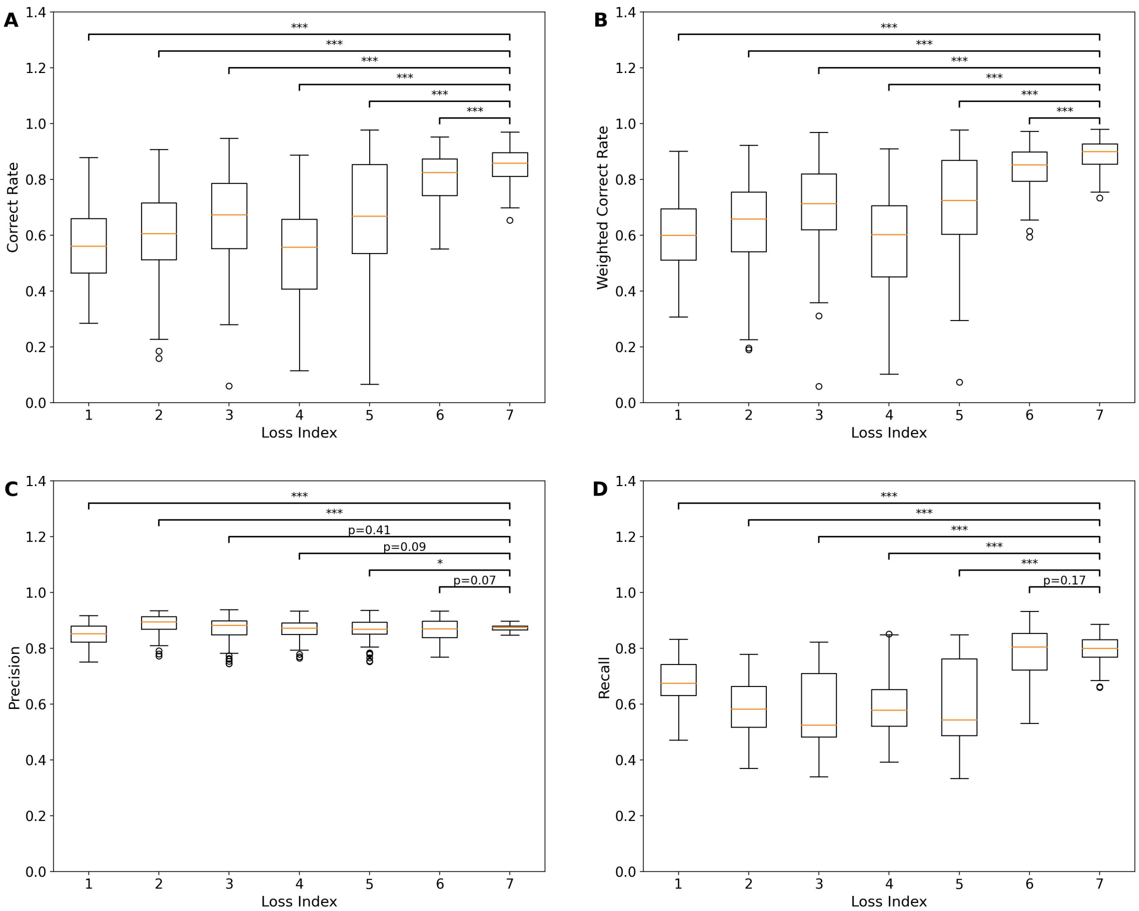

Figure 5.6: Quantitative comparison of ablated models for RPE flatmount image segmentation. The segmentation performance of the model with different training loss combinations is compared by (A) Correct Rate, (B) Weighted Correct Rate, (C) Precision, and (D) Recall. The x-axis represents the loss combination index given in Table 5.2. Paired sample t-tests between the developed combination for MultiHeadGAN and other six loss combinations suggest a statistically significant performance difference in most cases. The notations for *, **, and *** represent a p-value less than 0.05, 0.005, and 0.0005, respectively.

MultiHeadGAN model with all losses and achieves the best performance. Comparisons between loss combination 1 and 2 imply that the GAN mechanism improves supervised training outcome. Furthermore, comparisons between loss combination 2 and 3 suggest that the designed shape loss can boost the performance with annotated training data. Comparisons between loss combination 2 and 4 indicate that the addition of unsupervised learning to deep learning model training may degrade the model

Table 5.3: Quantitative performance comparison with different weight factor λ strategies by different evaluation metrics on the mouse dataset. textitCR: Correct Rate; *WCR*: Weighted Correct Rate; *P*: Precision; *R*: Recall. All value units are in percentage (%).

t_1	t_2	c	CR	WCR	P	R
0	0	0.5	48.9	53.3	**88.4**	58.0
10	40		70.2	74.4	86.5	64.7
40	70		79.6	83.2	83.4	71.7
70	100		66.3	70.2	85.5	65.4
0	0	0.7	50.6	55.0	87.8	58.7
10	40		74.5	78.7	86.7	64.9
40	70		**85.4**	**88.8**	87.3	**80.1**
70	100		73.3	77.2	84.3	65.4

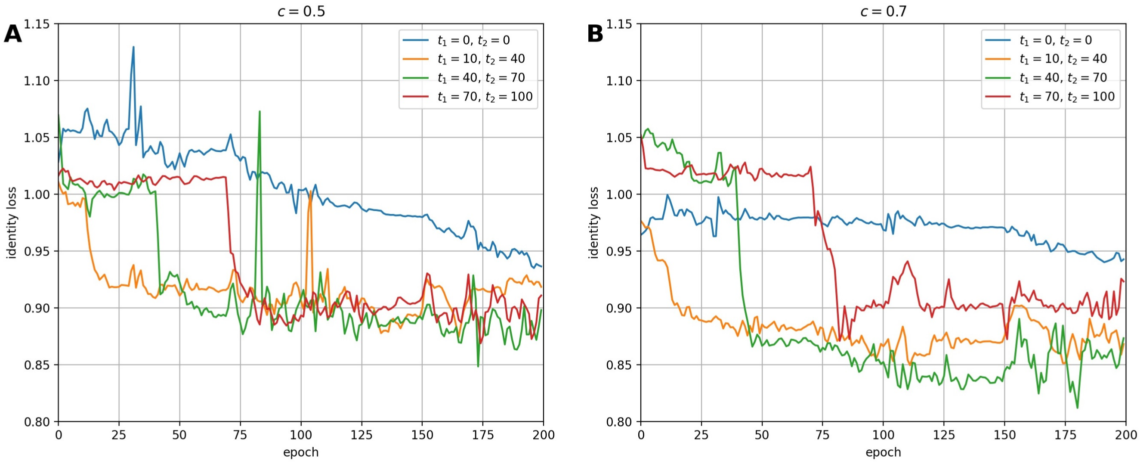

Figure 5.7: Comparison of the model training curves with different strategies for the weight factor λ. Training curves with stabilized weight factor $c = 0.5$ and $c = 0.7$ are presented in (A) and (B), respectively.

performance without use of appropriate constraints (i.e., identity loss and NCE loss). Figure 5.6 suggests that the adopted loss combination 7 statistically outperforms other loss combinations by CR and WCR.

Note that a dynamic weight factor λ is adopted to balance the supervised and unsupervised learning. To validate the effectiveness of this design and determine corresponding parameters, I have carried out ablation experiments. In Figure 5.7, the training curves of identity loss for the supervised learning suggest that the parameter

Table 5.4: Quantitative performance comparison with different training image and batch sizes on the mouse dataset. *CR*: Correct Rate; *WCR*: Weighted Correct Rate; *P*: Precision; *R*: Recall. All value units are in percentage (%).

Image size	Batch size	CR	WCR	P	R
32	36	58.9	63.3	85.0	70.1
64	24	**85.4**	**88.8**	**87.3**	**80.1**
96	12	65.4	69.6	86.4	61.2

setting with $t_1 = 40$, $t_2 = 70$, $c = 0.7$ achieves the best training performance. Furthermore, Table 5.3 compares the testing performance in . With a constant weight factor λ (i.e., $t_1 = t_2 = 0$), the model is trained with both supervised and unsupervised learning. The results suggest that a fixed λ has the worst performance across dynamic factor strategies for each given stabilized weight factor c. Next, I begin model training only with the unsupervised learning and linearly change the weight factor λ for a semi-supervised learning. The stage only with unsupervised learning is defined as pre-training. The performance is improved as the pre-training time increases. When the pre-training time is increased to 40 epochs, the developed model achieves the best performance. When the pre-training time is increased further, its performance becomes worse in large part due to overfitting in the unsupervised learning. After systematic investigations, I choose parameters $t_1 = 40$, $t_2 = 70$, $c = 0.7$ for the dynamic weight factor strategy by the experiment results.

Note that the performance of contrastive learning methods utilizing negative examples suffers from a small batch size [33]. Therefore, a reasonably large batch size is preferred in this study. Given a large batch size and a constant GPU memory, the resulting individual image region is limited to a small size. However, the selection of an unduly small image region size is subject to cell topological information loss. The best image region size is determined by ablation experiments. By Table 5.4, the best model performance is achieved when image region size is 64 × 64 in pixels with the batch size 24 in this study.

5.3 Discussion and summary

RPE cell morphological information plays a vital role to facilitate a better understanding of RPE aging and physiology. An accurate morphological characterization, however, is highly dependent on a high-quality segmentation. Due to lack of appropriate computational methods, the segmentation maps are often created by manual annotations, suffering from a large inter- and intra-variability. Additionally, such a human annotation process is time-consuming and insufficient to produce an adequately large number of annotated RPE images from impaired regions to support computer-based method training.

To address this challenge, a novel method (MultiHeadGAN) for segmenting RPE cells from 2D RPE flatmount microscopy images is developed in this study. The developed method takes a semi-supervised learning strategy and enables deep neural networks to learn from a small set of annotated and a large set of unlabeled image data, resulting in a more generalized feature extraction ability and learning outcome. The resulting deep neural network is designed within a GAN training framework equipped with one encoder, two decoders, and one feature extractor in the generator. Two heads are created in the generator for the contrast enhanced grayscale and binary segmentation output, respectively. Correspondingly, there are two discriminators in the developed model that force the generator to output images with strong borders. Additionally, a new shape loss term is created to encourages the model to produce closed cell borders.

Demonstrated in Figure 5.3, very few mis-classified pixels can either connect separate cells or divide a single cell, leading to a huge change in the RPE cell topology. two new metrics, i.e., Correct Rate (CR) and Weighted Correct Rate (WCR), are designed to better characterize such cell topology. CR represents the proportion of correctly segmented cells in the total cell population, while WCR indicates the proportion of areas of correctly segmented cells in the total cell area. As shown in Figure

Table 5.5: Comparison of deep learning segmentation models by model parameter number, Floating Point Operations Per Second (FLOPS), and the average processing time cost on a 96×96 image patch.

Model	Parameter number	FLOPS	Processing time (ms)
FCN	51.9M	7.6G	134.6
DeepLab	58.6M	8.5G	167.8
UNet	34.5M	9.2G	92.9
Cellpose	**6.6M**	**2.9G**	3,603.6
CycleGAN+UNet	43.0M	16.1G	132.6
CUT+UNet	43.0M	16.0G	125.1
MultiHeadGAN	8.5M	7.0G	**35.9**

5.5, the differences across different methods by CR and WCR are much larger than those suggested by Precision and Recall. CR and WCR can better reflect performance difference supported by the visual results in Figure 5.4.

I have systematically compared the developed MultiHeadGAN model with other state-of-the-art deep learning methods. MultiHeadGAN manifests a superior performance by RPE cell segmentation accuracy, model complexity, and computational efficiency. By both visual and quantitative evaluations, promising segmentation performance by MultiHeadGAN is demonstrated. In the ablation study, the contribution from each loss term is systematically compared and analyzed. Although the developed approach takes a semi-supervised training strategy for multiple network components in a GAN framework, only Encoder and Decoder 1 is necessary in the testing stage. As shown in Table 5.5, MultiHeadGAN includes the least parameters and requires the least Floating Point Operations Per Second (FLOPS) for processing except Cellpose. Although Cellpose has the least number of parameters, it requires a time-consuming gradient tracking process, resulting in the longest average processing time (3,604 ms). Additionally, models of CycleGAN+Unet and CUT+Unet leverage the GAN mechanism and include more parameters, leading to a slower analysis. By contrast, the developed MultiHeadGAN has the lowest average processing time cost (35.9 ms), promising to support large-scale RPE image data analysis.

Chapter 6

Biomedical Image Segmentation with Self-supervised Learning

With the rapid development of medical imaging technology in the past decade, computational image analysis has become a significant tool to facilitate a large number of image-related biomedical investigations. Images produced by various image acquisition techniques, e.g., Computed Tomography (CT), Magnetic Resonance (MR), Positron Emission Tomography (PET), ultrasound, X-ray, and histopathology microscopy, are widely used for the early detection, diagnosis, and treatment response assessment of numerous diseases [1].

The pigmented cell layer between the choroid and the neurosensory retina is known as the Retinal Pigment Epithelium (RPE). Its primary functions include secreting immunosuppressive factors, maintaining photoreceptor excitability, and transporting nutrients [114]. Therefore, RPE aging often results in secondary photoreceptor degradation, which ultimately leads to irreversible vision loss in turn. RPE cell morphological characteristics, such as area, perimeter, and aspect ratio, have previously been proven as good indicators of the cell pathophysiologic status and the degree of RPE aging [7, 116, 117]. RPE flatmount fluorescent microscopy images have been widely used to calculate RPE cell morphological features. Before cell morphological features can be computed, cell borders have to be detected from flatmount images. Some typical RPE cell examples in flatmount images are illustrated in Figure 6.1A. However, the flatmount image acquisition procedure inevitably produces damaged image regions where the RPE cells are often degraded by noise. Some RPE cells with blurred or missing cell borders from damaged image regions are presented in Figure 6.1B. To facilitate an accurate cell border recovery, I would like to suppress the nucleus contrast in the resulting flatmount images. In practice, the imperfect tissue preparation, fluorescent protein labeling, and imaging process often produce RPE cells with highlighted nuclei in practice (Figure 6.1C). As the manual annotation process for such degraded image regions are time-consuming and suffers from a large inter- and intra-variability, annotated training data is hardly adequate to support supervised learning methods. In the previous work [6], I developed a deep neural network for RPE cell border detection that was trained with a semi-supervised learning strategy. With this approach, the training set was further enriched by unlabeled data. The trained model, thus, benefited from such an increase in training data scale and sample diversity. However, its performance is still subject to the limited labeled data in the training dataset.

To address this problem, I present in this work a novel **S**elf-**S**upervised **S**emantic **S**egmentation (S^4) method that leverages a self-supervised learning strategy. Note that this RPE cell segmentation method only requires unlabeled flatmount image data to train deep neural networks. Specifically, the reconstruction and pairwise representation loss are utilized to train an encoder-decoder architecture that recovers degraded image regions. For the pairwise representation learning, a new image augmentation algorithm (AugCut) is used to generate correlated views of input images. Moreover, I formulate a novel morphology loss that incentivizes the network to generate binary outputs with closed cell borders. The developed approach is compared

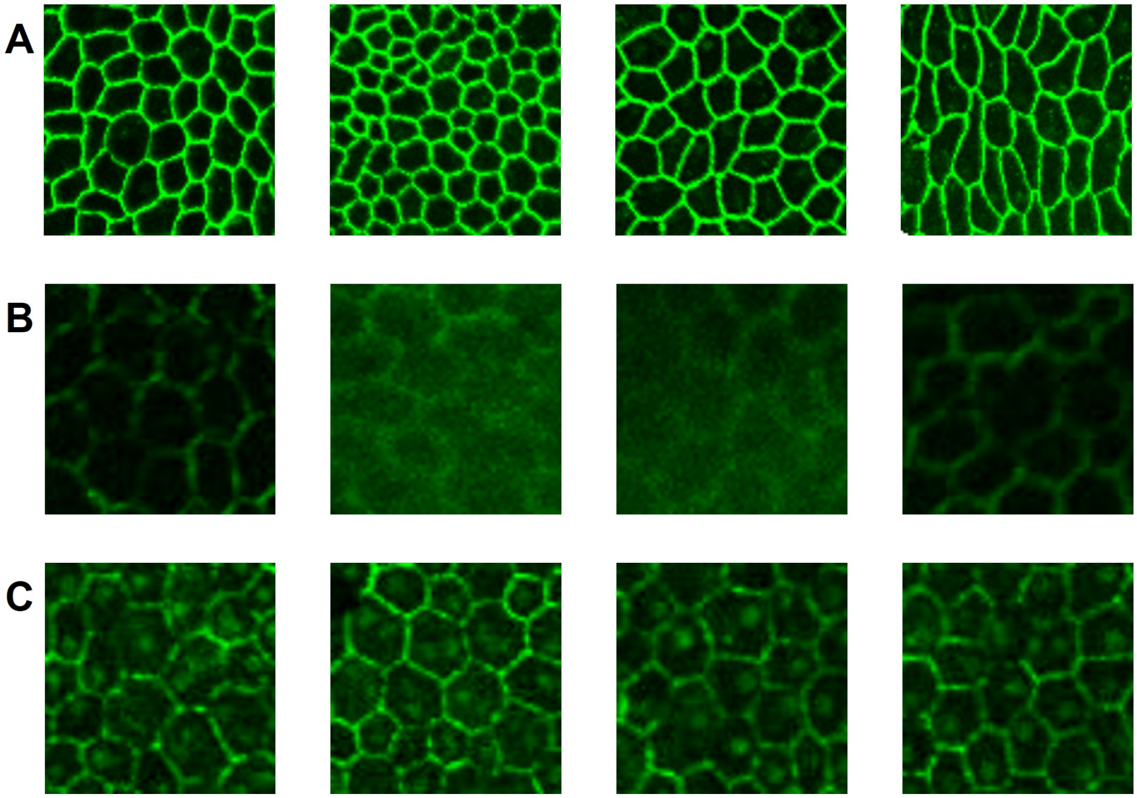

Figure 6.1: Representative RPE flatmount fluorescent microscopy image regions. (A) RPE cells in normal regions often have cell borders with a high contrast. (B) RPE cells in damaged regions present weak or missing cell borders with a partial or complete cell structure loss. (C) RPE cells in damaged regions can also present cell nuclei, making it more challenging for accurate RPE cell segmentation.

with the state-of-the-art deep learning approaches and demonstrates its superior performance for RPE cell segmentation with flatmount microscopy images. Ablation experiments present the necessity and efficacy of the developed training strategy design. With extensive tests and rigorous comparisons, the experimental performance of the developed deep learning method suggests its promising potential to support large-scale cell morphological analyses for RPE aging study.

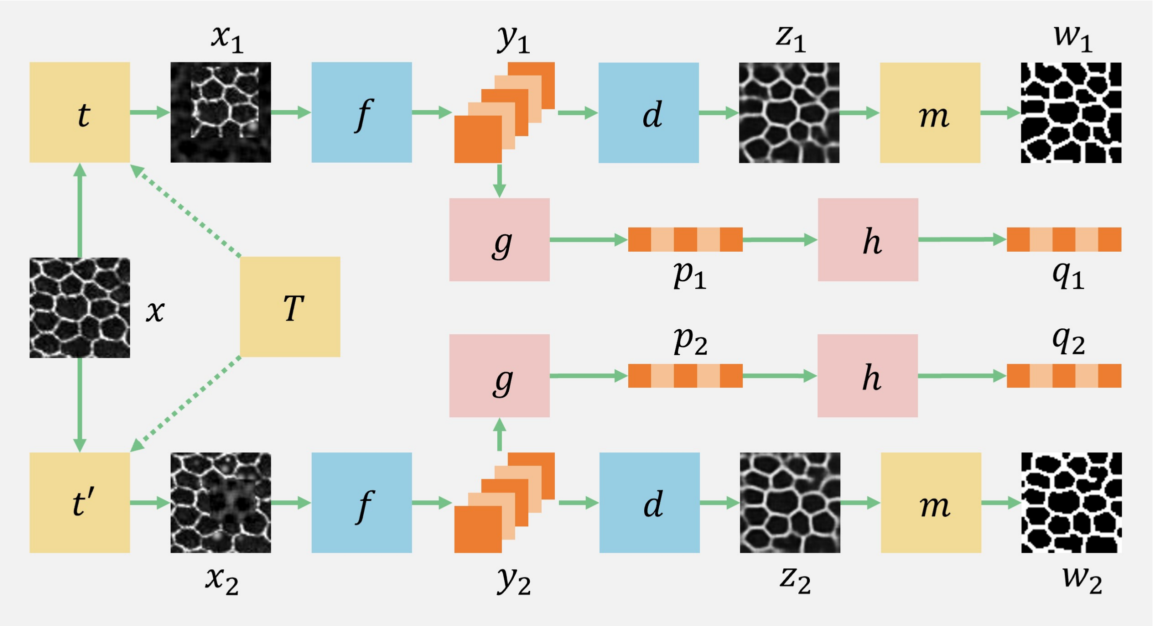

Figure 6.2: Overall schema of the developed self-supervised learning method S^4. Blue, pink, and yellow blocks represent convolutional layers, MLP layers, and traditional image processing methods, respectively. For each image input x, two different augmented views x_1 and x_2 are generated with operator t and t' randomly sampled from augmentation operator family T. Next, x_1 and x_2 are individually processed by the following convolutional and MLP layers for pairwise representation learning. The resulting latent feature vectors (p_1, p_2, q_1, q_2) are used for pairwise representation loss. Additionally, x_1 and x_2 are processed by an encoder-decoder network and morphology operations with output images (z_1, z_2, w_1, w_2) for reconstruction loss and morphological loss computation.

6.1 Methods

6.1.1 Deep neural network architecture

An overview of the developed self-supervised learning method S^4 is illustrated in Figure 6.2. In the training stage, each input image x, is transformed into two related views (i.e., x_1 and x_2) by two image augmentation operators randomly sampled from the augmentation operator family T. Next, an encoder network f down-samples x_1 and x_2 into latent image representations (i.e., y_1 and y_2) that are further transformed into latent representation vectors (i.e., p_1 and p_2) by a Multi-Layer Perceptron (MLP) projection head g. Another MLP prediction head h is used to match the mapped representation from one view to another latent representation vector (i.e., p_1 matching $q_2 = h(p_2)$, while p_2 matching $q_1 = h(p_1)$) [129]. The similarity between two views can be evaluated by the cosine similarity:

$$\mathcal{D}(q_1, p_2) = \frac{q_1 \cdot p_2}{\|q_1\|_2 \|p_2\|_2} \tag{6.1}$$

Prior studies have shown the necessity of using the stop-gradient operation, denoted as sg, in the pairwise learning [129, 130]. For symmetry, the pairwise representation loss is defined as:

$$\mathcal{L}_{PR} = -\frac{1}{2}\mathcal{D}\left(q_1, sg\left(p_2\right)\right) - \frac{1}{2}\mathcal{D}\left(q_2, sg\left(p_1\right)\right) \tag{6.2}$$

As the defined pairwise representation loss guides the encoder f to extract structural information from impaired image regions, I also need loss functions to train the decoder network d. I denote the outputs of the decoder on two branches as $z_1 = d(y_1)$ and $z_2 = d(y_2)$, respectively. The performance of the decoder can be improved by minimizing the difference between input x and outputs z_1, z_2:

$$\mathcal{L}_{Rec\_i} = \frac{1}{2}\|x - z_1\|_1 + \frac{1}{2}\|x - z_2\|_1 \tag{6.3}$$

In addition to the comparison between input and outputs, the reconstruction quality of the network can be evaluated by the difference between outputs z_1, z_2:

$$\mathcal{L}_{Rec\_o} = \|z_1 - z_2\|_1 \tag{6.4}$$

Combining loss term from Eq. 6.3 and Eq. 6.4, the reconstruction loss is defined as:

$$\mathcal{L}_{Rec} = \lambda_1 \mathcal{L}_{Rec\_i} + (1 - \lambda_1)\mathcal{L}_{Rec\_o} \tag{6.5}$$

where λ_1 is a weight factor ranging from 0 to 1.

In the ideal situation, the reconstruction loss $\mathcal{L}_Rec$ can be reduced to zero and outputs z_1, z_2 are exactly the same as input x. However, my goal is to train a network that can generate a binary segmentation output for each input image. Therefore, I utilize morphological transformations m to produce a binary map w for z and design a morphology loss as a function of the difference between w and z. By minimizing this difference, the network can be guided to generate binary segmentation maps. The morphology loss term is defined as:

$$\mathcal{L}_{Mor} = \frac{1}{2}\left\|w_1 - z_1\right\|_1 + \frac{1}{2}\left\|w_2 - z_2\right\|_1 \tag{6.6}$$

where $w_1 = m(z_1)$ and $w_2 = m(z_2)$.

Morphological transformation steps for the network training are illustrated in Figure 6.3. First, the output z from the decoder is binarized by the adaptive thresholding [131]. Next, the holes are removed by extracting and filling the external contours. An image opening operation with a 3×3 structural element is used to connect borders. Finally, I reverse the image and remove small regions with area less than 10 pixels.

Finally, the overall loss function of the method is formulated as:

$$\mathcal{L} = \mathcal{L}_{PR} + \lambda_2 \mathcal{L}_{Rec} + \lambda_3 \mathcal{L}_{Mor} \tag{6.7}$$

where λ_2 and λ_3 are loss weight factors.

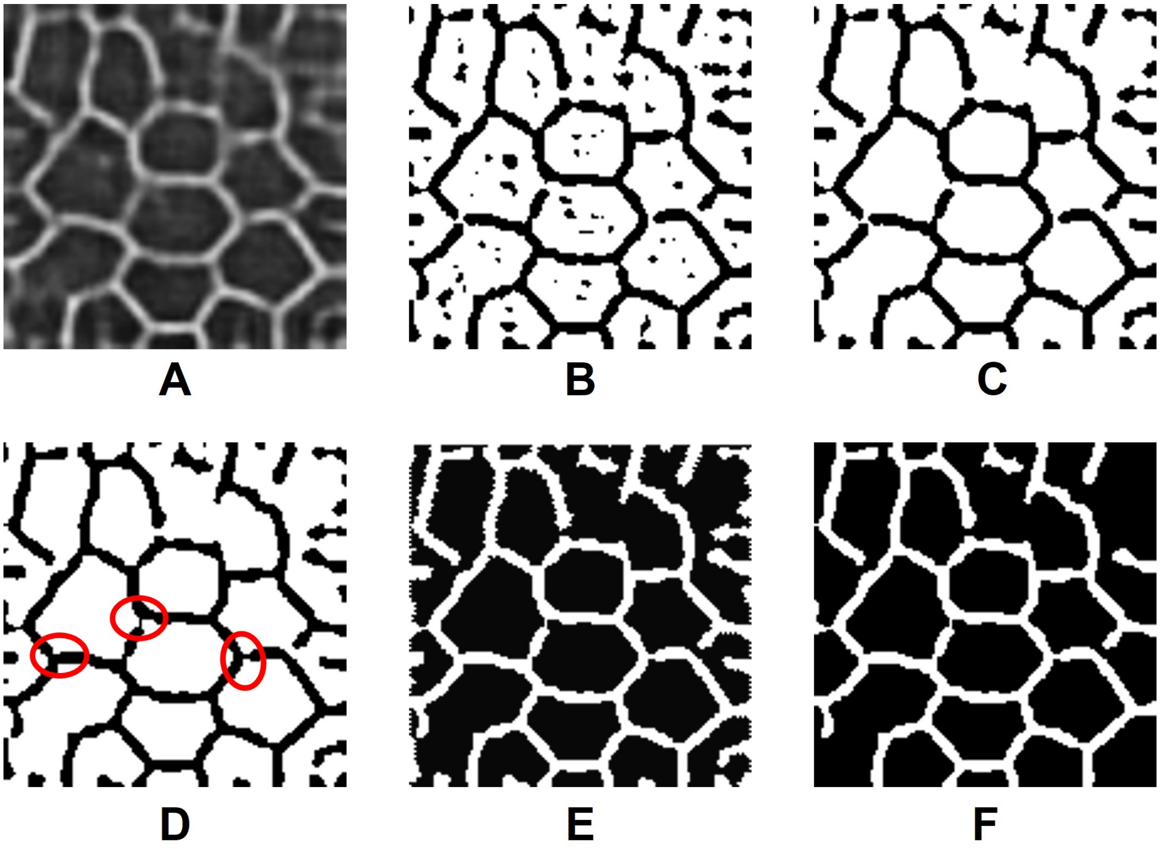

Figure 6.3: A typical example of the morphological transformation process. (A) A representative output image example z from the decoder; (B) The binarized result after the adaptive thresholding; (C) Hole exclusion by filling external contours; (D) Border connections by the image opening operation (red circles); (E) The reversed image; (F) Exclusions of small regions.

6.1.2 Image augmentation

The developed augmentation algorithm AugCut consists of two augmentation branches (Figure 6.4A). To emulate RPE cells in damaged tissue regions, augmented images are produced from training images with clear cell borders sampled from the dataset. On the top branch, input images are corrupted by T_1 with random Gaussian noise, Gaussian blur, and brightness reduction. This branch imitates the images of damaged tissue regions in Figure 6.1B where cell borders are blurred or even missing. By contrast, Gaussian distributed blobs are added to input images at random locations

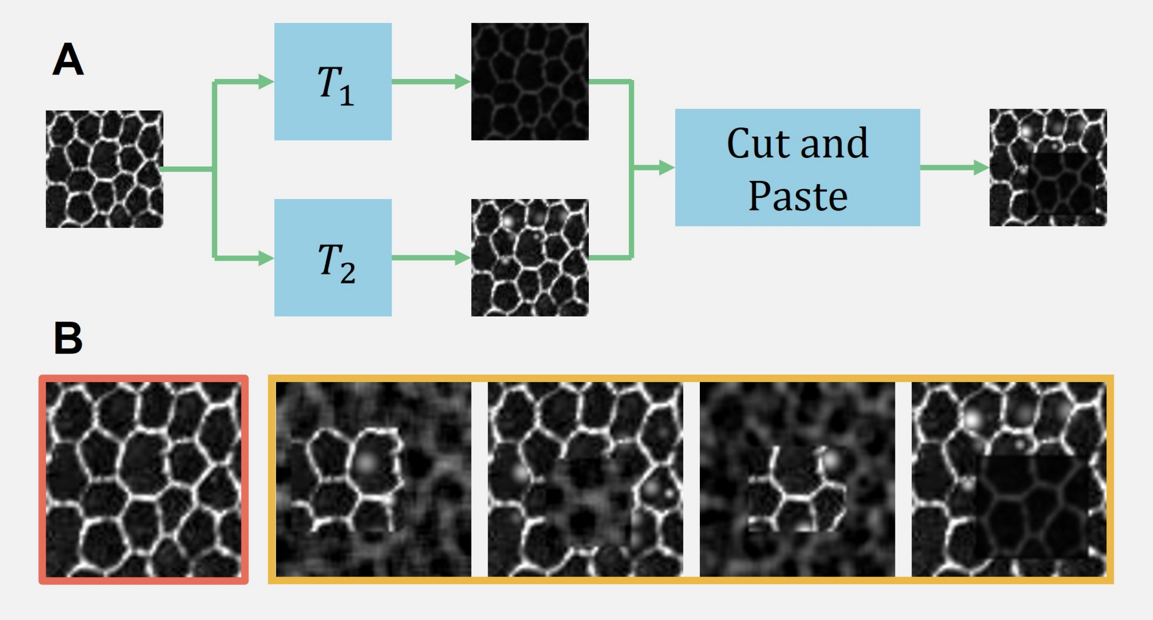

Figure 6.4: Developed image data augmentation algorithm AugCut. (A) Each input image is processed by two augmentation branches for brightness reduction, random image blur, and noise addition. With the resulting two output images, an image sub-region is randomly cut from one and pasted to the other at the same image location. (B) A typical input image (in red) is presented with its augmented views (in yellow).

by T_2 on the bottom branch, mimicking RPE cells with highlighted nuclei in Figure 6.1C. Similar to CutMix [81], I randomly cut an image sub-region from the output of T_1 on the top branch and paste it to the output of T_2 at the same position. Thus, the augmentation result can be described as:

$$T(x; i, j) = \begin{cases} T_1(x; i, j) & (i, j) \in R \\ T_2(x; i, j) & (i, j) \notin R \end{cases} \tag{6.8}$$

where R is a randomly selected rectangle region. In the implementation, T_1 and T_2 are swapped with a probability of 0.5 for symmetry. Figure 6.4B presents a typical input image (in red) and its augmented views (in yellow) by different augmentation operators sampled from the same augmentation operator family T.

6.1.3 Model Implementation

I implement the developed method S^4 with the Python 3.10 and PyTorch 1.8.1 deep learning toolkit [111] and execute programs on two NVIDIA Tesla K80 GPUs. The encoder f consists of an input layer, three down-sampling blocks using convolution layers with stride 2, and six residual blocks. The decoder d consists of three up-sampling blocks using deconvolution layers with stride 1/2 and an output layer. Balancing the trade-off between computational efficiency and deep network efficacy, I adopt 64, 128, 256, and 512 filters from the highest to the lowest resolution level, respectively. MLP head g and h include two and one hidden layers with 512 nodes, respectively. The length of the output representation vector from g and h is 2,048 each.

During training, the loss function first guides the network to extract structural information and recover the input image. After this, the loss function guides the network to produce binary segmentation images with closed cell borders. To realize this two-stage training, a dynamic factor λ is adopted:

$$\lambda(t) = \begin{cases} s & t \leq t_1 \\ s - \frac{s-e}{t_2-t_1}(t-t_1) & t_1 < t < t_2 \\ e & t \geq t_2 \end{cases} \tag{6.9}$$

where t_1 and t_2 are transition "time" cutoff values in the unit of epoch; s and e are the constant values before and after the transition. Factor λ_1 has settings: $s = 1, e = 0.5, t_1 = 40, t_2 = 70$. Factor λ_2 is set to constant 0.5. Factor λ_3 has settings: $s = 0, e = 1, t_1 = 40, t_2 = 70$.

6.2 Results

6.2.1 Training and testing datasets

In this study, the mouse RPE flatmount images are selected from our image database [127, 128]. As these RPE images have high image resolutions (around $4,000 \times 4,000$ pixels each), each image is divided into small image patches of 96×96 pixels for model training. Although the developed method S^4 only requires unlabeled images with strong borders, the state-of-the-art methods in the comparison study require additional training data. Therefore, both labeled and unlabeled RPE cells are included in the training set. The annotated set $\mathcal{P}$ has 155 patches with manually annotated ground truth. Regarding unlabeled RPE cells for training, I have two subsets. One high quality image set $\mathcal{X}$ includes 653 patches and another low quality $\mathcal{Y}$ includes 987 image patches. The testing set includes 43,258 RPE cells from 300 image patches.

Multiple evaluation metrics are used for RPE cell segmentation evaluation. In image segmentation tasks, metrics derived from the confusion matrix are frequently utilized [61, 62, 63]. In this work, I assign the positive class to border pixels and the negative class to pixels not on borders. Given these two classes, the confusion matrix has four values: TP (number of correctly classified pixels on borders), FP (number of incorrectly classified pixels not on borders), FN (number of incorrectly classified pixels on borders), and TN (number of correctly classified pixels not on borders). With these values, Precision (Pre), Recall (Rec), Intersection-Over-Union (IOU), and Dice Similarity Coefficient (DSC) are computed for method performance evaluation:

$$\begin{aligned} \text{Pre} &= \frac{\text{TP}}{\text{TP+FP}}, & \text{Rec} &= \frac{\text{TP}}{TP + FN} \\ \text{IOU} &= \frac{\text{TP}}{\text{TP+FP+FN}}, & \text{DSC} &= \frac{\text{TP} \times 2}{\text{TP} \times 2 + \text{FP+FN}} \end{aligned} \tag{6.10}$$

To utilize segmentation results for down-stream morphological feature extraction, Correct Rate (CR) and Weighted Correct Rate (WCR) described in Section 5.1.3 are adopted.

6.2.2 Model validation and performance comparison

The developed method S^4 is compared with four supervised learning models (i.e., UNet [63], DeepLab [62], MultiResUNet [51], and Cellpose [124]), and two semi-supervised learning methods (i.e., UNet enhanced with CUT [48] and MultiHeadGAN [6]). UNet, DeepLab, and MultiResUNet have been widely applied to a large number of biomedical image segmentation tasks [98]. Cellpose adopts a pre-trained UNet model to predict gradient vector fields and segment cells by gradient tracking. Two semi-supervised approaches using the pairwise learning mechanism have been developed to use unlabeled data for model training [6]. For method comparison, supervised learning approaches are trained with the labeled training set $\mathcal{P}$ and semi-supervised learning approaches are trained with training sets $\mathcal{P}$, $\mathcal{X}$, and $\mathcal{Y}$.

I present and compare typical RPE cell segmentation results of these models in Figure 6.5. By visual comparisons, the results from supervised learning approaches except for Cellpose have large under-segmented regions. Semi-supervised learning approaches can effectively mitigate under-segmentation, while the developed method S^4 achieves the best overall performance. Note that the performance of Cellpose is the second best as it is trained with a large amount of labeled data. However, Cellpose fails to produce reasonable RPE cell segmentation results when the gradient vectors in cells do not converge.

I also quantitatively evaluate and compare segmentation results from different models with multiple metrics (i.e., Pre, Rec, IOU, DSC, CR, and WCR). The quantitative comparison results are shown in Table 6.1. The developed method S^4 achieves the best performance by Pre (85.0%), IOU (76.8%), DSC (86.7%), CR (88.9%), and WCR (89.9%), respectively. Additionally, quantitative evaluation results are plotted in Figure 6.6 to present the statistical difference between the developed method and

Table 6.1: Quantitative performance comparison across the developed method S^4 and deep learning methods by multiple evaluation metrics. All values are in percentage (%).

Method	Pre	Rec	IOU	DSC	CR	WCR
Supervised						
UNet	74.4	67.1	54.5	70.0	53.1	57.0
Deeplab	74.4	68.6	55.5	71.0	62.6	66.3
MultiResUNet	76.7	79.9	64.2	78.0	62.3	65.2
Cellpose	68.2	89.0	62.8	76.7	88.8	89.2
Semi-supervised						
CUT+UNet	78.6	82.7	67.4	80.5	72.2	75.3
MultiHeadGAN	61.2	**90.1**	57.5	72.9	84.3	85.6
Self-supervised						
S^4 (Ours)	**85.0**	88.4	**76.8**	**86.7**	**88.9**	**89.9**

Table 6.2: Quantitative performance comparison across different training loss combinations by multiple evaluation metrics. Included and excluded terms are checked by "✓" and "✗" respectively. All values are in percentage (%).

Loss			Pre	Rec	IOU	DSC	CR	WCR
$\mathcal{L}_{Rec}$	$\mathcal{L}_{PR}$	$\mathcal{L}_{Mor}$						
✓	✗	✗	79.5	61.8	53.6	69.0	38.6	40.5
✓	✓	✗	83.6	65.3	58.1	73.1	40.4	43.5
✓	✗	✗	80.7	82.0	68.7	81.2	84.9	83.8
✓	✓	✓	**85.0**	**88.4**	**76.8**	**86.7**	**88.9**	**89.9**

others. By Pre, IOU, and DSC, the method S^4 is significantly better than all other approaches. By CR and WCR, S^4 is significantly better than all other approaches except Cellpose. Although MultiHeadGAN achieves the best performance by Rec, the difference between S^4 and MultiHeadGAN is negligible (i.e., 1.7% with p=0.08).

6.2.3 Ablation study

I present a set of ablation experiments to demonstrate the efficacy of 1) loss term combinations, 2) pairwise learning strategies, 3) data augmentation methods, and 4) dynamic loss term weight factors for training.

To study the contribution of individual training losses in the developed S^4 method, ablation experiments with different loss term combinations are carried out (Table 6.2).

Table 6.3: Quantitative performance comparison across different pairwise representation learning strategies by multiple evaluation metrics. I: input x. O: outputs z_1, z_2. All values are in percentage (%).

Learning Strategy	Loss from		Pre	Rec	IOU	DSC	CR	WCR
	I	O						
BYOL	✓	✓	84.4	84.4	73.2	84.2	81.4	84.3
	✓	✗	84.8	**88.6**	**76.8**	86.6	**89.2**	**90.2**
SimSiam	✓	✓	**86.9**	84.0	75.0	85.4	83.6	85.6
	✓	✗	85.0	88.4	**76.8**	**86.7**	88.9	89.9

As the reconstruction loss $\mathcal{L}_{Rec}$ is necessary for training, I keep this loss term and only test with different combinations of $\mathcal{L}_{PR}$ and $\mathcal{L}_{Mor}$. Note that the resulting output image is in grayscale when $\mathcal{L}_{PR}$ is not used. For fair comparisons, the same morphological transformations (Section 3.1) is used to binarize the grayscale output in such a case. By Table 6.2, when individually combined with $\mathcal{L}_{Rec}$, the loss term $\mathcal{L}_{Mor}$ presents better performance than $\mathcal{L}_{PR}$ by Rec (+25.6%), IOU (+18.2%), DSC (+11.1%), CR (+110.1%), and WCR (+92.6%), respectively. Additionally, the combination of all three loss terms used in the developed S^4 achieves the best performance. This suggests that the pairwise representation loss and the morphology loss are complementary to the reconstruction loss term, contributing to an enhanced method performance.

I further test the impact of different self-supervised learning strategies on segmentation performance. In the developed method, SimSiam [129] is adopted to compute the pairwise representation loss with augmented view x_1 and x_2. In the ablation study, an additional pairwise representation loss term $\mathcal{L}'_{PR}$ related to outputs (i.e., z_1 and z_2) is adopted. Thus, I process z_1 with encoder f, MLP head g, and h in sequence and obtain $p'_1 = g(f(z_1))$ and $q'_1 = h(p'_1)$. Similarly, p'_2 and q'_2 with z_2 are computed. The resulting overall pairwise representation loss is related to both input and outputs, and formulated as $\frac{1}{2}\mathcal{L}_{PR} + \frac{1}{2}\mathcal{L}'_{PR}$ with $\mathcal{L}'_{PR}$ defined as:

$$\mathcal{L}'_{PR} = -\frac{1}{2}\mathcal{D}\left(q'_1, sg\left(p'_2\right)\right) - \frac{1}{2}\mathcal{D}\left(q'_2, sg\left(p'_1\right)\right) \tag{6.11}$$

Table 6.4: Quantitative performance comparison among different image augmentation strategies by multiple evaluation metrics. All values are in percentage (%).

Augmentation Strategy	Pre	Rec	IOU	DSC	CR	WCR
T_1	86.9	84.0	75.0	85.4	83.6	85.7
T_2	80.3	72.8	61.9	76.1	53.4	56.0
Random	83.7	87.3	75.0	85.4	84.5	85.2
AugCut	**85.0**	**88.4**	**76.8**	**86.7**	**88.9**	**89.9**

Table 6.5: Quantitative performance comparison with different transition cutoff time values in the unit of epoch for weight factor λ_1 and λ_3. All evaluation metric values are in percentage (%).

t_1	t_2	Pre	Rec	IOU	DSC	CR	WCR
0	0	**86.0**	78.4	69.8	81.9	83.5	85.1
0	30	/	/	/	/	/	/
20	50	77.7	73.5	60.9	75.4	61.9	65.1
40	70	85.0	**88.4**	**76.8**	**86.7**	**88.9**	**89.9**
60	90	83.2	80.9	69.9	82.0	80.9	83.1
80	110	81.4	78.4	66.7	79.7	80.1	81.7

Similar to the ablation studies with SimSiam, I formulate the pairwise representation loss with another state-of-the-art strategy BYOL [130] with only inputs, and both inputs and outputs. The experimental results are presented in Table 6.3. The difference between BYOL and SimSiam is not significant by Pre (p=0.86), Rec (p=0.89), IOU (p=0.99), DSC (p=0.98), CR (p=0.75), or WCR (p=0.78). As BYOL includes an online and a target network, it is more time-consuming and takes more memory for model training than SimSiam. Note that the addition of the pairwise representation loss by outputs can impair the model performance.

Table 6.4 presents the effectiveness of the developed image augmentation algorithm AugCut by experimental results. For comparison, the testing performance associated with T_1 and T_2 is individually presented. Additionally, I show an enhanced testing performance with input views randomly augmented by either T_1 or T_2. Of all augmentation strategies for comparison, AugCut mixing results by T_1 and T_2 presents the best performance.

Table 6.6: Quantitative performance comparison among different stabilized values for weight factor λ_1 by multiple evaluation metrics. All values are in percentage (%).

Stabilized value e	Pre	Rec	IOU	DSC	CR	WCR
0.3	75.9	76.4	61.6	76.0	78.3	79.8
0.5	**85.0**	**88.4**	**76.8**	**86.7**	**88.9**	**89.9**
0.7	74.3	81.2	63.5	77.4	83.1	85.5

I investigate the optimal transition cutoff time settings (i.e., t_1 and t_2) for loss weight factor λ_1 and λ_3 (Table 6.5). The study results suggest the strategy with fixed weight factors (i.e., $t_1 = t_2 = 0$) can successfully train a network with an acceptable overall performance. However, such a strategy is very sensitive to the network parameter initialization and often results in a degenerated network with unmeaningful outputs. Next, I gradually increase the starting (i.e., t_1) and ending time (i.e., t_2) for the transition. The stage before the transition is defined as pre-training. The segmentation performance is improved as the pre-training time increases and achieves the best when it is increased to 40 epochs. When the pre-training time is increased further, the model performance becomes worse by all evaluation metrics.

I also study the impact of different stabilized values (i.e., e) for the weight factor λ_1 on the method performance. Figure 6.7 presents typical segmentation examples with different e. When e takes a large value, the loss term $\mathcal{L}_{Rec_i}$ between input x and outputs (i.e., z_1 and z_2) imposes a strong constraint on outputs and prevents them from being binary. As a result, some cell borders in the outputs are still in grayscale. By contrast, a small value for e tends to make cell borders deviate from true cell border structures in the input images. By the ablation study, I make $e = 0.5$ as this stabilized value selection can produce promising model performance by all evaluation metrics (Table 6.6).

6.3 Discussion and summary

A precise and complete understanding of the RPE cell morphology is the key to improve the comprehension of RPE physiology and aging [7, 128, 132]. In turn, an accurate segmentation is a prerequisite for the morphological characterization. Due to lack of appropriate computational tools, RPE cell segmentation often depends on manual annotations, a process suffering from a large inter- and intra-variability. Such a human annotation process is also too time-consuming to produce a sufficiently large number of annotated RPE cells in damaged tissue image regions for deep learning training. To address this challenge, a novel method (S^4) is developed for segmenting RPE cells in flatmount fluorescent microscopy images in this study.

The developed method takes a self-supervised learning strategy and enables deep neural networks to learn from unlabeled image data, resulting in a more generalized feature extraction ability and learning outcome. My idea is to produce synthetic damaged image regions by applying image augmentations to good quality image patches and train a network to recover good quality images by a reconstruction loss. To enhance the model performance, I combine this component with two MLP heads that extract representation vectors for the pairwise learning during the training. The ablation study results in Table 6.2 manifest that the model benefits from this combination strategy by all included performance metrics. However, this loss combination can only produce grayscale images. To achieve RPE cell segmentation results in binary, a RPE cell morphology loss is designed to compare decoder outputs z with their binary results w after the designed morphological transformations. With the addition of the morphology loss, the network generates decoder outputs approaching binarized segmentation maps. In practice, a dynamic strategy is adopted to adjust the composition of loss terms as a function of the training epoch. At the beginning of the training stage, the network is trained with pairwise representation loss $\mathcal{L}_{PR}$ and reconstruction loss $\mathcal{L}_{Rec\_i}$ between input x and outputs (i.e., z_1 and z_2). After the training performance becomes stable, the weight of $\mathcal{L}_{Rec\_i}$ is decreased to reduce the input-output similarity constraint. Concurrently, the weight of reconstruction loss $\mathcal{L}_{Rec\_o}$ is increased to support the supervision by semantic information. Meanwhile, the weight of morphology loss $\mathcal{L}_{Mor}$ is increased to force the model to produce binary segmentation outputs.

In the ablation study, the superiority of the developed method design is demonstrated by experimental results of multiple variants of the method. Specifically, it can be observed that the addition of pairwise representation loss PR from outputs decreases the segmentation performance in Table 6.3. The core task for the encoder f is to extract structural information from damaged RPE image regions. However, the loss term $\mathcal{L}'_{PR}$ makes use of the encoder f to extract information from the binary output z, resulting in a performance decrease.

Data augmentation results in Table 6.4 suggest that the augmentation with T_1 performs better than the augmentation with T_2 in general. Recall that T_1 augments weak cell border cases, while T_2 produces cells with strong nuclei. As missing cell borders in the segmentation results can significantly alter the RPE cell topology, these cells have a higher impact on the method performance. Additionally, as both weak RPE cell borders and noisy tissue regions with highlighted nuclei are augmented in the corrupted image views by AugCut, it creates a rich data diversity, contributing to its best performance.

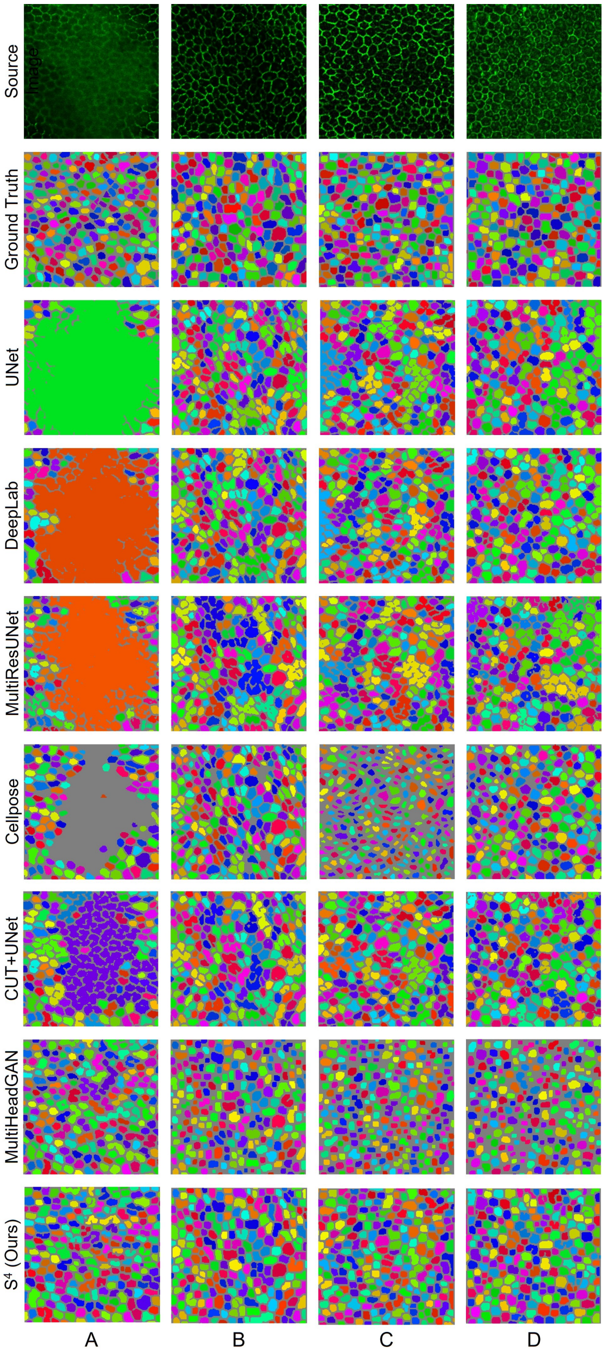

Figure 6.5: Qualitative comparison of deep learning approaches for RPE cell segmentation with flatmount microscopy images. Four typical impaired image regions are shown in columns (A-D) with rows for ground truth and corresponding segmentation results of UNet, DeepLab, MultiResUNet, Cellpose, CUT+UNet, MultiHeadGAN, and the developed S^4, respectively. Column (A) demonstrates the case that the whole region is severely blurred, while columns (B-C) present cases where cell borders are partially missing. Column (D) presents the case where RPE cells contain highlighted nuclei.

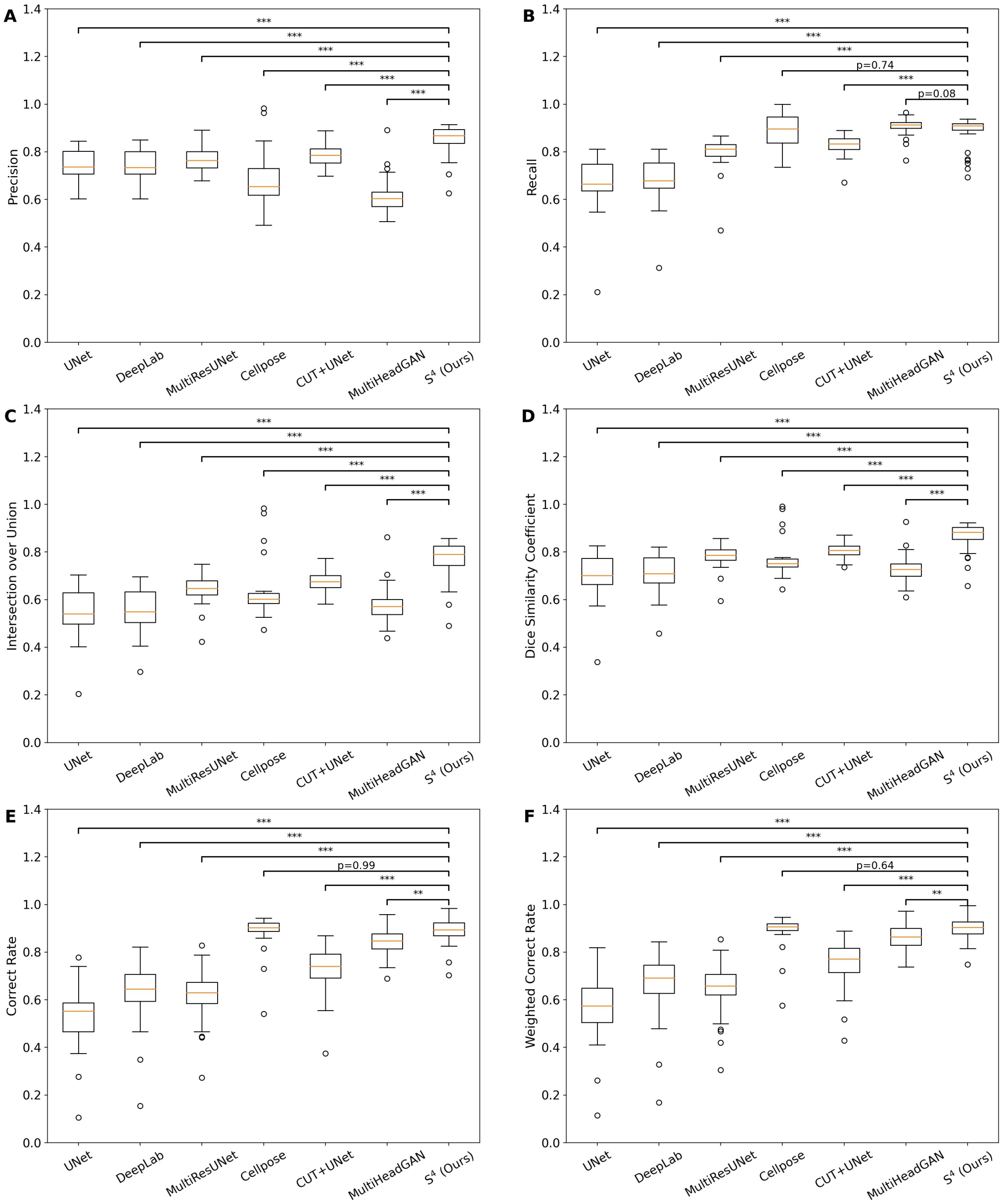

Figure 6.6: Quantitative comparison of deep learning approaches for RPE flatmount image segmentation. The RPE cell segmentation performance of deep learning approaches are compared by (A) Precision, (B) Recall, (C) Intersection over union, (D) Dice similarity coefficient, (E) Correct rate and (F) Weighted correct rate. Paired sample t-tests between the developed S^4 and other six state-of-the-art approaches suggest a statistically significant performance difference. The notations for *, **, and *** represent a p-value less than 0.05, 0.005, and 0.0005, respectively.

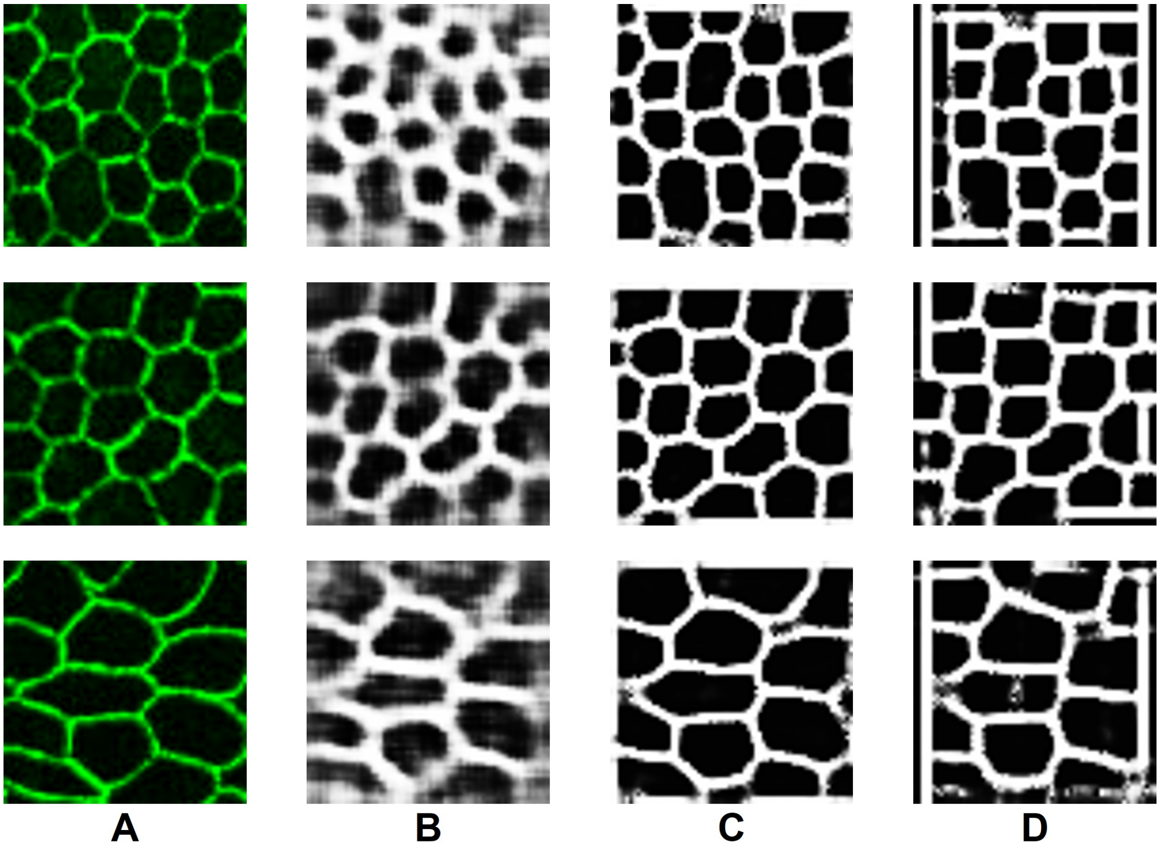

Figure 6.7: Typical output examples of networks trained with different stabilized values for weight factor λ_1. (A) Three typical input images; (B-D) Outputs associated with the stabilized value $e = 0.3, 0.5, 0.7$, respectively.

Chapter 7

Conclusion

In summary, this dissertation focuses on designing and applying computational methods to solve various biomedical image analysis tasks. Specifically, two ways are explored to boost model performance: 1) Enhancement of existing methods by the prior knowledge for specific tasks, and 2) Formulation of ways to leverage unlabeled data for semi-/unsupervised deep learning algorithms. In this dissertation, I focus on two common biomedical image analysis tasks: image segmentation and object tracking. Validated by different image types (e.g., fluorescence microscopy images and bright-field histopathology microscopy images) from various sources (e.g., bacteria, lung cancer spheroids, human liver biopsies, and retinal pigment epithelium tissues), the developed methods demonstrate their promising potential to support biomedical image analysis tasks.

For deep learning based image segmentation tasks, I have accumulated some knowledge on analysis pipeline setups after intensive experimental practices. If adequate high-quality labeled ground truth is available, supervised learning methods are preferred. Otherwise, training with unlabeled image data would be a choice. In this case, the training loss and strategy should be elaborately designed for semi-/unsupervised deep learning. This is suggested by ablation studies in Chapter 5 and 6 where significant differences in segmentation performance are presented when different loss terms and training strategies are applied. In general, loss terms are recommended to include prior knowledge of specific analysis tasks. For example, the shape loss in Chapter 5 is developed in the sense that all cells of interest for analysis have closed borders. Thus, the designed loss term increases when misclassification occurs on cell borders. Besides, multi-stage strategies are useful when unlabeled image data are leveraged for training. Forcing a deep learning model to generate binary segmentation results can possibly result in model degradation, especially when input images are corrupted by noise. Instead, we can first guide the model to produce an enhanced gray-scale image and gradually transit to the binary result after the training process is stabilized.

In the future, the developed methods in this dissertation can be generalized to support the analysis of biomedical images of different modalities. For further performance improvement, one approach is to adopt "network engineering" techniques. Emerging network architectures such as ViT [133] and MLP-Mixer [134] have presented their promising potential to strengthen or even replace convolutional neural networks. Finally, more learning methods (e.g., weakly supervised learning and few-shot learning) will be studied, along with their applications to more challenging biomedical image analysis tasks.

Bibliography

[1] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221, 2017.

[2] Nahum Kiryati and Yuval Landau. Dataset growth in medical image analysis research. *Journal of imaging*, 7(8):155, 2021.

[3] Hanyi Yu, Sung Bo Yoon, Robert Kauffman, Jens Wrammert, Adam Marcus, and Jun Kong. Non-gaussian models for object motion analysis with time-lapse fluorescence microscopy images. In *Modern Statistical Methods for Health Research*, pages 15–41. Springer, 2021.

[4] Robert C Kauffman, Oluwaseyi Adekunle, Hanyi Yu, Alice Cho, Lindsay E Nyhoff, Meagan Kelly, Jason B Harris, Taufiqur Rahman Bhuiyan, Firdausi Qadri, Stephen B Calderwood, et al. Impact of immunoglobulin isotype and epitope on the functional properties of vibrio cholerae o-specific polysaccharide-specific monoclonal antibodies. *Mbio*, 12(2):e03679–20, 2021.

[5] Hanyi Yu, Nima Sharifai, Kun Jiang, Fusheng Wang, George Teodoro, Alton B Farris, and Jun Kong. Artificial intelligence based liver portal tract region identification and quantification with transplant biopsy whole-slide images. *Computers in Biology and Medicine*, 150:106089, 2022.

[6] Hanyi Yu, Fusheng Wang, George Teodoro, John Nickerson, and Jun Kong. Multiheadgan: A deep learning method for low contrast retinal pigment epithe-

lium cell segmentation with fluorescent flatmount microscopy images. *Computers in Biology and Medicine*, 146:105596, 2022.

[7] Yong-Kyu Kim, Hanyi Yu, Vivian R Summers, Kevin J Donaldson, Salma Ferdous, Debresha Shelton, Nan Zhang, Micah A Chrenek, Yi Jiang, Hans E Grossniklaus, et al. Morphometric analysis of retinal pigment epithelial cells from c57bl/6j mice during aging. *Investigative ophthalmology & visual science*, 62(2):32–32, 2021.

[8] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[9] Junyi Chai, Hao Zeng, Anming Li, and Eric WT Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

[10] Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, 2018.

[11] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

[12] Kishor Barasu Bhangale and Mohanaprasad Kothandaraman. Survey of deep learning paradigms for speech processing. *Wireless Personal Communications*, pages 1–37, 2022.

[13] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

[14] Mihalj Bakator and Dragica Radosav. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, 2(3):47, 2018.

[15] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):1–9, 2021.

[16] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

[17] Mahmoud Abbasi, Amin Shahraki, and Amir Taherkordi. Deep learning for network traffic monitoring and analysis (ntma): A survey. *Computer Communications*, 170:19–41, 2021.

[18] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

[19] Man Zhang, Yong Zhou, Jiaqi Zhao, Yiyun Man, Bing Liu, and Rui Yao. A survey of semi-and weakly supervised semantic segmentation of images. *Artificial Intelligence Review*, 53(6):4259–4288, 2020.

[20] Lars Schmarje, Monty Santarossa, Simon-Martin Schröder, and Reinhard Koch. A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access*, 9:82146–82168, 2021.

[21] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[24] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[29] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[30] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2733–2742, 2018.

[31] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, pages 776–794. Springer, 2020.

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[34] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019.

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[36] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

[37] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adver-

sarial nets. In *Advances in neural information processing systems*, volume 27, pages 2672–2680, 2014.

[39] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv*, 2014.

[40] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[41] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. *arXiv*, 2018.

[42] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*, 2015.

[43] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in neural information processing systems*, volume 29, pages 5040–5048, 2016.

[44] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[46] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised

dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[47] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. PMLR, 2017.

[48] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020.

[49] Zitao Zeng, Weihao Xie, Yunzhe Zhang, and Yao Lu. RIC-Unet: An improved neural network based on Unet for nuclei segmentation in histology images. *IEEE Access*, 7:21420–21428, 2019.

[50] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-DenseUNet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE Transactions on Medical Imaging*, 37(12):2663–2674, 2018.

[51] Nabil Ibtehaz and M Sohel Rahman. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Networks*, 121:74–87, 2020.

[52] Mei Yu, Ming Han, Xuewei Li, Xi Wei, Han Jiang, Huiling Chen, and Ruiguo Yu. Adaptive soft erasure with edge self-attention for weakly supervised semantic segmentation: thyroid ultrasound image case study. *Computers in Biology and Medicine*, 144:105347, 2022.

[53] José Denes Lima Araújo, Luana Batista da Cruz, João Otávio Bandeira Diniz, Jonnison Lima Ferreira, Aristófanes Corrêa Silva, Anselmo Cardoso de Paiva,

and Marcelo Gattass. Liver segmentation from computed tomography images using cascade deep learning. *Computers in Biology and Medicine*, 140:105095, 2022.

[54] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[55] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International journal of computer vision*, 81(1):2–23, 2009.

[56] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th international conference on computer vision*, pages 670–677. IEEE, 2009.

[57] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, volume 25, pages 1097–1106, 2012.

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[62] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[63] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[64] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[65] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[66] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[67] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[68] D Luo, J Barker, JC McGrath, and CJ Daly. Iterative multilevel thresholding and splitting for three-dimensional segmentation of live cell nuclei using

laser scanning confocal microscopy. *Journal of Computer-Assisted Microscopy*, 10(4):151–162, 1998.

[69] Gang Lin, Umesh Adiga, Kathy Olson, John F Guzowski, Carol A Barnes, and Badrinath Roysam. A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 56(1):23–36, 2003.

[70] Jun Kong, Fusheng Wang, George Teodoro, Yanhui Liang, Yangyang Zhu, Carol Tucker-Burden, and Daniel J Brat. Automated cell segmentation with 3d fluorescence microscopy images. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 1212–1215. IEEE, 2015.

[71] Michael K Cheezum, William F Walker, and William H Guilford. Quantitative comparison of algorithms for tracking single fluorescent particles. *Biophysical journal*, 81(4):2378–2388, 2001.

[72] Olivier Debeir, Philippe Van Ham, Robert Kiss, and Christine Decaestecker. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE transactions on medical imaging*, 24(6):697–711, 2005.

[73] Daniel Sage, Franck R Neumann, Florence Hediger, Susan M Gasser, and Michael Unser. Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. *IEEE transactions on image processing*, 14(9):1372–1383, 2005.

[74] Ihor Smal, Wiro Niessen, and Erik Meijering. Advanced particle filtering for multiple object tracking in dynamic fluorescence microscopy images. In *2007 4th*

IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pages 1048–1051. IEEE, 2007.

[75] Auguste Genovesio, Tim Liedl, Valentina Emiliani, Wolfgang J Parak, Maité Coppey-Moisan, and J-C Olivo-Marin. Multiple particle tracking in 3-d+ t microscopy: method and application to the tracking of endocytosed quantum dots. *IEEE Transactions on Image Processing*, 15(5):1062–1070, 2006.

[76] WJ Godinez and K Rohr. Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. *IEEE transactions on medical imaging*, 34(2):415–432, 2015.

[77] Ricard Delgado-Gonzalo, Nicolas Chenouard, and Michael Unser. A new hybrid bayesian-variational particle filter with application to mitotic cell tracking. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1917–1920. IEEE, 2011.

[78] Yogesh Rathi, Namrata Vaswani, Allen Tannenbaum, and Anthony Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *IEEE transactions on pattern analysis and machine intelligence*, 29(8):1470, 2007.

[79] Christophe Zimmer, Elisabeth Labruyere, Vannary Meas-Yedid, Nancy Guillén, and J-C Olivo-Marin. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing. *IEEE transactions on medical imaging*, 21(10):1212–1221, 2002.

[80] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[81] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[82] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. *arXiv preprint arXiv:2007.13003*, 2020.

[83] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[84] Chaitanya K Ryali, David J Schwab, and Ari S Morcos. Leveraging background augmentations to encourage semantic focus in self-supervised contrastive learning. *arXiv preprint arXiv:2103.12719*, 2021.

[85] Ihor Smal, Erik Meijering, Katharina Draegestein, Niels Galjart, Ilya Grigoriev, Anna Akhmanova, ME Van Royen, Adriaan B Houtsmuller, and Wiro Niessen. Multiple object tracking in molecular bioimaging by rao-blackwellized marginal particle filtering. *Medical Image Analysis*, 12(6):764–777, 2008.

[86] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[87] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[88] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[89] Chenyang Xu and Jerry L Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on image processing*, 7(3):359, 1998.

[90] Kun Jiang, Mohammad K Mohammad, Wasim A Dar, Jun Kong, and Alton B Farris. Quantitative assessment of liver fibrosis by digital image analysis reveals correlation with qualitative clinical fibrosis staging in liver transplant patients. *Plos one*, 15(9):e0239624, 2020.

[91] Koichiro Yasaka, Hiroyuki Akai, Akira Kunimatsu, Osamu Abe, and Shigeru Kiryu. Liver fibrosis: deep convolutional neural network for staging by using gadoxetic acid-enhanced hepatobiliary phase mr images. *Radiology*, 287(1):146–155, 2018.

[92] RA Standish, E Cholongitas, A Dhillon, AK Burroughs, and AP Dhillon. An appraisal of the histopathological assessment of liver fibrosis. *Gut*, 55(4):569–578, 2006.

[93] Neil D Theise, Jidong Jia, Yameng Sun, Aileen Wee, and Hong You. Progression and regression of fibrosis in viral hepatitis in the treatment era: the beijing classification. *Modern Pathology*, 31(8):1191–1200, 2018.

[94] Yang Yu, Jiahao Wang, Chan Way Ng, Yukun Ma, Shupei Mo, Eliza Li Shan Fong, Jiangwa Xing, Ziwei Song, Yufei Xie, Ke Si, et al. Deep learning enables automated scoring of liver fibrosis stages. *Scientific reports*, 8(1):1–10, 2018.

[95] Yang Chen, Yan Luo, Wei Huang, Die Hu, Rong-qin Zheng, Shu-zhen Cong, Fan-kun Meng, Hong Yang, Hong-jun Lin, Yan Sun, et al. Machine-learning-based classification of real-time tissue elastography for hepatic fibrosis in pa-

tients with chronic hepatitis B. *Computers in biology and medicine*, 89:18–23, 2017.

[96] Stefan G Stanciu, Shuoyu Xu, Qiwen Peng, Jie Yan, George A Stanciu, Roy E Welsch, Peter TC So, Gabor Csucs, and Hanry Yu. Experimenting liver fibrosis diagnostic by two photon excitation microscopy and bag-of-features image classification. *Scientific reports*, 4(1):1–12, 2014.

[97] Shuoyu Xu, Yan Wang, Dean CS Tai, Shi Wang, Chee Leong Cheng, Qiwen Peng, Jie Yan, Yongpeng Chen, Jian Sun, Xieer Liang, et al. qFibrosis: a fully-quantitative innovative method incorporating histological features to facilitate accurate fibrosis scoring in animal model and chronic hepatitis B patients. *Journal of hepatology*, 61(2):260–269, 2014.

[98] S Kevin Zhou, Hayit Greenspan, Christos Davatzikos, James S Duncan, Bram Van Ginneken, Anant Madabhushi, Jerry L Prince, Daniel Rueckert, and Ronald M Summers. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, 109(5):820–838, 2021.

[99] Kyu Jin Choi, Jong Keon Jang, Seung Soo Lee, Yu Sub Sung, Woo Hyun Shim, Ho Sung Kim, Jessica Yun, Jin-Young Choi, Yedaun Lee, Bo-Kyeong Kang, et al. Development and validation of a deep learning system for staging liver fibrosis by using contrast agent–enhanced CT images in the liver. *Radiology*, 289(3):688–697, 2018.

[100] Koichiro Yasaka, Hiroyuki Akai, Akira Kunimatsu, Osamu Abe, and Shigeru Kiryu. Deep learning for staging liver fibrosis on CT: a pilot study. *European radiology*, 28(11):4578–4585, 2018.

[101] Jeong Hyun Lee, Ijin Joo, Tae Wook Kang, Yong Han Paik, Dong Hyun

Sinn, Sang Yun Ha, Kyunga Kim, Choonghwan Choi, Gunwoo Lee, Jonghyon Yi, et al. Deep learning with ultrasonography: automated classification of liver fibrosis using a deep convolutional neural network. *European radiology*, 30(2):1264–1273, 2020.

[102] Alex Treacher, Daniel Beauchamp, Bilal Quadri, David Fetzer, Abhinav Vij, Takeshi Yokoo, and Albert Montillo. Deep learning convolutional neural networks for the estimation of liver fibrosis severity from ultrasound texture. In *Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, pages 847–854. SPIE, 2019.

[103] Yuval Ramot, Ameya Deshpande, Virginia Morello, Paolo Michieli, Tehila Shlomov, and Abraham Nyska. Microscope-based automated quantification of liver fibrosis in mice using a deep learning algorithm. *Toxicologic Pathology*, 49(5):1126–1133, 2021.

[104] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[105] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017.

[106] Pius Kwao Gadosey, Yujian Li, Enock Adjei Agyekum, Ting Zhang, Zhaoying Liu, Peter T Yamak, and Firdaous Essaf. SD-UNet: Stripping down U-Net for segmentation of biomedical images on platforms with low computational budgets. *Diagnostics*, 10(2):110, 2020.

[107] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam:

Convolutional block attention module. In *European Conference on Computer Vision*, pages 3–19. Springer, 2018.

[108] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[109] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep learning and data labeling for medical applications*, pages 179–187. Springer, 2016.

[110] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv*, 2016.

[111] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[112] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[113] Kevin Trebing, Tomasz Stan̆czyk, and Siamak Mehrkanoon. Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. *Pattern Recognition Letters*, 145:178–186, 2021.

[114] Olaf Strauss. The retinal pigment epithelium in visual function. *Physiological reviews*, 85(3):845–881, 2005.

[115] Jayakrishna Ambati, Balamurali K Ambati, Sonia H Yoo, Sean Ianchulev, and

Anthony P Adamis. Age-related macular degeneration: Etiology, pathogenesis, and therapeutic strategies. *Survey of ophthalmology*, 48(3):257–293, 2003.

[116] Thomas Ach, Carrie Huisingh, Gerald McGwin, Jeffrey D Messinger, Tianjiao Zhang, Mark J Bentley, Danielle B Gutierrez, Zsolt Ablonczy, R Theodore Smith, Kenneth R Sloan, et al. Quantitative autofluorescence and cell density maps of the human retinal pigment epithelium. *Investigative ophthalmology & visual science*, 55(8):4832–4841, 2014.

[117] Shagun K Bhatia, Alia Rashid, Micah A Chrenek, Qing Zhang, Beau B Bruce, Mitchel Klein, Jeffrey H Boatright, Yi Jiang, Hans E Grossniklaus, and John M Nickerson. Analysis of rpe morphometry in human eyes. *Molecular vision*, 22:898, 2016.

[118] Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Kevin W Eliceiri, Johannes Schindelin, Albert Cardona, and H Sebastian Seung. Trainable Weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 33(15):2424–2426, 2017.

[119] David R Stirling, Madison J Swain-Bowden, Alice M Lucas, Anne E Carpenter, Beth A Cimini, and Allen Goodman. CellProfiler 4: improvements in speed, utility and usability. *BMC bioinformatics*, 22(1):1–11, 2021.

[120] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *arXiv*, 2022.

[121] Xiaoyuan Guo, Fusheng Wang, George Teodoro, Alton B. Farris, and Jun Kong. Liver steatosis segmentation with deep learning methods. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 24–27, 2019.

[122] Chengjia Wang, Tom MacGillivray, Gillian Macnaught, Guang Yang, and David Newby. A two-stage 3D Unet framework for multi-class segmentation on full resolution image. *arXiv*, 2018.

[123] Noah F Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature biotechnology*, pages 1–11, 2021.

[124] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[125] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[126] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018.

[127] Jeffrey H Boatright, Nupur Dalal, Micah A Chrenek, Christopher Gardner, Alison Ziesel, Yi Jiang, Hans E Grossniklaus, and John M Nickerson. Methodologies for analysis of patterning in the mouse RPE sheet. *Molecular Vision*, 21:40, 2015.

[128] Yi Jiang, Xin Qi, Micah A Chrenek, Christopher Gardner, Jeffrey H Boatright, Hans E Grossniklaus, and John M Nickerson. Functional principal component analysis reveals discriminating categories of retinal pigment epithelial morphol-

ogy in mice. *Investigative ophthalmology & visual science*, 54(12):7274–7283, 2013.

[129] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[130] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[131] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996.

[132] Micah A Chrenek, Nupur Dalal, Christopher Gardner, Hans GrossniklausGrossniklaus, Yi Jiang, Jeffrey H Boatright, and John M Nickerson. Analysis of the RPE sheet in the rd10 retinal degeneration model. In *Retinal Degenerative Diseases*, pages 641–647. Springer, 2012.

[133] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[134] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.