

THREE DIMENSIONAL SEGMENTATION AND  
DETECTION OF FLUORESCENCE MICROSCOPY IMAGES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

David Joon Ho

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2019

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL  
STATEMENT OF DISSERTATION APPROVAL**

Dr. Edward J. Delp, Co-Chair

School of Electrical and Computer Engineering

Dr. Paul Salama, Co-Chair

School of Electrical and Computer Engineering

Dr. Mary L. Comer

School of Electrical and Computer Engineering

Dr. Fengqing Maggie Zhu

School of Electrical and Computer Engineering

**Approved by:**

Dr. Pedro Irazoqui

Head of the School Graduate Program

The LORD your God himself will fight for you

## ACKNOWLEDGMENTS

First of all, I want to express my gratitude to my advisor, Professor Edward J. Delp. It was truly my privilege working with him at the Video and Image Processing (VIPER) laboratory for my Ph.D. study. He is a great engineer who loves solving challenging problems. Under his supervision, I had many great opportunities working on cutting edge research. I was able to accomplish many achievements with him that I would never imagine to even try before. I have learned how to push myself to the limit to challenge and improve myself to become a more mature engineer. Learning processes were never easy but I am really thankful that I was able to fulfill them. I thank him for trusting me and continuing to push me to move forward. I really appreciate his guidance and encouragement.

I want to show my appreciation to Professor Paul Salama. During weekly meetings he has provided me many comments which helped me to think more carefully on every single step in my research. I strongly believe all discussions with him have led into outstanding and thorough research. I really appreciate his time and effort for making me a more critical engineer. I also want to thank Dr. Kenneth W. Dunn for his instructions to move our project forward. He has given me lots of feedback to guide me into practical research topics that help biologists.

I want to say thank you to my committee members, Professor Mary L. Comer and Professor Fengqing Maggie Zhu, for their advice to continue and complete my study.

I want to express special thanks to my colleagues in the Microscopy Project, Dr. Neeraj J. Gadgil, Mr. Soonam Lee, Mr. Chichen Fu, and Ms. Shuo Han. It was a big pleasure working with them to solve many challenging problems in many evenings in the lab. Many discussions between us gave me lots of inspirations for my research.

I also want to thank all members in the VIPER lab who spent many years with me: Dr. Joonsoo Kim, Dr. Khalid Tahboub, Dr. Yu Wang, Dr. Bin Zhao, Ms. Blanca

Delgado, Mr. He Li, Ms. Chang Liu, Mr. Jiaju Yue, Ms. Thitiporn (Bee) Pramoun, Ms. Dahjung Chung, Mr. Shaobo Fang, Ms. Jeehyun Choe, Ms. Qingshuang Chen, Mr. Javier Ribera Prat, Mr. Daniel Mas Montserrat, Mr. David Guera Cobo, Mr. Yuhao Chen, Ms. Ruiting Shao, Ms. Di Chen, Mr. Sri Kalyan Yarlagadda, Mr. Janos Horvath, Mr. Yifan Zhao, Mr. Hanxiang (Hans) Hao, Mr. Sriram Baireddy, Mr. Changye Yang, Mr. Enyu Cai, Mr. Zeman Shao, Mr. Runyu Mao, Mr. Jiangpeng He, and Ms. Han Hu.

My Ph.D. study was partially supported by a George M. O'Brien Award from the National Institutes of Health under grant NIH/NIDDK P30 DK079312 and the endowment of the Charles William Harrison Distinguished Professorship at Purdue University. Thank you for the support to complete my study.

I want to show my gratitude to instructors who provided me an opportunity to serve as a teaching assistant in ECE 202, Linear Circuit Analysis II: Professor Raymond DeCarlo, Professor Eric Furgason, Professor Thomas M. Talavage, and Professor Shreyas Sundaram. Thank you for showing your passion on education. I also want to thank Professor Michael Melloch who gave me an opportunity to instruct a summer course in 2018. It was truly a valuable and unique experience and it definitely helped me to learn how to become a better teacher. I want to thank Mr. Hung-Yi Lo who had worked with me for many years and discussed with me on various topics on teaching and researching.

I want to show my thankfulness to all members in the Korean Presbyterian Church of Purdue. Thank you for spiritually feeding me to finish this study and thank you for being good friends with me especially when I was discouraged. All memories with them are priceless to me. They are truly my spiritual family in West Lafayette.

I want to express my gratitude to my family. Words cannot express how much I am thankful for their support and countless prayers. First of all, I want to thank my father, Dr. Yo-Sung Ho, my mother, Ms. Jaeshin Lim, my brother, Mr. Patrick Hyun Ho, and his wife, Ms. Sung Hong, for their encouragement. I also want to thank my father-in-law, Mr. Kyungkyu Park, my mother-in-law, Ms. Haengja Jang,

my brother-in-law, Mr. Yohan Park, my sister-in-law, Ms. Jinseon Park, and her husband, Mr. Jungwoo Park, for their trust. I want to give big thanks to my wife, Ms. Seojin Park, who shows endless love and support. This is not possible without her sacrifice. She has always been a big strength and encouragement to continue and complete my study. I appreciate her for being a great wife and mother. I want to thank my daughter, Lydia Soo Ho, for being a great joy and smile in my family.

Lastly, I want to thank Jesus Christ, my Lord and my Savior. Thank you for Your cross. Thank you for Your guidance. Thank you for Your faithfulness. This is possible because You have fought for us.

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	x
LIST OF FIGURES . . . . .	xii
ABSTRACT . . . . .	xvi
1 INTRODUCTION . . . . .	1
1.1 Background in Fluorescence Microscopy . . . . .	1
1.1.1 Physics of Fluorescence . . . . .	1
1.1.2 Types of Fluorescence Microscopy . . . . .	3
1.1.3 Limitations of Fluorescence Microscopy . . . . .	5
1.2 Challenges . . . . .	8
1.3 Notation . . . . .	10
1.4 Data Sets . . . . .	11
1.5 Contributions Of The Thesis . . . . .	13
1.6 Publications Results From Our Work . . . . .	16
2 LITERATURE REVIEW . . . . .	19
2.1 Computer Vision and Image Processing Based Methods . . . . .	19
2.2 Machine Learning Based Methods . . . . .	25
2.2.1 Detection and Segmentation Using Convolutional Neural Networks	26
2.2.2 Detection and Segmentation of Biomedical Images Using Con- volutional Neural Networks . . . . .	30
2.2.3 Generative Adversarial Networks . . . . .	34
3 BOUNDARY SEGMENTATION USING STEERABLE FILTERS . . . . .	37
3.1 Proposed Method . . . . .	37
3.1.1 Histogram Equalization and Foreground Segmentation . . . . .	38

	Page
3.1.2	Seed Selection Using Steerable Filters . . . . . 39
3.1.3	Small Blob Removal and Tubule/Lumen Separation . . . . . 41
3.1.4	$z$ -Propagation Refinement . . . . . 42
3.2	Experimental Results . . . . . 42
4	NUCLEI SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORKS . . . . . 55
4.1	Proposed Method . . . . . 55
4.1.1	Synthetic Volume Generation . . . . . 56
4.1.2	3D CNN Training . . . . . 62
4.1.3	3D CNN Inference . . . . . 63
4.2	Experimental Results . . . . . 64
5	NUCLEI DETECTION AND SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORKS . . . . . 73
5.1	Proposed Method . . . . . 73
5.1.1	3D Nuclei Detection . . . . . 74
5.1.2	3D Nuclei Segmentation . . . . . 77
5.2	Experimental Results . . . . . 80
6	CENTER-EXTRACTION-BASED NUCLEI INSTANCE SEGMENTATION . . . . . 89
6.1	Proposed Method . . . . . 89
6.1.1	Synthetic Volume Generation . . . . . 90
6.1.2	Nuclei Detection and Binary Segmentation . . . . . 92
6.1.3	Nuclei Instance Segmentation . . . . . 94
6.2	Experimental Results . . . . . 95
7	NUCLEI DETECTION USING SPHERE ESTIMATION NETWORKS . . . . . 103
7.1	Proposed Method . . . . . 103
7.1.1	Synthetic Volume Generation . . . . . 104
7.1.2	Pre-Processing Steps . . . . . 106

	Page
7.1.3 A 3D Convolutional Neural Network . . . . .	108
7.2 Experimental Results . . . . .	111
8 COLOR LABELING . . . . .	120
8.1 Proposed Color Labeling Method . . . . .	120
8.1.1 Labeling . . . . .	121
8.1.2 Colormap Generation . . . . .	121
8.1.3 Color Assignment . . . . .	122
8.2 Experimental Results . . . . .	123
9 CONCLUSIONS . . . . .	124
9.1 Summary . . . . .	124
9.2 Future Work . . . . .	126
9.3 Publications Results From The Thesis . . . . .	127
REFERENCES . . . . .	129
VITA . . . . .	148

## LIST OF TABLES

Table	Page
1.1 The size of data sets . . . . .	14
3.1 Accuracy, Type-I and Type-II errors for various methods on $I_{z_{95}}$ of the Data-T1 . . . . .	48
3.2 Accuracy, Type-I and Type-II errors for various methods on $I_{z_{17}}$ of the Data-T2 . . . . .	48
3.3 Accuracy, Type-I and Type-II errors for various methods on $I_{z_1}$ of the Data-T3 . . . . .	49
3.4 Accuracy, Type-I and Type-II errors for various methods on $I_{z_{81}}$ of the Data-T4 . . . . .	49
4.1 Accuracy, Type-I and Type-II errors for various methods on $I_{(241:272,241:272,31:62)}$ of Data-N1 . . . . .	69
4.2 Accuracy, Type-I and Type-II errors for various methods on $I_{(241:272,241:272,131:162)}$ of Data-N1 . . . . .	70
4.3 Accuracy, Type-I and Type-II errors for various methods on $I_{(241:272,241:272,231:262)}$ of Data-N1 . . . . .	70
5.1 Accuracy, Type-I and Type-II errors for various methods on $I_{(241:272,241:272,31:62)}$ of Data-N1 . . . . .	85
5.2 Accuracy, Type-I and Type-II errors for various methods on $I_{(241:272,241:272,131:162)}$ of Data-N1 . . . . .	86
5.3 Accuracy, Type-I and Type-II errors for various methods on $I_{(241:272,241:272,231:262)}$ of Data-N1 . . . . .	86
6.1 Dilation factors for the convolutional layers and their corresponding receptive fields . . . . .	95
6.2 Accuracy, Type-I and Type-II errors for various methods on $I_{(193:320,193:320,31:94)}$ of Data-N1 . . . . .	100
6.3 Precision, Recall, and F1 score for various methods on $I_{(193:320,193:320,31:94)}$ of Data-N1 . . . . .	100

7.1 Precision, Recall, and F1 score for various methods on $I_{(193:320,193:320,31:94)}$ of Data-N1 . . . . .	116
--	-----

## LIST OF FIGURES

Figure	Page
1.1 The Jablonski diagram that describes the basic principles of fluorescence microscopy . . . . .	2
1.2 Widefield microscopy . . . . .	4
1.3 Confocal microscopy . . . . .	5
1.4 Jablonski diagram of two-photon microscopy . . . . .	6
1.5 Notation used in thesis to describe a volume . . . . .	10
1.6 Sample images of data sets containing tubular structures . . . . .	12
1.7 Sample images of data sets containing nuclei structures . . . . .	13
2.1 U-Net Architecture . . . . .	31
3.1 Block diagram of the proposed method . . . . .	38
3.2 Spherical coordinates . . . . .	40
3.3 Example of $z$ -propagation refinement . . . . .	43
3.4 Original, intermediate, and segmented images of Data-T1 . . . . .	44
3.5 Original and segmented images of Data-T1 . . . . .	45
3.6 3D visualization of Data-T1 . . . . .	46
3.7 Comparison of other segmentation methods and our proposed method of $I_{z_{95}}$ in Data-T1 . . . . .	47
3.8 Original and segmented images of Data-T2 . . . . .	49
3.9 Original and segmented images of Data-T3 . . . . .	50
3.10 Original and segmented images of Data-T4 . . . . .	51
3.11 Comparison of other segmentation methods and our proposed method of $I_{z_{17}}$ in Data-T2 . . . . .	52
3.12 Comparison of other segmentation methods and our proposed method of $I_{z_1}$ in Data-T3 . . . . .	53

Figure	Page
3.13 Comparison of other segmentation methods and our proposed method of $I_{z_{81}}$ in Data-T4 . . . . .	54
4.1 Block diagram of the proposed method . . . . .	56
4.2 Block diagram for Synthetic Volume Generation . . . . .	56
4.3 Examples of synthetic image volumes and their labeled volumes . . . . .	61
4.4 Comparison between original volumes and a synthetic volume . . . . .	61
4.5 Architecture of our 3D CNN . . . . .	62
4.6 3D CNN Inference . . . . .	63
4.7 Original and segmented images of Data-N1 . . . . .	65
4.8 3D visualization comparison of other segmentation methods and our proposed method of $I_{(241:272,241:272,31:62)}$ of Data-N1 . . . . .	66
4.9 3D visualization comparison of other segmentation methods and our proposed method of $I_{(241:272,241:272,131:162)}$ of Data-N1 . . . . .	67
4.10 3D visualization comparison of other segmentation methods and our proposed method of $I_{(241:272,241:272,231:262)}$ of Data-N1 . . . . .	68
4.11 Original and segmented images of Data-N2 . . . . .	71
4.12 Original and segmented images of Data-N3 . . . . .	72
4.13 Original and segmented images of Data-N4 . . . . .	72
5.1 Block diagram of the proposed method . . . . .	73
5.2 Block diagram of the 3D nuclei detection . . . . .	74
5.3 Architecture of our 3D CNN for Classification . . . . .	76
5.4 Block diagram of the proposed 3D nuclei segmentation . . . . .	77
5.5 Examples of a real microscopy volume, a synthetic microscopy volume and synthetic ground truth volume . . . . .	79
5.6 Architecture of the proposed 3D CNN for Segmentation . . . . .	80
5.7 Original, intermediate, and segmented images in Data-N1 . . . . .	81
5.8 Original and segmented images of Data-N1 . . . . .	82
5.9 3D visualization comparison of other segmentation methods and our proposed method of $I_{(241:272,241:272,31:62)}$ of Data-N1 . . . . .	83

Figure	Page
5.10 3D visualization comparison of other segmentation methods and our proposed method of $I_{(241:272,241:272,131:162)}$ of Data-N1 . . . . .	84
5.11 3D visualization comparison of other segmentation methods and our proposed method of $I_{(241:272,241:272,231:262)}$ of Data-N1 . . . . .	85
5.12 Original and segmented images of Data-N5 . . . . .	87
5.13 Original and segmented images of Data-N6 . . . . .	88
6.1 Block diagram of the proposed method . . . . .	90
6.2 Examples of a real microscopy volume, a synthetic microscopy volume and synthetic ground truth volumes . . . . .	92
6.3 Our first CNN architecture (see Figure 6.1) . . . . .	93
6.4 Our second CNN architecture (see Figure 6.1) . . . . .	94
6.5 Original and segmented images of Data-N1 . . . . .	97
6.6 Original and segmented images of Data-N7 . . . . .	98
6.7 Original and segmented images of Data-N5 . . . . .	99
6.8 Comparison of other segmentation methods and our proposed method of the 97 <sup>th</sup> image of Data-N1 . . . . .	101
6.9 3D visualization comparison of other segmentation methods and our proposed method of $I_{(193:320,193:320,31:94)}$ of Data-N1 . . . . .	102
7.1 Block diagram of SphEsNet . . . . .	103
7.2 Examples of $I^{mask}$ , $I^{syn}$ , $I^{ctr}$ , and $I^{rad}$ . . . . .	107
7.3 Architecture of the Sphere Estimation Network . . . . .	108
7.4 Original and detected images of Data-N1 . . . . .	113
7.5 Original and detected images of Data-N5 . . . . .	114
7.6 Original and detected images of Data-N8 . . . . .	115
7.7 Original and detected images of Data-N4 . . . . .	115
7.8 Original and detected images of Data-N9 . . . . .	117
7.9 Comparison of other detection methods and our proposed method of $I_{z_{50}}$ of Data-N1 . . . . .	118
7.10 3D visualization comparison of other detection methods and our proposed method of $I_{(193:320,193:320,31:94)}$ of Data-N1 . . . . .	119

Figure	Page
8.1 Block diagram of our color labeling method . . . . .	120
8.2 The first 100 colors of our colormap . . . . .	121
8.3 3D window used for color assignment . . . . .	122
8.4 Color-Labeling result . . . . .	123

## ABSTRACT

Ho, David Joon PhD, Purdue University, May 2019. Three Dimensional Segmentation and Detection of Fluorescence Microscopy Images. Major Professors: Edward J. Delp and Paul Salama.

Fluorescence microscopy is an essential tool for imaging subcellular structures in tissue. Two-photon microscopy enables imaging deeper into tissue using near-infrared light. The use of image analysis and computer vision tools to detect and extract information from the images is still challenging due to the degraded microscopy volumes by blurring and noise during the image acquisition and the complexity of subcellular structures presented in the volumes. In this thesis we describe methods for segmentation and detection of fluorescence microscopy images in 3D. We segment tubule boundaries by distinguishing them from other structures using three dimensional steerable filters. These filters can capture strong directional tendencies of the voxels on a tubule boundary. We also describe multiple three dimensional convolutional neural networks (CNNs) to segment nuclei. Training the CNNs usually require a large set of labeled images which is extremely difficult to obtain in biomedical images. We describe methods to generate synthetic microscopy volumes and to train our 3D CNNs using these synthetic volumes without using any real ground truth volumes. The locations and sizes of the nuclei are detected using of our CNNs, known as the Sphere Estimation Network. Our methods are evaluated using real ground truth volumes and are shown to outperform other techniques.

# 1. INTRODUCTION

## 1.1 Background in Fluorescence Microscopy

Fluorescence microscopy is an optical microscopy imaging subcellular structures in three dimensions which are too small to see with the naked eye [1, 2]. Electron microscopy, another type of microscopy, uses electrons to image subcellular structures with a higher magnification [3]. Electron microscopy cannot image living and/or moving specimens because its high energy from the electron beam can harm the specimens. Optical microscopy, or light microscopy, using photons for visualization is preferred among biologists because it is harmless to specimens which enables visualizing living and/or moving specimens [4]. Optical microscopy can image multiple subcellular structures simultaneously using various fluorescent molecules emitting lights in distinct wavelengths. Optical microscopy does not cost as much as electron microscopy. In this section, we will address (1) physics of fluorescence, (2) types of fluorescence microscopy, and (3) limitations of fluorescence microscopy.

### 1.1.1 Physics of Fluorescence

Fluorescence is a phenomenon of emitting light by absorbing and releasing energy from fluorescent molecules called fluorophores. The process of fluorescence is shown in Figure 1.1, known as the Jablonski diagram [1, 5], which is composed of three steps: (1) excitation, (2) vibrational relaxation, and (3) fluorescence. When light with a specific wavelength is used as an excitation source, electrons in fluorophores can absorb photons and get excited. The electrons in the fluorophores move from the ground state to the excited state which occurs in femtoseconds ( $10^{-15}$  seconds). For the next few picoseconds ( $10^{-12}$  seconds), the excited fluorophores have vibrational

relaxation as they transfer some vibrational energy to heat energy. Most molecules collapse to the ground state again as the energy is released to fluorescence emission in nanoseconds ( $10^{-9}$  seconds). The energy of a photon is formulated as

$$E = \frac{hc}{\lambda} \quad (1.1)$$

where  $h$  is the Planck's constant,  $c$  is the speed of light,  $\lambda$  is the wavelength, and  $E$  is the energy of a photon [1]. The energy released during the emission is less than the energy absorbed during the excitation because some energy is lost during vibrational relaxation. According to Equation 1.1, the wavelength of emitted light is longer than the wavelength from the light source. The wavelength difference between the maximum intensity of absorption and the maximum intensity of emission is known as the Stokes shift [1]. By using interference filters to capture lights within a range of specific wavelength, fluorescence images can be generated. Using the property of the Stokes shift, the interference filters desire to capture fluorescent light and reject the source light.

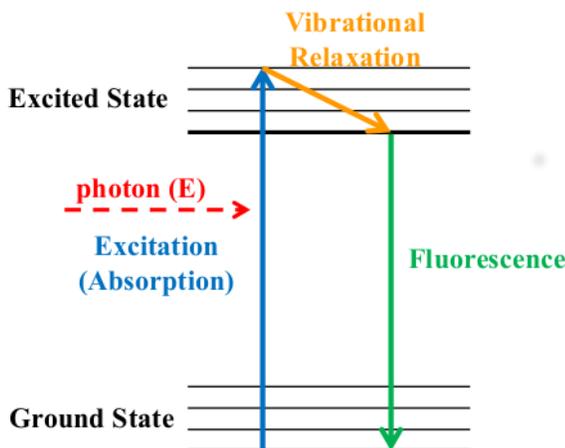


Fig. 1.1.: The Jablonski diagram that describes the basic principles of fluorescence microscopy

Although most of excited molecules collapse to the ground state, some molecules transit from the excited state to the triplet state. Photobleaching is a phenomenon

that fluorophores are permanently no longer able to fluoresce due to photon-induced chemical damage and covalent modification [1]. The triplet state potentially generates radical oxygen species which causing photobleaching. As the fluorophores collide with those radical oxygen species, an electron will be donated to be stable, and the fluorophores will be permanently destroyed. In order to reduce photobleaching, antifade reagents such as oxygen scavenger buffers can be used.

Fluorophores, also known as fluorescent molecules or fluorescent dyes, are particles which have fluorescent properties. Fluorophores having a larger Stokes shift are desired because it is easier to isolate the emitted light from the incoming light by interference filters. The quantum yield is another physical property of fluorophores, where the quantum yield is the ratio of the number of emitted fluorescent photons to the number of absorbed photon. Higher quantum yield is desired for generating brighter images from the same light intensity. Green fluorescent protein (GFP) [6–8], which is naturally from a jellyfish, *aequorea victoria*, and its variants are used as fluorophores due to their usage in living cells. Fluorophores are injected to specimens to image them using optical microscopes.

### 1.1.2 Types of Fluorescence Microscopy

We will investigate various types of fluorescence microscopy such as widefield microscopy, confocal microscopy, and two-photon microscopy.

Widefield microscopy, a type of an optical microscopy, is a technique which is based on Koehler illumination. Koehler illumination is a method for having uniform illumination on the sample [1]. If a fluorophore is located within the region illuminated from the light source, the emitted light from the fluorophore passes through dichroic mirror and a tube lens, and is imaged on a detector. Figure 1.2 shows a system of widefield microscopy.

One of the drawbacks of widefield microscopy is that emitted light from fluorophores in different focal planes would cause a blurred background and low-contrast

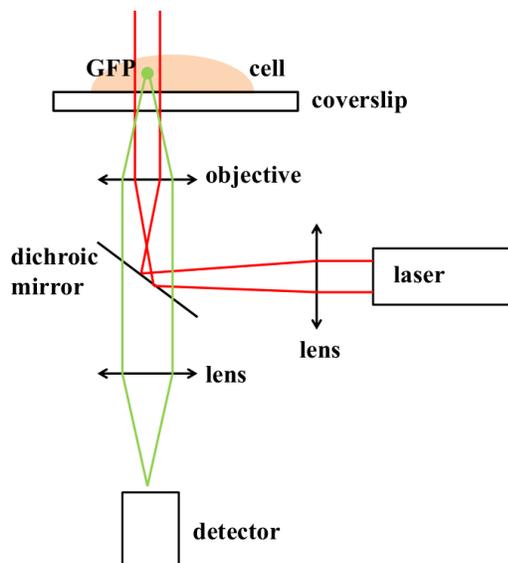


Fig. 1.2.: Widefield microscopy

images. To solve this issue, confocal microscopy, another optical microscopy method, was developed [1,9]. Confocal microscopy uses a pinhole in front of a detector. The light focuses on one focal plane after passing an objective lens. As the emitted light from a fluorophore on that focal plane passes through dichroic mirror, a tube lens, and pinhole, it will be imaged on a detector. Note other fluorophores from different focal plane would be filtered out at the pinhole. Using galvanometer mirrors, one for  $x$ -direction and the other one for  $y$ -direction, the light can scan the entire focal plane in a raster order. In order to image faster, multiple points can be scanned simultaneously using a spinning/Nipkow disk which is more efficient than other single-point scanning methods [10]. Figure 1.3 shows a system confocal microscopy. Note the source light in red in Figure 1.2 illuminates the sample uniformly in widefield microscopy whereas the source light in red in Figure 1.3 is focused on one focal plane using an objective lens in confocal microscopy.

Confocal microscopy has some limitations with detecting photons in deeper tissue. The number of photons from fluorescent molecules in deeper tissue may be

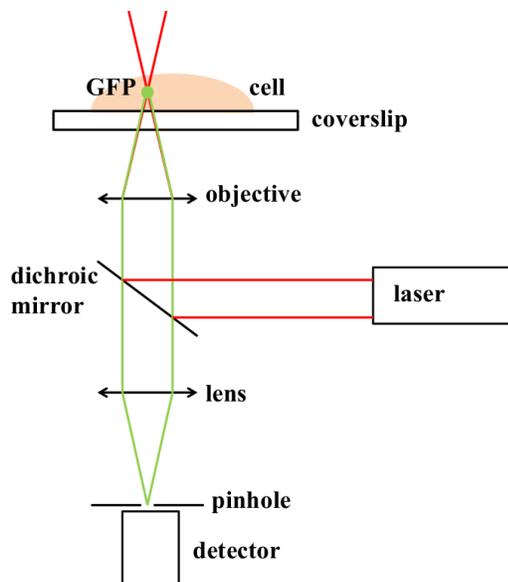


Fig. 1.3.: Confocal microscopy

reduced due to light scattering. Two-photon microscopy [1, 11–14] with near-infrared (IR) illumination can image deeper in tissue because IR light scatters less in deeper tissue. According to Equation 1.1, IR light contains less energy than visible light because the wavelength of IR light is larger than the wavelength of visible light. It is necessary that two or multiple photons must excite the fluorescent molecules simultaneously. Note that “simultaneously” means within about  $10^{-18}$  seconds [12]. Figure 1.4 shows the Jablonski diagram of two-photon microscopy. Using two-photon microscopy, photobleaching and photodamage can be reduced [1]. To visualize in deeper tissue, multi-photon microscopy techniques are also developed [15].

### 1.1.3 Limitations of Fluorescence Microscopy

Fluorescence microscopy images are suffered from lens and a detector in a microscope system. As a fluorescence microscope system is introduced, degradations of fluorescence microscopy images are addressed below.

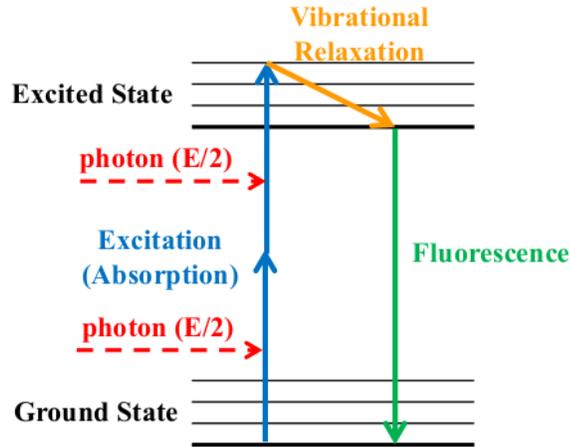


Fig. 1.4.: Jablonski diagram of two-photon microscopy

When the light pass through lens such as an objective lens or a tube lens in confocal microscopy or two-photon microscopy, blur can occur due to a point spread function (PSF) of the lens [16, 17]. The PSF is a diffraction pattern occurred from a point source through a microscope system. The PSF can also be defined as an impulse response of an imaging system. In an imaging system such as confocal microscopy, an image is generated as the convolution of the input light and the PSF. Due to the PSF, the image generated from a fluorescence microscope is blurred and it is extremely difficult to distinguish multiple objects which are overlapped. One way to reduce blurriness occurred by the PSF is increasing numerical aperture.

The emitted light from fluorophores will be captured by detectors [1, 2] to digitize light into an image. Photomultiplier tube (PMT), avalanche photodiode (APD), and multi-pixel detectors such as charge-coupled device (CCD), electron-multiplying CCD (EMCCD), and complementary metal-oxide semiconductor (CMOS) can be used as a detector in a microscope. The main idea of those detectors are from photoelectric effect which is the following: light can eject electrons from a metal. Those electrons ejected by light are called photoelectrons. We will investigate the CCD in detail below which is widely used as a detector in microscopes.

A CCD is composed of a CCD chip to detect photons, a shutter to expose and block light, a window to cover the chip from dusts, electronics for the readout of photoelectrons from pixels on the chip, and a cooler to reduce the thermal noise. A CCD chip, or an imager, is composed of a thin wafer of silicon to trap and hold photoelectrons. The silicon surface is covered with grids called pixels. To capture the emitted light using a CCD, two steps are needed, which are an exposure step and a readout step. During the exposure step, the CCD chip is exposed by the light, and pixels on the CCD chip can accumulate and store photoelectrons until the readout. During the readout step, each row on the parallel register containing accumulated photoelectrons in pixels will be transferred to the serial register which moves the accumulated photoelectrons on each pixel to an amplifier to convert them into the corresponding voltage. When the serial register becomes empty, the next row in the parallel register will be transferred to the serial register repeatedly until the entire parallel register becomes empty. Note pixels on the chip corresponds to pixels on a computer monitor when images are displayed.

An image generated by an ideal detector would be completely black if there is no sample. Unfortunately, even in those cases, the readout from some sensors in the detector is not completely zero. These undesired signals are known as noise [2]. If an input signal to a detector is strong, even with a small noise from the detector, noise may be ignored. If an input signal is weak which is especially true for fluorescence microscopy, noise can degrade images. Note there are only limited photons captured by the detector due to photobleaching which generates weak signals.

There are three main noises generated from a detector: dark current noise, photon noise, and readout noise [2,4,18]. Dark current noise are occurred when sensors detect electrons emitted by thermal motion. As the integration time and the temperature of a CCD chip increase, electrons excited by heat will be detected as noise. A cooler in CCD can reduce the dark current noise. Photon noise, also known as shot noise, happens due to the randomness of photons. Although the light is emitted uniformly, the frequency of photons arriving at a sensor is random. Therefore, photon noise

can be modeled as a Poisson distribution. Note fluorescence images may be highly affected by the Poisson noise due to the weak signals from fluorophores. Lastly, readout noise is generated while electrons detected by sensors are converted into voltage and digitization. Readout noise can be modeled as a Gaussian distribution.

As we investigated the principles of fluorescence microscopy, we observe many limitations: (1) photobleaching reduces the number photons to the detector so that most of the fluorescence microscopy images have low contrast and low signal-to-noise ratio, (2) a point spread function (PSF) from a microscope system blurs the images, (3) a detector causes a mixture of Poisson and Gaussian noises. It is impossible to generate perfect images due to the physical limitations in fluorescence microscopy. To solve these issues, techniques using image processing or machine learning are developed to analyze fluorescence microscopy images which will be described in detail in Chapter 2. Some techniques are available in platforms such as ImageJ [19, 20], Fiji [21], Icy [22], and CellProfiler [23–25].

## 1.2 Challenges

To analyze fluorescence microscopy images/volumes, detection and segmentation of subcellular structures are required steps. For example, an accurate segmentation of tubule borders can identify and characterize a single nephron in a kidney. Also, an accurate detection and segmentation of nuclei can help analyzing the status of tissue.

Analysis of fluorescence microscopy images can be challenging due to the following reasons:

- As mentioned above, fluorescence microscopy images are degraded by noise and blurring caused during image acquisition. When the ideal pixel intensity of fluorescence microscopy images is  $x$ , the real pixel intensity,  $z$  can be modeled as [26–28]:

$$z = y + b \tag{1.2}$$

where  $y \sim \mathcal{P}(x)$  is a Poisson random variable with a mean of  $x$ ,  $b \sim \mathcal{N}(\mu, \sigma^2)$  is a Gaussian random variable with a mean of  $\mu$  and a variance of  $\sigma^2$ . As mentioned above, the low level of fluorescence from photobleaching, low fluorophore concentrations, or short period of exposure time, only a limited number of photons is received in a detector causing noise. A point spread function (PSF) in a microscope system can blur microscopy volumes which can reduce the resolution. Therefore, boundaries of subcellular structures may not be well-defined due to noise and the PSF.

- Multiple structures can be presented in a fluorescence microscopy data set and the structures need to be distinguished to detect and segment specific structures. A single type of fluorophore may label multiple structures. For example, a phalloidin labels both the basement membrane of the tubules and the brush border of the proximal tubules [29, 30]. Additionally, crosstalk [5] can image structures in different color channel. Crosstalk can happen when fluorophores emits light with a wide range of wavelength. When multiple structures are labeled or presented, segmentation can be extremely challenging. For example, a simple thresholding [31] would segment all structures. An advanced technique may be required to distinguish them.
- Subcellular structures may have various sizes, shapes, and intensities [32, 33]. For example, nuclei in fluorescence microscopy volumes may have various sizes and shapes. Some fluorescence microscopy volumes have inhomogeneous intensity [29, 34], so a simple thresholding [31] may lose structures on the boundary of a volume and capture noise in the center of a volume. To correct inhomogeneous intensity, a pre-processing step such as adaptive histogram equalization [35] or inhomogeneous correction process [34] may be required.
- Touching or overlapping structures need to be separated. For example, separating touching or overlapping nuclei is required to count the number of nuclei in a fluorescence microscopy volume. By separating structures it is possible to

analyze structures individually. Watershed technique [36] which can be used to separate touching objects tends to over-segment nuclei due to irregular structure shapes. Separating touching nuclei is an open research problem.

### 1.3 Notation

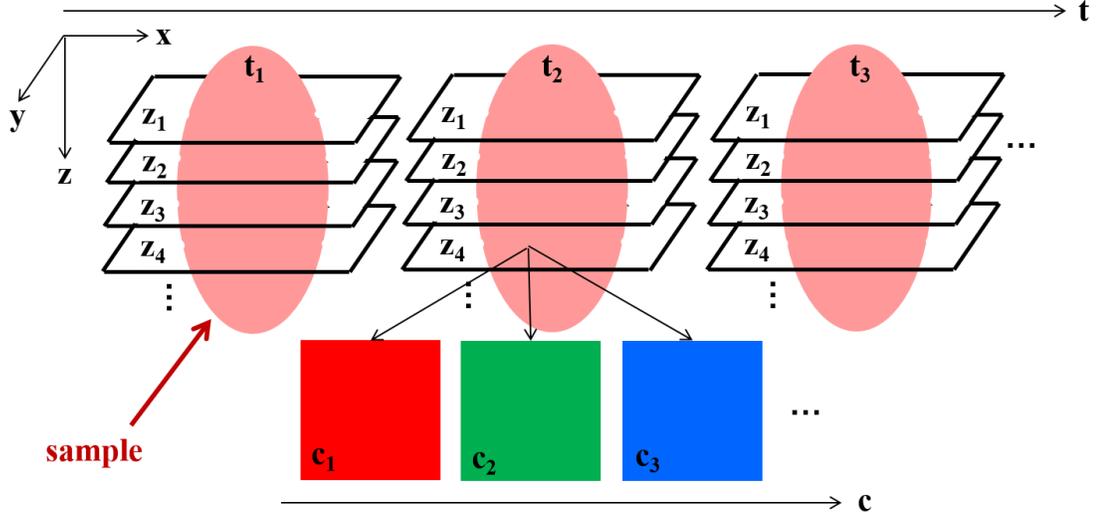


Fig. 1.5.: Notation used in thesis to describe a volume

Fluorescence microscopy data sets can be generated in 5D in width ( $x$ ), height ( $y$ ), depth ( $z$ ), time ( $t$ ), and color channel ( $c$ ), as shown in Figure 1.5. In general, we can represent  $I_{z_p, t_q, c_r}$  as a 2D grayscale image of size  $X \times Y$  with the  $p^{\text{th}}$  focal plane along the  $z$ -direction, the  $q^{\text{th}}$  time sample, and  $r^{\text{th}}$  color channel, where  $p \in \{1, \dots, Z\}$ ,  $q \in \{1, \dots, T\}$ , and  $r \in \{1, \dots, C\}$ . Note  $Z$  is the number of focal planes,  $T$  is the number of time samples, and  $C$  is the number of color channels of a fluorescence microscopy data set. In this thesis, we use data sets in 3D with a single time sample and a single color channel. Therefore, we denote  $I$  as a 3D image volume of size  $X \times Y \times Z$ . Also, we denote  $I_{z_p}$  as the  $p^{\text{th}}$  2D focal plane image of size  $X \times Y$  along the  $z$ -direction. For example,  $I_{z_{23}}^{\text{orig}}$  is the 23<sup>rd</sup> focal plane image of an original volume,

$I^{orig}$ . In addition, we denote  $I_{(q_i:q_f,r_i:r_f,p_i:p_f)}$  as a subvolume of  $I$ , whose  $x$ -coordinate is  $q_i \leq x \leq q_f$ ,  $y$ -coordinate is  $r_i \leq y \leq r_f$ ,  $z$ -coordinate is  $p_i \leq z \leq p_f$ , where  $q_i \in \{1, \dots, X\}$ ,  $q_f \in \{1, \dots, X\}$ ,  $r_i \in \{1, \dots, Y\}$ ,  $r_f \in \{1, \dots, Y\}$ ,  $p_i \in \{1, \dots, Z\}$ , and  $p_f \in \{1, \dots, Z\}$ . It is required to have  $q_i \leq q_f$ ,  $r_i \leq r_f$ , and  $p_i \leq p_f$ . For example,  $I_{(241:272,241:272,131:162)}^{seg}$  is a subvolume of a segmented volume,  $I^{seg}$ , whose  $x$ -coordinate is  $241 \leq x \leq 272$ ,  $y$ -coordinate is  $241 \leq y \leq 272$  and  $z$ -coordinate is  $131 \leq z \leq 162$ .

## 1.4 Data Sets

In this thesis, we analyze two types of data sets.

The first type contains images with tubular structures. There are four data sets belonging to this first type: Data-T1, Data-T2, Data-T3, and Data-T4<sup>1</sup>. Data-T1 is collected from rat kidney using two-photon microscopy. Both the basement membrane of the tubules (we will denote it as “tubule”) and the brush border of the proximal tubules (we will denote it as “lumen”) are labeled with phalloidin. The goal of Data-T1 is to distinguish tubule from lumen to segment the boundary of renal tubules to characterize a single nephron in the kidney. Data-T2, Data-T3, and Data-T4 are collected from liver samples using two-photon microscopy. Both cell boundaries and endothelia are labeled with a fluorescent tomato lectin. Our goal of Data-T2, Data-T3, and Data-T4 is to segment blood vessels and cell-cell junctions to characterize the vascular space and hepatocytes. Figure 1.6 shows sample images of Data-T1, Data-T2, Data-T3, and Data-T4.

The second type contains images with nuclei structures. There are nine data sets belonging to the second type: Data-N1, Data-N2, Data-N3, Data-N4, Data-N5, Data-N6, Data-N7, Data-N8, and Data-N9<sup>2</sup>.

<sup>1</sup>Data-T1 was provided by Malgorzata Kamocka of Indiana University and was collected at the Indiana Center for Biological Microscopy. Data-T2, Data-T3, and Data-T4 were provided by Sherry Clendenon and James Sluka of the Biocomplexity Institute, Indiana University at Bloomington.

<sup>2</sup>Data-N1 was provided by Malgorzata Kamocka of Indiana University and was collected at the Indiana Center for Biological Microscopy. Data-N2, Data-N3, and Data-N4 were provided by Tarek Ashkar of the Indiana University School of Medicine. Data-N5, Data-N6, and Data-N8 were provided by Kenneth W. Dunn of the Indiana University School of Medicine. Data-N7 was provided by Sherry

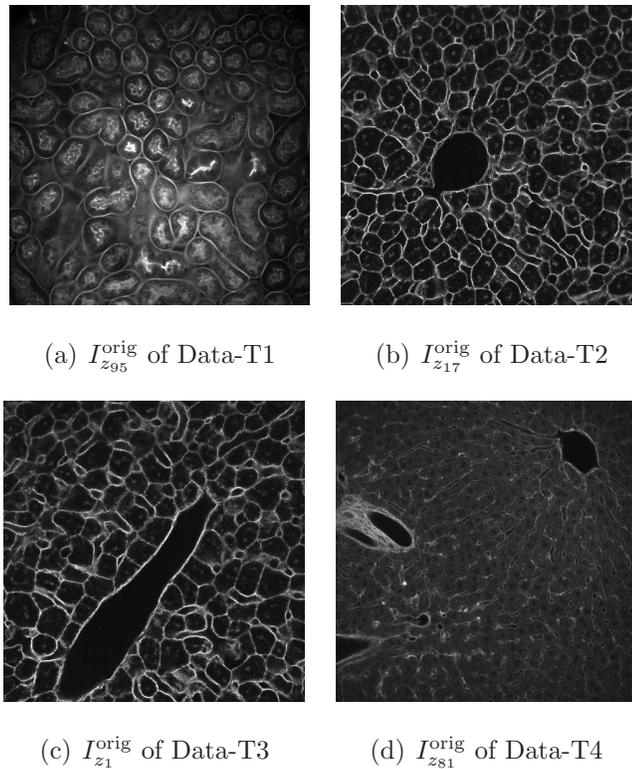


Fig. 1.6.: Sample images of data sets containing tubular structures

Data-N1, Data-N2, Data-N3, Data-N4, Data-N5, Data-N6, Data-N7, and Data-N8 are collected from rat kidney using two-photon microscopy where nuclei are labeled with Hoechst 33342. Data-N9 is collected from mouse intestine using two-photon microscopy where nuclei are labeled with DAPI. The goal is to segment nuclei by rejecting other subcellular structures. Figure 1.7 shows sample images of Data-N1, Data-N2, Data-N3, Data-N4, Data-N5, Data-N6, Data-N7, Data-N8, and Data-N9.

Table 1.1 lists the size of data sets in  $x$ ,  $y$ , and  $z$ -directions.

---

Clendenon collected while at the Indiana Center for Biological Microscopy. She is currently at the Department of Intelligent Systems Engineering of Indiana University. Data-N9 was provided by Mike Ferkowicz of the Indiana University School of Medicine.

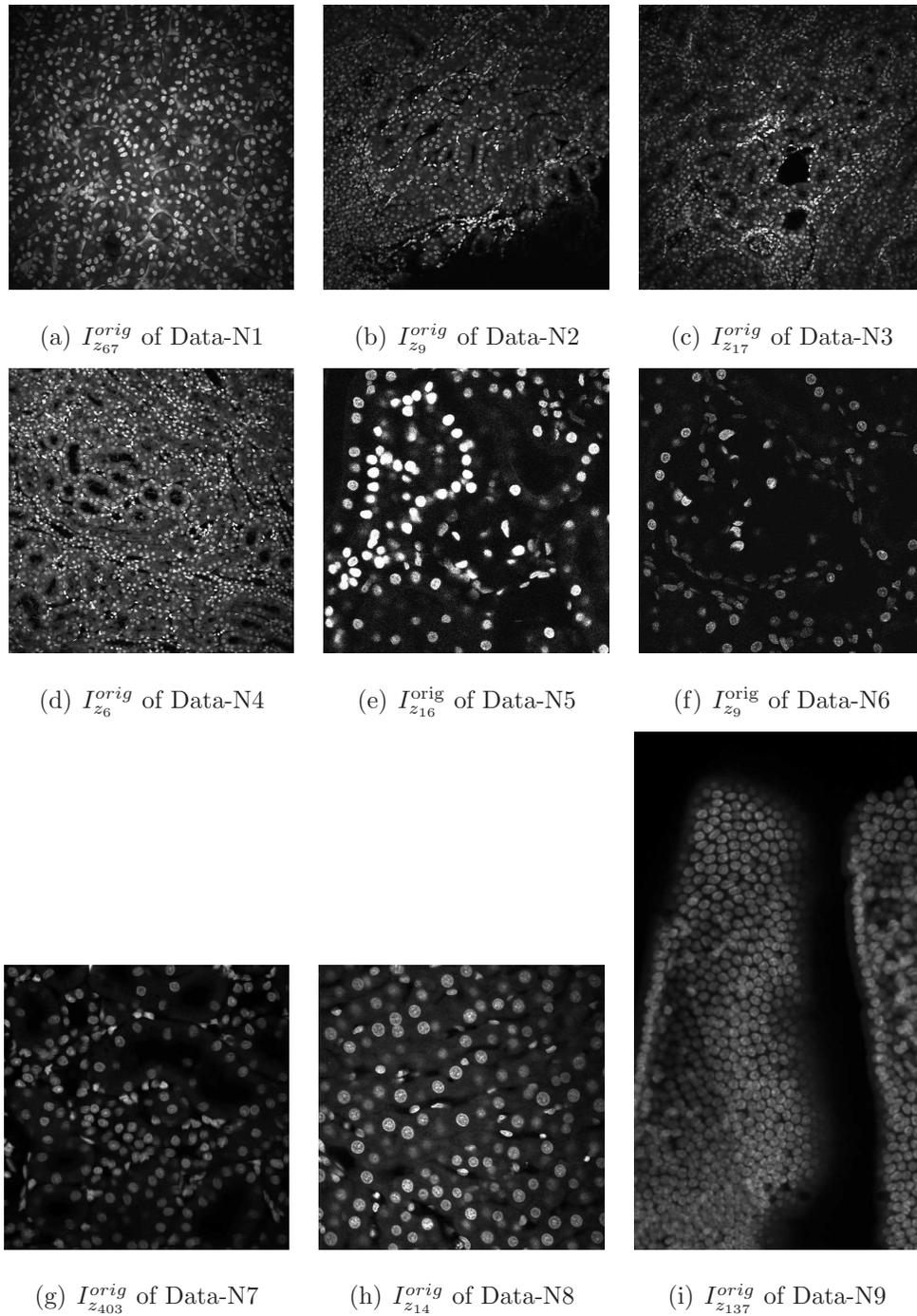


Fig. 1.7.: Sample images of data sets containing nuclei structures

## 1.5 Contributions Of The Thesis

In this thesis we describe multiple methods to analyze fluorescence microscopy images<sup>3</sup>.

<sup>3</sup>The work described in Chapter 4 was jointly done with Dr. Chichen Fu of Purdue University. The work described in Chapter 5 was jointly done with Dr. Chichen Fu of Purdue University. The

Table 1.1.: The size of data sets

Name	X	Y	Z
Data-T1	512	512	512
Data-T2	512	512	36
Data-T3	512	512	1
Data-T4	512	512	156
Data-N1	512	512	512
Data-N2	512	512	36
Data-N3	512	512	41
Data-N4	512	512	23
Data-N5	512	512	45
Data-N6	512	512	26
Data-N7	512	512	415
Data-N8	512	512	32
Data-N9	512	930	157

This thesis focuses on segmentation and detection of subcellular structures presented in three dimensional fluorescence microscopy images. We mainly focus on two types of microscopy data sets. Our method to analyze images of the first type containing tubular structures is described in Chapter 3. Our methods to analyze images of the second type containing nuclei structures are described in Chapter 4, Chapter 5, Chapter 6, and Chapter 7. Our color labeling method for nuclei segmentation is described in Chapter 8. The main contributions of this thesis are listed as follows:

- Boundary Segmentation Using Steerable Filters

---

work described in Chapter 6 was jointly done with Ms. Shuo Han and Dr. Chichen Fu of Purdue University. The work described in Chapter 7 was jointly done with Dr. Chichen Fu and Mr. Daniel Mas Montserrat of Purdue University. The work described in Chapter 8 was jointly done with Mr. Soonam Lee of Purdue University.

We describe a method to segment tubular boundary of biological structures. To segment foreground in fluorescence microscopy images having non-uniform intensity a 3D adaptive histogram equalization is used to correct inhomogeneity. The main challenge of this work is to segment only tubule structures while other structures such as lumen are presented in fluorescence microscopy images. To distinguish tubule in foreground from lumen, steerable filters [37] are used because steerable filters can capture strong directional tendencies on tubule. Our method can successfully segment tubule boundaries in various data sets from rat kidney and liver.

- Nuclei Segmentation Using Convolutional Neural Networks

We present a nuclei segmentation method of fluorescence microscopy images. We design and train three dimensional convolutional neural networks (CNNs). A large set of labeled training volumes is necessary to train CNNs. The labeling process especially in 3D is tedious. In this work we generate synthetic microscopy volumes to train our CNNs. We assume nuclei are in ellipsoidal shapes and we add a blurring operation modeled from a PSF and a noise operation which is a mixture of a Poisson noise and a Gaussian noise. Our 3D CNNs trained by synthetic microscopy volumes can successfully segment nuclei in real microscopy volumes without using any real ground truth volumes.

- Nuclei Detection and Segmentation Using Convolutional Neural Networks

We introduce a nuclei detection and segmentation method to label nuclei distinctly. To detect center locations of nuclei a 3D adaptive histogram equalization [38], a 3D distance transform, and a 3D classification CNN are used. After the detection stage, each nucleus is distinctly segmented using a segmentation CNN in a 3D patch surrounding the nucleus. The segmentation CNN is trained by a set of synthetic microscopy volumes. Due to the detection stage, non-nuclei structures presented in real fluorescence microscopy volumes are rejected which improves our segmentation accuracy.

- Center-Extraction-Based Nuclei Instance Segmentation

We describe a two-stage method of segmenting nuclei in distinct labels. Our first CNN detects the center locations of nuclei and generates a binary segmentation mask. Using the nuclei center locations and the binary segmentation mask from the first CNN, our second CNN individually segments nuclei in 3D patches surrounding each nucleus. To train our CNNs, realistic synthetic volumes generated by a spatially constrained cycle-consistent adversarial network (SpCycleGAN) [39] are used without using any real ground truth volumes. Our method can detect and segment nuclei in real microscopy volumes accurately.

- Nuclei Detection Using Sphere Estimation Network

We develop a Sphere Estimation Network (SphEsNet) which can detect nuclei in fluorescence microscopy volumes. Our network is a fully three dimensional CNN. In this work we assume nuclei presented in fluorescence microscopy volumes are spherical. Our network not only finds the locations of nuclei but also estimates the radii of nuclei. The SphEsNet is trained by realistic synthetic microscopy volumes using the SpCycleGAN [39] without using any real ground truth volumes. Our evaluation shows that the spheres at the estimated locations with the estimated radii generated by the SphEsNet highly overlap with nuclei presented in multiple real microscopy data sets.

## 1.6 Publications Results From Our Work

### Journal Papers

1. **D. J. Ho**, C. Fu, D. Mas Montserrat, P. Salama, K. W. Dunn and E. J. Delp, “Sphere Estimation Network: Three Dimensional Nuclei Detection of Fluorescence Microscopy Images”, *To be submitted to the IEEE Transactions on Medical Imaging*.

2. C. Fu, S. Han, S. Lee, **D. J. Ho**, P. Salama, K. W. Dunn and E. J. Delp, “Three Dimensional Nuclei Synthesis and Instance Segmentation”, *To be submitted to the IEEE Transactions on Medical Imaging*.

### Conference Papers

1. **D. J. Ho**, S. Han, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Center-Extraction-Based Three Dimensional Nuclei Instance Segmentation of Fluorescence Microscopy Images”, Submitted to *Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics*, May 2019, Chicago, IL.
2. C. Fu, S. Lee, **D. J. Ho**, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, “Three Dimensional Fluorescence Microscopy Image Synthesis and Segmentation,” *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 2302-2310, June 2018, Salt Lake City, UT.
3. **D. J. Ho**, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei Detection and Segmentation of Fluorescence Microscopy Images Using Three Dimensional Convolutional Neural Networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 418-422, April 2018, Washington D.C.
4. **D. J. Ho**, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei Segmentation of Fluorescence Microscopy Images Using Three Dimensional Convolutional Neural Networks,” *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 834-842, July 2017, Honolulu, HI.
5. C. Fu, **D. J. Ho**, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei Segmentation of Fluorescence Microscopy Images Using Convolutional Neural Networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 704-708, April 2017, Melbourne, Australia.

6. **D. J. Ho**, P. Salama, K. W. Dunn, and E. J. Delp, "Boundary Segmentation for Fluorescence Microscopy Using Steerable Filters," *Proceedings of the SPIE Conference on Medical Imaging*, pp. 101330E-1-11, February 2017, Orlando, FL.

## 2. LITERATURE REVIEW

In this chapter we will review previous techniques related to segmentation and detection of fluorescence microscopy images using computer vision, image processing, and machine learning methods.

### 2.1 Computer Vision and Image Processing Based Methods

Computer vision and image processing are research fields for analyzing images which are multi-dimensional arrays [40]. The goal of this thesis is to “segment” subcellular structures in fluorescence microscopy images. Therefore, we will focus on reviewing various segmentation techniques in this section.

One of the simplest methods of segmenting structures in foreground is thresholding [31, 40]:

$$I^{\text{seg}}(\mathbf{v}) = \begin{cases} 255, & \text{if } I^{\text{orig}}(\mathbf{v}) \geq \tau \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

where  $I^{\text{orig}}$  is an input (original) image,  $I^{\text{seg}}$  is an output (segmented) image, and  $\tau$  is the thresholding value. One of the challenges is to select the threshold to segment foreground regions accurately. Otsu developed a method to select the threshold automatically by maximizing an inter-class variance [41]. Some fluorescence microscopy images have inhomogeneous intensities, so global thresholding may not successfully segment structures especially on the boundary of volumes with lower pixel intensities. Adaptive thresholding methods may help enhancing structures in the boundary of volumes. For example, the thresholding value of each pixel can be selected by a mean value of a local window [42].

After segmenting foreground regions, the next step is to separate touching or overlapping structures. A simple connected component analysis [40] cannot separate

touching or overlapping structures. One method to separate touching structures is to use morphological operations [40]. The erosion of set  $A$  by set  $B$  is defined as

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2.2)$$

where  $(B)_z = \{c | c = b + z \text{ for } b \in B\}$  is the translation of set  $B$ . Similarly, the dilation of set  $A$  by set  $B$  is defined as

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (2.3)$$

where  $\hat{B} = \{w | w = -b \text{ for } b \in B\}$  is the reflection of set  $B$ . Morphological operations with the erosion and the dilation can be used to separate structures. Morphological multiscale decomposition is described in [43] to separate cell clusters. Markers for individual cell are selected during multiscale erosions. When the markers are found, dilation is performed to recover the markers to the original cell. It is still challenging to separate overlapping objects using morphological operations especially when the overlapping region between two structures is large.

Watershed [36, 40] can be used to separate touching or overlapping structures. Watershed has been used for cell segmentation [31–33]. Let us consider an image as a landscape where the pixel intensity illustrates its elevation. As water is flooded into the landscape, the water would be filled from low elevation, or local minima of an image. The local minima are denoted as watershed markers. One can construct dams between local minima to avoid merging different catchment basins. The dams are denoted as watershed lines. As the watershed lines are built, touching regions are split. The landscape can be generated by using a distance transform [44] of a thresholded image. One drawback of watershed is that watershed tends to over-segment structures when the structures are in irregular shapes. To avoid over-segmentation for nuclei segmentation in time-lapse microscopy, marker-controlled watershed is described [45]. Watershed markers are selected by condition erosion where the erosion occurs only when the size of the region is larger than a thresholding value. Additionally, fine erosion structures and coarse erosion structures are used based on the shape of the

region. Marker-controlled watershed can segment nuclei more accurately than the classical watershed technique [40]. Another marker-controlled watershed is introduced in [46] to segment surface-stained living cells in 3D where watershed markers are automatically selected after ridge enhancement.

To segment biological structures in irregular shapes, deformable models such as active contours [47], also known as snakes, are used in microscopy data sets [33, 48]. Active contour is an optimization problem to minimize an energy. As the energy is being minimized, a curve iteratively evolves towards the boundary of objects. The energy can be formulated as a sum of three terms:

$$E_{ac}(C) = E_{int}(C) + E_{ext}(C) + E_{const}(C) \quad (2.4)$$

where  $C$  is a curve,  $E_{int}$  is defined as an internal energy to smooth the curve,  $E_{ext}$  is defined as an external energy to attract the curve to the region of interest, and  $E_{const}$  is defined as a constraint energy to interact between a user and the curve. Then the final contour,  $C_{final}$ , would be

$$C_{final} = \arg \min_C E_{ac}(C) \quad (2.5)$$

Active contours are used to segment various biological structures such as nerve fibers [49], *Entamoeba histolytica* cells [50], and leukocytes (white blood cells) [51].

The main challenges for active contours are the following: (1) initial curves need to be manually placed near the region of interest, (2) curves tend to be sensitive to noise presented in fluorescence microscopy images, (3) it is challenging to capture the region of interest when concavities are presented. To solve these issues, gradient vector flow is used in the external energy [52, 53]. Vector field convolution is used in the external energy to improve segmentation performance to segment concave objects accurately especially when heavy noise is presented [54]. An automatic initialization method using Poisson inverse gradient is addressed in [55].

The previous methods [47, 49–55] are edge-based active contour models where the energy in Equation 2.4 is formulated as

$$E_{ac}(C) = \alpha_1 \int_0^1 |C'(s)|^2 ds + \alpha_2 \int_0^1 |C'''(s)|^2 ds - \alpha_3 \int_0^1 |\nabla I(C(s))|^2 ds \quad (2.6)$$

to evolve curves on edges which are the boundaries of objects.  $I$  is an image,  $C(s) : [0, 1] \rightarrow \mathbb{R}^2$  is a parametrized curve, and  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  are positive parameters. It is difficult for the edge-based active contour models to find edges when images are blurred and noisy such as fluorescence microscopy images. Inspired by the Mumford-Shah functional [56], Chan and Vese introduced region-based active contour model [57] where the energy can be formulated as:

$$\begin{aligned} E_{ac}(C) &= \lambda_{in} \int_{inside(C)} |I(x, y) - c_{in}|^2 dx dy \\ &+ \lambda_{out} \int_{outside(C)} |I(x, y) - c_{out}|^2 dx dy \\ &+ \mu \text{Length}(C) + \nu \text{Area}(inside(C)) \end{aligned} \quad (2.7)$$

where  $\lambda_{in}$ ,  $\lambda_{out}$ ,  $\mu$ ,  $\nu$  are weighting parameters greater than zero,  $c_{in}$  and  $c_{out}$  are the mean intensity of  $I$  inside  $C$  and outside  $C$ , respectively.

Using the level set method from [58], [57] defines a level set function  $\phi : \Omega \rightarrow \mathbb{R}$  such that

$$\begin{cases} C = \partial\omega = \{(x, y) \in \Omega : \phi(x, y) = 0\} \\ inside(C) = \omega = \{(x, y) \in \Omega : \phi(x, y) > 0\} \\ outside(C) = \Omega \setminus \bar{\omega} = \{(x, y) \in \Omega : \phi(x, y) < 0\} \end{cases} \quad (2.8)$$

where  $\omega \subset \Omega$  and  $C = \partial\omega$ . Then in Equation 2.7,  $C$  can be replaced by  $\phi$  in the energy:

$$\begin{aligned} E_{ac}(\phi) &= \lambda_{in} \int_{\Omega} |I(x, y) - c_{in}|^2 H(\phi(x, y)) dx dy \\ &+ \lambda_{out} \int_{\Omega} |I(x, y) - c_{out}|^2 (1 - H(\phi(x, y))) dx dy \\ &+ \mu \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\ &+ \nu \int_{\Omega} H(\phi(x, y)) dx dy \end{aligned} \quad (2.9)$$

where  $H$  is the Heaviside function and  $\delta$  is the Dirac function. With an initial  $\phi$  at  $t = 0$ , minimizing  $E_{ac}$  with respect to  $\phi$  can be done according to

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ -\lambda_{in}(I - c_{in})^2 + \lambda_{out}(I - c_{out})^2 + \mu \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \nu \right] \quad (2.10)$$

Region-based active contour models have been used in fluorescence microscopy images because they can segment objects accurately when blurring and noise are presented. In [59], the active contour model in [57] is expanded in three dimension, known as an active surface model, to segment nuclei in three dimensional fluorescence microscopy volumes from rat kidney. The work in [59] is further developed in [34] by including an inhomogeneity correction [60] to segment nuclei when intensity inhomogeneity is presented in fluorescence microscopy volumes.

To segment structures distinctly, multiple active surfaces can be used. Coupled active surfaces are developed to segment and track cells in 3D+time microscopy volumes [18]. In [18] multiple active surfaces are evolved to segment individual cells. A penalty term for overlapping surfaces and a constraint term for volume conservation are included in the energy to be minimized to separate touching cells. An improvement of the coupled active surfaces in [18] is done in [61] by incorporating the Radon transform, a non-partial differential equation (PDE)-based energy minimization method, and the watershed technique.

Active contour models are further developed including statistical modeling of images. A stochastic active contour scheme (STACS) is designed with the energy including an edge-based term, a region-based term, and a shape-based term to segment heart structures in cardiac MR images [62]. Active mask [63] is introduced by modifying [62] to segment cells in fluorescence microscopy images which have punctate pattern. In [63] active mask uses multiresolution, multiscale, and region-growing methods with an active contour model.

Many alternative methods have been developed used in fluorescence microscopy images. A model-based nuclei segmentation method by identifying nuclei using an attributed relational graph of nuclei boundaries and delineating them using a region-growing technique is presented in [64]. In [65], a method to segment fluorescently labeled endosomes when the number of object is unknown is presented using region-competition technique [66]. A method coupling image restoration and segmentation is introduced as Squassh (segmentation and quantification of subcellular shapes) [67,

68]. In [69], a method of counting and segmenting nuclei using a marked point process (MPP) [70] is introduced. Alternatively, a method to detect cells with various size using a denoising technique, a thresholding technique, a fast radial symmetric transform, and a dilation-based non-maxima suppression is described in [28].

The techniques described above cannot distinguish various structures presented in fluorescence microscopy images. Due to problems of a single type of fluorophores labeling multiple subcellular structures [29, 30] and crosstalk [5], distinguishing different structures can be an important task in fluorescence microscopy images. To segment a specific structure and reject other structures, a more careful analysis is needed.

The methods below are specifically designed to segment tubular structures in biomedical images. A minimal path technique by globally minimizing an energy is developed to detect boundaries between two points in 2D medical angiographic images [71]. In [72] the minimal path technique is extended into 3D to segment brain vessels in magnetic resonance angiography (MRA) images. Alternatively, a model-based segmentation method is used for vessels in 3D MRA images where the model couples a central vessel axis and a vessel wall surface [73]. A deformable model is designed in [74] where the curve evolution is controlled by geodesic active contours. A 4D curve method is developed by combining a minimal path technique [72] and active contour models [73, 74] used for vessel segmentation in 3D brain MRA images and aorta segmentation in 3D computed tomography angiography (CTA) images. A 4D curve represents with different points on the centerline (3D) with their radii (1D):

$$\tilde{C}(s) = (C(s), r(s)) \quad (2.11)$$

where  $\tilde{C}$  is a 4D curve representing a 3D tubular structure surface,  $C$  is a 3D location of the centerline,  $r$  is a radius of a sphere centered at  $C$ , and  $s \in [0, 1]$ . As a 3D vessel surface is represented as a 4D curve, a minimal path is used in 4D. Furthermore, Optically Oriented Flux (OOF) [75] is used for a faster propagation to the vessel's centerline [76]. Similarly, a vessel tree model requiring fewer degrees-of-freedom can

stably capture the centerline [77]. Only a few methods [29, 30] have investigated to segment tubule boundary in a rat kidney which is still an open research problem.

## 2.2 Machine Learning Based Methods

Machine learning is a field that a system can automatically learn some meaningful features in a training set to make decisions [78, 79]. One strength of machine learning methods is that the system can continue to learn and refine their performances when additional training sets are provided. Many decision-making techniques have been introduced such as a nearest neighbor classifier [78], a decision tree classifier [78], a random forest classifier [80], and a support vector machine (SVM) [78, 81]. For example, an interactive segmentation toolkit called ilastik [82] is developed to segment structures in biomedical images. A user can select initial masks/regions of multiple types of structures. Then a random forest classifier [80] trained by the initial masks labels pixel in an image based on features of color, edge, texture, and orientation. If the prediction is not satisfied, the user can label more masks until the prediction is satisfied.

Recently, deep learning [83, 84] has been shown remarkable performances in computer vision [85], natural language processing [86, 87], and graph analysis [88, 89]. Deep learning is a type of machine learning using neural networks with a series of multiple layers with non-linear activation functions. Deep learning was introduced several decades ago. For example, LeNet-5 trained by MNIST dataset can recognize digits [90]. Deep learning becomes more popular in a recent few years due to the advancement of graphics processing units (GPUs) [91] and the availability of public data sets such as ImageNet [92, 93], PASCAL VOC [94, 95], Microsoft COCO [96], and ADE20K [97, 98]. With the labeled data sets, the neural network can be trained based on its training loss function using optimization techniques such as stochastic gradient descent (SGD) [99] with momentum or the Adam optimizer [100]. Many deep learning frameworks such as Caffe [101], Caffe2, TensorFlow [102], Torch [103],

and Pytorch [104] are implemented. For the rest of the section, we will first address detection and segmentation methods using convolutional neural networks (CNNs) in computer vision. Then we will describe how these methods are used to analyze biomedical images. Lastly, we will describe generative adversarial networks (GANs) which are used to produce synthetic training images to train CNNs.

### 2.2.1 Detection and Segmentation Using Convolutional Neural Networks

Convolutional neural networks (CNNs) become a powerful tool in computer vision because local features are invariant in locations [83]. For example, if a convolutional filter can detect a nucleus in one location, the same filter with the same weights can highly detect another nucleus in a different location. The first few layers of the CNNs detect simple features such as edges whereas the deeper layers can incorporate previous features to detect more sophisticated features [105, 106]. CNNs have shown many progress in image classification, object detection, and image segmentation which are described in detail below.

Image classification is one of the popular topics in computer vision and many outstanding techniques have been developed from challenges such as ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). AlexNet [85] was the first CNN-based winner of ILSVRC. AlexNet uses five convolutional layers, three max-pooling layers, and three fully connected layers. AlexNet uses a Rectified Linear Unit (ReLU) as their non-linear activation function because it avoids vanishing gradient problem from the Sigmoid function or the tanh function. The ReLU function is defined as

$$f(x) = \max(x, 0) \tag{2.12}$$

AlexNet uses data augmentation techniques to avoid overfitting problems. Data augmentation is a technique to multiply labeled training images to produce synthetic images using simple transformation such as image translation, horizontal reflection, and color jittering. To have a large receptive field, AlexNet uses a large convolutional filters. For example, the filter size of the first convolutional layer is  $11 \times 11$  and the

filter size of the second convolutional layer is  $5 \times 5$ . To reduce the number of filter parameters, VGGNet [107] uses a series of convolutional filters with small size such as  $3 \times 3$ . For example, to have a receptive field of  $7 \times 7$ ,  $49C^2$  parameters are required using one  $7 \times 7$  filter, but  $3 \times 9C^2 = 27C^2$  parameters are required using three consecutive  $3 \times 3$  filters, where  $C$  is a channel number for the input and the output. Inception modules [108] are developed to have various receptive fields using  $1 \times 1$  convolutional filters,  $3 \times 3$  convolutional filters, and  $5 \times 5$  convolutional filters.  $1 \times 1$  convolutional bottleneck filters are used to avoid expansive computations by reducing the number of channels before  $3 \times 3$  and  $5 \times 5$  convolutional filters. By stacking multiple Inception modules, GoogLeNet [108] is introduced with a high classification performance. More recently, residual network (ResNet) is described in [109]. ResNet stacks residual blocks to go deeper such as 152 layers. A residual block contains two convolutional layers with a shortcut connection so the output of the residual block becomes  $F(x) + x$  where  $x$  is an input and  $F(x)$  is an output of two convolutional layers. With shortcut connections, it becomes possible to train deeper layers without degradation.

With the advancement in image classification, many object detection methods have been introduced [110]. Object detection is a task where objects in an image are detected and classified in bounding boxes. R-CNN [111] is developed for object detection. First of all, nearly 2000 region proposals are chosen using selective search [112]. After warping each region proposals, AlexNet [85] extracts features which are used to classify objects in region proposals using a linear SVM classifier. One of the drawbacks of R-CNN is an expansive computation because the CNN needs to process all region proposals. Fast R-CNN [113] uses a pre-trained CNN [107] to first generate a feature map on an entire input image. Then, region proposals on the input image are projected on the feature map and a region of interest (RoI) pooling layer extracts the corresponding features. The RoI pooling layer converts the projected regions into a small region in a fixed size using max pooling. Lastly, the RoI pooled regions are used to classify objects and regress bounding boxes. Majority computations in Fast

R-CNN are spent during the region proposal stage [112]. Therefore, a region proposal network (RPN) is developed in Faster R-CNN [114]. In the RPN, a small network is slid on a feature map generated by a pre-trained CNN. At each location, or anchor, various anchor boxes with different scales and aspect ratios are used to produce the classification (if there is an object or not) scores and bounding box coordinates. To avoid detecting multiple bounding boxes for one object, non-maximum suppression (NMS) is used on the classification score. Once the locations of objects with their bounding box coordinates are selected, then the object is classified after the RoI pooling layer as it is done in Fast R-CNN [113]. There are some other object detection approaches using a single-stage such as single shot multibox detector (SSD) [115] and you only look once (YOLO) [116–118] to be used in a light computational device but they do not perform as good as Faster R-CNN [114].

CNNs have impacted image segmentation. Semantic segmentation, a type of image segmentation, is defined as a pixel-wise classification problem where each pixel is classified to one of objects [119]. One of the naive methods for semantic segmentation is to classify each pixel in a patch centered the pixel to be classified [120]. The method in [120] is computationally inefficient because there are redundant computations on overlapping patches. For more efficient computation, Fully Convolutional Network (FCN) is designed [121, 122]. FCN is composed of an encoder and a decoder. An encoder finds low dimensional features and a decoder reconstructs feature map into the size of the original image. FCN uses an architecture from VGGNet [107] as the encoder but replaces fully connected layers to a transposed convolutional layer as the decoder. Transposed convolutional layers both convolve and upsample feature maps. One drawback of FCN is that the spatially detailed information can be lost during transposed convolutional layers with a large upsampling rate. To resolve this issue, many semantic segmentation networks have been developed. Instead of one transposed convolutional layer with a large upsampling rate, DeconvNet [123] contains multiple transposed convolutional layers with a small upsampling rate. SegNet [124] introduced max unpooling layers. A max unpooling layer in the decoder

is paired with the corresponding max pooling layer in the encoder where the pooling indices in the max pooling layer is stored and used during the max unpooling operation. U-Net [125] not only uses multiple transposed convolutional layers with a small upsampling rate but also introduces skip connections between an encoder and a decoder to transfer feature maps. Alternatively, DilatedNet [126] is introduced using dilated convolutions according to Equation 6.4. Dilated convolutions are originally developed for efficient wavelet transform [127] and they are used in [126] to exponentially increase receptive fields in convolutional layers. DilatedNet [126] develops a context module after their prediction module to refine segmentation map where the receptive field of the last layer in the context module is larger than the size of an input patch. Pyramid scene parsing network (PSPNet) is developed in [128] where a pyramid pooling module uses both local and global context information for accurate segmentation performance. RefineNet can achieve high resolution segmentation by fusing feature maps from various resolutions [129]. More recently, a series of DeepLab is presented [130–133]. DeepLab [130–133] uses atrous convolutions, also known as dilated convolutions [126], and develops atrous spatial pyramid pooling (ASPP) layer containing multiple filters with various atrous factors. In addition, DeepLab incorporates conditional random fields (CRFs) [134] to increase segmentation accuracy and depthwise separable convolutions and the Xception model [135] for efficient performance.

Due to the advancement in object detection and semantic segmentation, instance segmentation methods have been introduced. Instance segmentation, a type of image segmentation, is a task where objects are detected and segmented in distinct masks. One of the most popular instance segmentation methods is Mask R-CNN [136]. Mask R-CNN is developed from Faster R-CNN by adding the third branch which is a segmentation branch using the FCN [121, 122]. To avoid an expensive annotation process to train CNNs for instance segmentation, many techniques are being developed [137, 138].

### 2.2.2 Detection and Segmentation of Biomedical Images Using Convolutional Neural Networks

With the advancement of convolutional neural networks in computer vision, CNNs have made a big impact in biomedical image analysis especially for detection and segmentation problems [139]. To accurately detect location of objects in microscopy images, many CNN-based methods have been introduced. A patch-based method to detect mitosis in breast cancer histology images is described in [140]. A patch sliding over an input image generates a probability of the center pixel of the patch being to mitosis. Local maxima of the smoothed probability map are selected the centroids of mitosis. Similarly, Tyrosine Hydroxylase-containing cell detection method in larval zebrafish brain images uses a patch-based approach [141]. To collect a training set for the CNN in [141], an SVM classifier [142] is used to select patches centered at pixels in cells and background. One problem of patch-based methods is that they require an expensive computation because there are redundant convolutional operations for overlapping patches. Deep voting [143] uses fast scanning [144] for an efficient computation and produces a voting offset vector and a voting confidence vector to detect nuclei in microscopy images. Alternatively, a structured regression method is introduced where higher values are assigned toward the center of nuclei in a proximity mask for training a CNN [145]. In [146] a stacked sparse autoencoder (SSAE) is used to detect nuclei in breast cancer histopathology images. A spatially constrained CNN (SC-CNN) is developed where the CNN contains a parameter estimation layer and a spatially constrained layer for more accurate nuclei detection [147]. To accurately detect the location of nuclei, a method in [148] convolves ground truth points and the output of the CNN is deconvolved to produce nuclei center points. These methods can successfully find centroids of nuclei as a set of coordinates, but they do not generate segmentation masks of nuclei.

One of the most popular segmentation CNNs widely used in biomedical community is known as U-Net [125]. As mentioned above, U-Net is composed of an encoder part

and a decoder part and skip connections transfer feature maps from the encoder part to the decoder part to preserve spatial information which can be lost during pooling layers, as shown in Figure 2.1. Data augmentation techniques such as random elastic deformation are used to train the U-Net using 30 labeled images. U-Net [125] won the ISBI EM segmentation challenge 2012 [149] and the ISBI cell tracking challenge 2014 and 2015 [150].

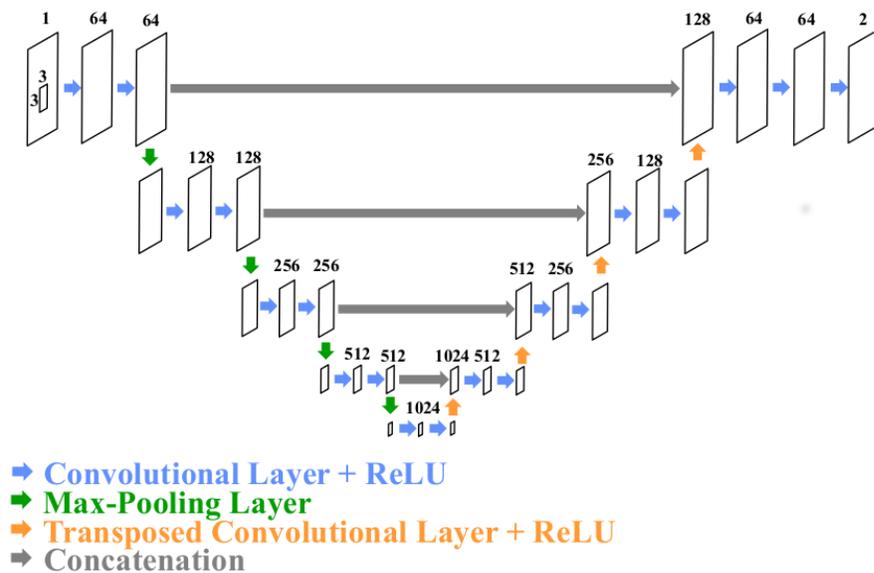


Fig. 2.1.: U-Net Architecture

U-Net [125] has been implemented and used in many platforms such as CellProfiler [25] and ImageJ [19]. CellProfiler is a platform specifically designed for cell analysis for biomedical images. CellProfiler provides many image processing tools such as Watershed technique and morphological operations. Users can select image processing tools provided by CellProfiler and build their own pipeline to analyze their data sets. Recently, a plugin known as ClassifyPixels-Unet is implemented in CellProfiler using a pretrained U-Net [125] to segment nuclei in input images. ClassifyPixels-Unet produces output images with three labels: nuclei interior, nuclei boundary, and background. This plugin can be included in a pipeline to analyze nuclei in

fluorescence microscopy images. To use this plugin, users need to install TensorFlow [102] and can run on local CPUs or GPUs. ImageJ [19] is another popular platform to analyze biomedical images and many image processing plugins are developed and shared with the public. More recently, an ImageJ plugin for cell counting, detection, and segmentation using U-Net [125] is presented in [151]. In this ImageJ plugin, pretrained networks are provided. To adapt the networks to users' own data sets, tools to train the networks are provided as well. More specifically, users can use the plugin to label their own data sets. After using data augmentation techniques such as rotation and elastic deformation, the labeled images can be used to train the networks to analyze input images more accurately. This adaptation step can be done either using local GPUs or using cloud services.

There are several other CNNs for segmenting subcellular structures. In [152] a multiple-input multiple-output (MIMO) network is described to segment cells in various size in fluorescence microscopy images. The work in [152] is extended in [153] to segment glands in histology images. To segment each gland in distinct labels, deep contour-aware network (DCAN) is developed [154]. The DCAN produces two outputs, gland objects and gland boundary. After separating touching gland using the boundary information, connected component can label each gland distinctly. The DCAN won the Gland Segmentation Challenge Contest in MICCAI 2015 [155]. In [156] the DCAN is further tested on nuclei segmentation. Similarly, a set of hematoxylin and eosin (H&E)-stained images is released where the annotation is done in three classes, foreground (inside nuclei), background (outside nuclei), and nuclei boundary, to separate touching nuclei [157]. To separate touching nuclei, assigning higher weights on cell contours to overcome the class imbalance problem during a training step is described in [158]. Similarly, a stain-aware multi-scale (SAMS) network is developed to assign higher weights on nuclei boundary to ensure separating touching nuclei [159]. In [160] detecting individual cells in star-convex polygons is described. An object probability and radial distances in 32 directions are produced using an U-Net architecture [125]. Alternatively, a cell proposal network (CPN) [161], similarly done as

a region proposal network (RPN) [114], produces bounding boxes surrounding cells. A cell segmentation proposal network [162] is further developed from [161] to generate not only bounding boxes but also masks on cells by adding region of interest (ROI) pooling layer [113]. In [163] Mask R-CNN [136] is directly used to label nuclei distinctly in microscopy images. A neural cell detection and segmentation method based on a SSD detector [115] is described in [164].

The work mentioned above are done using 2D operations. Some modalities including fluorescence microscopy generate images in various depth to produce 3D volumes. Instead of independently analyzing the images in 2D without using the depth information, analyzing them in 3D can produce a better segmentation performance. A triplanar CNN architecture is described in [165] to segment knee cartilage in MRI images. The outputs of three independent CNNs for  $xy$ ,  $xz$ , and  $yz$  planes are concatenated and processed in the last layer to classify each voxels. Similarly, a CNN with three input channels,  $xy$ ,  $xz$ , and  $yz$  planes, is developed to segment lymph node in computed tomography (CT) [166]. We described a 3D segmentation method using a 2D CNN on  $xy$ ,  $xz$ , and  $yz$  planes of a microscopy volume and combine the results using a majority voting technique in [167]. These pseudo-3D methods using 2D CNNs still cannot fully utilizing the entire 3D information. A fully 3D U-Net, an extension of a 2D U-Net [125], is introduced in [168] to analyze *Xenopus* kidney visualized by confocal microscopy. Fully 3D operations such as 3D convolutional layers, 3D max-pooling layers, and 3D transposed convolutional layers are used in the architecture. The 3D U-Net in [168] is trained by a set of sparsely labeled 2D images. A 3D CNN which can both segment and detect nuclei in microscopy images is developed [169]. The 3D CNN in [169] is composed of a 3D U-Net [168] for generating a segmentation mask followed by a series of convolutional layers for generating detection points. The 3D CNN in [169] is trained by real ground truth volumes which would be extremely tedious to manually generate them. A 3D cell instance segmentation method is described in [170] where bounding boxes on all cells and segmentation masks on some cells are required to train the network. As mentioned above, data augmentation

techniques are used to multiply ground truth images [85]. Data augmentation uses a simple linear and nonlinear transformation to produce synthetic training images from real ground truth images. Although it has shown a remarkable improvement especially to avoid overfitting problems, data augmentation techniques still require some real ground truth images. Our goal is to train 3D CNNs without using any real ground truth volumes. In the next subsection, we will describe generative adversarial networks which generates synthetic training images.

### 2.2.3 Generative Adversarial Networks

Generative adversarial networks, also known as GANs, have been used to produce synthetic images [171,172]. GANs are widely used in biomedical research community to synthesize images to train CNNs [173,174] because creating labeled training images from biomedical images is expensive and tedious. For example, synthetic cells in fluorescence microscopy [175], synthetic histopathology images [176,177], synthetic red blood cell images [178], synthetic lesions [179], and synthetic CT images [180] can be generated using GANs.

GANs are composed of two networks, a generative network,  $G$ , and a discriminative network,  $D$ . The goal of the generative network is to produce realistic synthetic images whereas the goal of the discriminative network is to discriminate synthetic images from real images. More specifically,  $G$  maps from an array of noise,  $z$ , to an image,  $x$ , whereas  $D$  determines whether  $x$  is real or not. We want to train  $D$  to maximize the probability of labeling  $x$  as a real image. We also want to train  $G$  to minimize the probability of labeling  $G(z)$ , a synthetic image from noise, as a synthetic image. Therefore,  $G$  and  $D$  are trained according to the following equation with a value function  $V(D, G)$  [171]:

$$\min_G \max_D V(D, G) = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))] \quad (2.13)$$

As both  $G$  and  $D$  are trained simultaneously, the generative network,  $G$ , can produce more realistic synthetic images.

GANs are further developed to translate labels to synthetic images. A conditional GAN, known as pix2pix, is presented in [181]. One drawback of pix2pix is that it requires real ground truth images to train GAN. To produce synthetic images without using any real ground truth images, a cycle-consistent GAN, known as CycleGAN, is developed [182]. A cycle consistency loss is included in the training loss function of the CycleGAN which is defined as [39, 182]:

$$\begin{aligned} L(G_1, G_2, D_1, D_2) = & L_{GAN}(G_1, D_1, I^{label}, I^{orig}) + L_{GAN}(G_2, D_2, I^{orig}, I^{label}) \\ & + \lambda L_{cycle}(G_1, G_2, I^{orig}, I^{label}) \end{aligned} \quad (2.14)$$

where  $\lambda$  is a weight coefficient and  $\|\cdot\|_1$  is  $L_1$  norm. The three loss terms in Equation 2.14 are defined as:

$$L_{GAN}(G_1, D_1, I^{label}, I^{orig}) = E_{I^{orig}}[\log(D_1(I^{orig}))] + E_{I^{label}}[\log(1 - D_1(G_1(I^{label})))] \quad (2.15)$$

$$L_{GAN}(G_2, D_2, I^{orig}, I^{label}) = E_{I^{label}}[\log(D_2(I^{label}))] + E_{I^{orig}}[\log(1 - D_2(G_2(I^{orig})))] \quad (2.16)$$

$$\begin{aligned} L_{cycle}(G_1, G_2, I^{orig}, I^{label}) = & E_{I^{label}}[\|G_2(G_1(I^{label})) - I^{label}\|_1] \\ & + E_{I^{orig}}[\|G_1(G_2(I^{orig})) - I^{orig}\|_1] \end{aligned} \quad (2.17)$$

In Equation 2.14,  $G_1$  is a generative network generating  $I^{orig}$  using  $I^{label}$ ,  $G_2$  is a generative network generating  $I^{label}$  using  $I^{orig}$ ,  $D_1$  is a discriminative network distinguishing between  $I^{orig}$  and  $G_1(I^{label})$ , and  $D_2$  is a discriminative network distinguishing between  $I^{label}$  and  $G_2(I^{orig})$ .

We developed a spatially constrained cycle consistent GAN (SpCycleGAN) by including a spatial constraint term in the training loss function to generate synthetic nuclei at spatially accurate locations [39]. Note our SpCycleGAN does not require any real ground truth images. The training loss function of the SpCycleGAN is defined as [39]:

$$\begin{aligned} L(G_1, G_2, H, D_1, D_2) = & L_{GAN}(G_1, D_1, I^{label}, I^{orig}) + L_{GAN}(G_2, D_2, I^{orig}, I^{label}) \\ & + \lambda_1 L_{cycle}(G_1, G_2, I^{orig}, I^{label}) + \lambda_2 L_{sp}(G_1, H, I^{orig}, I^{label}) \end{aligned} \quad (2.18)$$

where  $\lambda_1$  and  $\lambda_2$  are weight coefficients and  $\|\cdot\|_2$  is  $L_2$  norm. The first three terms are defined as Equation 2.15, Equation 2.16, Equation 2.17 and the last term is defined as:

$$L_{sp}(G_1, H, I^{orig}, I^{label}) = E_{I^{label}}[\|H(G_1(I^{label})) - I^{label}\|_2] \quad (2.19)$$

In Equation 2.18,  $H$  is a generative network generating a binary mask,  $H(G_1(I^{label}))$ , from  $G_1(I^{label})$ . By minimizing a  $L_2$  loss between  $H(G_1(I^{label}))$  and  $I^{label}$ , synthetic nuclei spatially match to the nuclei label. Using the synthetic microscopy images generated from the SpCycleGAN, our segmentation CNN has shown an outstanding performance [39]. The SpCycleGAN is used to generate synthetic microscopy images in Chapter 6 and Chapter 7.

### 3. BOUNDARY SEGMENTATION USING STEERABLE FILTERS

Boundary segmentation is an important step in analyzing subcellular structures in fluorescence microscopy images. For example, boundary segmentation can help characterizing a nephron in kidney or vascular space in liver. Segmentation of fluorescence microscopy is challenging due to blur and inhomogeneous intensity presented in fluorescence microscopy volumes. Boundary segmentation is especially challenging because other subcellular structures may present in fluorescence microscopy images. It is necessary to distinguish and separate other structures from the boundaries we desire to segment. We observe boundary structures generally have a thin shape. Using steerable filters [37], strong directional tendencies are captured at voxels on the boundary structures. In this chapter, we segment and separate boundaries using 3D adaptive histogram equalization, 3D steerable filters in various orientations, connected component analysis, and  $z$ -propagation refinement for data sets with tubular structures.

#### 3.1 Proposed Method

Figure 3.1 is a block diagram of the proposed method. 3D adaptive histogram equalization is first used on the original 3D stack of images for enhancement purposes. The histogram equalized images are then segmented by a foreground/background technique described below. Our goal is to segment tubule boundaries from lumen on foreground regions. We want to place “seeds” as growing points on tubule boundaries to separate tubule boundaries from lumen. Three-dimensional steerable filters are used to generate potential seeds that will be used to identify and grow tubule boundaries. This is followed by a small blob removal stage through which undesired

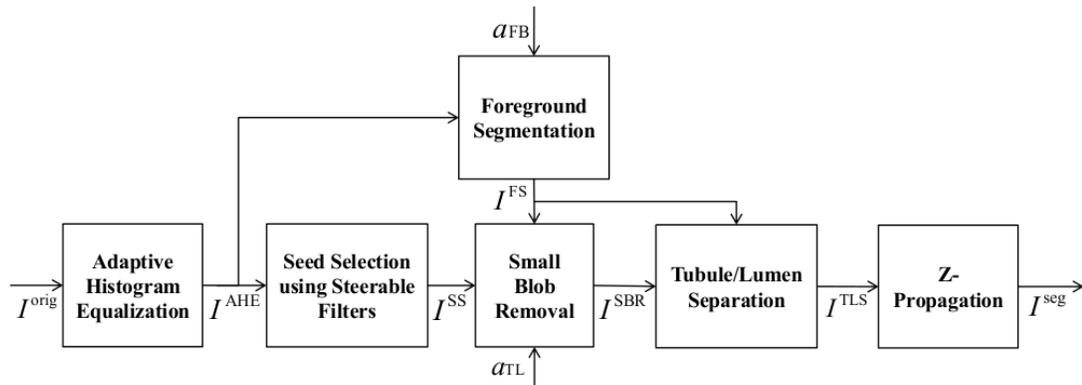


Fig. 3.1.: Block diagram of the proposed method

seeds are removed and the remaining seeds are used to separate tubule boundaries from lumen in images using two-dimensional connected components. Finally, the results for each image are correlated along the  $z$ -direction to remove any mis-labeled voxels.

### 3.1.1 Histogram Equalization and Foreground Segmentation

Since fluorescence microscopy images have non-uniform intensities, with low intensities being visible in the vicinity of the image boundaries, a pre-processing step is used to enhance regions close to the boundaries. This is achieved via 3D adaptive histogram equalization that employs a  $17 \times 17 \times 9$  rectangular window, where voxels near the stack boundaries are mirrored past the boundaries. Since the resolution along the  $z$ -direction is smaller than along the  $x$  and  $y$ -directions, the window dimensions were chosen accordingly. This step is used since it brightens only “foreground pixels” locally.

Foreground segmentation, in which the images are separated into “foreground” (**F**) and “background” (**B**) regions, is then done next. The foreground/background regions are determined by:

$$I^{FS}(\mathbf{v}) = \begin{cases} 255, & \text{if } I^{AHE}(\mathbf{v}) > 255 \cdot (1 - a_{FB}) \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

where  $I^{AHE}$  is the histogram equalized image volume,  $\mathbf{v} = (x, y, z)$  is a voxel location, and  $a_{FB}$  denotes the ratio of the number of voxels in the foreground to the total number of voxels. The value of  $a_{FB}$  is changed according to the desired size of foreground (**F**) and background (**B**) regions. For example, if the value of  $a_{FB}$  is increased, then the number of voxels belonging to foreground region may also increase.

### 3.1.2 Seed Selection Using Steerable Filters

In order to distinguish tubule boundaries within the foreground, “seeds” for constructing/growing each potential tubule boundary are needed. Since tubule boundaries tend to be thinner than lumen, seed selection is accomplished through the use of steerable filters [37] having various orientations that are capable of capturing the directional tendencies of thin regions. A property of steerable filters is that they can be synthesized as a linear combination of basis filters [37]. The advantage of steerable filters is that they can be used to detect local orientation of edges [183]. For example, facial wrinkles are detected using steerable filters in various orientations [184].

A 3D steerable filter with orientation angles  $\theta$  and  $\phi$  (see Figure 3.2),  $h^{\theta,\phi}$ , can be obtained as:

$$\begin{aligned} h^{\theta,\phi}(\mathbf{v}) = & \sin^2(\phi) \cos^2(\theta) g_{xx}(\mathbf{v}) + \sin^2(\phi) \sin^2(\theta) g_{yy}(\mathbf{v}) + \cos^2(\phi) g_{zz}(\mathbf{v}) \\ & - \sin^2(\phi) \sin(2\theta) g_{xy}(\mathbf{v}) - \sin(2\phi) \cos(\theta) g_{xz}(\mathbf{v}) - \sin(2\phi) \sin(\theta) g_{yz}(\mathbf{v}) \end{aligned} \quad (3.2)$$

where  $\mathbf{v} = (x, y, z)$  denotes voxel location,  $\theta$  an angle in the  $xy$ -plane relative to the  $x$ -axis,  $\phi$  an angle relative to the  $z$ -axis, and  $g(\mathbf{v})$  a Gaussian function given by  $g(x, y, z) = e^{-(x^2+y^2+z^2)}$  with rectangular support that is  $24 \times 24 \times 24$  in size. Allowing the values of  $\theta$  and  $\phi$  to vary between  $0^\circ$  to  $180^\circ$  in intervals of  $22.5^\circ$  results in 64 steerable filters.

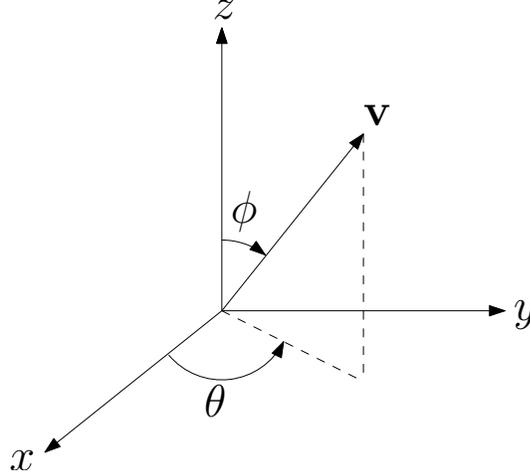


Fig. 3.2.: Spherical coordinates

Generally, tubules have thin boundaries and voxel intensities on tubule boundaries are higher than voxel intensities in the background. Therefore, a voxel intensity function will be locally concave along the normal to a tubule boundary. Consequently, the second derivative of the intensity function will typically be small near tubule boundaries. We utilize this fact and initially generate 64 steerable filter responses,  $R^{\theta,\phi}$ , which are responses with orientation  $\theta$  and  $\phi$ , respectively, of the 64 different steerable filters to the output of the adaptive histogram equalization step,  $I^{AHE}$ , that is:

$$R^{\theta,\phi} = I^{AHE} ** h^{\theta,\phi} \quad (3.3)$$

where  $**$  denotes 3D convolution.

Next we obtain the minimum responses:

$$I^{MR}(\mathbf{v}) = \min_{\theta,\phi} R^{\theta,\phi}(\mathbf{v}) \quad (3.4)$$

$$I^{\theta}(\mathbf{v}) = \arg \min_{\theta} R^{\theta,\phi}(\mathbf{v}) \quad (3.5)$$

$$I^{\phi}(\mathbf{v}) = \arg \min_{\phi} R^{\theta,\phi}(\mathbf{v}) \quad (3.6)$$

The smaller  $I^{MR}(\mathbf{v})$  is at voxel  $\mathbf{v}$  the stronger directional tendency that voxel will have. Similarly,  $I^{\theta}(\mathbf{v})$  and  $I^{\phi}(\mathbf{v})$  provide the strong directional tendencies at voxel  $\mathbf{v}$  along the  $\theta$  and  $\phi$  directions, respectively.

In order to decide how many voxels will be selected as seed candidates, we define the seed ratio,  $r_s$ , as the ratio of the number of seed candidates to the total number of voxels:

$$r_s = \frac{\sum_{x=1}^X \sum_{y=1}^Y \sum_{z=1}^Z I^{orig}(x, y, z)}{255XYZ} \quad (3.7)$$

where the size of the stack is  $X \times Y \times Z$ . The brighter the voxel intensities in  $I^{orig}$  are the more voxels are selected as seed candidates. For each focal plane image, we obtain the  $N_s = \lfloor r_s XY \rfloor$  smallest values of  $I_{z_p}^{MR}$  and choose these voxels to be seed candidates. Therefore, we have  $N_s$  seed candidates for each 2D image. The resulting image wherein seeds have been selected is denoted by  $I_{z_p}^{SS}$ .

### 3.1.3 Small Blob Removal and Tubule/Lumen Separation

In the prior seed selection stage, seed candidates were selected, but not all candidates lie on tubule boundaries. Thus, it is necessary to remove seed candidates not belonging to tubule boundaries. Initially, all seeds not belonging to foreground sections in  $I^{FS}$  are removed. In the previous step, we created 3D stacks,  $I^\theta(\mathbf{v})$  and  $I^\phi(\mathbf{v})$ , which contain the orientation of strong directional tendencies at  $\mathbf{v}$ . For each orientation pair  $(\theta, \phi)$  on every 2D image,  $I_{z_p}^{SS}$ , seed candidates having the orientation  $(\theta, \phi)$  and that reside in a connected region of size (size here denotes the number of connected components) less than a blobsize threshold,  $t_b$ , are removed using 2D connected component analysis where the threshold is given by

$$t_b = a_{TL} r_s XYZ \quad (3.8)$$

Note  $a_{TL}$  is the ratio of the number of seed candidates in the largest blob in a lumen to the total number of seed candidates in the image. After removing small blobs in each orientation, the remaining seeds are selected to be final candidates denoted as  $I^{SBR}$ .

Subsequent to small blob removal, tubule/lumen separation is performed based on the location of the final seeds in  $I^{SBR}$ . Using 2D connected component analysis all

foreground (**F**) blobs on every focal plane image,  $I_{z_p}^{FS}$ , that contain seeds are labeled as tubule boundaries (**T**), whereas the remaining voxels in the foreground are labeled as lumen (**L**). The value of  $a_{TL}$  can be tuned according to the desired size of tubule boundaries (**T**) and lumen (**L**) regions. For example, if  $a_{TL}$  is increased, then  $t_b$  will be increased and the number of final seed candidates will be reduced, which may result in smaller tubule regions in the foreground region.

### 3.1.4 $z$ -Propagation Refinement

We assume that sudden changes in the biological structure are unlikely to occur between adjacent slices along the  $z$ -direction and hence adjacent voxels along the  $z$ -direction will typically have the same label or belong to the same tubule boundary. Since some adjacent voxels along the  $z$ -direction may not have the same label, an extra post-processing step is necessary. We choose a set of voxels along the  $z$ -direction that have the same  $(x, y)$  position. Recall that each voxel is labeled as **T**, **L**, or **B**, representing tubule boundary, lumen, or background respectively. Next we divide this set into disjoint subsets such that each subset contains only voxels labeled as **T** or **L** (see Figure 3.3). For each such subset, a majority vote decides whether **T** or **L** is assigned to all pixels of that subset.

Figure 3.3 depicts a set of labeled voxels consisting of voxels belonging to a specific  $(x, y)$  position but differing  $z$ -planes. This set is divided into multiple subsets separated by the voxels labeled as **B**. In subset-1, most of the voxels are labeled as **L**, therefore all the voxels from subset-1 will be labeled as **L**, based on majority voting. In contrast, in subset-2 and subset-3, all voxels are labeled as **T**, based on respective majority votes.

## 3.2 Experimental Results

We tested our method on Data-T1. For this particular data  $a_{FB}$  was empirically found to be 0.2138 and  $a_{TL}$  empirically determined to be  $5.86 \times 10^{-7}$ . Figure 3.4

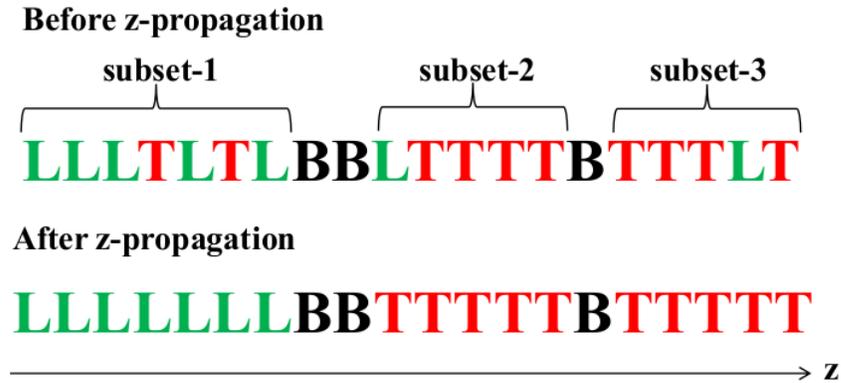


Fig. 3.3.: Example of  $z$ -propagation refinement

shows the images obtained at each step for  $I_{z_{95}}$  in Data-T1. In addition, Figure 3.5 shows the results produced by the proposed method for several images located at various depths. Figure 3.6 shows the 3D visualization of a subvolume of Data-T1 using Voxx [185].

We compare the performance of the proposed method to several other segmentation methods used in microscopy images including active contours [54], Jacob and Unser’s ImageJ plugin known as steerableJ [183], region competition [65], and Squash [67, 68], as shown in Figure 3.7.

It was observed that in the case of active contours, which required the manual selection of initial curves for each tubule boundary, most of the initial curves evolved poorly although they were initialized very closely to the actual boundaries. This was observed to be the case since the tubule boundaries in the Data-T1 were not well defined. As can be observed SteerableJ detects not only tubule boundaries but also lumens although it failed to detect tubule boundaries located at image boundaries where the pixel intensities were low. As far as region competition was concerned it had no ability to distinguish between tubule boundary and lumen. Similarly Squash also failed to distinguish between tubule boundary and lumen and it only segmented foreground regions located in the central area of the image.

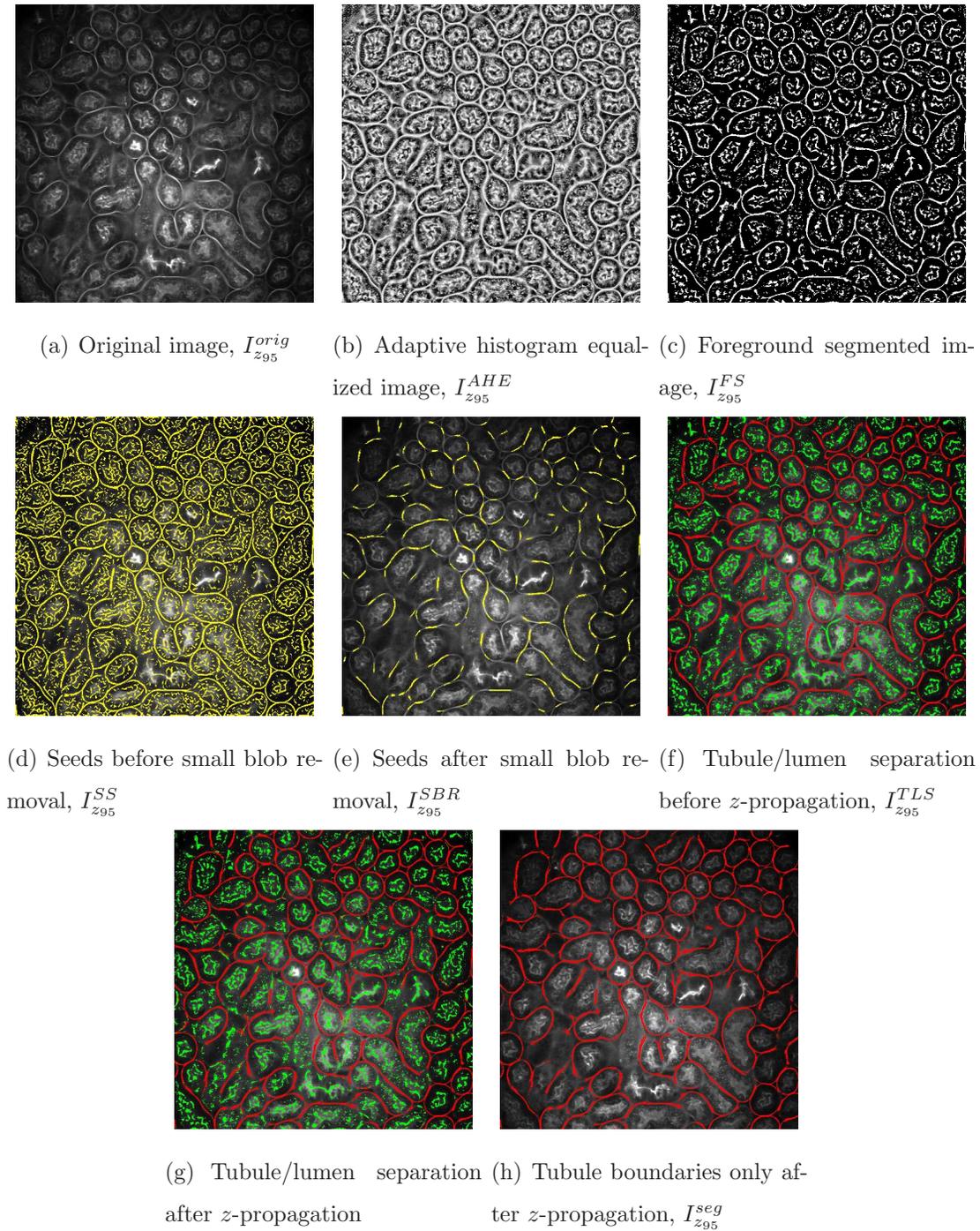


Fig. 3.4.: Original, intermediate, and segmented images of Data-T1

All methods were also evaluated using reference data that was generated by manually segmenting the images. The metrics used in the evaluation were segmentation

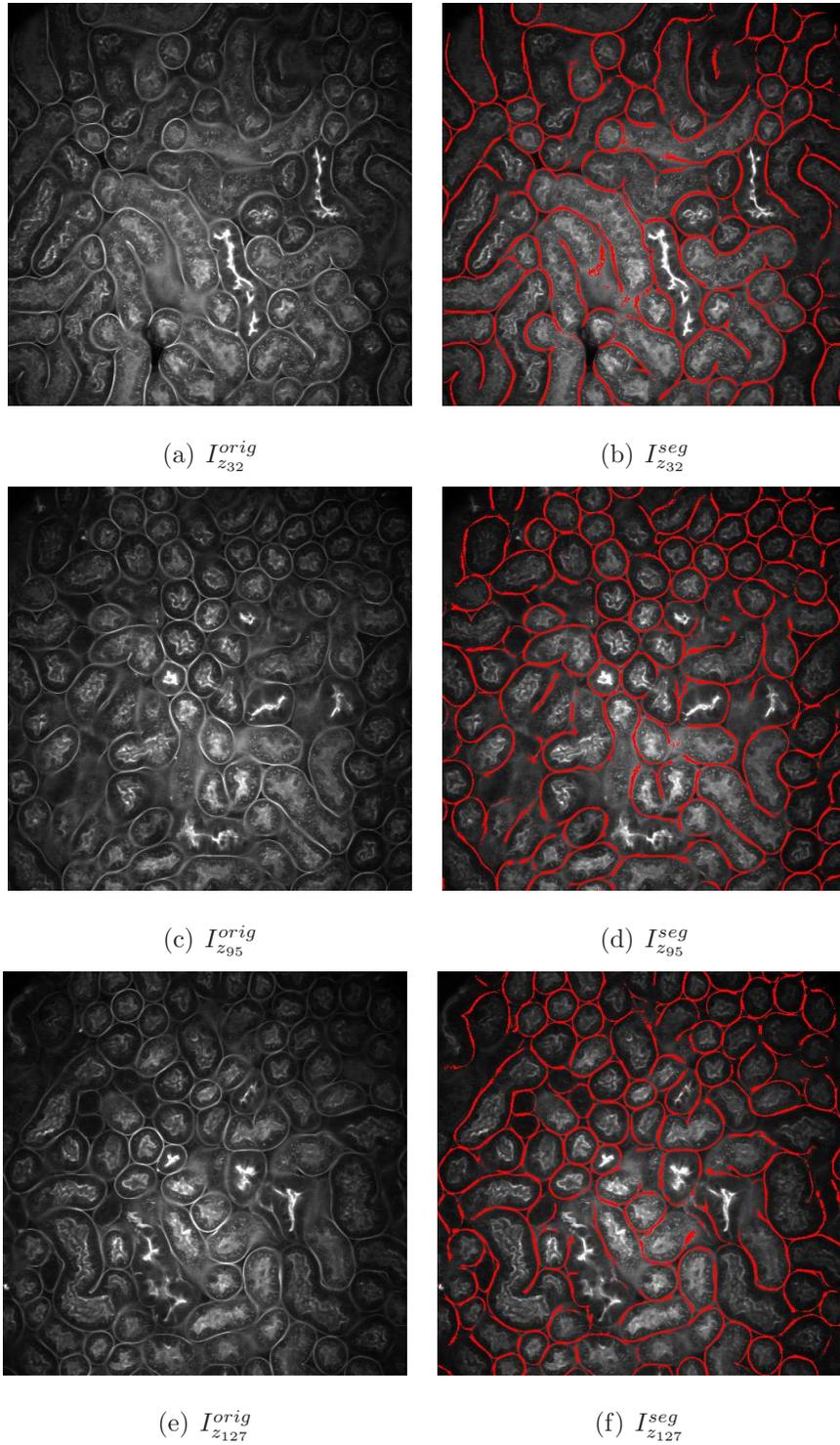


Fig. 3.5.: Original and segmented images of Data-T1

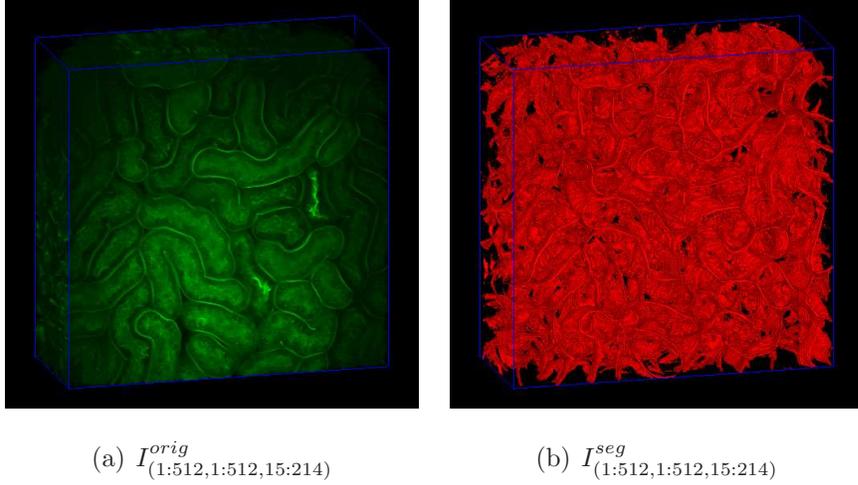


Fig. 3.6.: 3D visualization of Data-T1

accuracy and the Type-I and Type-II error ratios. Accuracy, Type-I error, and Type-II error ratios were defined as

$$\text{accuracy} = \frac{n_{\text{TP}} + n_{\text{TN}}}{n_{\text{total}}} \quad (3.9)$$

$$\text{Type-I error} = \frac{n_{\text{FP}}}{n_{\text{total}}} \quad (3.10)$$

$$\text{Type-II error} = \frac{n_{\text{FN}}}{n_{\text{total}}} \quad (3.11)$$

where  $n_{\text{TP}}$ ,  $n_{\text{TN}}$ ,  $n_{\text{FP}}$ ,  $n_{\text{FN}}$ ,  $n_{\text{total}}$  denote the number of true-positives (pixels on tubule boundary that are correctly labeled as belonging to a tubule boundary), true-negatives (lumen/background pixels that are correctly labeled as either lumen/background), false-positives (number of pixels belonging to lumen/background that are wrongly labeled as belonging to tubule boundary), false-negatives (number of pixels on tubule boundaries that are wrongly labeled as either lumen/background), and the total number of pixels, respectively [186, 187]. As can be observed from Table 3.1 our method outperformed the other techniques. It is to be noted that the parameters of the aforementioned techniques were tuned to produce the best results.

Our method was also tested on three liver datasets: Data-T2, Data-T3, and Data-T4. Figure 3.8, Figure 3.9, and Figure 3.10 show segmentation results in various

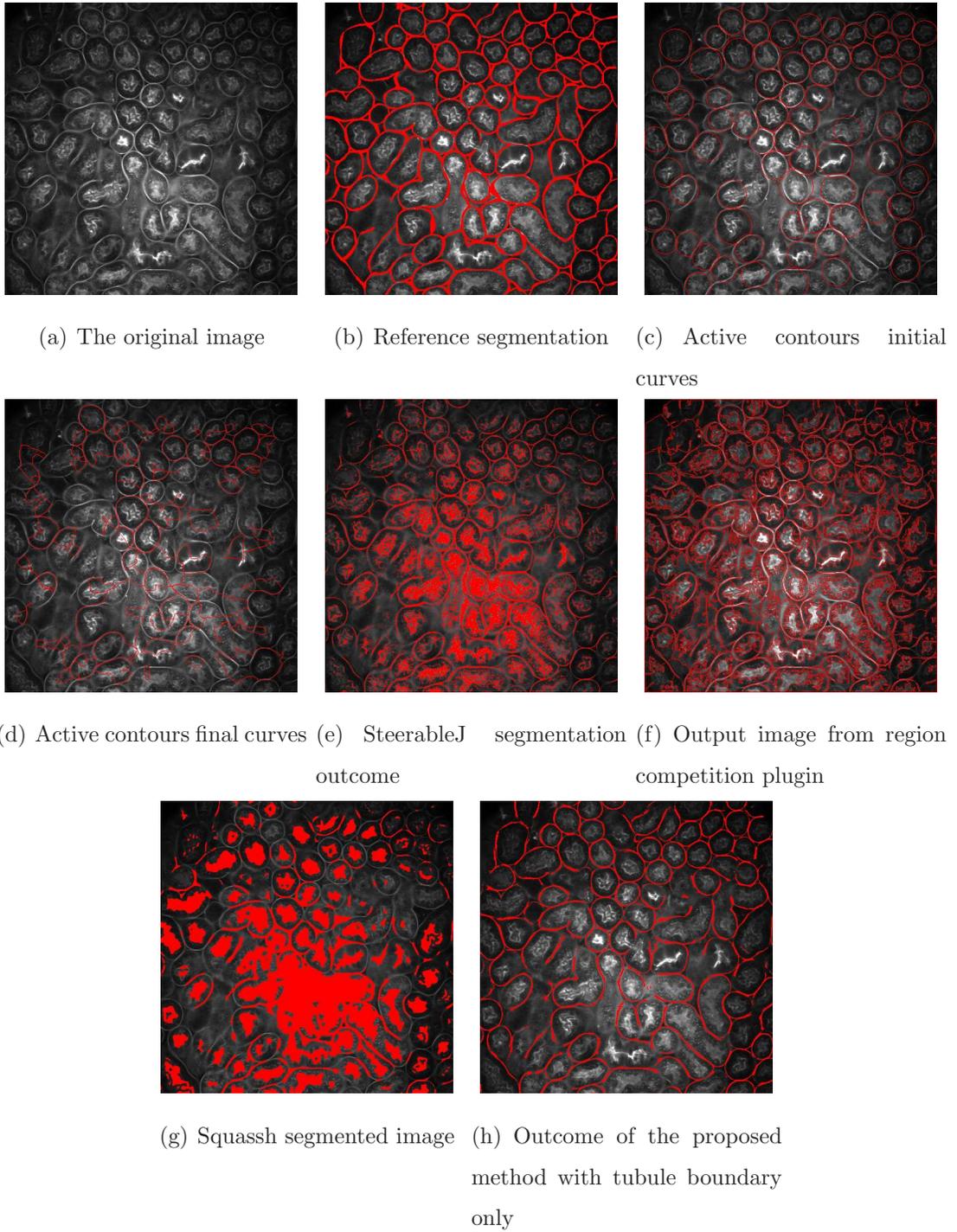


Fig. 3.7.: Comparison of other segmentation methods and our proposed method of  $I_{295}$  in Data-T1

Table 3.1.: Accuracy, Type-I and Type-II errors for various methods on  $I_{z_{95}}$  of the Data-T1

	Accuracy	Type I error	Type II error
Active Contours [54]	85.56%	1.83%	12.61%
steerableJ [183]	78.40%	12.16%	9.44%
Region Competition [65]	76.58%	12.99%	10.43%
Squassh [67, 68]	68.45%	22.33%	9.22%
Proposed Method	90.86%	1.63%	7.51%

Table 3.2.: Accuracy, Type-I and Type-II errors for various methods on  $I_{z_{17}}$  of the Data-T2

	Accuracy	Type-I error	Type-II error
Active Contours [54]	78.76%	0.56%	20.68%
steerableJ [183]	82.44%	4.18%	13.38%
Region Competition [65]	69.50%	11.64%	18.86%
Squassh [67, 68]	85.61%	10.09%	4.30%
Proposed Method	86.35%	6.62%	7.03%

depth of Data-T2, Data-T3, and Data-T4, respectively. As shown in Figure 3.11, Figure 3.12, Figure 3.13 and observed from Table 3.2, Table 3.3, and Table 3.4, our method successfully identified tubule and cell boundaries in various focal planes. In this case the values utilized for the parameters  $a_{FB}$  and  $a_{TL}$  were  $a_{FB} = 0.27$  and  $a_{TL} = 1.00 \times 10^{-5}$  for Data-T2,  $a_{FB} = 0.27$  and  $a_{TL} = 1.50 \times 10^{-3}$  for Data-T3, and  $a_{FB} = 0.20$  and  $a_{TL} = 6.41 \times 10^{-7}$  for Data-T4, respectively.

Table 3.3.: Accuracy, Type-I and Type-II errors for various methods on  $I_{z_1}$  of the Data-T3

	Accuracy	Type-I error	Type-II error
Active Contours [54]	82.04%	0.66%	17.30%
steerableJ [183]	84.68%	3.52%	11.80%
Region Competition [65]	72.92%	11.08%	16.00%
Squassh [67, 68]	85.83%	11.41%	2.76%
Proposed Method	88.05%	5.84%	6.11%

Table 3.4.: Accuracy, Type-I and Type-II errors for various methods on  $I_{z_{81}}$  of the Data-T4

	Accuracy	Type-I error	Type-II error
steerableJ [183]	87.38%	4.18%	8.44%
Region Competition [65]	85.42%	4.16%	10.41%
Squassh [67, 68]	90.07%	1.33%	8.60%
Proposed Method	88.27%	3.18%	8.55%

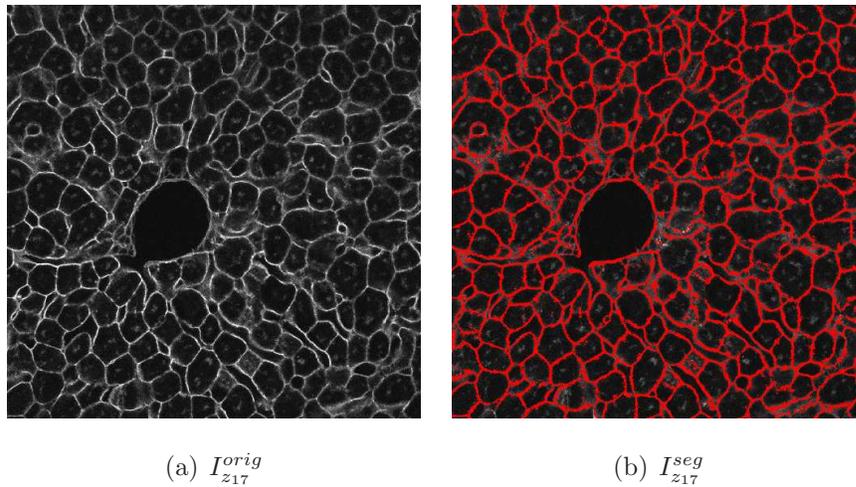
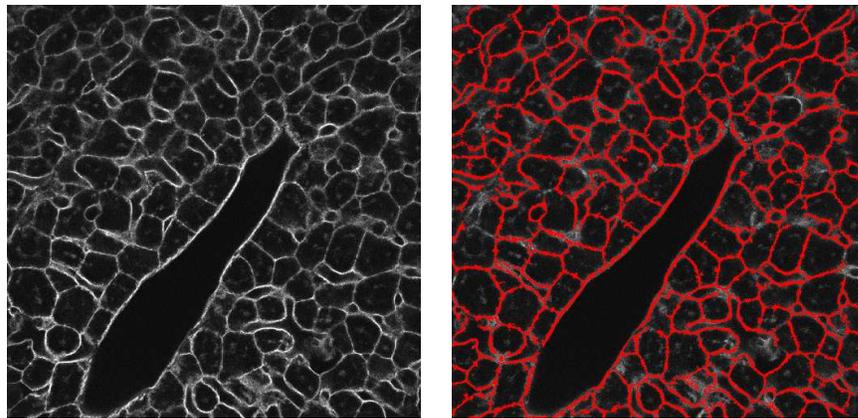


Fig. 3.8.: Original and segmented images of Data-T2



(a)  $I_{z_8}^{orig}$

(b)  $I_{z_8}^{seg}$

Fig. 3.9.: Original and segmented images of Data-T3

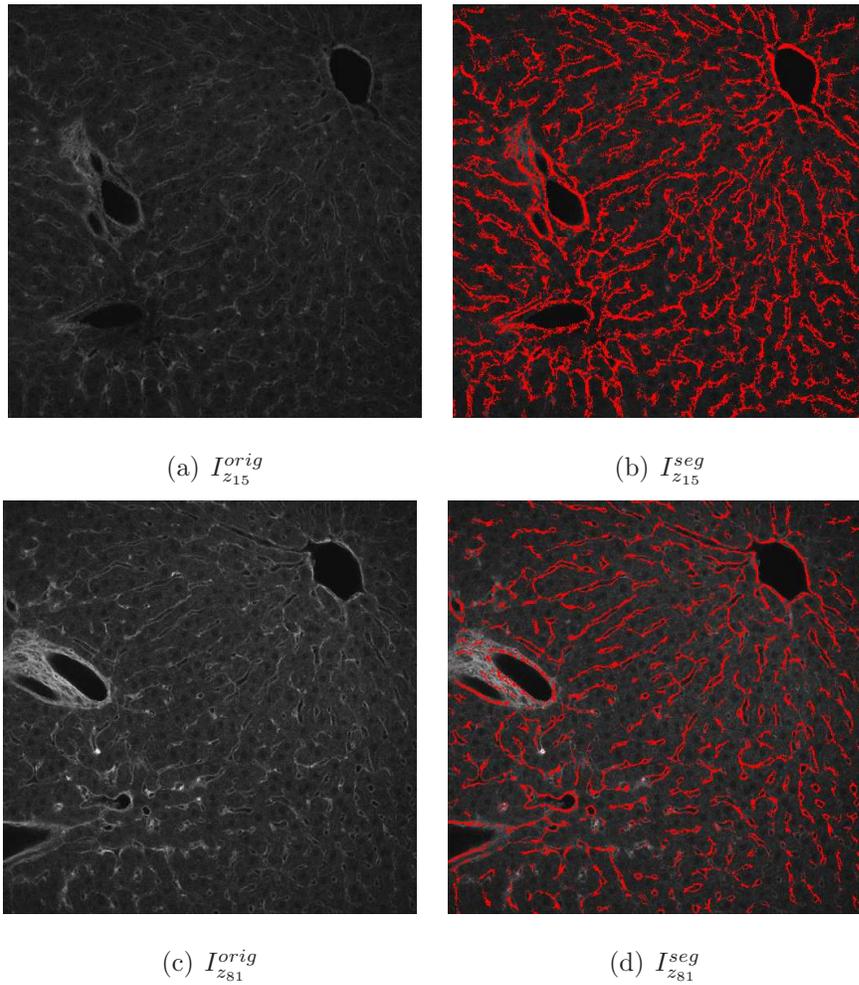


Fig. 3.10.: Original and segmented images of Data-T4

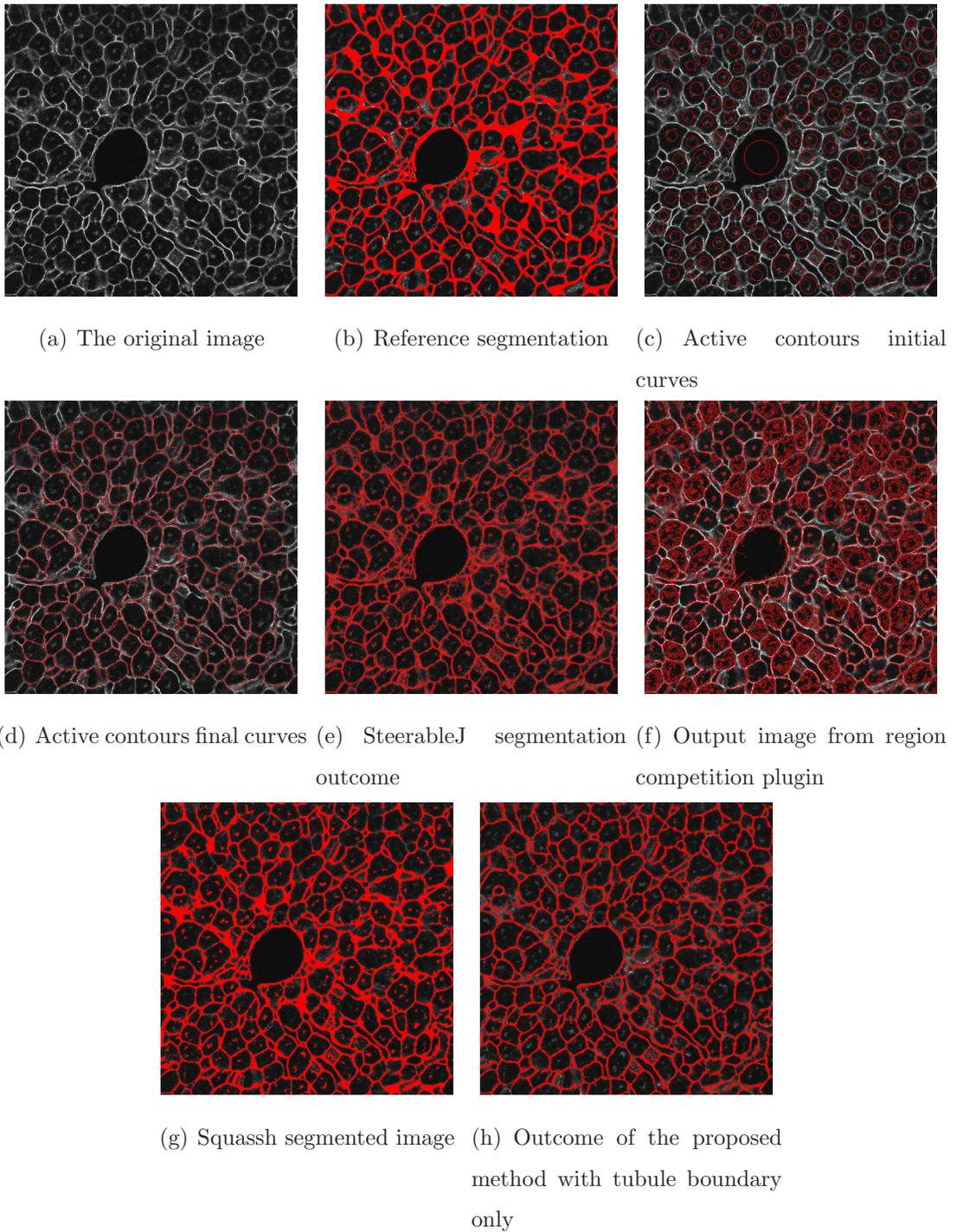


Fig. 3.11.: Comparison of other segmentation methods and our proposed method of  $I_{217}$  in Data-T2

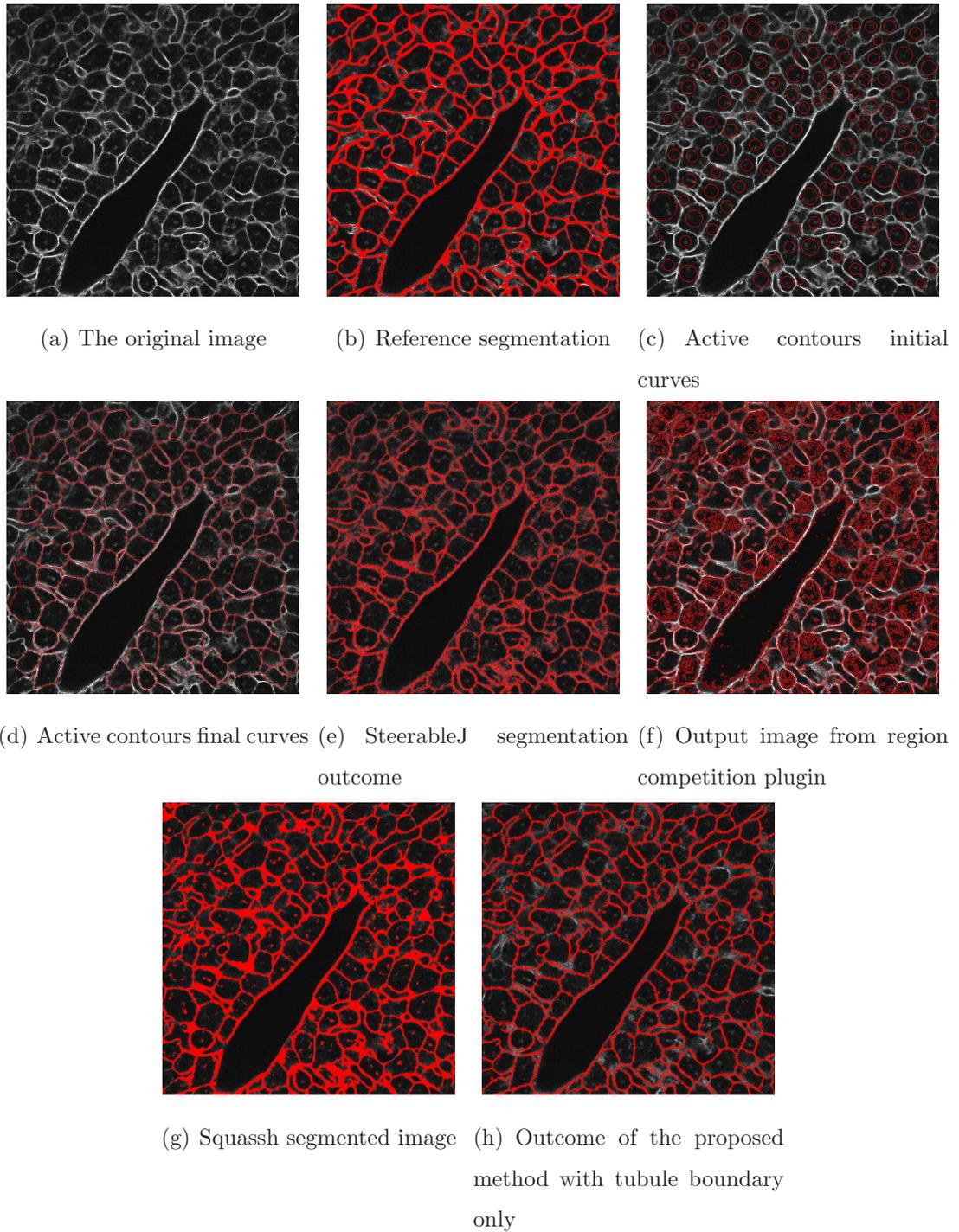


Fig. 3.12.: Comparison of other segmentation methods and our proposed method of  $I_{z_1}$  in Data-T3

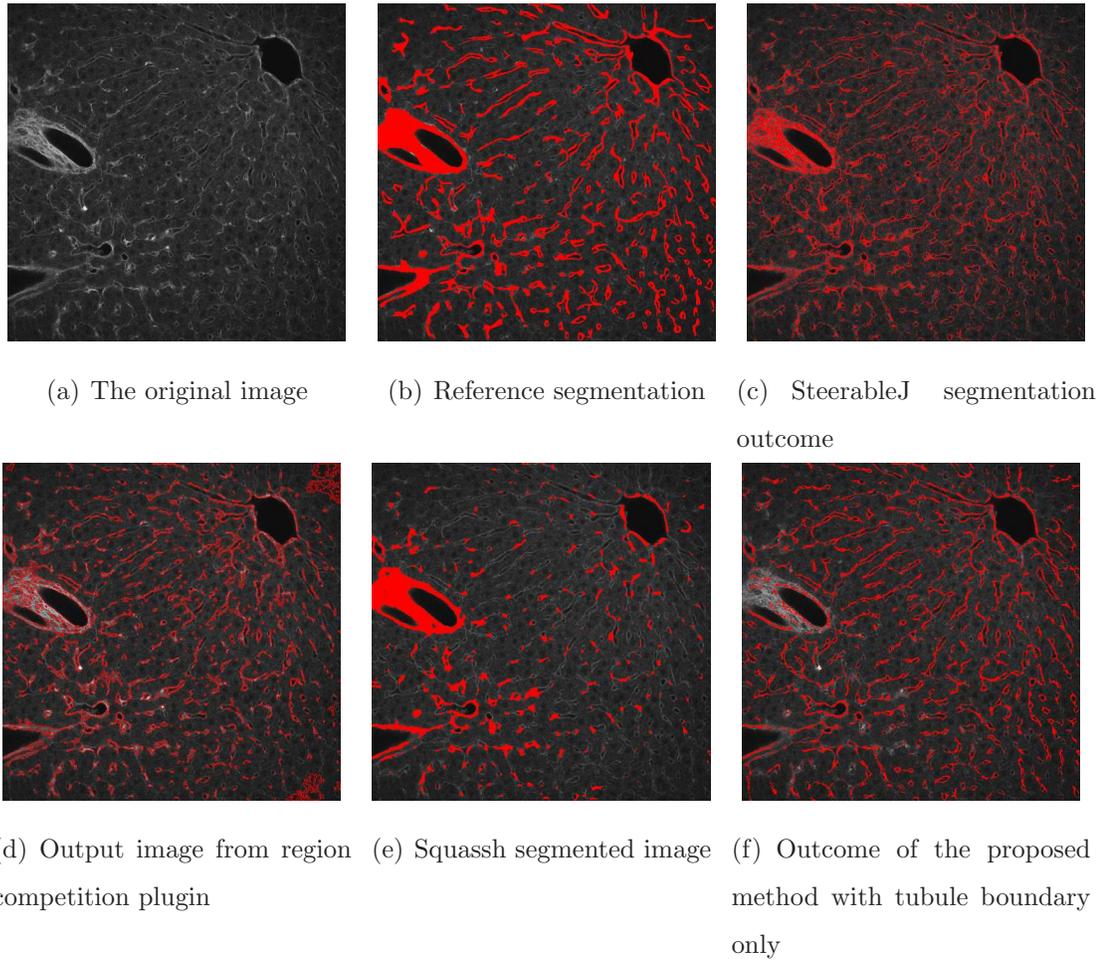


Fig. 3.13.: Comparison of other segmentation methods and our proposed method of  $I_{z_{81}}$  in Data-T4

## 4. NUCLEI SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORKS

In this chapter, we describe a nuclei segmentation method using convolutional neural networks (CNNs). To utilize the information in depth for more accurate segmentation, we develop a “fully” three dimensional CNN. To train the CNN, a large set of labeled training microscopy volume/images is required, the availability of which is extremely limited in biomedical data sets [139]. It is extremely tedious for one to manually generate labeled training images due to the complexity of data. Data augmentation approaches [85] are used to increase the amount of labeled training images. This has shown promising results to train CNNs using a limited number of labeled training images in biomedical applications [125, 167]. Data augmentation approaches still require a small number of actual ground truth images. Our goal is to train our CNNs without using any actual ground truth images. We automatically generate synthetic microscopy images to train our segmentation CNNs. We assume nuclei are in an ellipsoidal shape and we add blur and noise operations to produce realistic synthetic microscopy images.

### 4.1 Proposed Method

Figure 4.1 shows a block diagram of our proposed method to segment nuclei in three dimension. Synthetic volumes containing nuclei,  $I^{syn}$ , with their corresponding labeled volumes (synthetic ground truth volumes),  $I^{label}$ , are first randomly generated as described below. The synthetic volumes and labeled volumes then used to train a 3D CNN,  $M$ . Finally, the 3D CNN is used to segment nuclei in fluorescence microscopy images volumes,  $I^{orig}$ . The segmented image volume is denoted by  $I^{seg}$ .

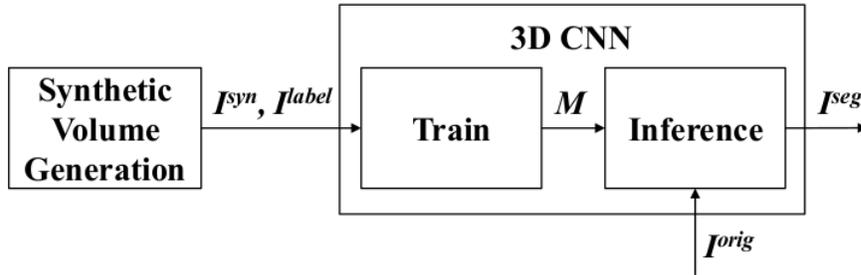


Fig. 4.1.: Block diagram of the proposed method

#### 4.1.1 Synthetic Volume Generation

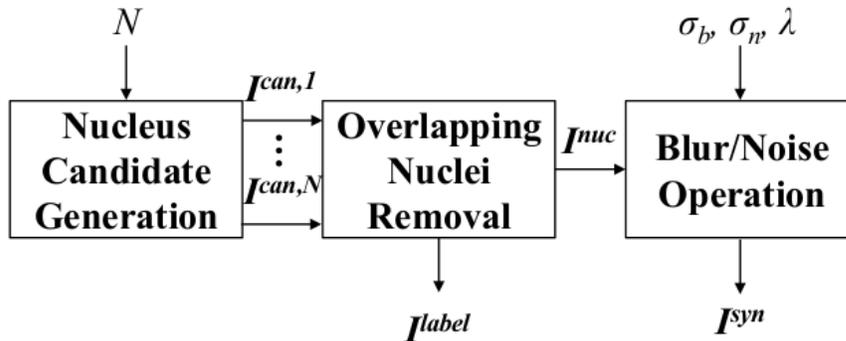


Fig. 4.2.: Block diagram for Synthetic Volume Generation

Figure 4.2 is a block diagram of the Synthetic Volume Generation stage. As we noted earlier, one of the major challenges in segmentation of biomedical images/volumes using a CNN is the lack of training images/volumes [139]. Manual segmentation to generate ground truth, especially in three dimension, is inefficient and laborious. Our approach is to use synthetic image volumes to train the CNN.

In synthetic image volumes, voxels belonging to nuclei which we are interested in segmenting are defined as foreground and other voxels are defined as background. We assume that nuclei can be modeled as an ellipsoid. Let  $N$  be the number of nucleus candidates. The  $j^{\text{th}}$  nucleus candidate,  $I^{can,j}$ , is first generated where  $1 \leq$

$j \leq N$ , with randomly generated lengths of the semi-axes, translation, rotation, and intensity of the ellipsoid. Then for each  $j$ ,  $I^{can,j}$  is included in the nuclei synthetic image volume,  $I^{nuc}$ .  $I^{nuc}$  is initialized to a volume with all voxel intensities to zero, unless it overlaps with other nucleus candidates. Note that  $I^{nuc}$  is composed of non-overlapping nuclei with randomly assigned nonzero intensities and voxel intensities in background are zero. From  $I^{nuc}$ , a labeled volume,  $I^{label}$ , is generated as a binary volume where foreground voxels (non-overlapping nuclei) are 1 and background voxels are 0. Fluorescence microscopy volumes are blurred and noisy due to the point spread function of the microscope and noise from the light detector [17, 18, 27]. We observe that the background in fluorescence microscopy volumes are not completely dark. To make our synthetic image volume resemble a fluorescence microscopy volume, we first increase the voxel values in the background to a randomly generated non-zero value. The volume is then filtered using a Gaussian filter with standard deviation,  $\sigma_b$ , to blur it and then Gaussian noise with standard deviation,  $\sigma_n$ , and a Poisson noise with mean,  $\lambda$ , are added to generate the final synthetic image volume,  $I^{syn}$ .

The synthetic image volumes will be used to train our 3D CNN. If the volumes are too small, not enough information can be learned. If the volumes are too big, the training time will take long. Therefore, we select the size of the synthetic volumes to be  $64 \times 64 \times 64$ .

Note that the forward model is used to model the image acquisition process of fluorescence microscopy [188]:

$$g[n] \sim Q(\mathcal{P}((A\{f\})[n] + b) + w[n]) \quad (4.1)$$

where  $f$  is the actual signal from fluorophores,  $A$  is an operator for a PSF,  $b$  is a background signal,  $\mathcal{P}$  is a Poisson distribution produced by photon noise,  $w$  is readout noise modeled as a Gaussian distribution,  $Q$  is a quantization operator, and  $g$  is the output signal acquired by a fluorescence microscope. To accurately analyze a 3D PSF in widefield microscopy, a more sophisticated model such as the Gibson and Lanni model [189] is used in [190]. The work in [190] localizes particles in a 3D volume generated by a widefield microscope and the work in [188] accurately restores volumes

containing *C. elegans* embryo generated by a widefield microscope but they do not generate images containing nuclei. In this chapter, we model a PSF, background noise, photon noise, and readout noise to synthesize nuclei in fluorescence microscopy images for training our 3D CNN.

## Nucleus Candidate Generation

In order to generate a synthetic volume with multiple nuclei, we first generate a single nucleus which can be potentially included in the synthetic volume. We first assume that nuclei are ellipsoidal shape. To train our CNN with various synthetic nuclei, we generate nuclei which have random size, are located in random positions, are oriented in random directions, and have random intensity. In this step,  $N$  nucleus candidates are generated in a volume of size  $64 \times 64 \times 64$ .

To generate the  $j^{\text{th}}$  nucleus candidate, where  $1 \leq j \leq N$ , we first find the translated and rotation coordinates,  $(\tilde{x}, \tilde{y}, \tilde{z})$ , from the original coordinates,  $(x, y, z)$  with a random translation vector,  $\mathbf{t} = (t_x, t_y, t_z)$ , and a random rotation vector,  $\mathbf{r} = (r_x, r_y, r_z)$ .  $t_x, t_y, t_z, r_x, r_y, r_z$  are the translations in  $x, y$ , and  $z$ -direction and the rotations around  $x, y$ , and  $z$ -axes, respectively.

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = R_z(r_z)R_y(r_y)R_x(r_x) \begin{bmatrix} x - t_x \\ y - t_y \\ z - t_z \end{bmatrix} \quad (4.2)$$

$R_x(\theta), R_y(\theta), R_z(\theta)$  are rotation matrices around the  $x, y, z$ -axes, respectively, with an angle  $\theta$ .

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) \\ 0 & \sin(-\theta) & \cos(-\theta) \end{bmatrix} \quad (4.3)$$

$$R_y(\theta) = \begin{bmatrix} \cos(-\theta) & 0 & \sin(-\theta) \\ 0 & 1 & 0 \\ -\sin(-\theta) & 0 & \cos(-\theta) \end{bmatrix} \quad (4.4)$$

$$R_z(\theta) = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

The  $j^{\text{th}}$  nucleus candidate,  $I^{\text{can},j}$ , is then generated with random length of semi-axes,  $\mathbf{a} = (a_x, a_y, a_z)$  and random intensity,  $i$ :

$$I^{\text{can},j} = \begin{cases} i, & \text{if } \frac{\tilde{x}^2}{a_x^2} + \frac{\tilde{y}^2}{a_y^2} + \frac{\tilde{z}^2}{a_z^2} < 1 \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

We set  $N$  to be 100. For each  $j$ , with uniform distribution,  $\mathbf{a}$  is randomly selected between 4 and 6,  $\mathbf{t}$  is randomly selected between 1 and 64,  $\mathbf{r}$  is randomly selected between 1 and 360, and  $i$  is randomly selected between 200 and 255. Those parameters are set based on nuclei in our actual microscopy volumes.

## Overlapping Nuclei Removal

A synthetic volume with multiple nuclei,  $I^{\text{nuc}}$ , can now be generated by adding  $N$  nuclei candidate volumes,  $I^{\text{can},j}$ , generated in the previous step. However, in a biological structure, no nuclei are physically overlapping. So it is necessary to remove nuclei overlapping with other nuclei.

First,  $I^{\text{nuc}}$  is initialized to zeros with the size of  $64 \times 64 \times 64$ . Note that  $I^{\text{nuc}}$  is initialized to be all background (non-nuclei region). For  $1 \leq j \leq N$ , in a sequential order, the  $j^{\text{th}}$  single nucleus candidate,  $I^{\text{can},j}$ , would be added in the synthetic volume,  $I^{\text{nuc}}$ , if there is no intersection between foreground region in  $I^{\text{can},j}$  and foreground region in  $I^{\text{nuc}}$ . However,  $I^{\text{can},j}$  would not be added in  $I^{\text{nuc}}$  if  $I^{\text{can},j}$  have intersection between foreground region in  $I^{\text{can},j}$  and foreground region in  $I^{\text{nuc}}$ . After this step, no nuclei will overlap to other nuclei in  $I^{\text{nuc}}$ .

A labeled volume (a synthetic ground truth volume),  $I^{label}$ , corresponding to the synthetic volume is generated by assigning 1 to foreground voxels and 0 to background voxels:

$$I^{label}(x, y, z) = \begin{cases} 1, & \text{if } I^{nuc}(x, y, z) \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.7)$$

## Blur and Noise

When images of specimens are acquired from fluorescence microscope, images are degraded by blurring and noise. First, the point spread function (PSF) of the microscope causes blurring [17]. Additionally, fluorescence microscopy images contain a combination of Gaussian noise and Poisson noise because only a limited number of photons are detected in the detector of microscope due to photobleaching, low fluorophore concentration, and short exposure time [18,27]. Therefore, it is necessary to include blur and noise in our generate synthetic image volumes.

Initially, background voxel intensities of  $I^{nuc}$  are set to  $b$  because voxel intensities of the background region for real fluorescence microscopy images are not completely zero. We let  $b$  be randomly selected between 50 and 100 with an uniform distribution. We then use a simple synthetic PSF to blur the volumes where we assume the PSF is a normalized Gaussian filter with window size of  $5 \times 5 \times 5$  and standard deviation of  $\sigma_n = 20$ . Lastly, zero-mean Gaussian noise with  $\sigma_n = 5$  and Poisson noise with  $\lambda = 5$  are added to the blurred volume to generate  $I^{syn}$ . Figure 4.3 shows three examples of synthetic image volumes with their corresponding labeled volumes. Figure 4.4 compares two original image volumes at various depths and a synthetic image volume with the size of  $64 \times 64 \times 64$ . The size of the nuclei, the intensity of the nuclei, and the background noise of synthetic image volumes are close to original image volumes.

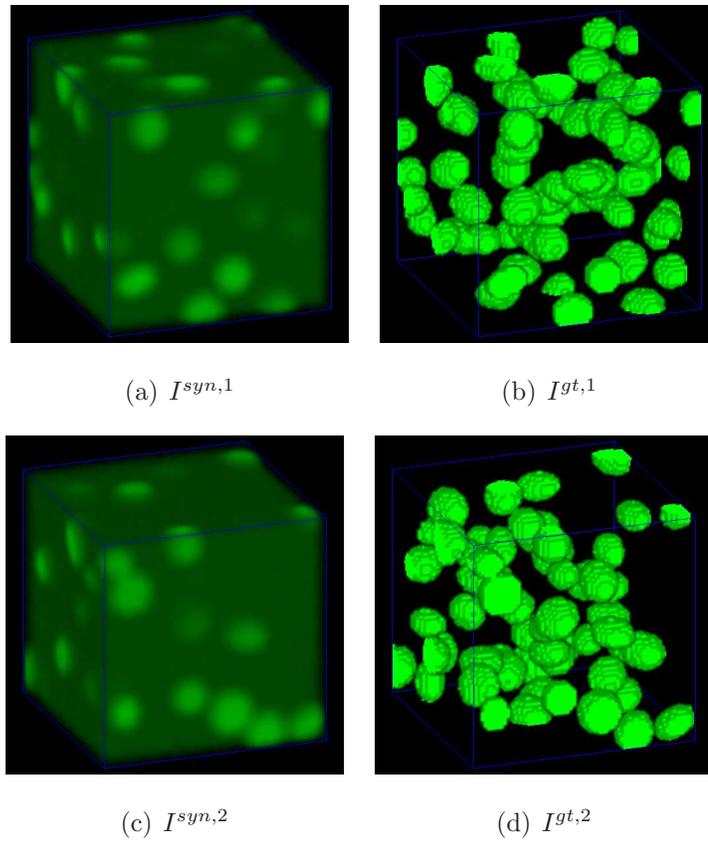


Fig. 4.3.: Examples of synthetic image volumes and their labeled volumes

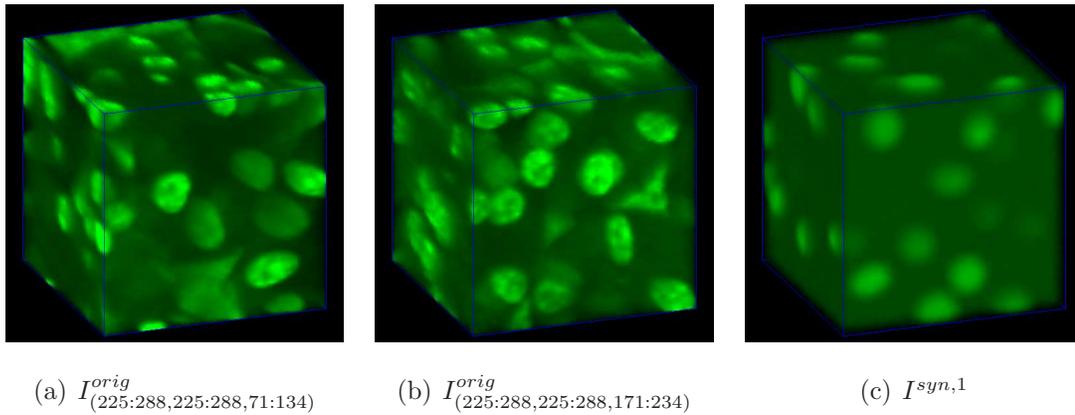


Fig. 4.4.: Comparison between original volumes and a synthetic volume

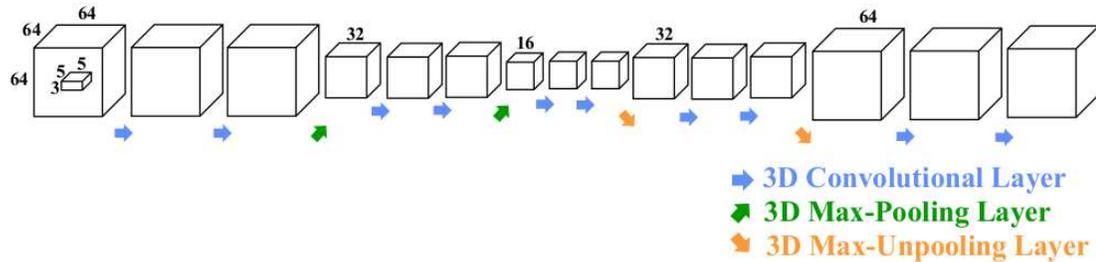


Fig. 4.5.: Architecture of our 3D CNN

#### 4.1.2 3D CNN Training

Figure 4.5 shows our architecture of a 3D CNN with an encoder-decoder structure. Each 3D convolutional layer consists of a convolutional operation with a  $5 \times 5 \times 3$  kernel with  $2 \times 2 \times 1$  voxel padding followed by 3D batch normalization [191] and a rectified-linear unit (ReLU) activation function. The kernel size is chosen accordingly because the resolution along the  $z$ -direction is smaller than along the  $x$  and  $y$ -directions. Voxel padding is used to maintain the same volume size during the convolutional operation. A 3D max-pooling layer uses  $2 \times 2 \times 2$  window with stride of 2 to preserve feature information while proceeding to the deep of the architecture. In the decoder, a 3D max-unpooling layer is used to retrieve feature information according to the indices that saved in the corresponding 3D max-pooling layer. An input volume to the network is a single channel volume with size of  $64 \times 64 \times 64$  and a 3D voxelwise classification with size of  $64 \times 64 \times 64$  is generated as an output volume of the network. Our architecture,  $M$ , is implemented in Torch [103]. To train our model,  $M$ , stochastic gradient descent (SGD) with a fixed learning rate of  $10^{-6}$  and a momentum of 0.9 is used. 100 pairs of synthetic image volumes,  $I^{syn}$ , and labeled image volumes (ground truth volumes for synthetic image volumes),  $I^{label}$ , are used to train the model. For each iteration, a randomly-selected training volume is used to train  $M$ . The training of our 3D CNN took approximately 2~3 days using NVIDIA’s GeForce GTX Titan X.

### 4.1.3 3D CNN Inference

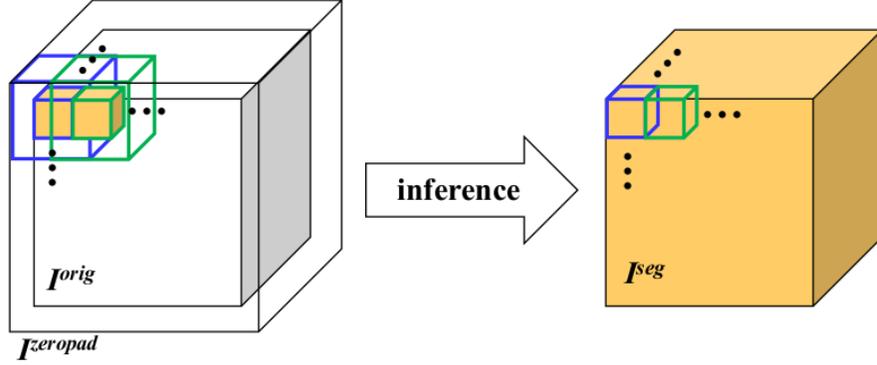


Fig. 4.6.: 3D CNN Inference

Our 3D CNN,  $M$ , segments a subvolume of size of  $64 \times 64 \times 64$ . While cropping the subvolume from the original volume, some nuclei on the boundary of the subvolume may be partially included and lose their shape information. This may lead to incomplete segmentation near the boundary of the subvolume. To avoid this, we use the central subvolume of the output volume with size of  $32 \times 32 \times 32$  to make sure nuclei in the output volume are completely segmented using their entire information. Additionally, if  $X$ ,  $Y$ , or  $Z$  of  $I^{orig}$  is greater than 64, it is necessary to slide a 3D window with size of  $64 \times 64 \times 64$  to segment the entire  $I^{orig}$ .

Figure 4.6 describes our method for 3D CNN inference. In order to have  $I^{seg}$  the same size as  $I^{orig}$ , we zero-pad  $I^{orig}$  by 16 voxels on boundaries, denoted as  $I^{zeropad}$ . In this case, the size of  $I^{zeropad}$  would be  $(X + 32) \times (Y + 32) \times (Z + 32)$ . Placing a 3D window on the top, left, frontal corner of  $I^{zeropad}$  (see blue window in Figure 4.6),  $I_{(1:64,1:64,1:64)}^{zeropad}$  becomes the input volume of the 3D CNN,  $M$ , with size of  $64 \times 64 \times 64$  to generate  $I_{(1:32,1:32,1:32)}^{seg}$ , a subvolume of  $I^{seg}$  on the top, left, frontal corner with size of  $32 \times 32 \times 32$ . Next, the 3D window is slid to  $x$ -direction by 32 voxels (see green window in Figure 4.6), then  $I_{(33:96,1:64,1:64)}^{zeropad}$  becomes the next input volume of the 3D CNN,  $M$ , to generate  $I_{(33:64,1:32,1:32)}^{seg}$ , a subvolume slid to  $x$ -direction by 32 voxels

from the previous subvolume,  $I_{(1:32,1:32,1:32)}^{seg}$ . This operation continues to  $x$ ,  $y$ , and  $z$ -direction until the entire volume is processed. The inference of a volume with size of  $512 \times 512 \times 512$  took approximately 2~3 minutes using NVIDIA’s GeForce GTX Titan X.

## 4.2 Experimental Results

We tested our method on four different rat kidney data sets: Data-N1, Data-N2, Data-N3, and Data-N4. Figure 4.7 shows the segmented images on Data-N1 located at various depths.

Our method was compared to other segmentation methods used in microscopy images including 3D active surface (3Dac) [59], 3D active surface with inhomogeneity correction (3DacIC) [34], 3D Squassh [67, 68], and 2D<sup>+</sup> convolutional neural network (2D<sup>+</sup> CNN) [167]. In order to evaluate our method, we used three 3D ground truth subvolumes of Data-N1,  $I_{(241:272,241:272,31:62)}^{gt}$ ,  $I_{(241:272,241:272,131:162)}^{gt}$ ,  $I_{(241:272,241:272,231:262)}^{gt}$  in different depth with size of  $32 \times 32 \times 32$ . The ground truth volumes for evaluation are manually generated from a real microscopy data set. Figure 4.8, Figure 4.9, Figure 4.10 are the 3D visualization of various segmentation method results on different subvolumes, helped by Voxx [185].

It was observed that 3D active surfaces had poor results as subvolumes are acquired deeper into tissue because voxel intensities on nuclei gets dimmer and more blurred. This issue was resolved by counting inhomogeneity correction, yet it has no ability to distinguish between nuclei and other subcellular structures. Squassh also failed to distinguish nuclei and other structures. Although 2D<sup>+</sup> CNN produced good results, discontinuity may be observed between planes because it has not utilized all three dimensional information. However, 3D CNN can segment in ellipsoidal shape which is close to the shape of nuclei. Note that the segmentation results are generated in a short running time (2~3 minutes) using NVIDIA’s GeForce GTX Titan X without any manually generated ground truth volumes to train the network.

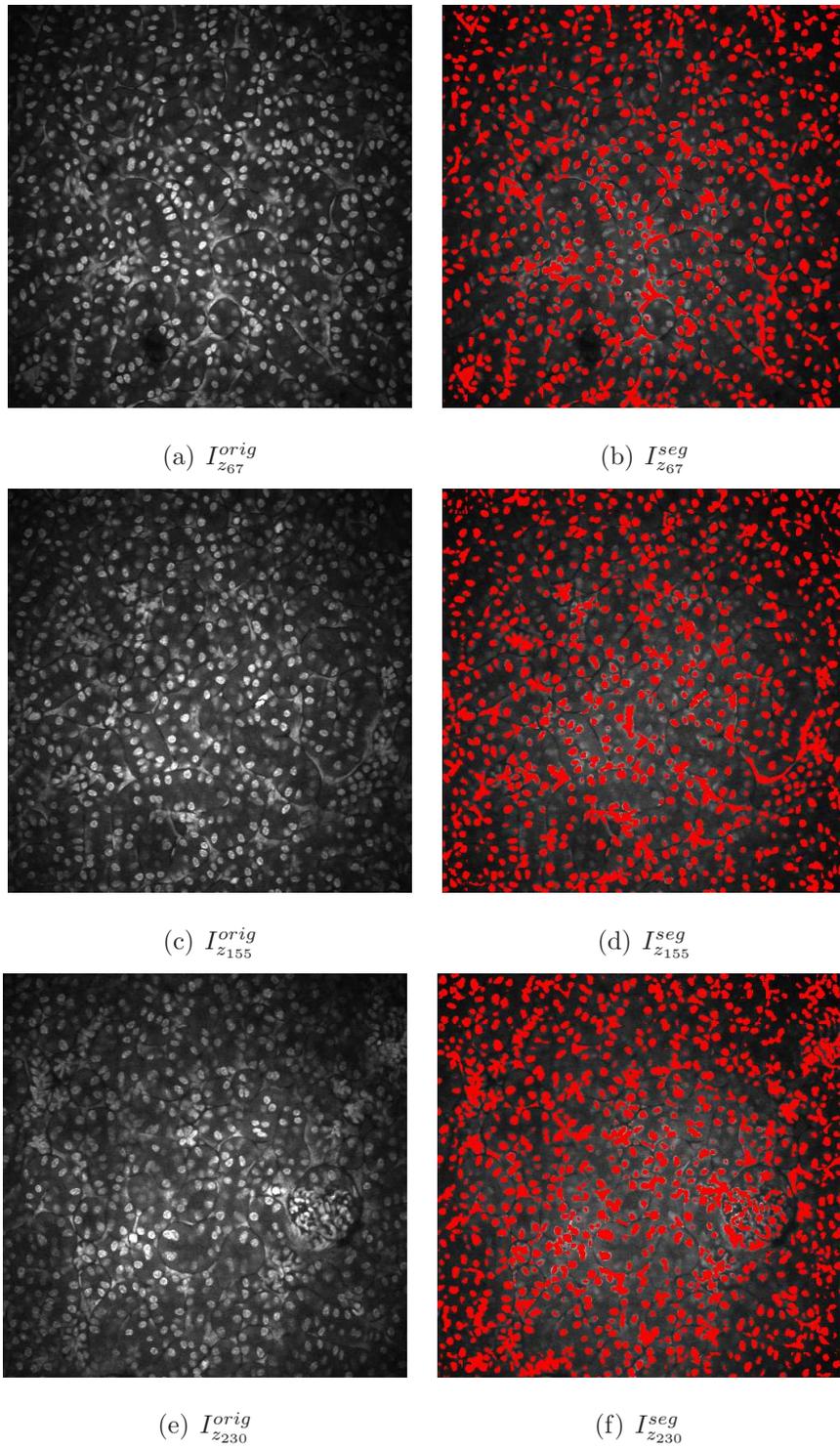


Fig. 4.7.: Original and segmented images of Data-N1

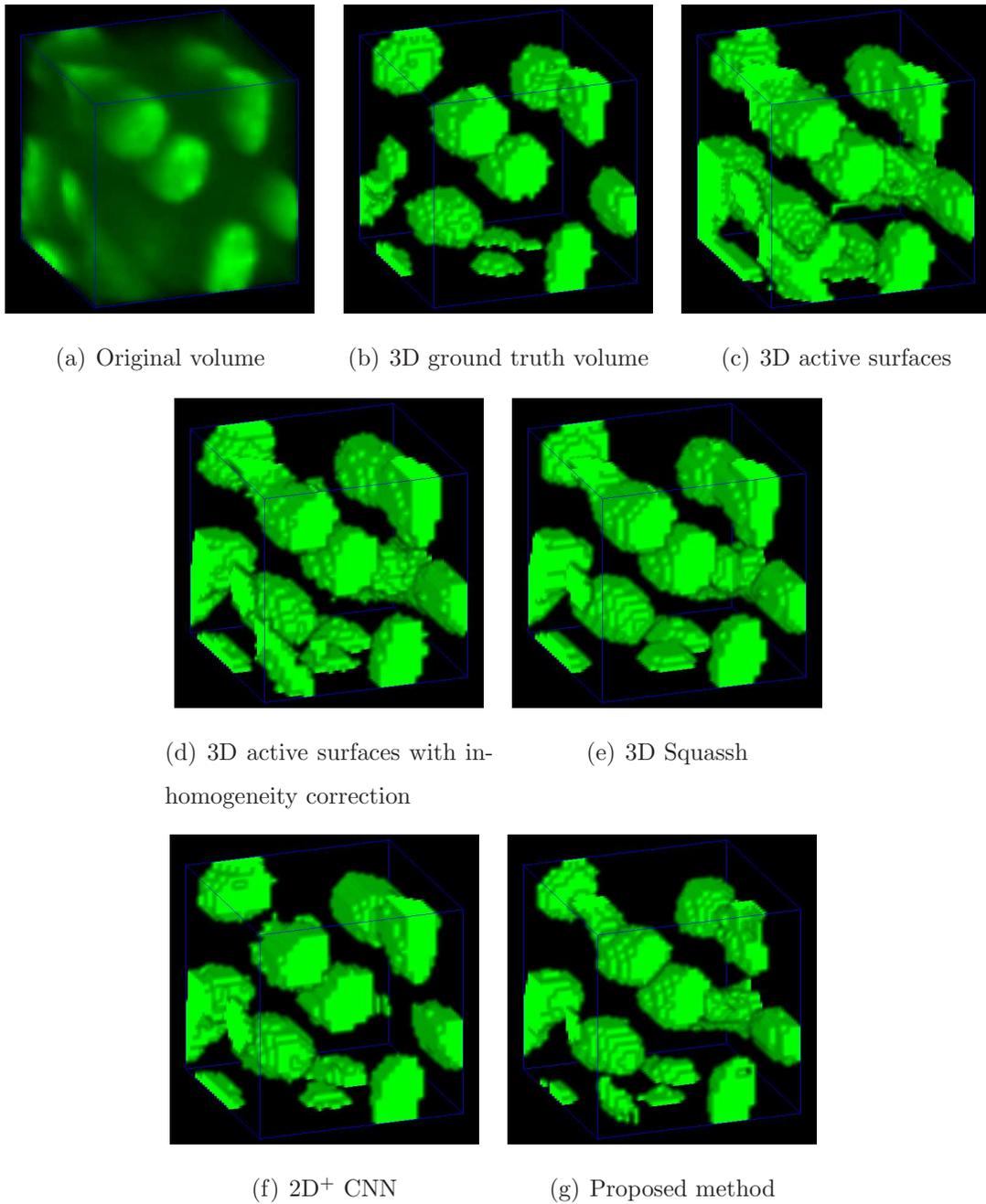


Fig. 4.8.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(241:272,241:272,31:62)}$  of Data-N1

All segmentation methods were evaluated using 3D ground truth volumes based on the accuracy, Type-I error, and Type-II error metrics. We define

$$\text{accuracy} = \frac{n_{\text{TP}} + n_{\text{TN}}}{n_{\text{total}}} \quad (4.8)$$

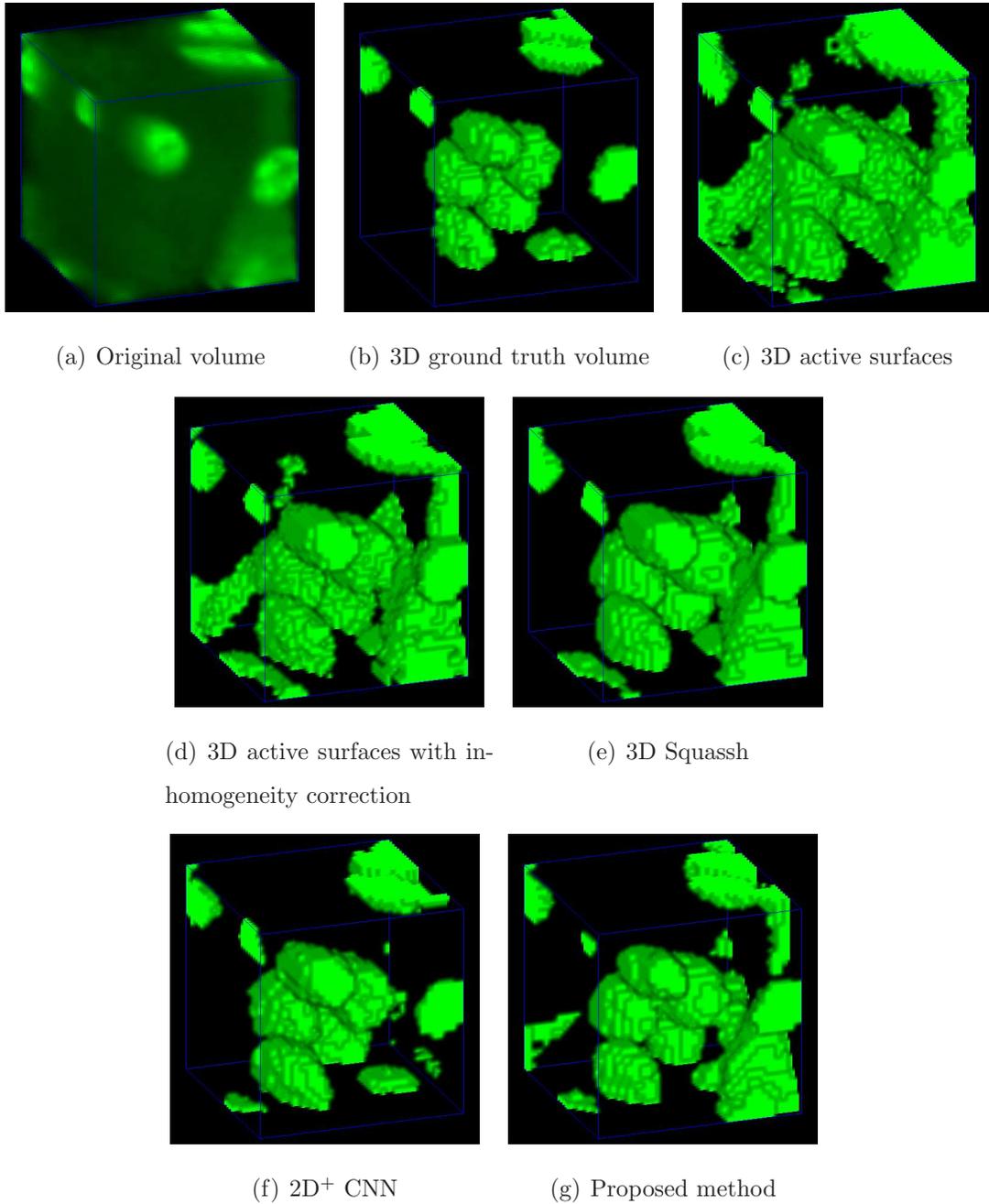


Fig. 4.9.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(241:272,241:272,131:162)}$  of Data-N1

$$\text{Type-I error} = \frac{n_{\text{FP}}}{n_{\text{total}}} \quad (4.9)$$

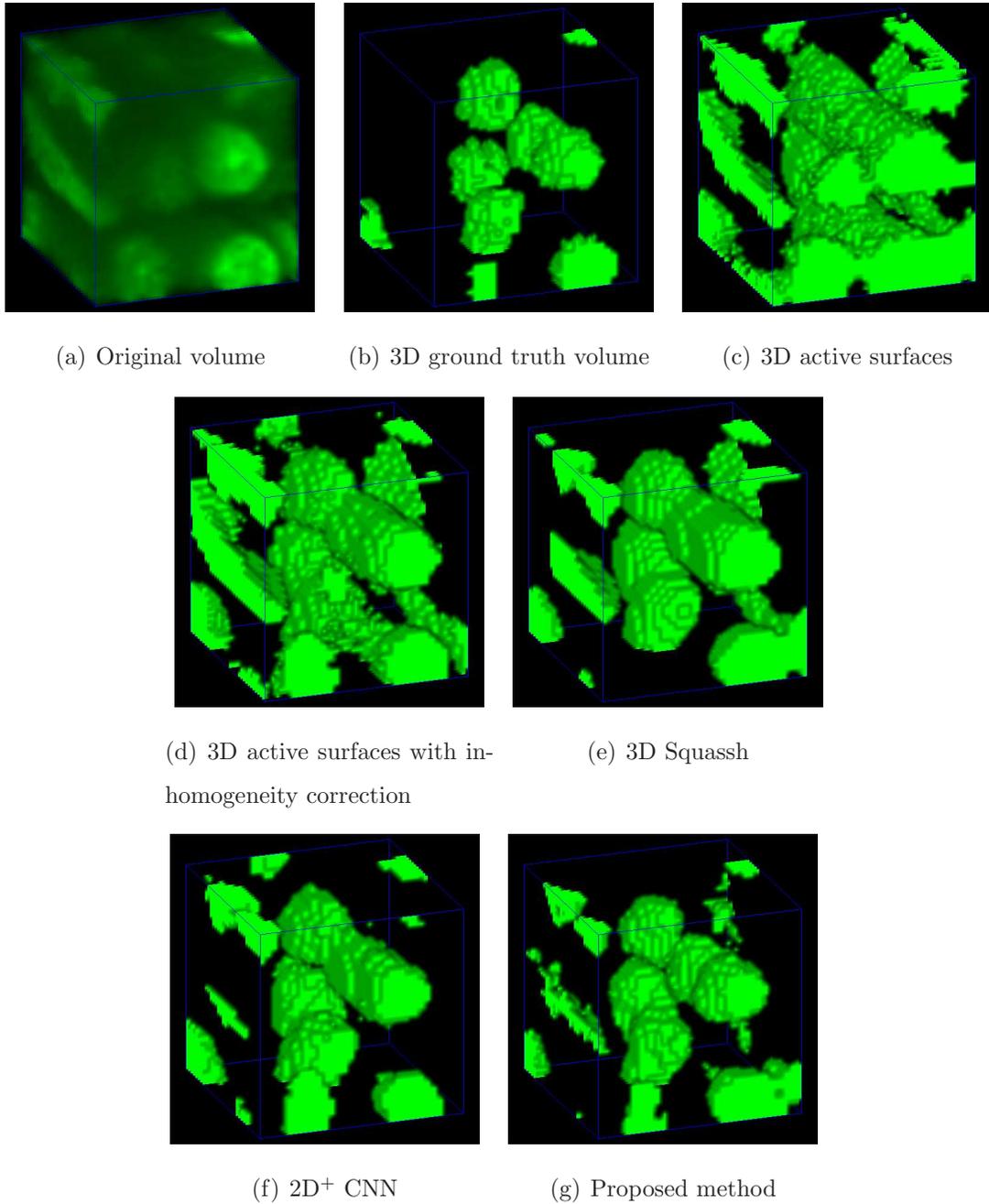


Fig. 4.10.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(241:272,241:272,231:262)}$  of Data-N1

$$\text{Type-II error} = \frac{n_{\text{FN}}}{n_{\text{total}}} \quad (4.10)$$

where  $n_{TP}$ ,  $n_{TN}$ ,  $n_{FP}$ ,  $n_{FN}$ ,  $n_{total}$  are defined to be the number of true-positives (voxels correctly segmented as nuclei), true-negatives (voxels correctly segmented as background), false-positives (voxels wrongly segmented as nuclei), false-negatives (voxels wrongly segmented as background), and the total number of voxels in a volume, respectively [186, 187]. Table 4.1, Table 4.2, Table 4.3 shows the accuracy for various segmentation methods and the proposed method on different subvolumes.

As shown in the 3D visualization from Figure 4.8, Figure 4.9, Figure 4.10, and accuracy test from Table 4.1, Table 4.2, Table 4.3, our 3D CNN achieved similar results “without” any ground truth volumes from Data-N1 during training. Training with synthetic volumes can be extremely helpful for automatic segmentation due to the difficulty of manually generating ground truth volumes in biomedical data sets.

Table 4.1.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(241:272,241:272,31:62)}$  of Data-N1

	Accuracy	Type-I error	Type-II error
3Dac [59]	84.09%	15.68%	0.23%
3DacIC [34]	87.36%	12.44%	0.20%
3D Squassh [67, 68]	90.14%	9.07%	0.79%
2D <sup>+</sup> CNN [167]	94.25%	5.18%	0.57%
Proposed Method	92.20%	5.38%	2.42%

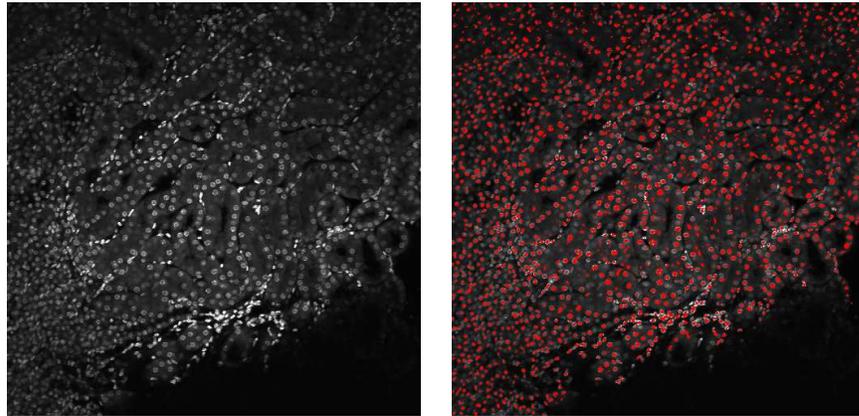
Our method can successfully segment nuclei from different rat kidney data (Data-N2, Data-N3, Data-N4). Since the size of nuclei in Data-N2, Data-N3, and Data-N4 is smaller than the size of nuclei in Data-N1, we used  $\mathbf{a}$ , length of semi-axes of a synthetic nucleus, randomly generated between 2 and 3. Figure 4.11, Figure 4.12, Figure 4.13 show the segmented images on different data sets.

Table 4.2.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(241:272,241:272,131:162)}$  of Data-N1

	Accuracy	Type-I error	Type-II error
3Dac [59]	79.25%	20.71%	0.04%
3DacIC [34]	86.78%	13.12%	0.10%
3D Squassh [67, 68]	88.26%	11.67%	0.07%
2D <sup>+</sup> CNN [167]	95.24%	4.18%	0.58%
Proposed Method	92.32%	6.81%	0.87%

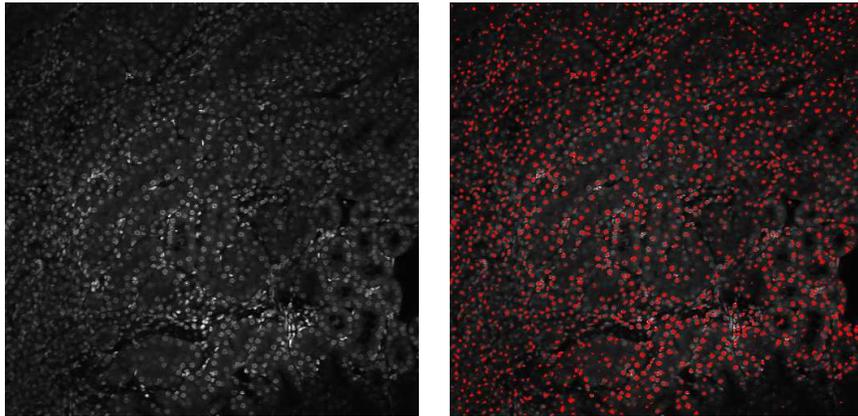
Table 4.3.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(241:272,241:272,231:262)}$  of Data-N1

	Accuracy	Type-I error	Type-II error
3Dac [59]	76.44%	23.55%	0.01%
3DacIC [34]	83.47%	16.53%	0.00%
3D Squassh [67, 68]	87.29%	12.61%	0.10%
2D <sup>+</sup> CNN [167]	93.21%	6.61%	0.18%
Proposed Method	94.26%	5.19%	0.55%



(a)  $I_{z_9}^{orig}$  of Data-N2

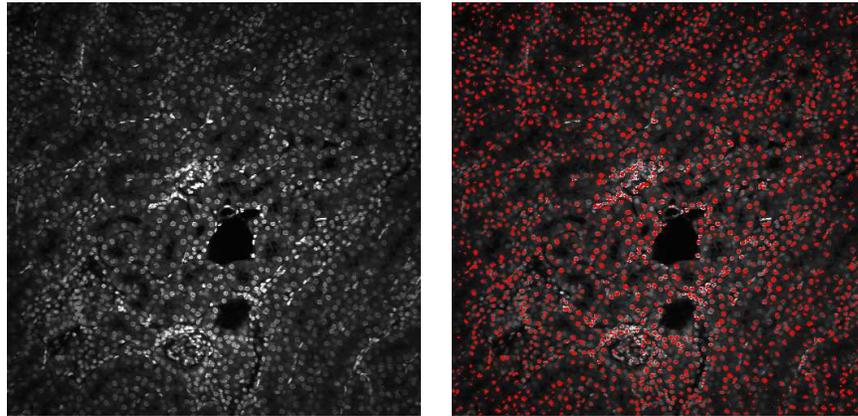
(b)  $I_{z_9}^{seg}$  of Data-N2



(c)  $I_{z_{17}}^{orig}$  of Data-N2

(d)  $I_{z_{17}}^{seg}$  of Data-N2

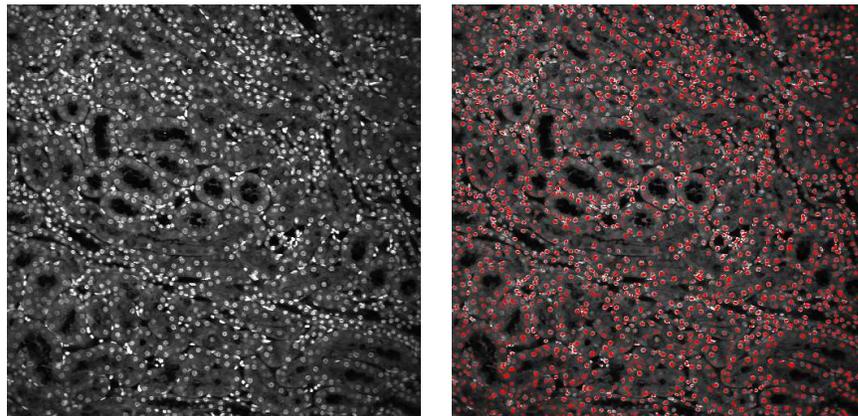
Fig. 4.11.: Original and segmented images of Data-N2



(a)  $I_{z_{17}}^{orig}$  of Data-N3

(b)  $I_{z_{17}}^{seg}$  of Data-N3

Fig. 4.12.: Original and segmented images of Data-N3



(a)  $I_{z_6}^{orig}$  of Data-N4

(b)  $I_{z_6}^{seg}$  of Data-N4

Fig. 4.13.: Original and segmented images of Data-N4

## 5. NUCLEI DETECTION AND SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORKS

In Chapter 4 we described a 3D CNN to segment nuclei in fluorescence microscopy volumes. We trained our CNN without using any real ground truth volumes. Nuclei were synthesized in ellipsoids in our synthetic microscopy training volumes so our CNN can successfully segment nuclei. We observed that our CNN still segment non-nuclei structures causing high Type-I error because non-nuclei structures were not synthesized in our synthetic microscopy training volumes. In this chapter we describe a nuclei detection and segmentation method. To reject non-nuclei structures presented in real microscopy volumes, we first detect the locations of nuclei by using a 3D adaptive histogram equalization [38], a 3D distance transform [44], and a 3D classification CNN. We denote the detected coordinates as seeds. We then segment each nucleus in subvolumes centered at the seeds using our 3D segmentation CNN which is similarly trained as [192]. Our evaluation shows that the method described in this chapter outperforms our previous method described in Chapter 4 by reducing Type-I error.

### 5.1 Proposed Method

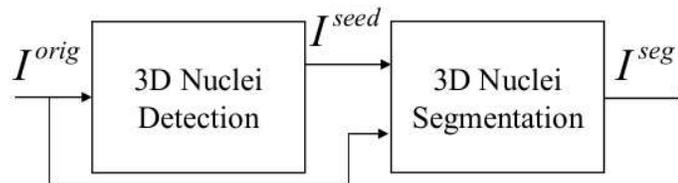


Fig. 5.1.: Block diagram of the proposed method

Figure 5.1 is a block diagram of the proposed method to segment individual nuclei in three dimensions. In this chapter, we define voxels in a nucleus as foreground and other voxels as background. Our goal is to segment nuclei individually. To achieve this, the first step is 3D nuclei detection to select seeds,  $I^{seed}$ , of a grayscale original microscopy volume,  $I^{orig}$ . Potential centers of nuclei are defined as “seeds.” After selecting seeds, the 3D nuclei segmentation stage segments nuclei surrounding each seed individually. The segmented image volume is denoted by  $I^{seg}$ . Our CNNs are implemented in Torch [103].

### 5.1.1 3D Nuclei Detection

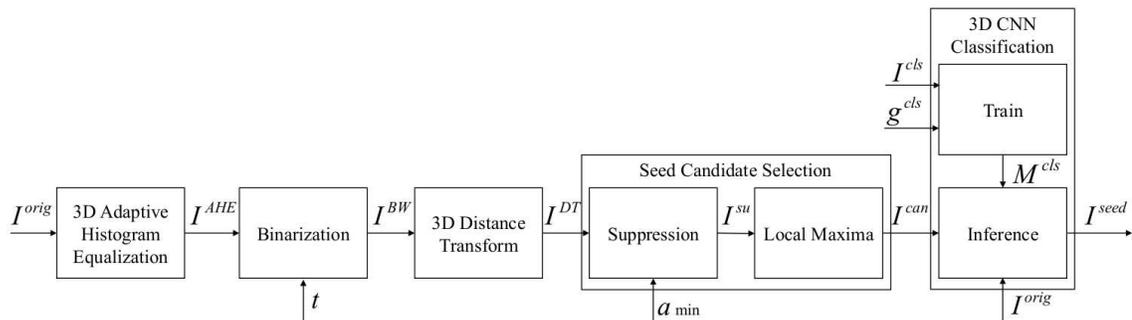


Fig. 5.2.: Block diagram of the 3D nuclei detection

Figure 5.2 shows the block diagram of the 3D nuclei detection stage. We assume that the nuclei are ellipsoidal and we use the 3D distance transform [44] to select seeds. To use the 3D distance transform, an original microscopy volume,  $I^{orig}$ , needs to be converted to a binary volume. Since fluorescence microscopy volumes generally have inhomogeneous intensities, a 3D adaptive histogram equalization is used as a pre-processing step to enhance the image volumes. The equalized image volume,  $I^{AHE}$ , is then binarized,  $I^{BW}$ , with a threshold,  $t$ . The 3D distance transform is then used to transform  $I^{BW}$  to generate  $I^{DT}$ . Next, seed candidates,  $I^{can}$ , are selected by the following steps: (1) suppress noise and non-nuclei structures by thresholding  $I^{DT}$  to generate  $I^{su}$  and (2) find local maxima from  $I^{su}$ . Finally, a model,  $M^{cls}$ , of a 3D

classification CNN is trained using subvolumes of a real microscopy volume,  $I^{cls}$ , with their ground truth labels,  $g^{cls}$ .  $M^{cls}$  is used to finalize a set of seeds,  $I^{seed}$ .

### 3D Adaptive Histogram Equalization and Binarization

To use the 3D distance transform, it is necessary to convert an original microscopy volume to a binary volume. Fluorescence microscopy volumes generally have non-uniform intensities, so using a fixed binary threshold for the original volume may not extract the foreground successfully. Therefore, a 3D adaptive histogram equalization is used to enhance the volume as a pre-processing step. A  $17 \times 17 \times 9$  rectangular window is used to equalize each voxel, where voxels near the volume boundaries are reflected past the boundaries. The size of the window is chosen according to the ratio of the resolution along the  $x$ ,  $y$ , and  $z$ -directions.

After enhancement, binarization is defined as:

$$I^{BW}(\mathbf{v}) = \begin{cases} 255, & \text{if } I^{AHE}(\mathbf{v}) > t \\ 0, & \text{otherwise} \end{cases} \quad (5.1)$$

where  $I^{AHE}$  is the histogram equalized volume,  $\mathbf{v}$  is a voxel location, and  $t$  is a threshold to have all voxels on nuclei in foreground region.  $t$  can be estimated by the ratio of the number of voxels in the background to the total number of voxels in a volume scaled by a factor of 255. This approach is the same as foreground segmentation step we presented in [38].

### 3D Distance Transform and Seed Candidate Selection

In this step, we use the 3D distance transform [44] on  $I^{BW}$  to find seed candidates. For each voxel location on  $I^{BW}$ , the minimum Euclidean distance to a background voxel is calculated.

To select seed candidates of centers of nuclei, we first suppress noise and non-nuclei structures by:

$$I^{su}(\mathbf{v}) = \begin{cases} I^{DT}(\mathbf{v}), & \text{if } I^{DT}(\mathbf{v}) > \frac{a_{min}}{2} \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

$a_{min}$  is the minimum semi-axis of the nuclei. The intensity of  $I^{DT}$  at the center of the nuclei must be greater than  $\frac{a_{min}}{2}$ , so the suppression process helps find seed candidates by removing noise and non-nuclei structures.

Since it is assumed that nuclei have ellipsoidal shape, voxels which is local maxima of  $I^{su}$  would be seed candidates. Hence, a voxel,  $\mathbf{v}$ , is selected as a seed candidate if (1)  $\mathbf{v}$  is the maximum in a window with size of  $13 \times 13 \times 9$  centered at  $\mathbf{v}$  of  $I^{su}$  and (2) there is no other seed candidates in the window to avoid selecting two or more seeds on one nucleus.

### A 3D Convolutional Neural Network for Classification

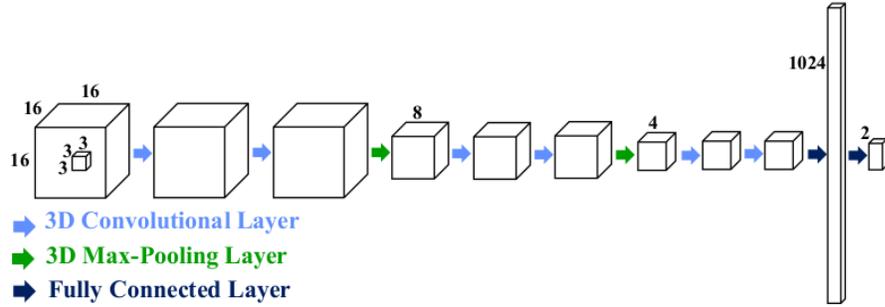


Fig. 5.3.: Architecture of our 3D CNN for Classification

The suppression process may not remove all non-nuclei structures. Therefore, a 3D classification CNN is used to filter out seed candidates which do not belong to nuclei.

Figure 5.3 shows the architecture of our 3D classification CNN. A 3D convolutional layer consists of a convolutional operation with a  $3 \times 3 \times 3$  kernel with 1 voxel padding,

3D batch normalization [191], and a rectified-linear unit (ReLU) activation function. A 3D max-pooling layer uses  $2 \times 2 \times 2$  window with a stride of 2. As a final stage, two fully connected layers with size of 1024 and 2 are used. An input volume to the network is a subvolume of  $I^{orig}$  with size of  $16 \times 16 \times 16$  centered at a seed candidate. The output of the network is a score vector with size of 2 to determine if the seed candidate is located in a nucleus. To train our model,  $M^{cls}$ , we use stochastic gradient descent (SGD) with a fixed learning rate of  $10^{-3}$  and a momentum of 0.9. Three hundred manually selected training volumes from a real fluorescence microscopy volume,  $I^{cls}$ , and their ground truth labels,  $g^{cls}$ , are used to train the model. 150 training volumes contain nuclei and the other 150 training volumes contain non-nuclei structures. The training of our 3D classification CNN took less than a day and the inference took approximately  $10^{-2}$  second using NVIDIA's GeForce GTX Titan X.

### 5.1.2 3D Nuclei Segmentation

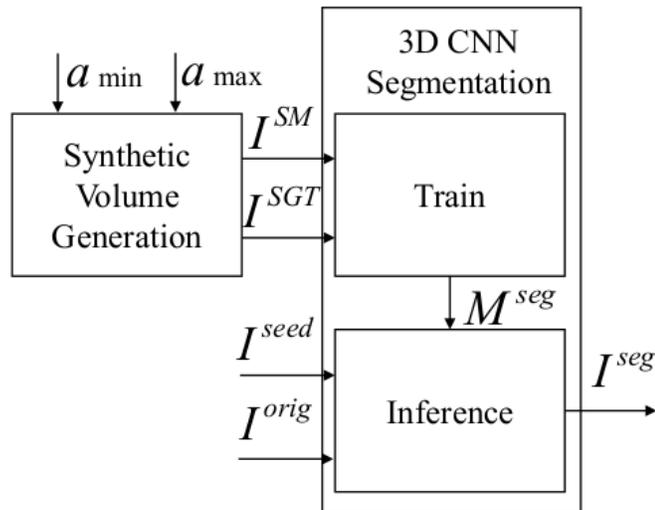


Fig. 5.4.: Block diagram of the proposed 3D nuclei segmentation

Figure 5.4 shows a block diagram of the proposed segmentation stage. For each seed,  $I^{seed}$ , selected from the 3D nuclei detection stage, 3D segmentation CNN,  $M^{seg}$ , segments nuclei individually on a subvolume of  $I^{orig}$  centered at the seed. Training 3D segmentation CNN requires many training volumes and ground truth volumes. It is extremely difficult to have manual 3D ground truth volumes from fluorescence microscopy volumes. Therefore, our approach is to generate synthetic microscopy volumes,  $I^{SM}$ , and their synthetic ground truth volumes,  $I^{SGT}$ , automatically to train the CNN. This approach is similar to the use of synthetic training we presented in [192]. The size of synthetic nuclei can be chosen based on the size of nuclei from the original microscopy volume to be segmented. After training the 3D segmentation CNN using  $I^{SM}$  and  $I^{SGT}$ ,  $M^{seg}$  can now segment individual nuclei.

### Synthetic Volume Generation

Synthetic ground truth volumes,  $I^{SGT}$ , and synthetic microscopy volumes,  $I^{SM}$ , are automatically generated to train our 3D segmentation CNN.  $I^{SGT}$  is first generated with an assumption that nuclei have ellipsoidal shape. Using  $I^{SGT}$ , we can create the corresponding  $I^{SM}$  by assigning intensities on foreground (nuclei) and background, blurring the volume, and adding noise to make synthetic microscopy volumes more realistic.

To generate  $I^{SGT}$ , ellipsoidal model is used:

$$I^{SGT}(\mathbf{v}) = \begin{cases} 1, & \text{if } \frac{\tilde{x}^2}{a_x^2} + \frac{\tilde{y}^2}{a_y^2} + \frac{\tilde{z}^2}{a_z^2} < 1 \\ 0, & \text{otherwise} \end{cases} \quad (5.3)$$

Here,  $(\tilde{x}, \tilde{y}, \tilde{z})$  is the translated and rotated coordinates from the original coordinates,  $\mathbf{v} = (x, y, z)$ , with  $t_x, t_y, t_z$ , the translations in  $x, y$ , and  $z$ -direction, respectively, and  $r_x, r_y, r_z$ , the rotations around  $x, y$ , and  $z$ -axes, respectively. Since synthetic nuclei would locate at the center of a volume with size of  $16 \times 16 \times 16$ , we set  $t_x = 8, t_y = 8$ , and  $t_z = 8$ , and  $r_x, r_y$ , and  $r_z$  are randomly chosen between 0 to 360. Also, we define  $(a_x, a_y, a_z)$  to be a vector of semi-axes length of synthetic nuclei.  $a_x, a_y$ , and  $a_z$  are

randomly selected between  $a_{min}$  and  $a_{max}$ , where  $a_{min}$  is the minimum semi-axis of nuclei and  $a_{max}$  is the maximum semi-axis of nuclei based on the semi-axes of nuclei in  $I^{orig}$ .

The synthetic microscopy volumes,  $I^{SM}$ , can be generated based on synthetic ground truth volumes,  $I^{SGT}$ . First, voxel intensities on foreground are set to be  $i_f$  and voxel intensities on background to be  $i_b$ . We set  $i_f$  to be randomly selected between 128 and 255 and set  $i_b = \frac{i_f}{3}$  by observing real fluorescence microscopy volumes. Next, the synthetic volumes are degraded by blurring and noise to model degradation seen in microscopy volumes. During the data acquisition, the point spread function (PSF) of the microscope blurs fluorescence microscopy volumes [17]. So we blur the volume using a 3D Gaussian filter with standard deviation of 1. Additionally, both Gaussian noise and Poisson noise are added to the volumes [28]. The noise model is  $I^{SM}(\mathbf{v}) = \alpha X + Y$  where  $X \sim \mathcal{P}(\lambda)$  and  $Y \sim \mathcal{N}(\mu, \sigma^2)$ . We set  $\alpha = 1$ ,  $\mu = 0$ ,  $\sigma = 5$ , and  $\lambda$  be  $i_f$  if  $\mathbf{v}$  belongs to foreground and  $i_b$  if  $\mathbf{v}$  belongs to background. During synthetic volume generation, the size of the nuclei, the intensity of the nuclei, and the background noise of synthetic microscopy volumes are close to original image volumes. Figure 5.5 shows an example of synthetic microscopy volume and its synthetic ground truth volume.

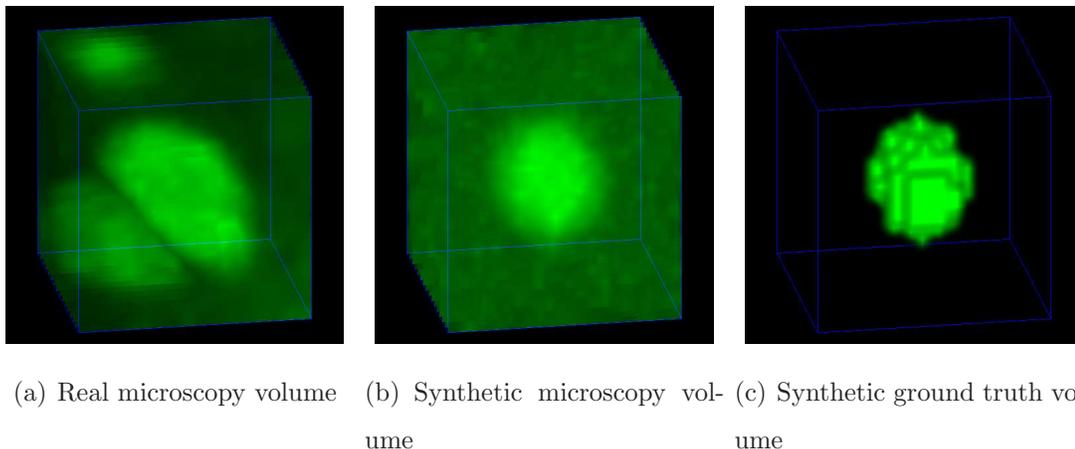


Fig. 5.5.: Examples of a real microscopy volume, a synthetic microscopy volume and synthetic ground truth volume

## A 3D Convolutional Neural Networks for Segmentation

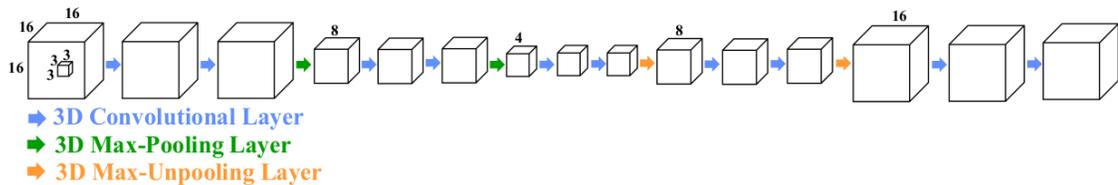


Fig. 5.6.: Architecture of the proposed 3D CNN for Segmentation

Figure 5.6 shows our architecture of a 3D CNN for segmentation. The architecture of the encoder part is identical to  $M^{cls}$  in Figure 5.3 until the first fully connected layer. In the decoder, 3D max-unpooling layers with 3D convolutional layers are used to reconstruct the output volume as the same size as the input volume. To train our model,  $M^{seg}$ , stochastic gradient descent (SGD) with a fixed learning rate of  $10^{-6}$  and a momentum of 0.9 is used. An input volume to the network is a subvolume of  $I^{orig}$  with size of  $16 \times 16 \times 16$  centered at a seed and an output volume is a binary volume with size of  $16 \times 16 \times 16$ . One hundred pairs of synthetic microscopy volumes,  $I^{SM}$ , and synthetic ground truth volumes,  $I^{SGT}$ , automatically generated in the previous step are used to train the model. The training of our 3D segmentation CNN took less than a day and the inference took approximately  $10^{-2}$  second using NVIDIA's GeForce GTX Titan X.

## 5.2 Experimental Results

We tested our method on three different rat kidney data sets: Data-N1, Data-N5, and Data-N6. Figure 5.7 shows the images obtained at each step for  $I_{z_{23}}$  in Data-N1, with  $t = 192$ ,  $a_{min} = 3$ , and  $a_{max} = 5$ . A classification CNN was trained by Data-N1. In Figure 5.7 (f), nuclei segmented on various seeds are labeled to different color. Figure 5.8 shows segmentation results in various depth.

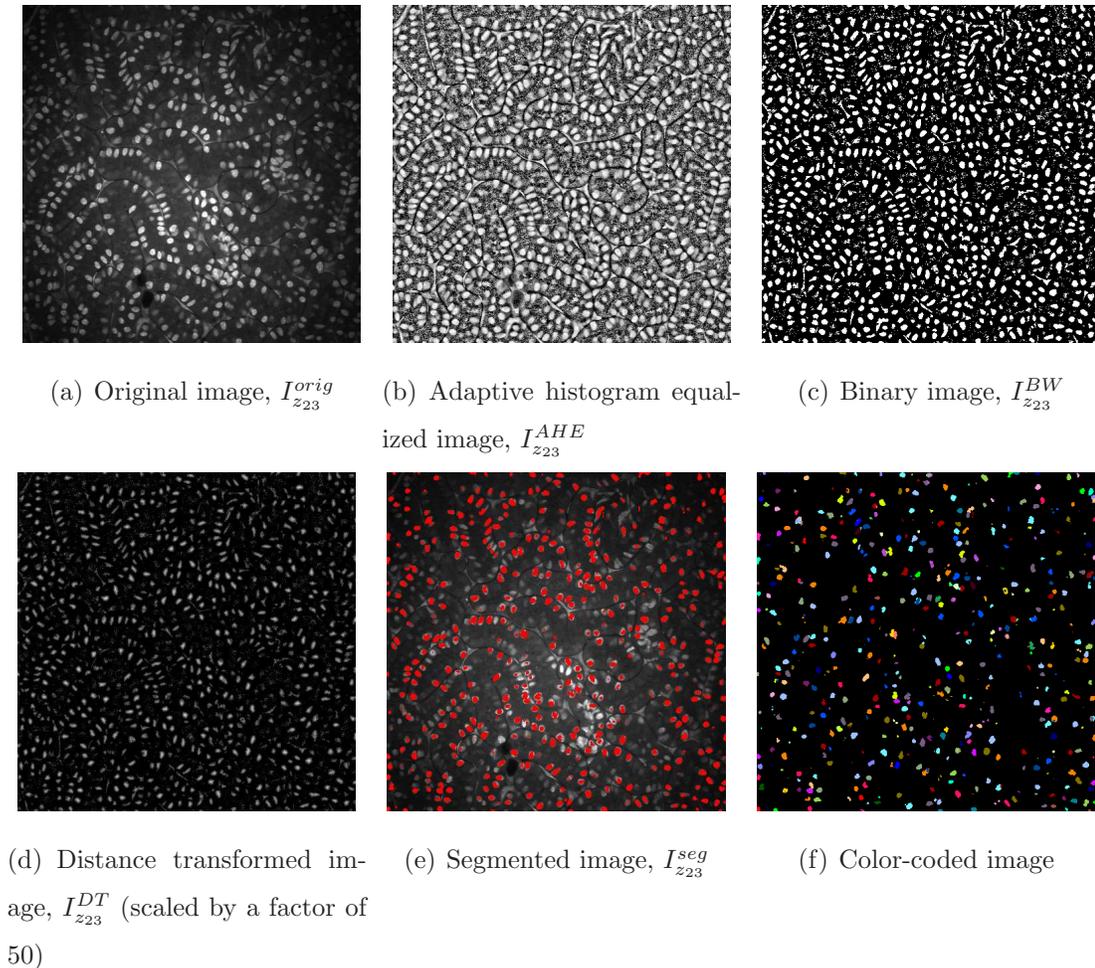


Fig. 5.7.: Original, intermediate, and segmented images in Data-N1

Our method was compared to other 3D segmentation methods used in microscopy images including 3D active surface with inhomogeneity correction (3DacIC) [34], 3D Squash [67,68], and 3D CNN we described in [192]. We evaluated our method using three subvolumes of Data-N1 by 3D real ground truth volumes. The real ground truth volumes for evaluation are manually generated from a real microscopy data set. We refer vol-1 as  $I_{(241:272,241:272,31:62)}$ , vol-2 as  $I_{(241:272,241:272,131:162)}$ , and vol-3 as  $I_{(241:272,241:272,231:262)}$ . Figure 5.9, Figure 5.10, and Figure 5.11 are the 3D visualization of various segmentation method results on different subvolumes, helped by Voxx [185]. It was observed that 3D active surface with inhomogeneity correction [34], 3D Squash

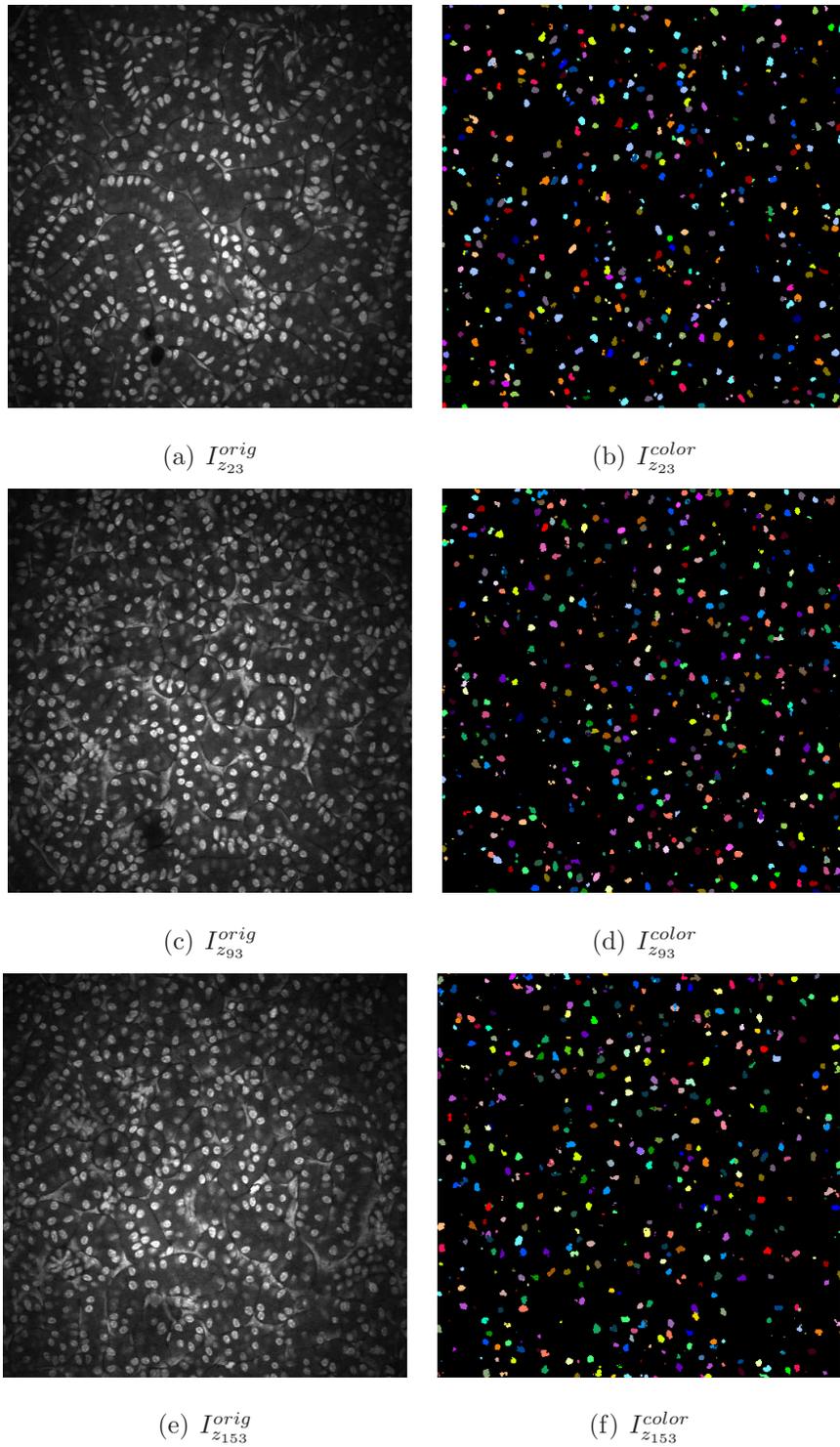


Fig. 5.8.: Original and segmented images of Data-N1

[67, 68], and our 3D CNN described in [192] cannot distinguish between nuclei and other subcellular structures. However, our method can distinguish between nuclei and other subcellular structures due to the suppression process and 3D classification CNN.

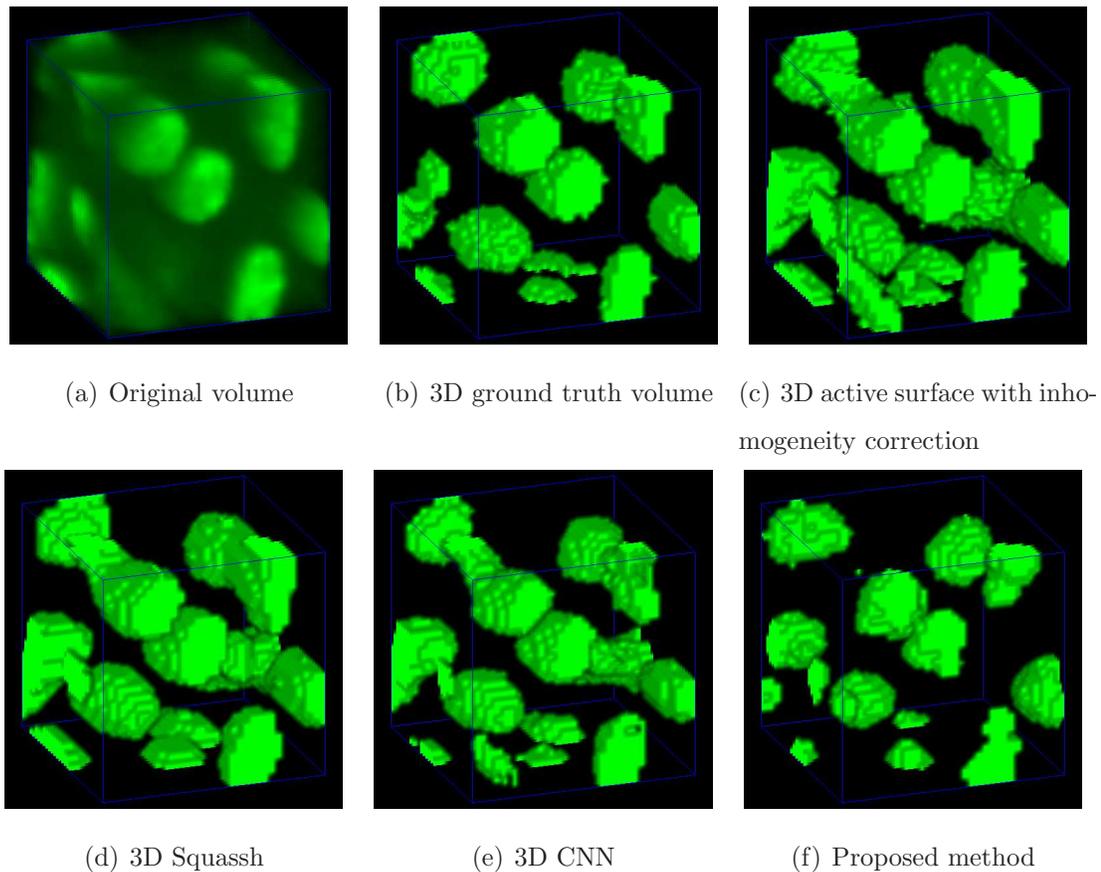


Fig. 5.9.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(241:272,241:272,31:62)}$  of Data-N1

All segmentation results were evaluated based on the following metrics:

$$\text{accuracy} = \frac{n_{\text{TP}} + n_{\text{TN}}}{n_{\text{total}}} \quad (5.4)$$

$$\text{Type-I error} = \frac{n_{\text{FP}}}{n_{\text{total}}} \quad (5.5)$$

$$\text{Type-II error} = \frac{n_{\text{FN}}}{n_{\text{total}}} \quad (5.6)$$

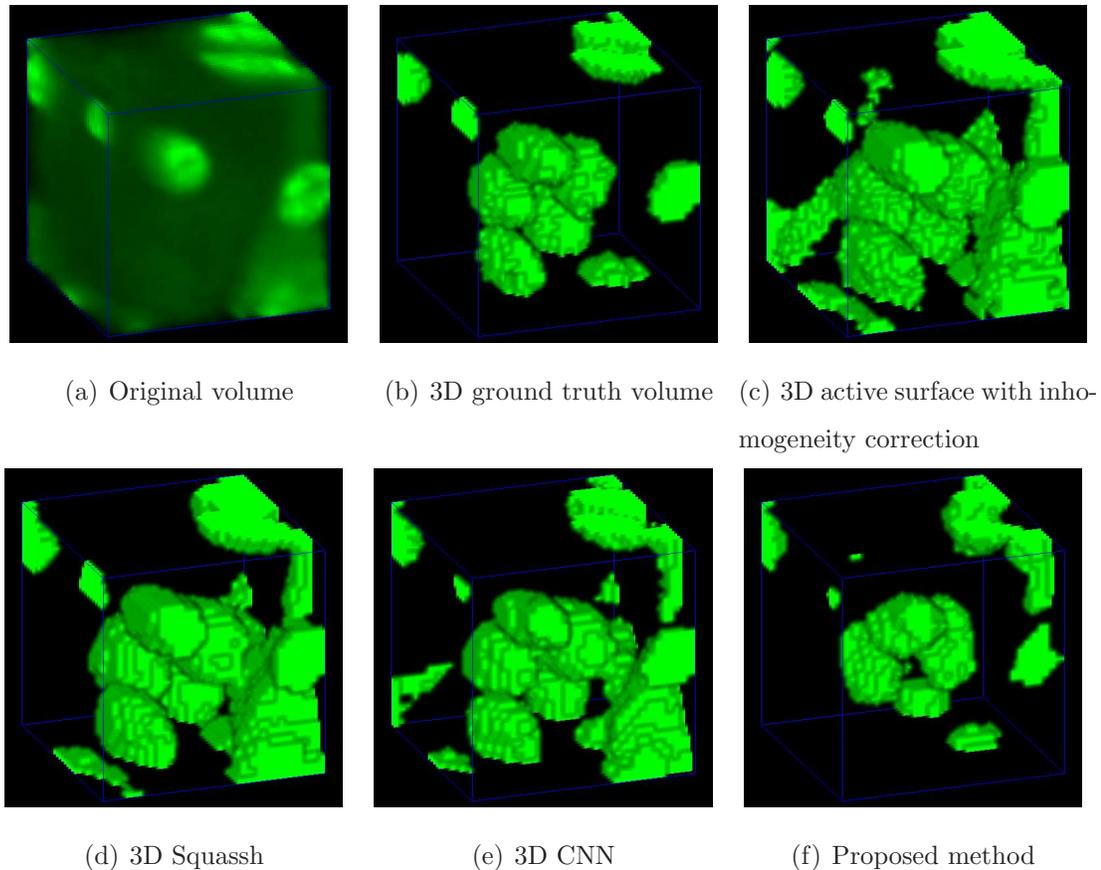


Fig. 5.10.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(241:272,241:272,131:162)}$  of Data-N1

where  $n_{TP}$ ,  $n_{TN}$ ,  $n_{FP}$ ,  $n_{FN}$ ,  $n_{total}$  are defined to be the number of true-positives (voxels correctly labeled as nuclei), true-negatives (voxels correctly labeled as background), false-positives (voxels wrongly labeled as nuclei), false-negatives (voxels wrongly labeled as background), and the total number of voxels in a volume, respectively [186, 187]. Table 5.1, Table 5.2, and Table 5.3 show the accuracy for various 3D segmentation methods and our method on three subvolumes on Data-N1.

Figure 5.12 and Figure 5.13 show that our method can successfully detect and segment nuclei from Data-N5 and Data-N6, respectively. Note that (1) Data-N5 and Data-N6 have uniform intensities so the 3D adaptive histogram equalization was not used (2) the suppression process was successful so the 3D classification CNN was not

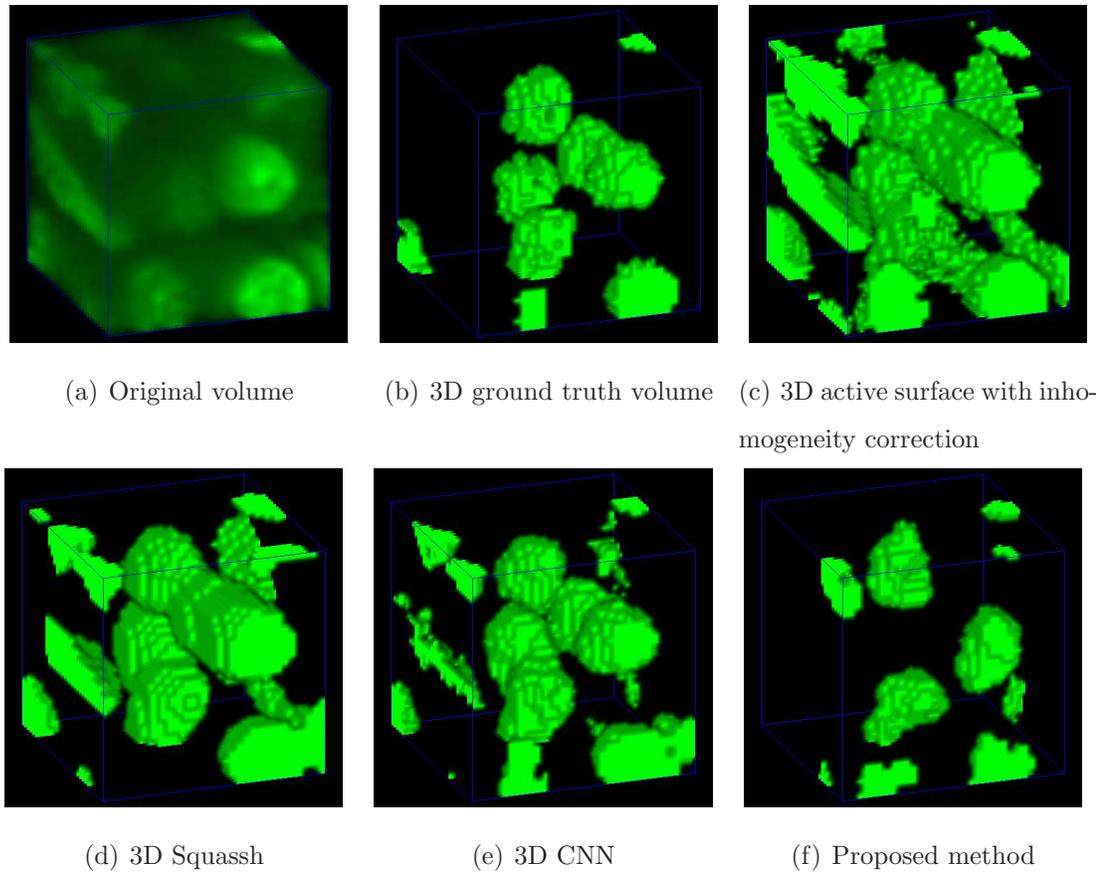


Fig. 5.11.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(241:272,241:272,231:262)}$  of Data-N1

Table 5.1.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(241:272,241:272,31:62)}$  of Data-N1

	Accuracy	Type-I error	Type-II error
3DacIC [34]	87.36%	12.44%	0.20%
3D Squassh [67, 68]	90.14%	9.07%	0.79%
3D CNN [192]	92.20%	5.38%	2.42%
Proposed Method	93.30%	1.65%	5.05%

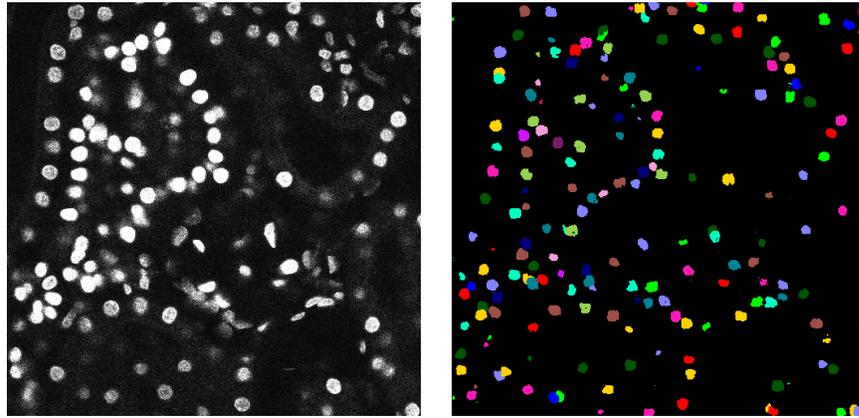
Table 5.2.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(241:272,241:272,131:162)}$  of Data-N1

	Accuracy	Type-I error	Type-II error
3DacIC [34]	86.78%	13.12%	0.10%
3D Squassh [67, 68]	88.26%	11.67%	0.07%
3D CNN [192]	92.32%	6.81%	0.87%
Proposed Method	93.48%	1.74%	4.78%

Table 5.3.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(241:272,241:272,231:262)}$  of Data-N1

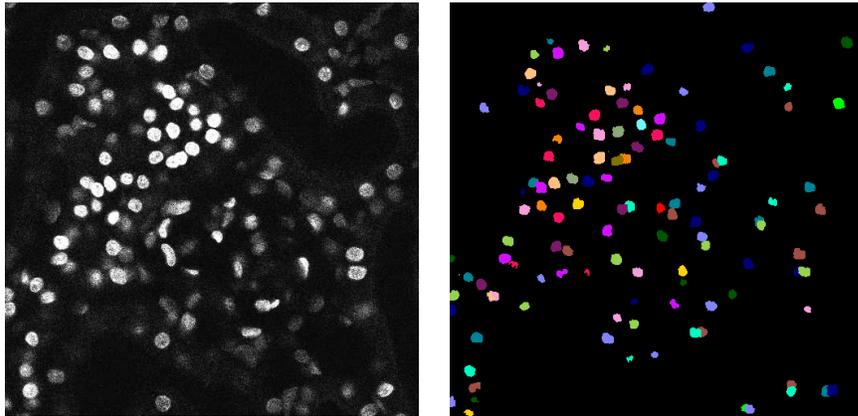
	Accuracy	Type-I error	Type-II error
3DacIC [34]	83.47%	16.53%	0.00%
3D Squassh [67, 68]	87.29%	12.61%	0.10%
3D CNN [192]	94.26%	5.19%	0.55%
Proposed Method	94.67%	1.56%	3.77%

used (3) we set  $t = 64$  for Data-N5 and  $t = 25$  for Data-N6 according to the intensity of nuclei (4) we set  $a_{min} = 6$  and  $a_{max} = 8$  for Data-N5 and Data-N6 according to the size of nuclei .



(a)  $I_{z_{16}}^{\text{orig}}$  of Data-N5

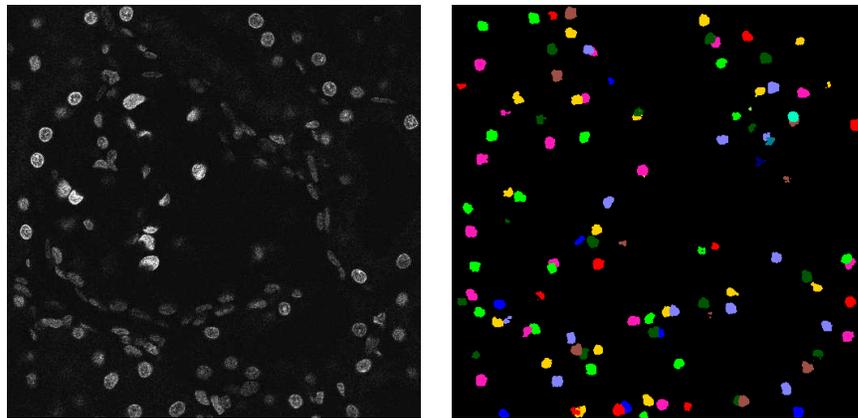
(b)  $I_{z_{16}}^{\text{seg}}$  of Data-N5



(c)  $I_{z_{27}}^{\text{orig}}$  of Data-N5

(d)  $I_{z_{27}}^{\text{seg}}$  of Data-N5

Fig. 5.12.: Original and segmented images of Data-N5



(a)  $I_{z_9}^{\text{orig}}$  of Data-N6

(b)  $I_{z_9}^{\text{seg}}$  of Data-N6

Fig. 5.13.: Original and segmented images of Data-N6

## 6. CENTER-EXTRACTION-BASED NUCLEI INSTANCE SEGMENTATION

In this chapter we describe a nuclei instance segmentation method. We define “instance segmentation” as a segmentation process that each nucleus is segmented with distinct labels. To compare to instance segmentation, we define “binary segmentation” as a segmentation process that each voxel is classified in binary whether it belongs to nuclei or not. Nuclei instance segmentation is more challenging than nuclei binary segmentation because touching or overlapping nuclei are required to be separated. We describe a two-stage method for nuclei instance segmentation. The CNN in the first stage detects a set of coordinates for centroids of nuclei by extracting the central region of nuclei and produces a binary segmentation mask. Using the set of coordinates and the binary segmentation mask the CNN in the second stage segments each nucleus in distinct labels. To train our CNNs we use a set of synthetic microscopy images generated by a spatially-constrained cycle-consistent generative adversarial networks (SpCycleGAN) described in [39]. The method described in this chapter outperforms our previous technique in Chapter 5 which detects the location of nuclei using a 3D distance transform and segments nuclei using a 3D CNN trained by a set of blurred and noisy synthetic volumes.

### 6.1 Proposed Method

Figure 6.1 is a block diagram of our proposed method for 3D nuclei instance segmentation. Our method consists of two CNNs as shown in Figure 6.1. The first CNN is used for nuclei detection and binary segmentation and has two outputs, a set of coordinates of the nuclei center location denoted as  $C^{det}$  and a nuclei mask volume denoted as  $I^{mask}$ . These two outputs are generated from an original microscopy volume,

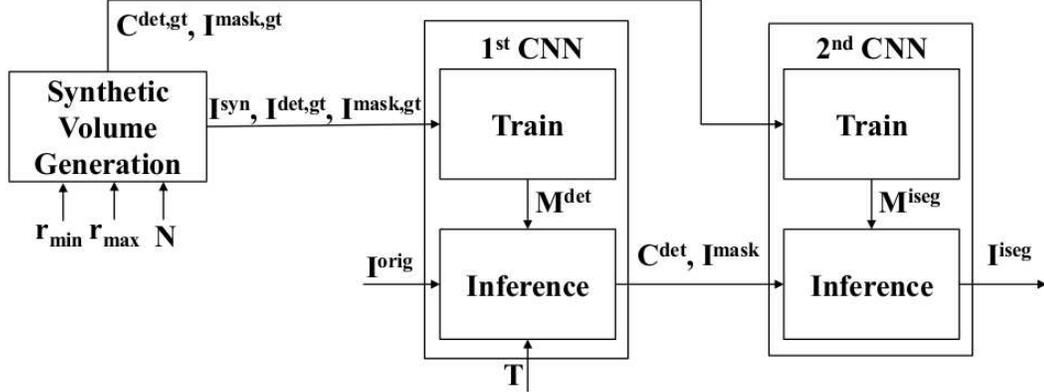


Fig. 6.1.: Block diagram of the proposed method

$I^{orig}$ .  $C^{det}$  is selected by finding centroid coordinates of components in a detection volume,  $I^{det}$ . To accurately select  $C^{det}$  at the nuclei center locations, especially when multiple nuclei are touching, the components in  $I^{det}$  are designed to have no touching or overlapping regions for distinct nuclei. Therefore,  $I^{det}$  contains components around the nuclei center locations.

The second CNN is used for nuclei instance segmentation where an individual nucleus is segmented in a 3D patch from  $I^{mask}$  centered at  $C^{det}$  and is color-coded. After color-coding individual nucleus, the final segmentation volume,  $I^{iseg}$ , is produced. To train the two CNNs a SpCycleGAN described in [39] is used to generate synthetic microscopy volumes,  $I^{syn}$ . Our implementation is done using PyTorch [104].

### 6.1.1 Synthetic Volume Generation

As we indicated above creating labeled ground truth 3D volumes is tedious. We use the SpCycleGAN we described in [39] to produce synthetic microscopy 3D volumes that we use for training. Note we do not need any actual ground truth volumes to use the approach described in this section. Synthetic microscopy volumes,  $I^{syn}$ , nuclei mask ground truth volumes,  $I^{mask,gt}$ , and detection ground truth volumes,  $I^{det,gt}$ , need to be generated. We start by generating a random nuclei mask 3D volume and

then use it to generate the synthetic volume. In our approach synthetic nuclei can be generated with either spherical or elliptical based nuclei shapes in  $I^{orig}$  with  $N$  synthetic nuclei. For spherical shapes the  $i^{\text{th}}$  synthetic nuclei,  $I^{mask,i}$ , is generated as a sphere with randomly selected radius,  $r_i$ , between  $r_{min}$  and  $r_{max}$ , and centered at a randomly selected coordinate,  $C^{det,i}$ , where  $1 \leq i \leq N$ .

$$I^{mask,i}(\mathbf{v}) = \begin{cases} 1, & \text{if } \|\mathbf{v} - C^{det,i}\|_2^2 < r_i^2 \\ 0, & \text{otherwise} \end{cases} \quad (6.1)$$

where  $\mathbf{v}$  is denoted as a voxel. Simultaneously, the  $i^{\text{th}}$  central region,  $I^{det,i}$ , is generated. We define a central region of a nucleus as a sub-region inside the nucleus where the centroid of the sub-region matches to the centroid of the nucleus.

$$I^{det,i}(\mathbf{v}) = \begin{cases} 1, & \text{if } \|\mathbf{v} - C^{det,i}\|_2^2 < \left(\frac{r_i}{2}\right)^2 \\ 0, & \text{otherwise} \end{cases} \quad (6.2)$$

We intentionally set the radius of  $I^{det,i}$  to be  $\frac{r_i}{2}$  to avoid multiple connected central regions although their corresponding synthetic nuclei may be touching. Once  $N$  synthetic nuclei and their central regions are produced, they are added to  $I^{mask,gt}$  and  $I^{det,gt}$  sequentially where  $I^{mask,gt}$  and  $I^{det,gt}$  are initialized to zero. If  $I^{mask,i}$  overlaps with any previous synthetic nuclei in  $I^{mask,gt}$ , then  $I^{mask,i}$  and  $I^{det,i}$  are not added to  $I^{mask,gt}$  and  $I^{det,gt}$ , respectively.

For elliptical nuclei  $I^{mask,i}$  is generated as an ellipsoid with randomly and independently selected three semi-axes between  $r_{min}$  and  $r_{max}$ , randomly rotated in  $x$ ,  $y$ , and  $z$ -axes, and centered at a randomly selected coordinate,  $C^{det,i}$ . In our experiments we used both shape types for generating synthetic images.

Once the nuclei mask ground truth volume,  $I^{mask,gt}$ , and the detection ground truth volume,  $I^{det,gt}$ , are generated, we use the SpCycleGAN to generate the corresponding synthetic volume,  $I^{syn}$ . For our experiments we generated 20 sets of synthetic volumes with a size of  $128 \times 128 \times 128$ . Figure 6.2 shows examples of a real microscopy volume, a synthetic microscopy volume, and synthetic ground truth volumes visualized by VoxX [185].

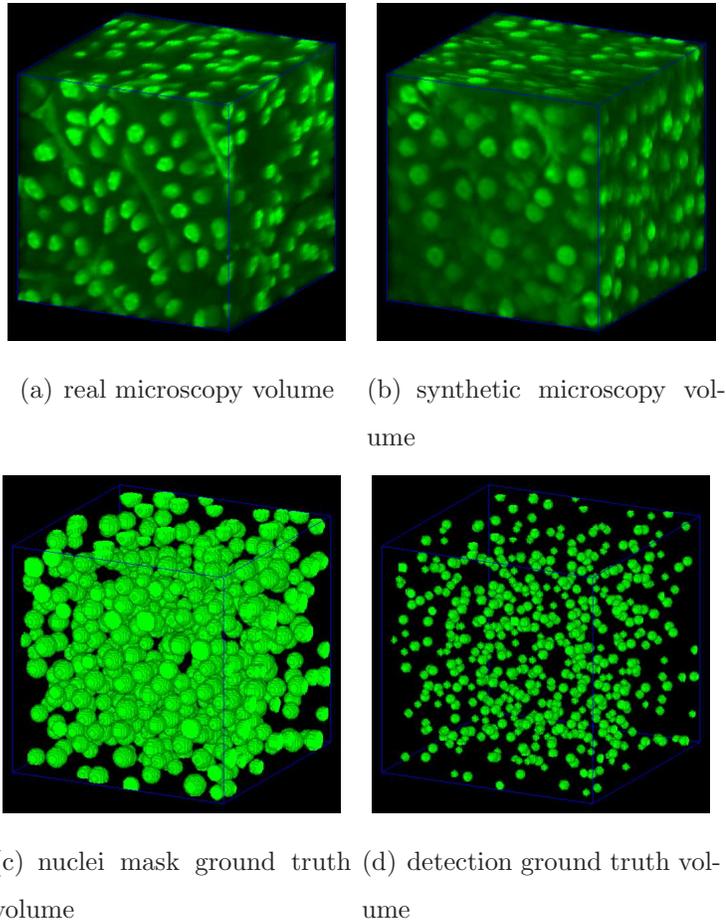


Fig. 6.2.: Examples of a real microscopy volume, a synthetic microscopy volume and synthetic ground truth volumes

### 6.1.2 Nuclei Detection and Binary Segmentation

Our first CNN used for nuclei detection and binary segmentation outputs nuclei center locations,  $C^{det}$ , and a nuclei mask volume,  $I^{mask}$  (Figure 6.1). This CNN is shown in more detail in Figure 6.3 and has an encoder-decoder architecture which is modified from 3D U-Net [168].  $C^{det}$  can be selected by finding centroids for components in  $I^{det}$ . To avoid false-detection, labels in  $I^{det}$  are labeled as background if  $I^{mask}$  at the same voxel locations are labeled as background. Also, components with the number of voxels less than  $T$  are not considered. A 3D convolutional layer consists

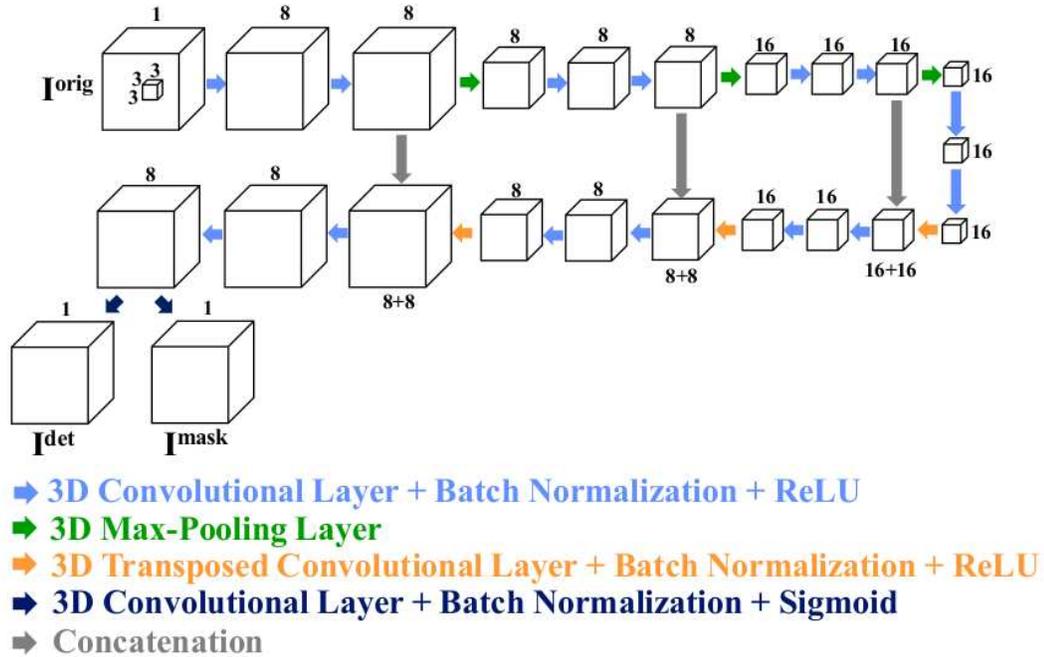


Fig. 6.3.: Our first CNN architecture (see Figure 6.1)

of a convolutional operation with a  $3 \times 3 \times 3$  kernel with 1 voxel padding, 3D batch normalization [191], and a rectified-linear unit (ReLU) activation function. Note that the Sigmoid function is used as an activation function for the last convolutional layers. In the encoder, 3D max-pooling layer uses  $2 \times 2 \times 2$  window with a stride of 2. In the decoder, a 3D transposed convolutional layer followed by 3D batch normalization [191] and ReLU activation function is used. In addition, concatenation transfers feature maps from the encoder to the decoder. The size of input/output volumes are  $64 \times 64 \times 64$ . If the size of  $I^{orig}$  is larger than  $64 \times 64 \times 64$ , then a 3D window with size of  $64 \times 64 \times 64$  is moved in the  $x$ ,  $y$ , and  $z$ -directions until the entire  $I^{orig}$  is processed [192]. During training, the Adam optimizer [100] is used with a learning rate of 0.001. The training loss function is a sum of the Binary Cross Entropy (BCE)



The second CNN in Figure 6.1,  $M^{iseg}$ , consists of a series of 3D convolutional layers. We use dilated convolutions [126] to have receptive field larger than the size of the patch. From the  $k^{\text{th}}$  feature map,  $I^k$ , with a convolution filter,  $h$ , the  $(k + 1)^{\text{th}}$  feature map,  $I^{k+1}$ , is generated using a  $d$ -dilated convolution at a voxel  $\mathbf{v}$  as

$$I^{k+1}(\mathbf{v}) = \sum_{\mathbf{u}} I^k(\mathbf{v} - d\mathbf{u})h(\mathbf{u}) \quad (6.4)$$

where  $d$  is known as the dilation factor. Dilated convolutions are used to exponentially increase the receptive fields. Table 6.1 shows the dilation factors for the convolutional layers such that the final receptive field is larger than  $32 \times 32 \times 32$ . During training, the Adam optimizer [100] is used with a learning rate of 0.001. The BCE loss is used as the training loss function. 300 patches from  $I^{mask,gt}$  centered at  $C^{det,gt}$  are used for the training. The training of our second CNN took less than a day and the inference took approximately  $10^{-2}$  second using NVIDIA’s GeForce GTX Titan X.

Table 6.1.: Dilation factors for the convolutional layers and their corresponding receptive fields

Conv. Layer	Filter Size	Dilation Factor	Receptive Field
1 <sup>st</sup> layer	$3 \times 3 \times 3$	1	$3 \times 3 \times 3$
2 <sup>nd</sup> layer	$3 \times 3 \times 3$	2	$7 \times 7 \times 7$
3 <sup>rd</sup> layer	$3 \times 3 \times 3$	4	$15 \times 15 \times 15$
4 <sup>th</sup> layer	$3 \times 3 \times 3$	8	$31 \times 31 \times 31$
5 <sup>th</sup> layer	$3 \times 3 \times 3$	1	$33 \times 33 \times 33$
6 <sup>th</sup> layer	$1 \times 1 \times 1$	1	$33 \times 33 \times 33$

## 6.2 Experimental Results

Our method is tested on three rat kidney data sets: Data-N1, Data-N7, and Data-N5. To match resolution in  $z$ -direction to resolution in  $x$  and  $y$ -directions, Data-N7

is downsampled in  $z$ -direction by a factor of 2 and Data-N5 is linearly interpolated in  $z$ -direction by a factor of 2.  $r_{min} = 4$ ,  $r_{max} = 6$ , and  $N = 1000$  with a spherical model and  $T = 10$  for Data-N1,  $r_{min} = 6$ ,  $r_{max} = 9$ , and  $N = 200$  with an ellipsoidal model and  $T = 20$  for Data-N7, and  $r_{min} = 6$ ,  $r_{max} = 9$ , and  $N = 50$  with a spherical model and  $T = 30$  for Data-N5 are used, respectively. Note that the size of synthetic nuclei for Data-N1 is small, so the size of patches during nuclei instance segmentation is reduced to  $16 \times 16 \times 16$  and the fourth convolutional layer in  $M^{iseg}$  is removed. Figure 6.5, Figure 6.6, Figure 6.7 show original images and segmented images for Data-N1, Data-N7, and Data-N5, respectively.

Our method was compared to other segmentation methods using voxel-wise and object-wise evaluation criteria [186, 187]. The other segmentation methods include watershed [36], Squassh [67, 68], our previous detection and segmentation method [193] that we will denote as Det-Seg, and our previous segmentation method using a SpCycleGAN [39] that we will denote as Seg-Morph. For the voxel-wise evaluation, we define

$$\text{accuracy} = \frac{n_{TP} + n_{TN}}{n_{total}} \quad (6.5)$$

$$\text{Type-I error} = \frac{n_{FP}}{n_{total}} \quad (6.6)$$

$$\text{Type-II error} = \frac{n_{FN}}{n_{total}} \quad (6.7)$$

where  $n_{TP}$ ,  $n_{TN}$ ,  $n_{FP}$ ,  $n_{FN}$ ,  $n_{total}$  are the number of true positive voxels (voxels correctly labeled as nuclei), true negative voxels (voxels correctly labeled as background), false positive voxels (voxels wrongly labeled as nuclei), false negative voxels (voxels wrongly labeled as background), and the total number of voxels in a volume, respectively. For the object-wise evaluation, Precision ( $P$ ), Recall ( $R$ ), and F1 score ( $F1$ ) are defined as

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (6.8)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (6.9)$$

$$F1 = \frac{2PR}{P + R} \quad (6.10)$$

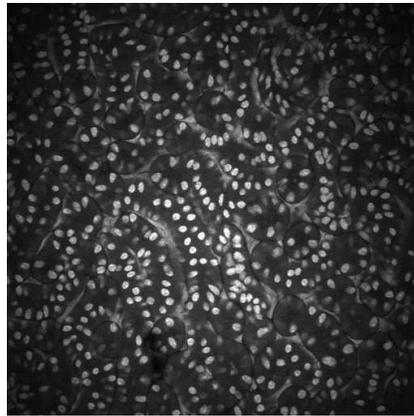
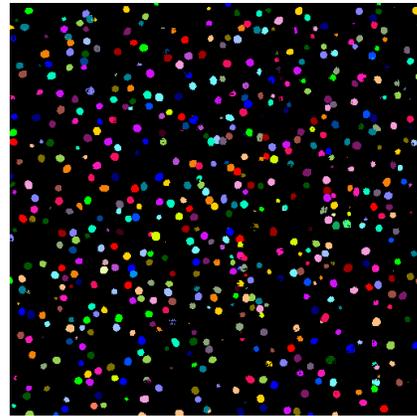
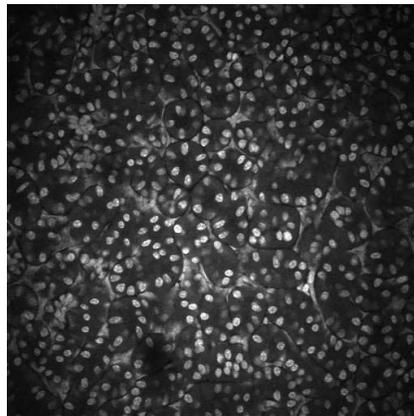
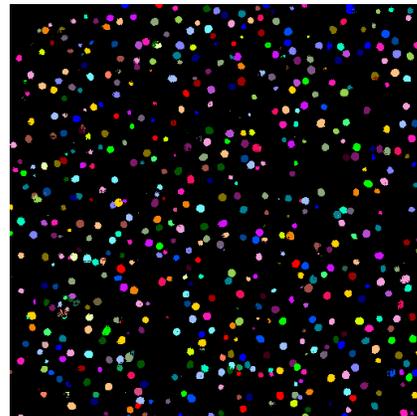
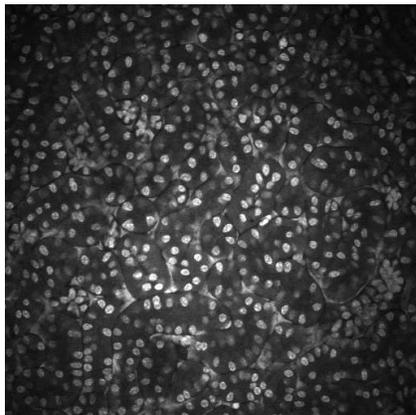
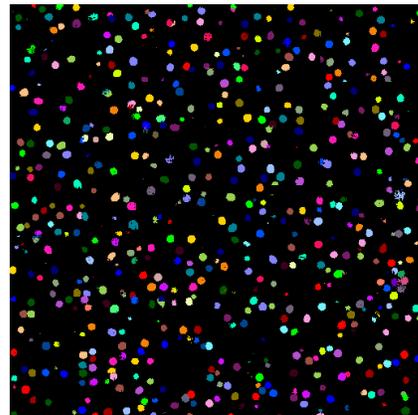
(a)  $I_{z_{36}}^{orig}$  for Data-N1(b)  $I_{z_{36}}^{iseg}$  for Data-N1(c)  $I_{z_{97}}^{orig}$  for Data-N1(d)  $I_{z_{97}}^{iseg}$  for Data-N1(e)  $I_{z_{159}}^{orig}$  for Data-N1(f)  $I_{z_{159}}^{iseg}$  for Data-N1

Fig. 6.5.: Original and segmented images of Data-N1

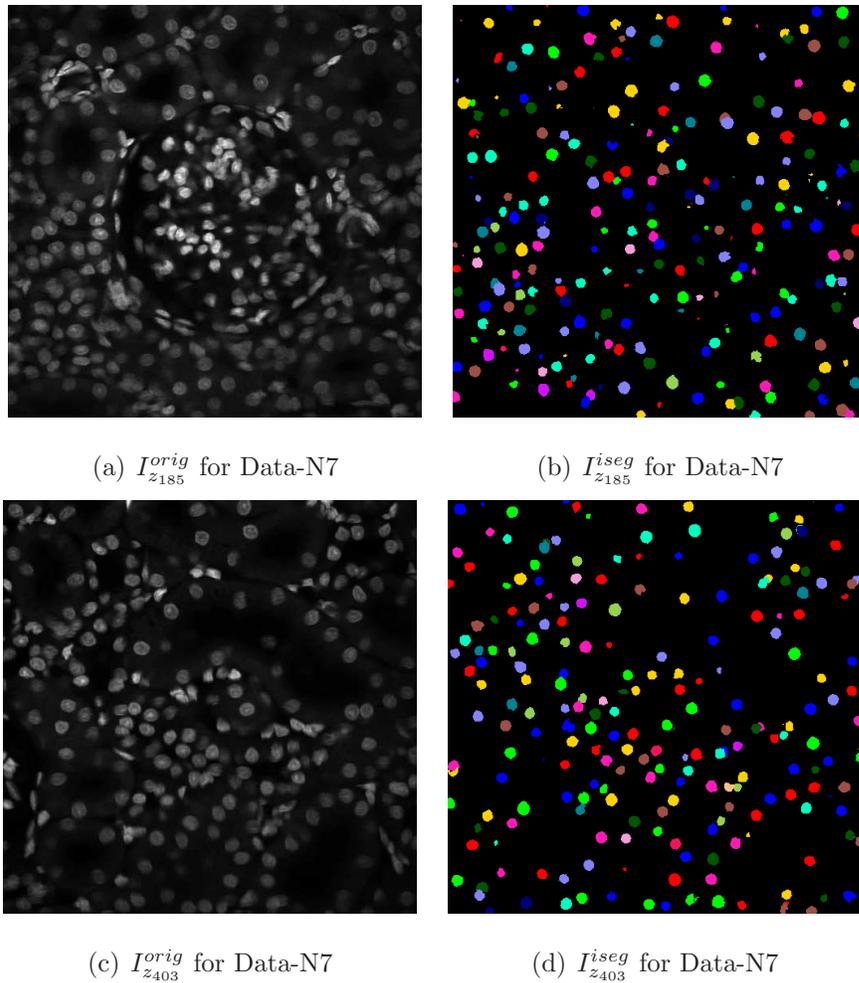


Fig. 6.6.: Original and segmented images of Data-N7

where  $N_{TP}$ ,  $N_{FP}$ , and  $N_{FN}$  are the number of true positive objects, the number of false positive objects, and the number of false negative objects, respectively. A segmented nucleus is defined as a true positive object if it intersects at least 50% of the corresponding ground truth nucleus. Otherwise, it is defined as a false positive object. A ground truth nucleus is defined as a false negative object if it intersects less than 50% of the corresponding segmented nucleus or there is no corresponding segmented nucleus. In our evaluation, we generated a 3D ground truth volume,  $I_{(193:320,193:320,31:94)}^{gt}$ , using ITK-SNAP [194] from Data-N1 with size of  $128 \times 128 \times 64$  containing 283 nuclei. Note that any components whose number of voxels is less

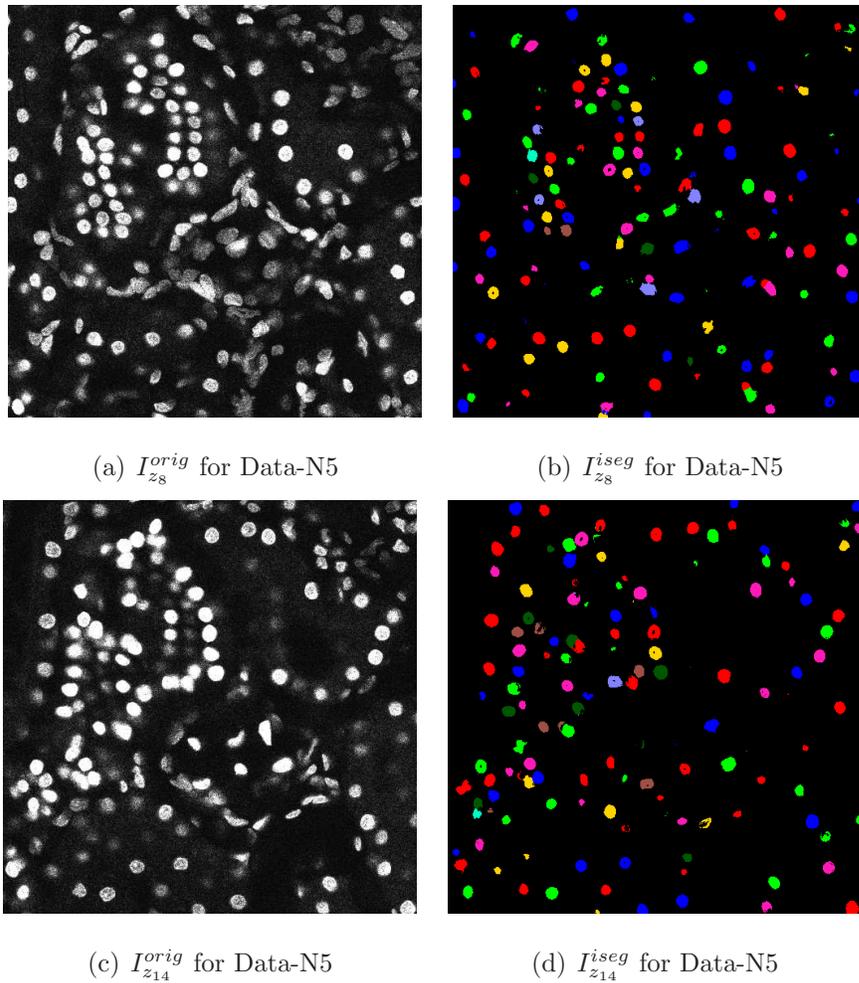


Fig. 6.7.: Original and segmented images of Data-N5

than 50 are removed on  $I_{(193:320,193:320,31:94)}^{iseg}$  and  $I_{(193:320,193:320,31:94)}^{gt}$  to remove partially included nuclei on the boundary of the subvolume.

Table 6.2 and Table 6.3 show the voxel-based evaluation and the object-based evaluation, respectively. Figure 6.8 shows the segmentation results of other methods and our new proposed method for  $I_{z_{297}}$  of Data-N1. Figure 6.9 shows the segmentation results of other methods and our new proposed method for Data-N1, visualized by Voxx [185]. For watershed [36],  $I^{orig}$  is first binarized by a manually-selected threshold value of 64. Thresholding cannot distinguish nuclei and non-nuclei structures and watershed technique over-segments foreground region. Squassh [67, 68] cannot

distinguish nuclei and non-nuclei structures and cannot separate touching nuclei successfully. Det-Seg [193] has a low Type-I error by rejecting non-nuclei structures but still have a poor F1 score. Seg-Morph [39] can generate an accurate binary segmentation mask. To separate touching nuclei, we used morphological operations with a 3D erosion, a 3D connected component for color-coding, and a 3D dilation with a sphere of radius of 1 used as the structuring element. We observe that morphological operations cannot separate all touching nuclei. Our proposed method, detecting the locations of nuclei and individually segmenting nuclei in 3D patches using the SpCycleGAN, can successfully segment and separate nuclei.

Table 6.2.: Accuracy, Type-I and Type-II errors for various methods on  $I_{(193:320,193:320,31:94)}$  of Data-N1

	Accuracy	Type-I error	Type-II error
Watershed [36]	59.25%	40.75%	0.00%
Squassh [67, 68]	80.45%	19.54%	0.01%
Det-Seg [193]	93.65%	1.71%	4.64%
Seg-Morph [39]	95.99%	2.32%	1.69%
Proposed Method	94.73%	3.85%	1.42%

Table 6.3.: Precision, Recall, and F1 score for various methods on  $I_{(193:320,193:320,31:94)}$  of Data-N1

	Precision	Recall	F1 score
Watershed [36]	51.14%	92.13%	65.78%
Squassh [67, 68]	85.07%	20.14%	32.57%
Det-Seg [193]	68.35%	90.22%	77.78%
Seg-Morph [39]	91.20%	82.01%	86.36%
Proposed Method	93.47%	96.80%	95.10%

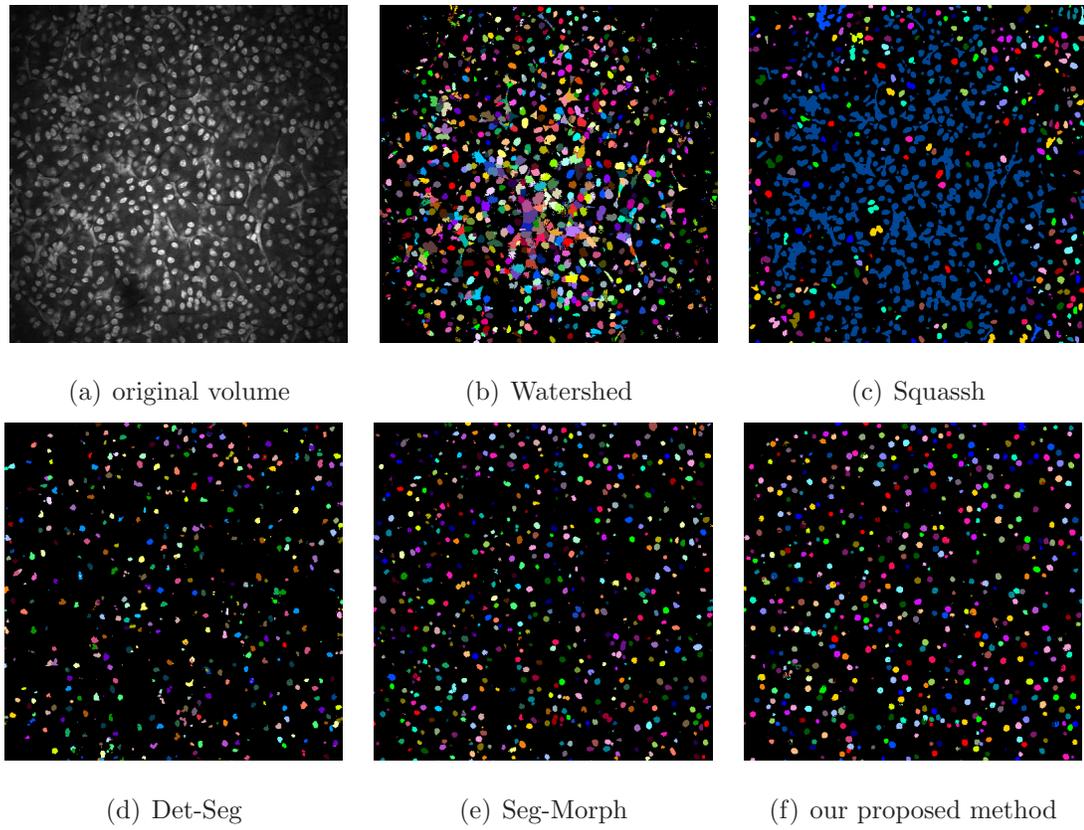


Fig. 6.8.: Comparison of other segmentation methods and our proposed method of the 97<sup>th</sup> image of Data-N1

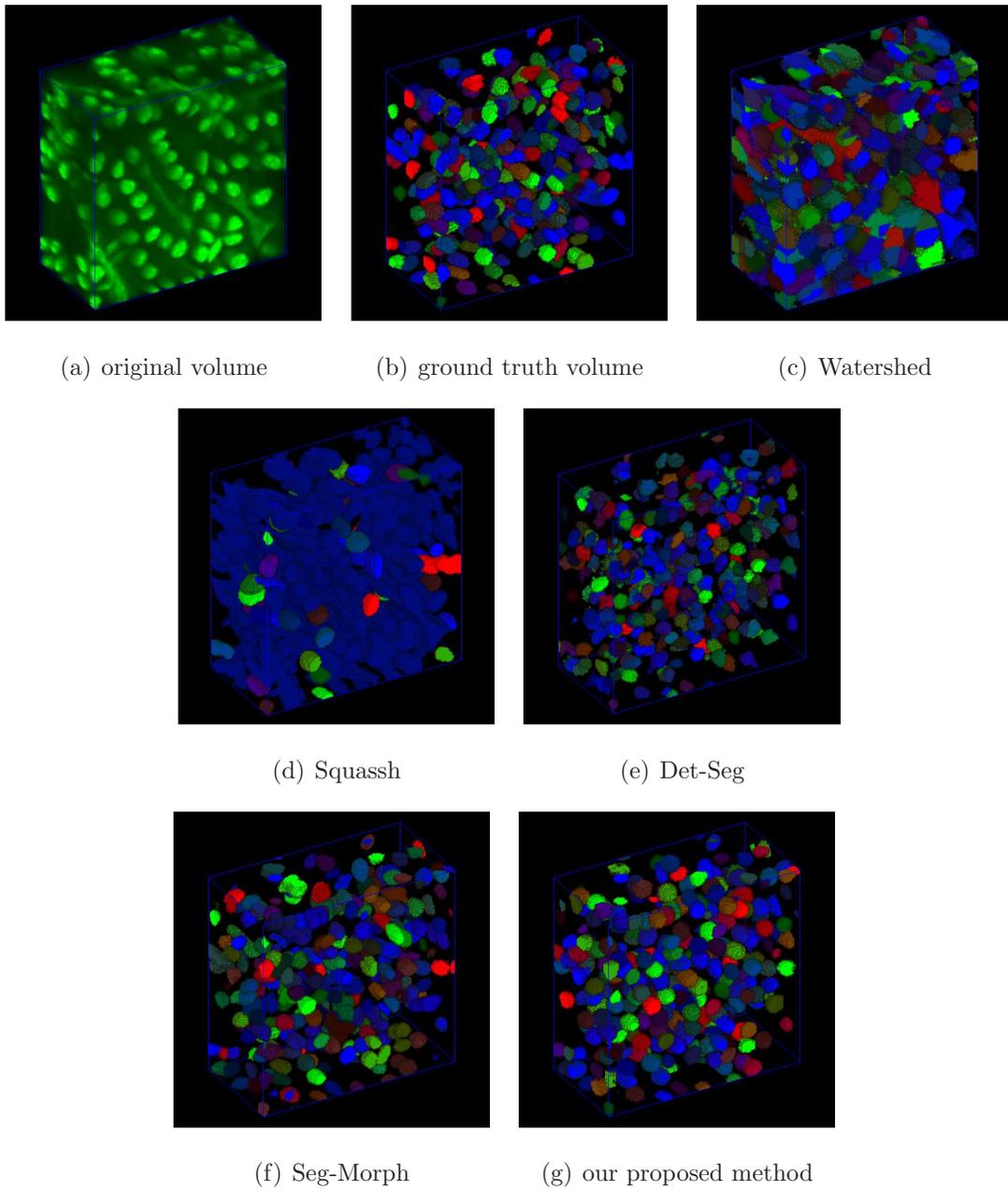


Fig. 6.9.: 3D visualization comparison of other segmentation methods and our proposed method of  $I_{(193:320,193:320,31:94)}$  of Data-N1



Figure 7.1 is a block diagram of the proposed method to detect nuclei in 3D. The goal is to estimate the location and size of nuclei presented in real fluorescence microscopy 3D volumes using our Sphere Estimation Network (SphEsNet). To train SphEsNet, we produce nuclei center volumes,  $I^{ctr}$ , to estimate the location of nuclei, nuclei radius volumes,  $I^{rad}$ , to estimate the size of nuclei, and nuclei mask volumes,  $I^{mask}$ , to generate synthetic microscopy volumes,  $I^{syn}$ , using a SpCycleGAN we described in [39]. Note  $I^{mask}$  is composed of multiple spherical masks. The location and size of synthetic nuclei in  $I^{syn}$  match to the location and size of spherical masks in  $I^{mask}$ . Once the SphEsNet model,  $M$ , is trained by  $I^{syn}$ ,  $I^{ctr}$ , and  $I^{rad}$ , it can be used on a real microscopy volume,  $I^{orig}$ , to detect nuclei. A pre-processing step on  $I^{orig}$  can be used to achieve a better result, where a pre-processed volume is denoted as  $I^{pre}$ . After processing  $I^{pre}$  using SphEsNet, the final detection volume,  $I^{det}$ , contains color-coded estimated spheres representing nuclei in the microscopy volume. SphEsNet is implemented in PyTorch [104].

### 7.1.1 Synthetic Volume Generation

The process of labeling ground truth for microscopy volumes in 3D is tedious. A method of generating synthetic volumes to train the SphEsNet is required. In this section we use a SpCycleGAN we described in [39] and in Section 2.2.3 to generate synthetic volumes for training. Note no real ground truth volumes are required in this technique. The Synthetic Volume Generation stage is composed of two steps: the Synthetic Groundtruth Volume Generation step and the Synthetic Microscopy Volume Generation step. The Synthetic Groundtruth Volume Generation step generates a set of nuclei center volumes,  $I^{ctr}$ , a set of nuclei radius volumes,  $I^{rad}$ , and a set of nuclei mask volumes,  $I^{mask}$ . The Synthetic Microscopy Volume Generation step generates a set of synthetic microscopy volume,  $I^{syn}$ , using the SpCycleGAN we described in [39]. Three sets of volumes,  $I^{syn}$ ,  $I^{ctr}$ , and  $I^{rad}$ , are then used to

train SphEsNet. In this stage, 20 synthetic volumes with size of  $128 \times 128 \times 128$  are generated.

### Synthetic Groundtruth Volume Generation

During the Synthetic Groundtruth Volume Generation step,  $I^{mask}$  is first generated. Note  $I^{mask}$  is used in the Synthetic Microscopy Volume Generation step to generate the corresponding  $I^{syn}$  where each mask matches to a synthetic nucleus. To produce  $I^{mask}$  with multiple masks,  $N$  mask candidate volumes are generated. Each mask candidate volume contains one spherical mask which can be potentially included in  $I^{mask}$ . The  $i^{\text{th}}$  mask candidate volume,  $I^{mask,i}$ , is a binary volume where a mask is centered at a random location  $\mathbf{c}_i = (x_i, y_i, z_i)$  with a random radius,  $r_i$ :

$$I^{mask,i}(\mathbf{v}) = \begin{cases} 1, & \text{if } \|\mathbf{v} - \mathbf{c}_i\|_2^2 \leq r_i^2 \\ 0, & \text{otherwise} \end{cases} \quad (7.1)$$

where  $1 \leq i \leq N$ ,  $1 \leq x_i \leq X$ ,  $1 \leq y_i \leq Y$ ,  $1 \leq z_i \leq Z$ , and  $r_{min} \leq r_i \leq r_{max}$ , and  $\mathbf{v}$  denotes a voxel location.  $r_{min}$  and  $r_{max}$  are the minimum possible radius of a synthetic nucleus in a volume and the maximum possible radius of a synthetic nucleus in a volume, respectively.  $N$  can be selected based on the density of nuclei in  $I^{orig}$ , and  $r_{max}$  and  $r_{min}$  can be selected based on the size of nuclei in  $I^{orig}$ .

After mask candidate volumes are generated,  $I^{mask}$  is generated by adding  $I^{mask,i}$  in a sequential order. First,  $I^{mask}$  is initialized to zeros. Note nuclei cannot be overlapped physically in a biological structure. For  $1 \leq i \leq N$ , the  $I^{mask,i}$  is added to  $I^{mask}$  if  $I^{mask,i}$  has no overlap with other masks already added in  $I^{mask}$ . If  $I^{mask,i}$  is overlapped with one of the masks in  $I^{mask}$ ,  $I^{mask,i}$  is not included in  $I^{mask}$ . Once this process is done, then  $I^{mask}$  contains  $N_{mask}$  number of nuclei where  $N_{mask} \leq N$ . This approach is similar to the synthetic binary volume generation we presented in [192].

To generate  $I^{ctr}$  and  $I^{rad}$ , we first define a ‘‘central region’’. For the  $j^{\text{th}}$  nucleus where  $1 \leq j \leq N_{mask}$  whose center location is located at  $\mathbf{c}_j = (x_j, y_j, z_j)$  and radius is  $r_j$ , the  $j^{\text{th}}$  central region is defined within a sphere centered at the  $\mathbf{c}_j$  with a radius of

$ar_j$ .  $a$  is a parameter determining the range of the central region where  $0 \leq a \leq 1$ . In this work, we set  $a = 0.5$ . Within the  $j^{\text{th}}$  central region,  $I^{ctr,j}$  contains information of the location of the  $j^{\text{th}}$  nucleus in  $I^{mask}$  and  $I^{rad,j}$  contains information of the radius of the  $j^{\text{th}}$  nucleus in  $I^{mask}$ . Voxel values within the central region of  $I^{ctr,j}$  become 1 and of  $I^{rad,j}$  become  $r_j$  whereas other voxel values not in the central region of both  $I^{ctr,j}$  and  $I^{rad,j}$  become 0.

$$I^{ctr,j}(\mathbf{v}) = \begin{cases} 1, & \text{if } \|\mathbf{v} - \mathbf{c}_j\|_2^2 \leq (ar_j)^2 \\ 0, & \text{otherwise} \end{cases} \quad (7.2)$$

$$I^{rad,j}(\mathbf{v}) = \begin{cases} r_j, & \text{if } \|\mathbf{v} - \mathbf{c}_j\|_2^2 \leq (ar_j)^2 \\ 0, & \text{otherwise} \end{cases} \quad (7.3)$$

Then  $I^{ctr}$  and  $I^{rad}$  are the sum of  $I^{ctr,j}$  and  $I^{rad,j}$  for all  $j$ , respectively.

$$I^{ctr} = \sum_{j=1}^{N_{mask}} I^{ctr,j} \quad (7.4)$$

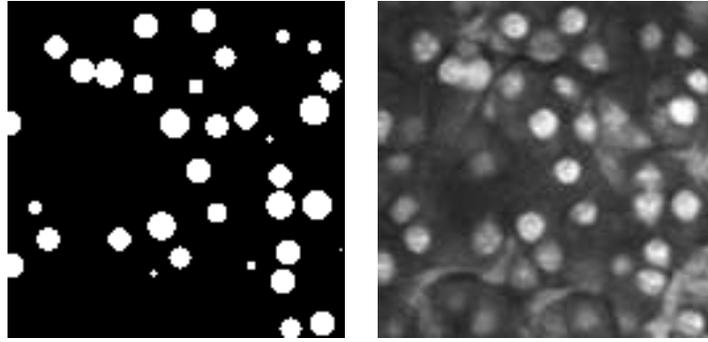
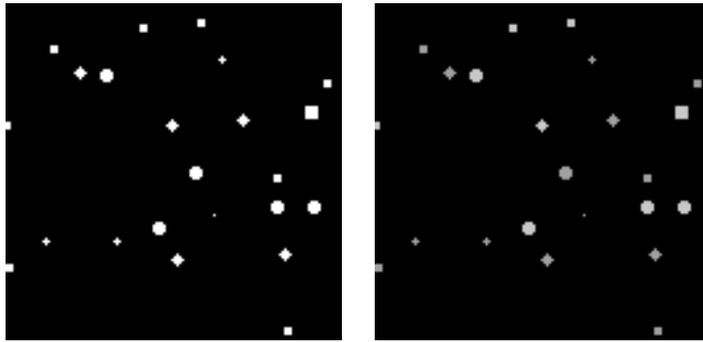
$$I^{rad} = \sum_{j=1}^{N_{mask}} I^{rad,j} \quad (7.5)$$

## Synthetic Microscopy Volume Generation

In the Synthetic Microscopy Volume Generation step,  $I^{mask}$  is used to generate the corresponding  $I^{syn}$  using the SpCycleGAN we described in [39]. Figure 7.2 shows an example of  $I^{mask}$ ,  $I^{syn}$ ,  $I^{ctr}$ , and  $I^{rad}$ .

### 7.1.2 Pre-Processing Steps

A pre-processing step can be used to  $I^{orig}$  before a SphEsNet-inference step as optional to improve detection result. We introduce three pre-processing steps: (1)  $z$ -directional interpolation, (2) 3D smoothing, and (3) clipping.

(a) a slice of  $I^{mask}$ (b) the corresponding slice of  $I^{syn}$ (c) the corresponding slice of  $I^{ctr}$ (d) the corresponding slice of  $I^{rad}$  (scaled by a factor of 40)Fig. 7.2.: Examples of  $I^{mask}$ ,  $I^{syn}$ ,  $I^{ctr}$ , and  $I^{rad}$ 

### $z$ -directional Interpolation

In this thesis, it is assumed that nuclei are spherical shape. If the resolution in  $z$ -direction is different with the resolutions in  $x$  and  $y$ -directions, nuclei may not look spherical. In order to make nuclei in spherical shape,  $z$ -directional interpolation needs to be done as a pre-processing step with the interpolation rate,  $r$ . If the number of images of the original volume,  $I^{orig}$ , in  $z$ -direction is  $Z$ , then the number of images after  $z$ -directional interpolation,  $I^{pre}$ , will be  $r \times (Z - 1) + 1$ .  $I_{z_p}^{orig}$  is the same as  $I_{z_{r \times (p-1)+1}}^{pre}$ . Images are interpolated linearly between  $I_{z_{r \times (p-1)+1}}^{pre}$  and  $I_{z_{r \times p+1}}^{pre}$ .

### 3D Smoothing

Fluorescence microscopy images are generally noisy which can cause some false nuclei detection. To remove noise, 3D smoothing is used as a pre-processing step with a 3D Gaussian filter with standard deviation of  $\sigma$ .

### Clipping

To remove noise in background, clipping can be used as a pre-processing step. Clipping can be done with a thresholding value,  $\tau$ :

$$I^{pre}(\mathbf{v}) = \begin{cases} I^{orig}(\mathbf{v}), & \text{if } I^{orig}(\mathbf{v}) > \tau \\ 0, & \text{otherwise} \end{cases} \quad (7.6)$$

#### 7.1.3 A 3D Convolutional Neural Network

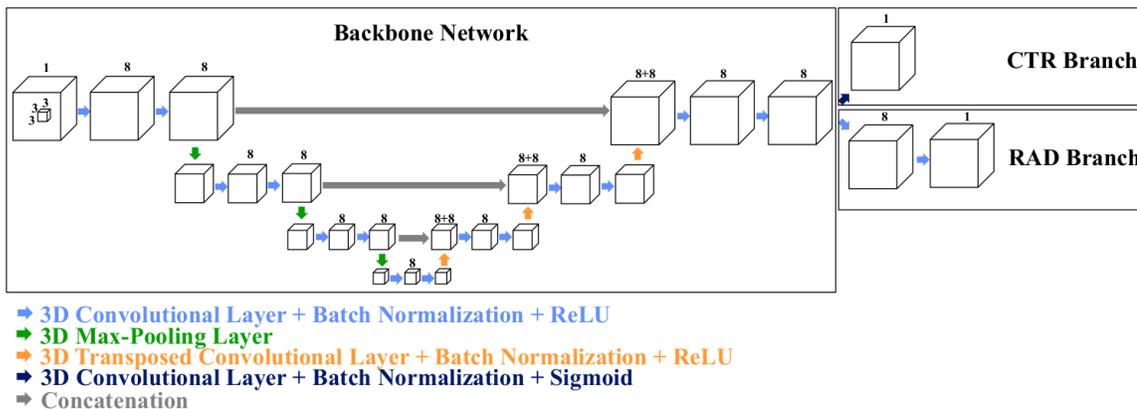


Fig. 7.3.: Architecture of the Sphere Estimation Network

Figure 7.3 shows the architecture of our Sphere Estimation Network. Our architecture consists of a backbone network and two branches. The first branch estimates center locations of nuclei and the second branch estimates radii of nuclei. We denote the first branch as the CTR branch and the second branch as the RAD branch. For the backbone network, we employ a modified 3D U-Net architecture [168]. Our

architecture uses the same number of convolutional layers, max-pooling layers, and transposed convolution layers in 3D U-Net, but the number of channels are reduced. A 3D convolutional layer consist of a convolutional operation with a  $3 \times 3 \times 3$  kernel with 1 voxel padding to maintain the same volume size, followed by 3D batch normalization [191] and a rectified-linear unit (ReLU) activation function. In the encoder part, a 3D max-pooling layer uses  $2 \times 2 \times 2$  window with a stride of 2. In the decoder part, a 3D transposed convolution layer is used in the upsampling path. Between the encoder part and the decoder part, shortcut connections transfer feature maps to preserve spatial information. For the CTR branch, one convolutional layer is used. The feature map intensities are between 0 and 1 to classify each voxel to center location or not, so the Sigmoid function is used as an activation function. For the RAD branch, two convolutional layers are used. The feature map intensities are greater than 0 so the ReLU function is used as an activation function. More details on train and inference are provided below.

## Training

During training,  $I^{syn}$ ,  $I^{ctr}$ , and  $I^{rad}$  generated from the Synthetic Volume Generation stage are used as an input volume, a ground truth volume in the CTR branch, and a ground truth volume in the RAD branch, respectively. Note that 20 sets of volumes with size of  $128 \times 128 \times 128$  are generated in the Synthetic Volume Generation stage. We divide each volume to 8 volumes with size of  $64 \times 64 \times 64$ , so 160 sets of volumes are used to train the model,  $M$ . Our training loss function,  $L$ , is a linear combination of the Binary Cross Entropy (BCE) loss,  $L^{BCE}$ , at the CTR branch and the Mean Squared Error (MSE) loss,  $L^{MSE}$ , at the RAD branch.

$$L(I^{ctr}, I^{rad}, I^{ctr,out}, I^{rad,out}) = L^{BCE}(I^{ctr}, I^{ctr,out}) + \lambda L^{MSE}(I^{rad}, I^{rad,out}) \quad (7.7)$$

where  $I^{ctr}$  is the ground truth volume in the CTR branch,  $I^{rad}$  is the ground truth volume in the RAD branch,  $I^{ctr,out}$  is the output volume in the CTR branch, and  $I^{rad,out}$  is the output volume in the RAD branch, respectively.  $\lambda$  is a weight coefficient

which can emphasize one loss term over the other loss term. For example, if  $\lambda$  is small, then the BCE loss will be more emphasized. Also, if  $\lambda$  is big, then the MSE loss will be more emphasized. In this work, we set  $\lambda = 1$  unless specified. More specifically, the BCE loss is defined as

$$L^{BCE}(I^{ctr}, I^{ctr,out}) = -\frac{1}{V} \sum_{\mathbf{v}=1}^V (I^{ctr}(\mathbf{v}) \log I^{ctr,out}(\mathbf{v}) + (1 - I^{ctr}(\mathbf{v})) \log(1 - I^{ctr,out}(\mathbf{v}))) \quad (7.8)$$

and the MSE loss is defined as

$$L^{MSE}(I^{rad}, I^{rad,out}) = \frac{1}{V} \sum_{\mathbf{v}=1}^V (I^{rad}(\mathbf{v}) - I^{rad,out}(\mathbf{v}))^2 \quad (7.9)$$

where  $V$  is the number of voxels in a volume. To train our model, the Adam optimizer [100] is used with a learning rate of 0.001. The training of our SphEsNet took approximately 2 days using NVIDIA’s GeForce GTX Titan X.

## Inference

SphEsNet is trained on volumes with size of  $64 \times 64 \times 64$  so an input volume to the model should be  $64 \times 64 \times 64$ . If an input microscopy volume is bigger than  $64 \times 64 \times 64$  then our model can only process a subvolume with size of  $64 \times 64 \times 64$ . To process the entire  $I^{orig}$ , a 3D window is slid with size of  $64 \times 64 \times 64$ , similarly done as [192]. First, the input microscopy volume is reflection-padded by 16 voxels. Then we slide the 3D window to  $x$ ,  $y$ , and  $z$ -direction by 32. Note that nuclei which are partially included on the boundary of the 3D window may generate inaccurate result. Therefore, only the central volume with size of  $32 \times 32 \times 32$  are used. We repeat this until the entire input microscopy volume is processed.

Once the process is done, then we have two output volumes,  $I^{ctr,out}$  and  $I^{rad,out}$ . Note  $I^{ctr,out}$  is the output volume from the CTR branch and  $I^{rad,out}$  is the output volume from the RAD branch. First of all, local maxima of  $I^{ctr,out}$ ,  $\mathbf{c}_k$ , are selected as the final center coordinate of the  $k^{\text{th}}$  nucleus. We avoid selecting more than two center coordinates within a cube with length of  $r_{min}$  because we set the smallest radius of

nuclei in a volume is  $r_{min}$ . Sometimes we observe that multiple local maxima with the same intensity are connected as a 3D region. In this case, a centroid of the 3D region is selected as the final center coordinate. After selecting center coordinates of nuclei, the radius for the  $k^{\text{th}}$  nucleus is estimated as  $I^{rad,out}(\mathbf{c}_k) + 1$  to generate a sphere centered at  $\mathbf{c}_k$ . We observed that the real nuclei in  $I^{orig}$  are not perfectly spherical and the radius is estimated to the shortest distance from the estimated center to the boundary of nuclei. So we increment the estimated radius,  $I^{rad,out}(\mathbf{c}_k)$ , by 1 to increase the overlapping region between the estimated sphere and the nucleus. As a final step, any other center locations within a sphere are removed because a center location of a nucleus cannot physically be in another nucleus. The final detection volume,  $I^{det}$ , is generated after color-labeling the spheres. The inference of SphEsNet on a volume with size of  $512 \times 512 \times 512$  took approximately 3~5 minutes using NVIDIA’s GeForce GTX Titan X.

## 7.2 Experimental Results

We tested our method on five different real fluorescence microscopy data sets: Data-N1, Data-N5, Data-N8, Data-N4, and Data-N9. In this work, we trained three models,  $M^{Data-N1}$ ,  $M^{Data-N5}$ , and  $M^{Data-N8}$ .  $M^{Data-N1}$  was trained by synthetic Data-N1 generated with  $r_{min} = 4$ ,  $r_{max} = 6$ , and  $N = 1000$ .  $M^{Data-N5}$  was trained by synthetic Data-N5 generated with  $r_{min} = 8$ ,  $r_{max} = 10$ , and  $N = 50$ .  $M^{Data-N8}$  was trained by synthetic Data-N8 generated with  $r_{min} = 8$ ,  $r_{max} = 12$ , and  $N = 200$ .  $r_{min}$ ,  $r_{max}$ , and  $N$  are selected based on the characteristics of the data sets. We observed that Data-N8 contains heavy noise so we selected  $\lambda = 0.1$  to emphasize more on nuclei location to avoid false detection. Data-N1 was tested on  $M^{Data-N1}$ , Data-N5 on  $M^{Data-N5}$ , and Data-N8 on  $M^{Data-N8}$ , respectively. In addition, Data-N4 was tested on  $M^{Data-N1}$  and Data-N9 was tested on  $M^{Data-N5}$  because the size of nuclei in Data-N4 is similar to the size of nuclei in Data-N1 and the size of nuclei in Data-N9 is similar to the size of nuclei in Data-N5. Pre-processing steps are used on

some data sets to improve detection results.  $z$ -directional interpolation with  $r = 2$  was used on Data-N5 to make nuclei in Data-N5 in spherical shapes. A 3D smoothing with  $\sigma = 2$  was used on Data-N8 because Data-N8 contains heavy noise. Lastly, clipping with  $\tau = 5$  was used on Data-N9 to remove noise in the background region. Figure 7.4, Figure 7.5, Figure 7.6, Figure 7.7, and Figure 7.8 show that our method can successfully detect nuclei on various data sets.

Our method was compared to other detection methods. To evaluate detection results, we use an object-level quantification. A detected nucleus is counted as true positive (TP) if it intersects at least 50% of the corresponding ground truth nucleus. Otherwise, it is counted as false positive (FP). A ground truth nucleus is counted as false negative (FN) if it intersects less than 50% of the corresponding detected nucleus or there is no corresponding detected nucleus.  $N_{TP}$ ,  $N_{FP}$ , and  $N_{FN}$  are denoted as the number of true positive, the number of false positive, and the number of false negative, respectively. Then Precision ( $P$ ), Recall ( $R$ ), and F1 score ( $F1$ ) are defined as the followings [186, 187]:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (7.10)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (7.11)$$

$$F1 = \frac{2PR}{P + R} \quad (7.12)$$

In our evaluation, we generated a 3D ground truth volume,  $I_{(193:320,193:320,31:94)}^{gt}$ , using ITK-SNAP [194] from Data-I with size of  $128 \times 128 \times 64$  containing 283 nuclei. Note any components whose number of voxels is less than 50 are removed on  $I_{(193:320,193:320,31:94)}^{det}$  and  $I_{(193:320,193:320,31:94)}^{gt}$  to remove partially included nuclei on the boundary of the subvolume during the evaluation.

Watershed [36] is generally used to separate touching nuclei in microscopy volumes. We used watershed on an original volume binarized by a manually-selected threshold value of 64. Due to inhomogeneous intensity presented in microscopy volumes, we observe that noise in the central region is detected, causing low precision. To correct inhomogeneity, a 3D adaptive histogram equalization [38] followed by binarization

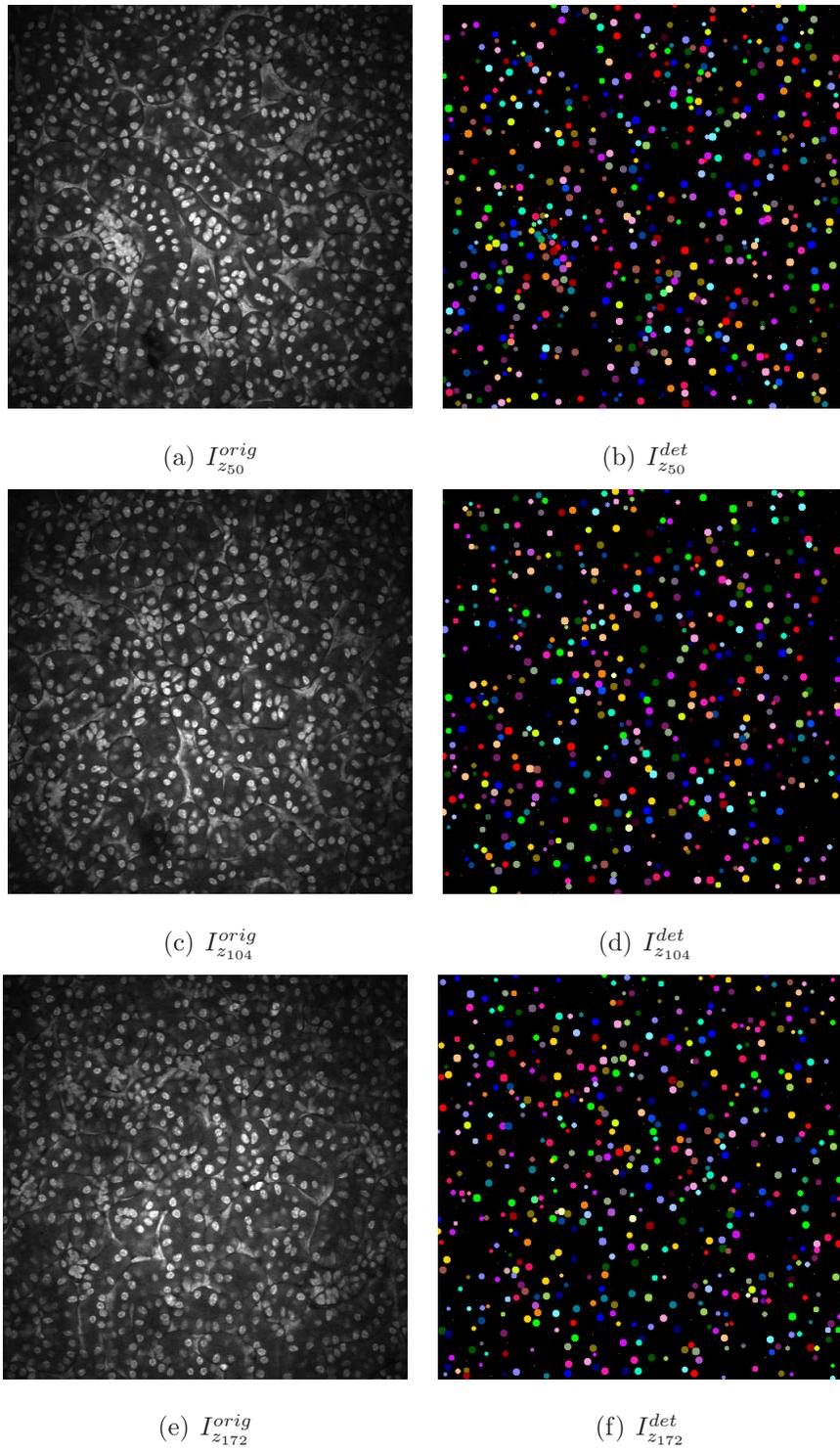


Fig. 7.4.: Original and detected images of Data-N1

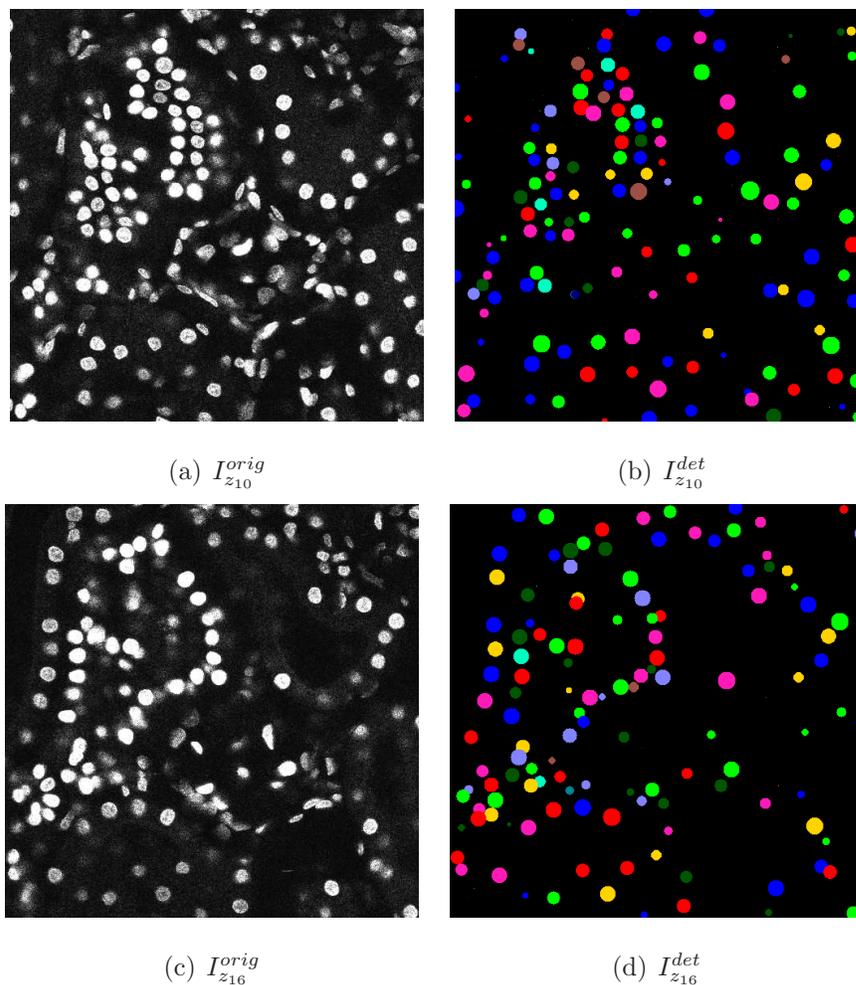


Fig. 7.5.: Original and detected images of Data-N5

with a manually-selected threshold value of 192 is used as a pre-processing step before watershed technique. We will denote the method of using a 3D adaptive histogram equalization and watershed technique as Watershed<sup>+</sup>. The 3D adaptive histogram equalization help removing noise in the microscopy volume but non-nuclei subcellular structures are still captured. Over-detection from the watershed reduces precision. A mask R-CNN [136] is a 2D segmentation method which can both detect and segment objects in distinct labels. A pre-trained Mask R-CNN with ResNet-50 [109] and a Feature Pyramid Network (FPN) [129] is fine-tuned using the same training images used for the SphEsNet. To combine 2D segmentation results into 3D, we use a

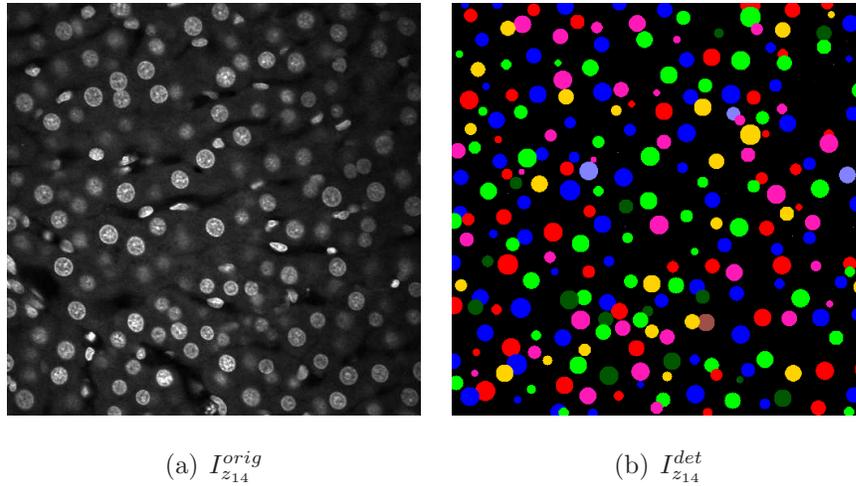


Fig. 7.6.: Original and detected images of Data-N8

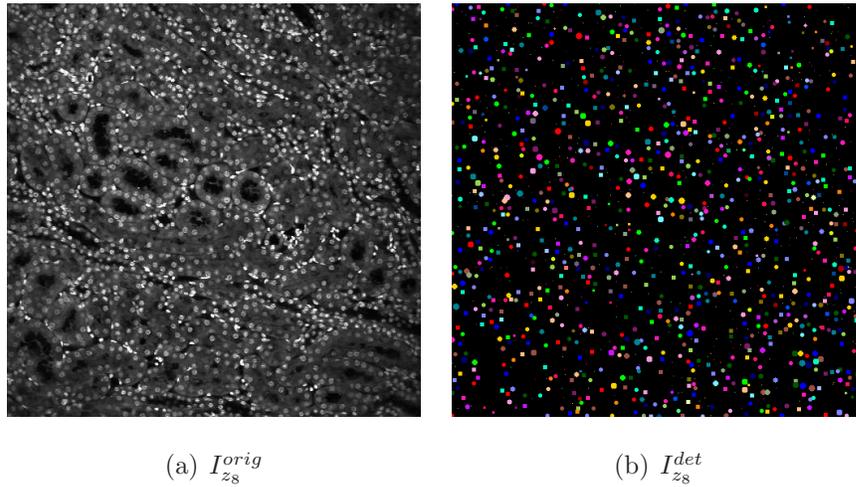


Fig. 7.7.: Original and detected images of Data-N4

$z$ -directional combination as a post-processing step. The  $m^{\text{th}}$  mask on  $I_{z_p}$  whose centroid located at  $(x_{m,p}, y_{m,p})$  and the  $n^{\text{th}}$  mask on  $I_{z_{p+1}}$  whose centroid located at  $(x_{n,p+1}, y_{n,p+1})$  are combined as the same nucleus if the distance between  $(x_{m,p}, y_{m,p})$  and  $(x_{n,p+1}, y_{n,p+1})$  is less than  $\frac{r_{max}}{2}$ . We will denote this method as Mask R-CNN $_z$ . A 2D Mask R-CNN cannot utilize depth information, so it performs poor segmentation. In addition, we observed the Mask R-CNN was overfitted. Alternatively, we developed

a two-stage method for 3D nuclei detection-segmentation [193]. We will denote this method as Det-Seg. The 3D segmentation CNN was trained by a set of synthetic volumes generated by blurring and noise operations. The synthetic volumes used in [193] were not realistic so segmentation masks were not accurate causing low precision. More recently, we generated an accurate binary segmentation mask using a 3D CNN trained by a set of synthetic volumes generated by a SpCycleGAN we described in [39]. To separate touching nuclei in the binary segmentation mask, morphological operations are used as a post-processing step. We used a 3D erosion, a 3D connected component (for color-coding), and a 3D dilation to separate touching nuclei. A ball with radius of 1 is used as a structuring element for the 3D erosion and the 3D dilation. We will denote this method as Seg-Morph. We observe that some nuclei are still not separated by morphological operations leading to low recall. Our method detects the locations of nuclei with their radii accurately so over-detection issues or touching nuclei issues can be avoided. Table 7.1 shows our method outperforms numerically. Figure 7.9 shows the original image and the segmentation results. The original volume, the ground truth volume, and 3D results are visualized using Voxx [185] shown in Figure 7.10.

Table 7.1.: Precision, Recall, and F1 score for various methods on  $I_{(193:320,193:320,31:94)}$  of Data-N1

	Precision	Recall	F1 score
Watershed [36]	51.14%	92.13%	65.78%
Watershed <sup>+</sup> [36, 38]	41.31%	97.17%	57.97%
Mask R-CNN <sub>z</sub> [136]	20.85%	39.20%	27.22%
Det-Seg [193]	68.35%	90.22%	77.78%
Seg-Morph [39]	91.20%	82.01%	86.36%
Proposed Method	84.04%	96.63%	89.90%

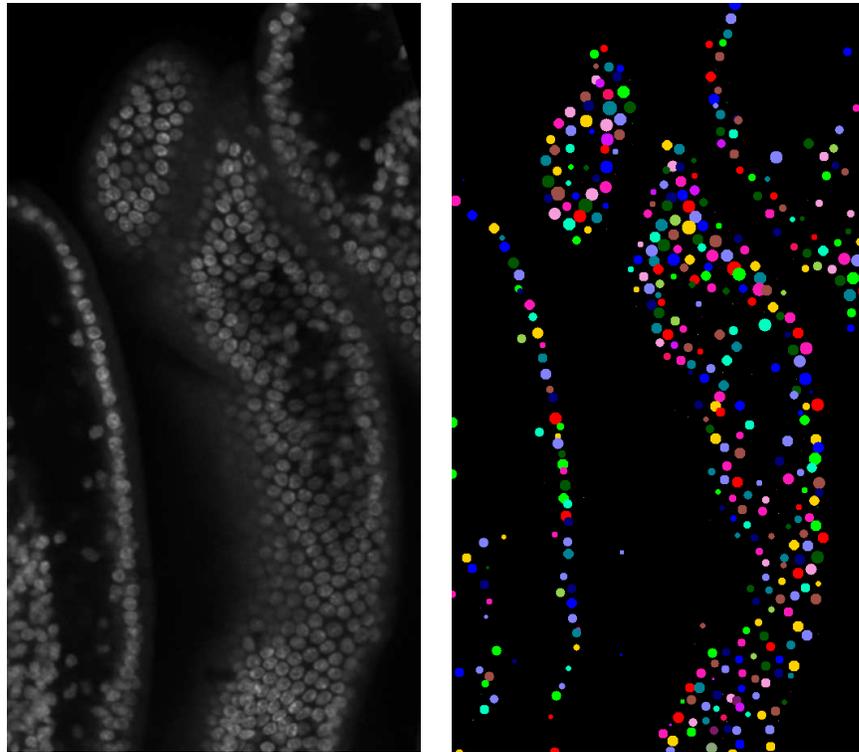
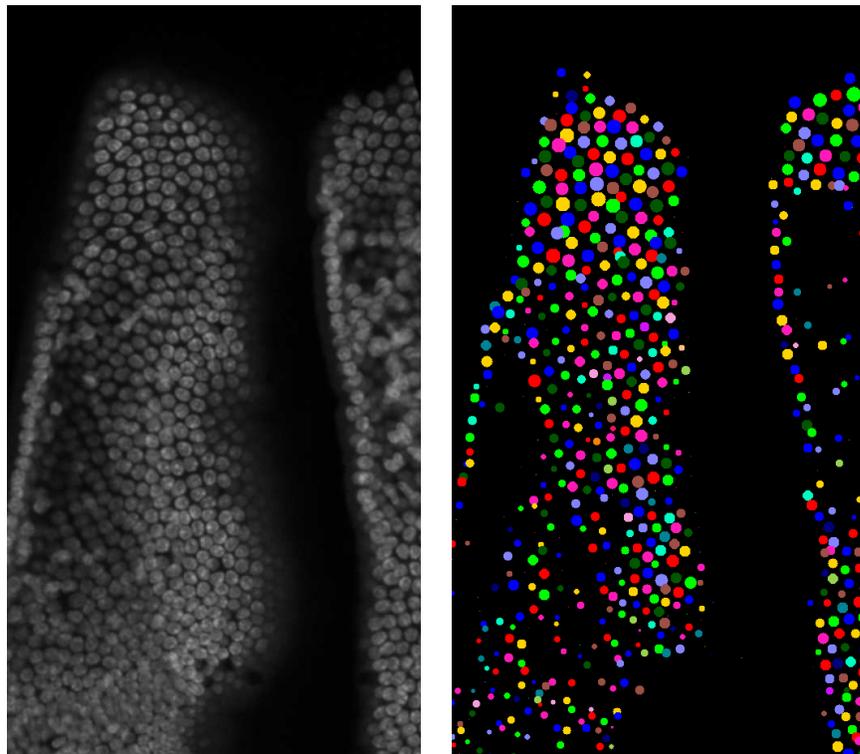
(a)  $I_{z_{31}}^{orig}$ (b)  $I_{z_{31}}^{det}$ (c)  $I_{z_{137}}^{orig}$ (d)  $I_{z_{137}}^{det}$ 

Fig. 7.8.: Original and detected images of Data-N9

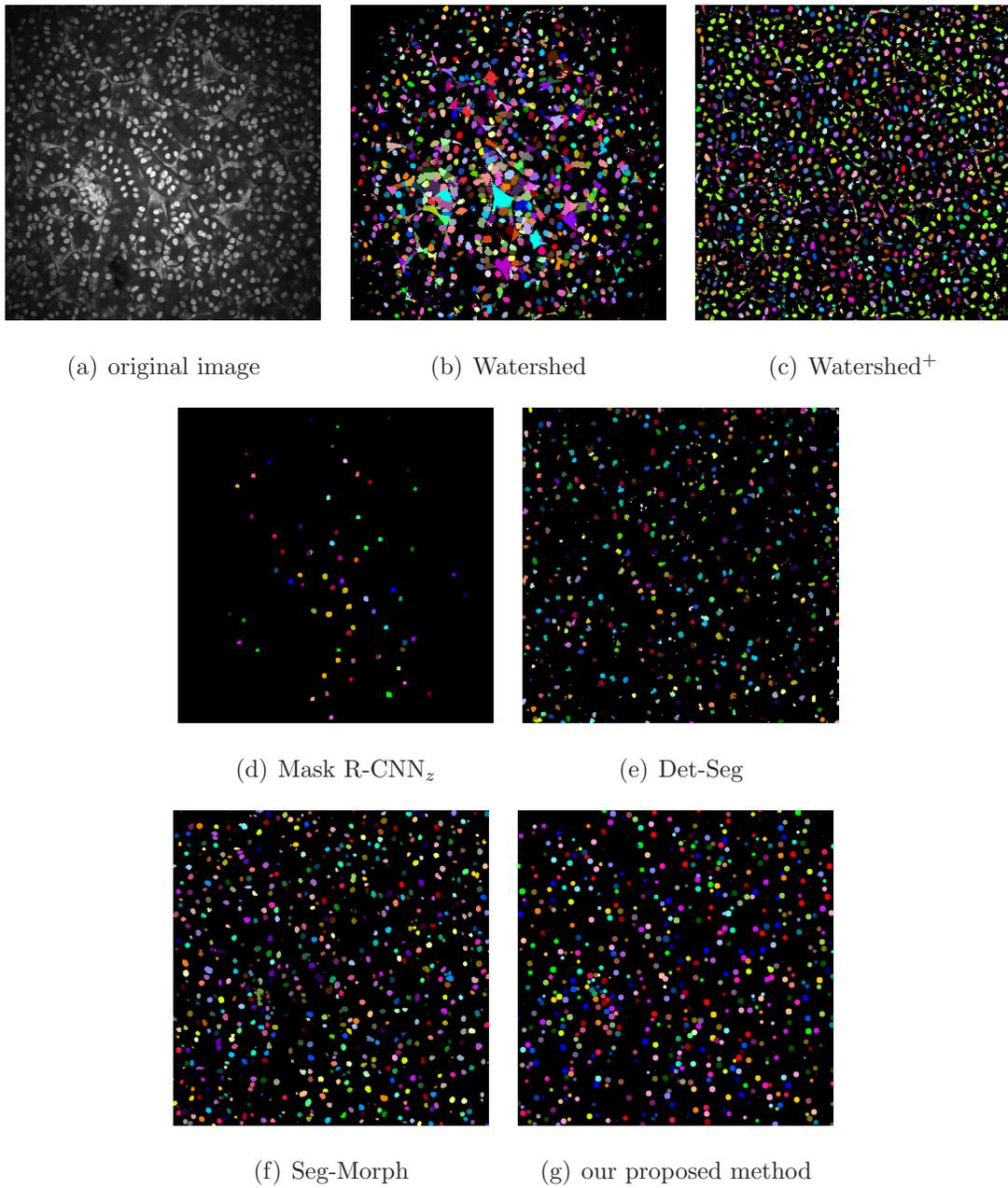


Fig. 7.9.: Comparison of other detection methods and our proposed method of  $I_{z50}$  of Data-N1

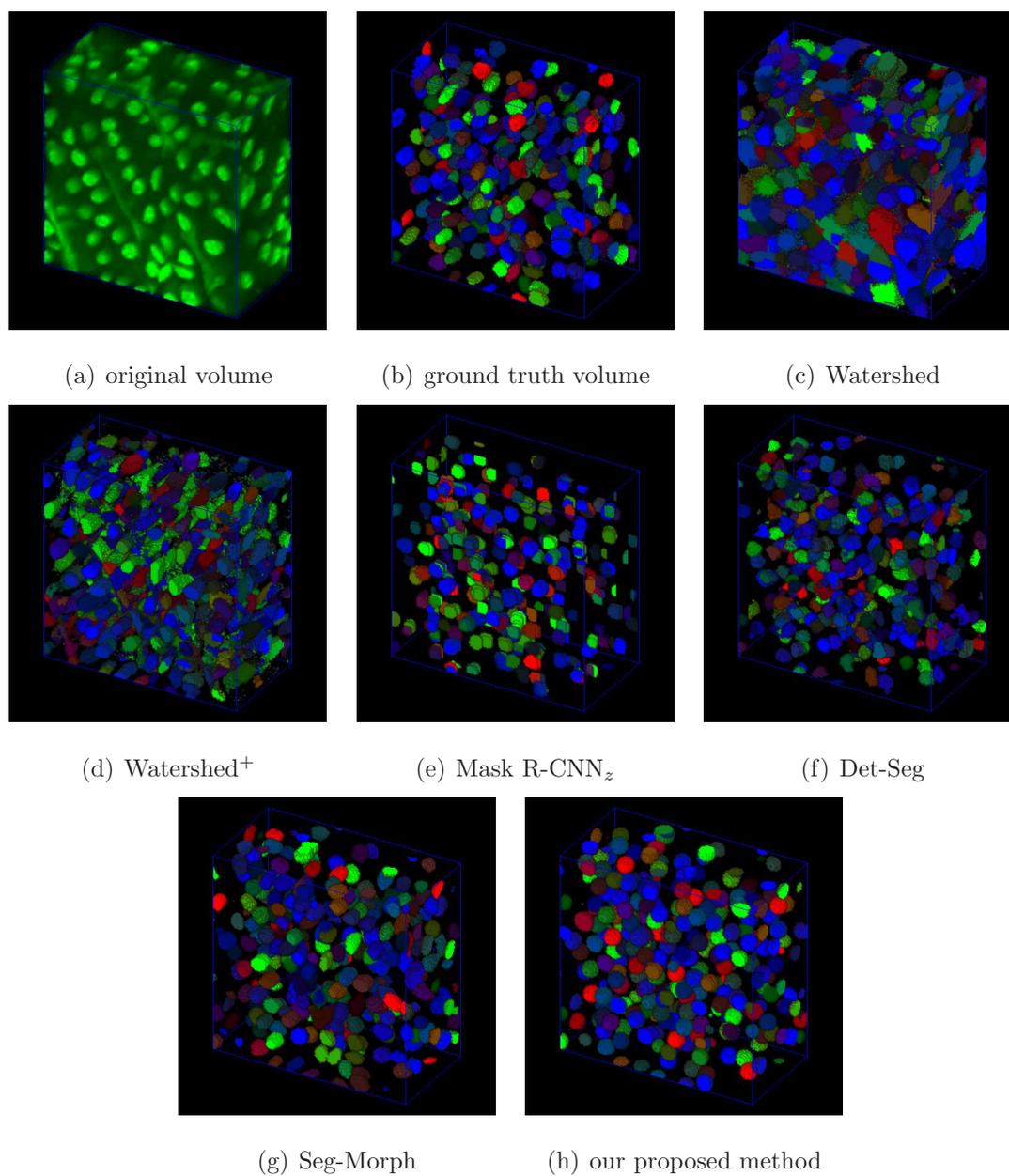


Fig. 7.10.: 3D visualization comparison of other detection methods and our proposed method of  $I_{(193:320,193:320,31:94)}$  of Data-N1

## 8. COLOR LABELING

We want to distinctly label nuclei with different colors for visualizing the segmentation results. Therefore, selecting proper colors for each nucleus is important. For example, we do not want to label adjacent nuclei using the same color or colors that human eye cannot distinguish immediately. In this chapter, we describe the color labeling method we developed.

### 8.1 Proposed Color Labeling Method

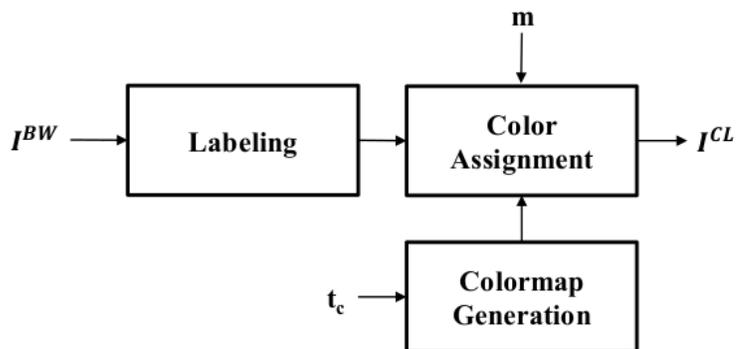


Fig. 8.1.: Block diagram of our color labeling method

Figure 8.1 is a block diagram of our color labeling method. The input is a binary image/mask after segmentation,  $I^{BW}$ , and the output is a color labeled image,  $I^{CL}$ . Nuclei in  $I^{BW}$  are labeled distinctly using 3D connected component with morphological operations [40] to separate touching nuclei. Also, we generate a colormap we use in color labeling. Lastly, colors are assigned to the labels on nuclei to produce  $I^{CL}$ . The details are described below.

### 8.1.1 Labeling

The goal of this step is to label nuclei distinctly in a binary image,  $I^{BW}$ . A simple 3D connected component [40] may not label successfully especially when two or more nuclei are touching. 3D watershed technique [36] tends to over-segment when nuclei are in irregular shapes. Therefore, we use 3D connected component with morphological operations [40] to label nuclei distinctly with separating touching nuclei.

To separate touching nuclei, we first use a 3D morphological erosion with a sphere of radius 1 as the structuring element. For the structuring element of the 3D erosion, we select the radius of a sphere to be 1 because we observe that the nuclei structures are not preserved or disappeared when the radius becomes too big. After the erosion, we use a 3D connected component with 6 point neighborhood to label each component distinctly. Finally, we want to recover nuclei structures by using a 3D morphological dilation with a sphere of radius 1 as the structuring element. Using this labeling technique, we can label nuclei distinctly even when some of them are touching.

### 8.1.2 Colormap Generation



Fig. 8.2.: The first 100 colors of our colormap

As mentioned above, we do not want to label adjacent nuclei using similar colors. To achieve this, we generate a colormap which is used in the color assignment step. We denote  $c_i$  to be an RGB coordinate for the  $i^{\text{th}}$  color. The first color is set to be blue where  $c_1 = (0, 0, 255)$ . For  $i \geq 2$ ,  $c_i$  is selected as the farthest distance from all previous colors in the *Lab* color space which is known to match human vision closely [195]. Note the background color,  $c_b$ , of  $I^{CL}$  is set to be black where

$c_b = (0, 0, 0)$ . To avoid selecting a color close to black, we remove any colors from the colormap whose  $R$ ,  $G$ , or  $B$  value is less than  $t_c$  where  $t_c$  is set to be 50 in our work. Figure 8.2 shows the first 100 colors of our colormap.

### 8.1.3 Color Assignment

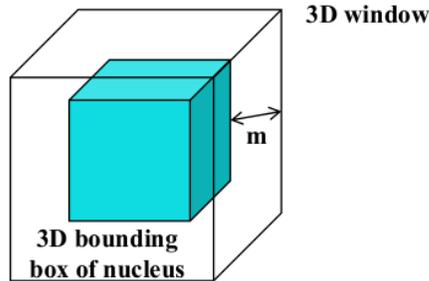


Fig. 8.3.: 3D window used for color assignment

The last step is to assign colors using our colormap to the labeled image. As mentioned above, we do not want to use the same color for adjacent nuclei. To make sure that the same color is not assigned to two or more nuclei within a certain distance, we use a 3D window. For each nucleus, we first find a 3D bounding box fitting the nucleus. Then we find a 3D window from the 3D bounding box with a margin of  $m$  as shown in Figure 8.3. If the 3D bounding box is  $I_{(x_1:x_2, y_1:y_2, z_1:z_2)}$ , then the 3D window becomes  $I_{(x_1-m:x_2+m, y_1-m:y_2+m, z_1-m:z_2+m)}$ . We set  $m$  to be 40 voxels. Lastly, we find the smallest  $j$  where  $c_j$  is not presented within the 3D window. The color of the nucleus is selected to be  $c_j$ . This procedure is repeated until color is assigned to all nuclei in the volume. Using this color assigning method, we can guarantee that the same color is used at least  $m$  voxels away.

## 8.2 Experimental Results

Figure 8.4 shows our color labeling result. The binary segmentation image is generated using our method described in [39]. Our color labeling method can successfully separate touching nuclei in 3D and label nuclei in distinguishable colors visually.

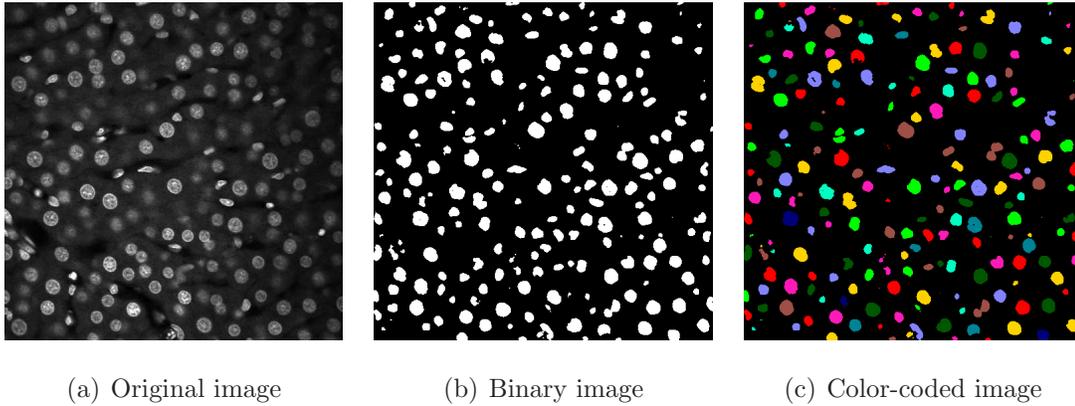


Fig. 8.4.: Color-Labeling result

## 9. CONCLUSIONS

### 9.1 Summary

In this thesis we describe multiple methods to analyze fluorescence microscopy images. Especially, this thesis focuses on segmentation and detection of subcellular structures presented in three dimensional fluorescence microscopy images. The main contributions of this thesis are listed as follows:

- Boundary Segmentation Using Steerable Filters

We describe a method to segment tubular boundary of biological structures. To segment foreground in fluorescence microscopy images having non-uniform intensity a 3D adaptive histogram equalization is used to correct inhomogeneity. The main challenge of this work is to segment only tubule structures while other structures such as lumen are presented in fluorescence microscopy images. To distinguish tubule in foreground from lumen, steerable filters [37] are used because steerable filters can capture strong directional tendencies on tubule. Our method can successfully segment tubule boundaries in various data sets from rat kidney and liver.

- Nuclei Segmentation Using Convolutional Neural Networks

We present a nuclei segmentation method of fluorescence microscopy images. We design and train three dimensional convolutional neural networks (CNNs). A large set of labeled training volumes is necessary to train CNNs. The labeling process especially in 3D is tedious. In this work we generate synthetic microscopy volumes to train our CNNs. We assume nuclei are in ellipsoidal shapes and we add a blurring operation modeled from a PSF and a noise operation which is a mixture of a Poisson noise and a Gaussian noise. Our 3D

CNNs trained by synthetic microscopy volumes can successfully segment nuclei in real microscopy volumes without using any real ground truth volumes.

- Nuclei Detection and Segmentation Using Convolutional Neural Networks

We introduce a nuclei detection and segmentation method to label nuclei distinctly. To detect center locations of nuclei a 3D adaptive histogram equalization [38], a 3D distance transform, and a 3D classification CNN are used. After the detection stage, each nucleus is distinctly segmented using a segmentation CNN in a 3D patch surrounding the nucleus. The segmentation CNN is trained by a set of synthetic microscopy volumes. Due to the detection stage, non-nuclei structures presented in real fluorescence microscopy volumes are rejected which improves our segmentation accuracy.

- Center-Extraction-Based Nuclei Instance Segmentation

We describe a two-stage method of segmenting nuclei in distinct labels. Our first CNN detects the center locations of nuclei and generates a binary segmentation mask. Using the nuclei center locations and the binary segmentation mask from the first CNN, our second CNN individually segments nuclei in 3D patches surrounding each nucleus. To train our CNNs, realistic synthetic volumes generated by a spatially constrained cycle-consistent adversarial network (SpCycleGAN) [39] are used without using any real ground truth volumes. Our method can detect and segment nuclei in real microscopy volumes accurately.

- Nuclei Detection Using Sphere Estimation Network

We develop a Sphere Estimation Network (SphEsNet) which can detect nuclei in fluorescence microscopy volumes. Our network is a fully three dimensional CNN. In this work we assume nuclei presented in fluorescence microscopy volumes are spherical. Our network not only finds the locations of nuclei but also estimates the radii of nuclei. The SphEsNet is trained by realistic synthetic microscopy volumes using the SpCycleGAN [39] without using any real ground

truth volumes. Our evaluation shows that the spheres at the estimated locations with the estimated radii generated by the SphEsNet highly overlap with nuclei presented in multiple real microscopy data sets.

## 9.2 Future Work

Our methods can be improved and extended in the following ways:

- Tubule Segmentation

In our work we segmented tubular boundary of biological structures such as a nephron in rat kidney or blood vessels in liver. We were able to successfully distinguish tubular structures from other subcellular structures presented in fluorescence microscopy volumes. Tubule segmentation is a prerequisite step to quantify the biological structures. Using our segmentation results we can characterize the biological structures such as volumes.

- Nuclei Segmentation

In our work we initially segmented nuclei and rejected other non-nuclei structures. Additionally, we labeled nuclei distinctly. To train our CNNs, the SpCycleGAN [39] produces realistic synthetic microscopy volumes without using any real ground truth volumes. We plan to improve the SpCycleGAN to produce multiple types of synthetic nuclei in various sizes and shapes. We expect the improvement of the SpCycleGAN can enable segmenting multiple types of nuclei presented in real microscopy volumes.

- Nuclei Detection

Nuclei detection is an important for analyzing nuclei in fluorescence microscopy images. In our work we not only find locations of nuclei but also estimate radii of nuclei using a CNN with two branches. We plan to design a multi-branch CNN which can obtain more information of nuclei such as types or textures.

### 9.3 Publications Results From The Thesis

#### Journal Papers

1. **D. J. Ho**, C. Fu, D. Mas Montserrat, P. Salama, K. W. Dunn and E. J. Delp, “Sphere Estimation Network: Three Dimensional Nuclei Detection of Fluorescence Microscopy Images”, *To be submitted to the IEEE Transactions on Medical Imaging*.
2. C. Fu, S. Han, S. Lee, **D. J. Ho**, P. Salama, K. W. Dunn and E. J. Delp, “Three Dimensional Nuclei Synthesis and Instance Segmentation”, *To be submitted to the IEEE Transactions on Medical Imaging*.

#### Conference Papers

1. **D. J. Ho**, S. Han, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Center-Extraction-Based Three Dimensional Nuclei Instance Segmentation of Fluorescence Microscopy Images”, Submitted to *Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics*, May 2019, Chicago, IL.
2. C. Fu, S. Lee, **D. J. Ho**, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, “Three Dimensional Fluorescence Microscopy Image Synthesis and Segmentation,” *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 2302-2310, June 2018, Salt Lake City, UT.
3. **D. J. Ho**, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei Detection and Segmentation of Fluorescence Microscopy Images Using Three Dimensional Convolutional Neural Networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 418-422, April 2018, Washington D.C.
4. **D. J. Ho**, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei Segmentation of Fluorescence Microscopy Images Using Three Dimensional Convolutional

- Neural Networks,” *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 834-842, July 2017, Honolulu, HI.
5. C. Fu, **D. J. Ho**, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei Segmentation of Fluorescence Microscopy Images Using Convolutional Neural Networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 704-708, April 2017, Melbourne, Australia.
  6. **D. J. Ho**, P. Salama, K. W. Dunn, and E. J. Delp, “Boundary Segmentation for Fluorescence Microscopy Using Steerable Filters,” *Proceedings of the SPIE Conference on Medical Imaging*, pp. 101330E-1-11, February 2017, Orlando, FL.

## REFERENCES

## REFERENCES

- [1] D. B. Murphy and M. W. Davidson, *Fundamentals of Light Microscopy and Electronic Imaging*. Wiley-Blackwell, 2012.
- [2] U. Kubitschek, *Fluorescence Microscopy: From Principles to Biological Applications*. Wiley-Blackwell, 2013.
- [3] D. B. Williams and C. B. Carter, *Transmission Electron Microscopy*. Springer, 2009.
- [4] C. Vonesch, F. Aguet, J. Vonesch, and M. Unser, “The colored revolution of bioimaging,” *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 20–31, May 2006. [Online]. Available: <https://doi.org/10.1109/MSP.2006.1628875>
- [5] J. W. Lichtman and J.-A. Conchello, “Fluorescence microscopy,” *Nature Methods*, vol. 2, no. 12, pp. 910–919, December 2005. [Online]. Available: <https://doi.org/10.1038/NMETH817>
- [6] M. Chalfie, Y. Tu, G. Euskirchen, W. Ward, and D. C. Prasher, “Green fluorescent protein as a marker for gene expression,” *Science*, vol. 263, no. 5148, pp. 802–805, February 1994. [Online]. Available: <https://doi.org/10.1126/science.8303295>
- [7] T. Stearns, “Green fluorescent protein: The green revolution,” *Current Biology*, vol. 5, no. 3, pp. 262–264, March 1995. [Online]. Available: [https://doi.org/10.1016/S0960-9822\(95\)00056-X](https://doi.org/10.1016/S0960-9822(95)00056-X)
- [8] O. Shimomura, “The discovery of aequorin and green fluorescent protein,” *Journal of Microscopy*, vol. 217, no. 1, pp. 3–15, January 2005. [Online]. Available: <https://doi.org/10.1111/j.0022-2720.2005.01441.x>
- [9] M. Minsky, “Memoir on inventing the confocal scanning microscope,” *Scanning*, vol. 10, no. 4, pp. 128–138, 1988. [Online]. Available: <https://doi.org/10.1002/sca.4950100403>
- [10] E. Wang, C. M. Babbey, and K. W. Dunn, “Performance comparison between the high-speed Yokogawa spinning disc confocal system and single-point scanning confocal systems,” *Journal of Microscopy*, vol. 218, no. 2, pp. 148–159, May 2005. [Online]. Available: <https://doi.org/10.1111/j.1365-2818.2005.01473.x>
- [11] W. Denk, J. H. Strickler, and W. W. Webb, “Two-photon laser scanning fluorescence microscopy,” *Science*, vol. 248, no. 4951, pp. 73–76, April 1990. [Online]. Available: <https://doi.org/10.1126/science.2321027>

- [12] D. W. Piston, “Imaging living cells and tissues by two-photon excitation microscopy,” *Trends in Cell Biology*, vol. 9, no. 2, pp. 66–69, February 1999. [Online]. Available: [https://doi.org/10.1016/S0962-8924\(98\)01432-9](https://doi.org/10.1016/S0962-8924(98)01432-9)
- [13] K. W. Dunn, R. M. Sandoval, K. J. Kelly, P. C. Dagher, G. A. Tanner, S. J. Atkinson, R. L. Bacallao, and B. A. Molitoris, “Functional studies of the kidney of living animals using multicolor two-photon microscopy,” *American Journal of Physiology-Cell Physiology*, vol. 283, no. 3, pp. C905–C916, September 2002. [Online]. Available: <https://doi.org/10.1152/ajpcell.00159.2002>
- [14] F. Helmchen and W. Denk, “Deep tissue two-photon microscopy,” *Nature Methods*, vol. 2, no. 12, pp. 932–940, December 2005. [Online]. Available: <https://doi.org/10.1038/NMETH818>
- [15] E. H. Hoover and J. A. Squier, “Advances in multiphoton microscopy technology,” *Nature Photonics*, vol. 7, pp. 93–101, February 2013. [Online]. Available: <https://doi.org/10.1038/NPHOTON.2013.361>
- [16] G. M. P. van Kempen, L. J. van Vliet, P. J. Verveer, and H. T. M. van der Voort, “A quantitative comparison of image restoration methods for confocal microscopy,” *Journal of Microscopy*, vol. 185, no. 3, pp. 354–365, March 1997. [Online]. Available: <https://doi.org/10.1046/j.1365-2818.1997.d01-629.x>
- [17] P. Sarder and A. Nehorai, “Deconvolution methods for 3-D fluorescence microscopy images,” *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 32–45, May 2006. [Online]. Available: <https://doi.org/10.1109/MSP.2006.1628876>
- [18] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J.-C. Olivo-Marin, and C. Zimmer, “Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces,” *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396–1410, September 2005. [Online]. Available: <https://doi.org/10.1109/TIP.2005.852790>
- [19] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “NIH Image to ImageJ: 25 years of image analysis,” *Nature Methods*, vol. 9, no. 7, pp. 671–675, July 2012. [Online]. Available: <https://doi.org/10.1038/nmeth.2089>
- [20] C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri, “ImageJ2: ImageJ for the next generation of scientific image data,” *BMC Bioinformatics*, vol. 18, no. 529, pp. 1–26, November 2017. [Online]. Available: <https://doi.org/10.1186/s12859-017-1934-z>
- [21] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: an open-source platform for biological-image analysis,” *Nature Methods*, vol. 9, no. 7, pp. 676–682, July 2012. [Online]. Available: <https://doi.org/10.1038/nmeth.2019>
- [22] F. de Chaumont, S. Dallongeville, N. Chenouard, N. Herve, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. L. Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin, “Icy: an open bioimage informatics platform for extended reproducible research,” *Nature Methods*, vol. 9, no. 7, pp. 690–696, July 2012. [Online]. Available: <https://doi.org/10.1038/nmeth.2075>

- [23] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini, “CellProfiler: image analysis software for identifying and quantifying cell phenotypes,” *Genome Biology*, vol. 7, no. R100, pp. R100–1–11, October 2006. [Online]. Available: <https://doi.org/10.1186/gb-2006-7-10-r100>
- [24] L. Kametsky, T. R. Jones, A. Fraser, M.-A. Bray, D. J. Logan, K. L. Madden, V. Ljosa, C. Rueden, K. W. Eliceiri, and A. E. Carpenter, “Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software,” *Bioinformatics*, vol. 27, no. 8, pp. 1179–1180, April 2011. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btr095>
- [25] C. McQuin, A. Goodman, V. Chernyshev, L. Kametsky, B. A. Cimini, K. W. Karhohs, M. Doan, L. Ding, S. M. Rafelski, D. Thirstrup, W. Wiegand, S. Singh, T. Becker, J. C. Caicedo, and A. E. Carpenter, “CellProfiler 3.0: Next-generation image processing for biology,” *PLOS Biology*, vol. 16, no. 7, pp. e2005970–1–17, July 2018. [Online]. Available: <https://doi.org/10.1371/journal.pbio.2005970>
- [26] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-B. Sibarita, and J. Salameo, “Patch-based nonlocal functional for denoising fluorescence microscopy image sequences,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 442–454, February 2010. [Online]. Available: <https://doi.org/10.1109/TMI.2009.2033991>
- [27] F. Luisier, T. Blu, and M. Unser, “Image denoising in mixed Poisson-Gaussian noise,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, March 2011. [Online]. Available: <https://doi.org/10.1109/TIP.2010.2073477>
- [28] S. Ram and J. J. Rodriguez, “Size-invariant detection of cell nuclei in microscopy images,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 7, pp. 1753–1764, July 2016. [Online]. Available: <https://doi.org/10.1109/TMI.2016.2527740>
- [29] S. Lee, P. Salama, K. W. Dunn, and E. J. Delp, “Boundary fitting based segmentation of fluorescence microscopy images,” *Proceedings of the SPIE/IS&T International Symposium on Electronic Imaging*, pp. 940 805–1–10, February 2015, San Francisco, CA. [Online]. Available: <https://doi.org/10.1117/12.2085417>
- [30] N. J. Gadgil, P. Salama, K. W. Dunn, and E. J. Delp, “Jelly filling segmentation of fluorescence microscopy images containing incomplete labeling,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 531–535, April 2016, Prague, Czech Republic. [Online]. Available: <https://doi.org/10.1109/ISBI.2016.7493324>
- [31] Q. Wu, F. A. Merchant, and K. R. Castleman, *Microscope Image Processing*. Academic Press, 2008.
- [32] E. Meijering, “Cell segmentation: 50 years down the road,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, August 2012. [Online]. Available: <https://doi.org/10.1109/MSP.2012.2204190>

- [33] F. Xing and L. Yang, "Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review," *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 234–263, January 2016. [Online]. Available: <https://doi.org/10.1109/RBME.2016.2515127>
- [34] S. Lee, P. Salama, K. W. Dunn, and E. J. Delp, "Segmentation of fluorescence microscopy images using three dimensional active contours with inhomogeneity correction," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 709–713, April 2017, Melbourne, Australia. [Online]. Available: <https://doi.org/10.1109/ISBI.2017.7950618>
- [35] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, September 1987. [Online]. Available: [https://doi.org/10.1016/S0734-189X\(87\)80186-X](https://doi.org/10.1016/S0734-189X(87)80186-X)
- [36] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, June 1991. [Online]. Available: <https://doi.org/10.1109/34.87344>
- [37] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, September 1991. [Online]. Available: <https://doi.org/10.1109/34.93808>
- [38] D. J. Ho, P. Salama, K. W. Dunn, and E. J. Delp, "Boundary segmentation for fluorescence microscopy using steerable filters," *Proceedings of the SPIE Conference on Medical Imaging*, pp. 101 330E–1–11, February 2017, Orlando, FL. [Online]. Available: <https://doi.org/10.1117/12.2254627>
- [39] C. Fu, S. Lee, D. J. Ho, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, "Three dimensional fluorescence microscopy image synthesis and segmentation," *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 2302–2310, June 2018, Salt Lake City, UT. [Online]. Available: <https://doi.org/10.1109/CVPRW.2018.00298>
- [40] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 2007.
- [41] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, January 1979. [Online]. Available: <https://doi.org/10.1109/TSMC.1979.4310076>
- [42] X. Zhou, F. Li, J. Yan, and S. T. C. Wong, "A novel cell segmentation method and cell phase identification using markov model," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 2, pp. 152–157, March 2009. [Online]. Available: <https://doi.org/10.1109/TITB.2008.2007098>
- [43] O. Schmitt and M. Hasse, "Morphological multiscale decomposition of connected regions with emphasis on cell clusters," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 188–201, February 2009. [Online]. Available: <https://doi.org/10.1016/j.cviu.2008.08.011>

- [44] C. R. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 265–270, February 2003. [Online]. Available: <https://doi.org/10.1109/TPAMI.2003.1177156>
- [45] X. Yang, H. Li, and X. Zhou, "Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and Kalman filter in time-lapse microscopy," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 11, pp. 2405–2414, November 2006. [Online]. Available: <https://doi.org/10.1109/TCSI.2006.884469>
- [46] E. Hodneland, N. V. Bukoreshtliev, T. W. Eichler, X.-C. Tai, S. Gurke, A. Lundervold, and H.-H. Gerdes, "A unified framework for automated 3-D segmentation of surface-stained living cells and a comprehensive segmentation evaluation," *IEEE Transactions on Medical Imaging*, vol. 28, no. 5, pp. 720–738, May 2009. [Online]. Available: <https://doi.org/10.1109/TMI.2008.2011522>
- [47] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988. [Online]. Available: <https://doi.org/10.1007/BF00133570>
- [48] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser, "Snakes on a plane: A perfect snap for bioimage analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 1, pp. 41–48, January 2015. [Online]. Available: <https://doi.org/10.1109/MSP.2014.2344552>
- [49] Y.-L. Fok, J. C. K. Chan, and R. T. Chin, "Automated analysis of nerve-cell images using active contour models," *IEEE Transactions on Medical Imaging*, vol. 15, no. 3, pp. 353–368, June 1996. [Online]. Available: <https://doi.org/10.1109/42.500144>
- [50] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, 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*, vol. 21, no. 10, pp. 1212–1221, October 2002. [Online]. Available: <https://doi.org/10.1109/TMI.2002.806292>
- [51] N. Ray, S. T. Acton, and K. Ley, "Tracking leukocytes in vivo with shape and size constrained active contours," *IEEE Transactions on Medical Imaging*, vol. 21, no. 10, pp. 1222–1235, October 2002. [Online]. Available: <https://doi.org/10.1109/TMI.2002.806291>
- [52] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, March 1998. [Online]. Available: <https://doi.org/10.1109/83.661186>
- [53] C. Xu and J. L. Prince, "Generalized gradient vector flow external forces for active contours," *Signal Processing*, vol. 71, no. 2, pp. 131–139, December 1998. [Online]. Available: [https://doi.org/10.1016/S0165-1684\(98\)00140-6](https://doi.org/10.1016/S0165-1684(98)00140-6)
- [54] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2096–2106, August 2007. [Online]. Available: <https://doi.org/10.1109/TIP.2007.899601>

- [55] B. Li and S. T. Acton, "Automatic active model initialization via Poisson inverse gradient," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1406–1420, August 2008. [Online]. Available: <https://doi.org/10.1109/TIP.2008.925375>
- [56] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, July 1989. [Online]. Available: <https://doi.org/10.1002/cpa.3160420503>
- [57] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001. [Online]. Available: <https://doi.org/10.1109/83.902291>
- [58] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, November 1988. [Online]. Available: [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
- [59] K. S. Lorenz, P. Salama, K. W. Dunn, and E. J. Delp, "Three dimensional segmentation of fluorescence microscopy images using active surfaces," *Proceedings of the IEEE International Conference on Image Processing*, pp. 1153–1157, September 2013, Melbourne, Australia. [Online]. Available: <https://doi.org/10.1109/ICIP.2013.6738238>
- [60] C. Li, R. Huang, Z. Ding, J. C. Gatenby, D. N. Metaxas, and J. C. Gore, "A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 2007–2016, July 2011. [Online]. Available: <https://doi.org/10.1109/TIP.2011.2146190>
- [61] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Transactions on Image Processing*, vol. 29, no. 3, pp. 852–867, March 2010. [Online]. Available: <https://doi.org/10.1109/TMI.2009.2038693>
- [62] C. Pluempitwiriyaewej, J. M. F. Moura, Y. L. Wu, and C. Ho, "STACS: New active contour scheme for cardiac MR image segmentation," *IEEE Transactions on Medical Imaging*, vol. 24, no. 5, pp. 593–603, May 2005. [Online]. Available: <https://doi.org/10.1109/TMI.2005.843740>
- [63] G. Srinivasa, M. C. Fickus, Y. Guo, A. D. Linstedt, and J. Kovacevic, "Active mask segmentation of fluorescence microscope images," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1817–1829, August 2009. [Online]. Available: <https://doi.org/10.1109/TIP.2009.2021081>
- [64] S. Arslan, T. Ersahin, R. Cetin-Atalay, and C. Gunduz-Demir, "Attributed relational graphs for cell nucleus segmentation in fluorescence microscopy images," *IEEE Transactions on Medical Imaging*, vol. 32, no. 6, pp. 1121–1131, June 2013. [Online]. Available: <https://doi.org/10.1109/TMI.2013.2255309>
- [65] J. Cardinale, G. Paul, and I. F. Sbalzarini, "Discrete region competition for unknown numbers of connected regions," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3531–3545, August 2012. [Online]. Available: <https://doi.org/10.1109/TIP.2012.2192129>

- [66] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, September 1996. [Online]. Available: <https://doi.org/10.1109/34.537343>
- [67] G. Paul, J. Cardinale, and I. F. Sbalzarini, "Coupling image restoration and segmentation: A generalized linear model/Bregman perspective," *International Journal of Computer Vision*, vol. 104, no. 1, pp. 69–93, March 2013. [Online]. Available: <https://doi.org/10.1007/s11263-013-0615-2>
- [68] A. Rizk, G. Paul, P. Incardona, M. Bugarski, M. Mansouri, A. Niemann, U. Ziegler, P. Berger, and I. F. Sbalzarini, "Segmentation and quantification of subcellular structures in fluorescence microscopy images using Squash," *Nature Protocols*, vol. 9, no. 3, pp. 586–596, February 2014. [Online]. Available: <https://doi.org/10.1038/nprot.2014.037>
- [69] N. J. Gadgil, P. Salama, K. W. Dunn, and E. J. Delp, "Nuclei segmentation of fluorescence microscopy images based on midpoint analysis and marked point process," *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 37–40, March 2016, Santa Fe, NM. [Online]. Available: <https://doi.org/10.1109/SSIAI.2016.7459169>
- [70] X. Descombes, R. Minlos, and E. Zhizhina, "Object extraction using a stochastic birth-and-death dynamics in continuum," *Journal of Mathematical Imaging and Vision*, vol. 33, no. 2, pp. 347–359, March 2009. [Online]. Available: <https://doi.org/10.1007/s10851-008-0117-y>
- [71] L. D. Cohen and R. Kimmel, "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, August 1997. [Online]. Available: <https://doi.org/10.1023/A:1007922224810>
- [72] T. Deschamps and L. D. Cohen, "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy," *Medical Image Analysis*, vol. 5, no. 4, pp. 281–299, December 2001. [Online]. Available: [https://doi.org/10.1016/S1361-8415\(01\)00046-9](https://doi.org/10.1016/S1361-8415(01)00046-9)
- [73] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. van Walsum, and M. A. Viergever, "Model-based quantitation of 3-D magnetic resonance angiographic images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 946–956, October 1999. [Online]. Available: <https://doi.org/10.1109/42.811279>
- [74] L. M. Lorigo, O. D. Faugeras, W. E. L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin, "CURVES: Curve evolution for vessel segmentation," *Medical Image Analysis*, vol. 5, no. 3, pp. 195–206, September 2001. [Online]. Available: [https://doi.org/10.1016/S1361-8415\(01\)00040-8](https://doi.org/10.1016/S1361-8415(01)00040-8)
- [75] M. W. K. Law and A. C. S. Chung, "Three dimensional curvilinear structure detection using optimally oriented flux," *Proceedings of the European Conference on Computer Vision*, pp. 368–382, October 2008, Marseille, France. [Online]. Available: [https://doi.org/10.1007/978-3-540-88693-8\\_27](https://doi.org/10.1007/978-3-540-88693-8_27)

- [76] F. Benmansour and L. D. Cohen, "Tubular structure segmentation based on minimal path method and anisotropic enhancement," *International Journal of Computer Vision*, vol. 92, no. 2, pp. 192–210, March 2010. [Online]. Available: <https://doi.org/10.1007/s11263-010-0331-0>
- [77] V. Mohan, G. Sundaramoorthi, and A. Tannenbaum, "Tubular surface segmentation for extracting anatomical structures from medical imagery," *IEEE Transactions on Medical Imaging*, vol. 29, no. 12, pp. 1945–1958, December 2010. [Online]. Available: <https://doi.org/10.1109/TMI.2010.2050896>
- [78] T. M. Mitchell, *Machine Learning*. McGraw-Hill Education, 1997.
- [79] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
- [80] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, October 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [81] M. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, July 1998. [Online]. Available: <https://doi.org/10.1109/5254.708428>
- [82] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht, "ilastik: Interactive learning and segmentation toolkit," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 230–233, March 2011, Chicago, IL. [Online]. Available: <https://doi.org/10.1109/ISBI.2011.5872394>
- [83] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015. [Online]. Available: <https://doi.org/10.1038/nature14539>
- [84] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [85] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Proceedings of the Neural Information Processing Systems*, pp. 1097–1105, December 2012, Lake Tahoe, NV. [Online]. Available: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
- [86] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, pp. 1–38, October 2015. [Online]. Available: <https://arxiv.org/abs/1506.00019>
- [87] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, October 2017. [Online]. Available: <https://doi.org/10.1109/TNNLS.2016.2582924>
- [88] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, July 2017. [Online]. Available: <https://doi.org/10.1109/MSP.2017.2693418>
- [89] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, pp. 1–14, February 2017. [Online]. Available: <https://arxiv.org/abs/1609.02907>

- [90] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998. [Online]. Available: <https://doi.org/10.1109/5.726791>
- [91] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," *Proceedings of the International Conference on Machine Learning*, pp. 873–880, June 2009, Montreal, Canada. [Online]. Available: <https://doi.org/10.1145/1553374.1553486>
- [92] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5122–5130, June 2009, Miami, FL. [Online]. Available: <https://doi.org/10.1109/CVPR.2009.5206848>
- [93] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, December 2015. [Online]. Available: <https://doi.org/10.1007/s11263-015-0816-y>
- [94] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010. [Online]. Available: <https://doi.org/10.1007/s11263-009-0275-4>
- [95] M. Everingham, S. M. A. Eslami, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, January 2015. [Online]. Available: <https://doi.org/10.1007/s11263-014-0733-5>
- [96] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft COCO: Common objects in context," *arXiv preprint arXiv:1405.0312*, pp. 1–15, February 2015. [Online]. Available: <https://arxiv.org/abs/1405.0312>
- [97] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through ADE20K dataset," *arXiv preprint arXiv:1608.05442*, pp. 1–12, August 2016. [Online]. Available: <https://arxiv.org/abs/1608.05442>
- [98] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5122–5130, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.544>
- [99] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. [Online]. Available: <https://doi.org/10.1162/neco.1989.1.4.541>
- [100] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, pp. 1–15, December 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>

- [101] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, pp. 1–4, June 2014. [Online]. Available: <https://arxiv.org/abs/1408.5093>
- [102] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: A system for large-scale machine learning,” *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, pp. 265–283, November 2016, Savannah, GA. [Online]. Available: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>
- [103] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning,” *Proceedings of the BigLearn workshop at the Neural Information Processing Systems*, pp. 1–6, December 2011, Granada, Spain. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.231.4195>
- [104] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” *Proceedings of the Autodiff Workshop at the Advances in Neural Information Processing Systems*, pp. 1–4, December 2017, Long Beach, CA. [Online]. Available: <https://openreview.net/forum?id=BJJsrnfCZ>
- [105] M. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *Proceedings of the European Conference on Computer Vision*, pp. 828–833, September 2014, Zurich, Switzerland. [Online]. Available: [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)
- [106] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” *arXiv preprint arXiv:1506.06579*, pp. 1–12, June 2015. [Online]. Available: <https://arxiv.org/abs/1506.06579>
- [107] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, pp. 1–14, April 2015. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [108] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, June 2015, Boston, MA. [Online]. Available: <https://doi.org/10.1109/CVPR.2015.7298594>
- [109] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016, Las Vegas, NV. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.90>
- [110] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikainen, “Deep learning for generic object detection: A survey,” *arXiv preprint arXiv:1809.02165*, pp. 1–30, September 2018. [Online]. Available: <https://arxiv.org/abs/1809.02165>

- [111] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, June 2014, Columbus, OH. [Online]. Available: <https://doi.org/10.1109/CVPR.2014.81>
- [112] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, January 2013. [Online]. Available: <https://doi.org/10.1007/s11263-013-0620-5>
- [113] R. Girshick, “Fast R-CNN,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, December 2015, Santiago, Chile. [Online]. Available: <https://doi.org/10.1109/ICCV.2015.169>
- [114] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Proceedings of the Neural Information Processing Systems*, pp. 91–99, December 2015, Montreal, Canada. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks>
- [115] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” *Proceedings of the European Conference on Computer Vision*, pp. 21–38, October 2016, Amsterdam, The Netherlands. [Online]. Available: [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- [116] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, June 2016, Las Vegas, NV. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.91>
- [117] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517–6525, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.690>
- [118] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, pp. 1–6, April 2018. [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [119] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A review on deep learning techniques applied to semantic segmentation,” *arXiv preprint arXiv:1704.06857*, pp. 1–23, April 2017. [Online]. Available: <https://arxiv.org/abs/1704.06857>
- [120] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” *Proceedings of the Neural Information Processing Systems*, pp. 2843–2851, December 2012, Lake Tahoe, NV. [Online]. Available: <http://papers.nips.cc/paper/4741-deep-neural-networks>
- [121] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, June 2015, Boston, MA. [Online]. Available: <https://doi.org/10.1109/CVPR.2015.7298965>

- [122] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017. [Online]. Available: <https://doi.org/10.1109/TPAMI.2016.2572683>
- [123] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528, December 2015, Santiago, Chile. [Online]. Available: <https://doi.org/10.1109/ICCV.2015.178>
- [124] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, December 2017. [Online]. Available: <https://doi.org/10.1109/TPAMI.2016.2644615>
- [125] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 231–241, October 2015, Munich, Germany. [Online]. Available: [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
- [126] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, pp. 1–13, April 2016. [Online]. Available: <https://arxiv.org/abs/1511.07122>
- [127] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, “A real-time algorithm for signal analysis with the help of the wavelet transform,” *Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference*, pp. 286–297, December 1987, Marseille, France. [Online]. Available: [https://doi.org/10.1007/978-3-642-75988-8\\_28](https://doi.org/10.1007/978-3-642-75988-8_28)
- [128] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6230–6239, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.660>
- [129] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5168–5177, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.549>
- [130] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” *arXiv preprint arXiv:1412.7062*, pp. 1–14, June 2016. [Online]. Available: <https://arxiv.org/abs/1412.7062>
- [131] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April 2018. [Online]. Available: <https://doi.org/10.1109/TPAMI.2017.2699184>

- [132] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Re-thinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*, pp. 1–14, December 2017. [Online]. Available: <https://arxiv.org/abs/1706.05587>
- [133] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *arXiv preprint arXiv:1802.02611*, pp. 1–18, August 2018. [Online]. Available: <https://arxiv.org/abs/1802.02611>
- [134] P. Krahenbuhl and V. Koltun, “Efficient inference in fully connected CRFs with gaussian edge potentials,” *Proceedings of the Neural Information Processing Systems*, pp. 109–117, December 2011, Granada, Spain. [Online]. Available: <http://papers.nips.cc/paper/4296-efficient-inference-in-fully-connected-crfs-with-gaussian-edge-potentials>
- [135] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800–1807, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.195>
- [136] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, October 2017, Venice, Italy. [Online]. Available: <https://doi.org/10.1109/ICCV.2017.322>
- [137] I. Radosavovic, P. Dollar, R. Girshick, G. Gkioxari, and K. He, “Data distillation: Towards omni-supervised learning,” *arXiv preprint arXiv:1712.04440*, pp. 1–10, December 2017. [Online]. Available: <https://arxiv.org/abs/1712.04440>
- [138] R. Hu, P. Dollar, K. He, T. Darrell, and R. Girshick, “Learning to segment every thing,” *arXiv preprint arXiv:1711.10370*, pp. 1–9, March 2018. [Online]. Available: <https://arxiv.org/abs/1711.10370>
- [139] G. Litjens, T. Kooi, B. Bejnordi, A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, and C. Sanchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, July 2017. [Online]. Available: <https://doi.org/10.1016/j.media.2017.07.005>
- [140] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Mitosis detection in breast cancer histology images with deep neural networks,” *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 411–418, September 2013, Nagoya, Japan. [Online]. Available: [https://doi.org/10.1007/978-3-642-40763-5\\_51](https://doi.org/10.1007/978-3-642-40763-5_51)
- [141] B. Dong, L. Shao, M. D. Costa, O. Bandmann, and A. F. Frangi, “Deep learning for automatic cell detection in wide-field microscopy zebrafish images,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 772–776, April 2015, Brooklyn, NY. [Online]. Available: <https://doi.org/10.1109/ISBI.2015.7163986>
- [142] M. Kolesnik and A. Fexa, “Multi-dimensional color histograms for segmentation of wounds in images,” *Proceedings of the International Conference Image Analysis and Recognition*, pp. 1014–1022, September 2005, Toronto, Canada. [Online]. Available: [https://doi.org/10.1007/11559573\\_123](https://doi.org/10.1007/11559573_123)

- [143] Y. Xie, X. Kong, F. Xing, F. Liu, H. Su, and L. Yang, "Deep voting: A robust approach toward nucleus localization in microscopy images," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 374–382, October 2015, Munich, Germany. [Online]. Available: [https://doi.org/10.1007/978-3-319-24574-4\\_45](https://doi.org/10.1007/978-3-319-24574-4_45)
- [144] A. Giusti, D. C. Cirean, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," *Proceedings of the IEEE International Conference on Image Processing*, pp. 4034–4038, September 2013, Melbourne, Australia. [Online]. Available: <https://doi.org/10.1109/ICIP.2013.6738831>
- [145] Y. Xie, F. Xing, X. Kong, H. Su, and L. Yang, "Beyond classification: Structured regression for robust cell detection using convolutional neural network," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 358–365, October 2015, Munich, Germany. [Online]. Available: [https://doi.org/10.1007/978-3-319-24574-4\\_43](https://doi.org/10.1007/978-3-319-24574-4_43)
- [146] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 1, pp. 119–130, January 2016. [Online]. Available: <https://doi.org/10.1109/TMI.2015.2458702>
- [147] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1196–1206, May 2016. [Online]. Available: <https://doi.org/10.1109/TMI.2016.2525803>
- [148] S. E. A. Raza, K. AbdulJabbar, M. Jamal-Hanjani, S. Veeriah, J. L. Quesne, C. Swanton, and Y. Yuan, "Deconvolving convolution neural network for cell detection," *arXiv preprint arXiv:1806.06970*, pp. 1–9, June 2018. [Online]. Available: <https://arxiv.org/abs/1806.06970>
- [149] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cirean, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, T. Liu, M. Seyedhosseini, T. Tasdizen, L. Kamentsky, R. Burget, V. Uher, X. Tan, C. Sun, T. D. Pham, E. Bas, M. G. Uzunbas, A. Cardona, J. Schindelin, and H. S. Seung, "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Frontiers in Neuroanatomy*, vol. 9, no. 142, pp. 1–13, November 2015. [Online]. Available: <https://doi.org/10.3389/fnana.2015.00142>
- [150] M. Maska, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. Espana, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolckova, M. Streitova, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G. Magnusson, J. Jalden, H. M. Blau, O. Dzyubachyk, P. Krizek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Munoz-Barrutia, E. Meijering, M. Kozubek, and C. O. de Solorzano, "A benchmark for comparison of cell tracking algorithms," *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, June 2014. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btu080>

- [151] T. Falk, D. Mai, R. Bensch, O. Cicek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jackel, K. Seiwald, A. Dovzhenko, O. Tietz, C. D. Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger, “U-Net: deep learning for cell counting, detection, and morphometry,” *Nature Method*, vol. 16, pp. 67–70, January 2019. [Online]. Available: <https://doi.org/10.1038/s41592-018-0261-2>
- [152] S. E. A. Raza, L. Cheung, D. Epstein, S. Pelengaris, M. Khan, and N. M. Rajpoot, “MIMO-Net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 337–340, April 2017, Melbourne, Australia. [Online]. Available: <https://doi.org/10.1109/ISBI.2017.7950532>
- [153] S. E. A. Raza, L. Cheung, M. Shaban, S. Graham, D. Epstein, S. Pelengaris, M. Khan, and N. M. Rajpoot, “Micro-Net: A unified model for segmentation of various objects in microscopy images,” *arXiv preprint arXiv:1804.08145*, pp. 1–32, April 2018. [Online]. Available: <https://arxiv.org/abs/1804.08145>
- [154] H. Chen, X. Qi, L. Yu, and P.-A. Heng, “DCAN: Deep contour-aware networks for accurate gland segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2487–2496, June 2016, Las Vegas, NV. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.273>
- [155] K. Sirinukunwattana, J. P. W. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, A. Böhm, O. Ronneberger, B. B. Cheikh, D. Racoceanu, P. Kainz, M. Pfeiffer, M. Urschler, D. R. J. Snead, and N. M. Rajpoot, “Gland segmentation in colon histology images: The glas challenge contest,” *Medical Image Analysis*, vol. 35, pp. 489–502, January 2017. [Online]. Available: <https://doi.org/10.1016/j.media.2016.08.008>
- [156] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, “DCAN: Deep contour-aware networks for object instance segmentation from histology images,” *Medical Image Analysis*, pp. 135–146, February 2017. [Online]. Available: <https://doi.org/10.1016/j.media.2016.11.004>
- [157] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, “A dataset and a technique for generalized nuclear segmentation for computational pathology,” *IEEE Transaction on Medical Imaging*, vol. 36, no. 7, pp. 1550–1560, July 2017. [Online]. Available: <https://doi.org/10.1109/TMI.2017.2677499>
- [158] F. A. Guerrero-Pena, P. D. M. Fernandez, T. I. Ren, M. Yui, E. Rothenberg, and A. Cunha, “Multiclass weighted loss for instance segmentation of cluttered cells,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 2451–2455, October 2018, Athens, Greece. [Online]. Available: <https://doi.org/10.1109/ICIP.2018.8451187>
- [159] S. Graham and N. M. Rajpoot, “SAMS-NET: Stain-aware multi-scale network for instance-based nuclei segmentation in histology images,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 590–594, April 2018, Washington, D.C. [Online]. Available: <https://doi.org/10.1109/ISBI.2018.8363645>

- [160] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, “Cell detection with star-convex polygons,” *arXiv preprint arXiv:1806.03535*, pp. 1–8, June 2018. [Online]. Available: <https://arxiv.org/abs/1806.03535>
- [161] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkila, “Cell proposal network for microscopy image analysis,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 3199–3203, September 2016, Phoenix, AZ. [Online]. Available: <https://doi.org/10.1109/ICIP.2016.7532950>
- [162] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkila, “Cell segmentation proposal network for microscopy image analysis,” *Proceedings of the Deep Learning in Medical Image Analysis workshop at the Medical Image Computing and Computer-Assisted Intervention*, pp. 21–29, October 2016, Athens, Greece. [Online]. Available: [https://doi.org/10.1007/978-3-319-46976-8\\_3](https://doi.org/10.1007/978-3-319-46976-8_3)
- [163] J. Johnson, “Adapting Mask-RCNN for automatic nucleus segmentation,” *arXiv preprint arXiv:1805.00500*, pp. 1–7, May 2018. [Online]. Available: <https://arxiv.org/abs/1805.00500>
- [164] J. Yi, P. Wu, D. J. Hoepfner, and D. Metaxas, “Pixel-wise neural cell instance segmentation,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 373–377, April 2018, Washington, D.C. [Online]. Available: <https://doi.org/10.1109/ISBI.2018.8363596>
- [165] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, “Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network,” *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 246–253, September 2013, Nagoya, Japan. [Online]. Available: [https://doi.org/10.1007/978-3-642-40763-5\\_31](https://doi.org/10.1007/978-3-642-40763-5_31)
- [166] H. R. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers, “A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations,” *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 520–527, September 2014, Boston, MA. [Online]. Available: [https://doi.org/10.1007/978-3-319-10404-1\\_65](https://doi.org/10.1007/978-3-319-10404-1_65)
- [167] C. Fu, D. J. Ho, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei segmentation of fluorescence microscopy images using convolutional neural networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 704–708, April 2017, Melbourne, Australia. [Online]. Available: <https://doi.org/10.1109/ISBI.2017.7950617>
- [168] O. Cicek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning dense volumetric segmentation from sparse annotation,” *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, October 2016, Athens, Greece. [Online]. Available: [https://doi.org/10.1007/978-3-319-46723-8\\_49](https://doi.org/10.1007/978-3-319-46723-8_49)
- [169] S. Ram, V. Nguyen, K. Limesand, and M. Sabuncu, “Joint cell nuclei detection and segmentation in microscopy images using 3D convolutional networks,” *arXiv preprint arXiv:1805.02850*, pp. 1–8, May 2018. [Online]. Available: <https://arxiv.org/abs/1805.02850>

- [170] Z. Zhao, L. Yang, H. Zheng, I. Guldner, S. Zhang, and D. Chen, "Deep learning based instance segmentation in 3D biomedical images using weak annotation," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 352–360, September 2018, Granada, Spain. [Online]. Available: [https://doi.org/10.1007/978-3-030-00937-3\\_41](https://doi.org/10.1007/978-3-030-00937-3_41)
- [171] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2672–2680, December 2014, Montreal, Canada. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets>
- [172] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernandez, J. Wardlaw, and D. Rueckert, "GAN augmentation: Augmenting training data using generative adversarial networks," *arXiv preprint arXiv:1810.10863*, pp. 1–12, October 2018. [Online]. Available: <https://arxiv.org/abs/1810.10863>
- [173] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *arXiv preprint arXiv:1809.07294*, pp. 1–20, September 2018. [Online]. Available: <https://arxiv.org/abs/1809.07294>
- [174] S. Kazemina, C. Baur, A. Kuijper, B. van Ginneken, N. Navab, S. Albarqouni, and A. Mukhopadhyay, "GANs for medical image analysis," *arXiv preprint arXiv:1809.06222*, pp. 1–41, December 2018. [Online]. Available: <https://arxiv.org/abs/1809.06222>
- [175] A. Osokin, A. Chessel, R. E. C. Salas, and F. Vaggi, "GANs for biological image synthesis," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2252–2261, October 2017, Venice, Italy. [Online]. Available: <https://doi.org/10.1109/ICCV.2017.245>
- [176] F. Mahmood, D. Borders, R. Chen, G. N. McKay, K. J. Salimian, A. Baras, and N. J. Durr, "Deep adversarial training for multi-organ nuclei segmentation in histopathology images," *arXiv preprint arXiv:1810.00236*, pp. 1–10, October 2018. [Online]. Available: <https://arxiv.org/abs/1810.00236>
- [177] L. Hou, A. Agarwal, D. Samaras, T. M. Kurc, R. R. Gupta, and J. H. Saltz, "Unsupervised histopathology image synthesis," *arXiv preprint arXiv:1712.05021*, pp. 1–11, December 2017. [Online]. Available: <https://arxiv.org/abs/1712.05021>
- [178] O. Bailo, D. Ham, and Y. M. Shin, "Red blood cell image generation for data augmentation using conditional generative adversarial networks," *arXiv preprint arXiv:1901.06219*, pp. 1–9, January 2019. [Online]. Available: <https://arxiv.org/abs/1901.06219>
- [179] C. Baur, S. Albarqouni, and N. Navab, "Generating highly realistic images of skin lesions with GANs," *arXiv preprint arXiv:1809.01410*, pp. 1–9, September 2018. [Online]. Available: <https://arxiv.org/abs/1809.01410>
- [180] Y. Huo, Z. Xu, S. Bao, A. Assad, R. G. Abramson, and B. A. Landman, "Adversarial synthesis learning enables segmentation without target modality ground truth," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 1217–1220, April 2018, Washington, D.C. [Online]. Available: <https://doi.org/10.1109/ISBI.2018.8363790>

- [181] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.632>
- [182] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, October 2017, Venice, Italy. [Online]. Available: <https://doi.org/10.1109/ICCV.2017.244>
- [183] M. Jacob and M. Unser, “Design of steerable filters for feature detection using canny-like criteria,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1007–1019, August 2004. [Online]. Available: <https://doi.org/10.1109/TPAMI.2004.44>
- [184] S. Jeong, Y. Tarabalka, and J. Zerubia, “Marked point process model for facial wrinkle detection,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 1391–1394, October 2014, Paris, France. [Online]. Available: <https://doi.org/10.1109/ICIP.2014.7025278>
- [185] J. L. Clendenon, C. L. Phillips, R. M. Sandoval, S. Fang, and K. W. Dunn, “Voxx: a PC-based, near real-time volume rendering system for biological microscopy,” *American Journal of Physiology-Cell Physiology*, vol. 282, no. 1, pp. C213–C218, January 2002. [Online]. Available: <https://doi.org/10.1152/ajpcell.2002.282.1.C213>
- [186] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006. [Online]. Available: <https://doi.org/10.1016/j.patrec.2005.10.010>
- [187] D. M. W. Powers, “Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, December 2011. [Online]. Available: <https://dSPACE2.flinders.edu.au/xmlui/handle/2328/27165>
- [188] D. Sage, L. Donati, F. Soulez, D. Fortun, G. Schmit, A. Seitz, R. Guiet, C. Vonesch, and M. Unser, “DeconvolutionLab2: An open-source software for deconvolution microscopy,” *Methods*, vol. 115, pp. 28–41, February 2017. [Online]. Available: <https://doi.org/10.1016/j.ymeth.2016.12.015>
- [189] S. F. Gibson and F. Lanni, “Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy,” *Journal of the Optical Society of America*, vol. 9, no. 1, pp. 154–166, January 1992. [Online]. Available: <https://doi.org/10.1364/JOSAA.9.000154>
- [190] H. Kirshner, F. Aguet, D. Sage, and M. Unser, “3D PSF fitting for fluorescence microscopy: implementation and localization application,” *Journal of Microscopy*, vol. 249, no. 1, pp. 13–25, January 2013. [Online]. Available: <https://doi.org/10.1111/j.1365-2818.2012.03675.x>
- [191] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, pp. 1–11, March 2015. [Online]. Available: <https://arxiv.org/abs/1502.03167>

- [192] D. J. Ho, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei segmentation of fluorescence microscopy images using three dimensional convolutional neural networks,” *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 834–842, July 2017, Honolulu, HI. [Online]. Available: <https://doi.org/10.1109/CVPRW.2017.116>
- [193] D. J. Ho, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, “Nuclei detection and segmentation of fluorescence microscopy images using three dimensional convolutional neural networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 418–422, April 2018, Washington, D.C. [Online]. Available: <https://doi.org/10.1109/ISBI.2018.8363606>
- [194] P. Yushkevich, J. Piven, H. Hazlett, R. Smith, S. Ho, J. Gee, and G. Gerig, “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,” *NeuroImage*, vol. 31, no. 3, pp. 1116–1128, July 2006. [Online]. Available: <https://doi.org/10.1016/j.neuroimage.2006.01.015>
- [195] C. Glasbey, G. van der Heijden, V. F. K. Toh, and A. Gray, “Colour displays for categorical images,” *Color Research and Application*, vol. 32, no. 4, pp. 304–309, August 2007. [Online]. Available: <https://doi.org/10.1002/col.20327>

VITA

## VITA

David Joon Ho was born in Santa Barbara, CA. He received his Bachelor of Science in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2010. He received his Master of Science in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2012.

Mr. Ho joined the Ph.D. program at the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana in August 2013. He worked at the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Edward J. Delp. He was an intern at HP Labs in Palo Alto, CA in the summer of 2017.

Mr. Ho's research interests include image processing, computer vision, and machine learning especially in biomedical images. He is a student member of the IEEE and the IEEE Signal Processing Society.