FLUORESCENCE MICROSCOPY IMAGES SEGMENTATION AND ANALYSIS USING MACHINE LEARNING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Shuo Han

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Edward J. Delp
School of Electrical and Computer Engineering
and Weldon School of Biomedical Engineering

Dr. Paul Salama
School of Electrical and Computer Engineering, IUPUI

Dr. Charles A. Bouman
School of Electrical and Computer Engineering
and Weldon School of Biomedical Engineering

Dr. Fang Huang
Weldon School of Biomedical Engineering

Approved by:

Dr. George R. Wodicka
Head of the Graduate Program

ACKNOWLEDGMENTS

I want to express my thankfulness to the following people mentioned in this section, who have helped me enormously during my doctorate study at Purdue University. Without their supportive advice, I could not have gone this far for my degree.

I am grateful for having Professor Edward J. Delp as my advisor for my graduate studies. Becoming a member of the Video and Image Processing Laboratory (VIPER) has been a life changing experience for me. Without the instruction and inspiration from Prof. Delp, I would not be able to understand and appreciate the essence of academic research. I value the experiences I gained and the problem-solving mindset I learned from working with him. Since joining VIPER, I have been given many research and development opportunities and I have gained a great deal while learning from Prof. Delp. He also helped me to learn how to solve real challenges and successfully implement practical solutions. I have become a better researcher and engineer through his critical thinking and target-oriented approach. I am proud and grateful that I have been working with him.

I wish to thank Professor Paul Salama, a member of my advisor committee. I appreciate his sharing of knowledge as well as his involvement with my research project. He has participated in our weekly meetings, proposed many new ideas and spent lots of hours in our paper reviews. I would also like to express my appreciation to Professor Charles Bouman and Professor Fang Huang, my other advisor committee members, for their valuable input.

I would like to thank Professor Kenneth W. Dunn for the help and lots of work he has contributed to this research, as well as the many microscopy datasets he provided. In addition, he provides the research with biological and medical based perspectives which helps the development of the scientific value of our work.

I would like to thank the former and current teammates Dr. Neeraj J. Gadgil, Dr. Chichen Fu, Dr. Soonam Lee, Dr. David Joon Ho, Alain Chen, Changye Yang and Liming Wu and for their help and support in the microscopy project. We have been through many time together solving challenging problems, building foundations for the microscopy project. Without the dedicated teamwork and collaborations from the talented team members, we could not accomplish the work we have done. In particular, I would like to thank Dr. Chichen Fu for his encouragement, showing me how to be a good problem-solver, and help me with my research. I would also like to thank Dr. Soonam Lee for helping me with my paper, working on new problems together, and contributed to my research. I would also like to thank Dr. David Joon Ho for inspiring me with new research ideas and always working with dedication to research. I would also like to thank Alain Chen for his hard work and lots of contributions to the Microscopy system project and the DINAVID system. I would also like to thank Changye Yang for his help and support for the project. I would also like to thank Liming Wu for getting hands on with the project at a fast pace.

I would like to extend my appreciation to all other great former and current colleagues in the VIPER laboratory: Dr. Khalid Tahboub, Dr. Joonsoo Kim, Dr. Yu Wang, Dr. Shaobo Fang, Dr. Javier Ribera Prat, Dr. Jeehyun Choe, Dr. Dahjung Chung, Dr. David Guera, Dr. Yuhao Chen, He Li, Chang Liu, Sri Kalyan Yarlagadda, Ruiting Shao, Di Chen, Qingshuang Chen, Daniel Mas, Runyu Mao, Zeman Shao, Sriram Baireddy, Emily Bartusiak, Jiaqi Guo, Hanxiang Hao, János Horváth, Enyu Cai, Jiangpeng He, Yifan Zhao, Han Hu, Yue Han, Mridul Gupta, and Justin Yang.

I want to show my gratitude to Purdue BME department and the professors and instructors who worked together with me as a teaching assistant for a long time. Without their support, I could not finish my Ph.D. study.

Finally, I would like to thank my family sincerely. Their unconditional support, love and trust make this work possible. Thank you for your endless love and encouragement, to my father, Zhong Han, and my mother, Xinyu Zhang. I would never be able to complete my Ph.D. without their unconditional support.

This work 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.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Han, Shuo Ph.D., Purdue University, August 2020. Fluorescence Microscopy Images Segmentation and Analysis Using Machine Learning. Major Professor: Edward J. Delp.

Microscopy image analysis can provide substantial information for clinical study and understanding of the biological structure. Two-photon microscopy is a type of fluorescence microscopy that can visualize deep into tissue with near-infrared excitation light. Large 3D image volumes of complex subcellular are often produced, which calls for automatic image analysis techniques. Automatic methods that can obtain nuclei quantity in microscopy image volumes are needed for biomedical research and clinical diagnosis. In general, several challenges exist for counting nuclei in 3D image volumes. These include "crowding" and touching of nuclei, overlapping of two or more nuclei, and shape and size variances of the nuclei. In this thesis, a 3D nuclei counter using two different generative adversarial networks (GAN) is proposed and evaluated. Synthetic data that resembles real microscopy image is generated with a GAN. The synthetic data is used to train another 3D GAN network that counts the number of nuclei. Our approach is evaluated with respect to the number of groundtruth nuclei and compared with common ways of counting used in the biological research. Fluorescence microscopy 3D image volumes of rat kidneys are used to test our 3D nuclei counter. The evaluation of both networks shows that the proposed technique is successful for counting nuclei in 3D. Then, a 3D segmentation and classification method to segment and identify individual nuclei in fluorescence microscopy volumes without having groundtruth volumes is introduced. Three dimensional synthetic data is generated using the Recycle-GAN with the Hausdorff distance loss introduced in to preserve the shape of individual nuclei. Realistic microscopy image volumes with nuclei segmentation mask and nucleus boundary groundtruth volumes are generated. A subsequent 3D CNN with a regularization term that discourages detection out of nucleus boundary is used to detect and segment nuclei. Nuclei boundary refinement is then performed to enhance nuclei segmentation. Experimental results on our rat kidney dataset show the proposed method is competitive with respect to several state-of-the-art methods. A Distributed and Networked Analysis of Volumetric Image Data (DINAVID) system is developed to enable remote analysis of microscopy images for biologists. There are two main functions integrated in the system, a 3D visualization tool and a remote computing tool for nuclei segmentation. The 3D visualization enables real-time rendering of large volumes of microscopy data. The segmentation tool provides fast inferencing of pre-trained deep learning models trained with 5 different types of microscopy data.

1. INTRODUCTION

1.1 Background in Optical Microscopy

Many different types of microscopes have been developed and used for a wide range of applications. There are three main categories of microscopy, which includes optical microscopes, electron microscopes, and scanning tunnel electron microscopes. Among the three categories, optical microscopy is preferred for biological studies. Optical microscopy imaging is less harmful to biological specimens, making it more compatible with clinical applications and biomedical research.

Optical microscopy is a type of microscopy that uses visible light to obtain images of small structures such as tissues or cells. Optical microscopy has been widely used in biomedical research and clinical studies, since it can visualize biological structures without the harm from invisible light. Thus, it serves as a powerful tool for clinical applications with the ability to observe living or moving specimens. In order to understand how light microscopy works we have to know how the optical microscope was made. When forming the image from samples, the objective and the condenser are considered as two major components. The objective is located at the top of the stage where a living specimen was laid which collects light diffracted by the specimen and forms a magnified real image. The condenser below the stage concentrates light from the illuminator on a small area of the [1] specimen. The Koehler illumination's [2] is used to place the lens in front of detectors so that light rays passing through focus in the lens go parallel to one another. The Koehler illumination provides bright and even illumination at the specimen plane due to these parallel light rays. The choice of illumination technique greatly determines the image quality.

Fluorescence microscopy is a type of optical microscopy that can visualize living tissues using fluorescence. Fluorescence describes a process during which a molecule first absorbs light energy and then releases it. The molecule that can absorb light is referred to as a “fluorophore”, also known as fluorescent molecules or fluorescent dyes. When the fluorophore absorbs energy and becomes excited, it reaches the excited state. Excited states include singlet states and triplet states, both have higher energy level than the ground state. The singlet state has its electron pairs spinning in the same orbital, whereas the triplet state has its electron pairs spinning in two orbitals forming three different possible configurations [3]. When the fluorophore emits light energy, it falls back to the ground state. The detailed process of fluorescence is illustrated in Figure 1.1, known as the Jablonski diagram [3].

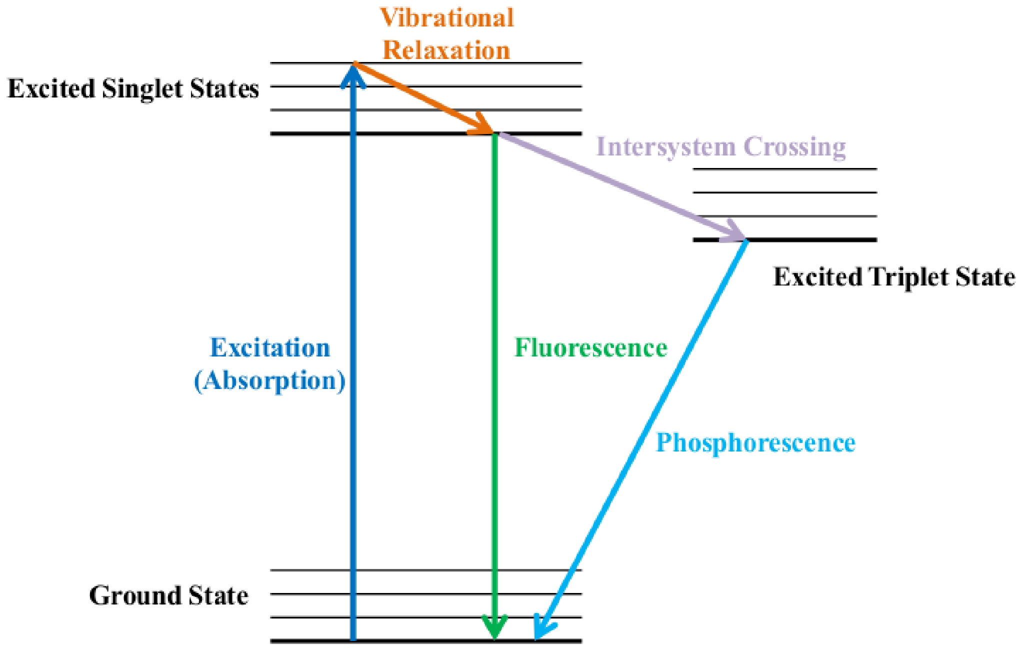


Fig. 1.1.: Jablonski diagram

A specific wavelength is used to excite the electrons of the fluorophore when it’s at ground state, which then absorbs photons and enters the excited states. Multiple excited states can be reached after excitation based on the difference of the energy level. The excitation of a fluorophore from ground state to excited states takes place in the scale of femtoseconds (10^{-15} seconds), leaving the molecule in the excited singlet states. Since fluorophores that are at the excited singlet states have high energy configurations, they are unstable. The excited fluorophores have a high tendency to transit to the excited singlet state that has the lowest vibrational energy. This process is referred to as vibrational relaxation which happens in the scale of around 10^{-11} - 10^{-14} seconds. The time that a molecule stays in its excited states is called the excited lifetime. During the excited lifetimes, most of molecules transit to the lowest excited singlet state with complete vibrational relaxation. The excess energy from vibrational relaxation is released into the environment in the form of heat energy. Finally, the molecules relax back to the ground state in around several nanoseconds (10^{-9}). At the same time, photons with different energy levels are emitted, i. e., fluorescence emission. Intersystem crossing is another process that can occur with relatively low probability. Instead of collapsing to the ground state, the molecules in the excited singlet states can transfer to the excited triplet state. If photons are emitted and the molecules relax to ground state from the excited triplet state, the process is referred to as phosphorescence. If the molecules transit to the excited singlet state from the excited triplet state, the process is referred to as delayed fluorescence. The energy of a photon is expressed as

$$E = \frac{hc}{\lambda} \tag{1.1}$$

here, Planck's constant is denoted as h, c denotes the speed of light, and λ denotes the wavelength. The wavelength of emitted light is longer than the wavelength from the light source, due to energy lost during vibrational relaxation which caused the energy released during the emission to be less than the energy absorbed during the excitation. This loss of energy during vibrational relaxation that leads to a difference between the absorption and emission spectra maxima can be referred to as the Stokes shift. An example of the Stokes shift in the excitation and emission spectra for green fluorescent protein (GFP) is shown in Figure 1.2. Fluorophores with a greater storage shift are desired because it is easier to isolate the emitted light through interference filters from the incoming light. The quantum yield is also another property of fluorophores in which the quantity yield is the ratio of the number of fluorescent photons emitted to the number of photons absorbed. Increased quantum output is required to generate brighter images with the same light intensity.

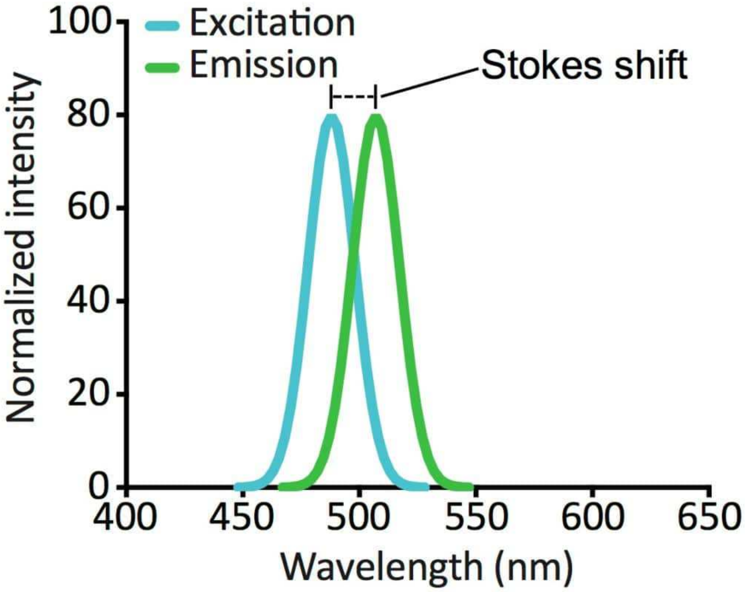


Fig. 1.2.: Stokes shift of GFP

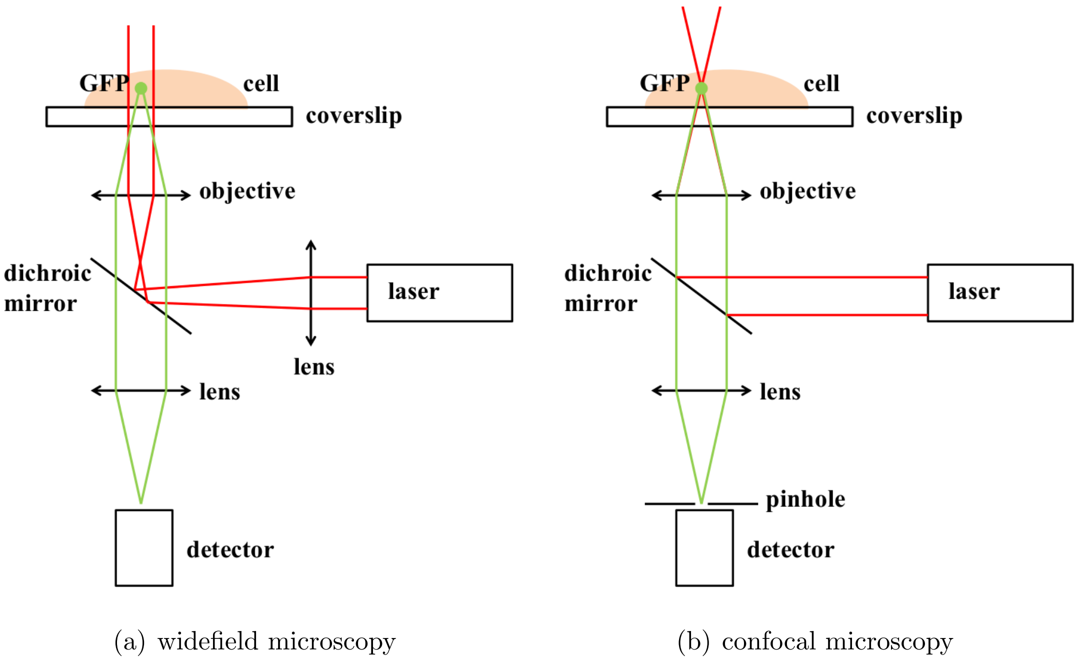
Although the excitation and fluorescence process is repeatable, some of the molecules can lose the ability to fluorescent if it is photobleached. Photobleaching occurs when the molecules in the excited triplet state react with oxidative surrounding agents and become permanently unable to fluoresce [1].

In summary, the common fluorescence process can be described as a three-step cyclical process:

- Step 1. The excitation of fluorophore with photon absorption.
- Step 2. The fluorophore losses some energy with vibrational relaxation during the excited lifetime.
- Step 3. The fluorophore emits photon and release back to the ground state.

Fluorescent microscopy is widely used in immunohistochemistry. The traditional method to localize protein is through immunohistochemistry in which a target protein is detected with fluorescent antibodies. The method has been applied to xanthine oxidase, carbonic anhydrase, and protein kinases C localization assays in skeletal muscle. Some limitations to this method include the large quantities of detection agent required and the need for cell permeabilization. The discovery of fluorescent proteins such green fluorescent protein (GFP) has contributed greatly to live fluorescence microscopy imaging. In a method called "genetic tagging", the GFP gene is commonly used as a reporter protein that helps in locating a co-expressed target protein. This method is suitable for identifying protein localization changes in response to external signals. However, it relies heavily on transfection and transgenic technology, which in turn limits its accessibility. When fluorophores are injected into species, the species can be magnified and visualized by microscopes. When the light passes through a target lens, the sample is illuminated evenly. If a fluorophore is located in the area where it is illuminated from the light source, the fluorophore emitted light passes through a dichroic mirror and a tube lens and is pictured on a detector. [1]

Confocal microscopy [1] is a type of optical microscopy which was invented by Minsky in 1955 [4]. A pinhole is added in confocal microscopy which is placed in front of electron detectors. Transitionally, widefield microscopy's signal acquisition suffers from emitted light from fluorophores in different focal planes. The images acquired from widefield microscopy are blurred by emission originating away from the focal plane. Blurred background and low-contrast images are produced from widefield microscopy. More specifically, thick fluorescent specimens may be challenging since bright signals from objects lying outside the focal plane increase the background and yield low contrast images. As opposed to widefield microscope, confocal microscope located a pinhole in front of detectors so that it successfully rejects signals from nearby sources above and below the focal plane. In confocal microscopy, the light focuses on one focal plane instead of uniformly illumination on the sample. A spinning disk is used to fast scan multiple points simultaneously [5].



(a) widefield microscopy

(b) confocal microscopy

Fig. 1.3.: Widefield microscopy and confocal microscopy

Figure 1.3 shows widefield microscopy and confocal microscopy. Confocal microscopy has certain limitations for deep tissue photon detection. If a sample's thickness is greater than the wavelength of visible light from a source, due to light scattering, the number of photons from fluorescent molecules in deep tissue can be reduced. With near-infrared (IR) illumination, two-photon microscopy [1, 6–9] can visualize deeper into tissue. Because IR light has longer wavelength than visible light, even in deep tissue, IR light scatters less. IR light contains less energy than visible light according to Equation 1.1, so two or more photons must simultaneously excite the fluorescent molecules. The signal is linear to the intensity of the incoming light in confocal microscopy. For two-photon microscopy, the signal is not linear to the intensity of the incoming light, but a square of the incoming light's intensity.

Compared with confocal microscopy, two-photon microscopy has many benefits specifically for deep tissue imaging. As indicated by confocal microscopy, [10] places pinhole aperture in front of the detector to reject unfocused dispersed light rays so that the signal strength is reduced. Two-photon microscopy, on the other hand, uses near-infrared rays that can image the subcellular structures without damaging *in vivo* [11] since each molecule requires photon with less energy to excite. Deep images acquisition inside intact living tissues becomes possible because of the preservation of signal power. In addition, the use of near-infrared rays decreases the scattering effect such that fluorescence emissions from the focal plane are collected efficiently over a large detector area. More specifically, the amount of scattering is inversely proportional to the fourth power of the wavelength in case the scattering particle is much smaller than, for example, less than $1/10\lambda$ [10]. Thus, the excitation of two-photon microscopy is limited to a small volume that allows for higher contrast, lower photo damage, and less photobleaching [1].

Multiphoton microscopy techniques are also being developed for visualization in deep tissue [11–14], which improved the two-photon microscopy technique. The use of more photons has a lower risk of destroying living tissues as longer wavelengths can be used compared to microscopy with two photons. For example, an infrared beam at $1050nm$ will produce three-photon excitation of an equivalent fluorophore absorbing ultraviolet light at $350nm$ whereas two-photon using green fluorophore at $525nm$ may allow the same illumination. In addition, excitation of three photons may be used from valuable imagery into deep ultraviolet.

1.2 Challenges

Several steps are required to analyze the images / volumes of a fluorescence microscopy, detect and segment subcellular structures. For example, an exact segmented tube boundary can identify a single nephron in the kidney and characterize it. A precise detection and segmentation of nuclei can also be used to analyze tissue status.

There are several main challenges involved in microscopy imaging. For the following reasons, analysis of fluorescence microscopy images may be difficult:

- First of all, noise is a big issue in fluorescence microscopy which can be caused by non-ideal detectors. There are three main noises in the camera: dark current noise, photon noise and reading noise [15, 16]. Dark noise occurs when sensors detect thermal-emitted electrons. As integration time and charge-coupled device (CCD) chip temperature increase, heat-excited electrons are detected as noise. A CCD cooler can reduce dark noise. Photon noise, also known as shot noise, is caused by photon randomness. Although the light is uniformly emitted, the frequency of photons that arrive at a sensor is alone. Photon noise therefore causes Poisson noise. Note that fluorescence images may contain stronger Poisson noise due to the weak signal from fluorophores. Finally, read noise is generated by converting electrons detected by sensors into voltage and digitization. Second of all, 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. Also, point spread function (PSF) from a microscope system blurs the images. Therefore, the main task of image processing is to restore the fluorescence images and extract useful content information from the images.

As a consequence, images of fluorescence microscopy are degraded by noise and blurring during the acquisition of the image. When the ideal pixels intensity is x, the real pixel intensity of fluorescence microscopies is z. [17–19]:

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

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 μ and a variance of σ^2. As noted above, only a limited number of photons are received in the noise detector because of low fluorescence from photobleaching, low fluorophore concentrations, and short duration of exposure. In a microscope device, the PSF can blur volumes that can decrease the resolution. Consequently, because of noise and PSF, limits to subcellular structures may not be well defined.

- A fluorescence microscopy data set can display multiple structures and structures to detect and segment specific structures that must be distinguished. Multiple structures can be labeled in one fluorophore form. A phalloidine marking, for example, is both the tube basement membrane and the proximal tubular brush frontier [20]. In addition, the crosstalk [21] can be used in different color channels for image structures. Crosstalk may occur when fluorophores emit wavelength light. Segmentation can be highly challenging when several constructs are labeled or introduced.

- Different sizes, shapes, and intensities of may exist in subcells. Cores can have various sizes and shapes, for example, in fluorescence microscopic volumes. Many of the volumes of fluorescence microscopy are inhomogeneously large [20, 22], so a simple threshold that loses structures at volume boundary and catches noise in the volume center. A pre-processing step such as adaptive histogram equalization or inhomogeneous correction [22] may be needed to correct the inhomogeneous intensity.
- It is important to distinguish touching or overlapping of neighboring structures. For example, the number of nuclei in the fluorescence microscopy volume must be counted by separating the contact or overlapping nuclei. Structures can be analyzed individually by separating structures. The [23] watershed technique that can be used to separate touching objects tends, because of irregular structural forms, to over-segment kernels. The separation of connected nuclei is still an open problem for research.

1.3 Notation

To easily describe our various processes, a general notation is introduced and used for this study for the microscopic volume analysis. The fluorescence data set can be expressed in 5D with a width (x), height (y), depth (z), time (t), and color channel (c), without loss of generality. For detailed 5D volume information, we use subscriptions for the 5D volume indices and superscripts. We denote I_{z_p,t_m,c_n} as a 2D grayscale image size of $X \times Y$ in p^{th} focal plane image along z-direction in

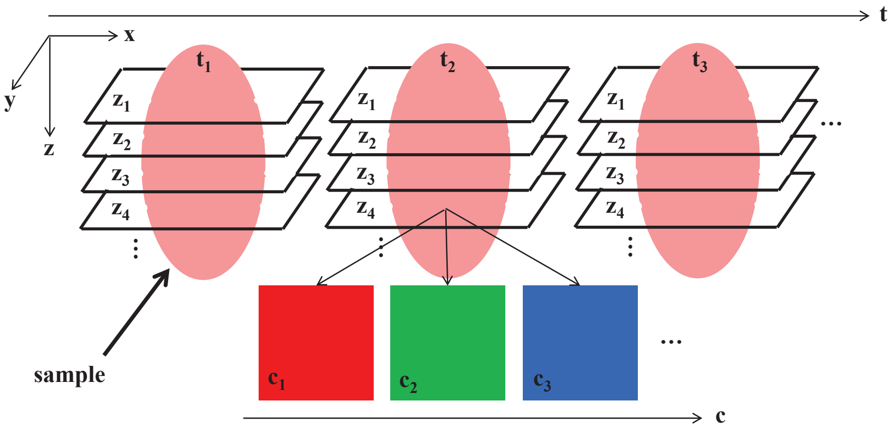
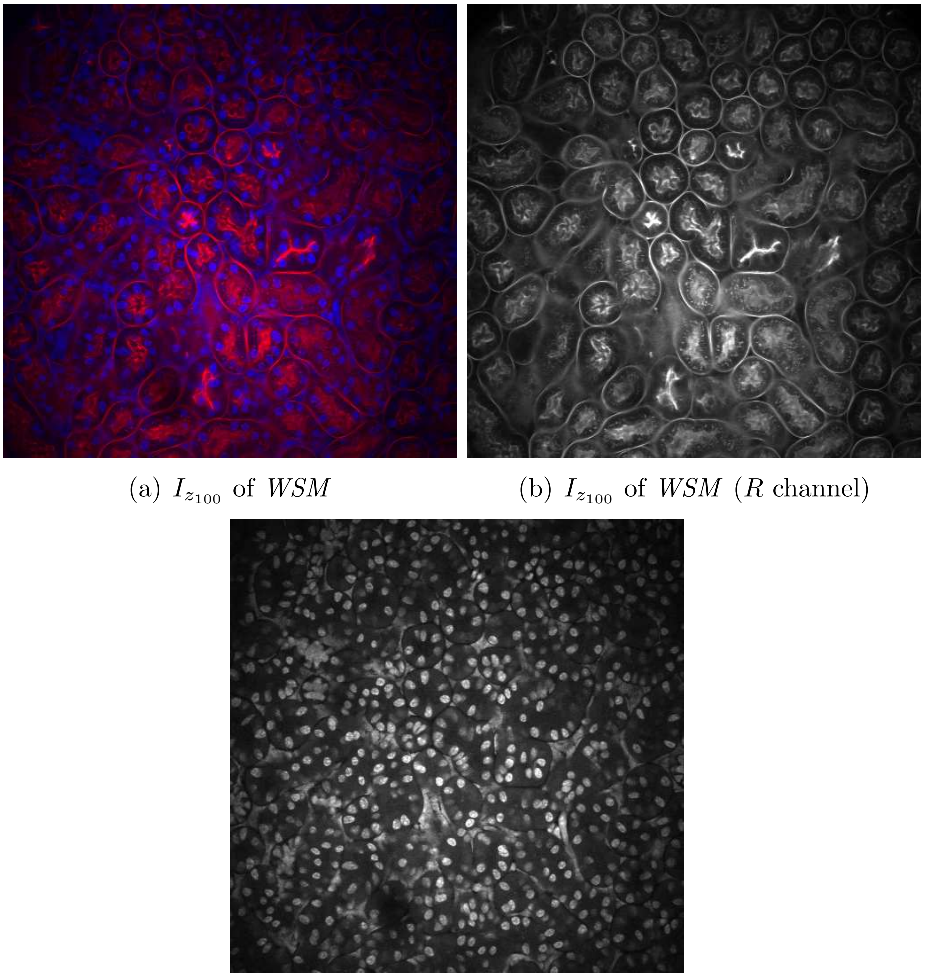


Fig. 1.4.: The notation for the microscopy volume in this thesis

a volume, the m^{th} time sample, and the n^{th} color channel, where $p \in \{1, \ldots, Z\}$, $m \in \{1, \ldots, T\}$, and $n \in \{1, \ldots, C\}$, respectively. Here, X and Y are image width and height, Z is the number of focal planes. Likewise, T is the sample number and C is the number of channels for color. Here, the original volume of fluorescence microscopy is a single grayscale volume. However, biologists often obtain volumes with multiple fluorescences that simultaneously highlight various structures. The volumes are then saved with different colored channels. In this study, we always separate each channel to obtain a 3D grayscale volume with structure for observation. In addition, we use a single timestamp for each 3D volume obtained. Therefore, for brevity, I can be defined as a $X \times Y \times Z$ 3D image dimension. The volume could be formed as a stack of multiple images in z-, y-, and x-direction. Therefore, we denoted I_{z_p} as a xy section with p^{th} focal plane along the z-direction in a volume, where $p \in \{1, \ldots, Z\}$. Similarly, I_{y_q} is a xz section with q^{th} focal plane along y-direction, where $q \in \{1, \ldots, Y\}$, and I_{x_r} is a yz section with r^{th} focal plane along x-direction, where $r \in \{1, \ldots, X\}$. For example, I^{orig}_{z23} is the 23rd focal plane image of an original volume, I^{orig}. In addition, let $I_{(r_i:r_f, q_i:q_f, p_i:p_f)}$ be a subvolume of I, whose x-coordinate is $r_i \leq x \leq r_f$, y-coordinate is $q_i \leq y \leq q_f$, and z-coordinate is $p_i \leq z \leq p_f$, where $r_i, r_f \in \{1, \ldots X\}$, $q_i, q_f \in \{1, \ldots Y\}$, and $p_i, p_f \in \{1, \ldots Z\}$. For example, $I^{seg}_{(241:272,241:272,131:162)}$ is a subvolume of segmented volume, I^{seg}, where the subvolume is cropped between 241^{st} slice and 272^{nd} slice in x-direction, between 241^{st} slice and 272^{nd} slice in y-direction, and between 131^{st} slice and 162^{nd} slice in z-direction.



(a) $I_{z_{100}}$ of *WSM*

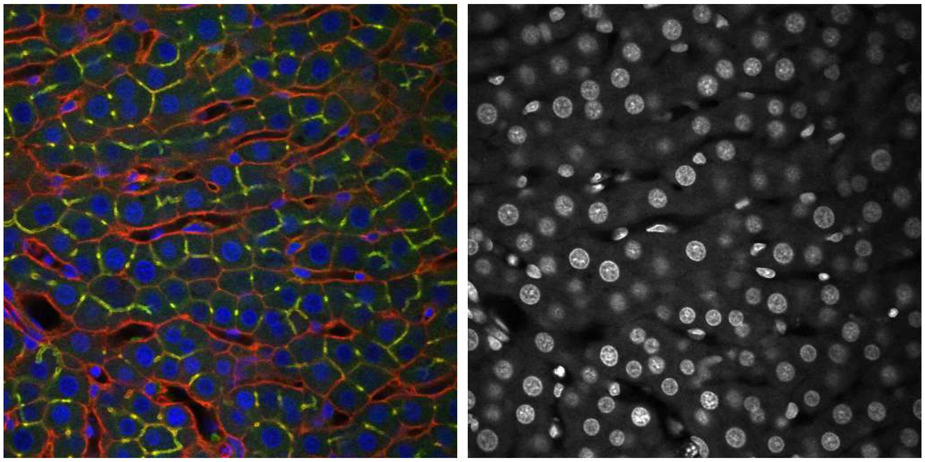
(b) $I_{z_{100}}$ of *WSM* (R channel)

(c) $I_{z_{100}}$ of *WSM* (B channel)

Fig. 1.5.: Sample images of *WSM* dataset

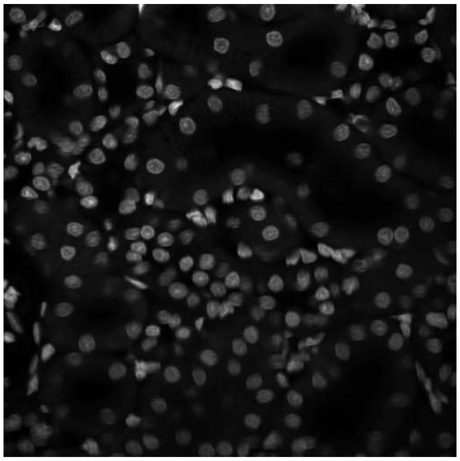
1.4 Data Sets

In this thesis, different methods are shown to analyze volumes of multiple microscopy image volumes, with a focus on segmentation and counting. Our data sets are acquired mainly by multiphoton microscopy. Intravital microscopy has been able



(a) $I_{z_{16}}$ of *immuno* (b) I_{z_{16},c_3} of *immuno* (B channel)

Fig. 1.6.: Sample images of *immuno* dataset



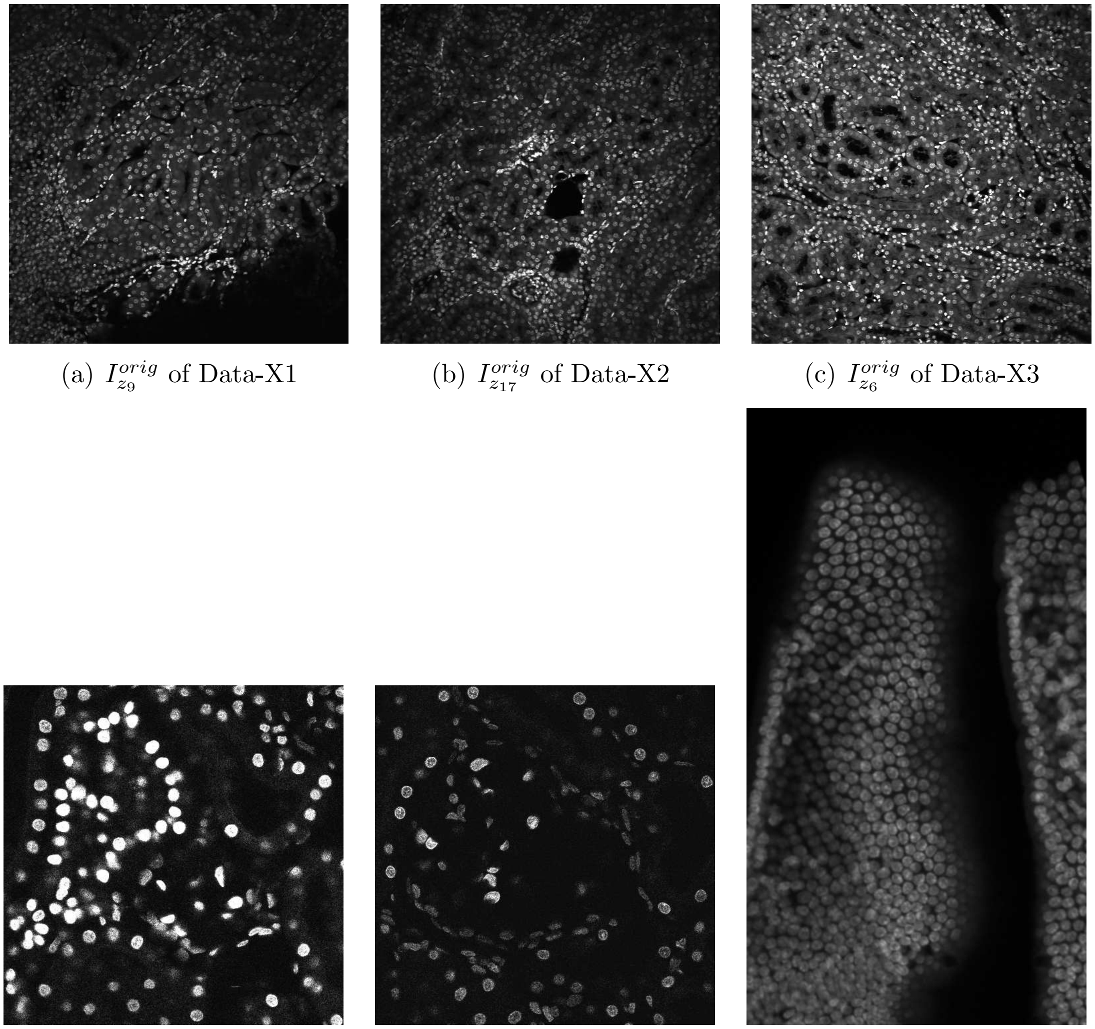
(a) $I^{orig}_{z_{403}}$ of *3color*

Fig. 1.7.: Sample images of *3color* dataset

to observe the physiological processes in cells with recent development of microscopy techniques such as confocal and microscopic microscopy. Due to this huge and complex collection of data, digital image processing has become a major component.

Different 3D microscopy volume datasets are used to test proposed segmentation and counting methods. Sample images of *WSM* [1] are shown in Figure 1.5 which each

[1] *WSM* dataset were provided by Malgorzata Kamocka of the Indiana Center for Biological Microscopy.



(a) $I_{z_9}^{orig}$ of Data-X1

(b) $I_{z_{17}}^{orig}$ of Data-X2

(c) $I_{z_6}^{orig}$ of Data-X3

(d) $I_{z_{16}}^{\mathrm{orig}}$ of Data-X4

(e) $I_{z_9}^{\mathrm{orig}}$ of Data-X5

(f) $I_{z_{137}}^{orig}$ of Data-X6

Fig. 1.8.: Sample images of data sets containing nuclei structures

of datasets are acquired from a 3D volume of rat kidney. Sample images of *immuno* dataset[2] is displayed in Figure 1.6. Along with *WSM* dataset this *immuno* dataset is used for testing our segmentation and visualization including color labeling. In Figure 1.7, an example slice of dataset *3color* [3]is displayed.

[2] *immuno* dataset was provided by Kenneth W. Dunn of the Indiana University School of Medicine.

[3] *3color* dataset was provided by Sherry Clendenon collected while at the Indiana Center for Biological Microscopy

For some additional examples, Figure 1.8 shows image slices from Data-X1, Data-X2, Data-X3, Data-X4, Data-X5, Data-X6 are datasets collected from rat using two-photon microscopy where nuclei are labeled with Hoechst 33342 or Dapi. The goal is to segment nuclei by rejecting other subcellular structures. [4]

1.5 Contributions of This Thesis

There are three main applications in fluorescence microscopy image analysis: restoration, registration, segmentation. Fluorescence microscopy image segmentation is challenging due to low contrast and inhomogeneous intensity in images or stacks potentially caused by photobleaching and noise. In addition, region boundaries are not well defined potentially caused by PSF. Here we discussed two different fields of microscopy image analysis including microscopy image synthesis and microscopy image segmentation. Microscopy image synthesis, segmentation and counting methods using generative adversarial networks are discussed in Chapter 3 and Chapter 4. A microscopy image synthesis and segmentation method is described in Chapter 5. A microscopy image analysis system is described in Chapter 6.

The main contributions of this thesis are listed below:

- A nuclei counting technique using a Spatial-Constrained Cycle-Consistent Generative Adversarial Networks (SpCycleGAN) and a 3D Generative Adversarial Networks (GAN) counting network is introduced. The 3D counting network is trained with synthetic microscopy data generated by the SpCycleGAN and tested with both synthetic and real microscopy data. A watershed-based counting method with Convolution Neural Network (CNN) based nuclei segmentation is also described. This method is consisting of two stages, a CNN nuclei segmentation with majority voting refinement and watershed nuclei counting. Our method achieves higher accuracy in terms of voxel evaluation.

[4] Data-X1, Data-X2, and Data-X3 were provided by Tarek Ashkar of the Indiana University School of Medicine. Data-X4, Data-X5 were provided by Kenneth W. Dunn of the Indiana University School of Medicine. Data-X6 was provided by Mike Ferkowicz of the Indiana University School of Medicine.

- A method for nuclei segmentation and detection with boundary refinement is also described. This method consists of three stages, a synthetic image volume generation step, a segmentation step, and a boundary refinement step. The generation of realistic synthetic data is extended from ReCycleGAN [24] which is initially developed for video to video style transfer.

- A method for synthetic microscopy image generation also described. This method consists of two stages, a synthetic image volume generation step and a segmentation step. The generation of realistic synthetic data is extended from style transfer method known as arbitrary style transfer in real-time with adaptive instance normalization.

- A Distributed and Networked Analysis of Volumetric Image Data (DINAVID) system is developed to enable remote analysis of microscopy images for biologists. There are two main functions integrated in the system, a 3D visualization tool and a remote computing tool for nuclei segmentation. The 3D visualization enables real-time rendering of large volumes of microscopy data. The segmentation tool provides fast inferencing of pre-trained deep learning models trained with 5 different types of microscopy data.

1.6 Publication Result From This Work

Journal Papers

1. K. W. Dunn, C. Fu, D. J. Ho, S. Lee, **S. Han**, P. Salama, and E. J. Delp, "DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data," *Scientific Reports*, Volume 9, Article number: 18295, December 2019. DOI: 10.1038/s41598-019-54244-5

Journal Papers In Preparation

1. E. J. Delp, **S. Han**, A. Chen, S. Lee, C. Fu, C. Yang, L. Wu, P. Salama, K. W. Dunn, and P. Salama, "DINAVID: A High-Performance Distributed and Networked Image Analysis System for Volumetric Image Data," *To be submitted*

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.*

3. S. Lee, **S. Han**, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Three dimensional tubule synthesis and segmentation for fluorescence microscopy using generative adversarial networks," *To be submitted to the IEEE Transactions on Medical Imaging.*

Conference Papers

1. **S. Han**, S. Lee, A. Chen, C. Yang, P. Salama, K. W. Dunn, and E. J. Delp, "Three Dimensional Nuclei Segmentation and Classification of Fluorescence Microscopy Images," *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, April 2020, Iowa City, Iowa.

2. **S. Han**, S. Lee, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Nuclei counting in microscopy images with three dimensional generative adversarial networks," *Proceedings of the SPIE Conference on Medical Imaging*, pp. 109492Y-1-11, February 2019, San Diego, CA.

3. 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,” *Proceedings of the IEEE International Conference on Biomedical and Health Informatics (BHI)*, May 2019, Chicago, IL.

4. S. Lee, **S. Han**, P. Salama, K. W. Dunn, and E. J. Delp, “Three Dimensional Blind Image Deconvolution for Fluorescence Microscopy Using Generative Adversarial Networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, April 2019, Venice, Italy.

5. 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*, June 2018, Salt Lake City, UT.

6. C. Fu, D. J. Ho, **S. Han**, P. Salama, K. W. Dunn, 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.

2. LITERATURE REVIEW

By developing and advancing fluorescence microscopy, biologists can see subcellular structures of living cells. Fluorescence microscopy generates a lot of microscopic images and stacks and these images and stacks need to be analyzed and quantified. Due to the large amount of data, however, it is not desirable to analyze and quantify the data manually in order to have an unbiased and effective process. Image and signal processing can be a great help in analyzing fluorescence microscopy images and stacks automatically. Automatic analysis of fluorescence microscopy images is challenging, as the images contain noise and blurriness, are not well aligned in z-direction and have inhomogeneous intensities between the center and the image boundary. Therefore, quantitative analysis of data from fluorescence microscopy, noise and blurriness reduction restoration, alignment registration in z-direction and segmentation for the partitioning regions of interest is required.

Common techniques for microscopy images analysis starts from identifying the region of interest. For region of interest that is in large 3D volumes, manual delineation is impossible. When a specific feature of the images is desired by biologists but a mathematical description is often not directly available, automatic image segmentation methods are of increasing demands in those cases. The use of deep learning based approach has become more popular with its ability to learn from the data. A set of user created training data is provided to train the deep learning models. This set of training data is usually not readily available as the desired type of outcome images need to be delineated by the user first. For example, a binary mask or the boundaries of the nuclei in 3D microscopy volumes need to be first defined in order to train a deep learning model to segment nuclei. To replace the need for hand annotation of images, synthetic images that can realistically describe the experimental data can be used. Generative adversarial networks come in handy for the creation of synthetic images.

2.1 Review of Nuclei Segmentation Methods

Fluorescence microscopy has become an important instrument in the investigation of cell biology, allowing 3D image volumes to be acquired from deep into the tissue and the visualization of complex subcellular structures. Nuclei segmentation enables the quantitative study of those structures that are important for characterizing the structure and creation of tissue volumes. Nevertheless, due to the size and complexity of data, manual segmentation is a laborious and unworkable operation.

Many different techniques of segmentation were developed. This included widely used techniques of active contours or snakes, such as Chan-Vese 2D region-based active contours model for the segmentation of the two-dimensional nuclei [25], which in [26] was expanded to 3D segment of cell-structures for rat kidney.

A fully automatic segmentation method with multi-level setting functionality with a penalty term and a volume conservation constraint for the separation of touching cells was describe in [27]. It was improved by a watershed approach for initialization, a non-PDE-based energy minimization for effective computing, and the Radon transformation to separate touching cells in [28]. In [29], a discrete multi-regional competition method was used where the number of regions is uncertain. Another image segmentation technique named Squassh was developed by [30, 31], which used an energy function derived from a generalized linear model.

When more subcellular structures are present, these techniques do not segment nuclei. A nuclei segmentation method has been developed to resolve this [32], which detects primitive nuclei boundaries, and uses regionally expanding to delineate nuclei. A technique based on midpoint analysis, distance functions for shape fitting and marked point processes (MPPs) has recently been described by [33]. Both [32] and [33] are 2-dimensional segmentation approaches which do not use z-directional information on volumes even though they can produce good results. The added z-directional information has increased the difficulty of the study.

Two or more nuclei are frequently found near or sometimes overlap in biological samples. In addition, in the present focal plane, photons emitted from nuclei can still be observed in an elongated and clustering nuclei. When fluorescent signal is emitted in all directions from each nucleus, focal objects appear to be produced in the vicinity of the nuclei. It will lead to expanded and overlapping segmentation areas. To fix this, the [34] watershed approach is widely used for demarcating the individual nucleus after processing. 3D watershed [23, 34] may be used as the post-processing technique to separate individual nucleus as watershed separates several overlapping items effectively. For instance, [35] uses a marker-controlled watershed to segment the nuclei.

Watershed generates a distinct label for each nucleus by finding local minima in the topographical distance transform [34]. If it is used on an original volume, local minima will be assigned not only to nuclei but also to other structures. Since the generated segmentation mask contains only the segmented nuclei, the watershed demarcates overlapping nuclei and labels adjacent nuclei. The method enables clear labeling of adjacent nuclei which is visually helpful for analyzing the images. The total number of nuclei in the volume can thus be estimated based on the number of labels. This information is particularly useful for analyzing properties such as cell livability for biological studies.

Recent deep learning approaches have been more used in biological volumetric analysis: [36–38]. Although deep learning is generally computationally intensive, the success of graphical processing units (GPUs) has reduced the execution times, which can be attributed to a renewed interest for deep learning. In several computer vision applications, such as object detection, human detection, or autonomous driving, Deep Neural Network (DNN) has been demonstrated to be helpful. The use of neural convolution networks (CNN), [39–42], forms the basis for a popular deep learning approach to image classification and segmentation. In [43] a max-pooling layer CNN is used for segment electron microscopy neuron membranes and in [44] a broad field microscopy CNN is used to detect tyrosine Hydroxy-containing cells of zebra-fish photos. The latter was used to help the collection of CNN training data with a Support Vector Machines classification [45].

In [46], this network named SegNet had the first objective as scene understanding, although its implementation is expendable. A semantic pixel wise segmentation deeply CNN was created. The innovation of SegNet is the way to higher resolution performance from the lower resolution input function charts. In the maximum pooling step of the corresponding encoder upsampling, the decoder explicitly uses pooling indexes calculated for non-linear efficiency. The segmentation network consists of an encoder, an encoder and a classification layer.

A combination of several convolutional layers with pooling layers, [47], and a rectified-linear unit (ReLU) activation function is used for each encoder layer. In order to maintain the images' original sizing, each convoluted layer performs the convolutional operation with a kernel of $3 \times 3$with 1pixel padding. Images are sampled in the pooling layer for a maximum of 2×2 max phase pooling operation. Conversely, a decoder consists of an up-sampling layer and multiple convolutional layers, which have the same convolutional layers as the encoder. The architecture's final layer is used to evaluate the likelihood that a pixel is or is not in the nucleus by a softmax classifier.

U-Net has also been used for segmentation of microscopic images, a completely integrated convolutional network [48]. Data augmentation using elastic deformations to train CNN architecture with a limited amount of training images has been developed. An extension of a [36] 3D U-Net was used to prevent 3D, dense segmentation by manual 2D sparse annotation preparation. The 3D density segment has been expected. Similarly, U-Net, a fully connected convolutional network [48], has been utilized for segmentation of microscopy images. Due to the lack of training data [48] developed data augmentation methods using elastic deformations to train the CNN architecture using a small number of training images. An extension of [48], a 3D U-Net [36] was used to generate 3D dense segmentation by using manual 2D sparse annotations. In addition, [49] used a deep CNN based shape initialization for the nuclei segmentation of histopathological images, while [50] developed a spatially restricted convolutional neural network (SC-CNN) which makes a probability map by using the distance from the center. A segmentation method using triplanar CNN [51] is represented by the independence and fusion into the final layer of three two-dimensional CNNs on horizontal, frontal and sagittal planes. Yet the training of individual CNNs will be computationally expensive. In [36], an extension of 2D U-Net [48] is defined as the 3D U-Net for volumetric segmentation. Although [36] 3D U-Net uses 3D operations to use volumetric data entirely, 2D slices must be annotated by hand.

2.2 Review of Synthetic Data Generation with Deep Neural Networks

Style transfer can render a content image in the style of another image. A style transfer method should understand and capture the style of given set of training images. Then, the method should be able to produce high-quality texture images on the basis of test images, but they may use a collection of training images to provide the final result with test image characteristics with trained styles. To achieve this transition, [52] uses CNN networks which, similar to the encoder-decoder structures, use VGG network [53] to minimize the average pixel differences between two generated Gram matrix represented in the encoder layer and corresponding layer in the decoder process. The CNN networks are stacked similar to encoder-decoder structures. Generative adversarial network (GAN)s have become an increasingly common model to learn how to produce loss functions along with the model itself because of [54]'s paper. Adverse loss models have been applied to a broad range of applications, including image generation, domain adaptation, text-to-image synthesis, 3D forms-to-structure synthesis, image-to-image translation, super resolution and even particle physics generation of radiation patterns. These models, however, suffer from problems such as mode crashes and training oscillations that make them difficult to use in practice. These problems are currently being addressed by the community in several ways. The architecture of the network was carefully optimized and best practices developed to optimize training were made extensive. The GAN might have a problem of not converging between the discriminatory and the generative networks. In order to overcome this problem, [55] demonstrated a deep convolutional GAN (DCGAN), which employs logarithms to increase network convergence. In addition to the Jensen-Shannon divergence distance, the Wasserstein GAN (W-GAN) [56] uses Earth-Mover distance to improve its convergence in training.

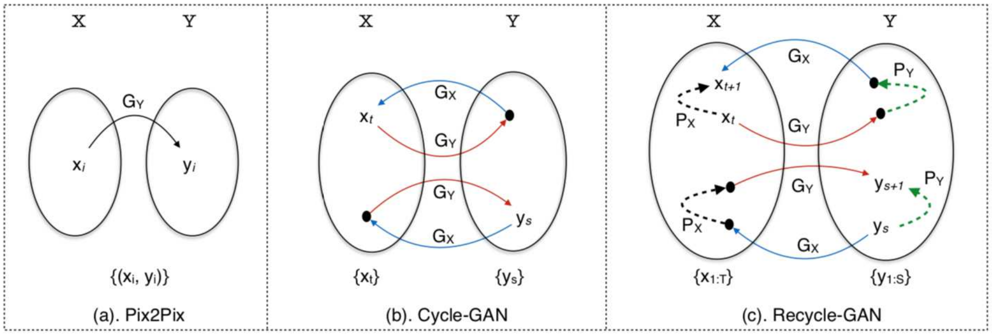


Fig. 2.1.: Comparison of Pix2Pix, CycleGAN, and ReCycleGAN [24]

Generating realistic synthetic microscopy image volumes still remain an onerous issue because a good model must be modeling different types of noise and biological structures of various kinds. A generative adversarial network (GAN) has recently been defined to resolve the image-to-image translation issues by means of two adversarial networks, a generative network and a discriminatory network.

New data similar to the expected one will be generated by the generator. The discriminator's objective is to determine whether an input data is real (it belongs to the original dataset) or false (generated by a forging). The discriminative network, in particular, learns a loss function to decide whether a real or false output image while the generative network attempts to minimize the loss function. In traditional GANs, a random image can be generated with a random code input, however, the level of user control and the resolution of the generated images are not ideal. Pix2Pix [57] is one of the GAN extensions that use conditional GANs to understand the correlation between the input and the output image that can produce realistic images. Since the input of Pix2Pix is user-defined images, the level of user control has greatly improved. For example, Pix2Pix is capable of transform line drawing of a cat into a photo-like image of cat. The problem with Pix2Pix [57] is that the training of networks still needs paired data. Later, cycle-consistent adversarial networks (CycleGAN) [58] used a cycle consistent term for adverse loss function without using paired training data for image generation. In addition, a space-restricted CycleGAN (SpCycleGAN) has been implemented to regularize the position of synthetic nuclei in order to boost nuclei segmentation efficiency. [59] Recently, a network called ReCycleGAN is introduced, which incorporates spatial and temporal information with adversarial losses in content translation and style preservation.

A comparison of Pix2Pix, CycleGAN, and ReCycleGAN is illustrated in Figure 2.1. Pix2Pix contains one generative model G_Y that can transfer image x_i in domain X into image y_i in domain Y. x_i and y_i are a set of paired images. CycleGAN contains two generative models, G_X that can transfer image y_s in domain Y into image y_t in domain X. G_Y that can transfer image x_t in domain X into image y_s in domain Y. x_t and y_s are a set of unpaired independent images, where x_t and y_s are not ordered in spatial or temporal domain. The introduction of cycle-consistency enables image-to-image translation tasks without any expensive labeling by hand. ReCycleGAN considered the case of unpaired but ordered streams $(x_1, x_2, ..., x_t, ...)$ and $(y_1, y_2..., y_s, ...)$. ReCycleGAN contains two generative models, G_X that can transfer image y in domain Y into image x in domain X. G_Y that can transfer image x in domain X into image y in domain Y. In addition, recurrent temporal predictor P_X and $P_Y$is trained to predict future samples in a stream given its past. P_X can predict x_{t+1} given $(x_1, x_2, ..., x_t, ...)$. P_Y can predict y_{s+1} given $(x_1, x_2, ..., x_s, ...)$. The recurrent loss is introduced to form better mapping by taking advantage of the temporal or spatial ordering. ReCycleGAN has been successfully used for the style transfer of video to video with continuous movement, for example, flower opening. An impressive feature is that there is no discontinuous artifact between each frame in time series.

2.3 Microscopy Image Analysis and Visualization Tools

Several popular tools are available among the biomedical community for the assisted analysis of medical images. Here are some examples of the tools that can segment nuclei in 3D microscopy image volumes:

- CellProfiler 3.0 [60]: CellProfiler is a user-oriented open source program to analyze biological images. CellProfiler has been developed to analyze images, the size, shape, intensity and textures of any cell (or other object) on a scalable and flexible and high performance basis in each image. Using a graphical user interface (GUI), users create a “pipeline” for image analysis, with sequential modules that each can perform a task of the image processing, including illumination correction, segmentation and object measurement. Users can select, choose, and change their settings to test the value of interest. It is also suitable for low performance assays though originally designed for high-performance images.

  CellProfiler can easily extract useful biological data from photographs and improve objective and statistical intensity of tests. It assists researchers in quantitatively addressing a range of biological issues, including standard testing and various morphological assessments. CellProfiler works through the design and arrangement of usable modules by designing custom task-specific pipelines.

  The default settings used to constitute a typical pipeline for segmentation include the “IdentifyPrimaryObject” module. The optimized settings can be developed by adding pre-processing and post-processing modules to the default pipeline based on our observations of the results obtained by the default settings. The final settings are chosen from the combination and arrangement of modules that provide the best results.

- Squaash [30, 31]: Squassh is a ImageJ tool that can segment subcellular structures in microscopy images in 2D and 3D. Three parameters in Squaash can be adjusted to produce the visually best segmentation results. The first parameter is the "Rolling ball window size" from background subtraction. The second parameter is the "Regularization parameter" for segmentation. Higher values are used to avoid segmenting noise-induced small intensity peaks. The third parameter is "Minimum object intensity" for segmentation. High values are used to force object separation.

  The first background subtraction step is carried out as the segmentation model assumes constant local intensities. Context variations are non-specific signals that this model does not represent. With the rolling-ball algorithm, it subtracts the background image. The entire image is segmented. It assigns to every pixel a number from 0 to 1, which is an object or background pixel score. We use k-mean clusters to estimate object intensity for this original, rough segmentation to the maximum intensity in the image. The score mask threshold displays the initial artifacts.

  The threshold value is set to the minimum intensity of the artifacts to be included in the analysis, as specified by the user. Each item with less than this threshold intensity is discarded. Connected areas are marked as separate objects. Individual segmentation of objects is obtained separately by operating the algorithm per image region. This move can be accomplished with a segmentation resolution higher than the image resolution, thereby providing subpixel precision. As this over-sample is done in local patches around the objects instead of the entire image, the measurement costs have only a moderate impact.

The final segmentation is obtained by using the optimal threshold for the last point, which reduces the segmentation error, automatically defined as mentioned above.

- FARSIGHT [61]: FARSIGHT is a popular toolkit for medical image analysis, particularly for neuronal analysis in microscopy images. It features automated and computer-assisted method for tracing neural structures and for cell analysis. FARSIGHT is written in C++ and python, using Kitware's Visualization Toolkit (VTK) and the Insight Toolkit (ITK). FARSIGHT is designed for the handling, pre-processing, segmentation and inspection of image data. These modules can be used to perform a variety of automated image processing tasks.

  The core high level image processing algorithms implemented in FARSIGHT are build using the filters developed by the Insight Segmentation and Registration Toolkit (ITK) and the visualization is built mainly using The Visualization ToolKit (VTK). Importantly, all modules are accessible through the Python scripting language which allows users to create scripts to accomplish sophisticated associative image analysis tasks over multi-dimensional microscopy image data. For FARSIGHT, four parameters σ_{min}, σ_{max}, r_{xy}, and r_z are adjustable. Here, σ_{min} and σ_{max} are minimum and maximum scale values of the Laplacian of Gaussian filter. r_{xy} and r_z are used to define clustering behaviors in corresponding directions. FARSIGHT automatically estimates the values of these four parameters and denotes their values as the default setting.

- ImageJ Based Visualization Tools: ImageJ can be used to visualize image with the help of plugins. One example that can provide cross-sectional views is known as 3D Viewer [62]. Another example that is capable of 3D rendering is Voxx [63] which can display entire stacked images as a single volume and performs 3D rendering to visualize 3D objects. Similarly, Volume Viewer [64], does real-time rendering to display stacked images. These 3D rendered viewing methods can provide an interactive display of 3D volume so that a user can interactively investigate the 3D volume.

- ImageVis3D [65]: ImageVis3D has been developed by the NIH / NIGMS Integral Biomedical Computing Center (CIBC) as a new volume rendering program. Simplification, scalability and interactivity are key design objectives of ImageVis3D. A modern user interface that provides unprecedented versatility is easy. Simplicity is achieved. Scalability and interactivity mean users can explore terabyte-sized hardware data sets, from mobile devices to high-end graphics workstations interactively. Finally it is possible not only to expand the ImageVis3D itself but also to reuse portions of it like the rendering core by exploiting the open source nature and the strict component-to-piece architecture. ImageVis3Dis an example of a OpenGL volumetric rendering tool that uses Open Graphics Library (OpenGL) [66] ImageVis3D also supports other viewing methods including 2D slices viewing and maximum intensity projection (MIP) viewing.

- Apeer [67]: Apeer is a cloud-based system is Apeer [67] from Zeiss. Apeer provides image processing tools through the use of Python-based modules. Apeer is powered by Azure from Microsoft for cloud computation and Git for version control and code collaboration. Users can create their own workflow or choose to use pre-built workflows. In terms of building workflows, Apeer is similar to CellProfiler in the ways each module is added and executed sequentially.

- WIPP [68]: For the purpose of facilitating the analysis of large size image data, the Web Image Processing Pipeline (WIPP) [68] was developed by the National Institute of Standards and Technology (NIST), who recently reported the development of plugins for cloud-based microscopy image analysis [69]. WIPP provides users with modules for image processing, image feature extraction based on various image characteristics, and machine learning-based regions of interest

(ROI) clustering. WIPP is expanding its capability for analysis by adding a module for statistical modeling.

- WebGL based visualization tools: BioWeb3D [70] is a network-based 3D visualization tool that uses WebGL. The limitation is that the data can only be visualized with each pixel as particles or with simple lines drawn between points. 3D Cell Viewer [71] from the Allen Cell Institute is a more advanced example that uses WebGL and ray-marched path tracing for 3D volume rendering and provides comprehensive imaging parameter tuning and channel toggle functions. With preloaded segmentation of cells, 3D Cell Viewer allows users to choose to turn on or off the visualization of a cell based on its characteristics such as location and intensity. So far, 3D Cell Viewer is limited to preloaded volumes of segmented cells. Similarly, Neuroglancer [72] allows examining 2D and 3D visualizations of data from different kinds of microscopes. Neuroglancer is more suitable for developers who are familiar with web development for a customized display of data for their projects.

3. NUCLEI COUNTING WITH GENERATIVE ADVERSARIAL NETWORK

3.1 Background

Quantitative information obtained from microscopy images is vital for biomedical research and clinical diagnosis. For example, analyzing the distribution of cells or extracellular structures can provide a better understanding of the physiological and pathological status of the tissue. Two-photon microscopy is a type of fluorescence microscopy that is favored for living tissue imaging. With the use of near-infrared excitation to increase penetration depth in tissue, large image volumes in 3D are produced. The large 3D image sizes and the high cost of manual processing require automated means to analyze quantitative biological information [73].

Many challenges exist for counting objects in 3D image volumes, which include "crowding" and touching of objects, overlapping of two or more objects, and variances in object shape and size. There are many real-life examples where counting is needed, for example, traffic surveillance, pedestrian counting, and cell density estimation.

Several semi-automatic techniques have been proposed for counting. In [74], a supervised machine learning counting method that can estimate the object count with manual annotation input was introduced. Another method that uses machine learning to count objects and estimates the density of the objects in images was described in [75]. The ImageJ toolset [76] has a 3D object counter, known as JACoP [77], that is a subcellular colocalization analysis tool that uses a statistical approach with a manually selected threshold to analyze intensity information to obtain the object count and location. In some counting approaches, segmentation is done prior to counting in order to distinguish the counting targets (foreground) from the background. The 3D nuclei segmentation presented in [22] classifies each voxel in the microscopy image volume as nuclei or non-nuclei using 3D active contours. Typically the nuclei count can be obtained from the segmentation results using 3D connected component analysis. However, multiple nuclei could be counted as one nucleus if they are close enough to be connected. Thus, using segmentation for counting is less accurate when nuclei are crowded. One way to address this problem is to use morphological operations [78]. Morphological watershed (2D or 3D) is one of the techniques commonly used for separating overlapping objects [79, 80]. In the watershed, the topographical distance transform is used to find the number of local minima in the image, which indicates the object count [81]. However, a drawback of the watershed is over-segmentation when more than one local minimum is found for each object, producing inaccurate counting results [82].

Deep learning has recently become a prominent approach to address the counting problem [83]. A way of counting that uses convolutional neural networks (CNN) to estimate the number of pedestrians in a video was presented in [84]. Similarly, a tumor cell counting CNN is trained to provide both cell count and the cell locations in [75]. In another example, cell counting using fully convolutional regression networks (FCRNs) was introduced [85]. The networks were trained to find the locations of the cells from which cell count was obtained. A combination of a CNN nuclei segmentation and 3D watershed was used for 3D segmentation and nuclei counting in [86].

Acquiring image groundtruth for the training of deep learning networks is labor intensive and in many applications difficult to obtain. Synthetic data has been used to train and test the networks [87]. An approach for generating synthetic microscopy images in 3D was presented in [88]. Recently, generative adversarial networks (GAN) [54] have been useful for generating realistic synthetic data. There are two different networks in a GAN, a generative network and a discriminative network. More specifically, the generative network is trained to generate sample images whereas the discriminative network is used to estimate the probability of a generated sample being a real image. Both networks are trained to minimize their loss functions simultaneously.

A deep convolutional generative adversarial network (DCGAN) was described in [55] for unsupervised learning which uses a GAN together with a CNN for synthesizing images. Another example of using GANs to synthesize image is the cycle-consistent adversarial network (CycleGAN) [58]. However, the synthetic images generated by CycleGAN can be geometrically distorted which includes spatially shifting of objects [89]. More recently, 3D realistic synthetic microscopy volumes were generated by a spatially constrained CycleGAN (SpCycleGAN) and utilized with corresponding synthetic binary volumes to achieve 3D nuclei segmentation [59, 90].

In this work, we introduce a nuclei counting technique using two GANs, a SpCycleGAN and a 3D GAN for nuclei counting. The 3D GAN is trained with synthetic microscopy data generated by the SpCycleGAN and tested with both synthetic and real microscopy data. Our approach is evaluated with respect to the number of groundtruth nuclei and compared with common ways of counting used in the biological research. Fluorescence microscopy 3D image volumes of rat kidneys are used to test our 3D nuclei counter. The data contains fluorescent labeled (Hoechst 33342 stain) nuclei of kidney cells collected using two-photon microscopy.

3.2 Proposed Method

In this work, I denotes a 3D image volume of size $X \times Y \times Z$. Figure 6.1 shows the block diagram of our proposed way of nuclei counting. There are two main steps in our nuclei counter: 3D synthetic data generation and 3D nuclei counting. I^{orig} denotes a subvolume of the original microscopy volume used for training, and I^{test} is a subvolume of original or synthetic microscopy volume that is used for testing. During the evaluation, I^{test} is used as an input to the 3D nuclei counting network to estimate the number of nuclei, N. I^{syn} denotes a synthetic microscopy volume generated with 3D synthetic data generation based on the features of I^{orig}. I^{dist} is a distance map volume that contains information regarding the locations of the nuclei of the microscopy image volumes. Here, I^{dist} is paired with I^{syn} to train the

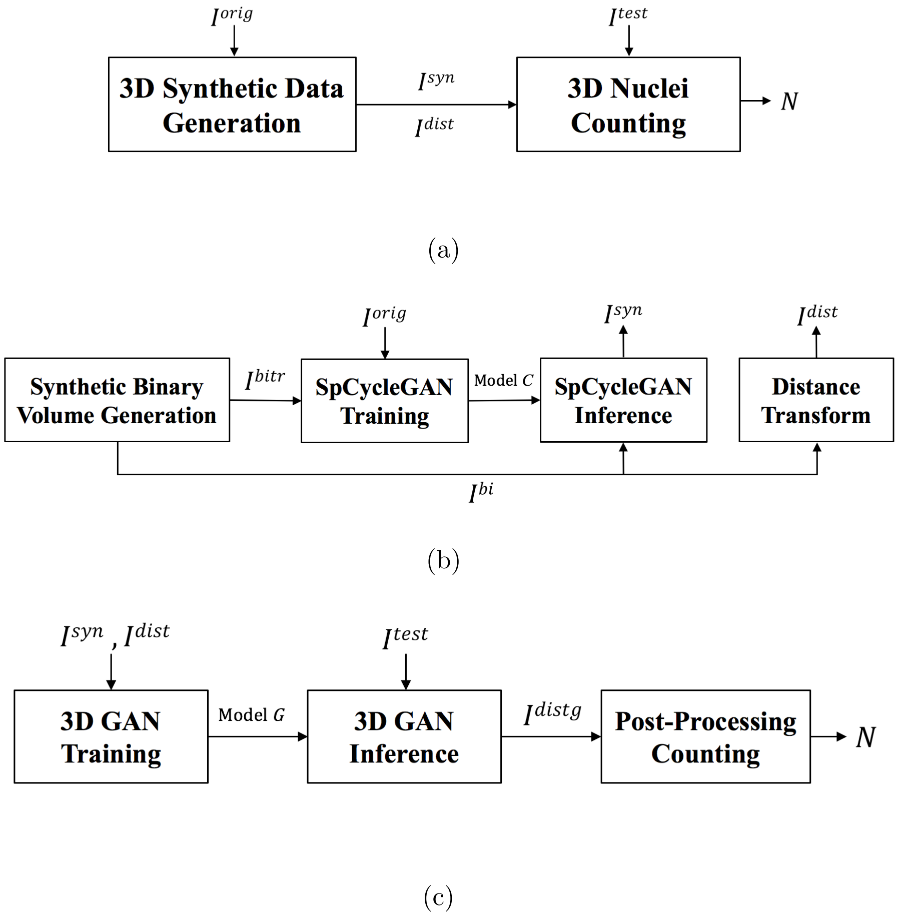


Fig. 3.1.: Block diagrams of (a) the proposed nuclei counter, (b) 3D synthetic data generation, (c) 3D nuclei counting

3D nuclei counting network. Figure 3.1(b) and Figure 3.1(c) are the detailed block diagrams for 3D synthetic image generation and 3D nuclei counting, respectively. I^{bitr} and I^{bi} denote the synthetic binary volumes used for SpCycleGAN training and inference, respectively. I^{distg} is a synthetic distance map volume generated by 3D nuclei counting network that contains nuclei location information of I^{test}.

As shown in Figure 3.1(b), synthetic binary volumes, I^{bitr}, are generated first. I^{bitr} is used together with I^{orig} to train a SpCycleGAN from which a generative model C is obtained. A different synthetic binary volume, I^{bi}, is then used to generate corresponding I^{syn} with the trained model C. For each I^{syn}, the corresponding I^{dist}

is generated. In Figure 3.1(c), paired I^{syn} and I^{dist} are used to train a 3D GAN for nuclei counting and obtain a 3D GAN generative model G. Model G is used to transform I^{test} to its synthetic distance map volume, I^{distg}, for counting purposes. Then, post-processing steps which include thresholding and connected component analysis are used on I^{distg} to obtain the number of nuclei N.

3.2.1 3D Synthetic Data Generation

Three dimensional synthetic data generation includes synthetic binary volume generation, distance transformation of the synthetic binary volume, SpCycleGAN training, and SpCycleGAN inference. A synthetic binary volume is generated by drawing ellipsoidal shapes in 3D according to randomly selected locations of nuclei centroids and their orientations.

Following synthetic binary volume generation, the 3D Euclidean distance transform [91] is used to obtain the distance transformation of nuclei. We denote the distance transform as a function $F_{DT} : I^{bi} \rightarrow I^{dist}$. The I^{dist} volume is obtained by the distance transform of the binary synthetic nuclei volume I^{bi}. F_{DT} assigns a number to each voxel of the binary volume that is the distance from the voxel to its nearest background voxel. Considering an individual nucleus in I^{bi} and its corresponding I^{dist}, the voxels close to the center of the nucleus have high distance values and the voxels at the boundary have low distance values. Thus, I^{dist} can be considered as a distance map of the centroid locations of nuclei.

The SpCycleGAN described in our work [59] was an extension of the CycleGAN [58] by considering spatial loss during the training. The goal is to obtain a generative model C that can transfer a binary synthetic nuclei volume to a synthetic microscopy volume with accurate nuclei locations. During SpCycleGAN training, the objective function is to solve the minimax problem where the training loss ($\mathcal{L}$) consists of the sum of an adversarial loss, a cycle consistency loss ($\mathcal{L}_{cyc}$), and a spatial loss ($\mathcal{L}_{spatial}$).

The cycle consistent term and spatial constrained term are used in the training loss for regularization of the SpCycleGAN model. The training loss can be expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda_1 \mathcal{L}_{cyc} + \lambda_2 \mathcal{L}_{spatial} \tag{3.1}$$

where λ_1 and λ_2 are the adjustable coefficients for $\mathcal{L}_{cyc}$ and $\mathcal{L}_{spatial}$. To find the spatial loss, a separate generative model H is trained by an additional generative network. Model H can generate a binary mask from a synthetic volume generated by model C. More specifically, model H uses $C(I^{bitr})$ as an input and generates $H(C(I^{bitr}))$. This generative model H minimizes $\mathcal{L}_2$ loss between I^{bitr} and $H(C(I^{bitr}))$. The spatial loss is then formed as:

$$\mathcal{L}_{spatial}(C, H, I^{orig}, I^{bitr}) = \mathbb{E}_{I^{bitr}}[||H(C(I^{bitr})) - I^{bitr}||_2] \tag{3.2}$$

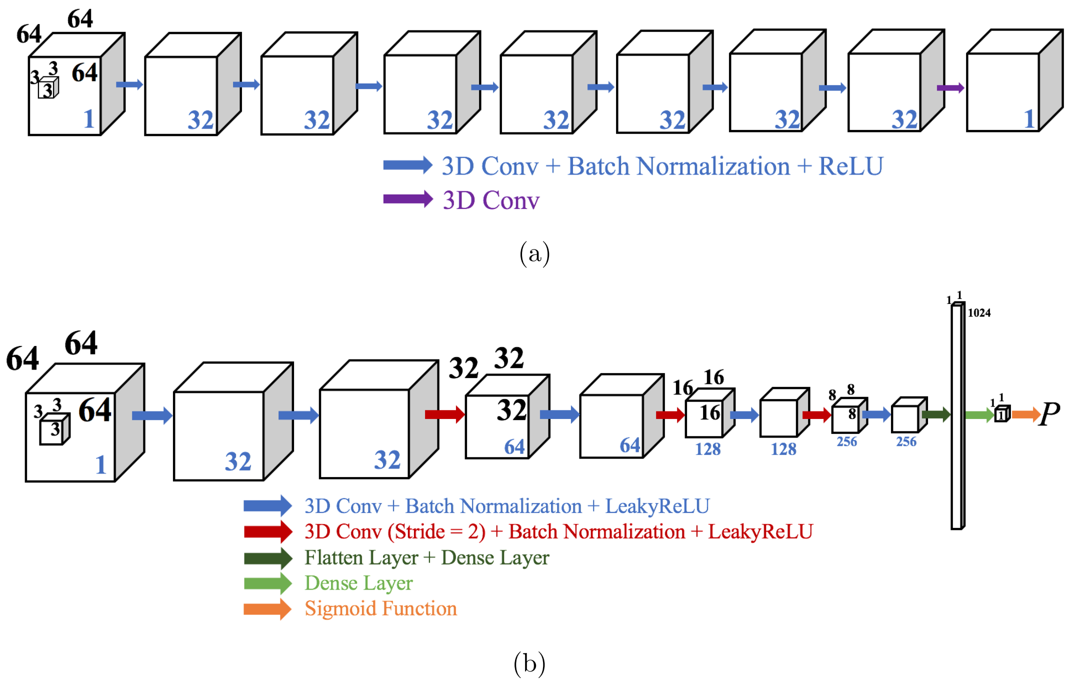
where $|| \cdot ||_2$ denotes the L_2 norm. Note that an unpaired set of training data, I^{orig} and I^{bitr}, is used for the training of model C.

During SpCycleGAN inference, the synthetic microscopy volume, I^{syn}, is generated from model C with the input as the synthetic binary volume, I^{bi}. Then, I^{bi} is used to obtain the distance map volume, I^{dist}, using distance transformation. Since the synthetic microscopy volume, I^{syn}, and the distance map volume, I^{dist}, are obtained from the same synthetic binary volume, I^{bi}, I^{syn} and I^{dist} are now a paired set of volumes and can be used for 3D nuclei counting network training.

3.2.2 3D Nuclei Counting

Three dimensional counting consists of 3D GAN training, 3D GAN inference, and post-processing counting. As shown in Figure 3.1(c), we utilize a paired set I^{syn} and I^{dist} to train a 3D GAN and obtain a generative model G. Here, the groundtruth distance map volume $I^{dist} = F_{DT}(I^{bi})$ provides information on the locations of the nuclei.

The 3D GAN generative model G is used to map nuclei in I^{test} to its synthetic distance map volume I^{distg}. This approach has the advantage over learning a direct mapping between original microscopy volume and nuclei count since the distance map volume preserves nuclei location information. The number of nuclei is then obtained from I^{distg} using thresholding and 3D connected component analysis. Thresholding of I^{distg} is done to keep voxels at or near the nuclei centroids and remove noises. By removing low distance transform voxels around nuclei boundaries, two or more connected nuclei can be identified as individual nuclei. Then, a 3D connected component analysis is done to assign different colors to individual nuclei at their centroid locations followed by obtaining N. The detail for post-processing color coding is available in [92] and [93].


64
64
3
3
3
64
1
32
32
32
32
32
32
32
1
3D Conv + Batch Normalization + ReLU
3D Conv
64
64
3
3
3
64
1
32
32
32
32
32
64
64
16
16
16
128
128
8
8
8
256
256
1
1
1024
1
1
1
P
3D Conv + Batch Normalization + LeakyReLU
3D Conv (Stride = 2) + Batch Normalization + LeakyReLU
Flatten Layer + Dense Layer
Dense Layer
Sigmoid Function

(a)

(b)

Fig. 3.2.: Architecture of 3D GAN for nuclei counting (a) generative network structure, (b) discriminative network structure

The architecture of the 3D GAN for nuclei counting is shown in Figure 3.2. For the generative network in Figure 3.2(a), a filter size of $3 \times 3 \times 3$ is used for the 3D convolution. Each 3D convolution is followed by a batch normalization [47] and a rectified-linear units (ReLU) activation function. Similarly, for the discriminative network shown in Figure 3.2(b), a batch normalization and a Leaky ReLU activation function are used to follow each 3D convolution layer. Convolutions (stride 2) are used in replacement of pooling layers, while the number of features is doubled. Finally, a flatten layer, two dense layers, and a sigmoid activation function are used to produce the probability of classification P. This probability is used to discriminate groundtruth distance map volume I^{dist} from the generated synthetic distance map volume $G(I^{syn})$. The architecture of the 3D GAN for counting is designed to have a lightweight generative network for efficient inference. The discriminator is used to supervise the prediction of the 3D GAN for nuclei counting to have the features of a real distance map.

The generative model G learns to transform I^{syn} into I^{dist}, whereas the discriminative model D distinguishes between I^{dist} and $G(I^{syn})$. As shown in Equation 3.3, the training loss function of the 3D GAN for nuclei counting is defined as the sum of an adversarial loss $\mathcal{L}_{\text{GAN}}$ (Equation 3.4) and a content loss $\mathcal{L}_{\text{MSE}}$ (Equation 3.5). Here, we used the mean square error between the synthetically generated distance map volume $G(I^{syn})$ and the groundtruth distance map volume I^{dist} as the content loss. λ is the weight coefficient used to control the relative importance of the adversarial loss and the content loss.

$$\mathcal{L}(G, D, I^{syn}, I^{dist}) = \mathcal{L}_{\text{GAN}}(G, D, I^{syn}, I^{dist}) + \lambda\mathcal{L}_{\text{MSE}}(G, I^{syn}, I^{dist}) \tag{3.3}$$

$$\mathcal{L}_{\text{GAN}}(G, D, I^{syn}, I^{dist}) = \mathbb{E}_{I^{dist}}[\log(D(I^{dist}))] + \mathbb{E}_{I^{syn}}[\log(1 - D(G(I^{syn})))] \tag{3.4}$$

$$\mathcal{L}_{\text{MSE}}(G, I^{syn}, I^{dist}) = \mathbb{E}_{I^{dist}, I^{syn}}[||G(I^{syn}) - I^{dist}||_2] \tag{3.5}$$

Table 3.1.: Experimental Settings

	Data-I	
Network	SpCycleGAN	3D GAN counting
Training volume size	128 × 128 × 128	64 × 64 × 64
Number of pairs of training volumes	1	80
Inference volume size	128 × 128 × 128	128 × 128 × 64
Number of inference volumes (total)	15	20
Model	C^{Data-I}	G^{Data-I}
	Data-II	
Network	SpCycleGAN	3D GAN counting
Training volume size	128 × 128 × 32	64 × 64 × 64
Number of pairs of training volumes	4	80
Inference volume size	128 × 128 × 128	128 × 128 × (64 *or* 32)
Number of inference volumes (total)	15	20
Model	$C^{Data-II}$	$G^{Data-II}$

3.3 Experimental Results

The testing of our counting approach involves two different rat kidney data sets which are denoted as Data-I and Data-II. Data-I has size of $X \times Y = 512 \times 512$ pixels with $Z = 512$ (grayscale images), whereas Data-II consists of $Z = 32$ with the same size in X and Y as Data-I. The experimental settings are listed in Table 3.1. The size of the training and inference volumes were selected according to the original size of the volumes. The total training volume size of SpCycleGAN is 128 × 128 × 128 for both data sets. Then, 10 synthetic microscopy volumes of size 128 × 128 × 128 were generated and divided into 80 pairs of synthetic microscopy volumes of size 64 × 64 × 64. These volumes were used together with their corresponded distance map volumes to train a 3D GAN for nuclei counting for each data set. We selected $\lambda = 10$ in Equation 3.4 for both data sets. Our network architecture is implemented in TensorFlow [94] using Adam optimizer [95] with a learning rate of 0.005.

Figure 4.6 shows the synthetic binary images, their corresponding distance map images, the synthetic microscopy images, and the original microscopy images. The

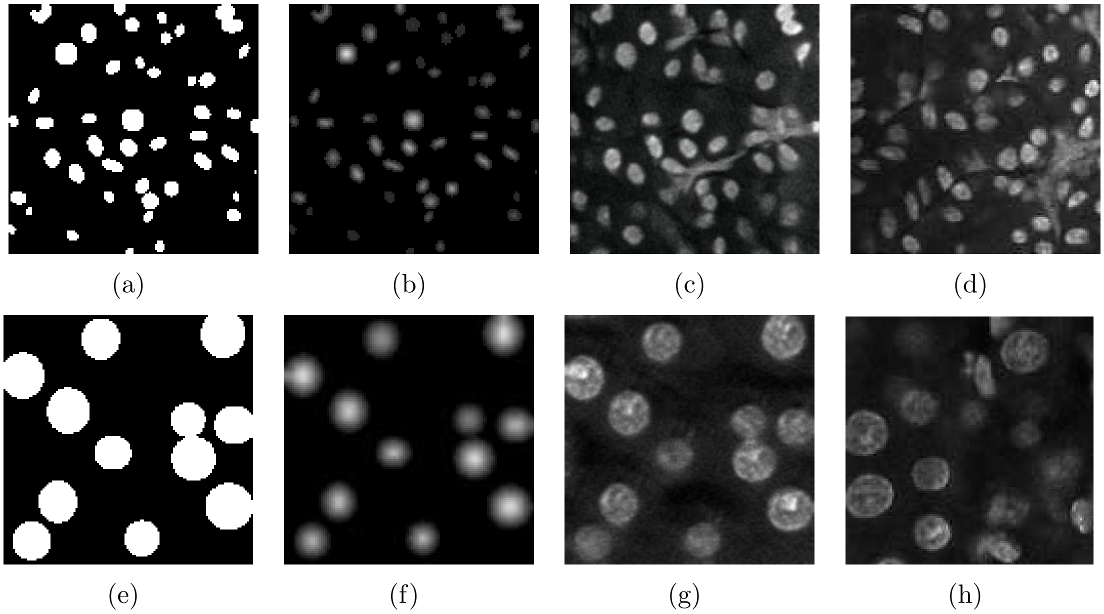


Fig. 3.3.: Comparison of slices of the synthetic binary volume, the distance map volume, the synthetic microscopy volume, and the original volume (a) Data-I synthetic binary image, (b) Data-I distance map image, (c) Data-I synthetic microscopy image, (d) Data-I original image, (e) Data-II synthetic binary image, (f) Data-II distance map image, (g) Data-II synthetic microscopy image, (h) Data-II original image

synthetic images are similar to the original microscopy images in terms of nuclei size and shape, nuclei distribution, intensity, and noise level. Using the trained model C, we can generate a paired set of training data for the 3D nuclei counting network. The trained model G is used to transform previously generated synthetic microscopy volumes into its corresponding synthetic distance map volumes, which contains nuclei centroid location information. Post-processing is done by first thresholding each voxel v with value v_i in I^{distg} as in Equation 3.6:

$$v_i = \begin{cases} 0 & \text{if } v_i \leq T \text{ or } 245 \leq v_i \\ v_i & \text{if } T \leq v_i \leq 245 \end{cases} \tag{3.6}$$

The threshold T is selected experimentally where $T = 45$ for Data-I and $T = 55$ for Data-II. Connected component analysis is then used to count and label each nucleus into a different color. For better visualization and evaluation on the object level, the labeled nuclei are dilated in 3D with a spherical mask to match the original nuclei sizes. In Figure 3.4, we show the post-processing steps of 3D nuclei counting with example slices from both synthetic and real microscopy image volumes.

The evaluation of microscopy image counting is difficult since groundtruth information is hard to obtain. We evaluate our results by comparing the results of nuclei counting N_i for the ith subvolume where $i \in \{1, \ldots, n\}$ with two other common ways of counting. N_i^{gt} is the groundtruth nuclei count for the ith subvolume where $i \in \{1, \ldots, n\}$. Here, n stands for the number of subvolumes. The mean absolute percentage error (MAPE) is then measured with Equation 3.7 and reported for each method.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{N_i - N_i^{gt}}{N_i^{gt}} \right| \tag{3.7}$$

The techniques compared with the proposed include the 3D watershed [81] and the ImageJ's 3D object counter (JACoP) [77]. Both counting approaches are semi-automatic since a threshold needs to be selected before counting. We initially selected the thresholding parameters based on the Otsu's method [96] and adjusted the parameters to best fit the volumes. Since the original microscopy volume suffers from a large variation of intensity and noise, we chose subvolumes from different locations of the original volume and processed each of them to obtain the nuclei count.

For evaluation, we generated $n = 10$ synthetic microscopy image subvolumes for both data sets denoted as $I^{syn_{Data-I}}$ and $I^{syn_{Data-II}}$, respectively. $I^{syn_{Data-I}}$ has a size of $128 \times 128 \times 64$ with the average number of nuclei being 256. $I^{syn_{Data-II}}$ has a size of $128 \times 128 \times 64$ with the average number of nuclei being 64. We processed these volumes with the ImageJ's 3D object counter (JACoP), the 3D watershed, and the proposed 3D nuclei counter to compare the counting results. The quantitative evaluations for the subvolumes are shown in Table 3.2. In Table 3.2, we can see that the 3D watershed and the JACoP method produce less accurate counting result comparing to our proposed method. Based on our observation for the JACoP, nuclei are often

Table 3.2.: Comparison of mean absolute percentage error (MAPE) of the proposed 3D nuclei counter with 3D watershed [81] and JACoP [77]

n=10	Synthetic Data-I	Synthetic Data-II
3D Watershed	14.69%	13.78%
JACoP	11.86%	16.10%
Proposed 3D Nuclei Counter	5.64%	2.13%
	Microscopy Data-I	Microscopy Data-II
3D Watershed	19.76%	10.88%
JACoP	12.67%	18.72%
Proposed 3D Nuclei Counter	6.68%	6.53%

missing around darker boundaries of the volume. Note that the JACoP has much lower detection accuracy for microscopy volume of large size due to inhomogeneity of the microscopy volume, so small subvolumes of microscopy image volume are used for counting. Our proposed nuclei counter achieves lower mean absolute percentage error in counting than the other two techniques.

We also selected $n = 10$ subvolumes from original microscopy data for the evaluation purpose. More specifically, we selected 10 subvolumes from I^{Data-I} with size of 128 × 128 × 64 with the average number of nuclei being 298 and 10 subvolumes of $I^{Data-II}$ with size of 128 × 128 × 32 with the average number of nuclei being 25, respectively. Note that the evaluation size of $I^{Data-II}$ is selected according to its total size 512 × 512 × 32. From Table 3.2, the proposed 3D nuclei counter achieves the lowest error among all three different counting approaches. Figure 3.5 shows the comparison of the color-coded images obtained from the 3D watershed, the JACoP, and our proposed method. It is observed that the 3D watershed and the JACoP method cannot distinguish between nuclei and non-nuclei structures. Our proposed method achieves better performance in finding nuclei and separating touching nuclei.

The color-coded images from the proposed method have less noise from the inhomogeneity of the nucleus compared with the results from the other two techniques.

Table 3.3.: Object-based comparison of the proposed 3D nuclei counter with 3D watershed [81] and JACoP [77] for Data-I

Microscopy Data-I						
	N_{TP}	N_{FP}	N_{FN}	P	R	$F1$
3D Watershed	121	9	153	93.08%	44.16%	59.90%
JACoP	146	19	128	88.48%	53.28%	66.51%
Proposed 3D Nuclei Counter	212	18	56	92.17%	79.10%	85.14%

Table 3.4.: Object-based comparison of the proposed 3D nuclei counter with 3D watershed [81] and JACoP [77] for Data-II

Microscopy Data-II						
	N_{TP}	N_{FP}	N_{FN}	P	R	$F1$
3D Watershed	171	13	50	92.93%	77.38%	84.44%
JACoP	151	16	62	90.42%	70.89%	79.47%
Proposed 3D Nuclei Counter	193	15	33	92.79%	85.40%	88.94%

To analyze the detection accuracy of the locations of the nuclei found with the counting methods, we use the object-based evaluation as described in [97]. The evaluation is based on manually annotated groundtruth of two subvolumes from I^{Data-I} with size of 128 × 128 × 64 and from $I^{Data-II}$ with size of 512 × 512 × 16. The groundtruth is obtained using ITK-SNAP [98] where each nucleus is manually labeled individually. If a nucleus overlaps equal or more than 50% with its corresponding groundtruth nucleus, it is counted as a true-positive, N_{TP}. If a nucleus overlaps less than 50% with its corresponding groundtruth nucleus or there is no corresponding groundtruth, it is counted as a false-positive, N_{FP}. A false negative, N_{FN}, is defined as when a nucleus is present in the groundtruth but no corresponding nucleus is found by the counting method. Then, the F1 score (F1), Precision (P) and Recall (R) described in [99] can be calculated as:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, R = \frac{N_{TP}}{N_{TP} + N_{FN}}, F1 = \frac{2PR}{P + R} \tag{3.8}$$

For fair comparisons, small objects removal is done to remove objects with sizes less that 5% of the average nuclei sizes in voxels. In Table 4.1 and Table 3.4, our proposed 3D nuclei counter achieves higher F1 score than the compared methods, which shows that the proposed method can successfully find nuclei at the correct locations. Both the 3D watershed and the JACoP method produce large numbers of N_{FN}, causing low R and F1 score. Overall, our proposed method outperforms the other two methods in both the MAPE evaluation and the object-based evaluation.

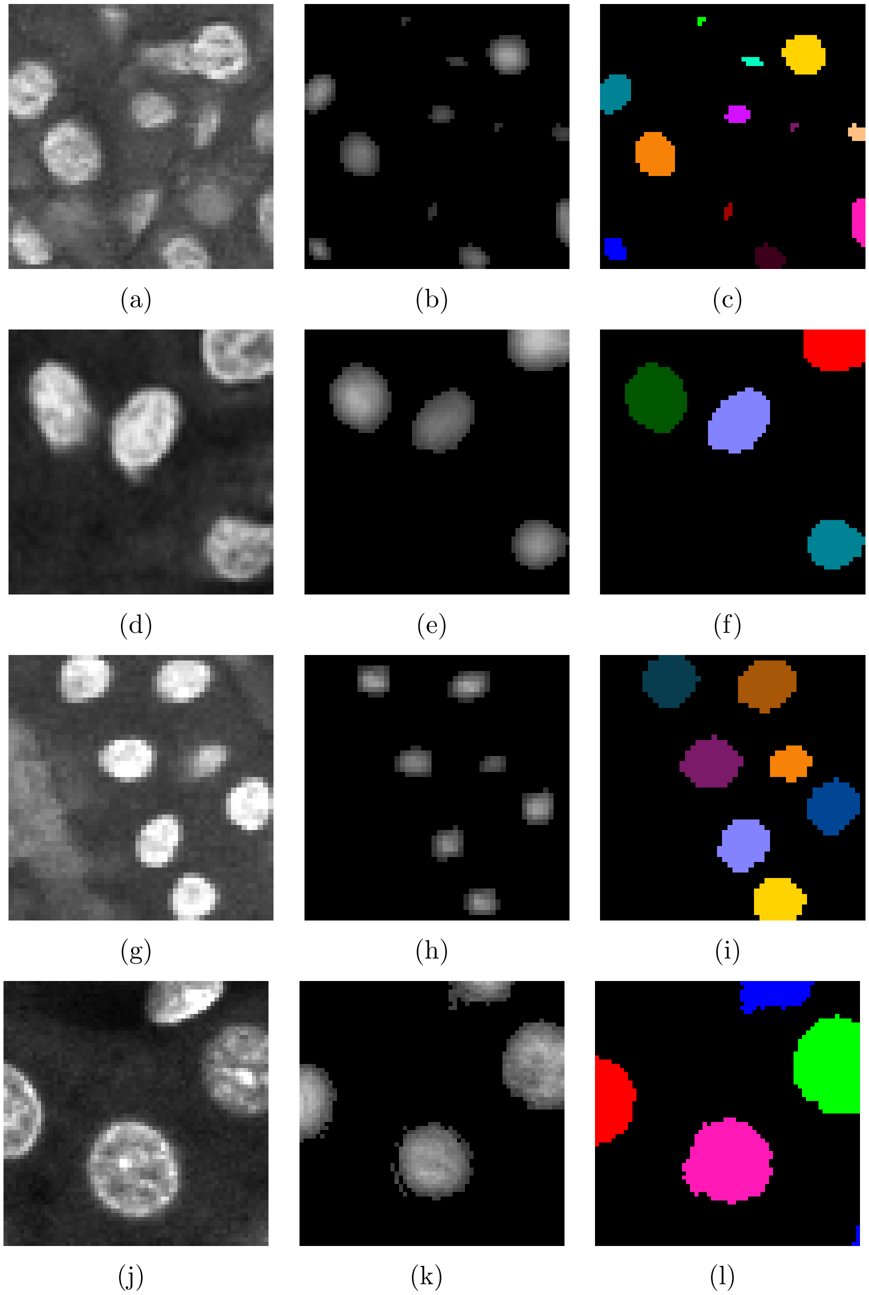
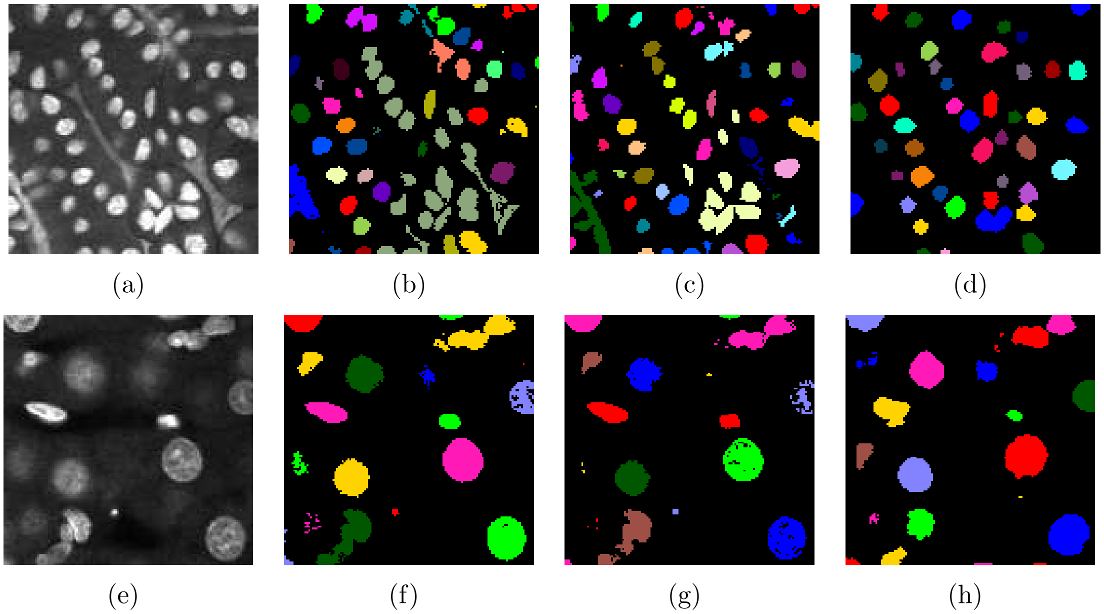


Fig. 3.4.: Example of slices of the testing volume, the synthetic distance map volume with thresholding, and the color-coded volume from 3D nuclei counting (a) Data-I synthetic image, (b) corresponding I^{distg} with thresholding for (a), (c) color-coded image for (a), (d) Data-II synthetic image, (e) corresponding I^{distg} with thresholding for (d), (f) color-coded image for (d), (g) Data-I real image, (h) corresponding I^{distg} with thresholding for (g), (i) color-coded image for (g), (j) Data-II real image, (k) corresponding I^{distg} with thresholding for (j), (l) color-coded image for (j)


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3.5.: Comparison of slices of the original image volume and results obtained from 3D watershed, JACoP, and our proposed method (a) Data-I original image, (b) Data-I color-coded image from 3D watershed, (c) Data-I color-coded image from JACoP, (d) Data-I color-coded image from our method, (e) Data-II original image, (f) Data-II color-coded image from 3D watershed, (g) Data-II color-coded image from JACoP, (h) Data-II color-coded image from our method

4. NUCLEI SEGMENTATION AND DETECTION WITH BOUNDARY REFINEMENT

4.1 Background

Fluorescence microscopy is an optical microscopy for the visualization of subcellular structures in living species using fluorescence. Two-photon microscopy is a type of precision fluorescence microscopy that allows biologists to visualize deeper into tissue. With the increased penetration depth of two-photon microscopy, large size data are often produced. An automatic image analysis tool that can extract important biological information in three dimension is required. A common application is the detection and segmentation of biological structures such as nuclei within microscopy images. Microscopy images that contains nuclei are hard to analyze due to several reasons: nuclei sizes and shapes are nonregular, nuclei are often crowded and "touching" to nearby nuclei, and the limited image quality with the presence of noise.

Segmentation and classification of cell nuclei in fluorescence 3D microscopy image volumes are fundamental steps for image analysis. However, accurate cell nuclei segmentation and detection in microscopy image volumes are hampered by poor image quality, crowding of nuclei, and large variation in nuclei size and shape. In this work, we present an unsupervised volume to volume translation approach adapted from the Recycle-GAN using modified Hausdorff distance loss for synthetically generating nuclei with better shapes. A 3D CNN with a regularization term is used for nuclei segmentation and classification followed by nuclei boundary refinement. Experimental results demonstrate that the proposed method can successfully segment nuclei and identify individual nuclei.

With the development of fluorescence microscopy, such as two-photon microscopy and multi-photon microscopy, large 3D volumes of microscopy data with deep penetration depth into tissue are possible [15]. For quantitative analysis of biomedical structures in these 3D volumes, cell nuclei have been extensively used since they form a highly organized structure and can easily be labeled with fluorescent markers [100]. By analyzing volumetric data with nuclei information, important information such as cell location, cell density, and cell type can be obtained.

There have been several methods developed for nuclei segmentation and classification. One example is the use of watershed [23] which is a combination of region growing and edge detection to achieve individual nuclei segmentation and classification. Another popular method for nuclei segmentation is active contours [101] which minimizes an energy functional iteratively to fit a contour to objects of interest. An early version of the edge-based active contours [101] tends to fail to segment objects in fluorescence microscopy because of its sensitiveness to initial contour placement and vulnerability to noise. Region-based active contours was introduced in [25] that seeks an energy balance between foreground and background regions. Region-based active contours was extended to 3D by developing an inhomogeneity intensity correction in [22]. Meanwhile, CellProfiler software [102] was developed for biomedical image analysis including cell segmentation and classification.

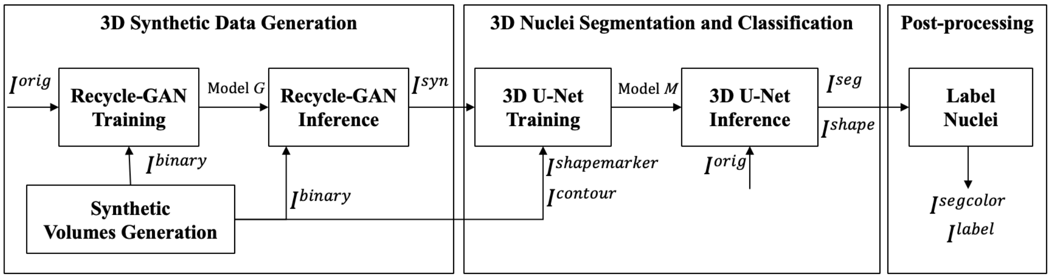


Fig. 4.1.: Block diagram of the proposed nuclei segmentation and classification method

Segmenting and detecting cell nuclei in microscopy image volumes are challenging due to poor image quality, crowding of nuclei, and large variation in nuclei sizes and shapes. Recent advancement in deep learning has shown success in finding and distinguish nuclei under different image contexts. To reduce the intensive work of groundtruth nuclei in 3D, carefully crafted synthetic training image pairs are used to train Convolutional Neural Networks (CNN). Amongst methods for image generation, unsupervised image to image translation with Generative Adversarial Networks (GAN) has become a prominent choice. Morphological operations are used as supplemental processing in those methods for result refinement.

Convolutional neural networks (CNN) has been popular to address various problems in medical imaging [103]. U-Net [48] is one of the most successful 2D CNN architectures used for medical image segmentation. U-Net uses an encoder-decoder architecture that transfers entire feature maps from each encoder layer to the corresponding decoder layer. Nuclei classification, segmentation, and detection using a combination of densely connected recurrent convolutional network and regression model with recurrent residual convolutional neural network based on U-Net was described in [104]. A 3D nuclei segmentation method that utilizes the SegNet [46] with data augmentation followed by a 3D watershed was presented in [86]. Additionally, a 3D U-Net which extends U-Net [48] to 3D to achieve volumetric image segmentation was introduced in [36]. Similarly, a V-Net which uses the Dice coefficient as a loss term in training to achieve volumetric segmentation was demonstrated in [105]. More recently, the Hausdorff distance which is typically used for the shape similarity measure for evaluating segmentation performance [97] was introduced as a loss term for locating objects [106] as well as medical image segmentation [107].

The previously discussed methods require huge volumes of manually annotated groundtruth for training. To address this problem, an approach to generate synthetic groundtruth and corresponding synthetic data was needed. The use of the cycle-consistent adversarial networks (CycleGAN) [58] for generating realistic 3D images from unpaired training data has shown promising results in [89] and [108]. Hence, we introduced a two-stage method known as spatially constrained cycle-consistent adversarial networks (SpCycleGAN) [59] that incorporates a spatially constrained loss to the CycleGAN to prevent misalignment between synthetically generated groundtruth volume and corresponding synthetically generated microscopy volume followed by 3D segmentation. This SpCycleGAN was utilized for nuclei detection and counting in [109] and further extended in [110] to utilize three directions in a volume along with axial, coronal, and sagittal sections (3-Way SpCycleGAN) to incorporate 3D information. One drawback of the SpCycleGAN and the 3-Way SpCycleGAN is that they are not fully 3D but 2D based methods. They generate synthetic groundtruth volumes and corresponding microscopy volumes that are not perfectly aligned in 3D. To address this problem, the Recycle-GAN [24] was introduced which adds a third dimension to the CycleGAN [58] to achieve video-to-video style transfer.

In this work, we describe a 3D segmentation and classification method to segment and identify individual nuclei in fluorescence microscopy volumes without having groundtruth volumes. Three dimensional synthetic data is generated using the Recycle-GAN [24] with the Hausdorff distance loss introduced in [107] to preserve the shape of individual nuclei. Realistic microscopy image volumes with nuclei segmentation mask and nucleus boundary groundtruth volumes are generated. A subsequent 3D CNN with a regularization term that discourage detection out of nucleus boundary is used to detect and segment nuclei. Nuclei boundary refinement is then performed to enhance nuclei segmentation. Experimental results on our rat kidney dataset show the proposed method is competitive with respect to several state-of-the-art methods. A 3D CNN network is then trained using the 3D synthetic data to segment and classify different kinds of nuclei structures. Our method is evaluated using hand segmented groundtruth volumes of real fluorescence microscopy data from a rat kidney. Our data was collected using two-photon microscopy with nuclei labeled with Hoechst 33342 staining.

4.2 Proposed Method

Figure 4.1 shows the block diagram of our proposed method for nuclei segmentation and classification. In this work, a 3D image volume of size $X \times Y \times Z$ is denoted with I. We use I_{z_p} to represent an xy slice of I at the p^{th} focal plane along the z-direction in a volume. Here, the range of p is $p \in \{1, \ldots, Z\}$. I_{y_q} is a xz slice of I at the q^{th} focal plane along the y-direction with $q \in \{1, \ldots, Y\}$. Similarly, I_{x_r} is a yz slice of I at the r^{th} focal plane along the x-direction with $r \in \{1, \ldots, X\}$. To indicate how a volume is sectioned from I, $I_{\left(r_i:r_f,q_i:q_f,p_i:p_f\right)}$ is used, where the coordinates of x is between $\{r_i, ..., r_f\}$, the coordinates of y is between $\{q_i, ..., q_f\}$, and the coordinate of z is between $\{p_i, ..., p_f\}$. For example, if a subvolume is a xy section from the 100^{th} slice to the 150^{th} slice along the z-direction, the subvolume can be denoted as $I_{\left(1:r_f,1:q_f,100:150\right)}$.

Our proposed method is composed of three main steps: 3D synthetic data generation, 3D nuclei segmentation and classification, and post-processing. I^{orig} denotes the original microscopy volume used for training during 3D synthetic data generation and testing during 3D nuclei segmentation and classification. I^{binary} represents the synthetic binary data generated during 3D synthetic data generation, which is used together with I^{orig} to train a Recycle-GAN model G. With a trained model G, I^{syn} can be generated as a synthetic microscopy volume based on the features of I^{orig}. During the generation of I^{binary}, two more synthetic volumes $I^{contour}$ and $I^{shapemarker}$ that are corresponding to I^{binary} are generated. Here, $I^{contour}$ contains the boundary of each nucleus whereas $I^{shapemarker}$ contains the ellipticity as a shape indicator of the nuclei. I^{syn} is paired with I^{binary} to train a 3D U-Net segmentation model M. Also, the corresponding $I^{shapemarker}$ and $I^{contour}$ are used during the training of the model to refine the shape of the segmentation. During the inference of the 3D U-Net, two different initial results volumes I^{seg} and I^{shape} are generated using model M. Here, I^{seg} contains the binary segmentation mask of the nuclei and I^{shape} contains the shape indicator of the nuclei volume. During the post-processing step, I^{seg} and I^{shape} are used to produce the final results of the color-coded segmentation volume $I^{segcolor}$ and the labeled volumed I^{label} which labels each nucleus by its type.

4.2.1 3D Synthetic Data Generation

Three dimensional synthetic data generation consists of three stages: the generation of the synthetic volumes, the training of the Recycle-GAN [24], and the inference of the Recycle-GAN. The synthetic volume I^{binary} is created by adding ellipsoidal nuclei candidates with different shape, size, and orientation to the volume at random locations. The range of the shape and size of the synthetic nuclei are appropriately chosen for the original microscopy volume. For i^{th} nucleus with semi-axes $\mathbf{a}_i = (a_{i_x}, a_{i_y}, a_{i_z})$ and centroid at $\mathbf{c}_i = (x_i, y_i, z_i)$, $I^{binary,i}$ can be expressed with Eq. (4.1) as:

$$I^{binary,i} = \begin{cases} 1, & \text{if} \left(\frac{x-x_i}{a_{i_x}}\right)^2 + \left(\frac{y-y_i}{a_{i_y}}\right)^2 + \left(\frac{z-z_i}{a_{i_z}}\right)^2 < 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where the value of $\mathbf{a}$ is selected from $r \in \{r_{min}, \ldots, r_{max}\}$. Let $a_{i_{min}}$ and $a_{i_{max}}$ represent the minimum and maximum value among $(a_{i_x}, a_{i_y}, a_{i_z})$, respectively. Here, the orientation of nuclei is selected with random rotation as described in [88]. For each I^{binary}, two corresponding synthetic volumes $I^{contour}$ and $I^{shapemarker}$ are generated using Eq. (4.2) and (4.3) such that

$$I^{contour,i} = \begin{cases} 1, & \text{if} 0.7 < \left(\frac{x-x_i}{a_{i_x}}\right)^2 + \left(\frac{y-y_i}{a_{i_y}}\right)^2 + \left(\frac{z-z_i}{a_{i_z}}\right)^2 < 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

$$I^{shapemarker,i} = 255 \cdot \left(\frac{a_{i_{max}} - a_{i_{min}}}{r_{max} - r_{min}}\right) \cdot I^{binary,i} \tag{4.3}$$

The generated I^{binary} is then processed with a 3D Gaussian filter with $\sigma = 2$ to simulate microscopy's image acquisition process where the target object is convolved with a 3D point spread function (PSF) [110]. The blurred I^{binary} volume is then used together with I^{orig} to train the Recycle-GAN.

The Recycle-GAN is an extension of the CycleGAN by replacing the cycle consistency loss with a recurrent loss and a recycle loss. The main difference between the CycleGAN and the Recycle-GAN is that the Recycle-GAN was originally proposed for use in the generation of synthetic videos using ordered sequential frames of images. Here, we used a 3D image volume instead of a video, which means the ordered sequential images are a series of images taken along the z direction. In addition to the generative models G_1 and G_2 and the discriminative models D_1 and D_2, the Recycle-GAN uses the predictive models P_1 and P_2 to predict the current frame based on S previous frames. For our method, we modified the predictive model to take an ordered series of S images (I_z) along the z direction in the spatial domain instead of the temporal domain. The recurrent loss is used to measure the prediction accuracy of the predictive model. Similarly, the recycle loss is a modified version of the cycle consistency loss which incorporates the predictive model for image reconstruction.

Eq. (4.4) shows the training losses of our proposed method. Note that λ_1, ..., λ_5 are the controllable coefficients for each loss term. G_1 is a generative model that can transfer I^{binary} to I^{syn} whereas G_2 is a generative model that can transfer I^{syn} to I^{binary}. P_1 is a generative model that can predict ${I_{z_{p+S+1}}}^{binary}$ from ${I_{z_p:z_{p+S}}}^{binary}$ whereas P_2 is a generative model that can predict ${I_{z_{p+S+1}}}^{syn}$ from ${I_{z_p:z_{p+S}}}^{syn}$. Additionally, D_1 is a discriminative model that distinguishes between I^{binary} and $P_1(G_2(I^{syn}))$. Also, D_2 is a discriminative model that distinguishes between I^{syn} and $P_2(G_1(I^{binary}))$.

$$\begin{aligned}
&\mathcal{L}(G_1, G_2, D_1, D_2, P_1, P_2, S)\{I^{binary}, I^{syn}\}\\
=&\mathcal{L}_{\text{GAN}}(G_1, D_2) + \mathcal{L}_{\text{GAN}}(G_2, D_1)\\
+&\lambda_1 \cdot \mathcal{L}_{recycle}(G_1, G_2, P_2, S) + \lambda_2 \cdot \mathcal{L}_{recycle}(G_2, G_1, P_1, S)\\
+&\lambda_3 \cdot \mathcal{L}_{recurrent}(P_1, S) + \lambda_4 \cdot \mathcal{L}_{recurrent}(P_2, S)\\
+&\lambda_5 \cdot \mathcal{L}_{contour}(G_2, P_1, S)
\end{aligned} \tag{4.4}$$

Moreover, we utilize a contour constraint loss term $\mathcal{L}_{contour}$ based on estimating the Hausdorff distance from the distance transform as described in [107]. Since the Hausdorff distance estimates the difference between two boundaries, adding this loss could refine the shape of the synthetic generated nuclei. We define the distance transform as a function $F_{DT} : I^{binary} \rightarrow F_{DT}(I^{binary})$ which finds the Euclidean distance between each voxel and its nearest background voxel. $\mathcal{L}_{contour}$ works by penalizing large errors at the boundaries of nuclei so that the generated nuclei have refined shape by minimizing $\mathcal{L}_{contour}$ as in Eq. (4.5):

$$\begin{aligned}
&\mathcal{L}_{contour}(G_2, P_1, S)\{I^{binary}, I^{syn}\} =\\
&\mathbb{E}_{I^{binary}}\{(||I^{binary} - P_1(G_2(I^{syn}))||_2 \cdot F_{DT}(I^{binary})^2\\
&+||I^{binary} - P_1(G_2(I^{syn}))||_2 \cdot F_{DT}(P_1(G_2(I^{syn})))^2)^{1/2}\}
\end{aligned} \tag{4.5}$$

During the Recycle-GAN inference, the synthetic microscopy volume, I^{syn}, is generated from using the model G on the synthetic binary volume, I^{binary}. For each synthetic volume I^{binary}, the corresponding synthetic volumes $I^{contour}$ and $I^{labelmarker}$ are generated. I^{binary} and I^{syn} form a paired set of volumes for the training of the 3D U-Net segmentation model.

4.2.2 3D Nuclei Segmentation and Classification

Three dimensional nuclei segmentation and classification consists of 3D U-Net training, 3D U-Net inference, and post-processing. As shown in Figure 4.1, we utilize a paired set I^{syn} and I^{binary} to train a 3D U-Net and obtain a generative model M_1. Here, the model M_1 is a function mapping nuclei in I^{orig} to the segmented volume in I^{seg}. Another paired set I^{syn} and $I^{shapemarker}$ are used to train a 3D U-Net model M_2. The model M_2 is a function that maps nuclei in I^{orig} to its corresponding shape information in I^{shape}. $I^{contour}$ is utilized during the training of the model M_1 and M_2 for refining the shape of nuclei in the segmented volume. I^{seg} and I^{shape} are then used to generate nuclei segmentation volume $I^{segcolor}$ and I^{label}.

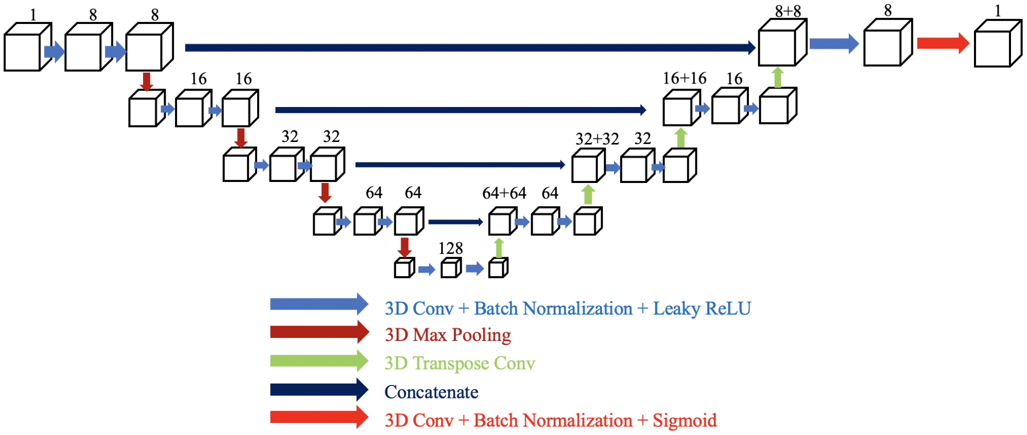


Fig. 4.2.: Architecture of the modified 3D U-Net

The architecture of the modified 3D U-Net [36] for nuclei segmentation and classification is shown in Figure 4.2. A filter size of 3×3×3 is used for 3D convolution. Each 3D convolution is followed by a batch normalization [47] and a rectified-linear unit (ReLU) activation function. Each convolutional layer contains a 3D convolution with filter size of $3 \times 3 \times 3$, a batch normalization, and a rectified-linear unit (ReLU) activation function. A 3D max-pooling of stride 2 with a window size of $2 \times 2 \times 2$ is used for the downsampling procedure. For upsampling, 3D transposed convolutional is used.

Between each encoding and decoding layer, a concatenation operation is used to preserve spatial information. Similarly,canny a 3D transposed convolution is used for an upsampling procedure. Corresponding encoder and decoder layers are connected with a concatenation operation to preserve spatial information. The training loss here is the same as described in [59] but with an addition of $\mathcal{L}_{MSE}\{I^{contour}, C(I^{seg})\}$, where $C(\cdot)$ is a Canny [111] edge function that finds the edges of a binary volume.

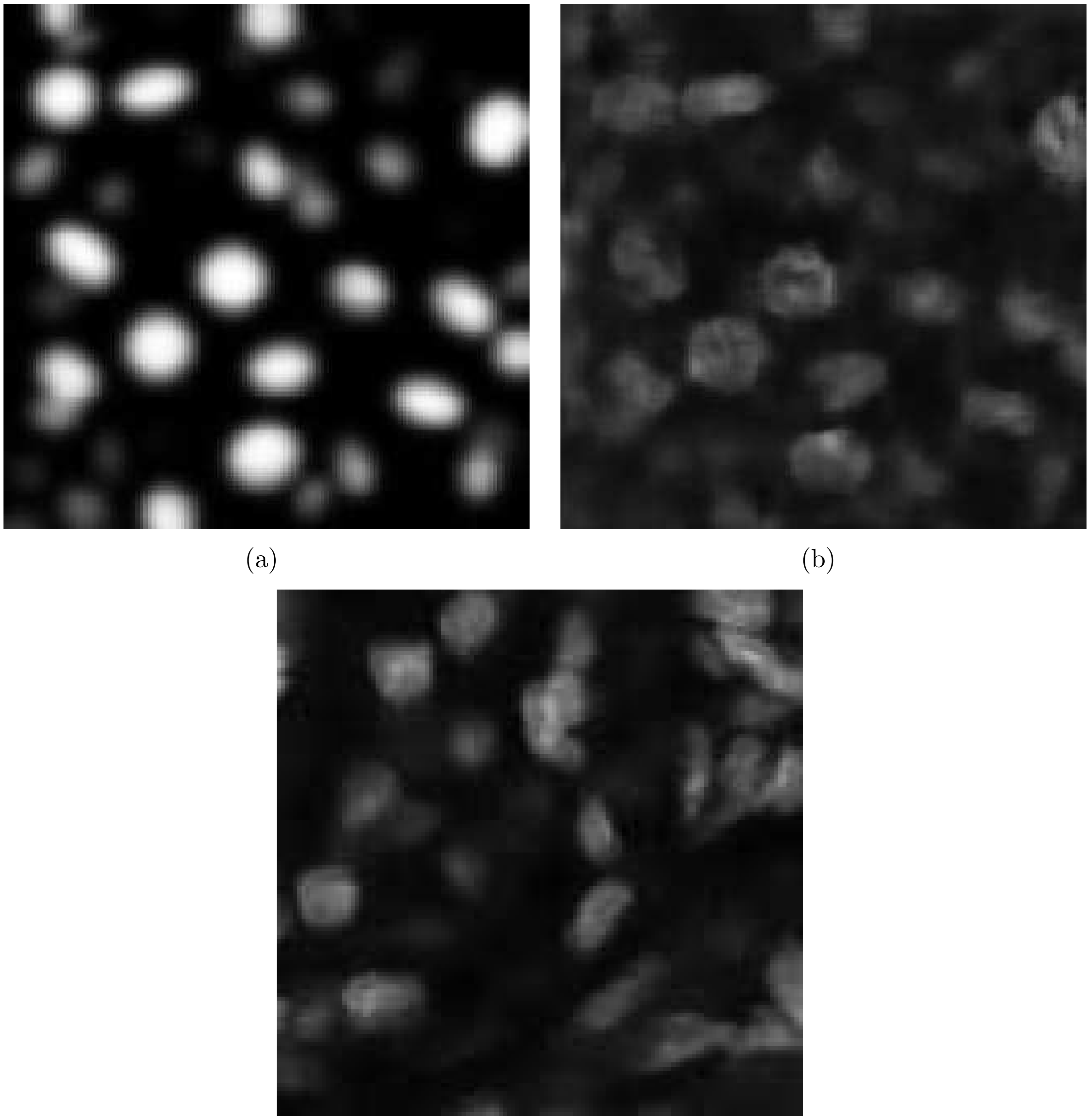
4.3 Experimental Results

We tested our proposed method on two different rat kidney data sets. These data sets contain grayscale images of size $X = 512 \times Y = 512$. Data-III consists of $Z = 415$ images.

The ReCycleGAN is implemented in Pytorch with default parameters given by [24] with $T_t = 10$. A $128 \times 128 \times 128$ synthetic binary volume for Data-III denoted as $I^{bi_{Data-III}}$ and a $128 \times 128 \times 128$ subvolume of original microscopy volume of Data-III denoted as $I^{orig_{Data-III}}$ were used to train model $C^{Data-III}$. .

Since our modified 3D segmentation architecture takes volumes of size of $64 \times 64 \times 64$, we divided $I^{bi_{Data-III}}$, $I^{bd_{Data-III}}$, and $I^{syn_{Data-III}}$ into adjacent non overlapping $64 \times 64 \times 64$. Thus, we have 80 pairs of synthetic binary volumes and corresponded synthetic microscopy volumes per each data to train our 3D segmentation network. Note that these 80 synthetic binary volumes per each data are used as groundtruth volumes to be paired with corresponding synthetic microscopy volumes. Model $M^{Data-III}$ are then generated.

Our 3D segmentation network is implemented in Pytorch using the Adam optimizer [95] with learning rate 0.001. For the evaluation purpose, we use different settings of using 3D synthetic data generation methods. Also, we use different loss functions with different settings of the μ_1 and μ_2. Moreover, we also compared our modified 3D U-Net with 3D encoder-decoder architecture [88]. Lastly, small objects which are less than 20 voxels were removed using 3D connected components.



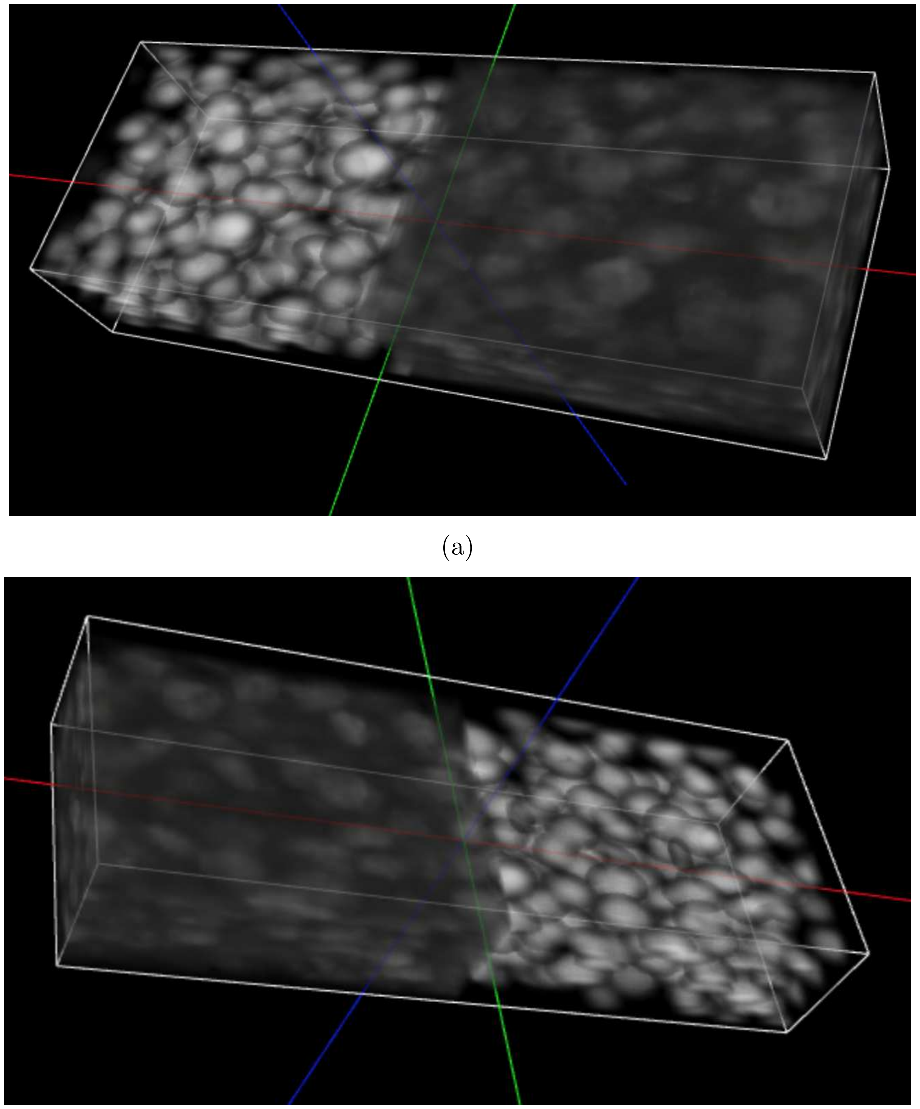
(a)

(b)

(c)

Fig. 4.3.: Comparison of the synthetic binary volume with Gaussian blur, the synthetic microscopy volume, and the original volume (a) Data-III synthetic binary image gaussian blur, (b) Data-III synthetic microscopy image, (c) Data-III original image

Figure 4.3 shows the synthetic images generated by our proposed method. The left side indicates synthetic binary images with blur whereas the middle shows synthetic image artificially generated from corresponding synthetic binary image, the right side shows the original image. As can be seen from Figure 4.3, the synthetic images reflect characteristics of the original microscopy images such as background noise, nuclei



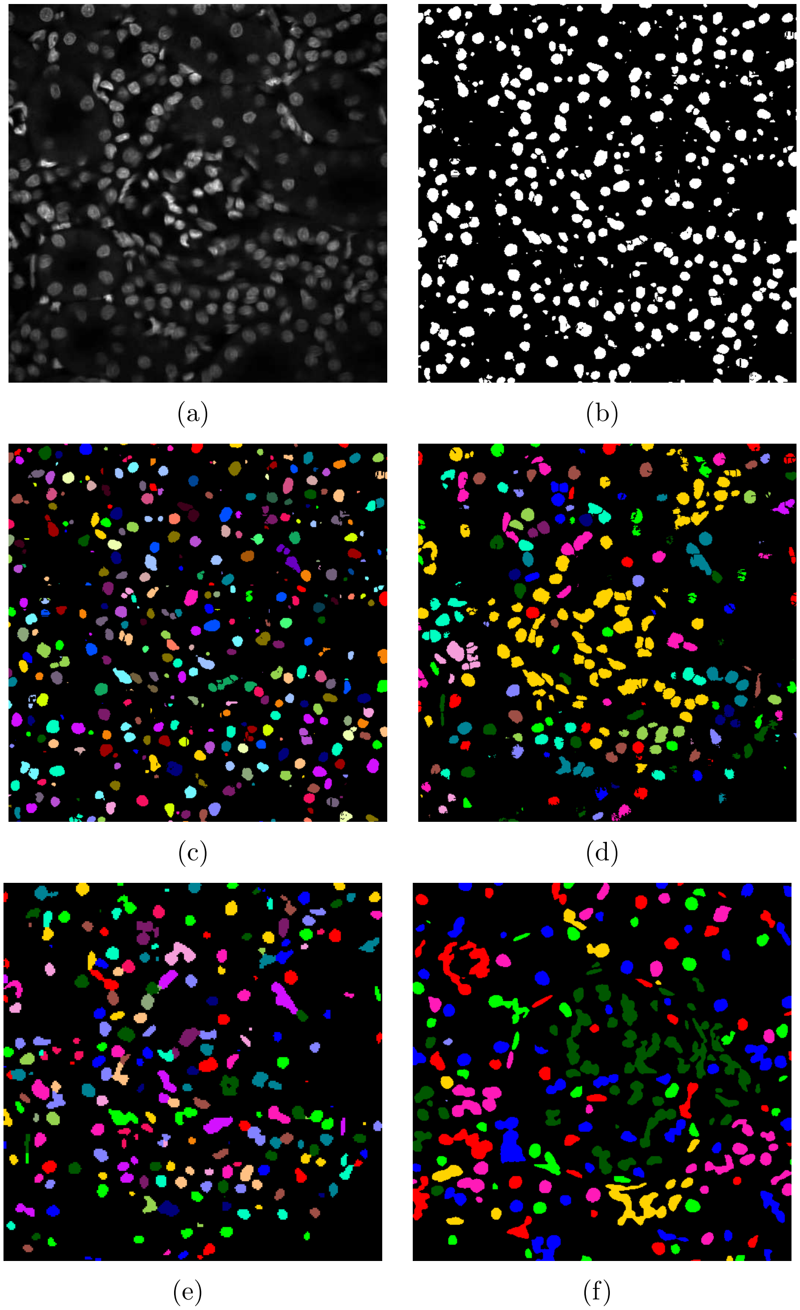
(a)

(b)

Fig. 4.4.: Comparison of the in 3D view of the synthetic binary volume with Gaussian blur, and the synthetic microscopy volume (a) left side angle view, (b) right side angle view

shape, orientation and intensity. Figure 4.4 shows two angles of the 3D visualization of the 128 × 128 × 128 volumes of the synthetic binary volume with blur and the synthetic microscopy volume used to train the ReCycleGAN model C.

Figure 4.5 shows the segmentation of result generated from the 3D segmentation network M. For the first row, the left images shows the original image, the middle



(a)

(b)

(c)

(d)

(e)

(f)

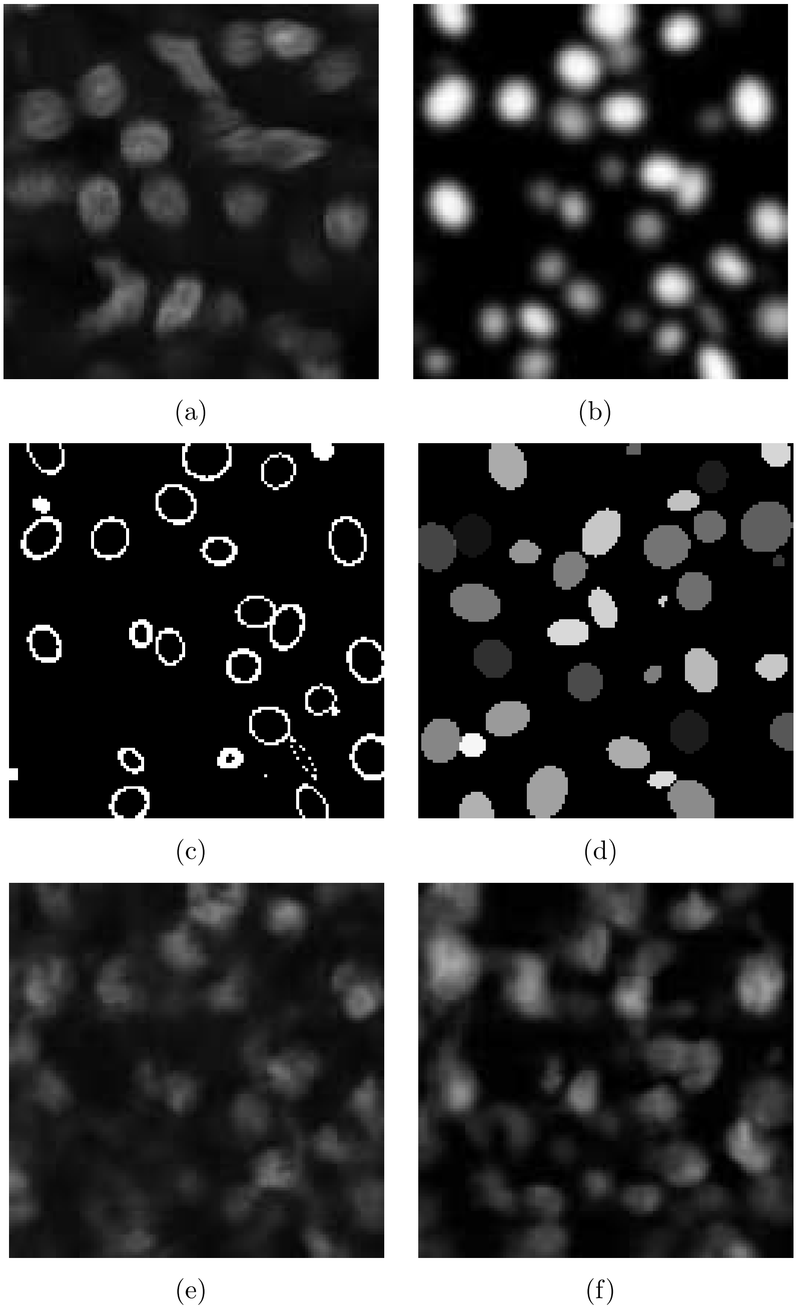
Fig. 4.5.: Comparison of slices of the original volume, the segmentation volume after 3-way watershed, and the color coded volume of the segmentation result (a) Data-III original volume, (b) Data-III segmentation volume after 3-way watershed, (c) Data-III color coded volume of the proposed method, (d) Data-III Watershed result color coded, (e) Data-III CellProfiler result color coded, (f) Data-III Squassh result color coded

image show the segmentation image of the proposed method after 3-way watershed, and the right side image shows the color-coded result of the segmentation result. For the second row, the left images shows the result from Watershed, the middle image shows the result from CellProfiler, and the right side image shows the color-coded result of Squassh. As can be observed, the segmentation result can detect nuclei regardless of the brightness and blurring from the original image. Nuclei that are of non-round shapes can be detected. Nuclei that are close to each other are separated from each other in the color-coded image. In Squassh and Watershed method, the nuclei at the center of the images are grouped into one cluster, greatly lowered the detection accuracy. In the CellProfiler result, miss detection of nuclei around the corner regions are observed, also, adjacent nuclei are more often detected as connected. By visual observation, the proposed method can segment and detect nuclei more accurately comparing to the other methods.

To test our proposed method, we used two rat kidney data sets denoted as Data-I and Data-II[1]. The size of Data-I and Data-II are $X \times Y \times Z = 512 \times 512 \times 415$ voxels and $X \times Y \times Z = 512 \times 512 \times 32$ voxels. A volume size of $128 \times 128 \times 128$ was selected for the Recycle-GAN training for both data sets. By inferencing with the trained model G, 10 synthetic microscopy volumes were generated, with the size being $128 \times 128 \times 128$. Those volumes were divided into 80 pairs of smaller volumes ($64 \times 64 \times 64$). These 80 pairs of synthetic volumes are then used for the training of model M_1 and M_2. Also, $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = \lambda_4 = 10$, $\lambda_5 = 0.1$, $S = 10$ in Eq. (4.4) were experimentally selected and used for both data sets. Note that we used PyTorch to implement our network architecture. We used Adam optimizer [95] with a learning rate of 0.0001.

Figure 4.6 shows the synthetic binary images, the corresponding synthetic contour images, and the corresponding synthetic microscopy images. An original microscopy image is shown in Figure 4.6(a). An example slice of the synthetic $I^{shapemarker}$ is

[1] All imaging data used for this work was obtained from animals and was collected at the Indiana Center for Biological Microscopy at the Indiana University Medical School. The studies used to collect this data were approved by the Indiana University animal use committee.


(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4.6.: Comparison of slices of the synthetic volumes with the original volume of Data-I (a) I^{orig} (b) I^{binary} with Gaussian blur, (c) $I^{contour}$, (d) $I^{shapemarker}$ at another plane, (e) I^{syn} from Recycle-GAN, (f) I^{syn} from Recycle-GAN with the Hausdorff distance loss

displayed in Figure 4.6(d). It can be observed that the synthetic microscopy images look like the original microscopy image with respect to nuclei characteristics. Then,

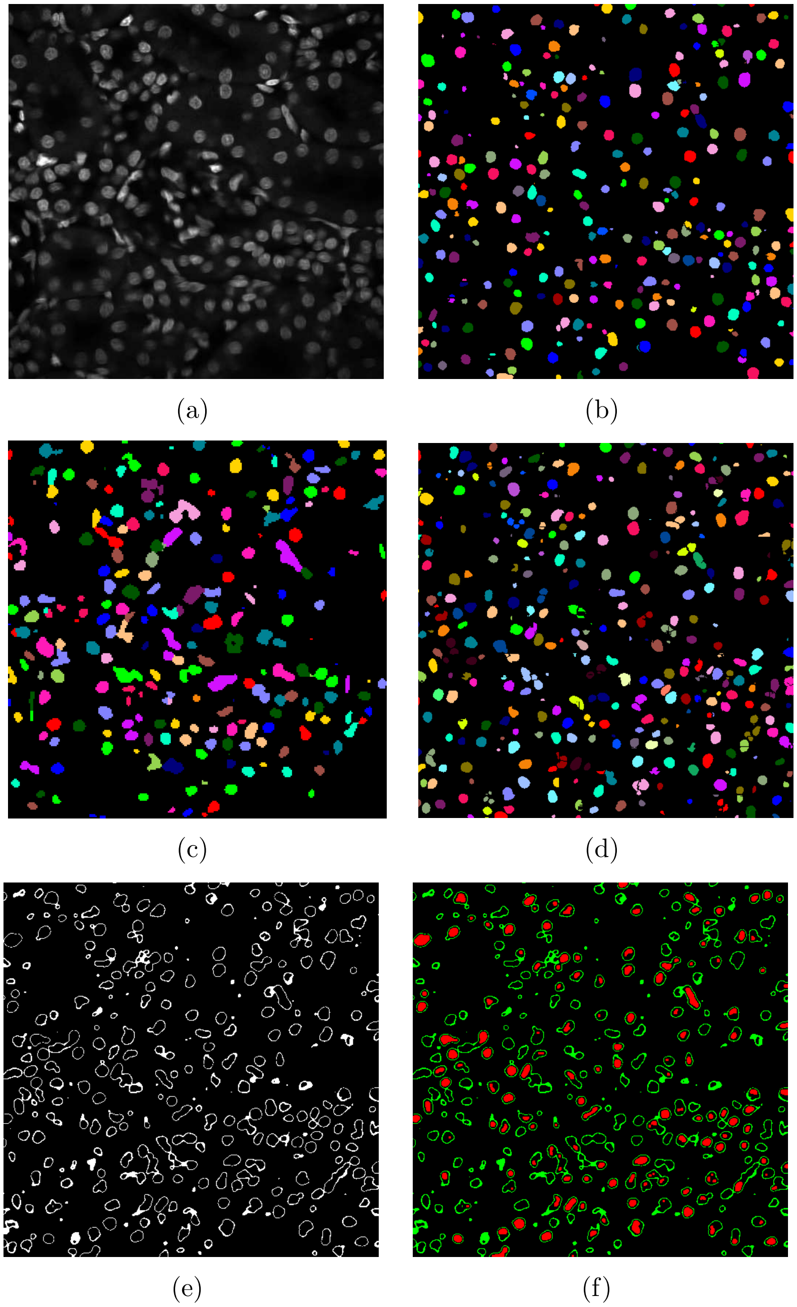


Fig. 4.7.: Comparison of slices of the original image volume and results obtained from SpCycleGAN [59], CellProfiler [102], and our proposed method of Data-I: (a) I^{orig}, (b) segmentation result from SpCycleGAN, (c) segmentation result from CellProfiler, (d) $I^{segcolor}$ from our method, (e) contour image from our method, (f) I^{label} image from our method

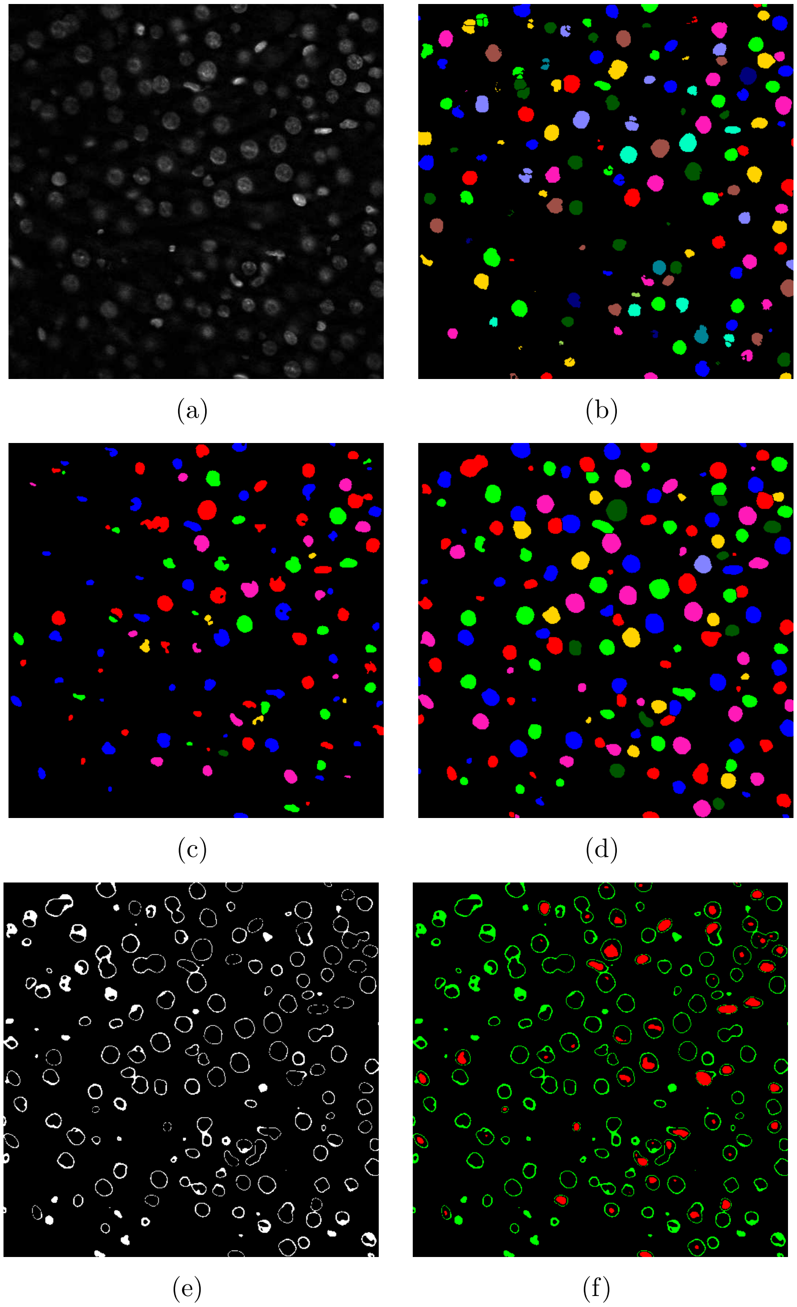


Fig. 4.8.: Comparison of slices of the original image volume and results obtained from SpCycleGAN [59], CellProfiler [102], and our proposed method of Data-II: (a) I^{orig}, (b) segmentation result from SpCycleGAN, (c) segmentation result from CellProfiler, (d) $I^{segcolor}$ from our method, (e) contour image from our method, (f) I^{label} from our method

we use the synthetic images obtained from model G to train a segmentation network and a classification network. In particular, since a 3D U-Net model M_1 is trained with paired sets I^{binary} and I^{syn}, this model is able to do segmentation. Similarly, the model M_2 is trained with paired sets $I^{shapemarker}$ and I^{syn} so that this model is able to generate I^{shape} from I^{orig} which contains the shape information of the nuclei. Lastly, post-processing on I^{seg} is done by using watershed in 3 dimensions and labeling the nuclei with different colors with 3D connected component. Then, a distance transform followed by thresholding is done on I^{seg} to obtain the contour images. The post-processing on I^{shape} is done by the Gaussian blurring with $\sigma = 2$ followed by thresholding with empirically selected thresholds. The thresholded images are then overlaid with the contour images to obtain I^{label}.

Our proposed method is compared with SpCycleGAN [59] and CellProfiler [102]. Note that both methods are tuned to produce the best results possible. Also, the SpCycleGAN is used together with a modified 3D U-Net whose training loss includes the Binary Cross Entropy (BCE) loss and the Dice loss [59]. Figure 4.7 and 4.8 show the comparison of the color-coded images obtained from the SpCycleGAN [59] with 3D U-Net for segmentation, CellProfiler [102] using the watershed for segmentation, and our proposed method. It is observed that the SpCycleGAN method tends to miss multiple nuclei and cannot capture the correct shape of the elongated nuclei while the CellProfiler method cannot distinguish and label a cluster of nuclei. Our method can find nuclei at the correct location with more accurate nuclei shape. Figure 4.7(e) and 4.8(e) shows the contour image generated from 3D U-Net and post-processing which labels the boundary of detected nuclei. Figure 4.7(f) and 4.8(f) is an example slice of I^{label} which exhibits nuclei classified as non-sphere shaped nuclei. The centroids of those nuclei are highlighted in red and the nuclei boundaries overlaid onto those centroids are highlighted in green. It can be observed that nuclei with red labels have more ellipsoidal shapes.

We used the object-based evaluation we presented in [97] to find the accuracy of the nuclei segmentation. Small objects removal is done to remove objects with small

Table 4.1.: Evaluation of the proposed method of Data-I and Data-II

	Microscopy Data-I			Microscopy Data-II		
	P	R	$F1$	P	R	$F1$
[102]	71.32%	65.25%	68.15%	81.21%	73.63%	77.23%
[59]	86.77%	68.62%	76.64%	89.32%	87.45%	88.37%
Proposed	76.99%	88.04%	82.14%	90.16%	95.82%	92.90%

sizes (fewer than 50 voxels). Manually annotate groundtruth is obtained using the ITK-SNAP [98] which generated distinct labels for each nucleus. The groundtruth we used here are a subvolume of Data-I with a size of $128 \times 128 \times 64$ and a subvolume of Data-II with a size of $256 \times 256 \times 32$. A true-positive, N_{TP}, is obtained if a nucleus has at least 50% overlap with its matching groundtruth. A false-positive, N_{FP}, is obtained if a nucleus has less than 50% overlap with its matching groundtruth. A false-negative, N_{FN}, is obtained if a nucleus in the groundtruth can not find a matching nucleus in the segmentation result. The F1 score (F1), Precision (P), and Recall (R) are then obtained as in [97]. Table 4.1 shows that our proposed method performs better than the compared methods with a higher F1 score. The object-based evaluation demonstrates that our method can segment and locate nuclei correctly.

4.4 Conclusions

In this work, we have demonstrated an approach for segmenting and classifying nuclei in 3D microscopy image volumes using synthetic training data. We first generated synthetic microscopy image volumes using the Recycle-GAN with a loss term using the Hausdorff distance. With the use of Recycle-GAN, three dimensional information can be fully utilized for generating synthetic image volumes. The use of the Hausdorff distance based loss function can refine the boundary of the generated nuclei by penalizing inaccuracies occurring around the boundaries of the synthetic nuclei. We then employed a 3D CNN for nuclei segmentation and another 3D CNN for nuclei classification based on the shape of the nuclei. The 3D CNN trained on synthetic data was capable of segmenting nuclei and generating volumes with nuclei shape information without fine-tuning. Our proposed method has the advantage to preserve nuclei shape and can be used to separate different types of nuclei within the same volume without additional information. In the future, we plan to produce groundtruth volumes with nuclei type information and evaluate the accuracy of nuclei type classification.

5. SYNTHETIC MICROSCOPY IMAGE GENERATION AND NUCLEI SEGMENTATION WITH STYLE TRANSFER

5.1 Background

Fluorescence microscopy is an optical microscopy for the visualization of fluorescence in living species of subcellular structures. Biologists are able to see more depth of the tissue through two-photon microscopy, a type of fluorescence precision microscopy. With a greater depth of penetration, large data are often generated by a two-photon microscopy. An automatic image analysis tool is required to extract significant biological information in three dimensions. The detection and division of biological structures such as nuclei within microscopic images is a common application. Nuclei containing microscopy images are difficult to examine, because of many factors.

The fluorescent multi-channel images vary significantly from those of natural images. The light and properties of a particular material in the scene decide color in natural photographs. A GAN needs to capture the connection between materials that make up a specific object and its colors, in order to produce realistic natural samples. In fluorescent images, by contrast, the light intensity in a given channel corresponds to the local concentration of the tagging pro-dye and the correlation between signals in the different Channels is an important piece of information about protein relation.

Segmenting cell nuclei in microscopy image volumes are challenging due to poor image quality, crowding of nuclei, and large variation in nuclei sizes and shapes. Recent advancement in deep learning has shown success in finding and distinguish nuclei under different image contexts. To reduce the intensive work of groundtruth nuclei in

3D, carefully crafted synthetic training image pairs are used to train Convolutional Neural Networks (CNN). Amongst methods for image generation, unsupervised image to image translation with Generative Adversarial Networks (GAN) has become a prominent choice.

In this work, we develop an unsupervised translation approach adapted from Adaptive Instance Normalization (AdaIN) [112] for synthetic microscopy image generation that utilizes three dimensional image information. Realistic microscopy image volumes with nuclei segmentation mask and nucleus boundary groundtruth volumes are generated. A subsequent 3D CNN with a regularization term that discourage detection out of nucleus boundary is used to detect and segment nuclei. Experimental results on our rat kidney dataset show the proposed method is competitive with respect to several state-of-the-art methods.

5.2 Proposed Method

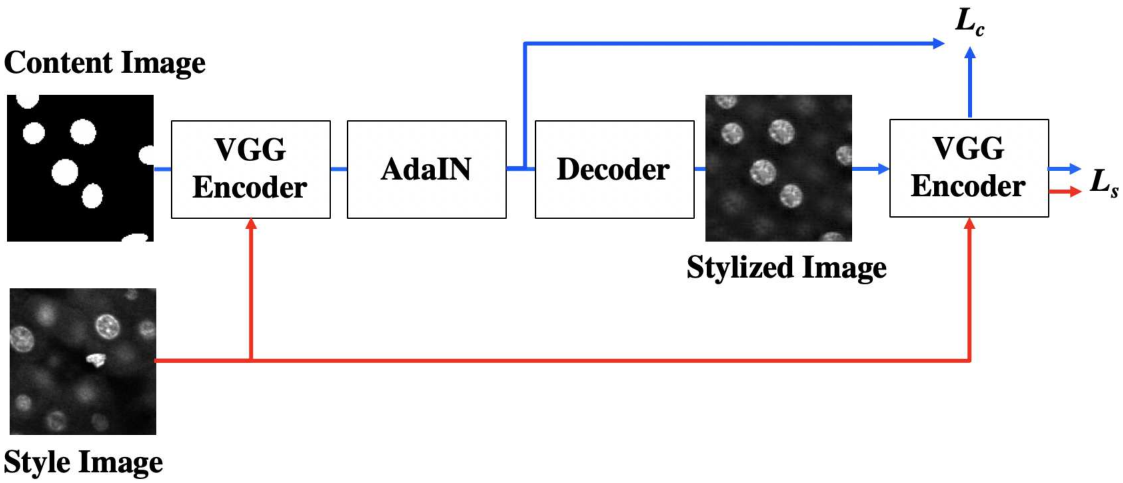
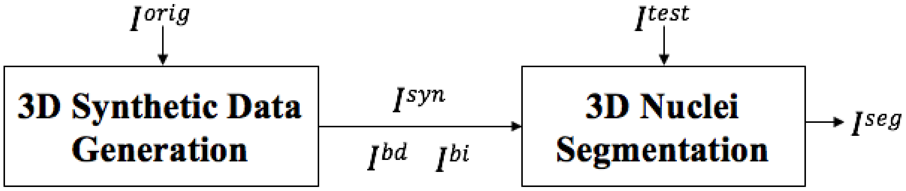
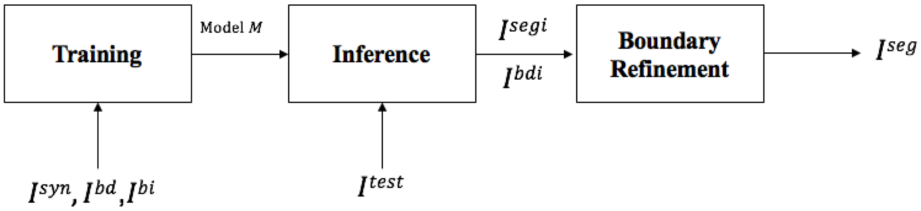


Fig. 5.1.: Block diagram of the proposed method

In this work, I denotes a 3D image volume of size $X \times Y \times Z$. Figure 5.1 and Figure 5.2(a) shows the block diagram of our proposed method. I^{orig} denotes a subvolume of



(a)



(b)

Fig. 5.2.: Block diagrams of the proposed method

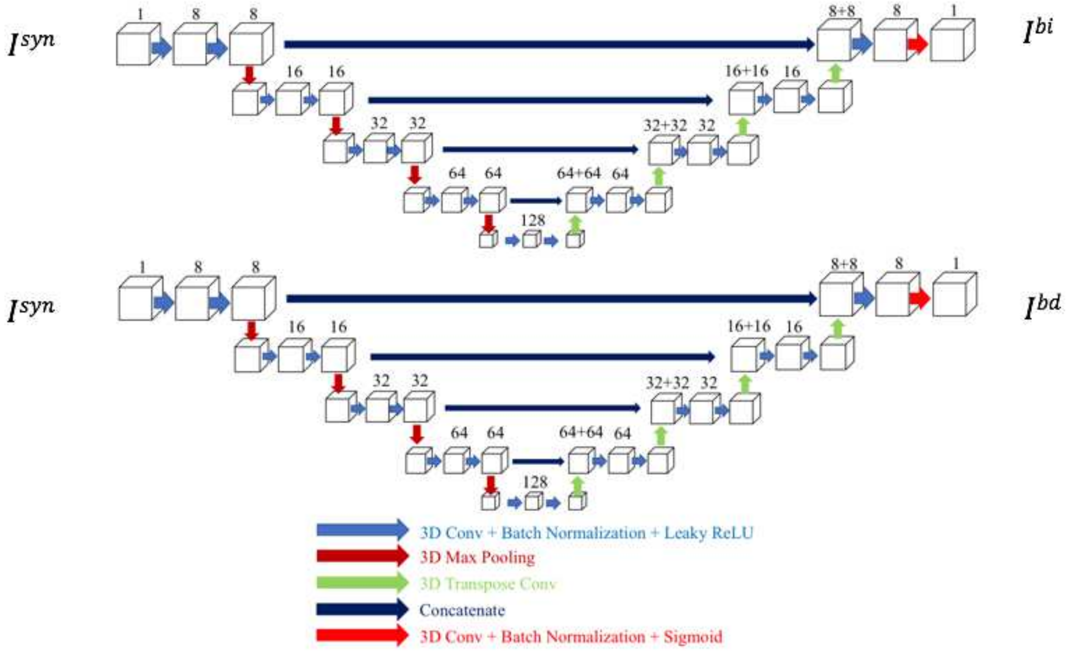
the original microscopy volume used for training, and I^{test} is a subvolume of original or synthetic microscopy volume that is used for testing. I^{syn} denotes a synthetic microscopy volume generated with 3D synthetic data generation based on the features of I^{orig}. I^{sty} denotes the stylized image generated with the style of the style image that is corresponding to the content image. I^{seg} denotes the segmentation result volume.

Synthetic binary volumes, I^{syn}, I^{bi} and I^{bd}, are generated first. I^{bi} is used together with I^{orig} to train a AdaIN model from which a generative model C is obtained. A different synthetic binary volume, I^{bi}, is then used to generate corresponding I^{syn} with the trained model C. For each I^{bi}, the corresponding I^{bd} is generated. In Figure 5.2(b), paired I^{syn} and I^{bd}, I^{bi} are used to train a CNN for nuclei segmentation and obtain a 3D CNN generative model M. Model M is used to transform I^{test} to its binary mask and boundary volume, I^{segi} and I^{bdi}. Then, by using Canny edge detection filter on I^{bi}, the resulting edge image is then compared with I^{bd}. The similarity of two edge images are measured and reported. The segmentation result is then refined by rejecting false positive detection outside of nuclei boundary.

Three dimensional synthetic data generation includes synthetic binary volume generation, synthetic boundary volume generation, AdaIN [112] training, and AdaIN inference. A synthetic binary volume is generated by drawing ellipsoidal shapes in 3D according to randomly selected locations of nuclei centroids and orientations.

The AdaIN layer is used to align the mean and variance of the content features with those of the style features. The input and output of an AdaIN layer is described as in Equation 5.1.

$$AdaIN(x,y) = \sigma(y)\frac{x-\mu(x)}{\sigma(x)} + \mu(y) \tag{5.1}$$



(a)

Fig. 5.3.: Architecture of the segmentation and refinement network structure

Figure 5.3 demonstrates the 3D CNN architecture for nuclei segmentation. For 3D convolution, the filter size is $3 \times 3 \times 3$. A $1 \times 1 \times 1$voxel padding is used to preserve the original volume size of the output of each convolution layer. Batch normalization [47] is used in each 3D convolution and the activation function is rectified linear units (ReLU). For downsampling path, 3D max pooling is used with a stride of 2, while a 3D transpose convolution is used in the upsampling direction between the layers. Information between the downsample path and the corresponding upsample path is passed by concatenation.

The training loss function is a linear combination of the Dice loss ($\mathcal{L}_{Dice}$) and the binary cross-entropy loss ($\mathcal{L}_{BCE}$) such that

$$\mathcal{L}_{seg}(T, S) = \mu_1 \mathcal{L}_{Dice}(T, S) + \mu_2 \mathcal{L}_{BCE}(T, S) \tag{5.2}$$

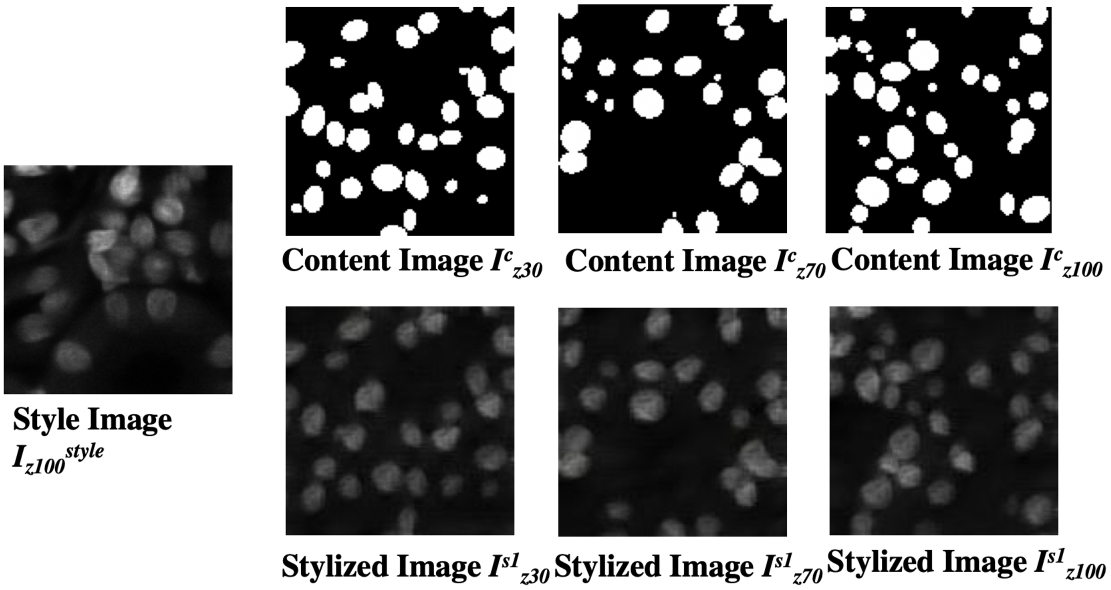
where

$$\begin{aligned} \mathcal{L}_{Dice}(T, S) &= \frac{2(\sum_{i=1}^{N} t_i s_i)}{\sum_{i=1}^{N} t_i^2 + \sum_{i=1}^{N} s_i^2} \\ \mathcal{L}_{BCE}(T, S) &= -\frac{1}{N} \sum_{i=1}^{N} t_i \log(s_i) + (1 - t_i) \log(1 - s_i), \end{aligned}$$

respectively.

Please note that T is the set of the targeted value of the groundtruth, and $t_i \in T$ is a targeted value at the voxel location i^{th}. Similarly, S is a binary segmentation probability map and $s_i \in S$ is the i^{th} voxel position probability map. Finally, N is the voxel total number and μ_1, μ_2 is the weights in Equation (5.2) between the loss terms. The network takes a $64 \times 64 \times 64$ grayscale input volume and produces the same size of the input volume for the classified 3D volume. In order to train our M model, V pairs of synthetic microscopy volumes, I^{syn}, and synthetic binary volumes, I^{label}, are used. Two separate 3D U-Net models are trained, producing I^{bi} which contains binary segmentation mask and I^{bd} which is the boundary information volume used for the refinement of results as describe in [113]

5.3 Experimental Results


Content Image I^{c}_{z30}
Content Image I^{c}_{z70}
Content Image I^{c}_{z100}
Style Image I_{z100}^{style}
Stylized Image I^{s1}_{z30}
Stylized Image I^{s1}_{z70}
Stylized Image I^{s1}_{z100}

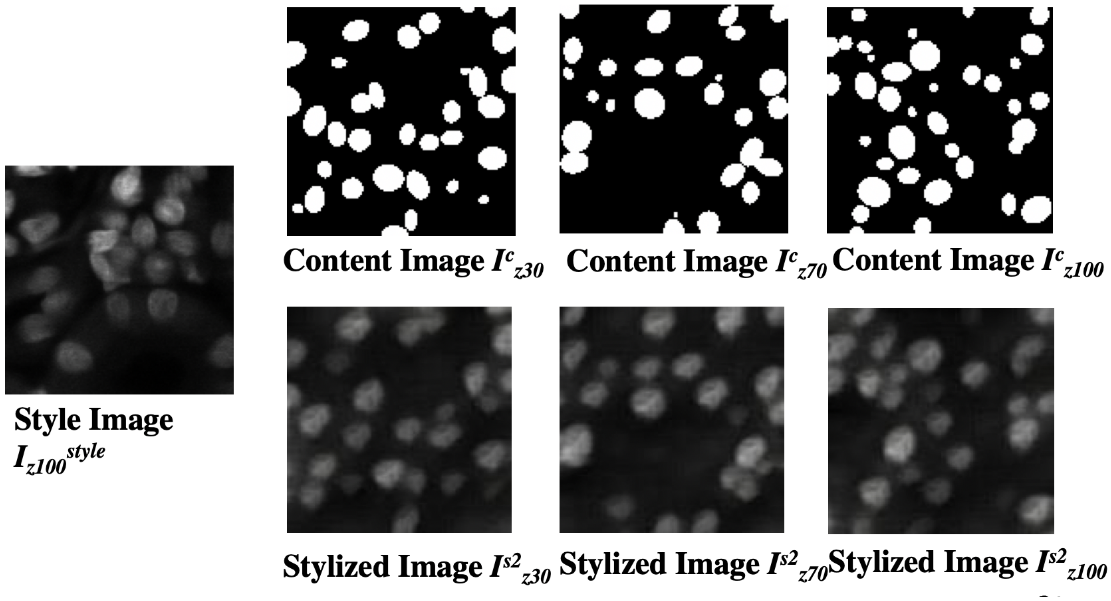
(a)

Fig. 5.4.: Comparison of the synthetic binary volume, the synthetic microscopy volume from style transfer, and the original volume

We tested our proposed method on two different rat kidney data sets. These data sets contain grayscale images of size $X = 512 \times Y = 512$. Data-III consists of $Z = 415$ images.

The AdaIN is implemented in PyTorch with default parameters given by [112]. A $128 \times 128 \times 128$ synthetic binary volume for Data-III denoted as $I^{bi_{Data-III}}$ and a $128 \times 128 \times 128$ subvolume of original microscopy volume of Data-III denoted as $I^{orig_{Data-III}}$ were used to train model $C^{Data-III}$. .

Because our modified 3D segmentation architecture uses volume size of $64 \times 64 \times 64$, we have divided the larger volumes into an adjacent, non-complete $64 \times 64 \times 64$, for $I^{bi_{Data-III}}$, $I^{bd_{Data-III}}$, and $I^{syn_{Data-III}}$. So we have 80 pairs of training volumes for each data to train our 3D segmentation network with synthetic binary volumes and the corresponding synthetic microscopy volumes. Remember that these 80 for each


Content Image I^{c}_{z30}
Content Image I^{c}_{z70}
Content Image I^{c}_{z100}
Style Image I_{z100}^{style}
Stylized Image I^{s2}_{z30}
Stylized Image I^{s2}_{z70}
Stylized Image I^{s2}_{z100}

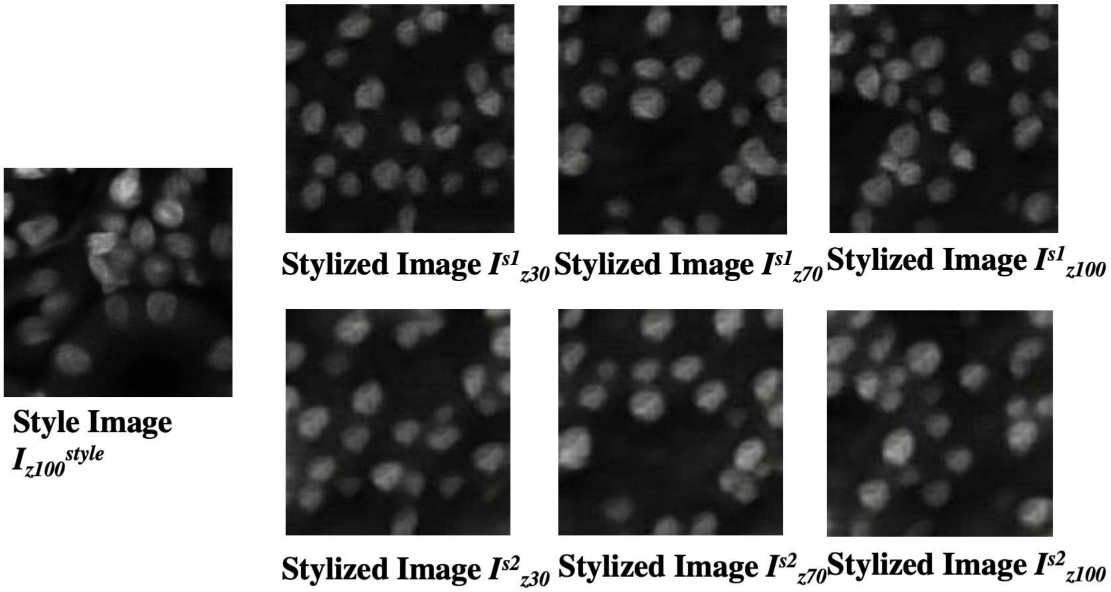
(a)

Fig. 5.5.: Comparison of the synthetic binary volume, the synthetic microscopy volume from style transfer, and the original volume

data is used to pair these synthetic binary volumes as groundtruth volumes with the corresponding synthetic microscopy volumes, generating $M^{Data-III}$.

The Adam optimizer [95] with learning rate 0.001 is used for the 3D segmentation network with PyTorch. We use different 3D synthetic data generation methods for the evaluation purpose. Furthermore, with the μ_1 and μ_2 settings, different loss functions are used. In addition, our updated 3D U-Net was compared to the [88] encoder-decoder architecture of 3D. Finally, with 3D connected components, small objects smaller than 20 have been eliminated.

Figure 5.4 shows the synthetic images generated by our proposed method. The left side indicates original style images whereas the top row shows synthetic binary image artificially generated, the bottom row shows the synthetic generated microscopy image. As can be seen from Figure 5.5 and 5.6, the synthetic images reflect characteristics of the original microscopy images such as background noise, nuclei shape, orientation and intensity.

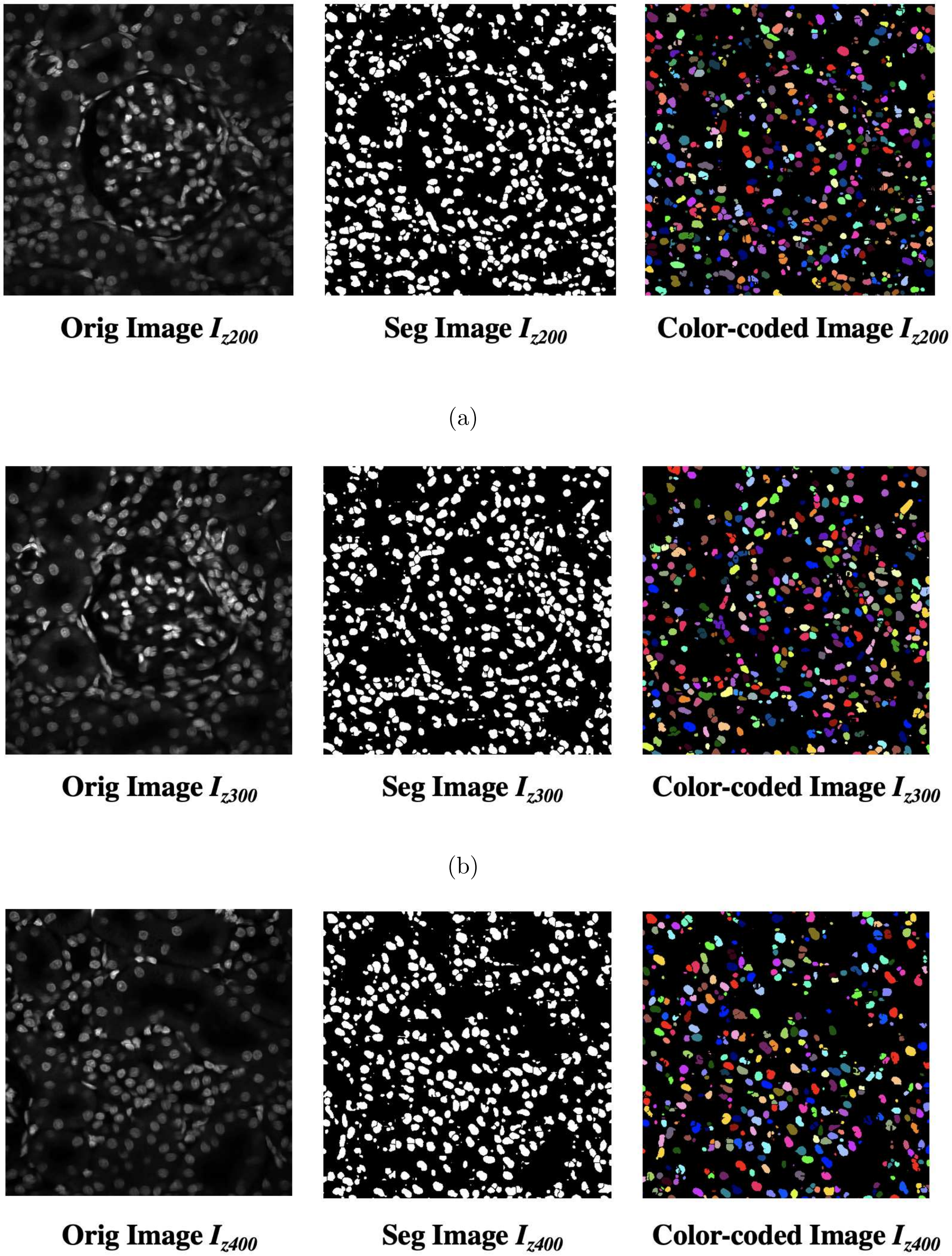

Style Image
I_{z100}^{style}
Stylized Image $I^{s1}{}_{z30}$
Stylized Image $I^{s1}{}_{z70}$
Stylized Image $I^{s1}{}_{z100}$
Stylized Image $I^{s2}{}_{z30}$
Stylized Image $I^{s2}{}_{z70}$
Stylized Image $I^{s2}{}_{z100}$

(a)

Fig. 5.6.: Comparison of the synthetic microscopy volume from style transfer model one, the synthetic microscopy volume from style transfer model two, and the original volume

Figure 5.7 shows the segmentation of result generated from the 3D segmentation network M. For the first row, the left images shows the original image, the middle image show the segmentation image of the proposed method after 3-way watershed, and the right side image shows the color-coded result of the segmentation result. Figure 5.7 compares the left images showing the original image, the middle image showing the result from this method, and the right side image showing the color-coded result of SpCycleGAN. As can be observed, the segmentation result can detect nuclei regardless of the brightness and blurring from the original image. Nuclei that are of non-sphere shapes can be detected. Nuclei that are close to each other are separated from each other in the color-coded image. In the compared method, the nuclei at the center of the images are grouped into one cluster, greatly lowered the detection accuracy. In the SpCycleGAN result, miss detection of nuclei around the corner regions are observed, also, adjacent nuclei are more often detected as connected.

By visual observation, the proposed method can segment and detect nuclei more accurately comparing to the other methods.



Orig Image I_{z200} Seg Image I_{z200} Color-coded Image I_{z200}

(a)

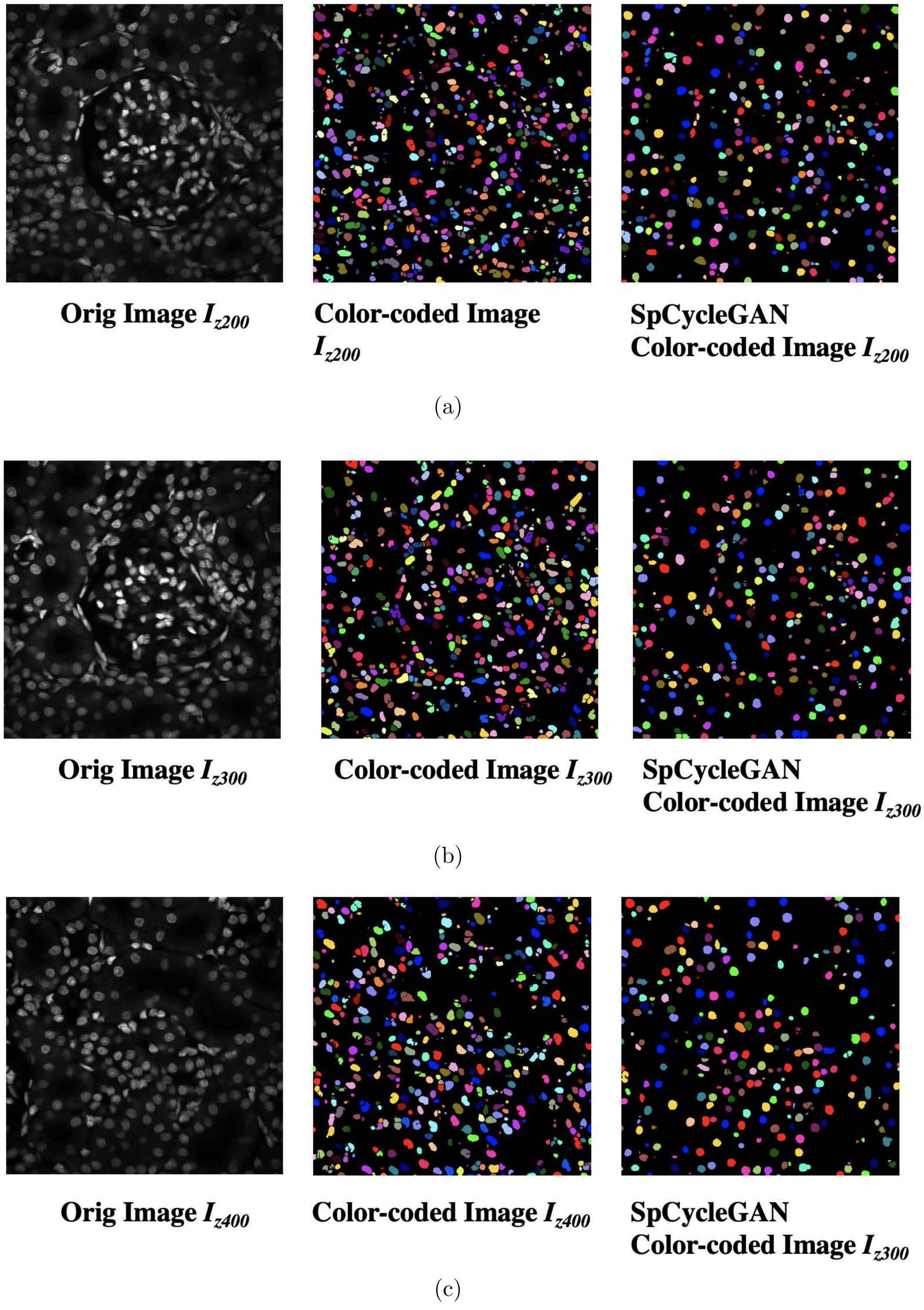
Orig Image I_{z300} Seg Image I_{z300} Color-coded Image I_{z300}

(b)

Orig Image I_{z400} Seg Image I_{z400} Color-coded Image I_{z400}

(c)

Fig. 5.7.: Comparison of slices of the original volume, the segmentation volume after 3-way watershed, and the color coded volume of the segmentation result (a) Data-III original volume, (b) Data-III segmentation volume after 3-way watershed, (c) Data-III color coded volume of the proposed method


Orig Image I_{z200}
Color-coded Image I_{z200}
SpCycleGAN Color-coded Image I_{z200}
Orig Image I_{z300}
Color-coded Image I_{z300}
SpCycleGAN Color-coded Image I_{z300}
Orig Image I_{z400}
Color-coded Image I_{z400}
SpCycleGAN Color-coded Image I_{z300}

(a)

(b)

(c)

Fig. 5.8.: Comparison of slices of the original volume, color coded volume of the segmentation result from this method, and the color coded segmentation volume from SpCycleGAN generated synthetic images (a) Data-III original volume, (b) Data-III color coded volume of the proposed method, (c) Data-III color coded volume of SpCycleGAN

6. DISTRIBUTED AND NETWORKED ANALYSIS OF VOLUMETRIC IMAGE DATA (DINAVID)

6.1 Background

Recent progress in microscopy technology has allowed the acquisition of large 3D volumetric data [114,115] including 3D multi-spectral data using fluorescence imaging.

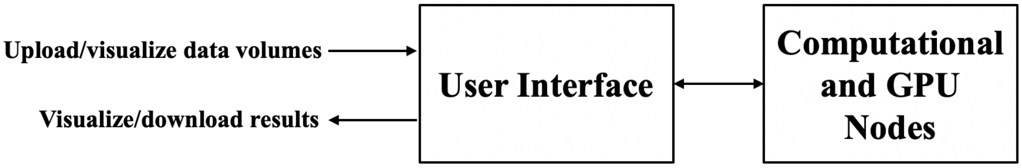

Upload/visualize data volumes
Visualize/download results
User Interface
Computational and GPU Nodes

Fig. 6.1.: Block diagram of the DINAVID system

Several challenges exist for the analysis of this 3D data. The first major challenge is the extraction of key information from a 3D volume. This can be a problem due to the characteristics of the biological structures, such as crowding of structures, blurred boundaries, and various noises. A common task for data analysis is to quantify the features of objects of interest. For example, image analysis tools are used to quantify the amount of fluorescence of different protein stains in the region around the cell [116]. CellProfiler [60] and ImageJ [117] are examples of popular image analysis packages in the biology community which we will discuss in more detail below. With tremendous expansion in the use of machine learning and deep learning, more tools use these techniques for extraction and quantification of important biological information. For example, U-Net [48] is a deep-learning architecture developed originally for

This is a joint work with Dr. Chichen Fu, Dr. Soonam Lee, Alain Chen, Changye Yang, and Liming Wu of Purdue University biomedical segmentation and is very popular for microscopy image segmentation. ImageJ [117] and CellProfiler [60] has been making progress in incorporating U-Net [48] and other machine learning tools. In some cases manual installation of libraries is needed which limits its use for non-expert users. In other cases [117], a user needs to provide the set of original images and corresponding groundtruth images to train the network.

The second challenge is to obtain representative and effective visualization. Since the visualization of living organisms and subcellular structures are vital for the analysis and understanding of related biological processes [118], traditional 2D visualization often lacks important perspectives. Effective 3D visualization is needed for a complete understanding of the data. A variety of ways have been used to achieve 3D visualization of an image volume, such as maximum intensity projection (MIP), three-dimension views, or cross sectional views [1]. We will review the commonly used visualization tools in the next section.

The final challenge is the computational requirements for large 3D multi-channel image analysis. The 3D microscopy volumes are often in the range of several gigabytes (GB)s to terabytes (TB)s of data. These 3D volumes require more computational resources (e.g. many GPUs) than a biologist may have access to in typical desktop or laptop computers especially if machine learning tools are being used for image analysis. With this increased amount of data analysis and computational complexity, there is a need for a more accessible, easy-to-use, and efficient network-based/cloud-based 3D image visualization and processing system. The Distributed and Networked Analysis of Volumetric Image Data (DINAVID) system is being developed to enable cloud-based analysis of microscopy images for biologists. The goal is to provide a system that biologists can use without worrying about managing the computational resources. DINAVID has a web user interface and requires that a user has a network connection of 100Mb/s and a less than $1000 graphics card [1] in the computer system they use to access DINAVID.

[1] An example is the NVIDIA GeForce RTX 2060.

DINAVID is designed for fast and accurate quantitative analysis of large 3D microscopy volumes. DINAVID supports traditional imaging processing tools such as filtering, thresholding, background subtraction, watershed [119], and morphological operations as well as deep learning-based 3D segmentation tools for fluorescent microscopy images of nuclei. Our system consists of network-based user interface and computing servers that contain high performance GPUs. It contains the functionality of an ImageJ plugin named VTEA [116] with a web-interface and leverage high-performance computing clusters and GPUs to support analysis of large scale microscopy volumes. Users are able to upload volumes and download analysis results via the network-based user interface. The system supports storage, management, quantitative analysis, visualization, and exploration of large image volumes, and provides real-time visualization. All data flow between DINAVID and the users is encrypted. In this work we will present an overview of the DINAVID system and compare it to other tools currently available for microscopy image analysis. A block diagram of the DINAVID system is shown in Figure 6.1.

6.2 Review of Existing Systems

In analyzing 3D microscopy volumes, a user would like to be able to visualize the volumes in 3D (or 2D) and then be able to use various image analysis methods to extract quantitative information about the structures in the volume. We below overview some of the existing systems and tools that are available for a user to analyze volumes. We will differentiate them not only by their capabilities to analyze volumes but also what biologists would need to manage using the systems tools with emphasis on whether the tools are network-based or require download and installation.

6.2.1 “Local” Image Analysis Systems

A “local” system is a set of tools that a user would need to install on their local computer to use. Open source local image processing packages are preferred by many biologists. Examples of these are ImageJ [117] and CellProfiler [60, 102]. ImageJ is an image processing system with installable plugins that enables many types of image processing and analysis. ImageJ provides support for displaying, editing, processing, and analyzing images in various formats. In addition, ImageJ is extendable via plugins and it is possible to use macros or scripts to execute repeating tasks. One challenge of working with ImageJ is that it does not have built-in 3D visualization. ImageJ has addressed this limitation by adding support for plugins such as 3D Filter [120] and 3D Object Counter [121]. ImageJ has limited support for user defined workflows, but has a well-developed and mature suite of image processing, analysis and visualization tools. One example of a set of tools based on ImageJ is the Volumetric Tissue Exploration and Analysis (VTEA) [116] for image processing, segmentation, and cytometry on 3D data. CellProfiler [60, 102] is another popular local open-source image processing tool that integrates image processing, analysis, and visualization. It has many modules that can be added by a user to define a particular analysis workflow. In CellProfiler, users can start by modifying existing workflows or choosing to develop their own workflows. The pre-built workflows are designed for various analysis tasks such as DNA gel analysis and cell counting. CellProfiler is user-friendly for biologists with repeating well-defined analysis without many parameter adjustments. Both ImageJ and CellProfiler are what we refer to as “local” systems in that they require installation on a user’s computer which means the computational capabilities are limited by the user’s “local” computer.

With respect to 2D/3D visualization several of the tools are also local. One example that can provide cross-sectional views is known as 3D Viewer [62] and is a plugin of ImageJ. Since cross sectional viewing could not display the objects-of-interest in 3D, a user needs to observe back and forth along the cross-sections to estimate the 3D surface. Thus, true 3D rendering is preferred. An example 3D rendering is Voxx [63] which can display entire stacked images as a single volume and performs 3D rendering to visualize 3D objects. Similarly, Volume Viewer [64], a plugin in ImageJ, does real-time rendering to display stacked images. These 3D rendered viewing methods can provide an interactive display of 3D volume so that a user can interactively investigate the 3D volume. One of the most popular approaches for volume rendering is the Open Graphics Library (OpenGL) [66]. An example of a OpenGL volumetric rendering tool is known as ImageVis3D [65]. ImageVis3D also supports other viewing methods including 2D slices viewing and maximum intensity projection (MIP) viewing. Additionally, Agave [122] is a software package that uses path trace rendering for speeding up the process in order to support large volume rendering.

6.2.2 “Network-Based” Image Analysis Systems

The above mentioned traditional image visualization and analysis tools are often limited to users with special research training, and the installation and maintenance of such software can be burdensome for some users. Thus, it is been observed that there is a growing trend for network-based image visualization tools as well as cloud-based image analysis tools. By a network-based (or cloud-based) system we mean a system that allows a user to process and visualize 3D volumes remotely. This does not require that they install and execute anything on their local computational resources. Users access the system via the web.

One example of a cloud-based system is Apeer [67] from Zeiss. Apeer provides image processing tools through the use of Python-based modules. Users can create their own workflow or choose to use pre-built workflows. In terms of building workflows, Apeer is similar to CellProfiler in the ways each module is added and executed sequentially. Apeer is powered by Azure from Microsoft for cloud computation and Git for version control and code collaboration. For the purpose of facilitating the analysis of large size image data, the Web Image Processing Pipeline (WIPP) [68] was developed by the National Institute of Standards and Technology (NIST), who recently reported the development of plugins for cloud-based microscopy image analysis [69]. WIPP provides users with modules for image processing, image feature extraction based on various image characteristics, and machine learning-based regions of interest (ROI) clustering. WIPP is expanding its capability for analysis by adding a module for statistical modeling. Another example of open source cloud-based platform is BisQue [123], which is designed for sharing and exchanging of published datasets. BisQue has a web interface that allows choosing and visualization of images. It also provides access to image analysis tools that come from CellProfiler.

There have been several examples of 2D/3D visualization tools that use the Web Graphics Library (WebGL) [124,125]. WebGL is derived from the OpenGL Embedded Systems 2.0 [125] and has a lighter computational load than OpenGL [66,126]. Due to the lighter computational load and its compatibility with any web browser, WebGL has become a popular tool for the network-based systems [127].

BioWeb3D [70] is a network-based 3D visualization tool that uses WebGL. The limitation is that the data can only be visualized with each pixel as particles or with simple lines drawn between points. 3D Cell Viewer [71] from the Allen Cell Institute is a more advanced example that uses WebGL and ray-marched path tracing for 3D volume rendering and provides comprehensive imaging parameter tuning and channel toggle functions. With preloaded segmentation of cells, 3D Cell Viewer allows users to choose to turn on or off the visualization of a cell based on its characteristics such as location and intensity. So far, 3D Cell Viewer is limited to preloaded volumes of segmented cells. Similarly, Neuroglancer [72] allows examining 2D and 3D visualizations of data from different kinds of microscopes. Neuroglancer is more suitable for developers who are familiar with web development for a customized display of data for their projects.

6.3 An Overview of DINAVID

The basic components of the DINAVID system are shown in Figure 6.1, which includes a user interface and computational and GPU nodes. Through a simple, easy to understand user interface, one can upload or visualize 3D volumes. Operations such as image analysis and visualization are sent to the high-performance computational

Table 6.1.: Comparison of Microscopy Image Analysis Tools I

Method	Local/ Network	2D Visualization	3D Visualization	Upload/ Show User's Volume
3D Cell Viewer [71]	Net	Yes	Yes	No
Apeer Core [67]	Net	Yes	Yes	Yes
WIPP [68]	Net	Yes	No	Yes
BisQue [123]	Net	Yes	Yes	Yes
bioWeb3D [70]	Net	No	Yes	Yes
Neuroglancer [72]	Net	Yes	Yes	No
ImageVis3D [65]	Local	Yes	Yes	Yes
Agave [122]	Local	Yes	Yes	Yes
CellProfiler [60, 102]	Local	Yes	No	Yes
VTEA [116]	Local	Yes	No	Yes
DINAVID	**Net**	**Yes**	**Yes**	**Yes**

Table 6.2.: Comparison of Microscopy Image Analysis Tools II

Method	Image Processing Tools	Segment Objects	Save Results
3D Cell Viewer [71]	No	No	No
Apeer Core [67]	Yes	No	Yes
WIPP [68]	Yes	Yes	Yes
BisQue [123]	Yes	Yes	Yes
bioWeb3D [70]	No	No	No
Neuroglancer [72]	No	No	No
ImageVis3D [65]	No	No	No
Agave [122]	No	No	No
CellProfiler [60, 102]	Yes	Yes	Yes
VTEA [116]	Yes	Yes	Yes
DINAVID	**Yes**	**Yes**	**Yes**

and GPU nodes. After processing the resultant volumes are available to the user via the interface to be visualized or downloaded by the user.

The architecture of the system is composed of the user interface and the DINAVID computational engines. The user interface was designed with web development tools that support interactive controls, real-time updating of processed information, and displaying images or image volumes. The visualization system allows for real-time interactive control and utilizes GPU acceleration of the user's local GPU in their desktop or laptop when possible. We assume that the user has a network connection of 100Mb/s and a graphics card with modest GPU capability. This provides reasonable interactivity and response time between DINAVID and the user. The DINAVID user interface system is responsible for managing the data and processed results and queuing of tasks from multiple users and between the computational nodes. The user interface also manages administration of DINAVID with user authentication, priority handling, file locking, secure login and logout, and changing of user credentials when needed.

The functionalities of the system include image and data analysis tools, 2D/3D visualization, and several auxiliary tools. The image analysis tools include Gaussian filtering, median background subtraction, median filter, thresholding/clamping, morphological operations, rolling ball subtraction, and machine learning segmentation tools. The statistics of segmented cells can be generated and displayed on a scatter plot. The scatter plot can be used to distinguish cells in certain statistical ranges and find gated cells in the original image, providing insights for further analysis.

The 2D visualization capability can load and adjust colors for multi-channel images, real-time previewing of image adjustments such as brightness tuning, and previewing of the image analysis operations. 3D visualization renders user defined subregions of an image volume and displays it in a separate window. 3D visualization has separate adjustments that allows for quick changing of the appearance of the rendered volume. DINAVID also provides a bandwidth testing function that will warn the user if their network bandwidth does not meet the 100 Mb/s suggested requirement, which means the system will not load images and perform functions with a high degree of interactivity.

6.4 Comparison of Systems

We compare characteristics of the above mentioned systems with DINAVID in Table 6.1 and Table 6.2. The comparison is in terms of if the system is local or network-based, its ability to do 2D or 3D visualization, and if it allows for uploading or visualizing a user supplied volume. We also compared the availability of image processing tools, the capability of object segmentation without needing to install external packages, and if it allows a user to save and download their processed results.

Most network-based systems lack image analysis tools while many local-based solutions are not as versatile in its visualization functions. 3D Cell Viewer [71] from the Allen Cell Institute is network-based and is capable of interactive 3D visualization of a single cell or a group of cells from their provided library of cell selections. Thus it is not able to visualize or process a user's own volume. Apeer [67] Core and WIPP [68] are both network-based which is targeted for processing of images based on user defined pipelines. Apeer only support 3D visualization through a separate viewer and it only support segmentation through user installation of external Python plugins. WIPP does not support 3D visualization. BisQue [123] has the ability for 2D/3D visualization but it only supports image analysis and segmentation through integrated CellProfiler [102] functions. BioWeb3D [70] and Neuroglancer [72] are both network-based software made for purpose of the demonstration of 3D visualization only. ImageVis3D [65] and Agave [122] are both local-based software for visualization of data, thus limited in their image processing abilities. CellProfiler and the VTEA [116] plugin from ImageJ [117] are both local-based software targeted for image analysis and qualitatively analysis which cannot render image volumes in 3D natively. Overall, most network-based solutions lack the capability of image analysis while many local-based solutions are not as versatile in their visualization functions. It can be observed that our method is able to perform the essential functions of 2D/3D visualization along with image processing, user uploading, and result downloading.

6.5 Architecture/Functionalities of DINAVID

In this section we describe in more detail the architecture and functionalities of the DINAVID system. The architecture of DINAVID include the user interface and the DINAVID server. The functionalities of the system include 2D visualization, 3D visualization, image processing tools, and quantitative data analysis tools.

6.5.1 User Interface

To make DINAVID accessible and easy to use, the staring point for using DINAVID is a web site that is accessed with a web browser. We use Apache [128] to establish the secure connections between the DINAVID system and the browser of the user. The biggest advantage of using DINAVID is that there is no need for a user to download and install anything locally on the user's system. The tools that the user interacts with are implemented in Hypertext Markup Language (HTML) [129], JavaScript [130], and Cascading Style Sheets (CSS) [131]. HTML is used to define the content and CSS is used to define the style and layout of HTML elements of the system web interface, while JavaScript provides user interaction and dynamic content updating within the browser. Examples of JavaScript enabled functions including hiding certain areas of the content, controlling multimedia, animating images, and sending the information to the system. Django [132], a Python-based web framework, is used to create the web interface and manages the communication between a user's request and the system response. Once the user uploads an image volume, they can interact with the DINAVID tools to process their volume. The processed image volume can then be visualized in the user's browser.

6.5.2 Data and User Request Handling

Once the user uploads volumes to DINAVID, Django [132] is used to process user requests and to schedule and load data for processing on the computational and

GPU nodes. Users send HTTP [133] requests to Django through their browser. For example, a request for displaying images will cause Django to forward the request to the appropriate "view." Each "view" is a Python script that defines how to process the request. During the processing of the requests, the views may interact with the "model" script for reading or writing data depending on what is required. For example, the "view" needs to know what images/volumes need to be retrieved from the system storage for the requested operation. Django will then return a response to the user's browser, often by dynamically creating an HTML page and by inserting the retrieved data into placeholders in an HTML template.

The DINAVID system employs Celery [134] to support task scheduling such as task queuing and distributed message passing. Task queues are used as a strategy to distribute the workload between multiple threads or computational nodes. Tasks can execute asynchronously (in the background) or synchronously (wait until ready). The DINAVID system supports parallel task processing for multiple users. This is achieved by RabbitMQ [135], a message broker, to accept and forward messages. The combination of RabbitMQ and Celery can achieve the function of managing and scheduling tasks from different users. It can automatically distribute the user's data to different computational nodes for processing. The resultant information will be stored in the system storage for users to access when needed.

6.5.3 Image Processing and Analysis

A common workflow in image processing for tissue cytometry includes image processing, segmentation, previewing and analysis. [116]. Processing of the data is often required to correct errors in image capture or to facilitate further analysis. Also, the processing of the data is often required to obtain the desired appearance of the microscopy volume for further analysis.

DINAVID supports the many image processing and operations as we indicated above. Here we describe these in a bit more detail. For Gaussian filtering to smooth or blur an image we allow the user to adjust the filter kernel size using the parameter σ, the standard deviation of the Gaussian filter. The median filter is used to remove noise from the image and we allow a user to adjust the window size of the kernel. Median background subtraction is used to remove background noise from an image by subtracting a median filtered image from the original image. Thresholding is also available including simple thresholding and Otsu's method [96]. Clamping removes low intensity values in the image. Pixels whose intensity is lower than the threshold are set to 0, while pixels whose intensity is greater than the threshold remains the same.

2D morphological operations for binary images are available to determine how local contents of the image are shaped relative to a given flat structuring element. The structuring element is defined by its shape and size with erosion, dilation, opening, and closing available. Erosion is used to shrinks objects, while dilation expands objects in a binary image. Opening is used to remove small objects, while closing removes small holes in a binary image. Rolling ball [136] is one of the options for background subtraction. We use a Python version implemented by [137]. Users can adjust the radius of the ball used in the rolling ball background subtraction. All of these functions are implemented via OpenCV [138], with the exception of rolling ball subtraction. DINAVID also allows the user to select Otsu's method [96] to determine a threshold value for the threshold operation. The user can select any combination of the operations or repeat them as needed. As in VTEA, the user can add as many of the above steps or repeat them and able to choose parameters for each operation.

6.5.4 Segmentation

In order to quantify the features of individual cells, it is important to identify the boundaries to distinguish between neighborhood cells. Since the boundaries of cells are not easy to segment directly, the workaround is to segment nucleus then find the region immediately around the nucleus. To quantify the protein label associated with each cell, segmentation is done followed by quantifing the amount of fluorescence of different colors in the region around the cell.

Segmentation of each cell is needed to quantify the features of individual cells. But the boundary of each individual cell is often difficult to find [116]. DINAVID took the approach of finding the cell nuclei first and then use the region immediately around the nuclei to represent the cells. This approach is also used in researches like [116] and [139].

The boundary of the cells is often hard to define in fluorescent microscopy, but cell nuclei are much easier to find [116]. Segmentation of cell nuclei is used instead and then the region immediately around the nucleus is considered as a part of that cell. DINAVID adopts this approach for segmentation.

Two nuclei segmentation tools are currently available in DINAVID, watershed [119] and a deep learning-based method known as DeepSynth [118]. Watershed was implemented via Scipy [140] and Scikit-Image [141]. The input to our watershed is assumed to a binary image.

A deep learning-based nuclei segmentation method, DeepSynth [59, 118], is implemented in our system. This method is unique in the sense that the segmentation system only uses synthetic data for training [59, 109, 110]. This method consists of two stages: 3D synthetic data generation and a 3D convolutional neural network (CNN) for 3D segmentation. More specifically, 3D synthetic data is generated using a spatially constrained cycle-consistent adversarial network (SpCycleGAN) without the need of manual segmented groundtruth volumes [59,109,110]. The synthetic data generated from the SpCycleGAN reflects characteristics of the original microscopy volumes such as background noise, nuclei shape, orientation, and intensity. The synthetic data is used to train our 3D segmentation CNN. Five pre-trained versions of DeepSynth [118] segmentation are available for users in DINAVID.

Since some image analysis processes require a long execution time, working directly on the entire image volume is often impractical or will take too long. A better practice is to first test the analysis on a sub-volume that shows results quickly before processing the entire volume. A previewing function is added for the image processing and segmentation steps. Previewing the image processing or segmentation steps is done on the selected nuclei channel on a user-selected region of interest on a single slice.

6.5.5 Visualization

DINAVID supports both 2D and 3D visualization. A user may upload a volume with as many 19 3D channels. For 2D visualization we display a slice from a 3D section of all of the channels.

Each channel in a 2D slice is assigned a color. The user can choose a color for each channel individually, a default color scheme is also provided. The maximum possible intensity in each channel is assigned to the selected color, while lower intensity values are assigned colors that are scaled to ratio of their relative intensity values to the maximum possible intensity. A maximum projection within each of the red, green, and blue components of the colors assigned to each channel is applied to generate the final color image that is displayed.

The default colors applied to each channel, in order, are = ['#0000ff', '#ff0000', '#00ff00' , '#f58231', '#4363d8', '#911eb4', '#42d4f4', '#f032e6', '#bfef45', '#fabebe', '#469990', '#e6beff', '#9a6324', '#fffac8', '#800000', '#aaffc3', '#808000', '#ffd8b1', '#000075']. This is approximately [primary blue, primary red, primary green, orange, blue, purple, cyan, magenta, lime, pink, teal, lavender, brown, beige, maroon, mint, olive, apricot, navy]. This list is adapted from [142].

DINAVID also has the capability of adjusting gamma, brightness, and offset for each of the individual channels, using Equation 6.1:

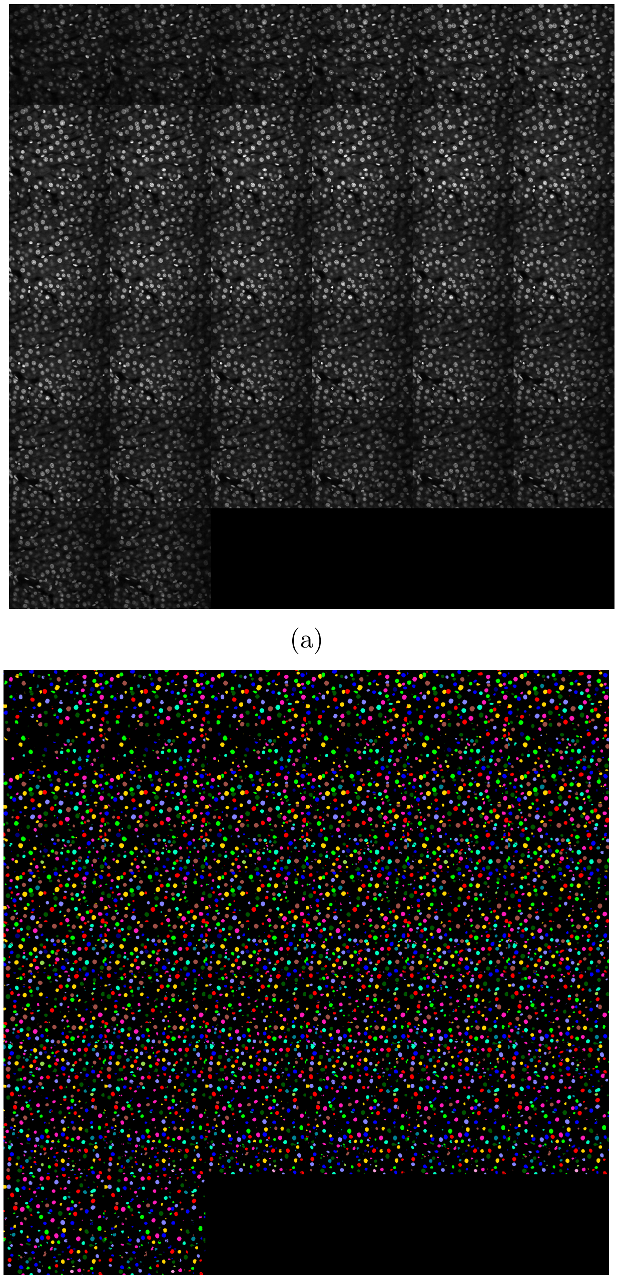
$$I^{Adjusted}(x,y) = max(min(B \times \frac{I^{orig}(x,y)}{255}^{\gamma} + C), 0), 1) \times 255 \tag{6.1}$$

where B is brightness, C is offset, and γ is gamma.

Whenever a visualization setting is changed, the change to the 2D slice is instantaneous. This is accomplished via an Asynchronous JavaScript and XML (AJAX) [143] request from the browser when a change in visualization parameter is detected. When a change is detected, the browser sends an AJAX request to the system. The system takes the user input and generates a new image with the new visualization parameters and sends this image back to the user's browser. Each image corresponds to a single slice of the 3D volume. Each AJAX request only handles one slice of the volume for efficiency. If the user wants to view different slices, a new image is generated each time a new slice is examined.

As described above 3D visualization uses WebGL [124] for interactive rendering and visualization. Volume rendering with ray casting is then implemented with WebGL for the ability to "see through" the 3D image volume. We first combine the images to be visualized into one large mosaic file which allows the data to be more accessible to the rendering engine. The tool is built with the goal of fast and accurate rendering that is capable to work on large data. The visualization is real-time and interactive and also supports a user adjusting parameters.

To generate a 3D visualization of a region of interest, the user needs to select a region on the slice by dragging a box that represents the region of interest. DINAVID generates the necessary files for rendering while the user's browser handles the rendering using local GPU acceleration in the users computer if available. The files needed for rendering are a 2D tiled image of the 3D subvolume and a JavaScript Object Notation (JSON) file that describes the size and shape of the subvolume. The rendering engine is a JavaScript library known as ShareVol [144] using WebGL. Similar to 2D visualization, we have the capability for adjusting gamma, brightness, and offset for each of the red, green, and blue channels as seen in Equation 6.1. Note that the input of the ShareVol is always a grayscale image and colors are randomly selected to differentiate between each 2D slice when all slices are stacked together in 3D. The colors present in the 3D visualization should match the colors used in the 2D visualization,



(a)

(b)

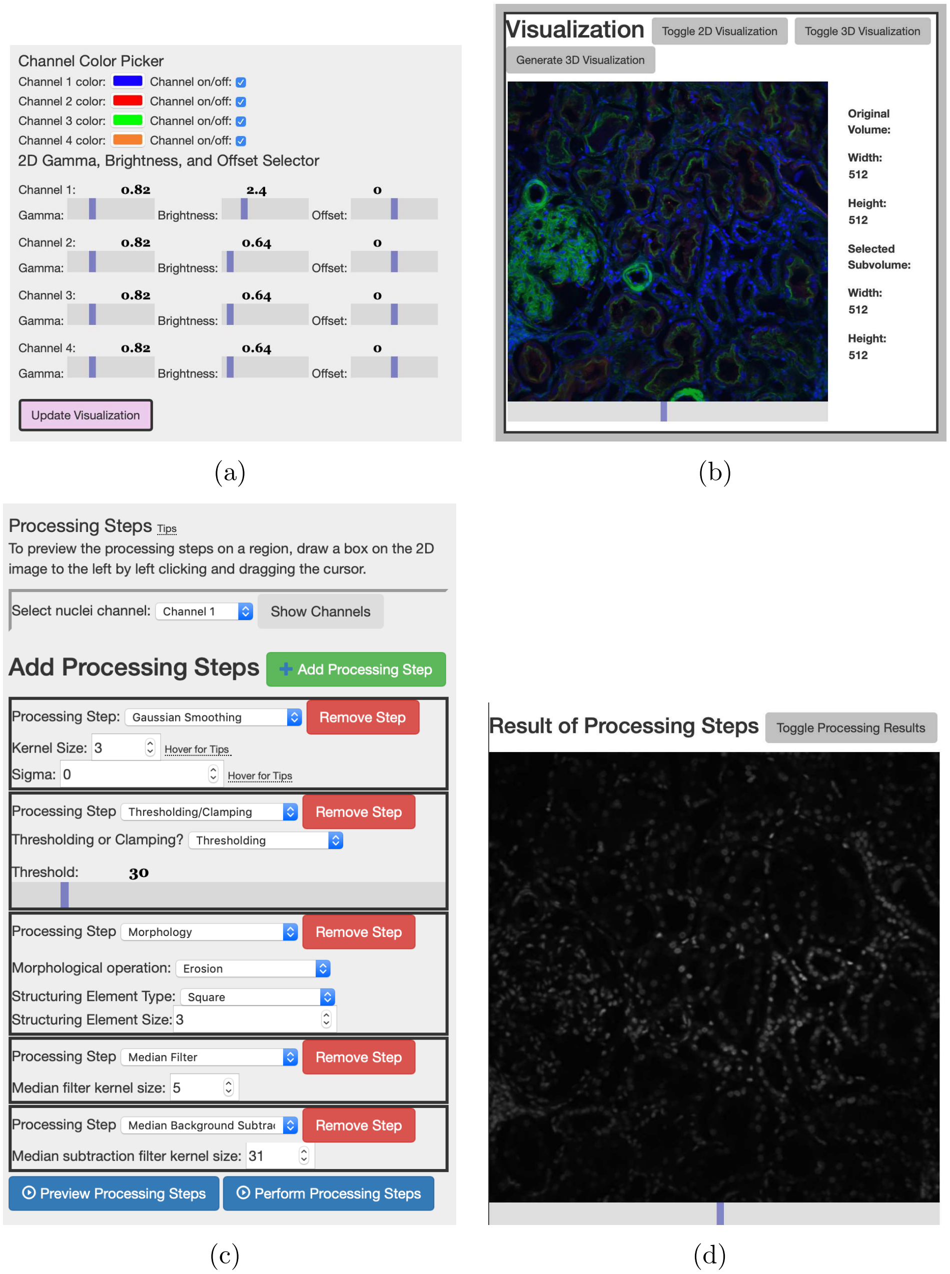
Fig. 6.2.: Examples of the mosaic images of original and color labeled of the nuclei dataset (a) Mosaic image of original volume, (b) Mosaic image of color labeled segmentation results.

so we had to modify ShareVol so that it accepts color images as input. To achieve developing the 3D volume rendering, we use a framework similar to lightweight volume viewer in WebGL called “ShareVol” [144]. For the dependencies, Javascript matrix and vector library for real time 3D graphics via WebGL called “glMatrix” and a lightweight graphical user interface for changing variables in JavaScript called “dat.GUI” are used. Here, the glMatrix supports the capability of handling the 3D volume with rotating (mouse left click) and panning (mouse right click) based on client input. Since this 3D volume rotating and panning motion are instantaneously and therefore real time 3D interactive visualization becomes possible. Additionally, there are red, green, and blue axes located at the volume to indicate which sections are displayed to help the client understand between 3D volume and 2D cross sections. Also, the dat.GUI supports the graphic user interface and allows parameter changes in the windows while the visualization is running. Again, this functionality is necessary for a client who would like to interactively adjust the parameters until the volume is displayed in the best setting.

One drawback in using WebGL is that WebGL does not support 3D textures because it is the lightweight version of OpenGL. Hence, a typical OpenGL approach such as loading the image stacks to generate a 3D texture input becomes impractical [125]. Alternatively, the stack of images are saved as a 2D texture with a mosaic configuration is described in [145]. Two examples of the mosaic images of original and color labeled segmentation results are shown in Figure 6.2. As can be seen in Figure 6.2, this dataset contains 32 images stacked to form an original volume and color labeled segmentation volume, respectively. DINAVID is also able to color label a segmented volume.

6.5.6 Quantitative Image Analysis

Cellular neighborhood contains critical information about the cell in fluorescent microscopy [116, 146]. For example, in [146] cellular neighborhood had a profound impact on the expression of protein receptors in immune cells. DINAVID estimates multiple statistics in a cellular neighborhood of each cell and generates a scatter plot.



Channel Color Picker

Channel 1 color: 🟦 Channel on/off: ☑

Channel 2 color: 🟥 Channel on/off: ☑

Channel 3 color: 🟩 Channel on/off: ☑

Channel 4 color: 🟧 Channel on/off: ☑

2D Gamma, Brightness, and Offset Selector

Channel 1: Gamma: 0.82 Brightness: 2.4 Offset: 0

Channel 2: Gamma: 0.82 Brightness: 0.64 Offset: 0

Channel 3: Gamma: 0.82 Brightness: 0.64 Offset: 0

Channel 4: Gamma: 0.82 Brightness: 0.64 Offset: 0

Update Visualization

(a)

Visualization

Toggle 2D Visualization Toggle 3D Visualization Generate 3D Visualization

Original Volume:

Width: 512

Height: 512

Selected Subvolume:

Width: 512

Height: 512

(b)

Processing Steps Tips

To preview the processing steps on a region, draw a box on the 2D image to the left by left clicking and dragging the cursor.

Select nuclei channel: Channel 1 Show Channels

Add Processing Steps + Add Processing Step

Processing Step: Gaussian Smoothing Remove Step

Kernel Size: 3 Hover for Tips

Sigma: 0 Hover for Tips

Processing Step Thresholding/Clamping Remove Step

Thresholding or Clamping? Thresholding

Threshold: 30

Processing Step Morphology Remove Step

Morphological operation: Erosion

Structuring Element Type: Square

Structuring Element Size: 3

Processing Step Median Filter Remove Step

Median filter kernel size: 5

Processing Step Median Background Subtra Remove Step

Median subtraction filter kernel size: 31

Preview Processing Steps Perform Processing Steps

(c)

Result of Processing Steps Toggle Processing Results

(d)

Fig. 6.3.: Examples of 2D image processing and visualization (a) Panel to adjust visualization parameters, (b) 2D Image visualization, (c) Panel to select image processing steps and adjust parameters, (d) An example slice after a Gaussian blurring operation

All the segmented cells are plotted in one scatter plot based on the statistics. Different types of cells might have different statistics in their cellular neighborhood. Ideally, a specific type of cell can be isolated by a region of interest (ROI) on the scatter plot. DINAVID allows users to choose an ROI on the scatter plot and highlights the cells inside the ROI on the original image volume. Users are able to see the distribution of a specific type of cell inside the imaged biostructure [116, 139].

We define the cellular neighborhood as the pixels surrounding each nucleus. With the segmented nuclei masks, pixels surrounding each nucleus are extracted using morphological dilation. Nuclei masks of each cell are dilated by a user-defined thickness. The difference between the resulting dilated mask and the original nuclei masks represents the pixels around the nuclei. Note the previous step only captures the layer of pixels wrapping outside the nuclei and the pixels belonging to the nuclei are not included. For each cell, we then extract six statistics (minimum, maximum, mean, standard deviation, sum, and count) using the pixels surrounding the cell nuclei. If the volume has more than one channel, for each individual channel all six statistics are calculated. DINAVID also generates a spreadsheet containing the estimated statistics of each cell.

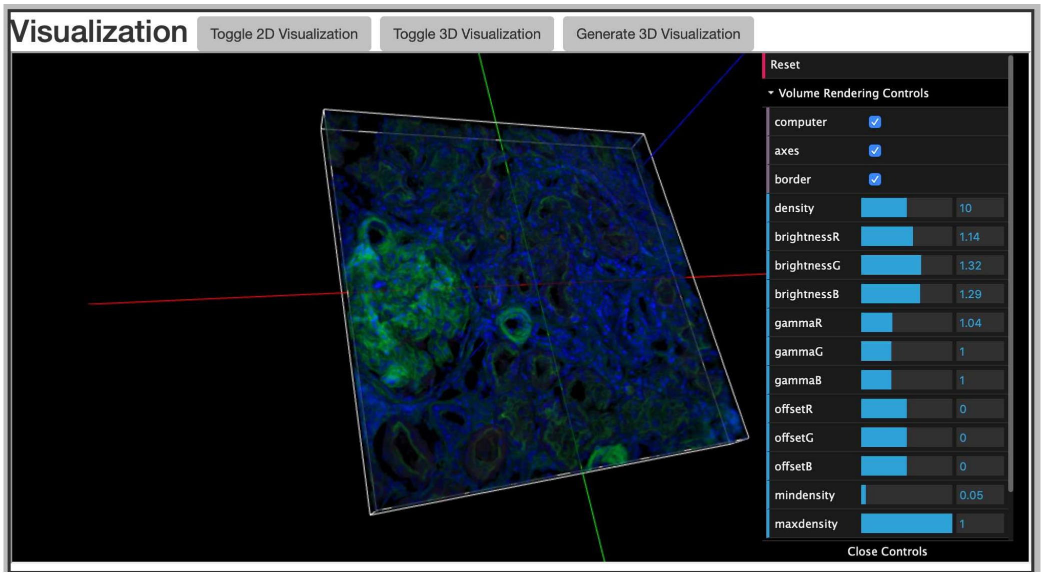
Using the estimated statistics of each cell, DINAVID generates a 2D scatter plot with user defined statistics. For example, the default statistics on the x-axis is channel one mean and the default statistics on the y-axis is channel two means. Each segmented cell from the volume will be part of the scatter plot. Users can change the type of statistics and channels used on the scatter plot axis. A region of interest (ROI) can also be selected on the scatter plot. DINAVID takes the cells inside the ROI and highlights the cell nuclei on the original image volume. Users can view the image volume with highlighted cell nuclei in 2D slices.

6.5.7 DINAVID Hardware Description

The hardware that we have for the current test version of the DINAVID is as follows:

Network-Based User Interface Server

- CPU: Intel Core i7-6900K



computer ☑

axes ☑

border ☑

density 10

brightnessR 1.14

brightnessG 1.32

brightnessB 1.29

gammaR 1.04

gammaG 1

gammaB 1

offsetR 0

offsetG 0

offsetB 0

mindensity 0.05

maxdensity 1

Fig. 6.4.: Example of a 3D Visualization

- RAM: 128GB
- GPU: NVIDIA GeForce GTX 1080 (1 GPU)
- Storage Driver: 1TB SSD + 10TB HDD

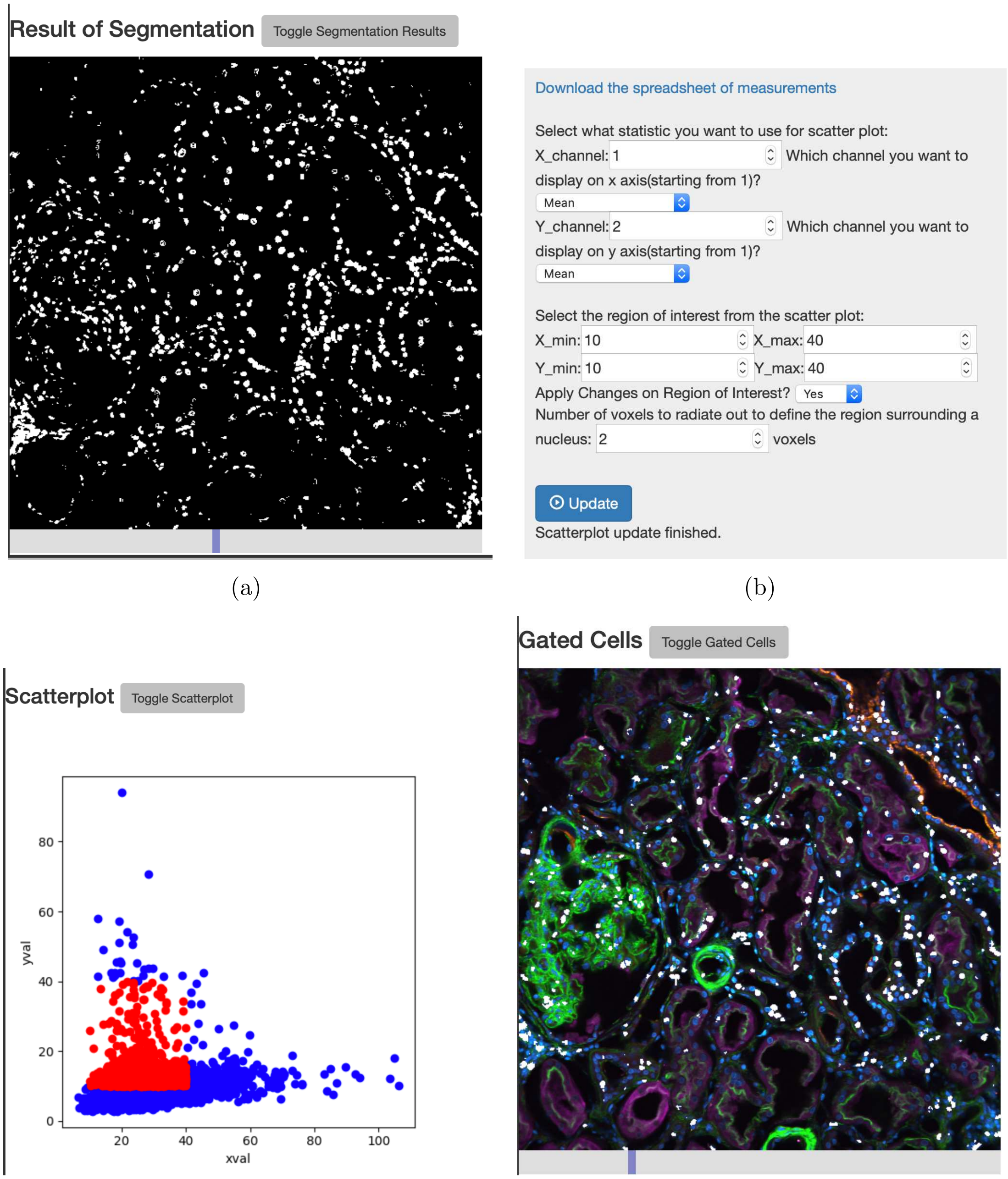
Compute/GPU Server

- CPU: Intel Core i7-6900K
- RAM: 128GB
- GPU: NVIDIA Titan XP 12GB RAM per GPU (4 GPUs)
- Storage Driver: 1TB SSD + 10TB HDD

We intend to deploy a larger version of DINAVID with multiple computational nodes in the near future. We will also deploy a version of DINAVID on the Amazon AWS system.

6.6 An Example of DINAVID Use Case

Users interact with DINAVID with a web browser. Users are required to log in using the credentials that are supplied to them on request. The login credentials are



Download the spreadsheet of measurements

Select what statistic you want to use for scatter plot:

X_channel: 1 Which channel you want to display on x axis(starting from 1)?

Mean

Y_channel: 2 Which channel you want to display on y axis(starting from 1)?

Mean

Select the region of interest from the scatter plot:

X_min: 10 X_max: 40

Y_min: 10 Y_max: 40

Apply Changes on Region of Interest? Yes

Number of voxels to radiate out to define the region surrounding a nucleus: 2 voxels

Update

Scatterplot update finished.

(a) (b)

(c) (d)

Fig. 6.5.: Examples of quantitative image analysis with scatter-plot and plotting gated nuclei (a) Example segmentation using watershed [119], (b) Panel for quantitative scatter plot settings, (c) Scatter plot of nuclei, (d) Mapping of gated nuclei.

secured and managed by Django's Admin tools. A greeting page will be shown to the users with tabs containing project information and instructions to use DINAVID.

There are four main tabs for navigating through DINAVID, “Tools” for accessing the processing and visualization tools, “Tutorial” for instructions on using DINAVID, “Demo” for trying the system with a preloaded example image, and “Contact” for any questions and contact information. Users can upload an image volume for processing and visualization. Figure 6.3 shows an example interface for selecting the parameters for 2D visualization. After choosing the desired channel and corresponding color to visualize in Figure 6.3 (a), the changes in brightness, gamma, and offset to the images will be reflected in Figure 6.3 (b). Figure 6.3 (c) shows the menu for adding processing steps to the user uploaded image. The number of steps and the sequence of processing is customizable. Figure 6.3 (d) shows an example slice of blurring operation on the nuclei channel of the sample image. Figure 6.4 shows an example interface for 3D rendered visualization with GUI for parameter tuning. The user can adjust the rendered volume and its appearance interactively in real-time. For segmentation with DeepSynth [118], five different segmentation models trained with different microscopy images are provided. An example of quantitative analysis after nuclei segmentation is shown in Figure 6.5. Using the segmentation result obtained from watershed [119], a scatter plot is generated by specifying the x-axis and y-axis statistics range. The gated nuclei are then overlaid on the original image.

Also, an image preview window shows the uploaded image. User can also process on a subvolume of the data by specifying a region of interest in the preview window. By pushing the blue “Process” button, our system will process the data at our computing clusters. Once our system finished the process, the web page will automatically be redirected to the result download page.

6.7 Building of the system

Biologists have been able to obtain large 3D microscopy data in recent developments in microscopy technology and computer system. This data is enormous in size and scope and provides many essential knowledge to understand biological processes.

A more efficient software and hardware for image processing, which can handle large and complex data quickly, is required. By designing the high-performance image processing framework for volumetric image analysis (DINAVID), It is hoped that machine learning and profound preparation would be brought into the hands of biologists who may not be specialists in areas of computer vision. The system supports large image volumes obtained by the 3D tissue cytometry center, storage, management, visualization, quantitative analysis, and exploration and supports remote investigators in real-time.

So far, we have designed and developed the prototype of a network-based microscopy image analysis system (DINAVID). This system is designed for fast and accurate analysis of large scale microscopy volumes. As shown in Figure 6.1, our system consists of network-based user interface and computing clusters that contains high performance GPUs. User will be able to upload and download data using network-based user interface. Also, built-in image previewer and built-in 3D volume visualization tools are integrated for visualizing data before and after processes.

The DINAVID system supports the function of image uploading and storage, image visualization, and segmentation of 3D microscopy image volumes. The distributed functionally of the system can support using multiple machines for deep learning based image volume processing and data transfer. The system has a embedded network-based visualization function. This system offers the capability to support client or server-side rendering options, to optimize performance for small versus large image volumes.

The DINAVID system has a user interface, a data and request handling mechanism, and an embedded network-based 3D data visualization tool.

- User Interface: The user interface of the website is written in JS, HTML and CSS. A Django framework is used to create the application. Django is python based web framework which allows quickly create web application. Django provides tools and components for the builder to add functions to the application.

The user data is managed by MySQL database server, which works with Django as well.

- Data and User Request Handling: The data and user request handling part of the website is written in Python, supported by Django as well. The language choice of Python allows us to integrate our deep learning functions which uses the PyTorch platform. To achieve distributed tasking and data transfer, Celery is used. Celery is an asynchronous task queue based on distributed message passing. Task queues are used as a strategy to distribute the workload between threads/machines. The messenger RabbitMQ is used to execute asynchronous in a Django application. RabbitMQ is required because Celery requires an external solution to send and receive messages. Those solutions are called message brokers. Once set up, the system is capable of managing and scheduling tasks from different web-users. The tasks are user data are securely sent to the GPU nodes machine to process. Once processed, the result data will be transfer back to the network node for download and display.

- 3D Visualization: The visualization tool is built with WebGL, which is a cross-platform web standard for 3D graphics API based on OpenGL ES. Volume rendering with ray casting is then implemented with WebGL for the ability to "see through" the uploaded 3D images data. We first combine the images to be visualized into one large mosaic file, which allows the data to be more accessible to the shader. The tool is built with the goal of fast and accurate rendering that is capable to work on large data. The visualization is real-time and interactive which also supports user adjusting parameters. Based on our testing, it can work on desktops, laptops, as well as on mobile devices.

6.8 Software Components Needed for the DINAVID System

List of required tools for building the system

(A) For the network user interface machine

1. Apache2

2. pip

3. mod-wsgi library (libapache2-mod-wsgi)

4. Virtualenv

5. Python 2.7 with libraries (numpy, spicy, scikit-image)

6. Django 1.11.11

7. PyTorch 0.3.1, torchvision, visdom

8. Celery 4.1.0 (No.5-8 above are installed under virtual environment installed at No.4)

9. RabbitMQ and Erlang

(B) For the computing GPU machine

1. pip

2. Python 2.7 with libraries (numpy, spicy, scikit-image)

3. Django 1.11.11

4. PyTorch 0.3.1, torchvision, visdom

5. Celery 4.1.0

6. RabbitMQ and Erlang

6.9 Setup Steps for the DINAVID System

- Django Setup and Database Setup In order to use Django for web development, one needs to consider the database driven design of the application. Although it is possible to use Django without a database, most Django applications use an object-relational mapper. The object-relational mapper can describe the layout of a database in Python code. For our case, several objects can be represented in the form of a "Model". For example, in order to make uploading a user image volume possible, the image volume is represented as a "Model" with several attributes, the location of the image file, the ownership of the image file, and the name of the image file and so on.

- Serving of the Django application with Apache Although Django is a powerful web framework that allows rapidly release your Python application or website, but a more secure and strong web server is necessary for anything that is even slightly related to production. Django also contains a simplified development server for local code testing. In order to install and configure Django in a virtual environment in Python, one needs to handle the client requests directly prior to passing requests which require an application logic to the Django app. The installation of Apache in our application is needed. It is achieved with mod wsgi Apache, which can interact with Django via the specification of the WSGI GUI. It is needed to configure Apache to interface with the Django app once we have the application running and up. The Apache module mod-wsgi is used to convert HTTP requests into a standard application format with a specification called WSGI.

- Django Architecture

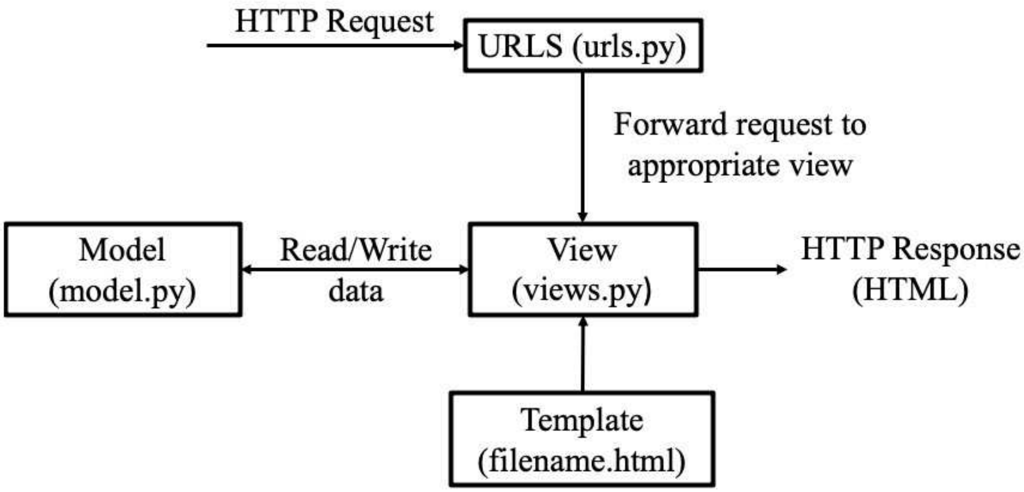


Fig. 6.6.: The architecture of Django [132]

Figure 6.6 shows the architecture of Django [132]. Once the user uploads volumes to DINAVID Django is used to process user requests and schedule and load data for processing on the computational and GPU nodes. The users send

HTTP requests to Django through their browser. For example, a request for displaying images will cause Django will forward the requests to the appropriate “views.” Each “view” is a Python script that defines how to process the request. During the processing of the requests, the views may interact with the “model” script for reading or writing data depending on what is required. For example, the “view” needs to know what images/volumes need to be retrieved from the system storage for the requested operation. Django will then return a response to the user’s browser, often by dynamically creating an HTML page and by inserting the retrieved data into placeholders in an HTML template.

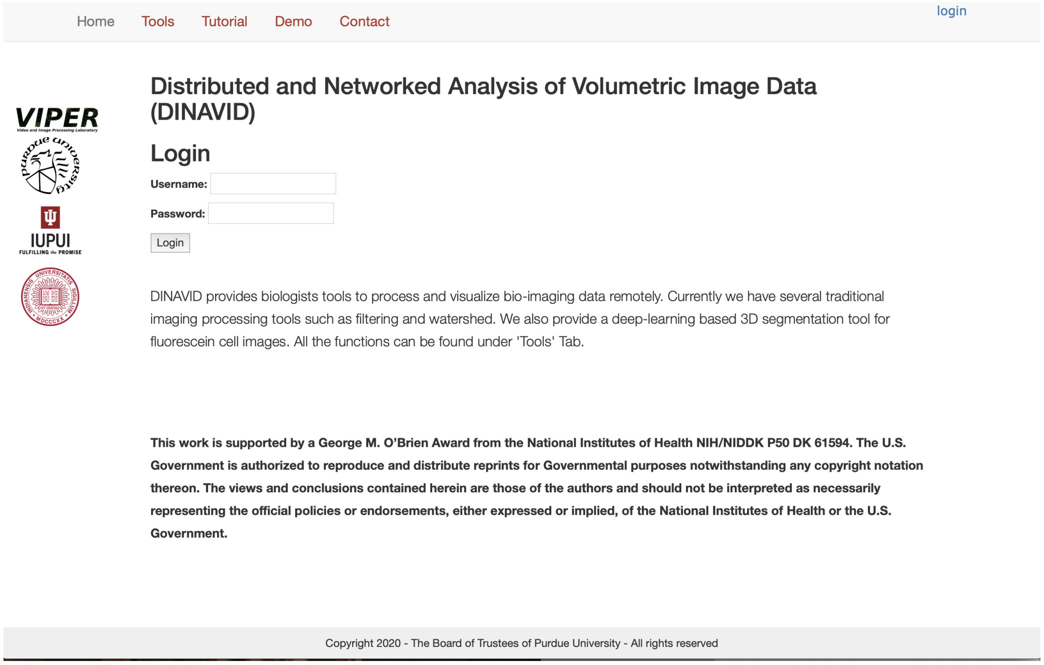
The DINAVID system employs Celery [134] to support task scheduling such as task queuing and distributed message passing. Task queues are used as a strategy to distribute the workload between multiple threads or computational nodes. Tasks can execute asynchronously (in the background) or synchronously (wait until ready). The DINAVID system supports parallel task processing for multiple users. This is achieved by RabbitMQ [135], a message broker, to accept and forward messages. The combination of RabbitMQ and Celery can achieve the function of managing and scheduling tasks from different users. It can automatically distribute users’ data to different computational nodes for processing. The resultant information will be stored in the system storage. The data will be saved for the user to access when needed.

6.10 Extensibility of the System

DINAVID is a system that can be extended with additional functionality built with Python. It is also possible to extend the system with web-based applications. One of the common use cases for extending the system is to add a user’s demand of ”pre-processing” techniques to the system. There are three parts of altering that is needed. Firstly, modify the webpage by adding a new JavaScript function. Then, add the processing step as a new operation with a new operation ID. Secondly, in the corresponding ”Views”, add the function you want to build into the system. Thirdly, for the function to work with Celery, modify the file that defines the task to be executed in the Celery task files. In the same way, a new segmentation method can be added.

6.11 Using the DINAVID System

Figure 6.7 shows the login pages. User will need to login to our system using our issued credential. The login credentials are secured and managed by the Django Admin tools. Django Admin tools are supported by Django with a user friendly administration interface. The metadata from the models can allow trusted users to mange and control the website.



Distributed and Networked Analysis of Volumetric Image Data (DINAVID)

Login

Username:

Password:

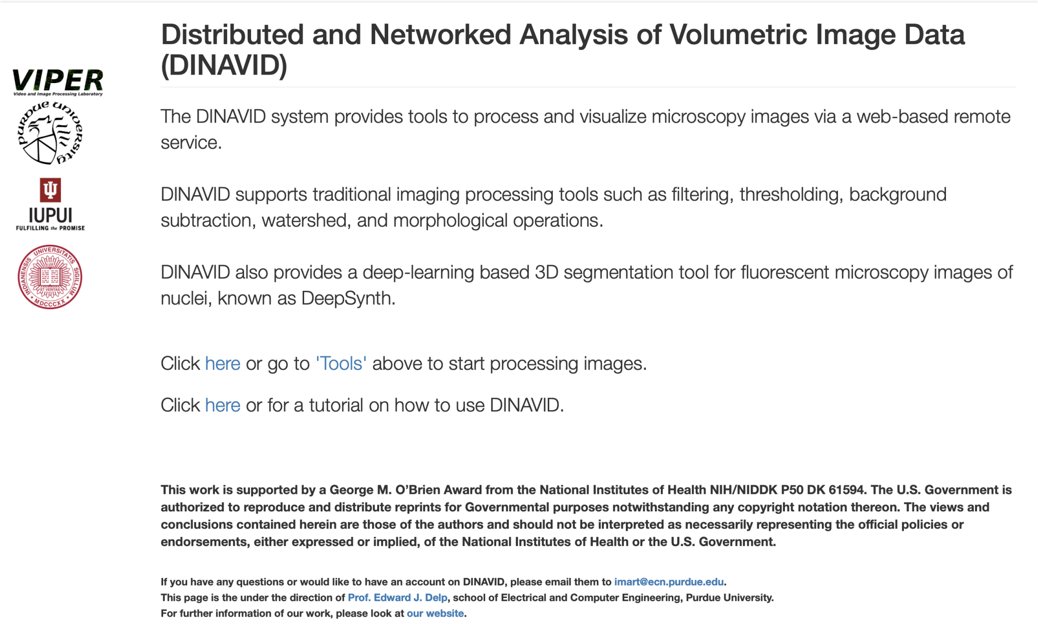
DINAVID provides biologists tools to process and visualize bio-imaging data remotely. Currently we have several traditional imaging processing tools such as filtering and watershed. We also provide a deep-learning based 3D segmentation tool for fluorescein cell images. All the functions can be found under 'Tools' Tab.

This work is supported by a George M. O'Brien Award from the National Institutes of Health NIH/NIDDK P50 DK 61594. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Institutes of Health or the U.S. Government.

Copyright 2020 - The Board of Trustees of Purdue University - All rights reserved

Fig. 6.7.: Login page of DINAVID

Figure 6.8 shows the welcome page, which has our project information and a brief tutorial. In the “Tool” tap, the upload function will allow user to uploaded they data into system using the blue “Upload Images” button.



Distributed and Networked Analysis of Volumetric Image Data (DINAVID)

The DINAVID system provides tools to process and visualize microscopy images via a web-based remote service.

DINAVID supports traditional imaging processing tools such as filtering, thresholding, background subtraction, watershed, and morphological operations.

DINAVID also provides a deep-learning based 3D segmentation tool for fluorescent microscopy images of nuclei, known as DeepSynth.

Click here or go to 'Tools' above to start processing images.

Click here or for a tutorial on how to use DINAVID.

This work is supported by a George M. O'Brien Award from the National Institutes of Health NIH/NIDDK P50 DK 61594. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Institutes of Health or the U.S. Government.

If you have any questions or would like to have an account on DINAVID, please email them to imart@ecn.purdue.edu.
This page is the under the direction of Prof. Edward J. Delp, school of Electrical and Computer Engineering, Purdue University.
For further information of our work, please look at our website.

Fig. 6.8.: Home page of DINAVID

After logging in, the user can see a list of uploaded image volumes. At the right top of the page, user can delete all the images using the red "Delete Uploaded Images" button. Multiple image formats are supported. Once the user has finished uploading images, by click the "visualize" button, a new window will be loaded and show the 3D view of the uploaded grayscale images.

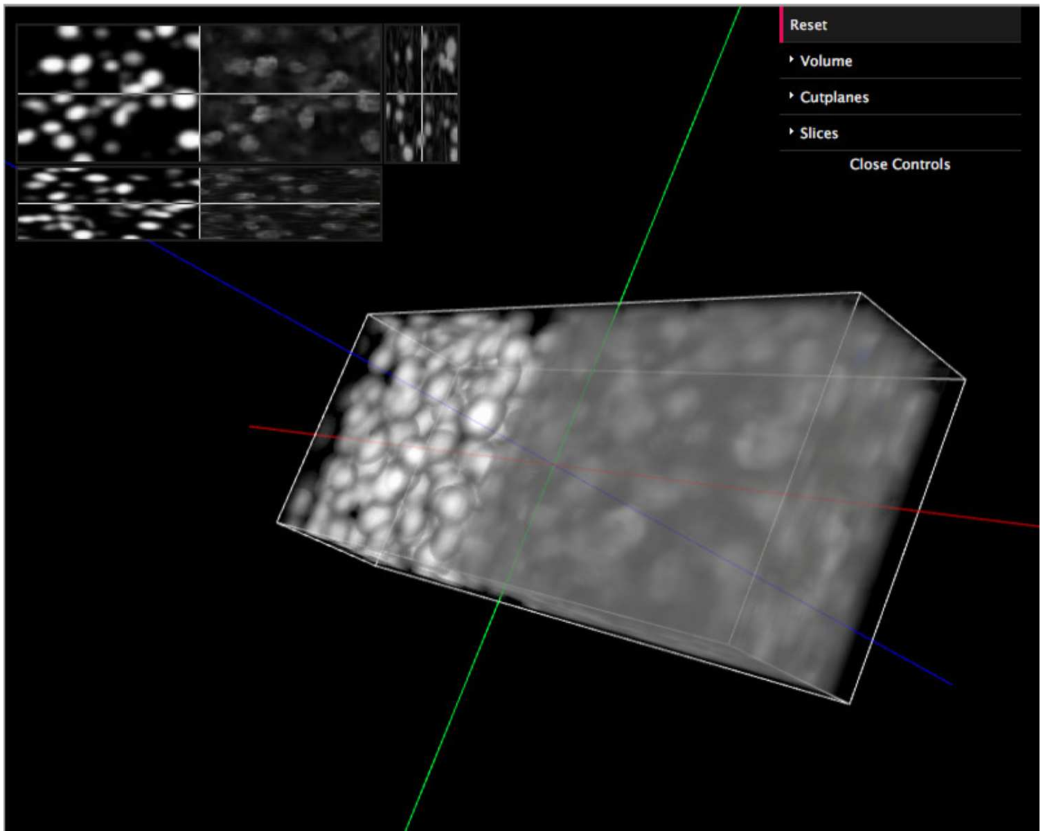
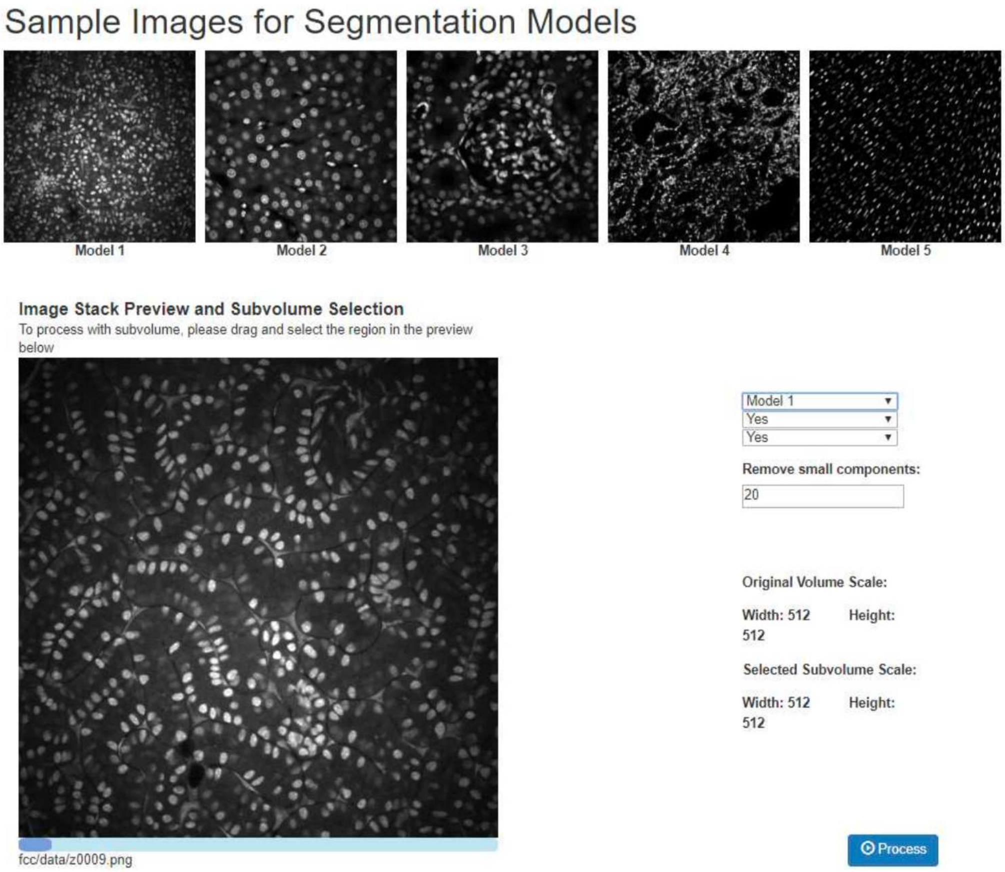


Fig. 6.9.: Visualize input image

A deep learning based nuclei segmentation method, DeepSynth, is implemented in our system. This is a 3D segmentation method is used to identify and segment nuclei in fluorescence microscopy volumes using machine-learning techniques, in particular deep learning. The method is unique in that the segmentation system only uses synthetic data for training. This method consists of two stages: 3D synthetic data generation and a 3D convolutional neural network (CNN) for 3D segmentation. More specifically, 3D synthetic data is generated using a spatially constrained cycle-



Sample Images for Segmentation Models

Image Stack Preview and Subvolume Selection

To process with subvolume, please drag and select the region in the preview below

fcc/data/z0009.png

Model 1 ▾

Yes ▾

Yes ▾

Remove small components:

20

Original Volume Scale:

Width: 512 Height: 512

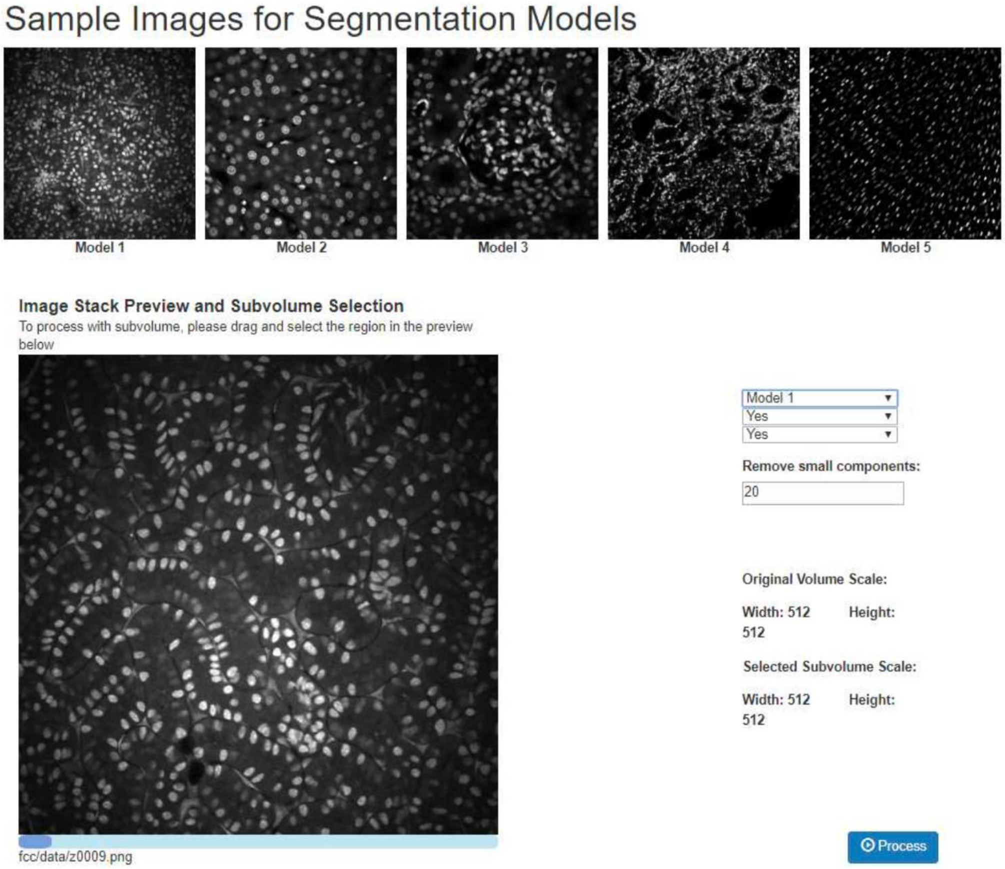
Selected Subvolume Scale:

Width: 512 Height: 512

Fig. 6.10.: Segmentation tool selection page

consistent adversarial network (SpCycleGAN) without the need of manual segmented groundtruth volumes. The synthetic data generated from the SpCycleGAN reflects characteristics of the original microscopy volumes such as background noise, nuclei shape, orientation and intensity. The synthetic data is used to train our 3D segmentation CNN. The 3D segmentation CNN is then used to segment nuclei structures in real microscopy volumes.

In Figure 6.11, five different segmentation models that trained with different microscopy images are provided. User can process their uploaded data with these models. Also, a image preview window shows the uploaded image.



Sample Images for Segmentation Models

Image Stack Preview and Subvolume Selection

To process with subvolume, please drag and select the region in the preview below

Model 1 ▾

Yes ▾

Yes ▾

Remove small components: 20

Original Volume Scale:

Width: 512 Height: 512

Selected Subvolume Scale:

Width: 512 Height: 512

fcc/data/z0009.png

Fig. 6.11.: Segmentation tool selection page

As shown in Figure 6.12, user can also process on a subvolume of the data by specifying a region of interest in the preview window. By pushing the blue "Process" button, our system will process the data at our computing clusters. Once our system finished the process, the web page will automatically redirected to the result download page.

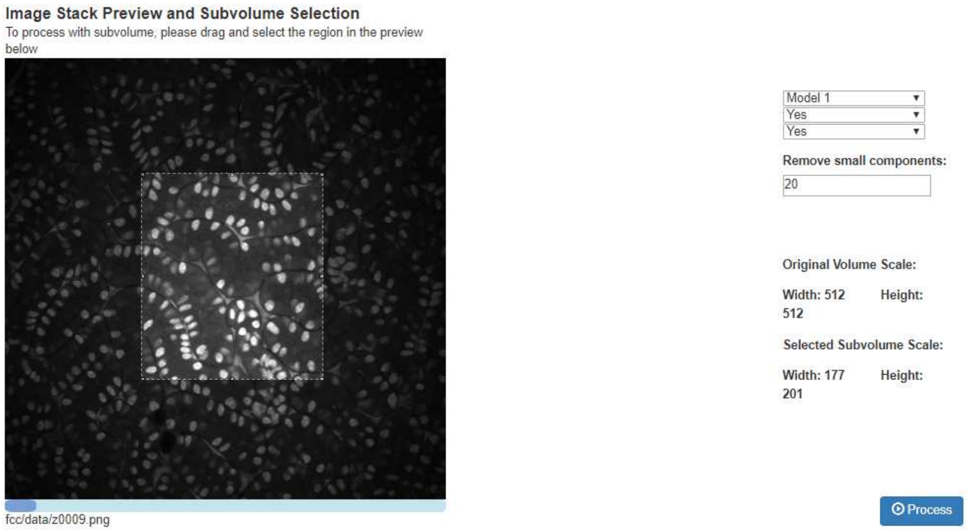


Image Stack Preview and Subvolume Selection

To process with subvolume, please drag and select the region in the preview below

fcc/data/z0009.png

Model 1

Yes

Yes

Remove small components: 20

Original Volume Scale:

Width: 512 Height: 512

Selected Subvolume Scale:

Width: 177 Height: 201

Fig. 6.12.: Subvolume selecting functionality

In Figure 6.13, our visualization tool can also provide subvolume visualization and 2D slices visualization. The visualization tool support visual settings of adjustable density, brightness, contrast, mindensity, and maxdensity. The default setting are as follows: Density: 10, Brightness: 0, Contrast: 1, Mindensity: 0.05, Maxdensity: 1.

With the advancement of high throughput optical microscopy very large 3D image datasets can be acquired. Image analysis tools and three dimensional visualization are critical for analyzing and interpreting these 3D image volumes. These large 3D microscopy volumes also require more computational resources than a biologist may have access to in typical desktop or laptop computers especially if machine learning tools are being used for image analysis. With this increased amount of data analysis and computational complexity, there is a need for a more accessible, easy-to-use, and efficient network-based/cloud-based 3D image processing system. The Distributed and Networked Analysis of Volumetric Image Data (DINAVID) system is being developed to enable remote analysis of microscopy images for biologists. DINAVID is designed using open source tools and has two main sub-systems, a computational

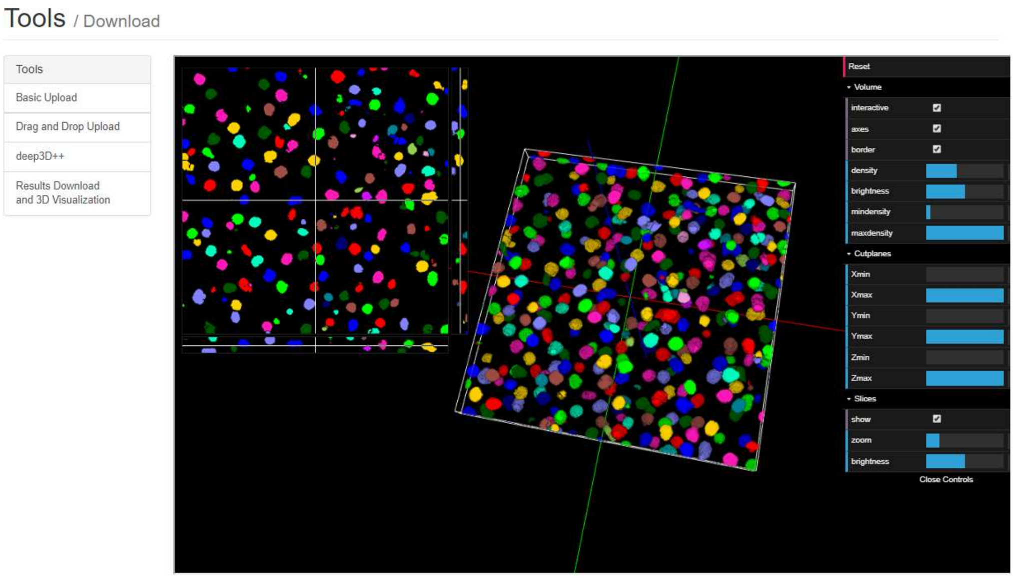


Fig. 6.13.: 3D visualization of DINAVID

system for 3D microscopy image processing and analysis and a 3D visualization system. The 3D visualization system enables real-time rendering of large 3D volumes of microscopy data. The computational system supports traditional imaging processing and analysis such as watershed [23], blurring, enhancement, and noise removal. The computational system also supports deep learning based tools for the detection and analysis of nuclei of 3D fluorescent microscopy images.

6.12 Conclusion

In this work, we described the DINAVID system that was developed for the analysis of 3D microscopy volumes. The goal is to provide a system capable of analyzing large 3D microscopy volumes using sophisticated machine learning methods that biologists can use without worrying about managing computational resources. We also compared DINAVID to existing systems. Implemented as a cloud-based system instead of a locally installable system, DINAVID removes the necessity for hardware and software installations beyond a computer with a less than $1000 GPU, 100Mb/s network connection, and a web browser. DINAVID provides real-time 2D focal plane and 3D volume rendering, along with a tissue cytometry tool with processing, segmentation, quantification, and data analysis steps. Each component of the tissue cytometry workflow is independent with one another, enabling the capability to add new features or methods to each component.

DINAVID is designed using open source tools. The source code and user access to DINAVID are available upon request.

In the future we will deploy more image analysis tools including machine learning architectures for microscopy image analysis from our research group. One example that can be added in the future is the ability to provide a way for a user to upload training data into DINAVID so that the machine learning models can be retrained. We also plan to add features to DINAVID to process more types of microscopy image data, such as supporting additional bio-imaging formats beyond 3D composite TIFFs or collections of 2D slices. We are also investigating adding online transfer learning tools that will allow users to investigate their own types of data by training machine learning methods.

7. CONCLUSIONS

7.1 Summary and Future Work

7.1.1 Summary of the contribution of the thesis

The main contributions of this thesis are:

- Microscopy Nuclei Counting and Segmentation: We have introduced a nuclei counting method using synthetic microscopy images generated with the previous mentioned SpCycleGAN. A 3D GAN counting network is used to estimate the number of nuclei in a microscopy image volume. The 3D counting network is evaluated with both synthetic and real microscopy data. Since the above mentioned nuclei counting method generates a segmentation mask before the post-processing counting is performed, the segmentation results are retained and used to find the object-based $F1$ score of the nuclei finding accuracy.

- Microscopy Nuclei Segmentation and Classification: In this thesis, I also described a method for nuclei segmentation and detection with boundary refinement. Three main stages of the method were described in the following order. The first step is a synthetic image generation step using GAN. The generation of realistic synthetic data is extended from ReCycleGAN [24] which is initially developed for video to video style transfer. ReCycleGAN extended CycleGAN into three dimensions with the added temporal dimension. Instead of using the temporal dimension, we used one additional dimension in the spatial dimension in the 3D image volume. This approach can generate synthetic images that are continuous along the z-axis, which contains more complete information of a 3D image volume. Following the synthetic image generation is the segmentation step with boundary refinement. Using the generated segmentation mask and shape indicator volumes, the method is able to predict the ellipsoidity of the nuclei without any prerequisite labeling of nuclei types.

- Distributed and Networked Analysis of Volumetric Image Data (DINAVID) System: We developed the Distributed and Networked Analysis of Volumetric Image Data (DINAVID) system with the purpose of providing biologists with a tool for convenient analyzing of large volumes of microscopy data. The system is designed according to the standard workflow and habits of microscopy image analysis guided by biologists' perspective. Two main components of system were described in this thesis, a 3D visualization tool that is capable of displaying original and colored images volume in 3D with interactive visualization, and a remote computing tool that provides multiple types of image processing and analysis functions including machine learning based nuclei segmentation. The 3D visualization enables real-time rendering of large volumes of microscopy data. The segmentation tool provides fast inferencing of pre-trained deep learning models trained with 5 different types of microscopy data.

7.1.2 Future Work

The proposed future work from this thesis is listed here:

- Microscopy Nuclei Image Segmentation and Counting: The previously mentioned SpCycleGAN is able to generate nuclei images without using any manually labeled data. The generated 2D images can be stacked to form 3D volume. Although the characteristic of nuclei is realistic in 2D, shape of the structures are not perfectly defined in 3D. In the future, we would like to expand our current method to a 3D technique. Also, our current method can be used on other applications such as image denoising, super resolution, and image restoration. We plan to investigate how to evaluate the quality of synthetically generated microscopy data in terms of noise, blur, and homogeneity. In the future, we would like to explore more on how to generalize our techniques. A unified model can provide a pre-trained models for the user to fine-tune, which is a useful expansion of application for generalized nuclei segmentation. In the future, we plan to test across different types of segmentation network and work on 3D ways to generate synthetic microscopy volumes. The 3D GAN counter can be used for the analysis of very crowd nuclei images, while still producing an accurate counting result. Since biologists are usually more interested in quantifying the number of nuclei, we want to focus the method on more efficient ways of producing the counting measurement. Also, nuclei that locate around the boundaries of the microscopy volumes are hard to detect, we plan to improve the detection accuracy around the boundaries with majority voting and refinement techniques.

- Microscopy Nuclei Segmentation and Boundary Refinement: In this work, a method for nuclei segmentation and detection with boundary refinement is also described. This method consists of three stages, a synthetic image volume generation step, a segmentation step, and a boundary refinement step. The generation of realistic synthetic data is extended from ReCycleGAN [24] which is initially developed for video to video style transfer. In the future, we hope to use this method to generate large sizes of synthetic data that is also meaningful and realistic in three dimensions. The plan for this work includes the evaluation of the microscopy images generated synthetically with a comparison with images generated by our SpCycleGAN and CycleGAN. Since different shapes of nuclei co-exist in a single volume, during the generation of synthetic images, we have manually crafted the ratio of round shaped and elongated nuclei. In the future, we plan to have machine learning based method for the analysis of the distribution ratio of these different types of nuclei. In this way, it is helpful for generating more realistic synthetic images that are more close in the characteristic and distribution of nuclei.

- In this work, we have proposed a Distributed and Networked Analysis of Volumetric Image Data (DINAVID) system is developed to enable remote analysis of microscopy images for biologists. There are two main functions integrated in the system, a 3D visualization tool and a remote computing tool for nuclei segmentation. The 3D visualization enables real-time rendering of large volumes of microscopy data. The segmentation tool provides fast inferencing of pre-trained deep learning models trained with 5 different types of microscopy data. In the future, we hope to process more types of microscopy image data, for cases when the users are prepared to provide training data. We hope to add online transfer learning functions that can allow users to try their own types of data and refine the results from the training of our provided segmentation network. Users can also preprocess their data with simple image processing tools, like blurring, background removal, histogram equalization, and resizing or scaling of the image volumes. The segmentation processing tools can have more options, for example, asking the user to resize their nuclei images to match the nuclei sizes between user data and pre-trained data. The future of this work including expanding the system to a larger scale with around 50 users with biological backgrounds. The users can test and share their results in real-time with co-workers and select sub-regions of their interest to follow up. The system will also have higher security features, that can preserve and protect user data while encouraging sharing and collaboration. We are also investigating the ability for remotely visualize 3D large volumes of microscopy data. The current 3D interactive visualization system is locally rendered which means the ability of rendering speed and size is dependent on the user's side. By utilizing server side GPU and send information to clients, we can lower the burden from client side. One potential issue is the remote sharing rendering will have a performance that is largely depended on the network speed. So we will look into methods for compression and speedy transferring of real-time rendered 3D image volumes.

7.2 Publication Result From This Work

Journal Papers

1. K. W. Dunn, C. Fu, D. J. Ho, S. Lee, **S. Han**, P. Salama, and E. J. Delp, "DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data," *Scientific Reports*, Volume 9, Article number: 18295, December 2019. DOI: 10.1038/s41598-019-54244-5

Journal Papers In Preparation

1. E. J. Delp, **S. Han**, A. Chen, S. Lee, C. Fu, C. Yang, L. Wu, P. Salama, K. W. Dunn, and P. Salama, "DINAVID: A High-Performance Distributed and Networked Image Analysis System for Volumetric Image Data," *To be submitted*

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.*

3. S. Lee, **S. Han**, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Three dimensional tubule synthesis and segmentation for fluorescence microscopy using generative adversarial networks," *To be submitted to the IEEE Transactions on Medical Imaging.*

Conference Papers

1. **S. Han**, S. Lee, A. Chen, C. Yang, P. Salama, K. W. Dunn, and E. J. Delp, "Three Dimensional Nuclei Segmentation and Classification of Fluorescence Microscopy Images," *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, April 2020, Iowa City, Iowa.

2. **S. Han**, S. Lee, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Nuclei counting in microscopy images with three dimensional generative adversarial networks," *Proceedings of the SPIE Conference on Medical Imaging*, pp. 109492Y-1-11, February 2019, San Diego, CA.

3. 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,” *Proceedings of the IEEE International Conference on Biomedical and Health Informatics (BHI)*, May 2019, Chicago, IL.

4. S. Lee, **S. Han**, P. Salama, K. W. Dunn, and E. J. Delp, “Three Dimensional Blind Image Deconvolution for Fluorescence Microscopy Using Generative Adversarial Networks,” *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, April 2019, Venice, Italy.

5. 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*, June 2018, Salt Lake City, UT.

6. C. Fu, D. J. Ho, **S. Han**, P. Salama, K. W. Dunn, 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.

REFERENCES

REFERENCES

[1] D. B. Murphy and M. W. Davidson, *Fundamentals of light microscopy and electronic imaging*, 2nd ed. Hoboken, NJ: Wiley-Blackwell, 2012. [Online]. Available: https://doi.org/10.1002/9781118382905

[2] A. Koehler, “New method of illumination for phomicrographical purposes,” *Journal of the Royal Microscopical Society*, vol. 14, pp. 261–262, 1894.

[3] J. W. Lichtman and J.-A. Conchello, “Fluorescence microscopy,” *Nature Methods*, vol. 2, no. 12, pp. 910–919, December 2005. [Online]. Available: http://doi.org/10.1038/nmeth817

[4] 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

[5] 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/10.1117/1.OE.58.4.046112

[6] 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

[7] 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

[8] 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

[9] 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

[10] S. G. Clendenon, P. A. Young, M. Ferkowicz, C. Phillips, and K. W. Dunn, “Deep tissue fluorescent imaging in scattering specimens using confocal microscopy,” *Microscopy and Microanalysis*, vol. 17, no. 4, pp. 614–617, August 2011. [Online]. Available: https://doi.org/10.1017/S1431927611000535

[11] W. Denk and K. Svoboda, “Photon upmanship: Why multiphoton imaging is more than a gimmick,” *Neuron*, vol. 18, no. 3, pp. 351–357, 1997. [Online]. Available: https://doi.org/10.1016/S0896-6273(00)81237-4

[12] 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.2012.361

[13] W. R. Zipfel, R. M. Williams, and W. W. Webb, "Nonlinear magic: Multiphoton microscopy in the biosciences," *Nature Biotechnology*, vol. 21, no. 11, pp. 1369–1377, October 2003. [Online]. Available: https://doi.org/10.1038/nbt899

[14] J. Mertz, "Nonlinear microscopy: New techniques and applications," *Current Opinion in Neurobiology*, vol. 14, no. 5, pp. 610–616, October 2004. [Online]. Available: https://doi.org/10.1016/j.conb.2004.08.013

[15] 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

[16] U. Kubitscheck, *Fluorescence Microscopy: From Principles to Biological Applications*. Wiley-Blackwell, 2013. [Online]. Available: http://doi.org/10.1002/9783527687732

[17] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-B. Sibarita, and J. Salamero, "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: http://doi.org/10.1109/TMI.2009.2033991

[18] 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

[19] 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: http://doi.org/10.1109/tmi.2016.2527740

[20] S. Lee, P. Salama, K. W. Dunn, and E. J. Delp, "Boundary fitting based segmentation of fluorescence microscopy images," *Proceedings of the IS&T/SPIE International Symposium on Electronic Imaging*, pp. 940 805–1–10, February 2015, San Francisco, CA. [Online]. Available: https://doi.org/10.1117/12.2085417

[21] R. Arppe, M. R. Carro-Temboury, C. Hempel, T. Vosch, and T. J. Sørensen, "Investigating dye performance and crosstalk in fluorescence enabled bioimaging using a model system," *PloS one*, vol. 12, no. 11, pp. e0 188 359–e0 188 359. [Online]. Available: https://doi.org/10.1371/journal.pone.0188359

[22] 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

[23] L. Vincent and P. Soille, “Watershed 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

[24] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, “Recycle-gan: Unsupervised video retargeting,” *Proceedings of the European Conference on Computer Vision*, pp. 122–138, September 2018, Munich, Germany. [Online]. Available: https://doi.org/10.1007/978-3-030-01228-1_8

[25] 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

[26] 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

[27] 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

[28] 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 Medical Imaging*, vol. 29, no. 3, pp. 852–867, March 2010. [Online]. Available: https://doi.org/10.1109/TMI.2009.2038693

[29] 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

[30] 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

[31] 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 Squassh,” *Nature Protocol*, vol. 9, no. 3, pp. 586–596, March 2014. [Online]. Available: https://doi.org/10.1038/nprot.2014.037

[32] 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

[33] N. 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

[34] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, no. 1, pp. 113–125, July 1994. [Online]. Available: https://doi.org/10.1016/0165-1684(94)90060-4

[35] 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*, vol. 53, no. 11, pp. 2405–2414, November 2006. [Online]. Available: https://doi.org/10.1109/TCSI.2006.884469

[36] O. Cicek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, vol. 9901, pp. 424–432, October 2016, Athens, Greece. [Online]. Available: https://doi.org/10.1007/978-3-319-46723-8-49

[37] Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V. Mok, L. Shi, and P.-A. Heng, "Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1182–1195, May 2016. [Online]. Available: https://doi.org/10.1109/TMI.2016.2528129

[38] H. Chen, Q. Dou, L. Yu, J. Qin, and P. A. Heng, "VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images," *Neuroimage*, vol. 170, no. 1, pp. 446–455, April 2018. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2017.04.041

[39] 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

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of the Advances in 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

[42] 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

[43] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Proceedings of the Neural Information Processing Systems*, pp. 1–9, December 2012, Lake Tahoe, NV.

[44] 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

[45] 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

[46] 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, January 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2644615

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the International Conference on Machine Learning*, vol. 37, pp. 448–456, July 2015, Lille, France. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, vol. 9351, pp. 234–241, October 2015, Munich, Germany. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28

[49] F. Xing, Y. Xie, and L. Yang, "An automatic learning-based framework for robust nucleus segmentation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 2, pp. 550–566, February 2016. [Online]. Available: http://doi.org/10.1109/TMI.2015.2481436

[50] 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

[51] 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

[52] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, June 2016, Las Vegas, NV. [Online]. Available: https://doi.org/10.1109/CVPR.2016.265

[53] 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

[54] 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: https://papers.nips.cc/paper/5423-generative-adversarial-nets

[55] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *International Conference on Learning Representations*, pp. 1–16, May 2016. [Online]. Available: https://arxiv.org/abs/1511.06434

[56] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, pp. 1–32, December 2017. [Online]. Available: https://arxiv.org/abs/1701.07875

[57] 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

[58] 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. 2242–2251, October 2017, Venice, Italy. [Online]. Available: https://doi.org/10.1109/ICCV.2017.244

[59] 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 IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pp. 2302–2310, June 2018, Salt Lake City, UT. [Online]. Available: https://doi.org/10.1109/CVPRW.2018.00298

[60] C. McQuin, A. Goodman, V. Chernyshev, L. Kamentsky, B. A. Cimini, K. W. Karhohs, M. Doan, L. Ding, S. M. Rafelski, D. Thirstrup, W. Wiegraebe, 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. 1–17, July 2018. [Online]. Available: https://doi.org/10.1371/journal.pbio.2005970

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

[62] B. Schmid, J. Schindelin, A. Cardona, M. Longair, and M. Heisenberg, "A high-level 3D visualization API for Java and ImageJ," *BMC Bioinformatics*, vol. 11, no. 274, pp. 1–7, May 2010. [Online]. Available: https://doi.org/10.1186/1471-2105-11-274

[63] 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

[64] K. U. Barthel, “3D-data representation with ImageJ,” *ImageJ User and Developer Conference*, pp. 1–4, May 2006.

[65] R. S. MacLeod, J. G. Stinstra, S. Lew, R. T. Whitaker, D. J. Swenson, M. J. Cole, J. Krüger, D. H. Brooks, and C. R. Johnson, “Subject-specific, multiscale simulation of electrophysiology: a software pipeline for image-based models and application examples,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1896, pp. 2293–2310, 2009. [Online]. Available: https://doi.org/10.1098/rsta.2008.0314

[66] D. Shreiner, G. Sellers, J. M. Kessenich, and B. M. Licea-Kane, *OpenGL programming guide: The official guide to learning OpenGL, Version 4.3*, 8th ed. Upper Saddle River, NJ: Addison-Wesley Professional, 2013.

[67] S. Bhattiprolu, “How to ensure fast, accurate and repeatable cell counting,” September 2019. [Online]. Available: https://www.apeer.com/blog-categories/how-tos

[68] P. Bajcsy, A. Vandecreme, J. Amelot, J. Chalfoun, M. Majurski, and M. Brady, “Enabling stem cell characterization from large microscopy images,” *Computer*, vol. 49, no. 7, pp. 70–79, 2016. [Online]. Available: https://doi.org/10.1109/MC.2016.191

[69] P. Bajcsy and N. Hotaling, “Interoperability of web computational plugins for large microscopy image analyses,” *National Institute of Standards and Technology Interagency or Internal Reports*, pp. 1–27, March 2020. [Online]. Available: https://doi.org/10.6028/NIST.IR.8297

[70] J. B. Pettit and J. C. Marioni, “bioweb3d: an online webgl 3d data visualisation tool,” *BMC bioinformatics*, vol. 1, no. 185, pp. 1–7, 2013. [Online]. Available: https://doi.org/10.1186/1471-2105-14-185

[71] A. I. for Brain Science, “Allen cell explorer: data portal provides an unprecedented view into human stem cells,” *Phys.org*, p. April, 2017. [Online]. Available: https://phys.org/news/2017-04-allen-cell-explorer-portal-unprecedented.html

[72] D. Haehn, J. Hoffer, B. Matejek, A. Suissa-Peleg, A. K. Al-Awami, L. Kamentsky, F. Gonda, E. Meng, W. Zhang, and R. Schalek, “Scalable interactive visualization for connectomics,” vol. 4, no. 3, p. 29, 2017. [Online]. Available: https://doi.org/10.3390/informatics4030029

[73] K. W. Dunn and P. A. Young, “Principles of multiphoton microscopy,” *Nephron Experimental Nephrology*, vol. 103, no. 2, pp. e33–e40, March 2006. [Online]. Available: http://doi.org/10.1159/000090614

[74] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1324–1332, December 2010, Whistler, Canada.

[75] Y. Xue, N. Ray, J. Hugh, and G. Bigras, “Cell counting by regression using convolutional neural network,” *Proceedings of the European Conference on Computer Vision*, pp. 274–290, October 2016, Amsterdam, Netherlands. [Online]. Available: https://doi.org/10.1007/978-3-319-46604-0-20

[76] 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. 1, p. 529, 2017. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw681

[77] S. Bolte and F. P. Cordelieres, "A guided tour into subcellular colocalization analysis in light microscopy," *Journal of microscopy*, vol. 224, no. 3, pp. 213–232, December 2006. [Online]. Available: https://doi.org/10.1111/j.1365-2818.2006.01706.x

[78] H. Tulsani, S. Saxena, and N. Yadav, "Segmentation using morphological watershed transformation for counting blood cells," *International Journal of Computer Applications and Information Technology*, vol. 2, no. 3, pp. 28–36, January 2013.

[79] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam, "A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks," *Cytometry Part A*, vol. 56, no. 1, pp. 23–36, November 2003. [Online]. Available: http://doi.org/10.1002/cyto.a.10079

[80] M. S. Rahman and M. R. Islam, "Counting objects in an image by marker controlled watershed segmentation and thresholding," *Proceedings of International Advance Computing Conference*, pp. 1251–1256, May 2013, Ghaziabad, India. [Online]. Available: http://doi.org/10.1109/IAdCC.2013.6514407

[81] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1, 2, pp. 187–228, 2000.

[82] A. E. Allaoui and M. Nasri, "Medical image segmentation by marker-controlled watershed and mathematical morphology," *The International Journal of Multimedia & Its Applications*, vol. 4, no. 3, p. 1, 2012. [Online]. Available: https://doi.org/10.5121/ijma.2012.4301

[83] E. Walach and L. Wolf, "Learning to count with CNN boosting," *Proceedings of the European Conference on Computer Vision*, pp. 660–676, October 2016, Amsterdam, Netherlands. [Online]. Available: http://doi.org/10.1007/978-3-319-46475-6_41

[84] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 833–841, June 2015, Boston, MA. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298684

[85] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting with fully convolutional regression networks," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 283–292, 2018. [Online]. Available: https://doi.org/10.1080/21681163.2016.1149104

[86] 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

[87] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam, "Learning classifiers from synthetic data using a multichannel autoencoder," *arXiv preprint arXiv:1503.03163*, pp. 1–11, March 2015. [Online]. Available: https://doi.org/10.13140/RG.2.1.1700.3367

[88] 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 IEEE Conferences on Computer Vision and Pattern Recognition Workshop*, pp. 834–842, July 2017, Honolulu, HI. [Online]. Available: https://doi.org/10.1109/CVPRW.2017.116

[89] Z. Zhang, L. Yang, and Y. Zheng, "Translating and segmenting multimodal medical volumes with cycle-and shapeconsistency generative adversarial network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9242–9251, June 2018, Salt Lake City, UT. [Online]. Available: https://doi.org/10.1109/CVPR.2018.00963

[90] C. Fu, "Microscopy image registration, synthesis and segmentation," Ph.D. dissertation, Purdue University, May 2019, West Lafayette, IN. [Online]. Available: https://hammer.figshare.com/articles/thesis/microscopy_image_registration_synthesis_and_segmentation/7754981

[91] 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

[92] S. Lee, "Segmentation and deconvolution of fluorescence microscopy volumes," Ph.D. dissertation, Purdue University, August 2019, West Lafayette, IN. [Online]. Available: https://figshare.com/articles/Segmentation_and_Deconvolution_of_Fluorescence_Microscopy_Volumes/8194214

[93] D. J. Ho, "Three dimensional segmentation and detection of fluorescence microscopy images," Ph.D. dissertation, Purdue University, May 2019, West Lafayette, IN. [Online]. Available: https://hammer.figshare.com/articles/thesis/three_dimensional_segmentation_and_detection_of_fluorescence_microscopy_images/7757120

[94] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "Tensorflow: A system for large-scale machine learning," *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, vol. 16, pp. 265–283, November 2016, Savannah, GA. [Online]. Available: https://research.google/pubs/pub45381/

[95] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, pp. 1–15, January 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[96] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. [Online]. Available: https://doi.org/10.1109/TSMC.1979.4310076

[97] S. Lee, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Tubule segmentation of fluorescence microscopy images based on convolutional neural networks with inhomogeneity correction," *Proceedings of the IS&T International Symposium on Electronic Imaging*, vol. 2018, no. 15, pp. 199–1–199–8, January 2018, Burlingame, CA. [Online]. Available: https://doi.org/10.2352/ISSN.2470-1173.2018.15.COIMG-199

[98] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. 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

[99] 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://doi.org/10.9735/2229-3981

[100] E. Meijering, "Cell segmentation: 50 years down the road," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, September 2012. [Online]. Available: http://doi.org/10.1109/MSP.2012.2204190

[101] 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

[102] 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. 10, pp. R100–1–11, October 2006. [Online]. Available: https://doi.org/10.1186/gb-2006-7-10-r100

[103] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, no. 1, pp. 60–88, December 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005

[104] M. Z. Alom, C. Yakopcic, T. M. Taha, and V. K. Asari, "Microscopic nuclei classification, segmentation and detection with improved deep convolutional neural network (DCNN) approaches," *arXiv preprint arXiv:1811.03447*, pp. 1–18, November 2018.

[105] F. Milletari, N. Navab, and S. A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," *Proceedings of the IEEE 2016 Fourth International Conference on 3D Vision*, pp. 565–571, October 2016, Stanford, CA. [Online]. Available: https://doi.org/10.1109/3DV.2016.79

[106] J. Ribera, D. Guera, Y. Chen, and E. J. Delp, "Locating objects without bounding boxes," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6479–6489, June 2019, Long Beach, CA. [Online]. Available: http://doi.org/10.1109/CVPR.2019.00664

[107] D. Karimi and S. E. Salcudean, "Reducing the Hausdorff distance in medical image segmentation with convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 499–513, February 2020. [Online]. Available: https://doi.org/10.1109/TMI.2019.2930068

[108] 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, DC. [Online]. Available: https://doi.org/10.1109/ISBI.2018.8363790

[109] S. Han, S. Lee, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Nuclei counting in microscopy images with three dimensional generative adversarial networks," *Proceedings of the SPIE Conference on Medical Imaging*, pp. 109 492Y–1–11, February 2019, San Diego, CA. [Online]. Available: https://doi.org/10.1117/12.2512591

[110] S. Lee, S. Han, P. Salama, K. W. Dunn, and E. J. Delp, "Three dimensional blind image deconvolution for fluorescence microscopy using generative adversarial networks," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 538–542, April 2019, Venice, Italy. [Online]. Available: https://doi.org/10.1109/ISBI.2019.8759250

[111] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, November 1986. [Online]. Available: https://doi.org/10.1109/TPAMI.1986.4767851

[112] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," pp. 1510–1519, 2017. [Online]. Available: https://doi.org/10.1109/ICCV.2017.167

[113] S. Han, S. Lee, A. Chen, C. Yang, P. Salama, K. W. Dunn, and E. J. Delp, "Three dimensional nuclei segmentation and classification of fluorescence microscopy images," *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, April 2020, Iowa City, IA. [Online]. Available: https://doi.org/10.1109/ISBI45749.2020.9098560

[114] D. Coutu, K. Kokkaliaris, L. Kunz, and T. Schroeder, "Three-dimensional map of nonhematopoietic bone and bone-marrow cells and molecules," *Nature Biotechnology*, vol. 35, no. 12, p. 1202, Novenmber 2017. [Online]. Available: http://doi.org/10.1038/nbt.4006

[115] M. Gerner, W. Kastenmuller, I. Ifrim, J. Kabat, and R. Germain, "Histo-cytometry: a method for highly multiplex quantitative tissue imaging analysis applied to dendritic cell subset microanatomy in lymph nodes," *Immunity*, vol. 37, no. 2, pp. 364–376, August 2012. [Online]. Available: https://doi.org/10.1016/j.immuni.2012.07.011

[116] S. Winfree, S. Khan, R. Micanovic, M. T. Eadon, K. J. Kelly, T. A. Sutton, C. L. Phillips, K. W. Dunn, and T. M. El-Achkar, "Quantitative three-dimensional tissue cytometry to study kidney tissue and resident immune cells," *Journal of the American Society of Nephrology*, vol. 28, no. 7, pp. 2108–2118, 2017. [Online]. Available: https://doi.org/10.1681/ASN.2016091027

[117] 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

[118] K. W. Dunn, C. Fu, D. J. Ho, S. Lee, S. Han, P. Salama, and E. J. Delp, “DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data,” *Scientific reports*, vol. 9, no. 1, pp. 1–15, 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-54244-5

[119] S. Beucher and F. Meyer, *The Morphological Approach to Segmentation: The Watershed Transformation*, 01 1993, vol. Vol. 34, p. 433–481. [Online]. Available: http://doi.org/10.1201/9781482277234-12

[120] E. Iannuccelli, F. Mompart, J. Gellin, Y. Lahbib-Mansais, M. Yerle, and T. Boudier, “Nemo: a tool for analyzing gene and chromosome territory distributions from 3d-fish experiments,” *Bioinformatics*, vol. 26, no. 5, pp. 696–697, 2010. [Online]. Available: https://doi.org/10.1093/bioinformatics/btq013

[121] F. P. Cordelieres and S. Bolte, “JACoP v2. 0: improving the user experience with co-localization studies,” in *Proceedings of the 2nd ImageJ user and developer conference*, 2008, pp. 1–8.

[122] A. I. for Brain Science, “Pathtrace rendering of 3d cells.” [Online]. Available: https://www.allencell.org/pathtrace-rendering.html

[123] K. Kvilekval, D. Fedorov, B. Obara, A. Singh, and B. S. Manjunath, “Bisque: a platform for bioimage analysis and management,” *Bioinformatics*, vol. 26, no. 4, pp. 544–552, 2010. [Online]. Available: https://doi.org/10.093/bioinformatics/btp699

[124] T. Parisi, *WebGL: Up and running*, 1st ed. Sebastopol, CA: O’Reilly Media, Inc., 2012. [Online]. Available: http://shop.oreilly.com/product/0636920024729.do

[125] J. M. Noguera and J. R. Jiménez, “Visualization of very large 3D volumes on mobile devices and WebGL,” *International Conference on Computer Graphics, Visualization and Computer Vision*, pp. 105–112, June 2012, Pilsen, Czech Republic. [Online]. Available: https://gggj.ujaen.es/publications/details/62

[126] D. Cantor and B. Jones, *WebGL beginner’s guide*, 1st ed. Birmingham, UK: Packt Publishing, 2012. [Online]. Available: https://www.packtpub.com/game-development/webgl-beginners-guide

[127] T. Parisi, *Programming 3D applications with HTML5 and WebGL*, 1st ed. Sebastopol, CA: O’Reilly Media, 2014. [Online]. Available: http://shop.oreilly.com/product/0636920029205.do

[128] B. Laurie and P. Laurie, *Apache: The Definitive Guide*. Sebastopol, CA: O’Reilly Media, Inc., 2002. [Online]. Available: https://www.oreilly.com/library/view/apache-the-definitive/0596002033/

[129] D. Raggett, A. L. Hors, and I. Jacobs, "HTML 4.01 specification," *W3C recommendation*, vol. 24, 1999. [Online]. Available: https://www.w3.org/TR/html401/

[130] D. Flanagan, *JavaScript: the definitive guide, 7th Edition.* Sebastopol, CA: O'Reilly Media, Inc., 2020. [Online]. Available: https://www.oreilly.com/library/view/javascript-the-definitive/9781491952016/

[131] T. A. Powell, *HTML & CSS: the complete reference.* New York, NY: McGraw-Hill New York, 2010. [Online]. Available: https://www.oreilly.com/library/view/html-css/9780071496292/

[132] A. Holovaty and J. Kaplan-Moss, *The definitive guide to Django: Web development done right.* New York, NY: Apress, 2009. [Online]. Available: https://www.apress.com/gp/book/9781430219361

[133] B. J. Deen, J. M. Soderberg, V. C. Van, and H. L. Sanders, "Methods and systems for processing http requests," September 2003, U.S. Patent 6,629,127.

[134] A. Solem, "Celery: Distributed task queue," 2013. [Online]. Available: http://docs.celeryproject.org/en/latest/index.html

[135] D. Dossot, *RabbitMQ essentials.* Birmingham, UK: Packt Publishing Ltd, 2014. [Online]. Available: https://www.oreilly.com/library/view/rabbitmq-essentials/9781783983209/

[136] S. R. Sternberg, "Biomedical image processing," *Computer*, vol. 16, pp. 22–34, January 1983. [Online]. Available: https://www.semanticscholar.org/paper/Biomedical-Image-Processing-Sternberg/7b003db87ab0ad84009a5a62c9baa5f37334fc01

[137] M. Balatsko, "Rolling ball and sliding paraboloid background subtraction algorithms," *GitHub repository*, 2018. [Online]. Available: https://github.com/mbalatsko/opencv-rolling-ball

[138] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, Inc., 2008. [Online]. Available: https://www.oreilly.com/library/view/learning-opencv/9780596516130/

[139] S. Winfree, M. Ferkowicz, P. Dagher, K. Kelly, M. Eadon, T. Sutton, T. Markel, M. Yoder, K. Dunn, and T. El-Achkar, "Large-scale 3-dimensional quantitative imaging of tissues: State-of-the-art and translational implications," *Translational Research*, vol. 189, pp. 1–12, June 2017. [Online]. Available: https://doi.org/10.1016/j.trsl.2017.07.006

[140] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, and J. Bright, "SciPy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, pp. 1–12, 2020. [Online]. Available: http://doi.org/10.1038/s41592-019-0686-2

[141] S. V. der Walt, J. L. Schonberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "Scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014. [Online]. Available: https://doi.org/10.7717/peerj.453

[142] S. Trubetskoy, "List of 20 simple, distinct colors," Jan 2017. [Online]. Available: https://sashat.me/2017/01/11/list-of-20-simple-distinct-colors/.

[143] J. Hayward, *Django javascript integration: AJAX and jQuery.* Birmingham,UK: Packt Publishing Ltd, 2011.

[144] D. Vohl, D. G. Barnes, C. J. Fluke, G. Poudel, N. Georgiou-Karistianis, A. H. Hassan, Y. Benovitski, T. H. Wong, O. L. Kaluza, T. D. Nguyen, and C. P. Bonnington, "Large-scale comparative visualization of sets of multidimensional data," *PeerJ Computer Science*, vol. 2, no. 88, pp. 1–36, October 2016. [Online]. Available: https://doi.org/10.7717/peerj-cs.88

[145] J. Congote, A. Segura, L. Kabongo, A. Moreno, J. Posada, and O. Ruiz, "Interactive visualization of volumetric data with WebGL in real-time," *Proceedings of the ACM International Conference on 3D Web Technology*, pp. 137–146, June 2011, Paris, France. [Online]. Available: http://doi.acm.org/10.1145/2010425.2010449

[146] Y. Goltsev, N. Samusik, J. Kennedy-Darling, S. Bhate, M. Hale, G. Vasquez, and G. Nolan, "Deep profiling of mouse splenic architecture with codex multiplexed imaging," *Cell*, vol. 174, no. 4, pp. 968–981, August 2018. [Online]. Available: https://doi.org/10.1016/j.cell.2018.07.010

VITA

VITA

Shuo Han was born in Dalian, China. She received the B.S. degree in Biomedical Engineering from University of Michigan, Ann Arbor, and another B.S. degree in Electrical and Computer Engineering from Shanghai Jiao Tong University, Shanghai, in 2014.

Shuo joined the Ph.D. program at the School of Biomedical Engineering, Purdue University, West Lafayette, Indiana in 2014. She has worked as a Research Assistant and a Teaching Assistant in the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Edward J. Delp since January 2016. While in the graduate program, she worked on projects sponsored by the National Institutes of Health (NIH).

Shuo was a research intern at the Canon Medical Research Institute, Vernon Hills, IL, during the summer of 2017. She worked on computed tomography (CT) image quality improvement with the CT reconstruction group. She was also a research intern at the flat panel display team in Apple Inc., Cupertino, CA in the summer of 2019. Her research interests includes image processing, computer vision, deep learning, and medical imaging. She is a student member of the IEEE, the IEEE Signal Processing Society.