Novel Computational Algorithms for Imaging Biomarker Identification

by

Yanzhe Xu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2022 by the
Graduate Supervisory Committee:

Teresa Wu, Chair
Ashif Iquebal
Hao Yan
Scott Beeman

ARIZONA STATE UNIVERSITY

August 2022

ABSTRACT

Over the past few decades, medical imaging is becoming important in medicine for disease diagnosis, prognosis, treatment assessment and health monitoring. As medical imaging has progressed, imaging biomarkers are being rapidly developed for early diagnosis and staging of disease. Detecting and segmenting objects from images are often the first steps in quantitative measurement of these biomarkers. While large objects can often be automatically or semi-automatically delineated, segmenting small objects (blobs) is challenging. The small object of particular interest in this dissertation are glomeruli from kidney magnetic resonance (MR) images. This problem has its unique challenges. First of all, the size of glomeruli is extremely small and very similar with noises from images. Second, there are massive of glomeruli in kidney, e.g. over 1 million glomeruli in human kidney, and the intensity distribution is heterogenous. A third recognized issue is that a large portion of glomeruli are overlapping and touched in images.

The goal of this dissertation is to develop computational algorithms to identify and discover glomeruli related imaging biomarkers. The first phase is to develop a U-net joint with Hessian based Difference of Gaussians (UH-DoG) blob detector. Joining effort from deep learning alleviates the over-detection issue from Hessian analysis. Next, as extension of UH-DoG, a small blob detector using Bi-Threshold Constrained Adaptive Scales (BTCAS) is proposed. Deep learning is treated as prior of Difference of Gaussian (DoG) to improve its efficiency. By adopting BTCAS, under-segmentation issue of deep learning is addressed. The second phase is to develop a denoising convexity-consistent Blob Generative Adversarial Network (BlobGAN). BlobGAN could achieve high denoising performance and selectively denoise the image without affecting the blobs.

These detectors are validated on datasets of 2D fluorescent images, 3D synthetic images, 3D MR (18 mice, 3 humans) images and proved to be outperforming the competing detectors. In the last phase, a Fréchet Descriptors Distance based Coreset approach (FDD-Coreset) is proposed for accelerating BlobGAN's training. Experiments have shown that BlobGAN trained on FDD-Coreset not only significantly reduces the training time, but also achieves higher denoising performance and maintains approximate performance of blob identification compared with training on entire dataset.

DEDICATION

To my dear wife and parents, who have always been emotionally supportive.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisor, Dr. Teresa Wu, for her continuous support on my Ph.D. study, research and dissertation. Dr. Wu is a fabulous mentor and research leader with empathy, patience, motivation and enthusiasm. She taught me how to do independent and collaborative research. Her working habits, research attitude and life advice will profoundly and consistently influence me through my entire life to become a better person and researcher. I could not have imagined having a better advisor and mentor for this Ph.D. journey.

I would also like to specially thank my committee members: Dr. Ashif Iquebal, Dr. Hao Yan and Dr. Scott Beeman, for their great encouragement, valuable suggestions, insightful ideas and kind guidance during my dissertation. I feel very lucky to have such amazing researchers to join my committee.

My sincere thanks also go to my collaborators, Dr. Kevin M. Bennett and Dr. Jennifer R. Charlton, for providing me great opportunities in working on such impactful, interdisciplinary and exciting projects. I have really learned a lot from this collaboration.

Thank you to all past and present members of AMCII and my great colleagues: Dr. Jing Li, Dr. Giulia Pedrielli, Dr. Feng Ju, Fei Gao, Hyunsoo Yoon, Min Zhang, Yinlin Fu, Congzhe Su, Can Cui, Jiajing Huang, Fulin Cai, Teng Li, Jay Shah, You Zhou, Abhidnya Patharkar, Matt Liljenstolpe, Mahfuzur Rahman Siddiquee, Firas AI-Hindawi, Suryadipto Sarkar, Shuluo Ning, Xiaonan Liu, Na Zou, Bing Si, Nathan Gaw, Xinyu Zhao, Lujia Wang, Dorukhan Sergin, Ting Yan (Betrica) Fok, Yunyi Kang, Xiushuang Li, Yitao Chen, Feifan Wang, Zhiyang Zheng and Kun Wang, for your accompany, help and guidance.

I would like to extremely acknowledge my parents: Jiguang Xu and Fengming Wang, for giving birth to me, raising me, and supporting me unconditionally, spiritually and materially throughout my life.

Finally, and most importantly, I would like to thank my dear wife: Yumeng Xie, for her endless support, encouragement, and love.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

Over the past few decades, medical imaging is becoming important in medicine for disease diagnosis, prognosis, treatment assessment and health monitoring. Various medical imaging modalities including computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET) and digital mammography (DM) are invented and introduced into clinical applications. As medical imaging has progressed, imaging biomarkers are being rapidly developed for early diagnosis and staging of disease. The development of these biomarkers requires advances in both image acquisition and analysis. Detecting and segmenting objects from images are often the first steps in quantitative measurement of these biomarkers. In image analysis, detection is used to identify objects such as organs and tumors, and segmentation is used to isolate the objects from an image. While large objects can often be automatically or semi-automatically delineated, segmenting small objects (blobs) is challenging. Blobs can vary in size and location in images. Examples of blobs include cells or cell nuclei in images from optical microscopy (M. Zhang et al., 2015), exudative lesions in images of the retina (Sánchez et al., 2012), breast lesions in ultrasound images (Moon et al., 2013), and glomeruli in magnetic resonance (MR) images of the kidney (Beeman et al., 2011, 2014; Gao et al., 2016). In computer vision field, this is known as small blob detection and segmentation.

A number of blob detectors have been developed to tackle this challenging problem. These include interest point detectors such as SIFT (Lowe, 2004a), SURF (Bay et al., 2008),

Radial-Symmetry (Loy & Zelinsky, 2003), BRISK (Leutenegger et al., 2011) and FREAK (Alahi et al., 2012). This type of detectors can extract scale invariant features, unfortunately, studies have shown that the performance for these detectors in medical images was suboptimal (Kong et al., 2013a). Recently, scale-space based blob detectors are proposed and have shown great potential in medical images related application (Kong et al., 2013a). For example, Kong et al. proposed generalized Laplacian of Gaussian (gLoG) (Kong et al., 2013a), which demonstrated good performance on detection of the blob with various scales, shapes and orientations from pathological and fluorescent microscopic images. Zhang et al. developed Hessian based blob detectors, HLoG (M. Zhang et al., 2015), HDoG (M. Zhang et al., 2016), to automatically detect glomeruli in MR images with high accuracy and efficiency. However, the comparison experiments in (Xu et al., 2019) have shown these blob detectors are not robust to noises. This results over-detection issue, that is, some noises are detected as blob leading to the number of blobs higher than the truth.

With the growing interests from deep learning (DL), we recognize there is a gap in applying deep model to small blob detections. It is until most recently, DL models for blob like objects (e.g. nucleus, cell) detection and segmentation emerged. For example, Ciresan et al. (Ciresan et al., 2013) applied the CNN model to automatically detect cell from breast cancer histology images. Based on the cell centroid's probability provided from CNN model, a non-maxima suppression method was then used for the final cell identification. Khoshdeli et al. (Khoshdeli & Parvin, 2018) used a non-negative matrix factorization method and Laplacian of Gaussian (LoG) filter for blob initialization and improved the performance of nuclear detection from CNN model. It is noted these traditional CNN-based methods from literature are multi-staged pipelines, in which the object identification results

significantly depend on earlier stages. Fully convolutional network (FCN) (Shelhamer et al., 2017a) and its extension models, on the other hand, are getting increasing attention because the models are single stage providing end-to-end solutions. Basically, FCN models transfer all CNN model's fully connected layers to convolutional layers and provide dense class maps with arbitrary-sized input image. One example is from Xing et al.(Xing et al., 2019) where a FCN based architecture is designed for nucleus detection under a weak supervised predicted label. However, FCN requires large datasets for training limiting its potentials in medical applications where the sample sizes are often small. To alleviate this issue, U-Net (Ronneberger et al., 2015a), a modified version of FCN, was proposed and achieved quite success in various medical tasks with fast and accurate segmentation performance (Esser et al., 2018; Falk et al., 2019; Gao et al., 2019). The output from U-Net is a probability map where each pixel in 2D or voxel in 3D indicates the likelihood of the pixel or voxel being within the imaging object. This map may be used to aid the small blob detections. However, there are several drawbacks of training deep learning models on public datasets. First, public datasets are from different imaging modalities and therefore have a different distribution of intensities than the target dataset. Consequently, pre-trained models require additional ad-hoc approaches for fine-tuning the model parameters (Tajbakhsh et al., 2016). Second, small blobs created by structures such as cells, nuclei, and glomeruli have distinct geometric properties. If public and target datasets have different geometric properties, pre-trained models may not be accurate. Third, public datasets have different noise distributions compared to the target datasets, making it challenging to filter noise during segmentation. These limitations of training deep learning models hinder its application to optimally denoise blob images.

Recently, another class of deep learning models called Generative Adversarial Networks (GANs) (Janowczyk & Madabhushi, 2016a) has shown great potential for denoising images without the need for annotations. The methodology, known as CycleGAN, is based on the framework of image-to-image translation (J.-Y. Zhu et al., 2017). GAN consists of two sub-networks: a generator and a discriminator. A generator is trained to synthesize fake versions of the original images, and a discriminator is employed to distinguish the fake images from the real ones. The generator and discriminator networks train together iteratively, resulting in fake images resembling the real images. CycleGAN is a generalization of GAN to translate image characteristics from one domain to another by simultaneously training two pairs of generators and discriminators, one for each domain. By considering noisy and denoised images as the two domains, image denoising can be performed using CycleGAN. Gu et al. (Gu & Ye, 2021) proposed a tunable CycleGAN with adaptive instance normalization to denoise low-dose CT images. Zhou et al. (Zhou et al., 2020) presented a supervised CycleGAN to denoise low-dose PET images. However, CycleGAN has some limitations. Zhang et al. (Z. Zhang et al., 2018) have shown that CycleGAN may geometrically distort image features. Gu et al. (Gu et al., 2021) showed that geometric distortion can destroy anatomical details in denoised images. These local anatomical structures can be critical in clinical decision-making. To alleviate geometric distortions, Gu et al. (Gu et al., 2021) used a wavelet transform to extract high-frequency signals that contain the noise, supporting CycleGAN in denoising cardiac CT images. Guo et al. (Guo et al., 2019) introduced structure-aware loss to constrain CycleGAN in denoising images of retinal pathologies. These previous studies focused on denoising and preventing distortion of images containing large objects. However, this might not be

effective for denoising blob images consisting of small blobs and background noise, in which geometric distortion occurs because the blobs and noise have similar size, shape, and distributions of intensity. The limitation of these CycleGAN-based models motivates us to explore image denoising and propose a convexity consistent constraint to preserve the geometric properties of blobs. Given the location, shape, and intensity distribution of the blobs as prior, the adversarial model can learn the noise distribution and identify the blobs with minimal geometric distortion.

While the performance of CyleGAN-based models in these applications is promising, a common critique is the training is slow and often limited by computing resources. For example, the original CycleGAN (Anoosheh et al., 2018; J.-Y. Zhu et al., 2017) took 220 hours of training with NVIDIA Titan X GPU on 10,000 2D paintings images. One may argue that the training time can be reduced using advanced GPUs. However, such computing resource may not be readily available. To reduce the training time, researchers started exploring the training using a subset instead of the whole dataset. Using false positive rate to measure the performance of generated image quality, Nuha et al. (Nuha & Afiahayati, 2018) proposed DCGAN model and showed the training on subsets can lead to the false positive rate comparable to the training using the entire dataset with less computing time. Unfortunately, it was noted that the performance may be unstable - for some experiments - DCGAN led to very low false positive rate (Nuha & Afiahayati, 2018). To maintain the quality of GAN's generated image during subset selection, DeVries et al. (DeVries et al., 2020) proposed a novel instance selection approach based on manifold density of dataset. They removed the low-density regions to improve subsets' samples quality. Yet, this assumes that the low-density regions are noisy data region which may not

always be true for medical images. Additionally, the computational time of this approach trained on ImageNet was reduced from 14.8 days to 3.7 days only (DeVries et al., 2020), which still is considered to be computational expensive even with advanced GPU power, in this case, 8 NVIDIA V100 GPUs.

In this dissertation, we are studying the problem of identifying glomeruli in cationic ferritin enhanced MRI (CFE-MRI) of the kidney. CFE-MRI provides unique challenges for image segmentation to identify and measure individual glomeruli in the kidney:

- First of all, the size of glomerular is extremely small and very similar with noises in images, which calls for algorithm with good denoising property. However, most of blob detectors are prone to detect non-blob objects as false blob candidates, which results in over-detection.
- Second, there are massive of glomeruli existing in kidney images, e.g. over 1 million in human kidney, and the intensity distribution is heterogenous. This poses the algorithmic challenges on detection accuracy, as well as the computation expenses.
- A third recognized issue is that a large portion of glomeruli are overlapping and touched in images. In this case, single fixed threshold or global optimal threshold applied on probability map from U-Net is not capable of splitting up all touched areas, which results in a lower number than true quantity of blobs. This becomes the critical under-segmentation issue.

1.2 Research Objective and Contributions

The overall objective of this dissertation is to develop novel computational algorithms for imaging biomarker identification using kidney MRI. The contributions in this dissertation include:

- **A joint constrained-based blob detector, UH-DoG, is proposed for improved small blob detection:** The proposed UH-DoG detector alleviates the over-detection from Hessian analysis and under-segmentation from deep learning. Three contributions come out of the UH-DoG: First, a global blob likelihood constraint from the U-Net probability map reduces over-detection by DoG. Second, a local convex constraint from the Hessian convexity map reduces under-segmentation. Third, integrating the probability map constraint with the Hessian convexity map eliminates the need for post-pruning. Details are discussed in Chapter 2.
- **An extension of UH-DoG, A multi-threshold, multi-scale blob detector, BTCAS, is proposed for improved small blob detection and segmentation:** The proposed BTCAS detector uncovers the relationship between the deep learning threshold and the Difference of Gaussian (DoG) scale and address computational challenge from UH-DoG with improved performance. Three contributions come out of the BTCAS: First, U-Net reduces over-detection when it is used in the initial denoising step. This results in a probability map with the identified centroid of blob candidates. Second, the monotonicity of the U-Net probability map is proved to lay the foundation. Distance maps are rendered with lower and upper probability bounds, which are used as the constraints for local scale search for the DoG. Third, a local optimum DoG

scale is adapted to the range of blob sizes to better separate touching blobs. The under-segmentation typical of the U-Net is resolved. Details are discussed in Chapter 3.

- **A denoising Convexity-Consistent Generative Adversarial Network, BlobGAN, is proposed for improved small blob identification:** The proposed BlobGAN detector could achieve high denoising performance and selectively denoise the image without affecting the blobs. Three contributions come out of the BlobGAN: First, a 3D elliptical Gaussian function is mathematically derived and 3D blob images are synthesized through this function to approximate the distribution of glomeruli and extend GAN to a noise translation model. Second, a convexity consistent constraint is proposed to prevent geometric distortions of blobs during translation. Third, based on the denoising output of GAN, a joint constraint operation is applied for final blobs identification. Details are discussed in Chapter 4.
- **A Fréchet Descriptors Distance based Coreset approach, FDD-Coreset, is proposed for accelerating GAN training**: BlobGAN trained on proposed FDD-Coreset not only significantly reduce the training time, but also achieve higher denoising performance and maintain approximate performance of blob identification compared with training on entire dataset. Two contributions come out of the FDD-Coreset: A Fréchet Descriptor Distance (FDD), derived from object statistics, is first proposed to measure the difference between each pair of blob images. Second, the Coreset with k samples is selected from the entire dataset based on the proposed FDD metric. Details are discussed in Chapter 5.

1.3 Dissertation Organization

The proposed dissertation research will be presented in the following four chapters. Specifically, Chapter 2 presents the development of topic (I): Improved Small Blob Detection in 3D Images using Jointly Constrained Deep Learning and Hessian Analysis. Chapter 3 presents the development of topic (II): Small Blob Detector Using Bi-Threshold Constrained Adaptive Scales. Chapter 4 presents the development of topic (III): BlobGAN: Denoising Convexity-Consistent Generative Adversarial Network for Small Blob Identification. Chapter 5 presents the development of topic (IV): GAN Training Acceleration using Fréchet Descriptors Distance based Coreset.

CHAPTER 2

IMPROVED SMALL BLOB DETECTION IN 3D IMAGES USING JOINTLY CONSTRAINED DEEP LEARNING AND HESSIAN ANALYSIS

Imaging biomarkers are being rapidly developed for early diagnosis and staging of disease. The development of these biomarkers requires advances in both image acquisition and analysis. Detecting and segmenting objects from images are often the first steps in quantitative measurement of these biomarkers. The challenges of detecting objects in images, particularly small objects known as blobs, include low image resolution, image noise and overlap between the blobs. The Difference of Gaussian (DoG) detector has been used to overcome these challenges in blob detection. However, the DoG detector is susceptible to over-detection and must be refined for robust, reproducible detection in a wide range of medical images. In this research, we propose a joint constraint blob detector from U-Net, a deep learning model, and Hessian analysis, to overcome these problems and identify true blobs from noisy medical images. We evaluate this approach, UH-DoG, using a public 2D fluorescent dataset for cell nucleus detection and a 3D kidney magnetic resonance imaging dataset for glomerulus detection. We then compare this approach to methods in the literature. While comparable to the other four comparing methods on recall, the UH-DoG outperforms them on both precision and F-score.

2.1 Introduction

There is great interest in tailoring diagnostic and therapeutic tools to individual patients. This concept reflects the growing recognition that there is significant variability between individuals. As therapies focus on molecular targets, diagnostic medical imaging tools must reveal focal pathologies and the effects of therapy in each patient. High-

resolution object detection and image segmentation are thus critical to obtaining meaningful data in a heterogeneous image.

In image analysis, detection is used to identify objects such as organs and tumors, and segmentation is used to isolate the objects from an image. While large objects can often be automatically or semi-automatically isolated, small objects (blobs) are difficult to detect and segment. Blobs can range in size and location in images. Examples of blobs include cells or cell nuclei in images from optical microscopy (M. Zhang et al., 2015), exudative lesions in images of the retina (Sánchez et al., 2012), breast lesions in ultrasound images (Moon et al., 2013), and glomeruli in magnetic resonance (MR) images of the kidney (Beeman et al., 2011, 2014; Gao et al., 2016). Major challenges to detecting these blobs include low image resolution and high image noise. The small blobs are often numerous and can overlap each other. Many approaches have been proposed for blob detection (Crocker & Grier, 1996; Lindeberg, 1993; Meijering et al., 2009) of which intensity thresholding is among the most common (Q. Wu et al., 2007). Intensity thresholding assumes that the blobs have consistently different intensities from the background. Global differences can be addressed with a fixed threshold and local differences can be addressed with an adaptive threshold (G. Li et al., 2007; Phansalkar et al., 2011). However, the assumptions required for consistent thresholding are often violated, and thresholding alone can lead to erroneous detection or segmentation. To address this, researchers have proposed multi-step pipelines (Bergmeir et al., 2012; Guo et al., 2014) in which thresholding is only the first step. Intensity-based features are then derived using filters for improved detection. One popular class of filters is based on mathematical morphology (Dalle et al., 2009; Malpica et al., 1997). Operators such as erosion, dilation, opening and closing allow

geometrical and topological properties of objects. This approach often begins with selected seed points in the image and iteratively adds connected points to form labeled regions. Mathematical morphology is preferred when the blobs are relatively large in size and small in number. Weaknesses of this approach include the tendency to under-segment and diminished performance in the presence of noise. Under-segmentation occurs when multiple blobs within close proximity are detected as one, resulting in an erroneously low detected number. Another type of filter is based on space transformation. For example, Radial-Symmetry (Loy & Zelinsky, 2003), a point detector for small blobs, uses radially symmetric space as a transformation space to detect radially symmetric blobs. SIFT (Lowe, 2004a), SURF (Bay et al., 2008) and BRISK (Leutenegger et al., 2011) are region detectors. Each of the region detectors extracts scale invariant features to detect small objects but may suffer from poor performance in optical imaging (Kong et al., 2013a). Recently, the Laplacian of Gaussian (LoG) detector (Koenderink, 1984; Witkin, 1983), from scale space theory, has attracted attention in blob detection (Lindeberg, 1993, 1998). Similar to the radially symmetric detector, the LoG detector is unreliable in detecting rotationally asymmetric blobs. To solve this, LoG extensions have been proposed, including the Difference of Gaussian (DoG) (Lowe, 2004a; Mikolajczyk & Schmid, 2004a, 2005; Tuytelaars & Mikolajczyk, 2007) and the Generalized Laplacian of Gaussian (gLoG) (Kong et al., 2013a). While each approach detects small blobs to some extent, non-blob objects are detected as false blob candidates resulting in over-detection. A post-pruning procedure can remove false blob candidates, but results have been inconsistent (M. Zhang et al., 2016).

Here we focus on detecting individual glomeruli in MR images of the kidney as a

specific blob detection problem. To date, most biomarkers of kidney pathology have come from histology using destructive techniques that estimate glomerular number(Bertram, 1995; Bertram et al., 1992; Cullen-Mcewen et al., 2012). A non-destructive imaging approach to measuring nephron endowment provides a new marker for renal health and susceptibility to kidney disease. Cationic ferritin enhanced MRI (CFE-MRI) enables the detection of glomeruli in animals(Baldelomar et al., 2016a; Bennett et al., 2008a) and in human kidneys(Baldelomar et al., 2016a, 2017; Beeman et al., 2014; Bennett et al., 2008a). Because each glomerulus is associated with a nephron, CFE-MRI may provide an important imaging marker to detect changes in the number of nephrons and susceptibility to renal and cardiovascular disease (Beeman et al., 2014). Glomerulus detection by CFE-MRI presents difficulties because glomeruli are small and have a spatial frequency similar to image noise. Zhang et al. developed the Hessian-based Laplacian of Gaussian (HLoG) detector (M. Zhang et al., 2015) and the Hessian-based Difference of Gaussian (HDoG) detector (M. Zhang et al., 2016) to automatically detect glomeruli in CFE-MR images. They employed the LoG or DoG to smooth the images, followed by Hessian analysis of each voxel for pre-segmentation. Since LoG and DoG suffer from over-detection, a Variational Bayesian Gaussian Mixture Model (VBGMM) was implemented as a final step. LoG and DoG were the first two detectors applied to MR images of the kidney to identify glomeruli. However, deriving Hessian-based features from each blob candidate is computationally expensive, limiting high-throughput studies. In addition, unsupervised learning using the VBGMM in the post-pruning procedure requires a number of carefully tuned parameters for optimal clustering. Here we propose a new approach, termed UH-DoG, which applies joint constraints from spatial probability maps derived from U-Net, a

deep learning model, and Hessian convexity maps derived from Hessian analysis on the DoG detector. The theoretical foundation of Hessian analysis guarantees that pre-segmentation will recognize all true convex blobs and some non-blob convex objects, resulting in a blob superset. Joining probability maps allows us to distinguish true blobs from the superset. The joint-constraint extension of the detector requires no post-pruning and thus is robust, generalizable and computationally efficient.

Within the field of deep learning, the Convolutional Neural Network (CNN) has been successfully implemented in medical imaging applications ranging from object detection and segmentation to classification (Gao et al., 2018; Hoo-Chang Member et al., 2016; Kamnitsas et al., 2017). The first generation of CNN models was used to classify images through fully connected layers. Shelhamer et al. (Shelhamer et al., 2017b) first proposed a Fully Convolutional Network (FCN) that transfers the fully connected layers to deconvolutional layers and provides a dense class map with arbitrarily-sized input image. The FCN changes "image-label" mapping to "pixel/voxel-label" mapping for object detection and image segmentation. One limitation of the FCN for medical imaging is the need for large training datasets. A lightly weighted FCN model, the U-Net (Ronneberger et al., 2015a), employs a modified FCN architecture to require fewer training images but yield precise, fast segmentation. U-Net has been implemented in various medical segmentation tasks such as nucleus, cell, and breast lesion segmentation (Gao et al., 2019; Raza et al., 2017; Yap et al., 2018), all drawn from limited datasets. The U-Net yields a probability map where each pixel or voxel indicates the likelihood of being within the imaging object. However, based on our previous study (Xu et al., 2019), U-Net does not reliably separate glomeruli within close proximity. Therefore, we choose to adopt the

probability map as part of UH-DoG in conjunction with Hessian analysis for glomerulus detection from CFE-MR images.

There are three main advantages of the UH-DoG method. First, a global blob likelihood constraint from the U-Net probability map reduces over-detection by DoG. Second, a local convex constraint from the Hessian convexity map reduces under-segmentation. Third, integrating the probability map constraint with the Hessian convexity map eliminates the need for post-pruning. To validate the performance of UH-DoG, four methods were chosen from the literature: HLoG (M. Zhang et al., 2015), gLoG (Kong et al., 2013a), LoG (Witkin, 1983), and Radial-Symmetry (Loy & Zelinsky, 2003). We tested these on dataset of 2D fluorescent images (n=200) where the locations of blobs were known. UH-DoG outperformed the other four methods in F-score and performed comparably to the other four methods in recall. Next, we compared blob detection of these methods on a 3D kidney MR dataset against the HDoG method. The differences between UH-DoG and HDoG were negligible but the average computation time of UH-DoG was 35% shorter than that of HDoG.

2.2 Methods

We propose UH-DoG, a joint constraint-based detector for glomeruli detection. UH-DoG consists of three steps (see Figure 1). Step 1 is to use the Difference of Gaussian (DoG) to smooth the images, followed by Hessian analysis to identify possible blob candidates based on local convexity. Step 2 is to use a trained U-Net to generate a probability map, which captures the most likely blob locations. Step 3 is to combine the probability map from Step 2 with blob candidates from Step 1 as joint constraints to identify true blobs. Each step is discussed in detail in the following sections.

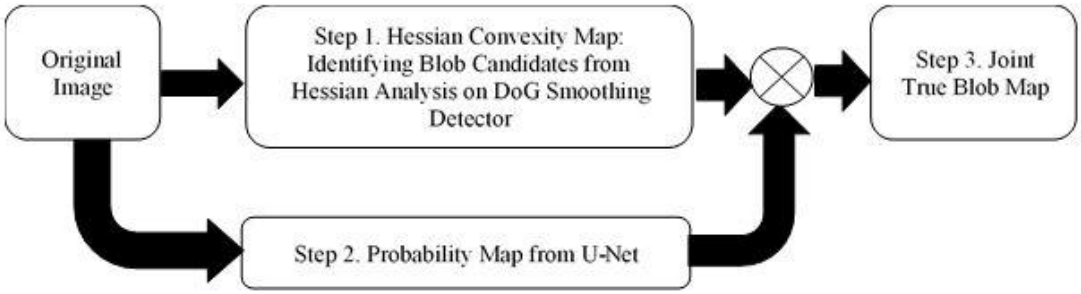


Figure 1 Proposed UH-DoG for Glomerulus Identification.

2.2.1 Hessian Analysis and Hessian Convexity Map

Before implementing Hessian analysis, DoG is used to smooth the images. By employing a convolution operator, DoG can filter image noise and enhance objects at the selected scale (Lindeberg, 1998). DoG is a fast approximation of the LoG filter to highlight blob structure (Beeman et al., 2011) and is thus computationally efficient (Lowe, 2004a).

Let a 3D image be $f: R^3 \rightarrow R$. The scale-space representation $L(x, y, z; \sigma)$ at point (x, y, z), with scale parameter σ, is the convolution of image $f(x, y, z)$ with the Gaussian kernel $G(x, y, z; \sigma)$:

$$L(x, y, z; \sigma) = G(x, y, z; \sigma) * f(x, y, z) \ , \tag{2.1}$$

where $*$ is the convolution operator and the Gaussian kernel $G(x, y, z; \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} e^{-\frac{(x^2+y^2+z^2)}{2\sigma^2}}$. The Laplacian of $L(x, y, z; \sigma)$ is:

$$\nabla^2 L(x, y, z; \sigma) = \frac{\partial^2 L(x,y,z;\sigma)}{\partial x^2} + \frac{\partial^2 L(x,y,z;\sigma)}{\partial y^2} + \frac{\partial^2 L(x,y,z;\sigma)}{\partial z^2}. \tag{2.2}$$

According to (Lowe, 2004a), $\sigma\nabla^2 L(x, y, z; \sigma) = L_\sigma(x, y, z; \sigma)$. We approximate the partial derivative $L_\sigma(x, y, z; \sigma)$ by a one-sided difference quotient, the DoG approximation of LoG is:

$$\nabla^2 L(x,y,z;\sigma) = \frac{L_\sigma(x,y,z;\sigma)}{\sigma} \approx \frac{L(x,y,z;\sigma+\Delta\sigma) - L(x,y,z;\sigma)}{\sigma\Delta\sigma}$$

$$= f(x,y,z) * \frac{(G(x,y,z;\sigma+\Delta\sigma) - G(x,y,z;\sigma))}{\sigma\Delta\sigma} \quad (2.3)$$

To locate an optimum scale for the blobs, similar to (M. Zhang et al., 2015), we add γ-normalization to form the normalized DoG detector $\sigma^\gamma \nabla^2 L(x,y,z;\sigma)$, which is:

$$DoG_{nor}(x,y,z;\sigma) = \sigma^{\gamma-1} f(x,y,z) * \frac{(G(x,y,z;\sigma+\Delta\sigma) - G(x,y,z;\sigma))}{\Delta\sigma} \quad , \quad (2.4)$$

where γ is introduced to automatically determine the optimum scale for the blobs. We set γ to 2 here. For details on tuning γ, refer to (M. Zhang et al., 2015). The normalized DoG transformation underlies Hessian-based convexity analysis to detect blobs.

After the image is smoothed by the normalized DoG, for a voxel (x,y,z) in the normalized DoG image $DoG_{nor}(x,y,z;\sigma)$ at scale σ, the Hessian matrix for this voxel is:

$$H(DoG_{nor}(x,y,z;\sigma)) = \begin{bmatrix} \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial x^2} & \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial x \partial z} \\ \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial y^2} & \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial y \partial z} \\ \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial x \partial z} & \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial y \partial z} & \frac{\partial^2 DoG_{nor}(x,y,z;\sigma)}{\partial z^2} \end{bmatrix} \quad .(2.5)$$

In a normalized DoG-transformed 3D image, each voxel of a transformed bright blob has a negative definite Hessian matrix (M. Zhang et al., 2016). We define a binary indicator matrix, $HI(x,y,z;\sigma)$, termed the Hessian convexity map. $HI(x,y,z;\sigma) = 1$ when $H\big(DoG_{nor}(x,y,z;\sigma)\big)$ is negative definite; otherwise, $HI(x,y,z;\sigma) = 0$.

To determine a single optimum scale σ^*, the maximum value of the normalized DoG is used here (M. Zhang et al., 2016). Let the average DoG value per blob candidate voxel measure B_{DoG} be:

$$B_{DoG}(\sigma) = \frac{\sum_{(x,y,z)} DoG(x,y,z) HI(x,y,z;\sigma)}{\sum_{(x,y,z)} HI(x,y,z;\sigma)} . \quad (2.6)$$

We have $\sigma^* = argmax\ B_{DoG}(\sigma)$. σ^* is used to generate the optimum Hessian convexity map $HI(x, y, z; \sigma^*)$. This map is the local convexity constraint for detecting the convex blob regions. Result is a set of convex objects including all true blobs and some non-blob convex objects.

2.2.2 U-Net and Probability Map

A classical CNN usually consists of multiple convolutional layers followed by pooling layers, activation layers, and fully connected layers. Convolutional layers learn hierarchical and high-level feature representation. Pooling layers can reduce feature dimensions and capture spatial feature invariance. The final fully connected layers categorize the images into different groups. Compared to classical CNNs, FCNs replace all fully connected layers with a fully convolutional layer. There are several advantages to this approach. First, the input image size can be arbitrary because all models consist of convolution layers, so output size only depends on input size. Second, the FCN can be trained from whole images without patch sampling, thus the effects of patch-wise training need not be considered. However, FCNs require a large dataset for training. To address this issue, a modified FCN model, the U-Net (Ronneberger et al., 2015b), was proposed. It consists of an encoding path (left) and a decoding path (right) (see Figure 2). The encoding path consists of 4 blocks; within each block, there are two 3×3 convolutional layers (Conv 3×3), a rectified linear unit (ReLU) layer, and a 2×2 max-pooling layer (Max pool 2×2). After the performance of each max-pooling layer, the resolution of the feature maps is halved, and the channel is doubled. The input images are compressed layer-wise, down through the encoding path. The corresponding decoding path has the inverse operation—which is, to reconstruct the output as a probability map back to the same size as the input

images. The resolution is increased layer-wise, up through the decoding path. To transfer information from the encoding path to the decoding path, concatenation paths are added in between the two paths (concatenation paths are marked by black arrows in Figure 2). The final layer is a 1x1 convolutional layer, followed by a sigmoid function. The sigmoid function ensures that the resultant output is a probability map in the range [0,1].

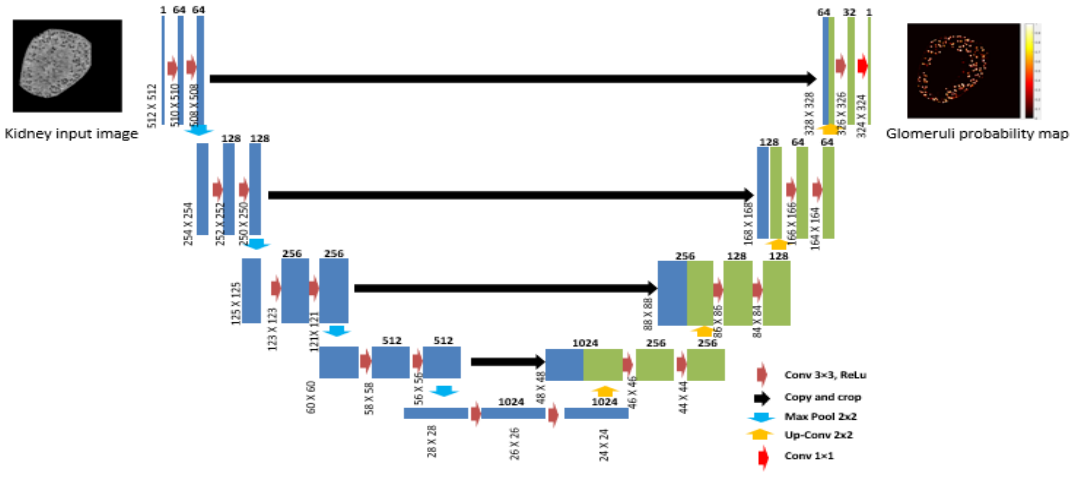


Figure 2 Architecture of U-Net Model

In U-Net, let the input images be $I \in R^{I_1 \times I_2 \times I_3}$, and the output map be $O \in R^{I_1 \times I_2 \times I_3}$, where $O^{i_1,i_2,i_3} \in [0,1]$. A binary cross entropy loss function is used in the training process to obtain the output map:

$$L = -\frac{1}{I_1 I_2 I_3} \sum_{k=1}^{I_1 \times I_2 \times I_3} O_k \cdot \log(\hat{O}_k) + (1 - O_k) \cdot \log(1 - \hat{O}_k), \qquad (2.7)$$

where O_k is the true label and $\hat{O}_k$ is the predicted probability for voxel k.

In our U-Net output, we obtain a probability map $U(x, y, z)$, where x and y are dimensions of each 2D image slice and z is the slice number. The probability map is the global blob likelihood constraint in our joint constraints operations to detect the most likely blob regions. By setting a probability threshold, most noise is removed. However, some

touching blobs could have a higher probability than the threshold in the boundary and might not be split up, resulting in reduced detection known as under-segmentation. Joining the Hessian convexity map with U-Net probability map will address the challenges of small blob detection.

2.2.3 Joint Constraint Operation for True Blob Identification

Given a 3D image $f: R^3 \rightarrow R$, Hessian analysis is applied to render a convexity map $HI(x, y, z; \sigma^*)$, U-Net is applied to render a probability map $U(x, y, z)$. We introduce a joint operator

$$UH(x, y, z) = HI(x, y, z; \sigma^*) \circ I(x, y, z) , \tag{2.8}$$

where $I(x, y, z)$ is a binary indicator matrix. Given a probability threshold δ_b, $I(x, y, z) = 1$ when $U(x, y, z) > \delta_b$; otherwise, $I(x, y, z) = 0$. We define the true blob candidate as a 27-connected voxel (Toolbox, 2004) , and the blob set is represented as:

$$S_{blob} = \{(x, y, z) | \, (x, y, z) \in DoG_{nor}(x, y, z; \sigma) \, , UH(x, y, z) = 1 \}. \tag{2.9}$$

To illustrate, Figure 3 shows images of blobs detected during the joint constraint operation of the U-Net probability map and the Hessian convexity map. The blue circle in Figure 3 (a) shows only one blob. The same blue circle on Figure 3 (b), after application of the Hessian convexity map, shows there is one "bigger" blob in the middle and a number of smaller blobs around the boundary of the blue circle. Figure 3 (c) shows the correct outcome, only one blob in the middle. This clearly illustrates the sensitivity of the Hessian matrix to noise. Even though the Hessian analysis guarantees the detection of the convex object, some non-blob convex objects (noise) will also be detected, resulting in over-detection. This noise can be readily filtered by the U-Net probability map (Figure 3 (c)).

We conclude that U-Net may be useful for denoising, which alleviates the over-detection of Hessian analysis.

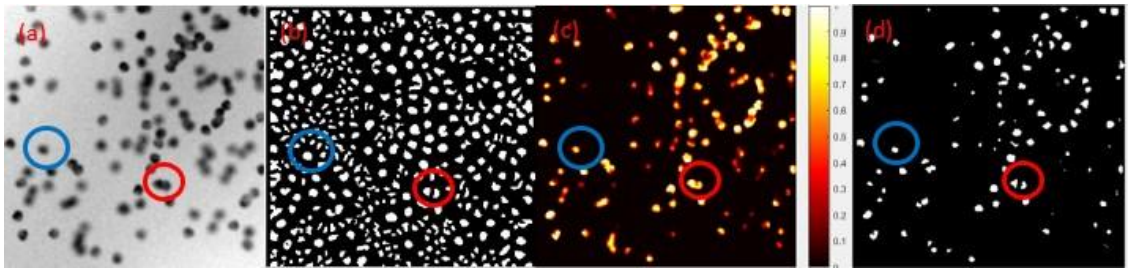


Figure 3 Visualization of Proposed UH-DoG Pipeline. (a) A 2D Gray Scale Image Preprocessed from Experiment I Fluorescent Image (b) Binary Hessian Convexity Map of (a), the Convex Pixels are Marked as the White Color. (c) U-Net Probability Map of (a), Pixel is Illustrated with a Color Indicating a Probability of the Pixel belonging to a Blob. (d) Blob Identification Map Joined from Hessian Convexity Map and U-Net Probability Map with 0.5 Threshold

The red circle in Figure 3 (a) shows overlapped blobs. They are still overlapping in the U-Net probability map from Figure 3 (c). But they are split up in the Hessian convexity map from Figure 3 (b). By joining the Hessian convexity map and U-Net probability map with a single global threshold, the overlapped blobs in the red circle are visualized as distinct entities, as shown in Figure 3 (d). We conclude Hessian analysis could alleviate the under-segmentation issue from U-Net.

Our proposed UH-DoG integrates the probability map from U-Net and convexity map from Hessian analysis to guarantee robustness to noise and effective blob detection. The detailed steps of UH-DoG are shown in Table 1.

Table 1 Detail Steps of Proposed UH-DoG

1. Use a pretrained model to generate a probability map of blobs from original image.
2. Initialize the normalization factor γ, and range and step-size of parameter σ, to transform the original image into normalized DoG space.
3. Calculate the Hessian matrix based on normalized DoG smoothed image and generate the Hessian convexity map $HI(x, y, z; \sigma)$.
4. Calculate average DoG intensity $B_{DoG}(\sigma) = \frac{\sum_{(x,y,z)} DoG(x,y,z) HI(x,y,z;\sigma)}{\sum_{(x,y,z)} HI(x,y,z;\sigma)}$ and find the optimum scale section by $\sigma^* = argmax\ B_{DoG}(\sigma)$.
5. Get the optimum Hessian convexity map $H(x, y, z; \sigma^*)$ under scale σ^*.
6. Join the probability map with Hessian convexity map to identify true blobs.

2.3 Experiments and Results

Two experiments were conducted to validate of the performance of our proposed UH-DoG detector. The first experiment validated the UH-DoG on 200 fluorescence, 2D light microscopy images for cell detection (Lempitsky & Zisserman, 2010). The 2D cell images were of interest because (1) to the best of our knowledge, there are no 3D small blob datasets available for comparison; (2) the blobs from these images are small and each image could be used to test the performance of the algorithm in the presence of background noise; (3) this dataset has the ground truth of the locations of each blob. The detection accuracy measured by recall, precision, and F-score can be used to compare this approach with methods from the literature. The second experiment validated the performance of UH-DoG on CFE-MR images of mouse kidneys where each glomerulus was detected. All experiments were approved by the University of Virginia Institutional Care and Use Committee, in accordance with the NIH Guide for the Care and Use of Laboratory Animals.

2.3.1 Training Dataset and Data Augmentation

We used a public dataset(Janowczyk & Madabhushi, 2016b) to train our deep learning model, based on optical images of cell nuclei. This dataset has 141 optical

microscopy pathology images (2,000 × 2,000 pixels), as shown in Figure 4 (a). The 12,000 ground truth annotations are typically done by an expert, which involves delineating object boundaries over 40 hours(Janowczyk & Madabhushi, 2016b). Due to the large amount of time and effort required, the annotated nuclei in this dataset only represents a small fraction of the total number of nuclei present in all images. Since we aim to facilitate U-Net to denoise our blob images based on the ground truth labeled images, as shown in Figure 4 (b), we generated Gaussian distributed noise with $\mu_{\text{noise}} = 0$ and $\sigma^2{}_{noise} = 0.01$ and we added it to the ground truth labeled images, resulting in 141 simulated training images, as shown in Figure 4 (c). Data were augmented to increase the in variance and robustness properties of U-Net (Ronneberger et al., 2015a). We generated the augmented data by a combination of rotation shift, width shift, height shift, shear, zoom, and horizontal flip.

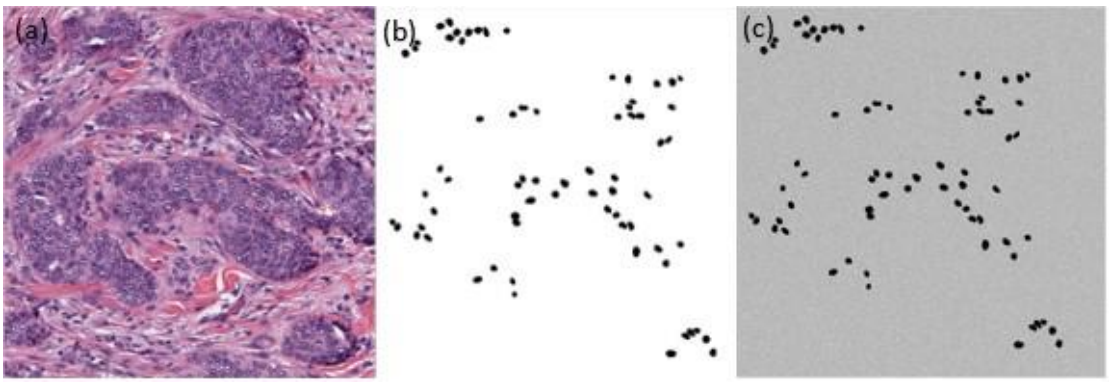


Figure 4 Training Dataset of U-Net. (a) Original Image. (b) Ground Truth Labeled Image. (c) Simulated Training Image.

2.3.2 Experiment I: Validation Experiments using 2D Fluorescent Images

Figure 5 illustrates an example fluorescent image (256 × 256 pixels). Since this was a 2D image, our proposed UH-DoG must incorporate a modified 2D DoG because comparison algorithms were from the 2D LoG and its extensions.

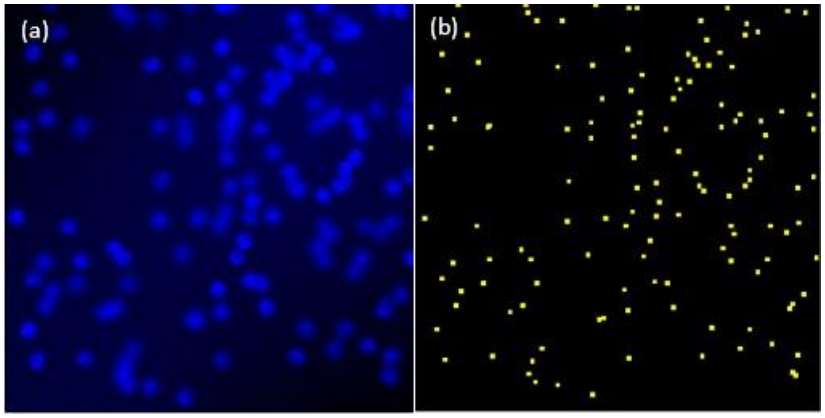


Figure 5 The 2D Fluorescent Images Dataset used in Experiment I. (a) Sample 2D Fluorescent Image. (b) Ground Truth Dots of (a).

To revise the DoG to a 2D version, for 2D images $f(x, y)$ with the Gaussian kernel $G(x, y; \sigma)$, we modified the normalized 3D DoG detector from equation (2.4) in a 2D format:

$$DoG_{nor}(x, y; \sigma) = \sigma^{\gamma-1} f(x, y) * \frac{\left(G(x,y;\sigma+\Delta\sigma) - G(x,y;\sigma)\right)}{\sigma\Delta\sigma} . \quad (2.10)$$

Then the corresponding Hessian matrix were modified from equation (2.5) as follows:

$$H(DoG_{nor}(x, y; \sigma)) = \begin{bmatrix} \frac{\partial^2 DoG_{nor}(x,y;\sigma)}{\partial x^2} & \frac{\partial^2 DoG_{nor}(x,y;\sigma)}{\partial x \partial y} \\ \frac{\partial^2 DoG_{nor}(x,y;\sigma)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y;\sigma)}{\partial y^2} \end{bmatrix} . \quad (2.11)$$

The parameter settings for Hessian analysis and DoG were as suggested in (M. Zhang et al., 2016). γ is set to 2. σ varies from 0.5 to 3 with step-size 0.5. $\Delta\sigma$ is set to 0.001.

We used precision, recall and F-score to evaluate the performance of our proposed algorithm. Precision measures the fraction of retrieved candidates confirmed by the ground-truth. Recall measures the fraction of ground-truth data retrieved. F-score measures

overall performance. Since ground truth data were provided in the form of dots (the coordinates of the blob centers), as in the literature (M. Zhang et al., 2015, 2016), a candidate was considered a true positive if its intensity centroid was within a threshold d of the corresponding ground truth dot. Specifically, if the Euclidian distance D_{ij} between dot i and blob candidate j was less than or equal to d, the blob was considered a true positive. To avoid duplicate counting, the number (#) of true positives TP was calculated by equation (2.12). Precision, recall, and F-score were calculated by equations (2.13), (2.14), (2.15) respectively:

$$TP = \min\left\{\#\left\{j: \min_{i=1}^{m} D_{ij} \le d\right\},\ \#\left\{i: \min_{j=1}^{n} D_{ij} \le d\right\}\right\}, \tag{2.12}$$

$$precision = \frac{TP}{n}, \tag{2.13}$$

$$recall = \frac{TP}{m}, \tag{2.14}$$

$$F - score = 2 \times \frac{precision \times recall}{(precision + recall)}, \tag{2.15}$$

where m is the number of ground-truth and n is the number of blob candidates; d is a thresholding parameter set to a positive value $(0, +\infty)$. If d is small, fewer blob candidates are counted since the distance between the blob candidate centroid and ground-truth should be small. If d is too large, more blob candidates are counted. Here, since local intensity extremes could be anywhere within a small blob with an irregular shape, we set d to the average diameter of the blobs: $d = 2 \times \sqrt{\frac{\sum_{(x,y)} I(x,y;\sigma)}{\pi}}$.

Since the results of detection by the complete versions of HLoG, gLoG, Radial-Symmetry and LoG on 200 pathological images are available online (Kong et al., 2013a; Loy & Zelinsky, 2003; M. Zhang et al., 2015), the results were directly used from these papers for comparison.

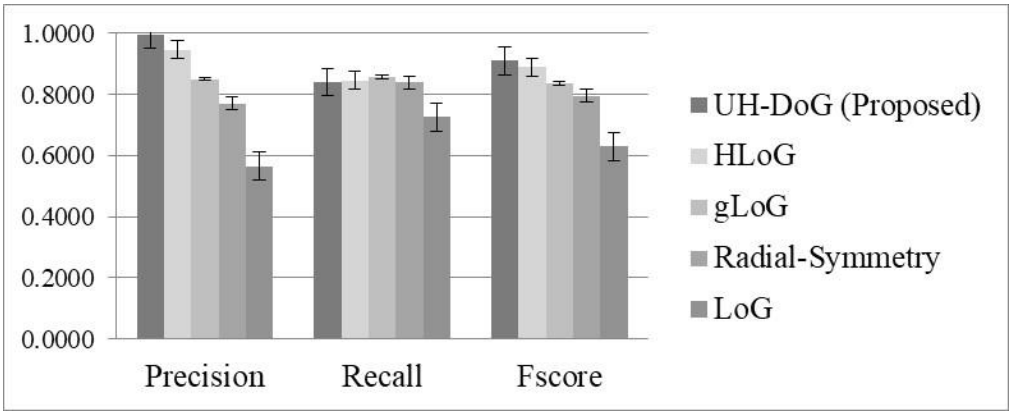


Figure 6 Comparison of Full Versions of UH-DoG, HLoG, gLoG, Radial-Symmetry and LoG on 200 Fluorescence Images. The Error Bar Indicates the Standard Deviation of the Corresponding Measure across 200 Images. For Precision and F-score, UH-DoG has Significant Different (See Table 2) with Others. For Recall, UH-DoG has Significant Difference with gLoG and LoG.

Figure 6 shows a comparison of UH-DoG to the HLoG, gLoG, LoG and Radial-Symmetry algorithms. While UH-DoG is comparable to HLoG, gLoG and Radial Symmetry algorithms in recall, it significantly outperforms the four algorithms in both precision and F-score (Table 2). The standard deviation of F-score in UH-DoG is 0.025, compared to 0.0377 with the HLoG method, compared to 0.1436 with the gLoG method, 0.0795 with the Radial-Symmetry method, and 0.0385 with the LoG method. We conclude that UH-DoG provides more accurate and robust detection of blobs in this dataset. In addition, statistical analysis was performed with the results summarized in Table 2. While comparable to the four algorithms on recall, our approach statistically outperformed the others on precision and F-score.

Table 2 ANOVA Using Tukey's HSD Pairwise Test on 200 Fluorescent Images

UH-DoG vs.	Precision	Recall	F-Score

HLoG	*< **0.0001**	0.207	***< 0.0001**
gLoG	*< **0.0001**	***0.001**	***< 0.0001**
Radial Symmetry	***< 0.0001**	0.963	***< 0.0001**
LoG	***< 0.0001**	***** **< 0.0001**	***< 0.0001**

*significance $\boldsymbol{p} < 0.05$

2.3.3 Experiment II: Validation Experiments using 3D Mouse Kidney MR Images

In this section, we conducted experiments on CF-labeled glomeruli from a dataset of 3D magnetic resonance images (256×256×256 voxels) to measure number (Nglom) and apparent size (aV_{glom}) of glomeruli in diseased kidneys and healthy control kidneys. Acute kidney injury was induced in adult male C57Bl/6 mice using an intraperitoneal injection of folic acid (125 mg). A subset of the group receiving folic acid, the AKI group (n=4) was euthanized 4 days after the folic acid was administered and the remainder of those that received folate were euthanized 4 weeks later and termed the chronic kidney disease (CKD) group, n=3. The control groups for AKI (n=5) and CKD (n=6) were age-matched adult male C57Bl/6 mice that received intraperitoneal sodium bicarbonate.

For improved detection, we adopted a preprocessing step to segment the medulla from the image because no glomeruli are located there. Based on the segmented kidney image, shown in Figure 7 (a), we converted it to a binary mask (Figure 7 (b)). Then we generated a distance mask, seen in Figure 7 (c). With the map showing the distance between each kidney's voxel and the kidney boundary, we set up a distance threshold to remove regions farther from the boundary than this threshold. Figure 7 (d) shows the 2D image slice after removing the medulla.

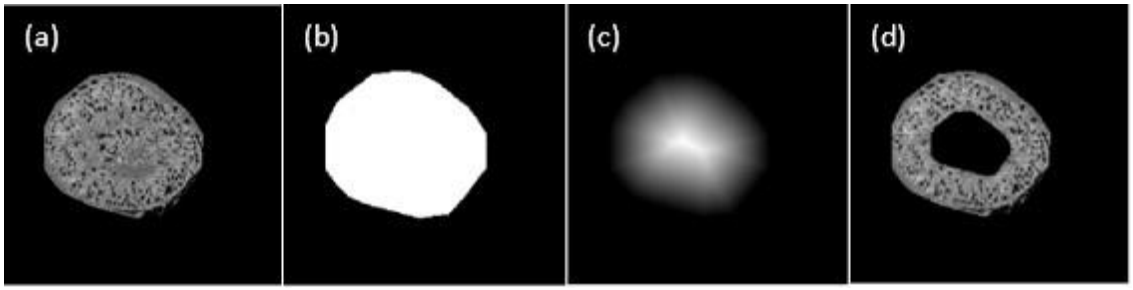


Figure 7 Preprocessing Steps for Mouse Kidneys. (a) One Slice of Healthy Mouse Kidney (ID: 477) Image. (b) Binary Image of (a). (c) Distance Mask of (b). (d) Remove Medulla from (a).

Then we performed the proposed UH-DoG method to segment the kidney glomeruli in Figure 7 (d). The parameter settings are as follows: γ is set to 2. σ varies from 0.5 to 1.8 with step-size 0.1. $\Delta\sigma$ is set to 0.001. Example segmentation results are shown in Figure 8, Figure 9, Figure 10, Figure 11. The number of glomeruli (N_{glom}), mean apparent glomerular volume (aV_{glom}) and median aV_{glom} are reported in Table 3, where the UH-DoG method is compared to the HDoG method. We used the method of calculating apparent glomerular volume from the paper(Baldelomar et al., 2017). Similarly, Table 4 summarizes the results from the AKI and control groups.

Table 3 Glomerular Number (N_{glom}) and Volume (aV_{glom}) for the CKD and Control Mouse Kidneys using the Proposed UH-DoG Method Comparing with HDoG Method

Mouse		N_{glom} (UH-DoG)	N_{glom} (HDo)	N_{glom} Difference Ratio (%)	Mean aV_{glom}* (UH-DoG)	Mean aV_{glom}* (HDoG)	Mean aV_{glom}* Difference Ratio (%)	Median aV_{glom}* (UH-DoG)	Median aV_{glom}* (HDoG)	Median aV_{glom}* Difference Ratio (%)
CKD	ID 429	7,346	7,656	**4.05**	2.92	2.57	**11.99**	1.74	1.48	**14.94**
	ID 466	8,138	8,665	**6.08**	2.06	2.01	**2.43**	1.15	0.94	**18.26**
	ID 467	8,663	8,549	**1.33**	2.32	2.16	**6.90**	1.47	1.28	**12.93**
	Avg	8,049	8,290	**2.91**	2.43	2.25	**7.67**	1.45	1.23	**15.14**

	Std	663	552		0.44	0.29		0.30	0.27	
Control	ID 427	12,701	12,724	**0.18**	1.61	1.49	**7.45**	1.26	1.15	**8.73**
	ID 469	11,347	10,829	**4.78**	2.20	1.91	**13.18**	1.41	1.20	**14.89**
	ID 470	11,309	10,704	**5.65**	2.04	1.98	**2.94**	1.50	1.37	**8.67**
	ID 471	12,279	11,943	**2.81**	1.56	1.5	**3.85**	1.22	1.13	**7.38**
	ID 472	12,526	12,569	**0.34**	1.49	1.35	**9.40**	1.16	1.06	**8.62**
	ID 473	11,853	12,245	**3.20**	1.58	1.50	**5.06**	1.25	1.18	**5.60**
	Avg	12,003	11,836	**1.41**	1.75	1.62	**7.16**	1.30	1.18	**9.10**
	Std	595	872		0.30	0.26		0.13	0.10	

*aV_{glom} unit $\boldsymbol{mm^3}$x $\boldsymbol{10^{-4}}$

Table 4 Glomerular Number (N_{glom}) and Volume (aV_{glom}) for the AKI and Control Mouse Kidneys using the Proposed UH-DoG Method Comparing with HDoG Method

Mouse		N_{glom} **(UH-DoG)**	N_{glom} **(HDoG)**	N_{glom} **Difference Ratio (%)**	**Mean** aV_{glom}* **(UH-DoG)**	**Mean** aV_{glom}* **(HDoG)**	**Mean** aV_{glom}* **Difference Ratio (%)**	**Median** aV_{glom}* **(UH-DoG)**	**Median** aV_{glom}* **(HDoG)**	**Median** aV_{glom}* **Difference Ratio (%)**
AKI	ID 433	11,033	11,046	**0.12**	1.63	1.53	**6.13**	1.27	1.17	**7.87**
	ID 462	10,779	11,292	**4.54**	1.48	1.34	**9.46**	1.17	1.00	**14.53**
	ID 463	10,873	11,542	**5.80**	2.61	2.35	**9.96**	1.60	1.25	**21.88**
	ID 464	11,340	11,906	**4.75**	2.40	2.31	**3.75**	1.59	1.17	**26.42**
	Avg	11,006	11,447	**3.85**	2.03	1.88	**7.27**	1.41	1.15	**18.47**
	Std	246	367		0.56	0.52		0.22	0.11	
Control	ID 465	10,115	10,336	**2.14**	2.40	2.30	**4.17**	1.66	1.42	**14.46**
	ID 474	11,157	10,874	**2.60**	2.52	2.44	**3.17**	1.70	1.44	**15.29**
	ID 475	10,132	10,292	**1.55**	1.70	1.74	**2.35**	1.26	1.16	**7.94**
	ID 476	10,892	10,954	**0.57**	1.62	1.53	**5.56**	1.21	1.09	**9.92**
	ID 477	11,335	10,885	**4.13**	1.70	1.67	**1.76**	1.27	1.19	**6.30**
	Avg	10,726	10,668	**0.54**	1.99	1.94	**2.62**	1.42	1.26	**11.27**
	Std	572	325		0.43	0.41		0.24	0.16	

*aV_{glom} unit $\boldsymbol{mm^3}$x $\boldsymbol{10^{-4}}$

We performed quality control by visually checking the identified glomeruli in kidney images. Figure 8 and Figure 9 show glomerular identification for CKD and control kidneys. Figure 10 and Figure 11 show glomerular identification for kidneys in the AKI and control groups.

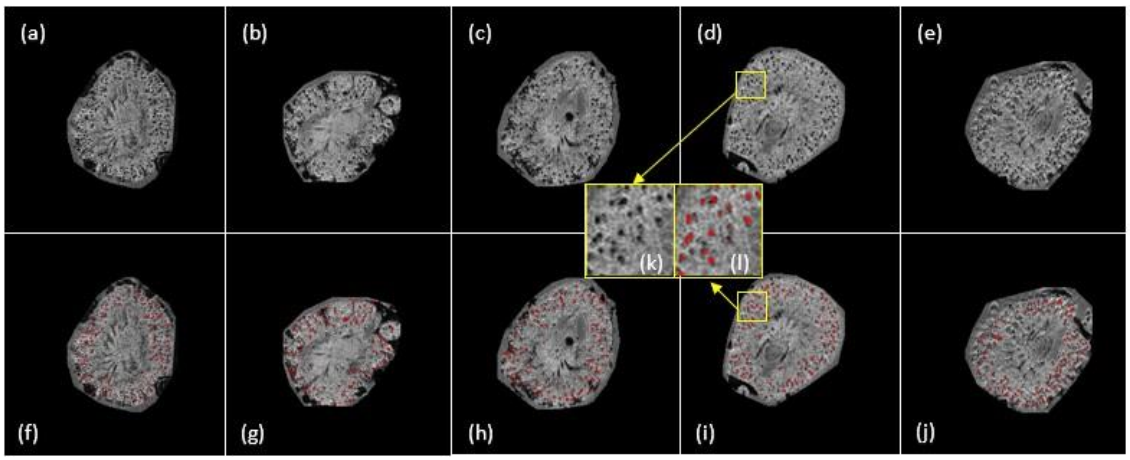


Figure 8 Glomerular Segmentation Results from 3D MR Images of Mouse Kidneys (Selected Slices Presented). (a–e) One Slice for the CKD Group. (f-j) Identified Glomeruli are Marked in Red. (k) is the Zoom-in Region of (d) while (l) is the Segmentation Result of (k).

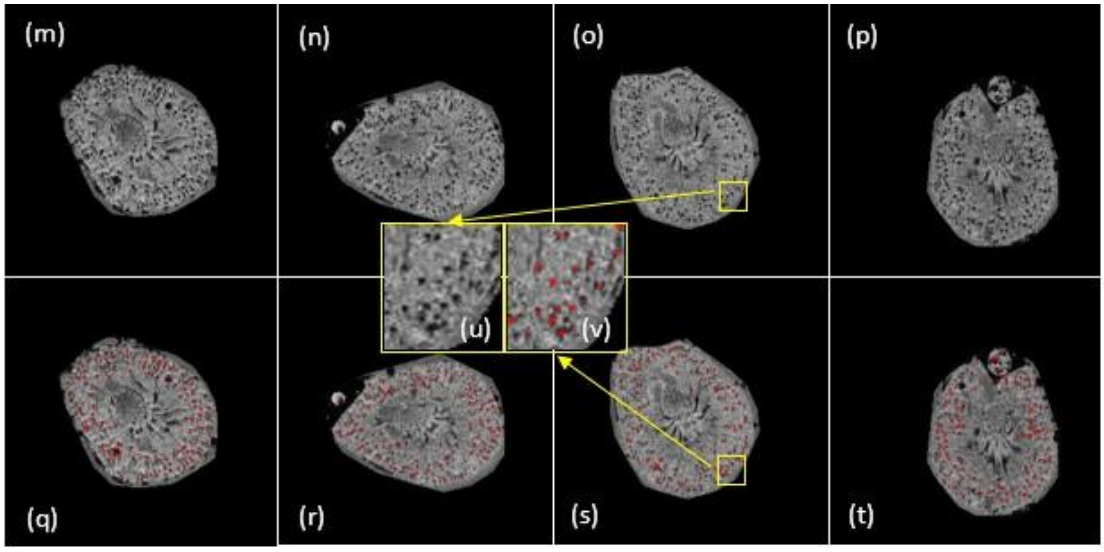


Figure 9 Glomerular Segmentation Results from 3D MR Images of Mouse kidneys (Selected Slices Presented). (m-p) One Slice for the Control Group. (q-t) Identified Glomeruli are Marked in Red. (u) is the Zoom-in Region of (o) while (v) is the Segmentation Results of (u).

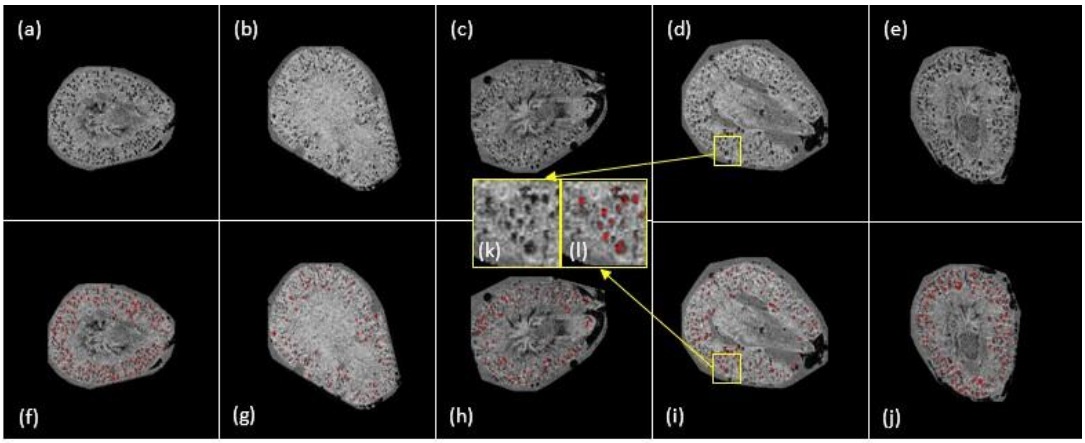


Figure 10 Glomerular Segmentation Results from 3D MR Images of Mouse Kidneys (Selected Slices Presented). (a–e) One Slice for the AKI Group. (f-j) Identified Glomeruli are Marked in Red. (k) is the Zoom-in Region of (d) while (l) is the Segmentation Result of (k)

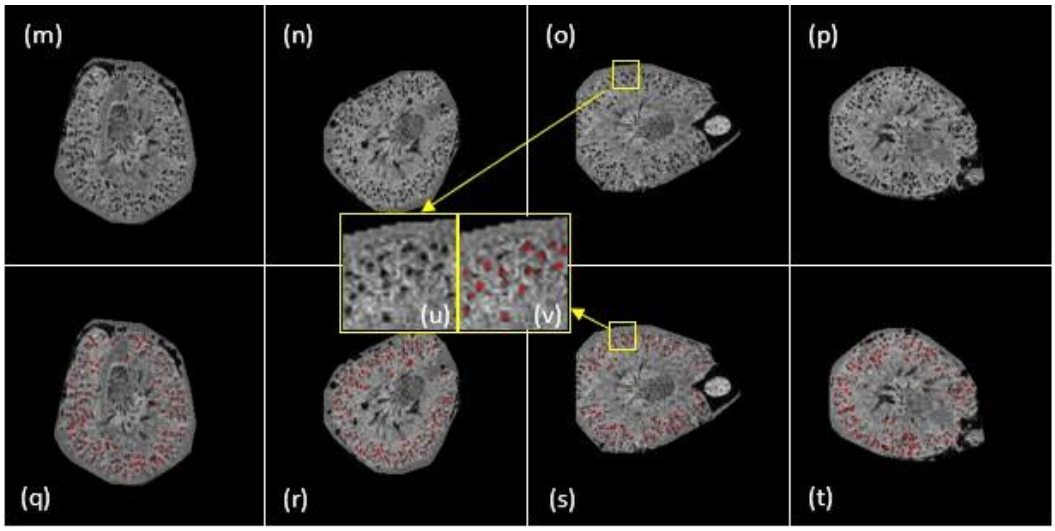


Figure 11 Glomerular Segmentation Results from 3D MR Images of Mouse Kidneys (Selected Slices Presented). (m-p) One Slice for the Control Group. (q-t) Identified Glomeruli are Marked in Red. (u) is the Zoom-in Region of (o) while (v) is the Segmentation Results of (u)

2.3.4 Discussion: Computation Cost

UH-DoG significantly decreases computation time compared to the HDoG algorithm (M. Zhang et al., 2016), as shown in Table 5 and Table 6. The training time of U-Net is not included in the estimates of computation time as it is trained beforehand and can be used to test on all images.

Table 5 Computation Time for CKD and Control Kidneys using HDoG and the Proposed Method with Scale = 1 (Intel Xeon 3.6 GHz CPU and 16 GB of Memory, NVIDIA TITAN XP and 12 GB of Memory)

Mouse		HDoG (seconds)	UH-DoG (seconds)
CKD	ID 429	9.3	7.3
	ID 466	9.5	7.3
	ID 467	11.4	7.6
	Avg	10.1	7.4
	Std	1.2	0.2
Control	ID 427	11.7	8.2
	ID 469	11.7	8.0
	ID 470	12.0	8.0
	ID 471	11.9	8.0
	ID 472	12.0	8.1
	ID 473	25.2	8.2
	Avg	14.1	8.1
	Std	5.5	0.1

Table 6 Computation Time for AKI and Control Kidneys Using HDoG and the Proposed Method with Scale = 1 (Intel Xeon 3.6 GHz CPU and 16 GB of Memory, NVIDIA TITAN XP and 12 GB of Memory)

Mouse		HDoG (seconds)	UH-DoG (seconds)
AKI	ID 433	13.7	7.9
	ID 462	13.4	8.0
	ID 463	13.1	8.0
	ID 464	14.3	8.3
	Avg	13.6	8.1
	Std	0.5	0.2

Control	ID 465	11.0	7.8
	ID 474	12.3	8.0
	ID 475	11.4	7.8
	ID 476	12.0	8.1
	ID 477	11.6	7.9
	Avg	11.7	7.9
	Std	0.5	0.1

2.3.5 Discussion: Clinical Translation

The use of imaging biomarkers in humans has increased both for disease early detection and disease severity assessment. Additionally, imaging biomarkers can serve as surrogate endpoints in clinical trials, reducing cost and burden associated with these studies. For example, total kidney volume has recently been accepted as a surrogate marker for disease progression in autosomal dominant polycystic kidney disease trials (Tangri et al., 2017). Although the importance of glomerular number has been universally accepted, the detection of glomerular number and size has been limited because the only methodology to obtain these metrics are destructive stereological approaches that could only be performed post-mortem. With the advent of CFE-MRI, the need for image analysis tools is paramount. However, it is critical to the success of any imaging biomarker that the marker be accurate and rapidly obtained. This study demonstrates some of the challenges in detecting small objects, such as glomeruli, particularly in the settings of low image resolution, image noise and overlap of objects. It also shows the promise of rapid acquisition where data can be used in a timeframe to influence patient care. Further work is necessary to validate the accuracy of the detection of diseased glomeruli to apply this algorithm to a wider range of renal disease models.

2.4 Conclusion

Discovering imaging biomarkers is important to inform disease diagnosis, prognosis, therapy development and treatment assessment. Of particular interest in this research is to identify quantitative glomeruli biomarkers from CFE-MR image. This is a challenging problem because the number of glomeruli is large, the size is small. In addition, the limitation from imaging acquisition such as hardware and variable acquisition parameters often renders the images with less desirable resolution resulting the overlapping glomeruli. In this paper, we demonstrated a new small blob detector by joining the Hessian convexity map and probability map from U-Net. This joint constraint-based approach overcomes under-segmentation by U-Net and over-detection by Hessian analysis. While it was successfully implemented in segmenting the kidney glomeruli, there are still some limitations. First, the assumption that the blobs are convex and similar in size may not be robust for non-convex objects with difference sizes. A future possible improvement is to enhance ability of U-Net to detect both convex and non-convex small objects. Second, the probability map is sensitive to the threshold. We plan to explore the use of thresholding to improve UH-DoG.

CHAPTER 3

SMALL BLOB DETECTOR USING BI-THRESHOLD CONSTRAINED ADAPTIVE SCALES

Recent advances in medical imaging technology bring great promises for medicine practices. Imaging biomarkers are discovered to inform disease diagnosis, prognosis, and treatment assessment. Detecting and segmenting objects from images are often the first steps in quantitative measurement of these biomarkers. The challenges of detecting objects in images, particularly small objects known as blobs, include low image resolution, image noise and overlap among the blobs. This research proposes a Bi-Threshold Constrained Adaptive Scale (BTCAS) blob detector to uncover the relationship between the U-Net threshold and the Difference of Gaussian (DoG) scale to derive a multi-threshold, multi-scale small blob detector. With lower and upper bounds on the probability thresholds from U-Net, two binarized maps of the distance are rendered between blob centers. Each blob is transformed to a DoG space with an adaptively identified local optimum scale. A Hessian convexity map is rendered using the adaptive scale, and the under-segmentation typical of the U-Net is resolved. To validate the performance of the proposed BTCAS, a 3D simulated dataset (n=20) of blobs, a 3D MRI dataset of human kidneys and a 3D MRI dataset of mouse kidneys, are studied. BTCAS is compared against four state-of-the-art methods: HDoG, U-Net with standard thresholding, U-Net with optimal thresholding, and UH-DoG using precision, recall, F-score, Dice and IoU. We conclude that BTCAS statistically outperforms the compared detectors.

3.1 Introduction

Imaging biomarkers play a significant role in medical diagnostics and in monitoring disease progression and response to therapy (Abramson et al., 2015; Gao et al., 2017; Litjens et al., 2014; Lord et al., 2016; T. Wu et al., 2017). The development and validation of imaging biomarkers involves the detection, segmentation and classification of imaging features. Deep learning tools have been recently developed to perform these functions. For example, convolutional neural networks (CNNs) have been applied to magnetic resonance (MR) and X-ray computed tomography (CT) images (Esteva et al., 2017; Gao et al., 2018, 2020; Lee et al., 2019; S. Liu et al., 2015; Rouhi et al., 2015) and recurrent neural networks (RNN) have been applied to functional and molecular images such as positron emission tomography (PET) (T. C. Chiang et al., 2019; Kashif et al., 2016; M. Liu et al., 2018) for image classification. A convolutional RNN model was designed to detect mitosis from cell videos (Phan et al., 2019), and a CNN model was developed to generate probability maps to initialize and model cell nuclear shape and fine-tune nuclei for segmentation in optical images (Xing et al., 2016). An ensemble of CNN models with differently sized filters was used to detect pulmonary nodules in CT images (Dou et al., 2017). However, deep learning tools are strongly affected by the quality of the images.

Recently, imaging tools have been developed to precisely map and measure individual glomeruli in the kidney using an injected contrast agent, (cationic ferritin, CF), which binds to the glomerular basement membrane and creates a dark spot in gradient-echo MR images (Baldelomar et al., 2016a, 2017; Bennett et al., 2008a). The emerging field of CFE-MRI provides comprehensive, 3D measurements of histologic features of the kidney that may aid in early detection of kidney pathology (Baldelomar et al., 2019; Beeman et al., 2011, 2014). Glomeruli appear as small blobs in CFE-MR images. A number of blob

detectors have been developed for small blob detection such as nuclei detection (Ho et al., 2018; Wahab et al., 2019), cell detection (Mahmood et al., 2019; Raza et al., 2019; Xue et al., 2019), among which scale-space based blob detectors have attracted great attention. For example, Kong et al. proposed the generalized Laplacian of Gaussian (gLoG) (Kong et al., 2013a), which accurately detected blobs of various scales, shapes and orientations from histologic and fluorescent microscopic images. Zhang et al. developed the Hessian-based blob detectors HLoG (M. Zhang et al., 2015) and HDoG (M. Zhang et al., 2016) to automatically detect glomeruli in CFE-MR images with high accuracy and efficiency. However, these blob detectors are not robust to noise (Xu et al., 2019), leading to high false positive rates. Deep learning has recently been applied to detect and segment blob- like objects. One approach is to apply a CNN to identify patches enclosing objects first, and then perform post-processing to segment the objects. For example, Ciresan et al. (Ciresan et al., 2013) applied a CNN to automatically detect cells in histologic images of breast cancer. Based on the probable location of the centroid of the cell, derived from the CNN, they used non-maxima suppression to identify the cells. Images can also be first pre-processed and then divided into patches, which are then confirmed by a CNN. For example, Khoshdeli et al. (Khoshdeli & Parvin, 2018) used non-negative matrix factorization and a Laplacian of Gaussian (LoG) filter to initially identify blobs, and used a CNN model to detect nuclei. Fully convolutional networks (FCNs) (Shelhamer et al., 2017b) have been proposed for segmentation at the pixel or voxel level. FCNs transfer the CNN's fully connected layers to convolutional layers, providing a map at pixel or voxel scale to detect the object (Xing et al., 2019). However, FCNs often require large datasets for training, limiting their potential use in medical applications where sample sizes are often small. To

resolve this issue, U-Net (Falk et al., 2019), a modified version of FCN, was developed to achieve fast and accurate segmentation (Esser et al., 2018; Gao et al., 2019; Xu et al., 2020).

CFE-MRI provides unique challenges for image segmentation to identify and measure individual glomeruli in the kidney. First, the size of a glomerulus is on the order of the image voxel (~ 100 μm) and the spatial frequency of glomeruli is close to that of image noise, requiring algorithms with good denoising ability. Since most existing blob detectors suffer from over-detection, post-pruning is often employed to correct false positives (M. Zhang et al., 2015). Second, a large fraction of glomeruli overlap in the images. A single threshold applied to the probability map derived from U-Net may not separate overlapping glomeruli, leading to under-segmentation and a high false negative rate. To address both over-detection and under-segmentation, the UH-DoG detector was proposed to take advantage of the complementary properties of U-Net and HDoG (Xu et al., 2020). The probability map from U-Net provides blob likelihood in the whole image, and the Hessian map from HDoG indicates local convexity among a group of neighborhood pixels or voxels. Joining the two maps was initially promising for detecting glomeruli (Xu et al., 2020). However, UH-DoG employs a single threshold-single scale approach, which may pose the following challenges: (1) A single threshold applied to the U-Net probability map may not be sensitive to noise to minimize under-segmentation. (2) A single optimum scale applied to the DoG space may overlook large variations in blob size. One possible solution is to exhaustively explore multiple thresholds in U-Net and multiple scales in DoG. Unfortunately, the massive number of glomeruli (> 1 million in a human kidney) with varying sizes makes such attempts computationally prohibitive.

This work aims to uncover the relationship between the U-Net threshold and the DoG scale to derive a multi-threshold, multi-scale small blob detector. We first prove the monotonicity of the U-Net probability map, laying the foundation for the proposed detector. With lower and upper bounds on the probability thresholds, we then render two binarized maps of the distance between blob centers. Since the true blob will fall between the two distance maps with a specified level of certainty, the search space for the DoG scales is bounded. Each blob can then be transformed to an optimum local DoG space locally, instead of by a single global optimum scale. A Hessian convexity map is rendered using an adaptive scale, and the under-segmentation typical of the U-Net is resolved. We term this approach the Bi-Threshold Constrained Adaptive Scale (BTCAS) blob detector. To validate the performance of BTCAS blob detector, we first study a 3D simulated dataset (n=20) where the locations of blobs are known. Four methods are chosen from the literature: HDoG (M. Zhang et al., 2016), U-Net with standard thresholding (Falk et al., 2019), U-Net with optimal thresholding (Xu et al., 2019), and UH-DoG (Xu et al., 2020) for comparison. Next, we compare blob detection using these methods applied to a 3D image of three human kidneys and a set of 3D image of mouse kidneys from CFE-MRI against the HDoG, UH-DoG and stereology.

3.2 Methods

Our proposed Bi-Threshold Constrained Adaptive Scale (BTCAS) blob detector consists of two steps to detect blobs (glomeruli) from CFE-MRI of the kidney: (1) Training U-Net to generate a probability map to detect the centroids of the blobs, and then deriving two distance maps with bounded probabilities; (2) Applying the Difference of Gaussian

(DoG) with an adaptive scale constrained by the bounded distance maps, followed by Hessian analysis for final blob segmentation.

3.2.1 Bi-Threshold Distance Maps from U-Net

U-Net consists of an encoding path (left) and a decoding path (right), (Figure 2). The encoding path has four blocks. Within each block, there are two 3×3 convolutional layers (Conv 3×3), a rectified linear unit (ReLU) layer, and a 2×2 max-pooling layer (Max pool 2×2). After each max-pooling layer, the resolution of the feature maps is halved and the channel is doubled. The input images are compressed by layer, through the encoding path. The corresponding decoding path performs the inverse operation to reconstruct the output as a probability map of the same size as the input images. The resolution is increased by layer through the decoding path. To transfer information from the encoding path to the decoding path, concatenation paths are added between them, marked by black arrows in Figure 2. The final layer is a 1x1 convolutional layer, followed by a sigmoid function. This sigmoid function ensures that the resultant output is a probability map. In supervised learning applications where the output labeling is known, U-Net can be directly used as a model for segmentation. When the output labeling is unknown, U-Net can be used to process and denoise the images (Batson & Royer, 2019; H. T. Chiang et al., 2019; Komatsu & Gonsalves, 2019; Song et al., 2019). Here, since the ground truth is unknown, we investigate the denoising capabilities of U-Net. It is common to use autoencoders to denoise images. However, in CFE-MRI of the kidney, the glomeruli are extremely small, similar to noise that can be potentially removed by autoencoders. The major difference between U-Net and autoencoders is that U-Net has concatenation paths, which can transfer fine-grained information from low layers to high layers to increase the performance of the

segmentation results. Therefore, U-Net may have the advantage over autoencoder model by removing background noise from the MR images and simultaneously enhancing the glomerular detection.

Let $X \in [0,1]^{I_1 \times I_2 \times I_3}$ be the input image and $Y \in \{0,1\}^{I_1 \times I_2 \times I_3}$ be the image after being denoised. For simplicity, assume input image X has Gaussian noise ε. We have

$$X = Y + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I). \tag{3.1}$$

U-Net is to obtain a function $\mathcal{F}(\cdot)$ mapping X to Y by learning and optimizing the parameters Θ of convolutional and deconvolutional kernels. This is achieved by minimizing the global loss function:

$$\mathcal{L}(\Theta) = \frac{1}{N} \sum_{i=1}^{n} loss\big(Y, \mathcal{F}(X; \Theta)\big), \tag{3.2}$$

Where N is the sample size, $\mathcal{F}(X; \Theta) \in [0,1]^{I_1 \times I_2 \times I_3}$ is the probability map followed by the sigmoid activation function, $loss(\cdot)$ is a binary cross entropy loss function defined as

$$loss\big(Y, \mathcal{F}(X; \Theta)\big) = -\frac{1}{I_1 I_2 I_3} \sum_{k=1}^{I_1 \times I_2 \times I_3} y_k \cdot log\big(\mathcal{F}_k(X; \Theta)\big) + (1 - y_k) \cdot log\ (1 - \mathcal{F}_k(X; \Theta)), \tag{3.3}$$

where $y_k \in \{0,1\}$ is the true label and $\mathcal{F}_k(X; \Theta) \in [0,1]$ is the predicted probability for voxel k. After denoising, the output of $\mathcal{F}(\cdot)$ approximates Y:

$$\mathcal{F}(X; \Theta) \approx Y. \tag{3.4}$$

Glomeruli in CFE-MR images are roughly spherical in shape, with varying image magnitudes. Based on this observation, we develop the **Proposition 1** in Appendix A.

The first use of **Proposition 1** is to identify the centroid of any blob. From **Proposition 1**, the centroid of any bright blob reaches maximum probability. Therefore, a regional maximum function RM can be applied to the probability map $U(x, y, z)$ to find

voxels with maximum probability from the connected neighborhood voxels as blob centroids:

$$RM(U) = \max_{\substack{u \in U(x,y,z) \\ \Delta u \in (-k,k)}} U(u + \Delta u)\,, \tag{3.5}$$

where k is the Euclidean distance between each voxel with its neighborhood voxels. The blob centroid set $C = \{C_i\}_{i=1}^{N}$ is defined as:

$$C = \{(x,y,z) | (x,y,z) \in \arg RM\big(U(x,y,z)\big)\}\,. \tag{3.6}$$

Here, k = 1. Each blob centroid $C_i \in C$ has maximum probability within 6-connected neighborhood voxels.

A second use of ***Proposition 1*** is to binarize the probability map with a confidence level. We first use Otsu's thresholding (Otsu & N., 1996) to remove noise and voxels in the blob centroids, and to extract the probability distribution of blob voxels. Next, instead of using single threshold, we apply the two-sigma rule to the distribution to identify the lower probability δ_L and higher probability δ_H covering 95% range of the probabilities. As a result, the probability map can be binarized to $B_L(x,y,z) \in \{0,1\}^{I_1 \times I_2 \times I_3}$ and $B_H(x,y,z) \in \{0,1\}^{I_1 \times I_2 \times I_3}$.

$$B_{L/H}(x,y,z) = \begin{cases} 1, & U(x,y,z) \geq \delta_{L/H} \\ 0, & U(x,y,z) < \delta_{L/H} \end{cases} \tag{3.7}$$

From ***Proposition 2***, $B_L(x,y,z)$ will approximate a blob with larger size and $B_H(x,y,z)$ will approximate a blob with smaller size. Without loss of generality, let $B(x,y,z)$ be a binarized probability map and define $\Omega = \{(x,y,z) | B(x,y,z) = 1\}$ as the set of blob voxels and $\partial\Omega$ the set of boundary voxels. $d(\cdot)$ is the Euclidean distance

function of any two voxels. The Euclidean distance of each voxel with the nearest boundary voxels is:

$$d(p, \partial\Omega) = \min_{\substack{p\in\Omega \\ q\in\partial\Omega}} d(p, q). \tag{3.8}$$

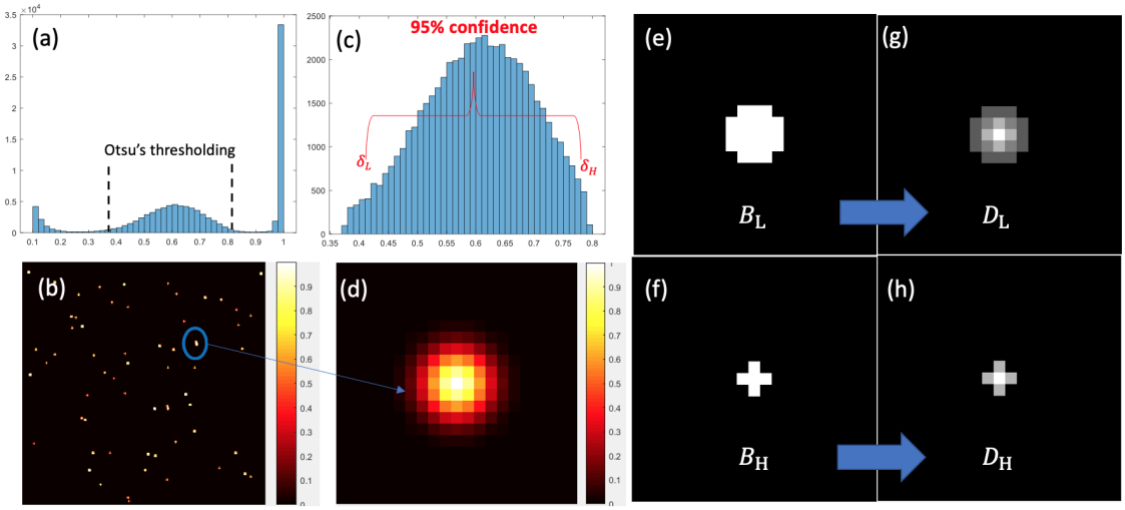


Figure 12 Approach to Derive the Distance Maps from Probability Map: (a) Probability Distribution of Probability Map. (b) Visualization of Probability Map. (c) Probability Distribution after Applying Otsu's Thresholding. (d) Visualization of Blob's Probability. (e) Binarized Probability Map B_L under Low Threshold δ_L. (f) Binarized Probability Map B_H under Low Threshold δ_H. (g) Distance Map D_L Derived from B_L. (h) Distance Map D_H Derived from B_H

Given $B_L(x, y, z)$ and $B_H(x, y, z)$, two distance maps are derived, $D_L(x, y, z) \in R^{I_1 \times I_2 \times I_3}$ and $D_H(x, y, z) \in R^{I_1 \times I_2 \times I_3}$ respectively. Figure 12 illustrates the process from the probability map, Bi-Threshold (lower and upper bound) to binarized distance maps.

For each blob centroid $C_i \in C$ (see equation (3.9)), we approximate radius r_i of blob i as:

$$r_i \in (D_H(C_i), D_L(C_i)\ ). \tag{3.9}$$

As proved in (Lindeberg, 1998), the smoothing scale in DoG is positively correlated with the blob radius. Here we will use this bounded radius information in equation (3.9) to constrain the adaptive scales in DoG imaging smoothing, as described in the next section.

3.2.2 Bounded Adaptive Scales in DoG and Hessian Analysis

For a normalized 3D image $X(x,y,z) \in [0,1]^{I_1 \times I_2 \times I_3}$, a DoG filter is

$$DoG(x,y,z;s) = X(x,y,z) * \frac{\left(G(x,y,z;s+\Delta s) - G(x,y,z;s)\right)}{\Delta s}, \tag{3.10}$$

where s is the scale value, $*$ is convolution operator, and Gaussian kernel $G(x,y,z;s) = \frac{1}{(2\pi s^2)^{\frac{3}{2}}} e^{-\frac{(x^2+y^2+z^2)}{2s^2}}$. The DoG filter smooths the image more efficiently in 3D than the LoG filter does (M. Zhang et al., 2016). However, determining the optimum DoG scale in blob detection is challenging. Zhang et al. (M. Zhang et al., 2015) proposed to use a single global optimal scale for all blobs by identifying the maximum DoG in the whole image. This approach guarantees efficient smoothing when all blobs are similar in size. For blobs with a wide range of sizes, if blob size is smaller than the DoG scale, the blob will be smoothed; if the blob size is larger than the DoG scale, only part of the blob will be smoothed. Adaptive scales have been proposed to alleviate this issue. One example is from Yousef et al. (Al-Kofahi et al., 2010) where a distance map was generated from the binarized map and graph-cut was applied to constrain the range of LoG scale. With the constrained range, each blob has an optimum scale when the LoG is at a local maximum. The authors acknowledged that one potential issue from graph-cut is under-segmentation (Al-Kofahi et al., 2010). In addition, the LoG is less computationally affordable compared to the DoG for 3D images.

Recognizing the merits and the challenges from the adaptive scales developed in (Al-Kofahi et al., 2010). Here we apply the distance maps (D_L and D_H) from U-Net to constrain the DoG scale for scale inference. Specifically, for a d-dimensional images, the DoG will reach a maximum response under scale $s = r/\sqrt{\mathrm{d}}$ (Lindeberg, 1998). In a 3D image, let the range of scale for each blob be $s_i \in (s_i^L, s_i^H)$. By substituting r with equation (3.9), we get:

$$s_i^L = D_H(C_i)/\sqrt{3} \tag{3.11}$$

$$s_i^H = D_L(C_i)/\sqrt{3} \tag{3.12}$$

For each blob, a normalized $DoG_{nor}(x, y, z; s_i)$ with multi-scale $s_i \in (s_i^L, s_i^H)$ is applied on a small 3D window with size $N \times N \times N$ ($N > 2 * D_L(C_i)$) and window center is the blob centroid $C_i \in C$.For each voxel (x, y, z) in $DoG_{nor}(x, y, z; s_i)$ at scale s_i, the Hessian matrix for this voxel is:

$$H(DoG_{nor}(x,y,z;s_i) = \begin{bmatrix} \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial x^2} & \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial x \partial z} \\ \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial y^2} & \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial y \partial z} \\ \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial x \partial z} & \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial y \partial z} & \frac{\partial^2 DoG_{nor}(x,y,z;s_i)}{\partial z^2} \end{bmatrix} \tag{3.13}$$

In a normalized DoG-transformed 3D image, each voxel of a transformed bright or dark blob has a negative or positive definite Hessian (M. Zhang et al., 2016). Taking a bright blob as an example, we define the Hessian convexity window, $HW(x, y, z; s_i)$, a binary indicator matrix:

$$HW(x,y,z;s_i) = \begin{cases} 1, & if \ \ H\big(DoG_{nor}(x,y,z;s_i)\big) \ is \ negative \ definite \\ 0, & otherwise \end{cases} \tag{3.14}$$

For each blob with centroid $C_i \in C$, let the average DoG value for each window BW_{DoG} be:

$$BW_{DoG}(s_i) = \frac{\sum_{(x,y,z)} DoG(x,y,z)HW(x,y,z;s_i)}{\sum_{(x,y,z)} HW(x,y,z;s_i)}. \quad (3.15)$$

The optimum scale s_i^* for each blob is determined if $BW_{DoG}(s_i^*)$ is maximum with $s_i \in (s_i^L, s_i^H)$. We derive the optimum scale s_i^* for each blob with centroid $C_i \in C$. The final segmented blob set S_{blob} is:

$$S_{blob} = \{(x,y,z) | \ (x,y,z) \in DoG_{nor}(x,y,z;s_i^*), HW(x,y,z;\ s_i^*) = 1\}. \quad (3.16)$$

The details of proposed BTCAS blob detector are summarized in Table 7.

Table 7 Pseudocode for BTCAS Blob Detector

1. Use a pretrained model to generate a probability map of blobs from original image.
2. Initialize probability range (δ_L, δ_H) and thresholding probability map to get binarized map $B(x,y,z)$ and distance map $D(x,y,z)$
3. Calculate the blob centroids set C from probability map $U(x,y,z)$. For each blob with centroid $C_i \in C$, get the scale range (s_i^L, s_i^H).
4. For each blob with centroid $C_i \in C$, transform raw image window of blob to multi-scale DoG space with scale $s_i \in (s_i^L, s_i^H)$.
5. Calculate the Hessian matrix based on normalized DoG smoothed window and generate the Hessian convexity window $HW(x,y,z;s_i)$.
6. Calculate average DoG intensity of each window $BW_{DoG}(s) = \frac{\sum_{(x,y,z)} DoG(x,y,z)HW(x,y,z;s)}{\sum_{(x,y,z)} HW(x,y,z;s)}$ and find the optimum scale for each blob by $s_i^* = argmax\ BW_{DoG}(s_i)$
7. Get the optimum Hessian convexity window $HW(x,y,z;s_i^*)$ under scale s_i^*.
8. Identify the final segmented blob voxels set S_{blob}.

3.3 Experiments and Results

3.3.1 Training Dataset and Data Augmentation

We used a public dataset (Janowczyk & Madabhushi, 2016b) of optical images of cell nuclei to train U-Net. This dataset contains 141 pathology images (2,000 × 2,000

pixels). The 12,000 ground truth annotations were provided by a domain expert, which involved delineating object boundaries over 40 hours. Since we aimed to facilitate U-Net to denoise our blob images based on the ground truth labeled images, we generated Gaussian distributed noise with $\mu_{niose} = 0$ and $\sigma^2_{noise} = 0.01$, which were added to the labeled images, resulting in 141 synthetic training images as shown in Figure 13 (g-i). Data were augmented to increase the invariance and robustness of U-Net. We generated the augmented data by a combination of rotation shift, width shift, height shift, shear, zoom, and horizontal flip. The trained model is validated using 3D synthetic image data and 3D MR image data.

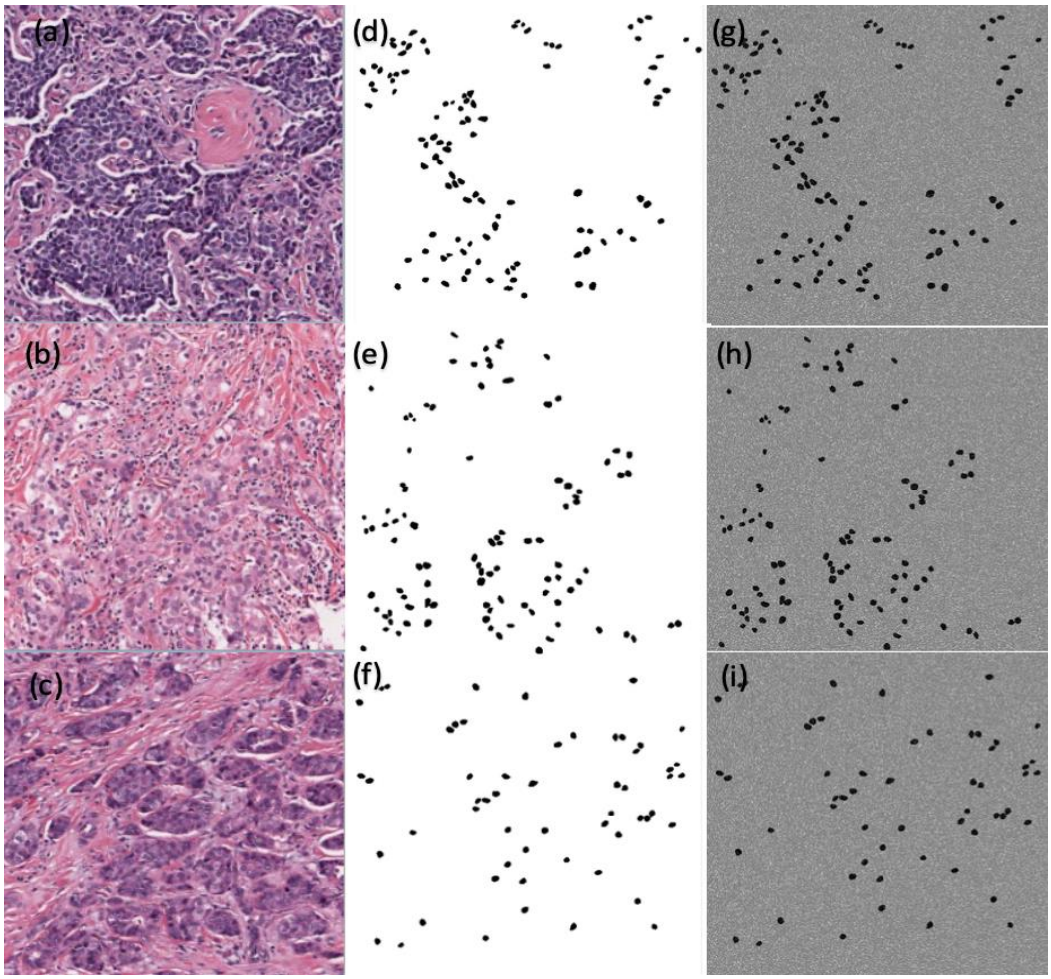


Figure 13 U-Net Training Dataset: (a-c) Original Images. (d-f) Ground Truth Labeled Images for (a-c). (g-i) Synthetic Training Images based on (d-f).

3.3.2 Experiment I: Validation Experiments using 3D Synthetic Image Data

We simulated 20, 3D images with 10 different numbers of blobs and two different levels of noise. From each 3D image (sized 256×256×256), blobs were generated using the Gaussian function with parameter $s = 1$ for blob size. The radii of the blobs was approximated as $(2 \times s + 0.5)$ voxels, based on observation. Blobs were spread on the images at random locations. The number of blobs (N) ranged from 5,000 to 50,000 with a step size of 5,000. Noise was generated by the Gaussian function with $\mu_{niose} = 0$ and σ^2_{noise} defined by:

$$\sigma^2{}_{noise} = \frac{\sigma^2{}_{image}}{10^{\frac{SNR}{10}}} \ . \tag{3.17}$$

The signal-to-noise ratio (SNR) was set at 1dB and 5dB for high noise and low noise, respectively. As the quantity of blobs increased, so did blob density, which resulted in a large number of blobs being closely clumped together (see Figure 14). We derived the ratio of overlap (O) of blobs in the 3D image:

$$O = \frac{N_o}{N_T} . \tag{3.18}$$

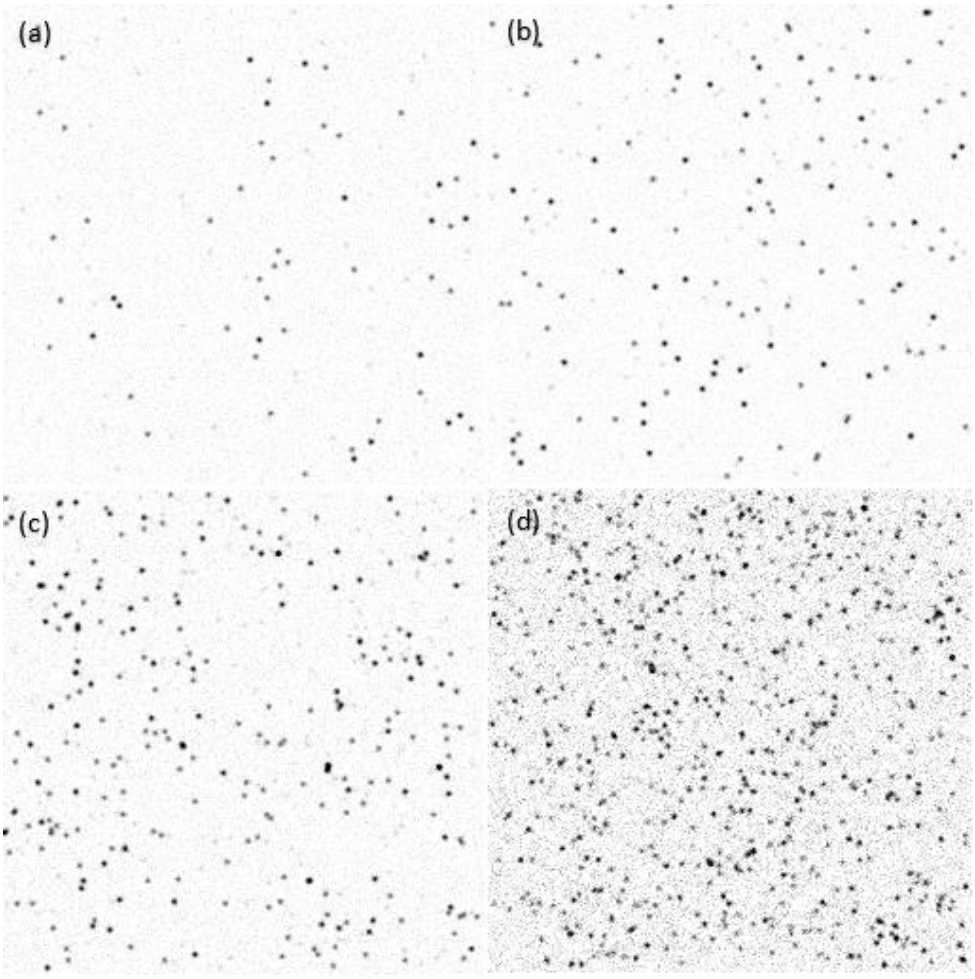


Figure 14 The 3D Synthetic Images Dataset in Experiment I. Slice 100 (of 256) from Simulated 3D Blob Images with Different Parameter Settings on the Number of Blobs and Signal-to-noise Ratio (*SNR*)(dB) (a) 3D Blob Image with N = 5,000 and SNR = 1dB, O = 0.04; (b) 3D Blob Image with N = 10,000 and SNR = 5dB, O = 0.07; (c) 3D Blob Image with N = 20,000 and SNR = 5dB, O = 0.14; and (d) 3D Blob Image with N = 50,000 and SNR = 1dB, O = 0.31.

Five methods were applied to the synthetic 3D blob images: the HDoG (M. Zhang et al., 2016), U-Net with standard thresholding (Falk et al., 2019), U-Net with optimal thresholding (OT U-Net) (Xu et al., 2019), the UH-DoG (Xu et al., 2020), and our proposed

BTCAS blob detector. The parameter settings of the DoG were as follows: window size N was 7. γ was 2. Δs was 0.001. To denoise the images of the 3D blobs using a trained U-Net, we first resized each 256×256 slice to 512×512 and each slice was fed into U-Net. We used the Adam optimizer in U-Net with a learning rate set to 0.0001. The dropout rate was set to 0.5. The threshold for the U-Net probability map in UH-DoG was set to 0.5. U-Net was implemented on a NVIDIA TITAN XP GPU with 12 GB of memory. Here we used a 2D U-Net and 2D probability maps were rendered on each slice then stacked together to form a 3D probability map.

3.3.3 Evaluating the Number of Blobs Detected

First, we compared the number of blobs detected from different algorithms and noisy image (Figure 15 and Figure 16) settings. The HDoG suffered from significant over-detection, yielding a high error rate in both experiments. In other methods, for the experiment on images with low noise, as the number of true blobs increased from 5,000 to 50,000, error rates for the U-Net, OT U-Net, and UH-DoG ranged from 4.96-38.78%, 4.28-32.22%, and 1.36-12.60% respectively. Our proposed BTCAS's error rates were significantly lower, ranging from 0.06-1.44%. For the experiment using images with high noise, as the number of true blobs increased from 5,000 to 50,000, error rates for the U-Net, OT U-Net, UH-DoG ranged from 4.68-39.87%, 4.08-32.96%, 1.38-12.79%. BTCAS had error rates of 0.08-10.20%. By integrating U-Net, the detection error decreased, and over-detection was reduced. However, both U-Net and OT U-Net detected fewer blobs than the ground truth. This can be explained by overlapping blobs; If the probability values at the boundaries of overlapping blobs are larger than the threshold, under-segmentation occurs, leading to fewer detected blobs. OT U-Net used Otsu's thresholding to find the

optimal threshold to somewhat reduce under-segmentation. With Hessian analysis, under-segmentation can be eliminated. The UH-DoG and BTCAS outperformed both U-Net and OT U-Net. The error rate of BTCAS slowly increased when the number of blobs increased from 5,000 to 50,000 with low noise and from 5,000 to 40,000 with high noise. Although the error rate of BTCAS increased when the number of blobs increased from 40,000 to 50,000 under high noise, this error rate was significantly lower than for UH-DoG. We conclude that BTCAS is much more robust in the presence of noise compared to the other four methods.

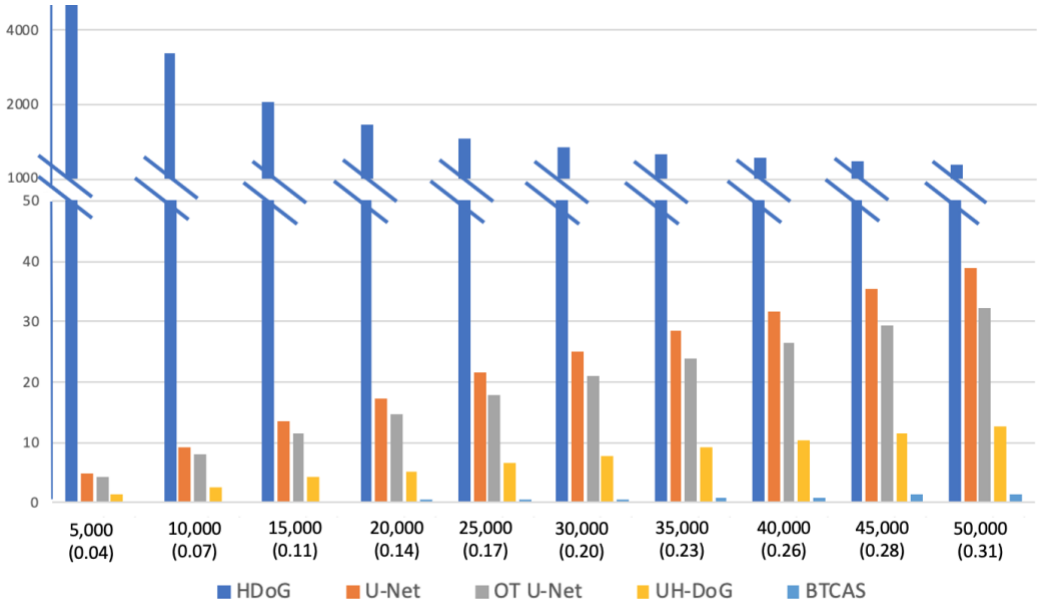


Figure 15 Comparison of Blob Detection Error Rate (%) of HDoG, U-Net, OT U-Net, UH-DoG and BTCAS in 3D Synthetic Blob Images with Low Noise (*SNR*=5dB). Number of True Blobs (Overlap Ratio) Ranges from 5,000 (0.04) to 50,000 (0.31).

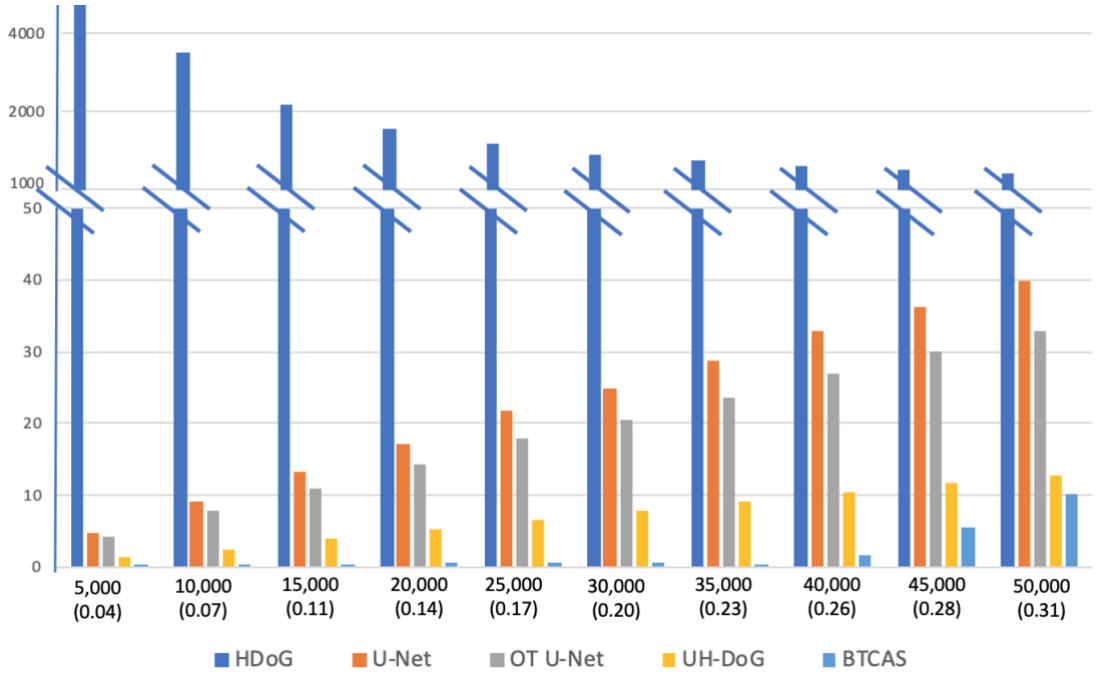


Figure 16 Comparison of Blob Detection Error Rate (%) of HDoG, U-Net, OT U-Net, UH-DoG and BTCAS in 3D Synthetic Blob Images with High Noise (*SNR*=1dB). Number of True Blobs (Overlap Ratio) Ranges from 5,000 (0.04) to 50,000 (0.31).

3.3.4 Evaluating Blob Detection and Segmentation Accuracy

Next, we evaluated algorithm performance by precision, recall, F-score, Dice coefficient, and Intersection over Union (IoU). For detection, Precision measures the fraction of retrieved candidates confirmed by the ground-truth. Recall measures the fraction of ground-truth data retrieved. F-score is an overall performance of precision and recall. For segmentation, the Dice coefficient measures the similarity between the segmented blob mask and the ground truth. IoU measures amount of overlap between the segmented blob mask and the ground truth. Ground truth voxels and blob locations (the coordinates of the blob centers) were already generated when synthesizing the 3D blob images. A candidate was considered as a true positive if the centroid of its magnitude was

in a detection pair (i, j) for which the nearest ground truth center j had not been paired and the Euclidian distance D_{ij} between ground truth center j and blob candidate i was less than or equal to d. To avoid duplicate counting, the number (#) of true positives TP was calculated by equation (3.19). Precision, recall, F-score were calculated by equations (3.20), (3.21), (3.22).

$$TP = \min\left\{\#\{(i,j): \min_{i=1}^{m} D_{ij} \le d\}, \#\{(i,j): \min_{j=1}^{n} D_{ij} \le d\}\right\}, \quad (3.19)$$

$$precision = \frac{TP}{n}, \quad (3.20)$$

$$recall = \frac{TP}{m}, \quad (3.21)$$

$$F - score = 2 \times \frac{precision \times recall}{(precision + recall)}, \quad (3.22)$$

where m is the number of true glomeruli and n is the number of blob candidates; d is a thresholding parameter set to a positive value $(0, +\infty)$. If d is small, fewer blob candidates are counted since the distance between the blob candidate centroid and ground-truth should be small. If d is too large, more blob candidates are counted. Here, since local intensity extremes could be anywhere within a small blob with an irregular shape, we set d to the average diameter of the blobs: $d = 2 \times \sqrt{\frac{\sum_{(x,y)} I(x,y;s)}{\pi}}$. The Dice coefficient and IoU were calculated by comparing the segmented blob mask and ground truth mask by equations (3.23) and (3.24).

$$Dice\ (B, G) = \frac{2|B \cap G|}{|B| + |G|}, \quad (3.23)$$

$$IoU\ (B, G) = \frac{B \cap G}{B \cup G}, \quad (3.24)$$

where B is the binary mask for segmentation result and G is the binary mask for the ground truth.

Comparisons between the models are shown in Table 8 and Table 9. ANOVA test was performed with Tukey's HSD multi-comparison at significance level 0.05. BTCAS significantly outperforms other four methods on Recall, F-Score for images with low and high noises. Compared to UH-DoG, BTCAS provides better performance on Recall, F-Score and is comparable on Precision, Dice and IoU. In this synthetic data, the blobs were generated with similar size ($s = 1$); we can thus still conclude that BTCAS can resolve under-segmentation by U-Net.

Table 8 Comparison (Avg ± Std) and ANOVA Using Tukey's HSD Pairwise Test of BTCAS, HDoG, UH-DoG, U-Net, OT U-Net on 3D Synthetic Images under *SNR* = 5db (Low Noise)

METRICS	BTCAS	HDOG	U-NET	OT U-NET	UH-DOG
PRECISION	1.00±0.00	0.10±0.07 (*< **0.0001**)	0.98±0.01 **(*<0.0001)**	1.00±0.00 **(*<0.0001)**	1.00±0.00 (0.172)
RECALL	0.99±0.00	0.99±0.01 (* **0.041**)	0.76±0.12 **(*< 0.001)**	0.81±0.09 **(*<0.0001)**	0.93±0.04 **(*<0.001)**
F-SCORE	1.00±0.00	0.18±0.11 **(*<0.0001)**	0.85±0.08 **(*< 0.001)**	0.89±0.06 **(*<0.001)**	0.96±0.02 **(*<0.001)**
DICE	0.96±0.03	0.26±0.14 **(*<0.0001)**	0.52±0.00 **(*< 0.0001)**	0.60±0.04 **(*<0.0001)**	0.97±0.02 **(*<0.0001)**
IOU	0.92±0.05	0.16±0.09 **(*<0.0001)**	0.35±0.00 **(*< 0.0001)**	0.43±0.04 **(*<0.0001)**	0.94±0.04 **(*<0.0001)**

*significance $\boldsymbol{p} < 0.05$

Table 9 Comparison (Avg ± Std) and ANOVA Using Tukey's HSD Pairwise Test of BTCAS, HDoG, UH-DoG, U-Net, OT U-Net on 3D Synthetic Images under *SNR* = 1db (High Noise)

METRICS	BTCAS	HDOG	U-NET	OT U-NET	UH-DOG

PRECISION	0.98±0.03	0.09±0.06 **(*< 0.0001)**	0.98±0.01 (0.338)	1.00±0.00 (0.063)	1.00±0.00 **(* 0.035)**
RECALL	0.99±0.00	0.99±0.01 **(* 0.026)**	0.76±0.12 **(*<0.001)**	0.81±0.10 **(*<0.001)**	0.93±0.04 **(*<0.001)**
F-SCORE	0.99±0.02	0.17±0.10 **(*<0.0001)**	0.85±0.08 **(*<0.001)**	0.89±0.06 **(*<0.0001)**	0.96±0.02 **(*<0.001)**
DICE	0.92±0.08	0.26±0.13 **(*<0.0001)**	0.51±0.01 **(*<0.0001)**	0.61±0.03 **(*<0.0001)**	0.94±0.04 (0.063)
IOU	0.85±0.13	0.15±0.09 **(*<0.0001)**	0.34±0.00 **(*<0.0001)**	0.44±0.03 **(*<0.0001)**	0.89±0.07 (0.061)

*significance $\boldsymbol{p} < 0.05$

3.3.5 Experiment II: Validation Using 3D Human Kidney CFE-MR Images

In this experiment we investigated blob segmentation applied to 3D CFE-MR images to measure number (N_{glom}) and apparent volume (aV_{glom}) of glomeruli in healthy and diseased human donor kidneys that were not accepted for transplant. Three human kidneys were obtained at autopsy through a donor network (The International Institute for the Advancement of Medicine, Edison, NJ) after receiving Institutional Review Board (IRB) approval and informed consent from Arizona State University (Beeman et al., 2014). They were imaged by CFE-MRI as described in (Baldelomar et al., 2016a, 2017; Beeman et al., 2014; Bennett et al., 2008a).

Each human MR image has pixel dimensions of 896×512×512. We applied the HDoG, UH-DoG and proposed BTCAS blob detector to segment glomeruli. The parameter settings of DoG are as follows: window size $N = 7$. $\gamma = 2$. $\Delta s = 0.001$. We first generated 14,336 2D patches, with each patch 128×128 in size and each patch was then fed into U-Net. The threshold for the U-Net probability map in UH-DoG was 0.5. We performed quality control by visually checking the identified glomeruli, visible as black spots in the images. For illustration, example results from CF2 which has more heterogenous pattern are shown in Figure 17. As seen, the BTCAS blob detector performed better than the HDoG

and the UH-DoG in segmentation. Several example glomeruli are marked with orange, green and blue circles. In Figure 17 (e-h), orange circles show that some noise is detected as false positives by the HDoG and UH-DoG, but the BTCAS blob detector performed well using the denoising provided by U-Net. Green circles show some blobs that are under-segmented in UH-DoG due to the fixed probability threshold. The BTCAS blob detector captured these. Blue circles show that for blobs with a range of sizes, the BTCAS blob detector delineated all voxels of blobs with the adaptive optimum DoG scale.

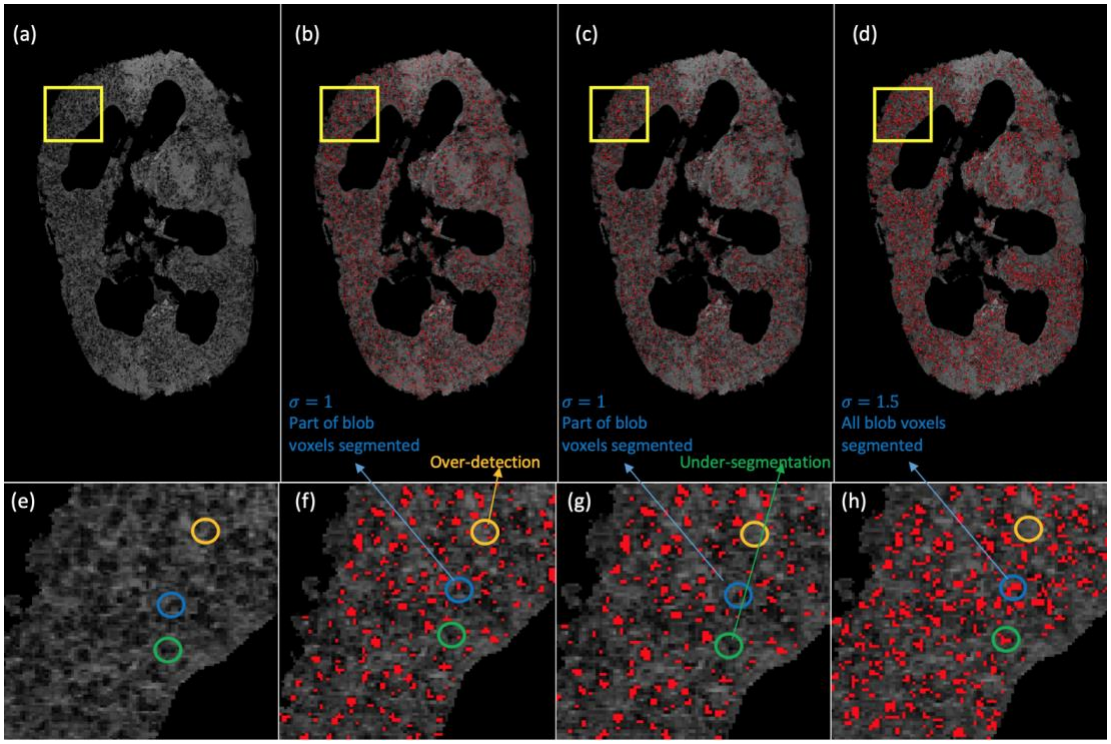


Figure 17 Glomerular Segmentation Results from 3D MR Images of Human Kidney (CF2 Slice 256). (a) Original Magnitude Image. (b) Glomerular Segmentation Results of HDoG. (c) Glomerular Segmentation Results of UH-DoG. (d) Glomerular Segmentation Results of BTCAS Blob Detector. (e-h) Magnified Regions (Yellow Box) from (a-d).

N_{glom} and aV_{glom} are reported in Table 10 and Table 11, where the HDoG, UH-DoG and proposed BTCAS blob detector are compared to data from unbiased dissector-fractionator stereology. representing a ground truth in the average measurements in each kidney. We used the stereology data from (Beeman et al., 2014) and the method of calculating aV_{glom} from (Baldelomar et al., 2016a). The differences between the results of the HDoG, UH-DoG, BTCAS methods and stereology data are also listed in Table 10 and Table 11. Compared to stereology, the HDoG identified more glomeruli and the difference with stereology is much larger than the other two methods, indicating over-detection under the single optimal scale of DoG and lower mean aV_{glom} than stereology. UH-DoG identified fewer glomeruli due to under-segmentation when using the single thresholding (0.5) on the probability map of U-Net combined with the Hessian convexity map. BTCAS provided the most accurate measurements of N_{glom} and mean aV_{glom} than the other two methods.

Table 10 Human Kidney Glomerular Segmentation (N_{glom}) from CFE-MRI Using HDoG, UH-DoG and the Proposed BTCAS Blob Detectors Compared to Dissector-fractionator Stereology

Human Kidney	Nglom ($\times 10^6$) (Stereology)	Nglom ($\times 10^6$) (BTCAS)	Difference Ratio (%)	Nglom ($\times 10^6$) (UH-DoG)	Difference Ratio (%)	Nglom ($\times 10^6$) (HDoG)	Difference Ratio (%)
CF 1	1.13	1.16	2.65	0.66	41.60	2.95	>100
CF 2	0.74	0.86	16.22	0.48	35.14	1.21	63.51
CF 3	1.46	1.50	2.74	0.85	41.78	3.93	>100

Table 11 Human Kidney Glomerular Segmentation (Mean aV_{glom}) from CFE-MRI using HDoG, UH-DoG and the Proposed BTCAS Blob Detectors Compared to Dissector-fractionator Stereology

Human Kidney	Mean aV_{glom}($\times 10^{-3}mm^3$) (Stereology)	Mean aVglom ($\times 10^{-3}mm^3$) (BTCAS)	Difference Ratio (%)	Mean aVglom($\times 10^{-3}mm^3$) (UH-DoG)	Difference Ratio (%)	Mean aVglom($\times 10^{-3}mm^3$) (HDoG)	Difference Ratio (%)
CF 1	5.01	5.32	**6.19**	7.36	**46.91**	4.8	**4.19**
CF 2	4.68	4.78	**2.14**	5.62	**20.09**	3.2	**31.62**
CF 3	2.82	2.55	**9.57**	3.73	**32.37**	3.2	**13.48**

3.3.6 Experiment III: Validation using 3D Mouse Kidney CFE-MR Images

We conducted experiments on CF-labeled glomeruli from a dataset of 3D MR images to measure N_{glom} and aV_{glom} of glomeruli in healthy and diseased mouse kidneys. This dataset includes chronic kidney disease (CKD, n=3) vs. controls (n=6), acute kidney injury (AKI, n=4) vs. control (n=5). The animal experiments were approved by the Institutional Animal Care and Use Committee (IACUC) under protocol #3929 on 04/07/2020 at the University of Virginia, in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals. They were imaged by CFE-MRI as described in (Charlton et al., 2020).

Each MRI image has pixel dimensions of 256×256×256. We applied the HDoG, HDoG with VBGMM, UH-DoG and proposed BTCAS blob detector to segment glomeruli. The parameter settings of DoG were: window size N = 7. γ = 2. Δs = 0.001. To denoise the 3D blob images by using trained U-Net, we first resized each slice to 512×512 and each slice was fed into U-Net. The threshold for the U-Net probability map in UH-DoG was 0.5.

N_{glom} and mean aV_{glom} are reported in Table 12 and Table 13, where the HDoG, UH-DoG and proposed BTCAS blob detector are compared to HDoG with VBGMM from

(M. Zhang et al., 2016). The differences between the results are also listed in Table 12 and Table 13. Compared to HDoG with VBGMM, the HDoG identified more glomeruli and the difference with HDoG with VBGMM is much larger than for the other two methods, indicating over-detection under the single optimal scale of the DoG and lower mean aV_{glom} than HDoG with VBGMM. UH-DoG identified fewer glomeruli and larger mean aV_{glom} due to under-segmentation when using the single thresholding (0.5) on the probability map of U-Net combined with the Hessian convexity map. BTCAS provided the most accurate measurements of N_{glom} and mean aV_{glom} compared to the other two methods.

Table 12 Mouse Kidney Glomerular Segmentation (N_{glom}) from CFE-MRI using HDoG, UH-DoG and the Proposed BTCAS Compared to HDoG with VBGMM Method

Mouse kidney		N_{glom} **(HDoG with VBGMM)**	N_{glom} **(BTCAS)**	**Difference Ratio (%)**	N_{glom} **(UH-DoG)**	**Difference Ratio (%)**	N_{glom} **(HDoG)**	**Difference Ratio (%)**
CKD	ID 429	7,656	7,719	**0.82**	7,346	**4.05**	10,923	**42.67**
	ID 466	8,665	8,228	**5.04**	8,138	**6.08**	9,512	**9.77**
	ID 467	8,549	8,595	**0.54**	8,663	**1.33**	12,755	**49.20**
	Avg	8,290	8,181	**2.13**	8,049	**2.91**	11,063	**33.88**
	Std	552	440		663		1626	
Control for CKD	ID 427	12,724	12,008	**5.63**	12,701	**0.18**	15,515	**21.93**
	ID 469	10,829	11,048	**2.02**	11,347	**4.78**	15,698	**44.96**
	ID 470	10,704	10,969	**2.48**	11,309	**5.65**	13,559	**26.67**
	ID 471	11,943	12,058	**0.96**	12,279	**2.81**	16,230	**35.90**
	ID 472	12,569	13,418	**6.75**	12,526	**0.34**	17,174	**36.64**
	ID 473	12,245	12,318	**0.60**	11,853	**3.20**	15,350	**25.36**
	Avg	11,836	11,970	**3.07**	12,003	**1.41**	15,588	**31.91**
	Std	872	903		595		1193	
AKI	ID 433	11,046	10,752	**2.66**	11,033	**0.12**	12,315	**11.49**
	ID 462	11,292	10,646	**5.72**	10,779	**4.54**	17,634	**56.16**
	ID 463	11,542	11,820	**2.41**	10,873	**5.80**	20,458	**77.25**
	ID 464	11,906	12,422	**4.33**	11,340	**4.75**	25,233	**>100**
	Avg	11,447	11,410	**3.78**	11,006	**3.85**	18,910	**64.21**
	Std	367	858		246		5401	
Control for AKI	ID 465	10,336	10,393	**0.55**	10,115	**2.14**	13,473	**30.35**
	ID 474	10,874	11,034	**1.47**	11,157	**2.60**	16,934	**55.73**
	ID 475	10,292	9,985	**2.98**	10,132	**1.55**	12,095	**17.52**
	ID 476	10,954	11,567	**5.60**	10,892	**0.57**	15,846	**44.66**
	ID 477	10,885	11,143	**2.37**	11,335	**4.13**	14,455	**32.80**
	Avg	10,668	10,824	**2.59**	10,726	**0.54**	14,561	**36.21**
	Std	325	630		572		1908	

Table 13 Mouse Kidney Glomerular Segmentation (Mean aV_{glom}) from CFE-MRI using HDoG, UH-DoG and the Proposed BTCAS Compared to HDoG with VBGMM Method

Mouse kidney		Mean aV_{glom}* (HDoG with VBGMM)	Mean aV_{glom}* (BTCAS)	Difference Ratio (%)	Mean aV_{glom}* (UH-DoG)	Difference Ratio (%)	Mean aV_{glom}* (HDoG)	Difference Ratio (%)
CKD	ID 429	2.57	2.63	**2.33**	2.92	**11.99**	2.46	**4.28**
	ID 466	2.01	2.01	**0.00**	2.06	**2.43**	1.75	**12.94**
	ID 467	2.16	2.20	**1.85**	2.32	**6.90**	1.9	**12.04**
	Avg	2.25	2.28	**1.40**	2.43	**7.67**	2.04	**9.75**
	Std	0.29	0.32		0.44		0.37	
Control for CKD	ID 427	1.49	1.57	**5.37**	1.61	**7.45**	1.49	**0.00**
	ID 469	1.91	1.95	**2.09**	2.20	**13.18**	1.76	**7.85**
	ID 470	1.98	2.05	**3.54**	2.04	**2.94**	1.73	**12.63**
	ID 471	1.5	1.58	**5.33**	1.56	**3.85**	1.4	**6.67**
	ID 472	1.35	1.36	**0.74**	1.49	**9.40**	1.35	**0.00**
	ID 473	1.5	1.56	**4.00**	1.58	**5.06**	1.39	**7.33**
	Avg	1.62	1.68	**3.51**	1.75	**7.16**	1.52	**5.75**
	Std	0.26	0.26		0.30		0.18	
AKI	ID 433	1.53	1.64	**7.19**	1.63	**6.13**	1.38	**9.80**
	ID 462	1.34	1.41	**5.22**	1.48	**9.46**	1.3	**2.99**
	ID 463	2.35	2.4	**2.13**	2.61	**9.96**	1.94	**17.45**
	ID 464	2.31	2.36	**2.16**	2.40	**3.75**	1.78	**22.94**
	Avg	1.88	1.95	**4.18**	2.03	**7.27**	1.60	**13.29**
	Std	0.52	0.50		0.56		0.31	
Control for AKI	ID 465	2.3	2.46	**6.96**	2.40	**4.17**	2.11	**8.26**
	ID 474	2.44	2.34	**4.10**	2.52	**3.17**	2.14	**12.30**
	ID 475	1.74	1.86	**6.90**	1.70	**2.35**	1.58	**9.20**
	ID 476	1.53	1.57	**2.61**	1.62	**5.56**	1.49	**2.61**
	ID 477	1.67	1.68	**0.60**	1.70	**1.76**	1.61	**3.59**
	Avg	1.94	1.98	**4.23**	1.99	**2.62**	1.79	**7.19**
	Std	0.41	0.40		0.43		0.31	

*aV_{glom} unit $\boldsymbol{mm^3}$x $\boldsymbol{10^{-4}}$

Table 14 Comparison of Computation Time Between DoG under Glomerulus-specific Optimal Scale and Proposed BTCAS Method

Human Kidney	DoG under glomerulus-specific optimal scale (second)	BTCAS (second)	Difference Ratio (%)
CF 1	51,238	34,792	32.10
CF 2	39,616	28,156	28.93
CF 3	59,703	41,425	30.61
AVG ± STD	50,186 ± 10,085	34,791 ± 6,635	30.55 ± 1.59

3.3.7 Discussion of Computation Time

Our proposed method uses U-Net for pre-processing, followed by the DoG where the scales vary depending on sizes of the glomeruli. The computational time of U-Net is satisfactory. For example, it takes < 5 minutes for training and < 1 second per slice or per patch for testing. Therefore, we focus on the discussion of computation efforts related to the DoG implementation. Given a 3D image in $N_1 \times N_2 \times N_3$ and a convolutional filtering kernel size as $r_1 \times r_2 \times r_3$, the computational complexity of HDoG (M. Zhang et al., 2016) is $O(\, N_1\, N_2\, N_3(\, r_1 + r_2 + r_3))$. Considering our proposed method BTCAS, let N_S be the number of scales searched ($N_S > 1$), the computational complexity is $O(\, N_S\, N_1\, N_2\, N_3(\, r_1 + r_2 + r_3))$. We conclude that BTCAS requires more computing effort comparing to HDoG (M. Zhang et al., 2016) since $N_S > 1$. Yet, HDoG (M. Zhang et al., 2016), the single scale approach suffers from performances as shown in the comparison experiments (See Figure 17 in Section 3.3.5, Tables 8-13 in Sections 3.3.4, 3.3.5, 3.3.6). Exhaustively searching optimal scale for each glomerulus however is computational prohibitive. Table 14 Comparison of Computation Time Between DoG under Glomerulus-specific Optimal Scale and Proposed BTCAS Methodsummarizes the computational time for DoG under exhaustive search on scales (note the scale ranges [0, 1.5]) using stereology knowledge) for each glomerulus and that for BTCAS. As seen, BTCAS saves about 30% computing time.

3.4 Conclusion

In this research, we develop a new small blob detector (BTCAS). This work provides three main contributions to the literature. First, U-Net reduces over-detection

when it is used in the initial denoising step. This results in a probability map with the identified centroid of blob candidates. Second, distance maps are rendered with lower and upper probability bounds, which are used as the constraints for local scale search for the DoG. Third, a local optimum DoG scale is adapted to the range of blob sizes to better separate touching blobs. In two experiments, this adaptive scale based on deep learning greatly decreases under-segmentation by U-Net with over 80% increase in Dice and IoU and decreases over-detection by DoG with over 100% decrease in error rate of blob detection.

While the results of this study are encouraging, there is room for improvement. First, the proposed method consists of two sequential steps, where adaptive DoG and Hessian analysis are based on the probability and distance maps predicted from U-Net. This approach may not be computationally efficient. It is our intention to integrate the DoG and Hessian analysis as layers of in the overall deep learning network for comprehensive glomerular segmentation. Second, we use 2D U-Net instead of 3D U-Net to perform on 3D images, so each slice is processed independently. The performance might be different under a 3D U-Net. We also plan to explore semi-supervised learning by incorporating domain knowledge of glomeruli to further improve glomerular detection and segmentation. All the future work is built upon the BTCAS which we believe has shown to be an adaptive and effective tuning-free detector for blob detection and segmentation and has the potential for kidney biomarker identification for clinical use.

CHAPTER 4

BLOBGAN: DENOISING CONVEXITY-CONSISTENT GENERATIVE ADVERSARIAL NETWORK FOR SMALL BLOB IDENTIFICATION

Medical imaging tools are becoming increasingly important in the clinic. Biomarkers extracted from these images inform disease diagnosis, prognosis, and plans for treatment. Identifying (detecting and segmenting) objects from medical images is often the first step in the quantitative measurement of these biomarkers. Magnetic resonance imaging (MRI) is advantageous for detecting small structures in soft tissue, which are represented by small blobs in the image (e.g., kidney glomeruli), but other medical imaging modalities also reveal small blob (e.g., cell nuclei in histopathology images or fluoroscopic images). However, there are critical challenges to identifying small blobs including image noise and low image resolution relative to the distance between the physical structures that are represented by the blobs. To address these challenges, we propose a BlobGAN (Blob Generative Adversarial Network) to detect and segment blobs. First, we propose a 3D elliptical Gaussian function to synthesize 3D blobs that can approximate the blobs of interest. The synthesized 3D images containing blobs and their masks are then employed as training input to a convexity- consistent 3D GAN to preserve the geometric properties of the blobs. Given the location, shape, and data distribution of blobs as prior, BlobGAN iteratively learns the noise distribution from target noisy images containing blobs, and simultaneously de-noises the images. Subsequently, two types of masks are derived from the denoised images: Hessian convexity masks containing voxel-based convexity information and blob masks containing the location information. Finally, we apply a joint constraint operation for

blob identification to address over-segmentation from the Hessian convexity mask and under-segmentation from the blob mask. To validate the performance of the proposed BlobGAN, a 3D simulated dataset (n=1,000) of blobs, a 3D MRI dataset of human kidneys, and a 3D MRI dataset of mouse kidneys are studied. BlobGAN is compared against six state-of-the-art methods: HDoG, U-Net with standard thresholding, U-Net with optimal thresholding, UH-DoG, BTCAS, CycleGAN using detection error rate, precision, recall, F-score, Dice, IoU, and blobness. We conclude that BlobGAN statistically outperforms the compared detectors.

4.1 Introduction

Imaging technologies are being increasingly integrated into clinical practice. There is a growing need to advance the field of medical image quantification for accurate diagnosis of the underlying health conditions and determining the optimal treatment strategies. Quantification of medical images often begins with identification (detection and segmentation) of the pertinent region of interest (ROI). These ROIs range from large tumorous regions in the breast and brain mammograms to nanoparticle - labeled glomeruli in magnetic resonance (MR) images of the kidney. Several supervised and semi-supervised approaches have emerged to identify large ROIs (Samala et al., 2019; Shin et al., 2019). However, the identification of small blobs has received very little attention. Small blobs are easily confused with image artifacts, and traditional object detection and segmentation methods are insufficient. Additionally, low image resolution relative to the size of the blobs can complicate their identification.

To overcome the challenges of blob identification, scale-space blob detectors such as Laplacian of Gaussian (LoG) (Koenderink, 1984; Witkin, 1983) and Difference of

Gaussian (DoG) (Lowe, 2004b; Mikolajczyk & Schmid, 2004b, 2005; Tuytelaars & Mikolajczyk, 2007) have attracted considerable attention. For example, Kong et al. proposed a generalized Laplacian of Gaussian (gLoG) (Kong et al., 2013b) to detect the centroid of nuclei and estimate their sizes and orientations from histologic and fluorescent microscopic images. Zhang et al. developed a series of blob detectors such as the Hessian-based Laplacian of Gaussian (HLoG) (M. Zhang et al., 2015) and Hessian-based Difference of Gaussian (HDoG) (M. Zhang et al., 2016), based on Hessian analysis, to automatically detect glomeruli in cationic ferritin enhanced MRI (CFE-MRI). Recently, deep learning has been investigated for biomarker discovery from medical images (Chang et al., 2021; Dolz et al., 2019; Gupta et al., 2018; W. Li et al., 2019; Tan et al., 2020; Zreik et al., 2019). However, this approach has been limited to detecting and segmenting large ROIs such as organs and tumors; its performance in small blob segmentation is uncertain. Deep learning models are inefficient at segmenting overlapping blobs, leading to under-segmentation and a high false negative rate (Xu et al., 2019). Deep learning also requires large, annotated datasets. Generating annotations for blobs, such as glomeruli in MR images of the kidney, is difficult because each image may contain hundreds of blobs, as shown in Figure 23. Recently, UH-DoG was proposed that used a pre-trained U-Net to denoise CFE-MRI images to derive a probability map of glomeruli. Joining the probability map with a Hessian convexity map, true glomeruli were identified (Xu et al., 2020). UH-DoG is a single-threshold, single-scale approach. Such an approach may not be sensitive to noise and may overlook large variations in blob size. Therefore, UH-DoG was later extended to a multi-threshold, multi-scale approach called Bi-Threshold Constrained Adaptive Scales (BTCAS) (Xu et al., 2021) to uncover the relationship between the U-Net threshold and

DoG scale. By adopting BTCAS, the computational efficiency and accuracy of segmentation were both improved. However, these deep learning models were pre-trained on publicly available optical images of cell nuclei (Janowczyk & Madabhushi, 2016a). While these approaches detect blobs to some extent, there are several drawbacks to training deep learning models on public datasets. First, public datasets are from different imaging modalities and therefore have a different distribution of intensities than the target dataset. Consequently, pre-trained models require additional ad-hoc approaches for fine-tuning the model parameters (Tajbakhsh et al., 2016). Second, small blobs created by structures such as cells, nuclei, and glomeruli have distinct geometric properties. If public and target datasets have different geometric properties, pre-trained models may not be accurate. Third, public datasets have different noise distributions compared to the target datasets, making it challenging to filter noise during segmentation. These limitations of training deep learning models hinder its application to optimally denoise blob images.

Recently, another class of deep learning models called Generative Adversarial Networks (GANs) (Goodfellow et al., 2020) has shown great potential for denoising images without the need for annotations. The methodology, known as CycleGAN, is based on the framework of image-to-image translation (J.-Y. Zhu et al., 2017). GAN consists of two sub-networks: a generator and a discriminator. A generator is trained to synthesize fake versions of the original images, and a discriminator is employed to distinguish the fake images from the real ones. The generator and discriminator networks train together iteratively, resulting in fake images resembling the real images. CycleGAN is a generalization of GAN to translate image characteristics from one domain to another by simultaneously training two pairs of generators and discriminators, one for each domain.

By considering noisy and denoised images as the two domains, image denoising can be performed using CycleGAN. Gu et al. (Gu & Ye, 2021) proposed a tunable CycleGAN with adaptive instance normalization to denoise low-dose CT images. Zhou et al. (Zhou et al., 2020) presented a supervised CycleGAN to denoise low-dose PET images. However, CycleGAN has some limitations. Zhang et al. (Z. Zhang et al., 2018) have shown that CycleGAN may geometrically distort image features. Gu et al. (Gu et al., 2021) showed that geometric distortion can destroy anatomical details in denoised images. These local anatomical structures can be critical in clinical decision-making. To alleviate geometric distortions, Gu et al. (Gu et al., 2021) used a wavelet transform to extract high-frequency signals that contain the noise, supporting CycleGAN in denoising cardiac CT images. Guo et al. (Guo et al., 2019) introduced structure-aware loss to constrain CycleGAN in denoising images of retinal pathologies. These previous studies focused on denoising and preventing distortion of images containing large objects. However, this might not be effective for denoising blob images consisting of small blobs and background noise, in which geometric distortion occurs because the blobs and noise have similar size, shape, and distributions of intensity. In this work, we apply CycleGAN for image denoising and propose a convexity consistent constraint to preserve the geometric properties of blobs. Given the location, shape, and intensity distribution of the blobs as prior, the adversarial model can learn the noise distribution and identify the blobs with minimal geometric distortion.

Here we apply this new approach to the problem of identifying glomeruli in CFE-MR images of the kidney. Glomeruli are roughly spherical and have a blob-like shape in 3D space. There are several challenges. First, the glomeruli are small relative to the

imaging resolution and have a similar visual appearance as the noise. Second, there are over a million glomeruli in the cortex of the human kidney, and the image intensity distribution is heterogeneous. Third, a significant proportion of glomeruli overlap due to the low image resolution, making it difficult to identify them as individual blobs. Note one can argue the image resolution in MRI can be adjusted, it comes with the expenses of long acquisiton time. We propose a 3D BlobGAN model to address these challenges for glomeruli detection and segmentation. The overall architecture of 3D BlobGAN is motivated by CycleGAN (J.-Y. Zhu et al., 2017), but with three key innovations. First, we propose a 3D elliptical Gaussian function to define the blobs, as a model of the glomeruli. The randomly generated, 3D images of the blobs, and the corresponding masks, comprise the training data, eliminating the need for public datasets. Second, we propose a convexity consistent constraint to preserve blobs' geometric properties, location, and intensity distribution. Because the GAN-translated images mimic the images of glomeruli, the BlobGAN essentially learns the noise distribution. The two generators then form a cycle for adding and removing noise. Finally, our model is able to denoise glomerular images and derive the Hessian convexity mask for blob identification. To validate the performance of BlobGAN, we perform three case studies. The first case study involves the segmentation of blobs in a 3D simulated blob image dataset (n=1,000) where the locations of blobs are known. We choose six state-of-the-art methods from the literature for comparison: HDoG (M. Zhang et al., 2016), U-Net with standard thresholding (Falk et al., 2019), U-Net with optimal thresholding (Xu et al., 2019), UH-DoG (Xu et al., 2020), BTCAS (Xu et al., 2021), and CycleGAN (J.-Y. Zhu et al., 2017). In the next two case studies, we implement

the BlobGAN method on 3D human kidney images and 3D mouse kidney images obtained from CFE-MRI and compare against the HDoG, UH-DoG, BTCAS, and stereology.

The remainder of the paper is organized as follows: Section 4.2 describes our proposed BlobGAN in detail. Section 4.3 demonstrates the comparative results on the 3D synthetic images, 3D human kidney images, and 3D mouse kidney images. Finally, the conclusions are presented in Section 4.4.

4.2 Methods

Our proposed BlobGAN (Figure 18) consists of three steps for the identification of glomeruli from kidney CFE-MRI images: (1) synthesis of 3D blobs using 3D elliptical Gaussian function and random generation of 3D blob images with their blob masks as training input; (2) training CycleGAN with convexity consistency constraint to denoise the noisy blob images (kidney CFE-MRI); and (3) applying joint constraint operation on the blob mask and the Hessian convexity mask obtained from step 2 to derive the final identification mask of blobs.

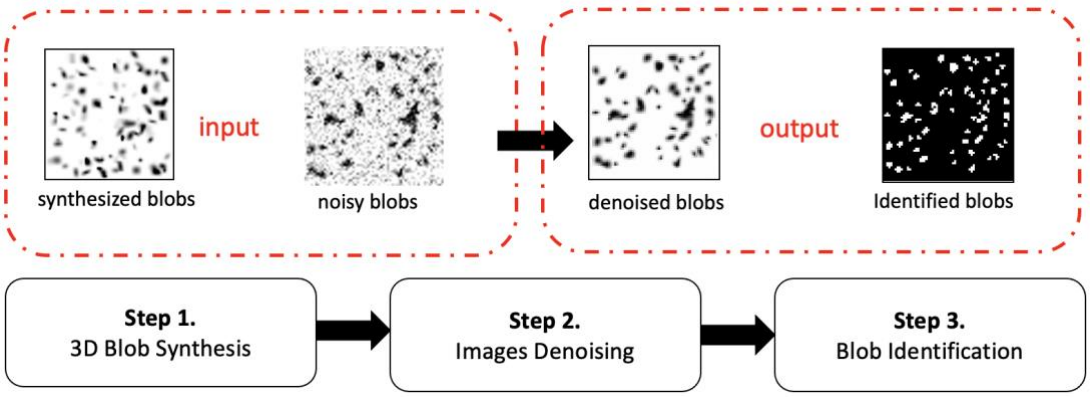


Figure 18 The Overview of the Proposed BlobGAN Model for Glomerular Identification.

4.2.1 3D Blob Synthesis using 3D Elliptical Gaussian Function

In the existing literature, Gaussian functions have been employed to synthesize blobs. For instance, Wang et al.(Wang et al., 2017) proposed a second-order Gaussian function for constructing small blobs with a fixed diameter. Kong et al. (Kong et al., 2013b) introduced additional shape and orientation parameters into the Gaussian function as a kernel of the blob detector. However, these approaches are only suited for 2D blob synthesis. To the best of our knowledge, there is no systematic approach for 3D blob synthesis. In the 3D space, the shape and orientation of blobs are more complex. In this work, the 3D blobs of interest are the glomeruli from kidney CFE-MRI. So, we focus on synthesizing 3D blobs that have similar geometric properties as the glomeruli. To this end, we propose a 3D elliptical Gaussian function with the following general form (see ***Proposition 1*** in the Appendix B):

$$F(x,y,z) = \mathcal{A} \cdot e^{-(a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 + d(x-x_0)(y-y_0) + e(y-y_0)(z-z_0) + f(x-x_0)(z-z_0))}, \quad (4.1)$$

where $\mathcal{A}$ is a normalization factor, and x_0, y_0 and z_0 are the coordinates of the center of the Gaussian function $F(x,y,z)$. The coefficients a, b, c, d, e and f control the shape and orientation of $F(x,y,z)$ via θ, φ, σ_x, σ_y and σ_z. as given in the following:

$$a = \frac{sin^2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin^2\theta sin^2\varphi}{\sigma_y^2} + \frac{cos^2\theta}{\sigma_z^2}$$

$$b = \frac{cos^2\theta cos^2\varphi}{\sigma_x^2} + \frac{cos^2\theta sin^2\varphi}{\sigma_y^2} + \frac{sin^2\theta}{\sigma_z^2}$$

$$c = \frac{sin^2\varphi}{\sigma_x^2} + \frac{cos^2\varphi}{\sigma_y^2}$$

$$d = \frac{sin2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin2\theta sin^2\varphi}{\sigma_y^2} - \frac{sin2\theta}{\sigma_z^2}$$

$$e = -\frac{cos\theta sin2\varphi}{\sigma_x^2} + \frac{cos\theta sin2\varphi}{\sigma_y^2}$$

$$f = -\frac{sin\theta sin2\varphi}{\sigma_x^2} + \frac{sin\theta sin2\varphi}{\sigma_y^2}$$

(4.2)

Without any loss of generality, we let x_0, y_0 and z_0 equal to zero in our blob synthesis model. Therefore, the 3D elliptical Gaussian function is simplified as:

$$F(x, y, z) = \mathcal{A} \cdot e^{-(ax^2 + by^2 + cz^2 + dxy + eyz + fxz)} \quad (4.3)$$

To diversify the blobs, parameters θ, φ, σ_x, σ_y and σ_z are randomly selected during the blob synthesis. Figure 19 illustrates the synthesized 3D blobs for different parameter combinations.

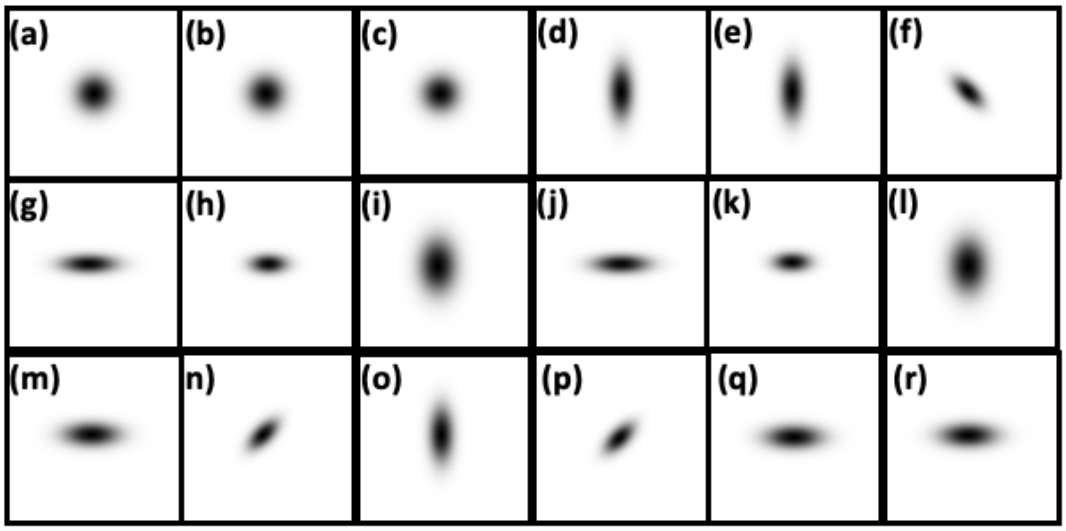


Figure 19 Synthesized 3D Blobs using the Proposed 3D Elliptical Gaussian Function with Blob Parameters θ, φ, σ_x, σ_y and σ_z. All Images are Shown as 2D Slices (Slice 50 out of 99) for All Planes. (a-c) XY, XZ, YZ 2D Planes of 3D Blob with $\theta = 0°$, $\varphi = 0°$, $\sigma_x = 10$, $\sigma_y = 10$, $\sigma_z = 10$. (d-f) XY, XZ, YZ 2D Planes of 3D Blob with $\theta = 0°$, $\varphi = 45°$, $\sigma_x = 5$, $\sigma_y = 10$, $\sigma_z = 15$. (g-i) XY, XZ, YZ 2D Planes of 3D Blob with $\theta = 90°$, $\varphi =$

0°, $\sigma_x = 5$, $\sigma_y = 10$, $\sigma_z = 15$. (j-l) XY, XZ, YZ 2D Planes of 3D Blob with $\theta = 45°$, $\varphi = 90°$, $\sigma_x = 15$, $\sigma_y = 10$, $\sigma_z = 5$. (m-o) XY, XZ, YZ 2D Planes of 3D Blob with $\theta = 90°$, $\varphi = 45°$, $\sigma_x = 10$, $\sigma_y = 5$, $\sigma_z = 15$. (p-r) XY, XZ, YZ 2D Planes of 3D Blob with $\theta = 45°$, $\varphi = 90°$, $\sigma_x = 15$, $\sigma_y = 5$, $\sigma_z = 10$.

Using the 3D elliptical Gaussian function, we generate training images containing blobs and their corresponding masks. Within each image, the number of blobs, the individual blob parameters, and the location of blobs are randomly chosen. Using BlobGAN, we translate the synthesized blob images (without noise) to mimic the real glomeruli images and then translate them back to the original synthesized blob domain (without noise). The goal of synthesizing 3D blobs is to approximate the distribution of glomeruli so that only the context in images is added and removed in this translation cycle. Because the context (background) of glomeruli images is noise, BlobGAN is essentially an approach of noisy to denoised domain translation. By doing so, glomeruli images are denoised. We will discuss the detailed approach in the next section.

4.2.2 3D Blob Image Denoising through 3D GAN with Convexity Consistency Constraint

BlobGAN adopts the domain translation architecture from CycleGAN (J.-Y. Zhu et al., 2017). In the following, we use CycleGAN to illustrate our model. Given a source domain S and a target domain T, the goal of CycleGAN is to train a generator G_{ST} to learn the mapping from S to T and another generator G_{TS} to learn the mapping from T to S iteratively until the generated images approximate the real images from the target domain. In each iteration, we also maintain a pair of discriminators D_S and D_T to distinguish the real images in domains S and T from the synthesized images obtained from $G_{TS}(T)$ and

$G_{ST}(S)$, respectively. With training samples of clean synthetic blob images $\{I_c\} \in S$ and real noisy blob images $\{I_n\} \in T$, we denote their respective data distributions as $I_c \sim p_{data}(I_c)$ and $I_n \sim p_{data}(I_n)$ such that the adversarial loss for each generator-discriminator pair can be written as:

$$\mathcal{L}_{GAN}(G_{ST}, D_T) = \mathrm{E}_{I_n \sim p_{data}(I_n)}[\log D_T(I_n)] + \mathrm{E}_{I_c \sim p_{data}(I_c)}[\log(1 - D_T(G_{ST}(I_c)))] \quad (4.4)$$

$$\mathcal{L}_{GAN}(G_{TS}, D_S) = \mathrm{E}_{I_c \sim p_{data}(I_c)}[\log D_S(I_c)] + \mathrm{E}_{I_n \sim p_{data}(I_n)}[\log(1 - D_S(G_{TS}(I_n)))] \quad (4.5)$$

To ensure that the translated images can be transferred back to the domain they are generated from, CycleGAN (J.-Y. Zhu et al., 2017) uses an effective cycle consistency constraint, i.e., $G_{TS}(G_{ST}(I_c)) \approx I_c$ and $G_{ST}(G_{TS}(I_n)) \approx I_n$ and the cycle consistency loss can be written using L1 loss over all the image voxels:

$$\mathcal{L}_{cycle}(G_{ST}, G_{TS}) = \mathrm{E}_{I_c \sim p_{data}(I_c)}[\|G_{TS}(G_{ST}(I_c)) - I_c\|_1] + \mathrm{E}_{I_n \sim p_{data}(I_n)}[\|G_{ST}(G_{TS}(I_n)) - I_n\|_1] \quad (4.6)$$

Besides the cycle consistency loss, we also employ an identity loss function to constrain the generator to provide the identity mapping when images from the target domain are provided as input. Here, the identity loss can be written using L1 loss as:

$$\mathcal{L}_{identity}(G_{ST}, G_{TS}) = \mathrm{E}_{I_n \sim p_{data}(I_n)}[\|G_{ST}(I_n) - I_n\|_1] + \mathrm{E}_{I_c \sim p_{data}(I_c)}[\|G_{TS}(I_c) - I_c\|_1] \quad (4.7)$$

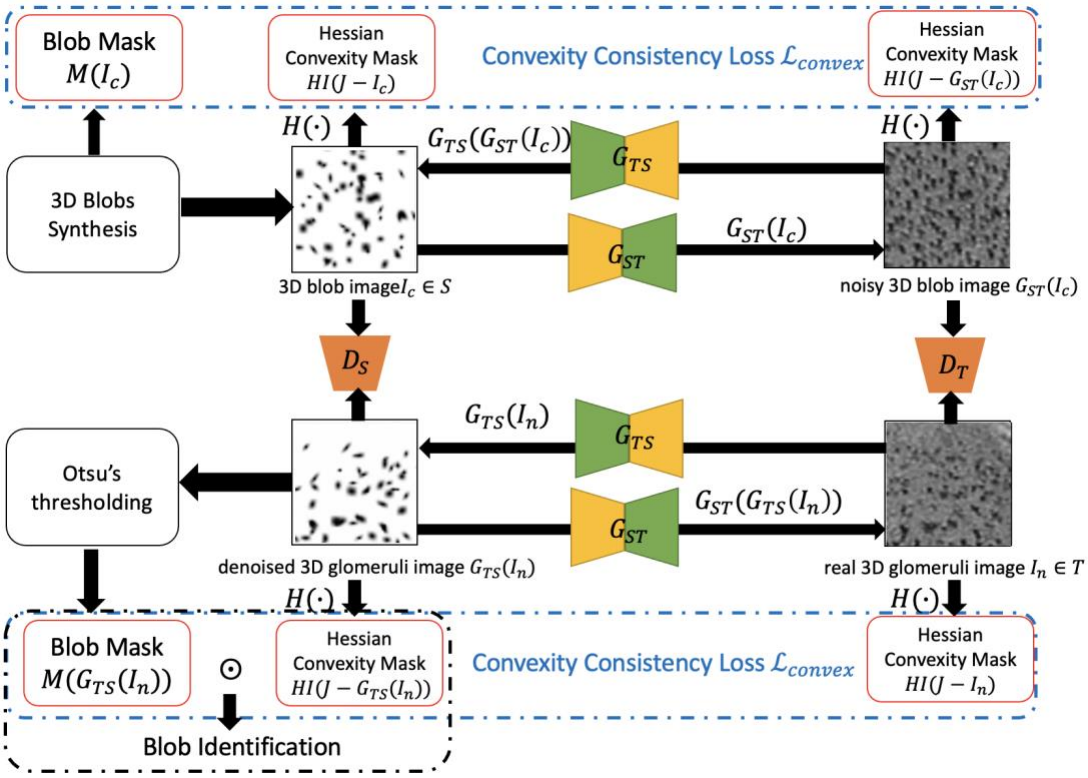


Figure 20 Illustration of 3D Blob Image Denoising through BlobGAN for Glomerular Identification. G_{ST} is the Generator to Generate the Noisy Images from 3D Blob Image $I_c \in S$. G_{TS} is the Generator to Denoise the Real Glomeruli Images $I_n \in T$ (3D Patch from Human Kidney CFE-MRI). D_S is the Discriminator to Distinguish the Synthetic 3D Blob Image I_c and the Denoised 3D Glomeruli Image $G_{TS}(I_n)$. D_T is the Discriminator to Distinguish the Noisy 3D Blob Image $G_{ST}(I_c)$ and the Real 3D Glomeruli Image I_n. $HI(J - I_c)$ is the Hessian Convexity Mask of 3D Blob Image I_c. $HI(J - G_{ST}(I_c))$ is the Hessian Convexity Mask of Noisy 3D Blob Image $G_{ST}(I_c)$. $HI(J - I_n)$ is the Hessian Convexity Mask of Real 3D Glomeruli Image I_n. $HI(J - G_{TS}(I_n))$ is the Hessian Convexity Mask of 3D Denoised Glomeruli Image $G_{TS}(I_n)$. $M(I_c)$ is the Blob Mask of I_c. $M(G_{TS}(I_n))$ is the Blob Mask of $G_{TS}(I_n)$.

A schematic of BlobGAN is shown in Figure 20. The loss functions in equations (4.4)-(4.7) ensure that the distribution gap between the real 3D glomeruli images and the translated 3D blob images gradually decrease until $G_{ST}(I_c) \overset{p_{data}}{\approx} I_n$. However, it is possible for the generator to generate synthetic images where the geometric properties of the blobs are distorted and still minimize the adversarial loss since it is based only on the voxel intensities and the geometric distortions in the translated images cannot be identified by discriminators. The translated 3D blobs in $G_{ST}(I_c)$ with geometric distortions will have inconsistent geometric properties with the glomeruli in I_n. To overcome this limitation, we propose a convexity consistency loss function based on the Hessian in BlobGAN to preserve the geometric properties of blobs using a convexity consistent constrain. For any voxel (i, j, k) in 3D image I_f: $R^3 \rightarrow R$, the corresponding Hessian matrix $H(\cdot)$ is written as:

$$H\left(I_f(i,j,k)\right) = \begin{bmatrix} \frac{\partial^2 I_f(i,j,k)}{\partial i^2} & \frac{\partial^2 I_f(i,j,k)}{\partial i \partial j} & \frac{\partial^2 I_f(i,j,k)}{\partial i \partial k} \\ \frac{\partial^2 I_f(i,j,k)}{\partial i \partial j} & \frac{\partial^2 I_f(i,j,k)}{\partial j^2} & \frac{\partial^2 I_f(i,j,k)}{\partial j \partial k} \\ \frac{\partial^2 I_f(i,j,k)}{\partial i \partial k} & \frac{\partial^2 I_f(i,j,k)}{\partial j \partial k} & \frac{\partial^2 I_f(i,j,k)}{\partial k^2} \end{bmatrix} \tag{4.8}$$

For a blob brighter than the background, authors in (Frangi et al., 1998) have shown that each voxel in the blob has a negative value in the Hessian matrix. Assuming I_f consists of bright blobs, we define a binary Hessian indicator matrix $HI(\cdot)$ to measure the convexity of voxel (i, j, k) in I_f as:

$$HI(i,j,k) = \begin{cases} 1, \ H\left(I_f(i,j,k)\right) \prec 0 \\ 0, \ H\left(I_f(i,j,k)\right) \succcurlyeq 0 \end{cases} \tag{4.9}$$

This binary indicator matrix (referred subsequently to as Hessian convexity mask) identifies the blob voxels as 1 and the rest as 0. To preserve the geometry of 3D blobs in the translated images $G_{ST}(I_c)$, $G_{ST}(I_c)$ should have more convex voxels than I_c. Since I_c is a normalized image with darker blobs as compared to the background, we define the convexity consistent constraint for G_{ST} as:

$$\sum_{v=1}^{N} HI_v(J - I_c) \leq \sum_{v=1}^{N} HI_v\big(J - G_{ST}(I_c)\big), \tag{4.10}$$

where J denotes the unit matrix with all ones, N denotes the total number of voxels in I_c and v denotes index of voxel in I_c. In Section 4.2.1, we have shown that the distribution of the synthetic 3D blob images is similar to the distribution of the translated glomeruli images, which means $I_c \overset{p_{data}}{\approx} G_{TS}(I_n)$. This indicates that $G_{TS}(I_n)$ is the translated image with only glomeruli. To preserve the glomeruli voxels in the translated glomeruli images $G_{TS}(I_n)$, $G_{TS}(I_n)$ should have less convex voxels than I_n. Similar to equation (4.10), we derive the following convexity consistent constraint for G_{TS}:

$$\sum_{v=1}^{N} HI_v\big(J - G_{TS}(I_n)\big) \leq \sum_{v=1}^{N} HI_v(J - I_n), \tag{4.11}$$

To integrate the convexity constraint into BlobGAN, we define a convexity consistent loss function that minimizes the L1 norm between the Hessian indicator matrix of the original image and the translated image. To ensure that we only consider the convexity of blob voxels, we only include the voxels that belong to the blobs by using the blob mask $M(I_c)$ and $M\big(G_{TS}(I_n)\big)$ respectively for the synthetic blob image and the real glomeruli image (see Section 4.2.1).

$$\mathcal{L}_{convex}(G_{ST}, G_{TS}, H, M) = \mathrm{E}_{I_c \sim p_{data}(I_c)}\left[\left\|(HI(J - I_c) - HI\big(J - G_{ST}(I_c)\big))\odot M(I_c)\right\|_1\right] +$$

$$\mathrm{E}_{I_n \sim p_{data}(I_n)}\left[\left\|(HI\big(J - G_{TS}(I_n)\big) + HI(J - I_n))\odot M(G_{TS}(I_n))\right\|_1\right]. \quad (4.12)$$

Here the blob mask $M\big(G_{TS}(I_n)\big)$ is a binary mask obtained through Otsu's thresholding (Otsu & N., 1996) of $G_{TS}(I_n)$. The convexity consistency loss regularizes the generators to preserve voxel convexity. This will ensure that the synthetic blob voxels are fixed during translation from I_c to $G_{ST}(I_c)$ and glomeruli voxels are fixed during translation from I_n to $G_{TS}(I_n)$. Then we can write the data distribution of $G_{ST}(I_c)$ and $G_{TS}(I_n)$ as the following linear functions:

$$G_{ST}(I_c) \overset{p_{data}}{=} I_c + \varepsilon_{G_{ST}(I_c)} \quad (4.13)$$

$$G_{TS}(I_n) \overset{p_{data}}{=} I_n - \varepsilon_{I_n}, \quad (4.14)$$

where $\varepsilon_{G_{ST}(I_c)}$ is the noise generated by G_{ST} on $G_{ST}(I_c)$ and ε_{I_n} is noise deducted from I_n by G_{TS}. Since we know that $I_c \overset{p_{data}}{\approx} G_{TS}(I_n)$ and $G_{ST}(I_c) \overset{p_{data}}{\approx} I_n$, it is clear to see that:

$$\varepsilon_{G_{ST}(I_c)} \overset{p_{data}}{\approx} \varepsilon_{I_n}. \quad (4.15)$$

Equation (4.15) shows that G_{ST} is able to selectively generate the noise of real glomeruli images I_n on the synthesized 3D blob images I_c and G_{TS} is able to remove the noise from real glomeruli images I_n to generate $G_{TS}(I_n)$. This indicates that BlobGAN is essentially a GAN with the cycle of generating noise through G_{ST} and denoising through G_{TS}.

By combining the loss functions described in the foregoing, the complete objective function of BlobGAN is given as:

$$\mathcal{L}(G_{ST}, G_{TS}, D_S, D_T, H, M) = \mathcal{L}_{GAN}(G_{ST}, D_T) + \mathcal{L}_{GAN}(G_{TS}, D_S) +$$

$$\lambda_{cycle}\mathcal{L}_{cycle}(G_{ST}, G_{TS}) + \lambda_{identity}\mathcal{L}_{identity}(G_{ST}, G_{TS}) +$$

$$\lambda_{convex}\mathcal{L}_{convex}(G_{ST}, G_{TS}, H, M)\,, \tag{4.16}$$

where λ_{cycle}, $\lambda_{identity}$ and λ_{convex} are positive weighted coefficients to control the importance of cycle consistency loss, identity loss, and convexity consistency loss, respectively. We aim to solve:

$$G_{ST}^*, G_{TS}^*, HI^*(J - G_{TS}(I_n)), M^*(G_{TS}(I_n)) = \arg \min_{G_{ST}, G_{TS}, H, M} \max_{D_S, D_T} \mathcal{L}(G_{ST}, G_{TS}, D_S, D_T, H, M) \tag{4.17}$$

To optimize $\mathcal{L}_{GAN}$, $\mathcal{L}_{cycle}$, $\mathcal{L}_{identity}$ and $\mathcal{L}_{convex}$, we update them alternatively. D_S, D_T, $HI(J - G_{TS}(I_n)), M(G_{TS}(I_n))$ are fixed when optimizing G_{ST} and G_{TS}. G_{ST} and G_{TS} are fixed when optimizing D_S, D_T, $HI(J - G_{TS}(I_n)), M(G_{TS}(I_n))$. After BlobGAN satisfies some stopping criteria, we get the denoised glomeruli images $G_{TS}(I_n)$, Hessian convexity mask $HI^*(J - G_{TS}(I_n))$ and blob mask $M^*(G_{TS}(I_n))$ are also derived. The final step is to use Hessian analysis to identify the 3D blobs (glomeruli) from these two masks. We discuss the final step in the next section.

4.2.3 3D Blob Identification through Joint Constraint Operation

The BlobGAN approach presented in Section 4.2.2 provides the Hessian convexity mask $HI^*(J - G_{TS}(I_n))$ and the blob mask $M^*(G_{TS}(I_n))$. Let the final 3D denoised blob image be represented as $I_d: R^3 \rightarrow R$, From Section 4.2.2, we note that $I_d = G_{TS}(I_n)$ with the corresponding Hessian convexity mask and blob mask of denoted as $HI(J - I_d)$ and $M(I_d)$. Prior works (Xu et al., 2019) have shown that Hessian analysis tends to over-segments the blobs and the U-Net probability map tends to under-segment the blobs. To resolve the issues associated with Hessian analysis and U-Net, we use a joint constraint

operation to derive the final blobs as discussed in (Xu et al., 2020). Towards this, the final blob identification mask I_b is derived by applying the joint operation on the two masks:

$$I_b = HI(J - I_d) \odot M(I_d) \,, \tag{4.18}$$

where the joint operator $\odot$ is the Hadamard product. Figure 21 shows the step-by-step methodology of blob identification through BlobGAN and joint constraint operation. The blue circle in the original noisy blob image and corresponding denoised blob image in Figure 21 (a) and Figure 21 (b), respectively, show two overlapping blobs. In the blob mask shown in Figure 21 (d), the overlapping blobs are incorrectly segmented as one blob. But in the Hessian convexity mask shown in Figure 21 (c), they are correctly identified as two distinct blobs. However, the Hessian convexity mask over-segments certain regions causing the noisy regions to be detected as blobs (see the yellow circle in Figure 21 (c)). In contrast, the blob mask correctly filters the noise, as shown by the yellow circle in Figure 21 (d). Therefore, by taking the intersection of the Hessian convexity mask and the blob mask, their respective issues of under-segmentation and over-segmentation are alleviated. The final blob mask derived by the joint constraint operation is shown in Figure 21 (e). Clearly, it is able to selectively segment the blobs while avoiding over-segmentation of noisy regions.

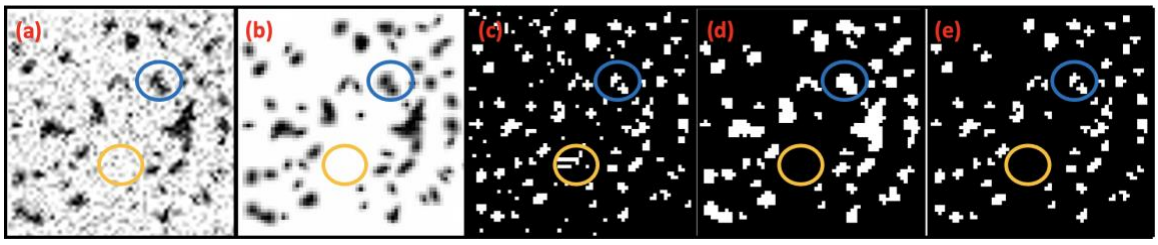


Figure 21 Illustration of Blob Identification through Joint Constraint Operation. (a) Original Noisy Blob Image. (b) Denoised Blob Image I_d. (c) Hessian Convexity Mask $HI(J - I_d)$. (d) Blob Mask $M(I_d)$. (e) Final Blob Identification Mask I_b.

Compared with UH-DoG (Xu et al., 2020), the benefit of BlobGAN is that our denoised 3D blob image I_d is already smoothed without the need for additional smoothing steps. In addition, the Hessian convexity mask and the blob mask in BlobGAN are derived from I_n. But in UH-DoG, the two masks come from two different approaches (U-Net and HDoG), which increases the inconsistency in the blob identification. We define the true 3D blob candidate as a 6-connected voxel, so the final identified blobs set are represented as:

$$S_{blob} = \{(i,j,k) \mid (i,j,k) \in I_d, I_b(i,j,k) = 1\} \tag{4.19}$$

4.2.4 Network Architecture of BlobGAN

BlobGAN is directly trained on 3D images. We provide 3D images with 64×64×32 voxels as input, then resize them to 128×128×64 voxels and normalize the voxel intensities to $[0,1]$. The generators G_{ST} and G_{TS} adopt the encoder-decoder structure with residual blocks, similar to (Johnson et al., 2016). Each generator consists of a $7 \times 7 \times 7$ 3D Convolution-InstanceNormalization-ReLU layer followed by two downsampling layers, six residual blocks, two upsampling layers, and a tanh loss function. Each downsampling layer is a 3×3×3 3D Convolution-InstanceNormalization-ReLU layer with stride size two. Each residual block consists of two 3×3×3 3D Convolution-InstanceNormalization-ReLU layers. Each upsampling layer is a $3 \times 3 \times 3$ 3D fractional-strided-Convolution-InstanceNormalization-ReLU layer with a stride size of two. A 3D replication padding with padding size one is used in all convolution layers. Each of the discriminators D_S and D_T adopt a 70×70 PatchGAN (Isola et al., 2017) for classifying the real images and translated images. It consists of three 4×4×4 3D Convolution-InstanceNormalization-LeakyReLU layer with stride size 2, one 4×4×4 3D Convolution-InstanceNormalization-LeakyReLU

layer with a stride size of one, and one 4×4×4 3D Convolution layer with stride size of one. The LeakyReLU in each layer has a slope of 0.2. Finally, a sigmoid activation function is applied in the output layer.

4.3 Experiments and Results

4.3.1 Training Dataset

We evaluated our method using three case studies. BlobGAN needs training datasets from source domain S and target domain T as input. Since the source domain S in BlobGAN is fixed, the training dataset from source domain S for three case studies are synthesized 3D blobs using the function in Section 4.2.1. For the training dataset from target domain T, we used the simulated noisy images in one case study and the real-world CFE-MR images of the kidney in the other two case studies.

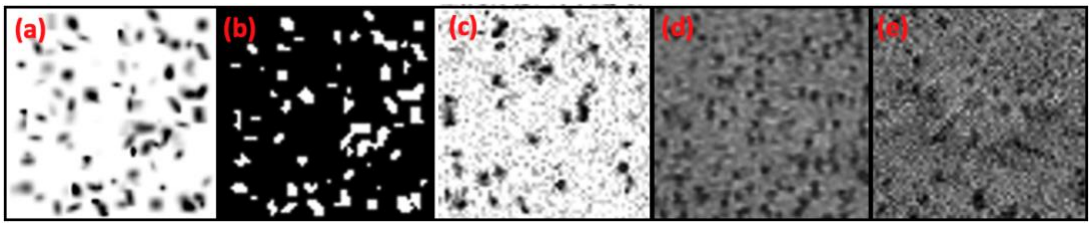


Figure 22 Illustration of Training Input Images of BlobGAN (a) Synthesized 3D Blob Image from Domain S. (b) Blob Mask of (a) from Domain S. (c) Synthesized 3D Noisy Blob Image from Domain T. (d) 3D CFE-MR Image of the Human Kidney: Patch from Domain T. (e) Kidney Image Patch from Domain T in 3D.

In the first case study, we randomly synthesized 1,000 3D blob images (64×64×32 voxels) to construct the source domain S. For each 3D training image (Figure 22 (a)), blobs were scattered randomly in the image space. The number of blobs for each image is generated using a random number generator and ranged from 500 to 800. The parameters

of 3D elliptical Gaussian function for each synthesized blob are as follows: $\theta, \varphi \in [0, 180^{\circ}]$, σ_x, σ_y, $\sigma_z \in [0.5, 1.5]$. We record the blob mask (Figure 22 (b)) of each image. For the target domain Y, we synthesized another 1,000 3D blob images using the same 3D blob synthesis function. To simulate noisy 3D blob images, we added random Gaussian noise to the synthesized images. The noise was generated by the Gaussian function with $\mu_{noise} = 0$ and σ^2_{noise} defined by:

$$\sigma^2{}_{noise} = \frac{\sigma^2{}_{image}}{10^{\frac{SNR}{10}}} \ , \tag{4.20}$$

where the signal-to-noise ratio (SNR) lies in the interval $[0.01dB, 1dB]$. The 1,000 3D blob images and 1,000 3D noisy blob images comprise the training dataset for the first case study. The validation experiments of the first case will be discussed in the Section 4.3.2.

In the second case study, the source domain S contains 1,000 3D blob images (64×64×32 voxels), which is the same as the first case study. For target domain T, we studied three 3D human kidneys CFE-MR images. Each human kidney CFE-MR image has voxel dimensions of 896×512×512. These three human kidneys were obtained after autopsy through a donor network (The International Institute for the Advancement of Medicine, Edison, NJ) after receiving Institutional Review Board (IRB) approval and informed consent from Arizona State University (Beeman et al., 2014). They were imaged by CFE-MRI as described in (Baldelomar et al., 2016b, 2017; Beeman et al., 2014; Bennett et al., 2008b). To appropriately validate our model, we train the BlobGAN on two human kidneys and test on the third one. We randomly sampled 1,000 3D non-overlapping patch images (64×64×32 voxels, Figure 22 (c)) from the two human kidneys in the training dataset. The sampling process is performed in the cortex region because the medulla does not have any glomeruli. The medulla and cortex regions of human kidneys were annotated

by a domain expert. The resulting 1,000 3D blob images and 1,000 3D human kidney patch images are treated as the training dataset for the second case study. The validation experiments of the second case study will be discussed in Section 4.3.3.

In the third case study, the source domain S contains 800 3D blob images (64×64×32 voxels). For the target domain T, we studied 62 mice kidney CFE-MR images. Each mouse kidney MR image has voxel dimensions of 256×256×256. This dataset includes chronic kidney disease (CKD, n=26) vs. controls (n=18), acute kidney injury (AKI, n=10) vs. control (n=8). The animal experiments were approved by the Institutional Animal Care and Use Committee (IACUC) under protocol #3929 on 04/07/2020 at the University of Virginia, in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals. They were imaged by CFE-MRI as described in (Charlton et al., 2020). For validation, we split the 62 mice kidneys into a training set and a validation set. The training dataset includes 44 mice kidneys and the validation dataset includes 18 mice kidneys. We randomly sampled 800 disjoint 3D patch images (64×64×32 voxels) from 44 mice kidneys (Figure 22 (d)). The sampling process is performed in the cortex region only. The 800 3D blob images and 800 3D mice kidney patch images comprise the training dataset for the third case study. The validation experiments of the third case study will be discussed in Section 4.3.4.

4.3.2 Experiment I: Validation Experiments using 3D Noisy Synthetic Image Data

To validate the performance of BlobGAN, we synthesized an independent group of 1,000 3D blob images (64×64×32 voxels) with Gaussian distributed noise as the test data. The version of these images without noise and their blob masks are recorded as ground truth to compare the performance with other methods. Seven methods were applied to the

synthetic 3D blob images: the HDoG (M. Zhang et al., 2016), U-Net with standard thresholding(Falk et al., 2019), U-Net with optimal thresholding (OT U-Net) (Xu et al., 2019), the UH-DoG (Xu et al., 2020), BTCAS (Xu et al., 2021), CycleGAN (J.-Y. Zhu et al., 2017) and our proposed BlobGAN. The goal of applying CycleGAN in this experiment is to show that BlobGAN is capable of denoising and preserving the geometric property of blobs during the image translation. The comparison of denoising blob images will be discussed in Section 4.3.5. Original CycleGAN is for 2D images, so we modified 2D CycleGAN to 3D CycleGAN in this experiment.

The parameter settings of the DoG are as follows: window size is set to 7, the normalization factor is set to 2, and the increment of scale is set to 0.001. To denoise the images of the 3D blobs using a trained U-Net, we first resized each 256×256 slice to 512×512, and each slice was fed into the U-Net. We used the Adam optimizer in U-Net with a learning rate of 0.0001. The dropout rate was set to 0.5. The threshold for the U-Net probability map in UH-DoG was set to 0.5. A 2D U-Net probability map was rendered on each slice, then stacked together to construct the 3D probability map of blobs from U-Net, U-Net with Optimal Thresholding (OT U-Net), UH-DoG, and BTCAS. U-Net, CycleGAN and BlobGAN were all implemented on an NVIDIA TITAN XP GPU with 12 GB of memory. To train the CycleGAN and BlobGAN, λ_{cycle} and λ_{convex} were set to 10, $\lambda_{identity}$ was set to 0.5. CycleGAN and BlobGAN were both trained from scratch with a learning rate 0.0002. The training typically took about 20 epochs to converge, so we did not set up the decay policy for the learning rate. We used the Adam optimizer with a batch size set to 1.

We evaluated the performance of these seven methods by the following seven metrics: Detection Error Rate (DER), Precision, Recall, F-score, Dice coefficient, Intersection over Union (IoU), and Blobness. For detection, DER measures the difference ratio between the number of detected blobs and the ground truth. Precision measures the fraction of retrieved blobs confirmed by the ground truth. Recall measures the fraction of ground-truth data retrieved. F-score is the overall performance of precision and recall. For segmentation, the Dice coefficient measures the similarity between the segmented blob mask and the ground truth. IoU measures the amount of overlap between the segmented blob mask and the ground truth. For synthesis, the Blobness measures the likelihood of the objects with blobs shape.

Since the blob locations (the coordinates of the blob centers) were already generated when synthesizing the 3D blob images, so ground truth number of blobs for all 3D images was already recorded. DER can be calculated by equation (4.21).

$$DER = \frac{|N_{GT} - N_{Det}|}{N_{GT}} , \tag{4.21}$$

where N_{GT} represents the ground truth number of blobs and N_{Det} represents the number of detected blobs. A candidate was considered as a true positive if the centroid of its magnitude was in a detection pair (p_i, p_j) for which the nearest ground truth center p_j had not been paired, and the Euclidian distance $D_{p_i p_j}$ between ground truth center p_j and blob candidate p_i was less than or equal to $dist$. To avoid duplicate counting, the number (#) of true positives TP was calculated by equation (4.22). Precision, Recall, F-score were calculated by equations (4.23), (4.24), (4.25).

$$TP = min\left\{\#\left\{(p_i, p_j): min_{i=1}^{m} D_{p_i p_j} \leq dist\right\}, \#\left\{(p_i, p_j): min_{j=1}^{n} D_{p_i p_j} \leq dist\right\}\right\}, \tag{4.22}$$

$$Precision = \frac{TP}{N_B}, \tag{4.23}$$

$$Recall = \frac{TP}{N_G}, \tag{4.24}$$

$$F - score = 2 \times \frac{Precision \times Recall}{(Precision + Recall)}, \tag{4.25}$$

where N_G is the number of true glomeruli and N_B is the number of blob candidates; $dist$ is a thresholding parameter set to a positive value $(0, +\infty)$. If $dist$ is small, fewer blob candidates are counted since the distance between the blob candidate centroid and ground-truth should be small. If $dist$ is too large, more blob candidates are counted. Here, since local intensity extremes could be anywhere within a small blob with an irregular shape, we set $dist$ to the average diameter of the blobs: $dist = 2 \times \sqrt{\frac{\sum_{(i,j,k)} I_f(i,j,k)}{\pi}}$. The Dice coefficient and IoU were calculated by comparing the segmented blob mask and ground truth mask by equations (4.26) and (4.27).

$$Dice\ (B_S, B_G) = \frac{2|B_S \cap B_G|}{|B_S| + |B_G|}, \tag{4.26}$$

$$IoU\ (B_S, B_G) = \frac{B_S \cap B_G}{B_S \cup B_G}, \tag{4.27}$$

where B_S is the binary mask for segmentation result and B_G is the binary mask for the ground truth. Based on identified blobs set S_{blob} from equation (4.19), blobness for each blob candidate b_i is calculated by equation (4.28):

$$Blobness_{b_i \in S_{blob}} = \frac{3 \times \left|\det\left(H(J - I_f)\right)\right|^{\frac{2}{3}}}{pm(H(J - I_f))}, \tag{4.28}$$

where I_f is the normalized 3D blob image, H represents the Hessian matrix and pm represents the principal minors of Hessian matrix. We assume blobs in I_f are dark blobs in equation (4.28) to keep consistent with other equations. Note that in (M. Zhang et al., 2016), blobness is calculated based on DoG transformed blob image, but here we use the normalized 3D blob image instead without DoG transformation. As a result, we just need to turn these dark blobs into bright blobs by $J - I_f$.

Comparisons between the models are shown in Table 15. Two sample t-test was performed at a significance level of 0.05. BlobGAN significantly outperforms the other five methods (HDoG, U-Net, OT U-Net, BTCAS, CycleGAN) on all seven metrics. Compared to UH-DoG, BlobGAN provides better performance on DER, Recall, F-Score, Dice, IoU and is comparable on Precision and Blobness. Note that since we already know the blob mask of each synthetic image, we calculated the average blobness of all images as ground truth. In this experiment, the average blobness for all images is 0.519. Table 15 shows that BlobGAN achieves the closest number with ground truth.

Table 15 Comparison (Avg ± Std) and Two Sample T-test of BlobGAN, HDoG, U-Net, OT U-Net, UH-DoG, BTCAS, CycleGAN on 3D Synthetic Images

METRICS	BLOBGAN	HDOG	U-NET	OT U-NET	UH-DOG	BTCAS	CYCLEGAN
DETECTION ERROR RATE	0.081±0.039	1.362 ±0.390 **(*<0.0001)**	0.380 ±0.074 **(*<0.0001)**	0.385±0.081 **(*<0.0001)**	0.166±0.040 **(*<0.0001)**	0.683±0.301 **(*<0.0001)**	0.384±0.052 **(*<0.0001)**
PRECISION	0.940±0.013	0.337±0.045 **(*<0.0001)**	0.883±0.096 **(*<0.0001)**	0.864±0.108 **(*<0.0001)**	0.940±0.040 **(0.723)**	0.602±0.086 **(*<0.0001)**	0.978±0.008 **(*<0.0001)**
RECALL	0.864±0.034	0.780±0.029 **(*<0.0001)**	0.545±0.059 **(*<0.0001)**	0.528±0.060 **(*<0.0001)**	0.786±0.029 **(*<0.0001)**	0.988±0.004 **(*<0.0001)**	0.602±0.052 **(*<0.0001)**
F-SCORE	0.900±0.019	0.468±0.041 **(*<0.0001)**	0.672±0.066 **(*<0.0001)**	0.653±0.071 **(*<0.0001)**	0.855±0.026 **(*<0.0001)**	0.744±0.070 **(*<0.0001)**	0.744±0.041 **(*<0.0001)**

DICE	0.796±0.070	0.323±0.005 **(*<0.0001)**	0.463±0.020 **(*<0.0001)**	0.481±0.020 **(*<0.0001)**	0.296±0.005 **(*<0.0001)**	0.348±0.012 **(*<0.0001)**	0.334±0.015 **(*<0.0001)**
IOU	0.666±0.082	0.193±0.004 **(*<0.0001)**	0.301±0.017 **(*<0.0001)**	0.317±0.017 **(*<0.0001)**	0.174±0.003 **(*<0.0001)**	0.210±0.009 **(*<0.0001)**	0.200±0.011 **(*<0.0001)**
BLOBNESS (GROUND TRUTH: 0.519)	0.528±0.289	0.861±0.213 **(*<0.0001)**	0.555±0.299 **(*<0.05)**	0.566±0.301 **(*<0.01)**	0.550±0.301 **(0.094)**	0.662±0.300 **(*<0.0001)**	0.645±0.304 **(*<0.0001)**

*significance $\boldsymbol{p} < 0.05$

4.3.3 Experiment II: Validation Experiments using 3D Human Kidney CFE-MR Images

In this experiment, we investigated the blob detection and segmentation approach on 3D CFE-MR images to measure the number (N_{glom}) and apparent volume (aV_{glom}) of glomeruli in healthy and diseased human donor kidneys that were not accepted for transplant. We applied the HDoG, UH-DoG, BTCAS and our proposed BlobGAN blob detectors to detect and segment glomeruli. The parameter settings of DoG are as follows: window size is set to 7, the normalization factor is set to 2, and the increment of scale is set to 0.001. We first generated 14,336 2D patches, with each patch 128×128 in size and each patch was then fed into U-Net. The final probability map of the whole kidney is reconstructed by stacking all the 2D patches and used for UH-DoG and BTCAS. The threshold for the U-Net probability map in UH-DoG was set to 0.5. Since the human kidney has voxel dimensions of 896×512×512 and the input of BlobGAN is a 3D patch with voxel dimensions of 64×64×32, we divided each human kidney into 1,792 3D patches (64×64×32) to validate the performance of BlobGAN. The final identification mask of the whole kidney is reconstructed by stacking all 3D patches. To train the BlobGAN, λ_{cycle} and λ_{convex} were set to 10, $\lambda_{identity}$ was set to 0.5. BlobGAN was trained from scratch with a learning rate 0.0002. The training typically took about 20 epochs to converge, so we

did not set up the decay policy for the learning rate. We used the Adam optimizer with a batch size set to 1.

N_{glom} and aV_{glom} are reported in Table 16 and Table 17, where the HDoG, UH-DoG, BTCAS, and the proposed BlobGAN blob detectors are compared to the data from unbiased dissector-fractionator stereology (Beeman et al., 2011). We used these stereology data from (Beeman et al., 2014) as ground truth and calculated aV_{glom} based on the method from (Baldelomar et al., 2016a). The differences between the results of the HDoG, UH-DoG, BTCAS, BlobGAN methods and stereology data are also listed in Table 16 and Table 17.

Compared to stereology, the HDoG identified more glomeruli and the difference with stereology is much larger than the other two methods, indicating over-detection under the single optimal scale of DoG and lower mean aV_{glom} than stereology. UH-DoG identified fewer glomeruli due to under-segmentation when using the single thresholding (0.5) on the probability map of U-Net combined with the Hessian convexity map. BTCAS identified more glomeruli due to over-detection when detecting glomerular centroids on the probability map of pre-trained U-Net. BlobGAN provided the most accurate measurements of N_{glom} and mean aV_{glom} than the other three methods.

Table 16 Human Kidney Glomerular Segmentation Results (N_{glom}) using BTCAS, HDoG, UH-DoG and Proposed BlobGAN Methods Comparing with Stereology

HUMAN KIDNEY	N_{glom} ($\times 10^6$) (STEREOLOGY)	N_{glom} ($\times 10^6$) (BLOBGAN)	Difference Ratio (%)	N_{glom} ($\times 10^6$) (BTCAS)	Difference Ratio (%)	N_{glom} ($\times 10^6$) (UH-DOG)	Difference Ratio (%)	N_{glom} ($\times 10^6$) (HDOG)	Difference Ratio (%)
CF 1	1.13	1.05	**7.08**	1.16	**2.65**	0.66	**41.60**	2.95	**>100**
CF 2	0.74	0.71	**4.05**	0.86	**16.22**	0.48	**35.14**	1.21	**63.51**

CF 3	1.46	1.48	**1.37**	1.50	**2.74**	0.85	**41.78**	3.93	**>100**

Table 17 Human Kidney Glomerular Segmentation Results (Mean aV_{glom}) using BTCAS, HDoG, UH-DoG and Proposed BlobGAN Methods Comparing with Stereology

HUMAN KIDNEY	Mean aV_{glom}($\times 10^{-3}mm^3$) (STEREOLOGY)	Mean aV_{glom}($\times 10^{-3}mm^3$) (BLOBGAN)	Difference Ratio (%)	Mean aV_{glom} ($\times 10^{-3}mm^3$) (BTCAS)	Difference Ratio (%)	Mean aV_{glom}($\times 10^{-3}mm^3$) (UH-DOG)	Difference Ratio (%)	Mean aV_{glom}($\times 10^{-3}mm^3$) (HDOG)	Difference Ratio (%)
CF 1	5.01	5.19	**3.59**	5.32	**6.19**	7.36	**46.91**	4.8	**41.91**
CF 2	4.68	4.80	**2.56**	4.78	**2.14**	5.62	**20.09**	3.2	**31.62**
CF 3	2.82	2.81	**0.35**	2.55	**9.57**	3.73	**32.37**	3.2	**13.48**

4.3.4 Experiment III: Validation Experiments using 3D Mouse Kidney CFE-MR Images

We conducted experiments on CF-labeled glomeruli from a dataset of 3D MR images to measure N_{glom} and aV_{glom} of glomeruli in healthy and diseased mouse kidneys. We applied the HDoG (M. Zhang et al., 2016), UH-DoG(Xu et al., 2020), BTCAS (Xu et al., 2021), and the proposed BlobGAN blob detectors to segment glomeruli. The results of HDoG with VBGMM from (M. Zhang et al., 2016), are used as the ground truth in this paper. The parameter settings of DoG are as follows: window size is set to 7, the normalization factor is set to 2, and the increment of scale is set to 0.001. To denoise the 3D blob images by using trained U-Net, we first resized each slice to 512×512 and each slice was fed into U-Net. The final probability map of the whole kidney is reconstructed by stacking all 2D patches and used for UH-DoG and BTCAS. The threshold for the U-Net probability map in UH-DoG was 0.5. Since the images of the mouse kidney were acquired with voxel dimensions of 256×256×256 and the input of BlobGAN is a 3D patch with voxel dimensions of 64×64×32, we divided each mouse kidney into 128 3D patches (64×64×32) to validate the performance of BlobGAN. The final identification mask of the whole kidney is also reconstructed by stacking all 3D patches. To train the BlobGAN,

λ_{cycle} and λ_{convex} were set to 10, $\lambda_{identity}$ was set to 0.5. The BlobGAN was trained from scratch with a learning rate 0.0002. The training typically took about 20 epochs to converge, so we did not set up the decay policy for the learning rate. We used the Adam optimizer with a batch size set to 1.

We perform quality control by visually checking the identified glomeruli, visible as black spots in the images. For illustration, Figure 23 and Figure 24 show the comparison of glomerular segmentation results on mouse kidneys of CKD group and AKI group using HDoG, UH-DoG, BTCAS, and BlobGAN, respectively. From the zoom-in regions from Figure 23 and Figure 24, it is clear that HDoG, UH-DoG, BTCAS have over-detection or under-segmentation and BlobGAN outperforms these blob detectors.

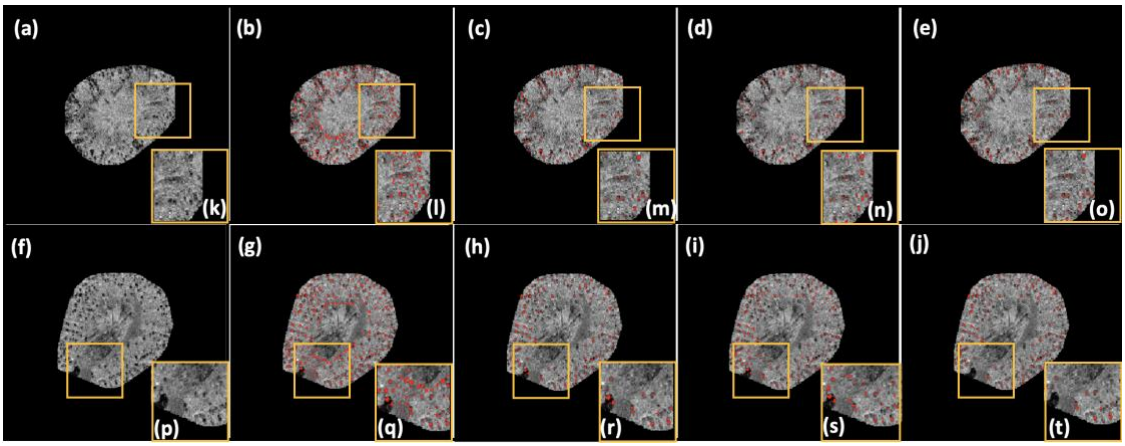


Figure 23 Comparison of Glomerular Segmentation Results from 3D MR Images of Mouse Kidneys (CKD and Control) using HDoG, UH-DOG, BTCAS and BlobGAN. Identified Glomeruli are Marked in Red. (a) Original Magnitude Image of CKD Mouse Kidney # 466 (Slice 100 out of 150). (b) Glomerular Segmentation Results of HDoG for (a). (c) Glomerular Segmentation Results of UH-DoG for (a). (d) Glomerular Segmentation Results of BTCAS for (a). (e) Glomerular Segmentation Results of

BlobGAN for (a). (f) Original Magnitude Image of Control Mouse Kidney # 427 (Slice 100 out of 182). (g) Glomerular Segmentation Results of HDoG for (a). (h) Glomerular Segmentation Results of UH-DoG for (a). (i) Glomerular Segmentation Results of BTCAS for (a). (j) Glomerular Segmentation Results of BlobGAN for (a). (k-o) are the Zoom-in Regions of (a-e) Respectively. (p-t) are the Zoom-in Regions of (f-j), Respectively.

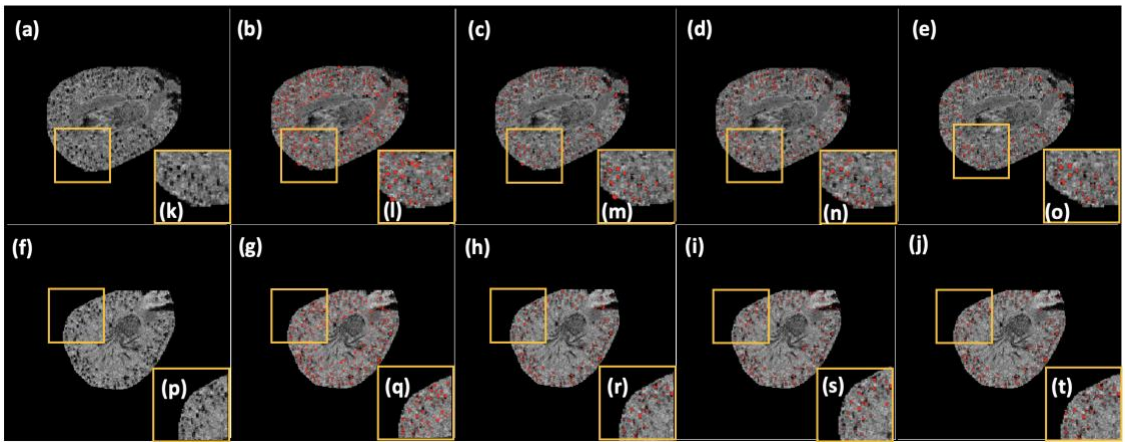


Figure 24 Comparison of Glomerular Segmentation Results from 3D MR Images of Mouse Kidneys (AKI and Control) using HDoG, UH-DOG, BTCAS and BlobGAN. Identified Glomeruli are Marked in Red. (a) Original Magnitude Image of AKI Mouse Kidney # 463 (Slice 100 out of 178). (b) Glomerular Segmentation Results of HDoG for (a). (c) Glomerular Segmentation Results of UH-DoG for (a). (d) Glomerular Segmentation Results of BTCAS for (a). (e) Glomerular Segmentation Results of BlobGAN for (a). (f) Original Magnitude Image of Control Mouse Kidney # 475 (Slice 100 out of 170). (g) Glomerular Segmentation Results of HDoG for (a). (h) Glomerular Segmentation Results of UH-DoG for (a). (i) Glomerular Segmentation Results of BTCAS for (a). (j) Glomerular Segmentation Results of BlobGAN for (a). (k-o) are the

Zoom-in Regions of (a-e) Respectively. (p-t) are the Zoom-in Regions of (f-j) Respectively.

N_{glom} and mean aV_{glom} are reported in Table 18 and Table 19, where the HDoG, UH-DoG, BTCAS and the proposed BlobGAN blob detectors are compared to HDoG with VBGMM from (M. Zhang et al., 2016). The differences between the results are also listed in Table 18 and Table 19. Compared to HDoG with VBGMM, the HDoG identified more glomeruli and the difference with HDoG with VBGMM is much larger than for the other three methods, indicating over-detection under the single optimal scale of the DoG and lower mean aV_{glom} than HDoG with VBGMM. UH-DoG identified fewer glomeruli and larger mean aV_{glom} due to under-segmentation when using the single thresholding (0.5) on the probability map of U-Net combined with the Hessian convexity map. BTCAS identified more glomeruli due to over-detection when detecting glomerular centroids on the probability map of pre-trained U-Net. BlobGAN provided the most accurate measurements of N_{glom} and mean aV_{glom} compared to HDoG and UH-DoG. Compared to BTCAS, BlobGAN provides more accurate N_{glom} but comparable on mean aV_{glom} because BTCAS seeks the local optimal DoG scale for each blob and provides the capability of measuring glomerular volume through best scale.

Table 18 Mouse Kidney Glomerular Segmentation (N_{glom}) from CFE-MRI using HDoG, UH-DoG, BTCAS and the Proposed BlobGAN Compared to HDoG with VBGMM Method

Mouse kidney	N_{glom} **(HDoG with VBGMM)**	N_{glom} **(BLOB GAN)**	**Difference Ratio (%)**	N_{glom} **(BTCAS)**	**Difference Ratio (%)**	N_{glom} **(UH-DoG)**	**Difference Ratio (%)**	N_{glom} **(HDoG)**	**Difference Ratio (%)**

CKD	ID 429	7,656	7633	**0.30**	7,719	**0.82**	7,346	**4.05**	10,923	**42.67**
	ID 466	8,665	8912	**2.85**	8,228	**5.04**	8,138	**6.08**	9,512	**9.77**
	ID 467	8,549	8802	**2.96**	8,595	**0.54**	8,663	**1.33**	12,755	**49.20**
	Avg	8,290	8449	**1.92**	8,181	**2.13**	8,049	**2.91**	11,063	**33.88**
	Std	552	709		440		663		1626	
Control for CKD	ID 427	12,724	12683	**0.32**	12,008	**5.63**	12,701	**0.18**	15,515	**21.93**
	ID 469	10,829	10921	**0.85**	11,048	**2.02**	11,347	**4.78**	15,698	**44.96**
	ID 470	10,704	10774	**0.65**	10,969	**2.48**	11,309	**5.65**	13,559	**26.67**
	ID 471	11,943	12692	**6.27**	12,058	**0.96**	12,279	**2.81**	16,230	**35.90**
	ID 472	12,569	12786	**1.73**	13,418	**6.75**	12,526	**0.34**	17,174	**36.64**
	ID 473	12,245	12058	**1.53**	12,318	**0.60**	11,853	**3.20**	15,350	**25.36**
	Avg	11,836	11986	**1.27**	11,970	**3.07**	12,003	**1.41**	15,588	**31.91**
	Std	872	920		903		595		1193	
AKI	ID 433	11,046	11618	**5.18**	10,752	**2.66**	11,033	**0.12**	12,315	**11.49**
	ID 462	11,292	11445	**1.35**	10,646	**5.72**	10,779	**4.54**	17,634	**56.16**
	ID 463	11,542	11544	**0.02**	11,820	**2.41**	10,873	**5.80**	20,458	**77.25**
	ID 464	11,906	11562	**2.89**	12,422	**4.33**	11,340	**4.75**	25,233	**≥100**
	Avg	11,447	11542	**0.84**	11,410	**3.78**	11,006	**3.85**	18,910	**64.21**
	Std	367	72		858		246		5401	
Control for AKI	ID 465	10,336	10214	**1.18**	10,393	**0.55**	10,115	**2.14**	13,473	**30.35**
	ID 474	10,874	10955	**0.74**	11,034	**1.47**	11,157	**2.60**	16,934	**55.73**
	ID 475	10,292	10222	**0.68**	9,985	**2.98**	10,132	**1.55**	12,095	**17.52**
	ID 476	10,954	11452	**4.55**	11,567	**5.60**	10,892	**0.57**	15,846	**44.66**
	ID 477	10,885	10929	**0.40**	11,143	**2.37**	11,335	**4.13**	14,455	**32.80**
	Avg	10,668	11154	**0.81**	10,824	**2.59**	10,726	**0.54**	14,561	**36.21**
	Std	325	643		630		572		1908	

Table 19 Mouse Kidney Glomerular Segmentation from CFE-MRI (Mean aV_{glom}) using HDoG, UH-DoG, BTCAS and the Proposed BlobGAN Compared to HDoG with VBGMM Method

Mouse kidney		**Mean aV_{glom}* (HDoG with VBGMM)**	**Mean aV_{glom}* (BLOBGAN)**	**Difference Ratio (%)**	**Mean aV_{glom}* (BTCAS)**	**Difference Ratio (%)**	**Mean aV_{glom}* (UH-DoG)**	**Difference Ratio (%)**	**Mean aV_{glom}* (HDoG)**	**Difference Ratio (%)**
CKD	ID 429	2.57	2.63	**2.28**	2.63	**2.33**	2.92	**11.99**	2.46	**4.28**
	ID 466	2.01	1.86	**8.06**	2.01	**0.00**	2.06	**2.43**	1.75	**12.94**
	ID 467	2.16	2.27	**4.85**	2.20	**1.85**	2.32	**6.90**	1.9	**12.04**
	Avg	2.25	2.25	**0.15**	2.28	**1.40**	2.43	**7.67**	2.04	**9.75**

	Std	0.29	0.39		0.32		0.44		0.37	
Control for CKD	ID 427	1.49	1.53	**2.61**	1.57	**5.37**	1.61	**7.45**	1.49	**0.00**
	ID 469	1.91	2.08	**8.17**	1.95	**2.09**	2.20	**13.18**	1.76	**7.85**
	ID 470	1.98	2.12	**6.60**	2.05	**3.54**	2.04	**2.94**	1.73	**12.63**
	ID 471	1.5	1.63	**7.98**	1.58	**5.33**	1.56	**3.85**	1.4	**6.67**
	ID 472	1.35	1.44	**6.25**	1.36	**0.74**	1.49	**9.40**	1.35	**0.00**
	ID 473	1.5	1.65	**9.09**	1.56	**4.00**	1.58	**5.06**	1.39	**7.33**
	Avg	1.62	1.74	**6.89**	1.68	**3.51**	1.75	**7.16**	1.52	**5.75**
	Std	0.26	0.29		0.26		0.30		0.18	
AKI	ID 433	1.53	1.61	**4.97**	1.64	**7.19**	1.63	**6.13**	1.38	**9.80**
	ID 462	1.34	1.45	**7.59**	1.41	**5.22**	1.48	**9.46**	1.3	**2.99**
	ID 463	2.35	2.42	**2.89**	2.4	**2.13**	2.61	**9.96**	1.94	**17.45**
	ID 464	2.31	2.34	**1.28**	2.36	**2.16**	2.40	**3.75**	1.78	**22.94**
	Avg	1.88	1.96	**3.84**	1.95	**4.18**	2.03	**7.27**	1.60	**13.29**
	Std	0.52	0.52		0.50		0.56		0.31	
Control for AKI	ID 465	2.3	2.36	**2.54**	2.46	**6.96**	2.40	**4.17**	2.11	**8.26**
	ID 474	2.44	2.67	**8.61**	2.34	**4.10**	2.52	**3.17**	2.14	**12.30**
	ID 475	1.74	1.79	**2.79**	1.86	**6.90**	1.70	**2.35**	1.58	**9.20**
	ID 476	1.53	1.58	**3.16**	1.57	**2.61**	1.62	**5.56**	1.49	**2.61**
	ID 477	1.67	1.71	**2.34**	1.68	**0.60**	1.70	**1.76**	1.61	**3.59**
	Avg	1.94	2.02	**4.06**	1.98	**4.23**	1.99	**2.62**	1.79	**7.19**
	Std	0.41	0.47		0.40		0.43		0.31	

*aV_{glom} unit $\boldsymbol{mm^3}$x $\boldsymbol{10^{-4}}$

4.3.5 Discussion of Image Denoising

Image denoising plays a critical role in blob identification from noisy images. The major challenge in segmenting the blobs is that they are small objects and may have similar intensity and size distribution as the background noise. Authors in (Xu et al., 2019) have shown that Hessian analysis is not robust to noise and leads to a high false positive rate. This explains the high value of *DER* for HDoG (see Table 15). To resolve this issue, UH-DoG and BTCAS use the probability map from a pretrained U-Net to denoise blob image. However, because the U-Net is pre-trained on a different dataset with potentially different noise distribution, the glomeruli denoising is only suboptimal. To illustrate, Figure 25

shows the comparison of denoised results of U-Net, CycleGAN, and BlobGAN. In Figure 25 (b), we note that the region enclosed within the blue circle in Figure 25 (a) is noise but is segmented as blobs in Figure 25 (c). In contrast, CycleGAN and BlobGAN correctly identify this region as noise, as shown in Figure 25 (d) and Figure 25 (e), respectively. This demonstrates CycleGAN and BlobGAN have better denoising performance than U-Net.

Noise can also have an indirect effect on the blobs during blob translation. For instance, there is a notable difference in the blob morphology within the yellow circle in Figure 25 (b) and Figure 25 (d). While CycleGAN identifies blobs in this region, the number and morphology of blobs is affected. By accounting for the convexity consistent constrain in BlobGAN, blobs within the yellow circle in Figure 25 (e) have a similar shape to those in Figure 25 (b). This shows that BlobGAN preserves the geometric properties of the blobs during translation. In the absence of convexity constraint, we observe geometric distortion for CycleGAN.

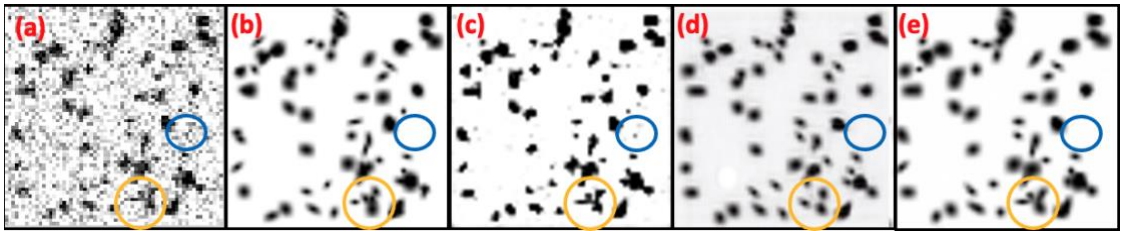


Figure 25 Denoising Results of Noisy Synthetic Blob Image using U-Net, CycleGAN, BlobGAN and Compared with Ground Truth. (a) Original Noisy Blob Image. (b) Ground Truth of (a). (c) Denoised Result of U-Net. (d) Denoised Result of CycleGAN. (e) Denoised Result of BlobGAN

4.4 Conclusion

In this research, we propose BlobGAN for small blob identification. This work provides three main contributions to the literature. First, we synthesized 3D blob images through a 3D elliptical Gaussian function to approximate the distribution of glomeruli and extend GAN to a noise translation model. Second, we proposed a convexity consistent constraint to prevent geometric distortions of blobs during translation. Third, based on the denoising output of GAN, we applied a joint constraint operation for final blobs identification. We conclude that BlobGAN could achieve high denoising performance and selectively denoise the image without affecting the blobs. Also, BlobGAN significantly outperformed the other six blob detectors in the validation experiments, leading to a more accurate imaging biomarker. While the results of this study are encouraging, there is room for improvement. The proposed method assumes that the blobs from 3D noisy images satisfy the convexity assumption. It might not be capable of detecting those blobs without this geometric property. Especially for CFE-MRI, varied parameters and artifacts affect blobs' shape during the imaging process. In the future, we plan to generate more realistic 3D blob images by integrating these physical constraints into the GAN architecture. Alternatively, we plan to explore the development of two models, one from GAN - a data-driven model, one from physics-based model. We plan to develop model cross-talk architecture for small blob synthesis and identification. Second, this approach may not be computationally efficient. The major reason is that it is a 3D GAN, so the size of network parameters is much larger than 2D GAN. If we reduce the model to 2D, the performance might be affected due to the loss of constraining the 3rd dimension. We plan to develop an approach to reduce this computation power based on a core set of training data. By training

these representative core samples from the whole data, the model will be able to reach a comparable performance.

CHAPTER 5

GAN TRAINING ACCELERATION USING FRÉCHET DESCRIPTOR BASED CORESET

Generative Adversarial Networks (GANs) are a class of deep learning models being applied to image processing, that consist of generators and discriminators. The generator is trained to synthesize fake versions of the original images, and the discriminator is employed to distinguish fake images from the real ones. The generator and discriminator networks are trained together iteratively, resulting in generated images similar to the real images. GANs have demonstrated state-of-the-art performances in applications such as image generation, image-to-image translation, just to name a few. With the success, it comes to realize the training of GANs takes long time and often is limited by available computing resources. In this research, we propose to construct Coreset using Fréchet Descriptor Distances (FDD-Coreset) to accelerate the training process of GAN for blob identification application. We first propose a Fréchet Descriptor Distance (FDD) to measure the difference between each pair of blob images based on the statistics derived from blob distribution. The Coreset is then employed using our proposed FDD metric to select samples from the entire dataset for GAN training. For validation purpose, a 3D simulated dataset of blobs and a 3D MRI dataset of human kidneys are studied. Using computation time and eight performance metrics including peak signal to noise ratio, detection error rate, precision, recall, F-score, Dice, IoU, and Blobness, the GAN model trained on FDD-Coreset is compared against with the model trained on entire dataset (1,000 random samples) and Inception and Euclidean Distance based Coreset (IED-Coreset). We conclude that FDD-Coreset not only significantly reduces the training time, but also

achieves higher denoising performance and maintains approximate performance of blob identification compared with training on entire dataset.

5.1 Introduction

Generative Adversarial Networks (GANs), proposed by Ian Goodfellow (Goodfellow et al., 2020) in 2014, are a class of deep learning models, applied to image processing, that consist of two sub-networks: a generator and a discriminator. The generator is trained to synthesize artificial versions of the original images, and a discriminator is employed to distinguish the artificial images from the real images. During the GAN training, both the generator network and the discriminator network interacts with each other iteratively, resulting in generated images resembling as close as to the real images. GANs have demonstrated state-of-the-art performance in many applications including medicine, such as medical image generation (Han et al., 2018), medical image-to-image translation(Sandfort et al., 2019), etc. Most recently, CycleGAN(J.-Y. Zhu et al., 2017) has attracted great attention to translate unpaired images from one domain (known as source domain, e.g., one image modality) to another (known as target domain, e.g., a different image modality) by simultaneously training two sets of generators and discriminators, one for each domain. CycleGAN has been applied to organ segmentation (Sandfort et al., 2019), tumor detection (H. Zhu et al., 2020), medical image denoising (Gu et al., 2021) , medical image synthesis (Hiasa et al., 2018), blob detection, as shown in CHAPTER 4, etc. In general, there are two types of CycleGAN-based models: (1) imaging level translation; (2) object focused image translation. For imaging level translation, the model transfers whole images from one domain to another domain on a pixel or voxel basis. For example, Oulbacha et al.

(Oulbacha & Kadoury, 2020) proposed a pseudo-3D CycleGAN to synthesize CT images from MRI for surgical guidance. Yang et al. (Yang et al., 2021) proposed a switchable CycleGAN with adaptive instance normalization to generate synthesized images for hypopharyngeal cancer diagnosis. For object focused image translation, the model considers the domain knowledge from the target objects as added constraints to regularize the image translation to alleviate the geometric distortion issue. For example, Zhang et al.(Z. Zhang et al., 2018)utilized segmentation to derive a shape consistency constraint and applied it on CycleGAN to solve the shape distortion in cross-modality synthesis. Ma et al. (Ma et al., 2021) introduced illumination regularization and structure loss function for medical image enhancement. Xu et al. leveraged the geometric information of the target objects to assist the synthetic image rendering as shown in CHAPTER 4.

While the performance of CyleGAN-based models in these applications are promising, a common critique is the model training is slow and often limited by computing resources. For example, the original CycleGAN (Anoosheh et al., 2018; J.-Y. Zhu et al., 2017) took 220 hours of training with NVIDIA Titan X GPU on 10,000 2D paintings images. BlobGAN in CHAPTER 4 trained on 1,000 3D blob images took ~26 hours for 50 epochs. One may argue that the training time can be reduced using advanced GPUs. However, such computing resources may not be readily available. To reduce the training time, researchers began exploring training using a subset instead of the whole dataset. Using false positive rate to measure the performance of generated image quality, Nuha et al. (Nuha & Afiahayati, 2018) proposed DCGAN model and showed the training on subsets can lead to a false positive rate comparable to the training using the entire dataset with less computing time. Unfortunately, it was noted that the performance may be unstable - for

some experiments - DCGAN led to very low false positive rate (Nuha & Afiahayati, 2018). To maintain the quality of GAN's generated image during subset selection, DeVries et al. (DeVries et al., 2020) proposed a novel instance selection approach based on manifold density of dataset. They removed the low-density regions to improve subsets' samples quality. Yet, this assumes that the low-density regions are noisy data region which may not always be true for medical images. Additionally, the computational time of this approach trained on ImageNet was reduced from 14.8 days to 3.7 days only (DeVries et al., 2020), which still is considered to be computationally expensive even with advanced GPU power, for example, 8 NVIDIA V100 GPUs in this case.

Recently, Coreset has attracted considerable attention. In Coreset, a subset of the data is to be optimally selected such that model trained on the selected subset will perform as closely as possible to the model trained on the entire dataset. Huang et al. (Huang et al., 2019) constructed Coreset to accelerate the computation of K-means and K-median clustering algorithms. Sener et al. (Sener & Savarese, 2017) formulated the active learning problem as a Coreset selection problem where only core samples are to be labeled to improve the performance of Convolutional Neural Network (CNN) models. Coreset has also been adopted in GAN training. For example, Sinha et al. (Sinha et al., 2019) developed a Coreset sampling approach to speed up GAN training. A pre-trained Inception classifier was first implemented to extract the Inception embedding from the whole dataset based on which pairwise Euclidean distances were calculated. The Coreset is then derived for training in each iteration through comparing the distance between data samples. While this approach accelerates the GAN training to some extent, it is noted the Coreset needs to be generated within each iteration of the training, in other words, within the training loop.

From iteration to iteration, it is highly likely the same images will be selected for the Coreset which is computationally inefficient, especially considering the long computational time required to generate the Inception embedding. And (Sinha et al., 2019) focused on the GAN model taking a batch of images as Coreset for training, while CycleGANs utilize the Instance Normalization (Ulyanov et al., 2016) as its normalization method for each layer and thus the training of CycleGAN often needs one image instead of a batch of images per iteration. Recognizing the potentials of Coreset and issues from deriving Coreset within the training loop as in (Sinha et al., 2019), we argue the training of GAN, in this study, CycleGANs can be greatly accelerated if the Coreset selection is taken out of the training loop.

We propose to implement Coreset as a pre-training step instead of within-training process, and in this case, the Euclidian distance calculated from Inception embedding (derived during the GAN training) for Coreset construction in (Sinha et al., 2019) does not apply. Though an Inception classifier can be employed to derive Inception embedding, additional computational cost occurs which is against the goal of this study. In addition, the Inception embedding may be problematic to calculate the distance metric due to the fact medical images are known to be noisy. Motivated by object focused image translation approaches, we introduce Fréchet Descriptor Distance (FDD) derived from object related statistics to accelerate GAN's training. To demonstrate this idea, we focus on one specific type of imaging problem: blob images. This type of imaging has shown in a number of medical applications such as nuclei in 2D microscopy images (Lee et al., 2018), glomeruli in 3D kidney cationic ferritin enhanced MRI (Xu et al., 2020, 2021). These images have some common characteristics: number of blobs is large, and the shape of these blobs

roughly follows Gaussian distribution. In this research, we introduce a new blob- based FDD to measure the distances between the image pair based on object (blob) statistics. The Coreset is then constructed using the new FDD metric on the image pairs, then the Coreset selection step can be taken out of the GAN training loop. To validate the performance of out of loop FDD-Coreset for GAN training, we conduct two experiments. The first experiment is to identify the blobs in a 3D simulated blob image dataset where the locations of blobs are known. We choose the naive random sampling method and the Inception and Euclidean Distance based Coreset (IED-Coreset) (Sinha et al., 2019) for comparison. Other than computation time, eight performance metrics are used including peak signal to noise ratio, detection error rate, precision, recall, F-score, Dice, IoU, and Blobness. In the second experiment, we implement the FDD-Coreset on 3D human kidney images obtained from MRI. Since there is no ground truth available, we compare against the above two methods using stereology results. Both experiments support the conclusion that FDD-coreset can significantly accelerate the GAN training with comparable performance.

The remainder of the paper is organized as follows: Section 5.2 describes our proposed FDD-Coreset in detail. Section 5.3 demonstrates the comparative results on the 3D synthetic images and 3D human kidney images. Finally, the conclusions are represented in Section 5.4.

5.2 Methods

In the proposed FDD-Coreset approach, given the group of descriptors of blob distribution from the entire blob images dataset, a Fréchet Descriptor Distance (FDD) is first derived to measure the difference between each pair of blob images; a Coreset is then selected from the entire dataset based on FDD to train a GAN model. To demonstrate the

concept of FDD-Coreset, the GAN of interest in this research is BlobGAN from CHAPTER 4, a model designed for small blob identification. Because the FDD-Coreset is a generalized approach, it can be applied to other GAN models.

5.2.1 BlobGAN and Blob Descriptors

BlobGAN (Figure 26) consists of three steps: (1) 3D synthetic blobs are rendered using a 3D elliptical Gaussian function. The 3D blobs with respect to the corresponding masks and true images comprise the training input; (2) a 3D GAN is trained to denoise the images; (3) denoised 3D blobs are identified from the denoised images.

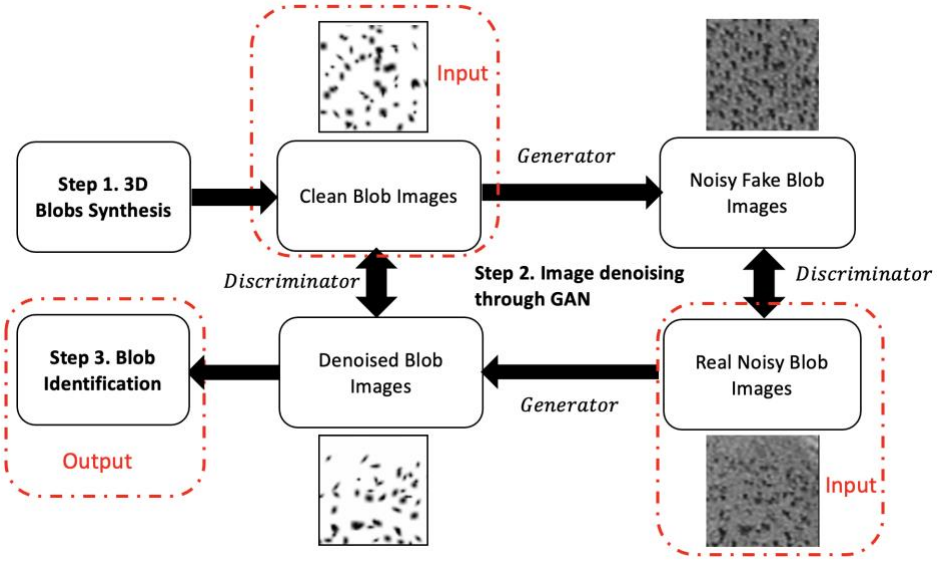


Figure 26 The Overview of the BlobGAN Model for Blob Identification.

Here we apply the elliptical Gaussian function to identify the blob descriptors. A 3D elliptical Gaussian function is in the general form as following:

$$F(x,y,z) = \mathcal{A} \cdot e^{-(a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 + d(x-x_0)(y-y_0) + e(y-y_0)(z-z_0) + f(x-x_0)(z-z_0))}, \quad (5.1)$$

where $\mathcal{A}$ is a normalization factor, and x_0, y_0 and z_0 are the coordinates of the center of the Gaussian function $F(x, y, z)$. The coefficients a, b, c, d, e and f control the shape and orientation of $F(x, y, z)$ via θ, φ, σ_x, σ_y and σ_z as given:

$$a = \frac{sin^2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin^2\theta sin^2\varphi}{\sigma_y^2} + \frac{cos^2\theta}{\sigma_z^2}$$

$$b = \frac{cos^2\theta cos^2\varphi}{\sigma_x^2} + \frac{cos^2\theta sin^2\varphi}{\sigma_y^2} + \frac{sin^2\theta}{\sigma_z^2}$$

$$c = \frac{sin^2\varphi}{\sigma_x^2} + \frac{cos^2\varphi}{\sigma_y^2}$$

$$d = \frac{sin2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin2\theta sin^2\varphi}{\sigma_y^2} - \frac{sin2\theta}{\sigma_z^2}$$

$$e = -\frac{cos\theta sin2\varphi}{\sigma_x^2} + \frac{cos\theta sin2\varphi}{\sigma_y^2}$$

$$f = -\frac{sin\theta sin2\varphi}{\sigma_x^2} + \frac{sin\theta sin2\varphi}{\sigma_y^2}, \qquad (5.2)$$

Based on the 5 descriptors θ, φ, σ_x, σ_y and σ_z from blob distribution, a FDD distance metric is proposed to measure the difference between each pair of images.

5.2.2 Fréchet Descriptor Distance

Fréchet Distance (FD) is used to measure the similarity between curves (Aronov et al., 2006) and has been used in drug discovery (Preuer et al., 2018) and video applications (Unterthiner et al., 2018), and others. In deep learning research, FD on Inception, termed Fréchet Inception Distance (FID) (Heusel et al., 2017) is commonly used in GAN models. To compute FID, an Inception model $\mathcal{I}(\cdot)$ needs to be pre-trained, the features from the penultimate layer of Inception model are extracted. Let $I_f: R^3 \rightarrow R$ be a 3D input image

and assume these features from I_f come from a multivariate Gaussian, the image features can be represented as:

$$\mathcal{I}(I_f;\Theta)\sim\mathcal{N}_f(\mu_f,\sum_f), \tag{5.3}$$

where Θ is the parameter set of Inception model, μ_f and $\sum_f$ are mean and covariance matrices of multivariate Gaussian $\mathcal{N}_f$. Let the image features on a 3D real image I_r: $R^3 \rightarrow R$ be $\mathcal{I}(I_r;\Theta)\sim N_r(\mu_r,\sum_r)$ and the image features on a 3D synthesized image I_s: $R^3 \rightarrow R$ be $\mathcal{I}(I_s;\Theta)\sim N_s(\mu_s,\sum_s)$, FID is defined as:

$$d_{FID}(N_r(\mathcal{I}(I_r;\Theta)),N_s(\mathcal{I}(I_s;\Theta))) = \|\mu_r-\mu_s\|_2^2 + tr(\sum_r+\sum_s-2\sqrt{\sum_r\sum_s}), \tag{5.4}$$

where tr is the trace of matrix. While FID is mainly used to evaluate the quality or effectiveness of GAN by measuring the similarity between real images and synthesized image, it has not been used as a distance metric in Coreset mainly due to its computing costs. The computational complexity of FID is $\mathcal{O}(\mathcal{I}(I_r;\Theta)+\mathcal{I}(I_s;\Theta)+d_{FID})$. It is known computing Θ and deep layers in Inception model is costly with largest portion of computation dedicated for Inception model, that is, $\mathcal{O}\big(\mathcal{I}(I_r;\Theta)+\mathcal{I}(I_s;\Theta)\big) >> \mathcal{O}(d_{FID})$.

To accelerate GAN training, we propose Fréchet Descriptor Distance (FDD) as an alternative distance metric. Compared to FID, FDD is directly derived from object related statistics to measure the distances between the image pairs. In ***II.A***, we synthesize 3D blobs using 3D elliptical Gaussian function and we know that each blob has 5 descriptors: θ, φ, σ_x, σ_y and σ_z. These blob descriptors are object related statistics in 3D blob images. Assume each descriptor follows a Gaussian and a 3D blob image consist of M blobs, the descriptors set $\Omega=\{\theta,\varphi,\sigma_x,\sigma_y,\sigma_z\}_{i\in M}$ follows a multivariate Gaussian. Let the descriptor set on a 3D blob image I_b: $R^3 \rightarrow R$ be $\Omega_b\sim N_b(\mu_b,\sum_b)$, FDD for 2 pair wisely compared 3D blob images I_{b1} and I_{b2} is defined as:

$$d_{FDD}(N_{b1}(\Omega_{b1}), N_{b2}(\Omega_{b2})) = \|\mu_{b1} - \mu_{b2}\|_2^2 + tr(\sum_{b1} + \sum_{b2} - 2\sqrt{\sum_{b1}\sum_{b2}}). \quad (5.5)$$

The computational complexity of FDD is $\mathcal{O}(d_{FDD})$. $\mathcal{O}(d_{FDD})$ is a function of M, the number of blobs and $\mathcal{O}(d_{FID})$ is a function of the image features from Inception model, for example, 64. We conclude $\mathcal{O}(d_{FDD}) \approx \mathcal{O}(d_{FID})$, $\mathcal{O}\big(\mathcal{I}(I_r;\Theta) + \mathcal{I}(I_s;\Theta)\big) >> \mathcal{O}(d_{FID}) \approx \mathcal{O}(d_{FDD})$, thus $\mathcal{O}(\mathcal{I}(I_r;\Theta) + \mathcal{I}(I_s;\Theta) + d_{FID}) >> \mathcal{O}(d_{FDD})$, FDD is more computationally efficient for Coreset selection.

5.2.3 Dataset Sampling based on Coreset

Models based on Coreset will provide approximate performance of using the whole dataset. Given an image dataset D, a Coreset C of D is a subset $C \subset D$ that approximates the "shape" of D. Let the cost function of model be $\mathcal{L}(\cdot)$, the objective function to select Coreset can be represented as:

$$\min_{C:|C|=k} |\mathcal{L}(D) - \mathcal{L}(C)|, \quad (5.6)$$

where k is desired size of C. equation (5.6) indicates the performance under the selected Coreset C with size k to be as close as possible with the performance under the whole dataset D. We formulate it as a k-center problem (minimax facility location (Wolf, 2011)) to choose k core sample images such that the largest distance between a sample image with its nearest center is minimized. This is represented as:

$$\min_{C:|C|=k} \max_{I_i \in D} \min_{I_j \in C} d(I_i, I_j), \quad (5.7)$$

where $d(.,.)$ is a distance metric on D. This is an NP-Hard problem. However, it is possible to obtain an approximation solution efficiently using a greedy approach. Here we use FDD as the distance metric $d(.,.)$ in equation (5.7) and the details of FDD-Coreset is shown in Table 20.

Table 20 Pseudocode for FDD-Coreset

Input: target size k, 3D blob image datasets D ($|D| > k$) with descriptors sets $\{\Omega_p\}_{p\in|D|}$
Output: Coreset C ($|C| = k$)
1. **Initialize** Coreset $C = \{\}$
2. **While** $|C| < k$:
3. **For** 3D blob image I_i from $D\backslash C$:
4. **Extract** the blob descriptors sets Ω_i of I_i
5. **For** 3D blob image I_j from C:
6. **Extract** the blob descriptors sets Ω_j of I_j
7. **Calculate** FDD distance between I_i and I_j: $d_{FDD}(N_i(\Omega_i), N_j(\Omega_j))$
8. **Iteratively** until find sample image $S = \underset{I_i\in D\backslash C}{\operatorname{argmax}} \min_{I_j\in C} d_{FDD}(N_i(\Omega_i), N_j(\Omega_j))$
9. **Add** sample image S in Coreset: $C = C \cup \{S\}$
10. **End While**
11. **Return** Coreset C

5.3 Experiments and Results

5.3.1 Training Dataset

We evaluated our method using two experiments. BlobGAN's training datasets in the source domain and target domain are different. In the source domain, we used the blob images synthesized by the 3D elliptical Gaussian function in Section 5.2 for both experiments. In the target domain, we used simulated noisy images in the first experiment and the real-world 3D human MR images of the kidney in the second experiment.

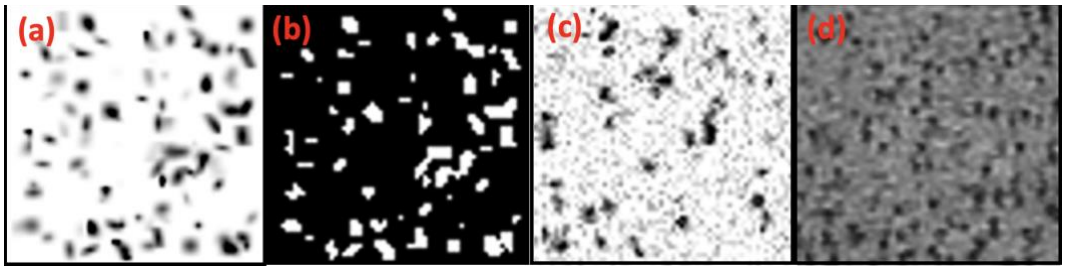


Figure 27 Illustration of Training Input Images of BlobGAN (a) Synthesized 3D Blob Image from Source Domain. (b) Blob Mask of (a) from Source Domain. (c) Synthesized 3D Noisy Blob Image from Target Domain. (d) 3D MR Image of the Human Kidney: Patch from Target Domain.

In the first experiment, we randomly synthesized 1,000 3D blob images (64×64×32 voxels) to construct the source domain of BlobGAN. 3D training image (Figure 27 (a)) blobs were scattered randomly in the image space. These blobs were generated using a random number generator and blobs' number ranged from 500 to 800. The parameters of 3D elliptical Gaussian function for each synthesized blob are as follows: $\theta, \varphi \in [0, 180^{\circ}]$, σ_x, σ_y, $\sigma_z \in [0.5, 1.5]$. The blob mask (Figure 27 (b)) was recorded for each blob image. For the target domain, we synthesized another 1,000 3D blob images using the same 3D blob synthesis function. To simulate noisy 3D blob images (Figure 27 (c)), we added random Gaussian noise to the synthesized images. The noise was generated by the Gaussian function with $\mu_{noise} = 0$ and σ^2_{noise} defined by:

$$\sigma^2{}_{noise} = \frac{\sigma^2 image}{10^{\frac{SNR}{10}}} , \tag{5.8}$$

where the signal-to-noise ratio (SNR) lies in the interval $[0.01dB, 1dB]$. The 1,000 3D blob images and 1,000 3D noisy blob images comprise the training dataset of baseline in the first case study.

In the second experiment, the source domain contains 1,000 3D blob images (64×64×32 voxels). For target domain, we studied three 3D human kidneys MR images. Each human kidney MR image has voxel dimensions of 896×512×512. These three human

kidneys were obtained after autopsy through a donor network (The International Institute for the Advancement of Medicine, Edison, NJ) after receiving Institutional Review Board (IRB) approval and informed consent from Arizona State University (Beeman et al., 2014). They were imaged by as described in (Baldelomar et al., 2016b, 2017; Beeman et al., 2014; Bennett et al., 2008b). To validate our model, each time we trained the BlobGAN on two human kidneys and tested on the other one. We randomly sampled 1,000 3D non-overlapping patch images (64×64×32 voxels, Figure 27 (d)) from the two human kidneys in the training dataset. The sampling process was performed in the cortex region because the medulla region does not have any glomeruli. The medulla and cortex regions of human kidneys were annotated by a domain expert. The resulting 1,000 3D blob images and 1,000 3D human kidney patch images are treated as the training dataset of baseline in the second case study.

5.3.2 Experiment I: Validation Experiments using 3D Synthetic Image Data

To validate the performance of FDD-Coreset on BlobGAN, we synthesized an independent dataset of 1,000 3D blob images ($64 \times 64 \times 32$ voxels) with Gaussian distributed noise as the test data. The version of these images without noise and their blob masks were recorded as ground truth to compare the performance with other approaches. The goal of this experiment is to show that FDD-Coreset is capable of accelerating the training of BlobGAN and meanwhile maintaining the BlobGAN's performance on blob identification. First, we selected k Coreset from these 1,000 3D blob images to train BlobGAN. To train the BlobGAN, λ_{cycle} and λ_{convex} were set to 10, $\lambda_{identity}$ was set to 0.5. BlobGAN was trained using a learning rate of 0.0002 with the Adam optimizer (batch size =1).

We compared the computation time of IED-Coreset (Sinha et al., 2019) with proposed FDD-Coreset under varying sample size k (see Table 21). As shown in Table 21, FDD-Coreset is significantly faster (266 – 42,784 times) than IED-Coreset for size k from 1 to 10. We compared the total computation time from Coreset selection to training Coreset on BlobGAN. We randomly sampled 1,000 3D blob images to train BlobGAN as baseline. We generated Coreset using IED-Coreset (size $k = 10$) and FDD-Coreset (size $k = 10$, 20, 30), respectively and trained these groups of Coreset on BlobGAN by 20 epochs. Note we did not evaluate the Coreset via IED-Coreset with $k = 20$,30. This is because with $k = 10$, IED-Coreset took 42,784 times than the FDD-Coreset with $k = 10$. Once these Coresets were generated, we compared the total computational time from the Coreset generation to the BlobGAN training (see Table 22). It is apparent that BlobGAN trained on IED-Coreset with $k = 10$ takes the longest time. The reason is that the Coreset selection in IED-Coreset takes almost 99% of the whole process. BlobGAN trained on FDD-Coreset with $k = 10$, 20 and 30 all perform significantly faster (31 – 87 times) than training on entire dataset (1,000 random samples). This is because the FDD-Coreset reduces the entire 1,000 3D images to k core images which significantly reduces the BlobGAN training time. BlobGAN trained on FDD-Coreset with $k = 10$, 20 and 30 perform significantly faster (189 – 530 times) than training on IED-Coreset with $k = 10$. The reason is that the IED-Coreset calculated the distance between images based on Inception embedding from images and there exists duplicated computation in the batch selection. We conclude that the total computation time for training BlobGAN on FDD-Coreset with $k = 10$, 20 and 30 is significantly less than entire dataset and IED-Coreset even with $k = 10$.

Table 21 Computation Time Comparison of IED-Coreset and FDD-Coreset on 3D Synthetic Blob Images (unit: second)

K CORESET SAMPLES	IED-CORESET	FDD-CORESET
1	2.66	0.01
2	3,325.27	1.17
3	13,366.92	2.00
4	33,394.27	3.39
5	66,640.30	4.66
6	116,509.83	5.89
7	186,813.33	7.67
8	279,370.76	9.30
9	398,713.87	10.71
10	546,784.21	12.78

Table 22 Computation Time Comparison of BlobGAN Trained on Entire Dataset (1,000 Random Samples), IED-Coreset (k =10) and Proposed FDD-Coreset (k =10, 20, 30) on 3D Synthetic Blob Images (unit: second)

k CORESET SAMPLES	ENTIRE DATASET	IED-CORESET	FDD-CORESET
10	89,927.46	547,825.60	1,032.19
20	89,927.46	--	1,995.25
30	89,927.46	--	2,896.44

We evaluated the model performance (blob identification) using eight metrics: Peak Signal to Noise Ratio (*PSNR*), Detection Error Rate (*DER*), Precision, Recall, F-score, Dice coefficient, Intersection over Union (*IoU*), and Blobness, as follows:

1. Peak Signal to Noise Ratio (*PSNR*) metric is to measure the performance of image denoising. Let the final 3D denoised blob image be x: $R^3 \rightarrow R$ and the 3D blob image without noises be y: $R^3 \rightarrow R$, then *PSNR* is defined as follows:

$$PSNR = 20log_{10}\frac{MAX_x}{||x-y||_2}, \tag{5.9}$$

where MAX_x is the possible maximum voxel values of x.

2. Detection Error Rate (*DER*) is to measure the difference ratio between the number of detected blobs and the ground truth. *DER* can be calculated by equation (5.10).

$$DER = \frac{|N_{GT}-N_{Det}|}{N_{GT}}, \tag{5.10}$$

where N_{GT} represents the # of ground truth blobs and N_{Det} represents the # of detected blobs.

To avoid duplicate counting, the number (#) of true positives TP was calculated by equation (5.11)

$$TP = min\left\{\#\{(i,j): min_{i=1}^{m} D_{ij} \leq d_\delta\}, \#\{(i,j): min_{j=1}^{n} D_{ij} \leq d_\delta\}\right\}, \tag{5.11}$$

where m is the number of true glomeruli and n is the number of blob candidates; d_δ is a thresholding parameter set to a positive value $(0, +\infty)$. If d_δ is small, fewer blob candidates are counted since the distance between the blob candidate centroid and ground-truth should be small. If d_δ is too large, more blob candidates are counted. A candidate was considered as TP if the centroid of its magnitude was in a detection pair (i, j) for which the nearest ground truth center j had not been paired and the Euclidian distance D_{ij} between ground truth center j and blob candidate i was less than or equal to d_δ. Here, since local intensity extremes could be anywhere within a small blob with an irregular shape, we set d to the average diameter of the blobs: $d_\delta = 2 \times$

$\sqrt{\frac{\sum_{(i,j,k)} f(i,j,k)}{\pi}}$. TP is used for the precision in equation (5.12), recall in equation (5.13) and F-score in equation (5.14).

3. Precision is to measure the fraction of retrieved blobs confirmed by the ground truth

$$Precision = \frac{TP}{n}, \tag{5.12}$$

4. Recall is to measure the fraction of ground-truth data retrieved

$$Recall = \frac{TP}{m}, \tag{5.13}$$

5. F-score is the overall performance of precision and recall

$$F - score = 2 \times \frac{Precision \times Recall}{(Precision + Recall)}, \tag{5.14}$$

6. Dice coefficient is to measure the similarity between the segmented blob mask and the ground truth.

$$Dice\ (BM, BG) = \frac{2|BM \cap BG|}{|BM| + |BG|}, \tag{5.15}$$

where BM is the binary mask for segmentation result and BG is the binary mask for the ground truth.

7. Intersection over Union (*IoU*) is to measure the amount of overlap between the segmented blob mask and the ground truth

$$IoU\ (BM, BG) = \frac{BM \cap BG}{BM \cup BG}, \tag{5.16}$$

where BM is the binary mask for segmentation result and BG is the binary mask for the ground truth.

8. Blobness is to measure the likelihood of the objects with blobs shape. Blobness for each blob candidate bi from blobs set S_{blob} is calculated by equation (5.17):

$$Blobness_{bi \in S_{blob}} = \frac{3 \times |\det(H(J-f))|^{\frac{2}{3}}}{pm(H(J-f))}, \quad (5.17)$$

where f is the normalized 3D dark blob image, H represents the Hessian matrix and pm represents the principal minors of Hessian matrix.

The performance comparison between BlobGAN trained on entire dataset (1,000 random samples), IED-Coreset ($k = 10$), FDD-Coreset ($k = 10, 20, 30$) are shown in Table 23. Compared to the BlobGAN trained on entire dataset, BlobGAN trained on FDD-Coreset ($k = 20$) provides better performance on *PSNR* and BlobGAN trained on FDD-Coreset ($k = 30$) provides better performance on *DER*, comparable performance on Recall and F-score. BlobGAN trained on IED-Coreset ($k = 10$) and FDD-Coreset ($k = 10, 20, 30$) all give the lower performance on Dice and IoU than BlobGAN trained on entire dataset but FDD-Coreset ($k = 30$) gives the closest performance. BlobGAN trained on FDD-Coreset ($k = 10$) has the closest Blobness value with the BlobGAN trained on entire dataset and the ground truth. We conclude that BlobGAN trained on FDD-Coreset provides comparable performance to the model trained on entire dataset.

Table 23 Performance Comparison (Avg ± Std) of BlobGAN Trained on Entire Dataset (1,000 Random Samples), IED-Coreset (k =10) and Proposed FDD-Coreset (k =10, 20, 30) on 3D Synthetic Images

METRICS	ENTIRE DATASET	IED-CORESET (k =10)	FDD-CORESET (k = 10)	FDD-CORESET (k = 20)	FDD-CORESET (k = 30)
PEAK SIGNAL TO NOISE RATIO (*PSNR*)	12.539±0.143	13.000±0.753	12.084±0.423	15.758±0.373	11.489±0.216
DETECTION ERROR RATE (*DER*)	0.091±0.037	0.124±0.131	0.171±0.156	0.219±0.061	0.052±0.037

PRECISION	0.941±0.013	0.759±0.069	0.671±0.056	0.923±0.018	0.827±0.024
RECALL	0.855±0.033	0.781±0.052	0.771±0.042	0.720±0.049	0.845±0.032
F-SCORE	0.896±0.018	0.766±0.030	0.715±0.028	0.807±0.029	0.835±0.014
DICE	0.825±0.017	0.548±0.046	0.662±0.014	0.502±0.039	0.671±0.022
IoU	0.702±0.024	0.379±0.044	0.495±0.015	0.336±0.035	0.505±0.025
BLOBNESS (GROUND TRUTH: 0.519)	0.538±0.279	0.609±0.302	0.577±0.308	0.583±0.290	0.610±0.298

5.3.3 Experiment II: Validation Experiments using 3D Human Kidney MR Images

In this experiment, we investigated the proposed FDD-Coreset approach on 3D MR images to measure the number (N_{glom}) and apparent volume (aV_{glom}) of glomeruli in healthy and diseased human donor kidneys that were not accepted for transplant. Since we have three 3D MR human kidney images and each of them has voxel dimensions of 896×512×512 and the input of BlobGAN is a 3D patch with voxel dimensions of 64×64×32, we divided each human kidney into 1,792 3D patches (64×64×32) to validate the performance of BlobGAN. The final identification mask of the whole kidney was reconstructed by stacking all 3D patches. To train the BlobGAN, λ_{cycle} and λ_{convex} were set to 10, $\lambda_{identity}$ was set to 0.5, learning rate was 0.0002, Adam optimizer with a batch size set to 1.

We compared the total computation time from Coreset selection to training Coreset on BlobGAN on 3D MR human kidney images. Same with Section 5.3.2, we have original random sampled 1,000 3D blob images to train BlobGAN as baseline, so we generated Coreset (size $k = 10$) through IED-Coreset and FDD-Coreset (size $k = 10$), respectively and trained these groups of Coreset on BlobGAN by 20 epochs. Once these Coreset are

generated, we compare the total computational time from the Coreset generation to the BlobGAN training and summarize them in the Table 24, from where, we can see the results have shown that BlobGAN trained on IED-Coreset with $k = 10$ on three human kidneys takes the longest time. BlobGAN trained on FDD-Coreset with $k = 10$ on three kidneys all perform significantly faster (76 times) than training on entire dataset (1,000 random samples). BlobGAN trained on FDD-Coreset with $k = 10$ on three kidneys all perform significantly faster (460 times) than training on IED-Coreset with $k = 10$. We can conclude that training BlobGAN on FDD-Coreset with $k = 10$ on 3D MR human kidney images is significantly computationally efficient than entire dataset and IED-Coreset with $k = 10$ and this is consistent with the comparison of computation time in Section 5.3.2.

Table 24 Computational Time Comparison of BlobGAN Trained on Entire Dataset (1,000 Random Samples), IED-Coreset (k =10) and Proposed FDD-Coreset (k =10) on 3D Human Kidney MR Images (unit: second)

HUMAN KIDNEY	ENTIRE DATASET	IED-CORESET (k =10)	FDD-CORESET (k= 10)
CF1	91,325.20	547,800.21	1,195.58
CF2	91,382.40	547,801.81	1,180.38
CF3	91,515.60	547,772.61	1,199.58

N_{glom} and mean aV_{glom} are reported in Table 25 and Table 26, where BlobGAN trained on entire dataset, BlobGAN trained on IED-Coreset with $k = 10$ and BlobGAN trained on FDD-Coreset with $k = 10$ are compared to the data from unbiased dissector-

fractionator stereology (Beeman et al., 2011). We used these stereology data from (Beeman et al., 2014) as ground truth and calculated aV_{glom} based on the method from (Baldelomar et al., 2016b). The differences between the results of these three training approaches and stereology data are also listed in Table 25 and Table 26. As shown in Table 25 and Table 26, N_{glom} and mean aV_{glom} derived from BlobGAN trained on FDD-Coreset with $k = 10$ provides smaller difference ratio with stereology than BlobGAN trained on IED-Coreset with $k = 10$. While BlobGAN trained on entire dataset has smallest difference ratio with stereology, it takes more than 96% computational time than FDD-Coreset with $k = 10$, as shown in Table 5. We could conclude FDD-Coreset not only significantly reduces the training time, but also maintain approximate performance of glomerular segmentation compared with training on entire dataset.

Table 25 Human Kidney Glomerular Segmentation Results (N_{glom}) using BlobGAN Trained on Entire Dataset (1,000 Random Samples), IED-Coreset (k =10) and Proposed FDD-Coreset (k =10) Comparing with Stereology

HUMAN KIDNEY	N_{glom} ($\times 10^6$) (STEREOLOGY)	N_{glom} ($\times 10^6$) (ENTIRE DATASET)	Difference Ratio (%)	N_{glom} ($\times 10^6$) (IED-CORESET (k =10))	Difference Ratio (%)	N_{glom} ($\times 10^6$) (FDD-CORESET (k =10))	Difference Ratio (%)
CF 1	1.13	1.05	**7.08**	1.58	**39.82**	1.26	**11.50**
CF 2	0.74	0.71	**4.05**	0.96	**29.73**	0.78	**5.41**
CF 3	1.46	1.48	**1.37**	1.77	**21.23**	1.53	**4.79**

Table 26 Human Kidney Glomerular Segmentation Results (Mean aV_{glom}) using BlobGAN Trained on Entire Dataset (1,000 Random Samples), IED-Coreset (k =10) and Proposed FDD-Coreset (k =10) Comparing with Stereology

HUMAN KIDNEY	Mean aV_{glom} ($\times 10^{-3}mm^3$) (STEREOLOGY)	Mean aV_{glom} ($\times 10^{-3}mm^3$) (ENTIRE DATASET)	Difference Ratio (%)	Mean aV_{glom} ($\times 10^{-3}mm^3$) (IED-CORESET (k =10)	Difference Ratio (%)	Mean aV_{glom} ($\times 10^{-3}mm^3$) (FDD-CORESET (k =10))	Difference Ratio (%)
CF 1	5.01	5.19	**3.59**	4.48	**10.58**	5.20	**3.79**
CF 2	4.68	4.80	**2.56**	3.61	**22.86**	5.02	**7.26**
CF 3	2.82	2.81	**0.35**	2.25	**20.21**	2.93	**3.90**

For illustration, example results from CF1 are shown in Figure 28. As seen, BlobGAN trained on FDD-Coreset have the more similar glomerular segmentation results with the BlobGAN trained on the entire dataset (1,000 random samples) than BlobGAN trained on IED-Coreset.

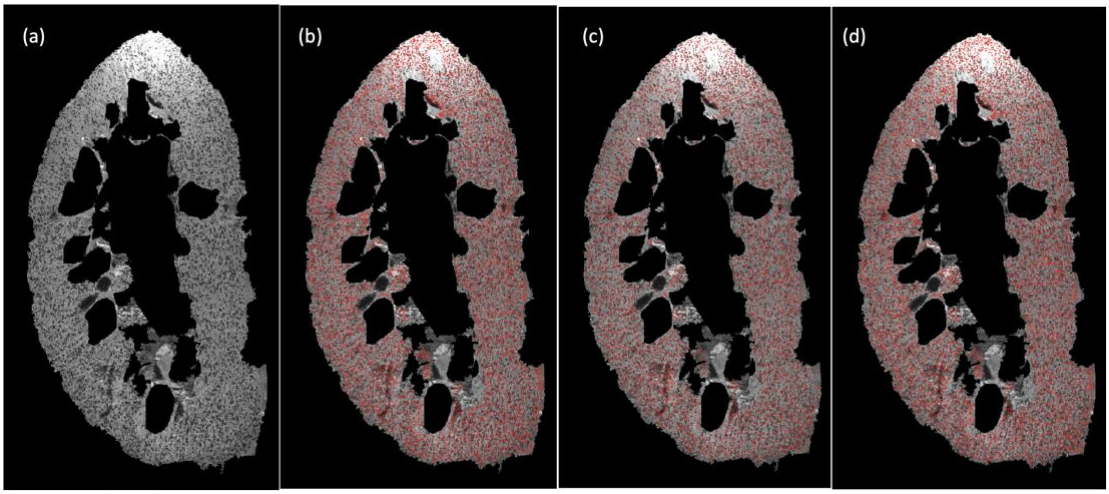


Figure 28 Glomerular Segmentation Results from 3D MR Images of Human Kidney (CF1 Slice 256). (a) Original Magnitude Image. (b) Glomerular Segmentation Results of BlobGAN Trained on Entire Dataset (1,000 Random Samples). (c) Glomerular Segmentation Results of BlobGAN Trained on IED-Coreset (k = 10). (d) Glomerular Segmentation Results of BlobGAN Trained on FDD-Coreset (k =10).

5.4 Conclusion

In this research, a FDD-Coreset approach is proposed to select subset samples used in GAN training to address the computational challenges. A Fréchet Descriptor Distance (FDD) is first proposed to measure the difference between each pair of blob images. Second, we select the Coreset from the entire dataset using FDD metric. We conducted two experiments. In the first experiment, we evaluated the performance of FDD-Coreset on BlobGAN using a set of 3D synthetic blob images. Computational time and eight performance metrics including Peak Signal to Noise Ratio (*PSNR*), Detection Error Rate (*DER*), Precision, Recall, F-score, Dice coefficient, Intersection over Union (*IoU*), and Blobness, were used to compare the performance of FDD-Coreset with entire dataset (1,000 random samples) and IED-Coresets. Compared to training on entire dataset, BlobGAN trained on FDD-Coreset greatly achieve over 96% decrease in computational time, over 25% increase in *PSNR*, over 40% decrease in *DER* and provides comparable performance in blob detection (Precision, Recall, F-score), blob segmentation (Dice, *IoU*), blob synthesis (Blobness). Three 3D MR human kidney images were studied in the second experiment. Compared to training on entire dataset, BlobGAN trained on FDD-Coreset greatly achieve over 96% decrease in computational time and comparable performance in N_{glom} and aV_{glom}. From the results of the two experiments, we conclude that BlobGAN trained on proposed FDD-Coreset significantly reduces the training time of BlobGAN, achieves higher denoising performance (*PSNR*) and maintains approximate performance of blob identification compared to training on entire dataset.

While the results of this study are encouraging, there is room for improvement. The proposed FDD is derived from object statistics, which are blob descriptors from blob distribution. This could be potentially used for accelerating other object focused image

translation model where the shape of objects (blob, nuclei, glomeruli, etc.) follow Gaussian distribution. However, if the shape of objects (tumor, organ, etc.) is irregular and the distribution is unknown, the proposed FDD metric will not be applicable. It is our plan to explore distribution agnostic based Coreset approach as the next step. In addition, the blob images studied in this research have large number of objects to derive the statistic descriptors. Some other medical image applications may have smaller number of objects (e.g., tumors, lesions) to be investigated. We plan to study other discriminative features instead of shape as object statistics to increase the effectiveness of Coreset.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The overall objective of this research is to develop novel computational algorithms for imaging biomarker identification using kidney MRI. I first develop a novel small blob detector, UH-DoG, which joins the Hessian convexity map and probability map from U-Net (Xu et al., 2020). This joint constraint-based approach alleviates the under-segmentation issue from U-Net and the over-detection issue from Hessian analysis. While UH-DoG is successfully implemented in segmenting the 3D mouse kidneys glomeruli using MR images, there are limitations. First, a single threshold applied to U-Net probability map may not be sensitive to noise. Second, a single optimal scale applied to HDoG may overlook large variations in blobs size. One possible solution is to exhaustively explore multiple thresholds in U-Net and multiple scales in DoG, but the massive number of glomeruli (> 1 million in a human kidney) with varying sizes makes such attempts computationally prohibitive. To solve this problem, I propose an extension of UH-DoG, a small blob detector using Bi-Threshold Constrained Adaptive Scales (BTCAS) (Xu et al., 2021). This multi-thresholds, multi-scales approach identifies the relationship between U-Net threshold and DoG scale and addresses computational challenge from UH-DoG with improved performance. Specifically, the local optimum DoG scale is adapted to the range of blob sizes to better separate touching blobs thus under-segmentation issue of U-Net is addressed. Its capability of segmenting blobs with varied sizes and decreasing under-segmentation has been demonstrated through both 3D synthetic blob images and human kidney MR images. By adopting BTCAS, the

computational efficiency and accuracy of segmentation are both improved. However, U-Net in UH-DoG and BTCAS are pre-trained on publicly available optical images of cell nuclei. While UH-DoG and BTCAS detect blobs to some extent, there are several drawbacks of training U-Net on public datasets. First, public datasets are from different imaging modalities and therefore have different intensity distributions than the target datasets. Consequently, pre-trained models require additional ad-hoc approaches to fine-tune the model parameters. Second, small blobs representing structures such as cells, nuclei, and glomeruli have distinct geometric properties. If public and target datasets have different geometric properties, pre-trained models may suffer from the adaption to new domain dataset. Third, public datasets have different noise distributions compared to the target datasets, making it challenging to filter noise during segmentation. To address these limitations from training U-Net and optimally denoising blob images, I develop a denoising Convexity-Consistent Generative Adversarial Network, BlobGAN, for improved small blob identification. This blob detector could achieve high denoising performance and selectively denoise the image without affecting the properties of blobs. However, BlobGAN trained on 1,000 3D blob images takes ~26 hours for 50 epochs, which is computational expensive. To improve the training time of BlobGAN, I propose a Fréchet Descriptors Distance based Coreset approach, FDD-Coreset, for accelerating BlobGAN's training. BlobGAN trained on proposed FDD-Coreset not only significantly reduces the training time, but also achieves higher denoising performance and is able to maintain approximate performance of blob identification compared with training on entire dataset.

6.2 Discussion and Future work

The proposed BlobGAN detector assumes that the blobs from 3D noisy images satisfy the convexity. It might not be capable of detecting those blobs without such geometric property. Especially for CFE-MRI, varied parameters and artifacts affect blobs' shape during the imaging acquisition. In the future, we plan to generate more realistic 3D blob images by integrating physics-based constraints into the GAN architecture. Second, the algorithms developed in this dissertation still have potentials for further computational efficiency improvement. Considering BlobGAN, the two generators and two discriminators are not fully utilized and redundant in training. If we could reduce the number of generators and discriminators while maintain its performance, the training time of BlobGAN will be further reduced. Additionally, there are also some potential works related to FDD-Coresets which are yet to be done. The proposed FDD is derived from object statistics, which are blob descriptors from blob distribution. This could be potentially used for accelerating other object focused image translation model where the shape of objects (tumor, organ, etc.) is irregular and their distribution is unknown. In the future, we plan to explore distribution agnostic based Coreset approach to accelerate training GAN without the need of the object distribution as a prior. Second, the number of objects may not be large in other applications. This will reduce the sample size of object statistics and thus reduce the performance of Coreset. A possible solution is to find discriminative features as object statistics to increase the effectiveness of Coreset. Third, blob descriptors are derived from clean blob images, which only consist of blobs. However, the images may contain noise or other objects, as a context in the image, and these contexts may be needed in the training to assist image translation. This constraint may be resolved

by combining the context statistics and object statistics together as descriptors to derive FDD-Coreset.

REFERENCES

Abramson, R. G., Burton, K. R., Yu, J. P. J., Scalzetti, E. M., Yankeelov, T. E., Rosenkrantz, A. B., Mendiratta-Lala, M., Bartholmai, B. J., Ganeshan, D., Lenchik, L., & Subramaniam, R. M. (2015). Methods and Challenges in Quantitative Imaging Biomarker Development. In *Academic Radiology*. https://doi.org/10.1016/j.acra.2014.09.001

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). FREAK: Fast retina keypoint. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2012.6247715

Al-Kofahi, Y., Lassoued, W., Lee, W., & Roysam, B. (2010). Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*. https://doi.org/10.1109/TBME.2009.2035102

Anoosheh, A., Agustsson, E., Timofte, R., & van Gool, L. (2018). ComboGAN: Unrestrained Scalability for Image Domain Translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 896–8967. https://doi.org/10.1109/CVPRW.2018.00122

Aronov, B., Har-Peled, S., Knauer, C., Wang, Y., & Wenk, C. (2006). *Fréchet Distance for Curves, Revisited* (pp. 52–63). https://doi.org/10.1007/11841036_8

Baldelomar, E. J., Charlton, J. R., Beeman, S. C., & Bennett, K. M. (2017). Measuring rat kidney glomerular number and size in vivo with MRI. *American Journal of Physiology-Renal Physiology*, *314*(3), F399–F406. https://doi.org/10.1152/ajprenal.00399.2017

Baldelomar, E. J., Charlton, J. R., Beeman, S. C., Hann, B. D., Cullen-McEwen, L., Pearl, V. M., Bertram, J. F., Wu, T., Zhang, M., & Bennett, K. M. (2016a). Phenotyping by magnetic resonance imaging nondestructively measures glomerular number and volume distribution in mice with and without nephron reduction. *Kidney International*. https://doi.org/10.1038/ki.2015.316

Baldelomar, E. J., Charlton, J. R., Beeman, S. C., Hann, B. D., Cullen-McEwen, L., Pearl, V. M., Bertram, J. F., Wu, T., Zhang, M., & Bennett, K. M. (2016b). Phenotyping by magnetic resonance imaging nondestructively measures glomerular number and volume distribution in mice with and without nephron reduction. *Kidney International*. https://doi.org/10.1038/ki.2015.316

Baldelomar, E. J., Charlton, J. R., DeRonde, K. A., & Bennett, K. M. (2019). In vivo measurements of kidney glomerular number and size in healthy and Os/+ mice using MRI. *American Journal of Physiology - Renal Physiology*. https://doi.org/10.1152/ajprenal.00078.2019

Batson, J., & Royer, L. (2019). Noise2Seif: Blind denoising by self-supervision. *36th International Conference on Machine Learning, ICML 2019*.

Bay, H., Ess, A., Tuytelaars, T., & van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*. https://doi.org/10.1016/j.cviu.2007.09.014

Beeman, S. C., Cullen-McEwen, L. A., Puelles, V. G., Zhang, M., Wu, T., Baldelomar, E. J., Dowling, J., Charlton, J. R., Forbes, M. S., Ng, A., Wu, Q. -z., Armitage, J. A., Egan, G. F., Bertram, J. F., & Bennett, K. M. (2014). MRI-based glomerular morphology and pathology in whole human kidneys. *AJP: Renal Physiology*. https://doi.org/10.1152/ajprenal.00092.2014

Beeman, S. C., Zhang, M., Gubhaju, L., Wu, T., Bertram, J. F., Frakes, D. H., Cherry, B. R., & Bennett, K. M. (2011). Measuring glomerular number and size in perfused kidneys using MRI. *AJP: Renal Physiology*, *300*(6), F1454–F1457. https://doi.org/10.1152/ajprenal.00044.2011

Bennett, K. M., Zhou, H., Sumner, J. P., Dodd, S. J., Bouraoud, N., Doi, K., Star, R. A., & Koretsky, A. P. (2008a). MRI of the basement membrane using charged nanoparticles as contrast agents. *Magnetic Resonance in Medicine*. https://doi.org/10.1002/mrm.21684

Bennett, K. M., Zhou, H., Sumner, J. P., Dodd, S. J., Bouraoud, N., Doi, K., Star, R. A., & Koretsky, A. P. (2008b). MRI of the basement membrane using charged nanoparticles as contrast agents. *Magnetic Resonance in Medicine*. https://doi.org/10.1002/mrm.21684

Bergmeir, C., García Silvente, M., & Benítez, J. M. (2012). Segmentation of cervical cell nuclei in high-resolution microscopic images: A new algorithm and a web-based software framework. *Computer Methods and Programs in Biomedicine*. https://doi.org/10.1016/j.cmpb.2011.09.017

Bertram, J. F. (1995). Analyzing Renal Glomeruli with the New Stereology. *International Review of Cytology*. https://doi.org/10.1016/S0074-7696(08)62497-3

Bertram, J. F., Soosaipillai, M. C., Ricardo, S. D., & Ryan, G. B. (1992). Total numbers of glomeruli and individual glomerular cell types in the normal rat kidney. *Cell & Tissue Research*. https://doi.org/10.1007/BF00381877

Chang, S., Chen, X., Duan, J., & Mou, X. (2021). A CNN-Based Hybrid Ring Artifact Reduction Algorithm for CT Images. *IEEE Transactions on Radiation and Plasma Medical Sciences*, *5*(2), 253–260. https://doi.org/10.1109/TRPMS.2020.2983391

Charlton, J. R., Xu, Y., Wu, T., deRonde, K. A., Hughes, J. L., Dutta, S., Oxley, G. T., Cwiek, A., Cathro, H. P., Charlton, N. P., Conaway, M. R., Baldelomar, E. J., Parvin, N., & Bennett, K. M. (2020). Magnetic resonance imaging accurately tracks kidney pathology and heterogeneity in the transition from acute kidney injury to chronic kidney disease. *Kidney International*. https://doi.org/10.1016/j.kint.2020.08.021

Chiang, H. T., Hsieh, Y. Y., Fu, S. W., Hung, K. H., Tsao, Y., & Chien, S. Y. (2019). Noise Reduction in ECG Signals Using Fully Convolutional Denoising Autoencoders. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2912036

Chiang, T. C., Huang, Y. S., Chen, R. T., Huang, C. S., & Chang, R. F. (2019). Tumor detection in automated breast ultrasound using 3-D CNN and prioritized candidate aggregation. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2018.2860257

Ciresan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images. *International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, Heidelberg*. https://doi.org/10.1007/978-3-642-40763-5_51

Crocker, J. C., & Grier, D. G. (1996). Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science*. https://doi.org/10.1006/jcis.1996.0217

Cullen-Mcewen, L. A., Armitage, J. A., Nyengaard, J. R., & Bertram, J. F. (2012). Estimating nephron number in the developing kidney using the physical disector/fractionator combination. *Methods in Molecular Biology*. https://doi.org/10.1007/978-1-61779-851-1_10

Dalle, J., Racoceanu, D., & Putti, T. C. (2009). Nuclear Pleomorphism Scoring by Selective Cell Nuclei Detection. *IEEE Workshop on Applications of Computer Vision*.

DeVries, T., Drozdzal, M., & Taylor, G. W. (2020). *Instance Selection for GANs*.

Dolz, J., Gopinath, K., Yuan, J., Lombaert, H., Desrosiers, C., & ben Ayed, I. (2019). HyperDense-Net: A Hyper-Densely Connected CNN for Multi-Modal Image Segmentation. *IEEE Transactions on Medical Imaging*, *38*(5), 1116–1126. https://doi.org/10.1109/TMI.2018.2878669

Dou, Q., Chen, H., Yu, L., Qin, J., & Heng, P. A. (2017). Multilevel Contextual 3-D CNNs for False Positive Reduction in Pulmonary Nodule Detection. *IEEE Transactions on Biomedical Engineering*. https://doi.org/10.1109/TBME.2016.2613502

Esser, P., Sutter, E., & Ommer, B. (2018). A Variational U-Net for Conditional Appearance and Shape Generation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2018.00923

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. https://doi.org/10.1038/nature21056

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T. L., Prinz, M., Palme, K., Simons, M., … Ronneberger, O. (2019). U-Net: deep learning for cell counting, detection, and morphometry. *Nature Methods*. https://doi.org/10.1038/s41592-018-0261-2

Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998). *Multiscale vessel enhancement filtering* (pp. 130–137). https://doi.org/10.1007/BFb0056195

Gao, F., Wu, T., Chu, X., Yoon, H., Xu, Y., & Patel, B. (2019). Deep Residual Inception Encoder-Decoder Network for Medical Imaging Synthesis. *IEEE Journal of Biomedical and Health Informatics*. https://doi.org/10.1109/JBHI.2019.2912659

Gao, F., Wu, T., Li, J., Zheng, B., Ruan, L., Shang, D., & Patel, B. (2018). SD-CNN: A shallow-deep CNN for improved breast cancer diagnosis. *Computerized Medical Imaging and Graphics*. https://doi.org/10.1016/j.compmedimag.2018.09.004

Gao, F., Xu, Y., Panda, A., Zhang, M., Hanson, J., Su, C., Wu, T., Pavlicek, W., & James, J. R. (2017). MR efficiency using automated MRI-desktop eProtocol. *Medical Imaging 2017: Imaging Informatics for Healthcare, Research, and Applications*. https://doi.org/10.1117/12.2249712

Gao, F., Yoon, H., Xu, Y., Goradia, D., Luo, J., Wu, T., & Su, Y. (2020). AD-NET: Age-adjust neural network for improved MCI to AD conversion prediction. *NeuroImage: Clinical*. https://doi.org/10.1016/j.nicl.2020.102290

Gao, F., Zhang, M., Wu, T., & Bennett, K. M. (2016). 3D small structure detection in medical image using texture analysis. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. https://doi.org/10.1109/EMBC.2016.7592201

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gu, J., Yang, T. S., Ye, J. C., & Yang, D. H. (2021). CycleGAN denoising of extreme low-dose cardiac CT using wavelet-assisted noise disentanglement. *Medical Image Analysis*, *74*, 102209. https://doi.org/10.1016/j.media.2021.102209

Gu, J., & Ye, J. C. (2021). AdaIN-Based Tunable CycleGAN for Efficient Unsupervised Low-Dose CT Denoising. *IEEE Transactions on Computational Imaging*, *7*, 73–85. https://doi.org/10.1109/TCI.2021.3050266

Guo, Y., Wang, K., Yang, S., Wang, Y., Gao, P., Xie, G., Lv, C., & Lv, B. (2019). *Structure-Aware Noise Reduction Generative Adversarial Network for Optical Coherence Tomography Image* (pp. 9–17). https://doi.org/10.1007/978-3-030-32956-3_2

Guo, Y., Xu, X., Wang, Y., Wang, Y., Xia, S., & Yang, Z. (2014). An image processing pipeline to detect and segment nuclei in muscle fiber microscopic images. *Microscopy Research and Technique*. https://doi.org/10.1002/jemt.22373

Gupta, H., Jin, K. H., Nguyen, H. Q., McCann, M. T., & Unser, M. (2018). CNN-Based Projected Gradient Descent for Consistent CT Image Reconstruction. *IEEE Transactions on Medical Imaging*, *37*(6), 1440–1453. https://doi.org/10.1109/TMI.2018.2832656

Han, C., Hayashi, H., Rundo, L., Araki, R., Shimoda, W., Muramatsu, S., Furukawa, Y., Mauri, G., & Nakayama, H. (2018). GAN-based synthetic brain MR image generation. *Proceedings - International Symposium on Biomedical Imaging*. https://doi.org/10.1109/ISBI.2018.8363678

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*.

Hiasa, Y., Otake, Y., Takao, M., Matsuoka, T., Takashima, K., Carass, A., Prince, J. L., Sugano, N., & Sato, Y. (2018). *Cross-Modality Image Synthesis from Unpaired Data Using CycleGAN* (pp. 31–41). https://doi.org/10.1007/978-3-030-00536-8_4

Ho, D. J., Fu, C., Salama, P., Dunn, K. W., & Delp, E. J. (2018). Nuclei detection and segmentation of fluorescence microscopy images using three dimensional convolutional neural networks. *Proceedings - International Symposium on Biomedical Imaging*. https://doi.org/10.1109/ISBI.2018.8363606

Hoo-Chang Member, S., Roth hoochangshin, H. R., Gao, M., Lu Senior Member, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R. M., Shin, H.-C., Roth, H. R., & Lu, L. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning and Daniel Mollura are with Center for Infectious Disease Imaging HHS Public Access. *IEEE Trans Med Imaging*, *35*(5), 1285–1298. https://doi.org/10.1109/TMI.2016.2528162

Huang, L., Jiang, S., & Vishnoi, N. (2019). Coresets for clustering with fairness constraints. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 13285–13296.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Janowczyk, A., & Madabhushi, A. (2016a). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*, *7*(1), 29. https://doi.org/10.4103/2153-3539.186902

Janowczyk, A., & Madabhushi, A. (2016b). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*. https://doi.org/10.4103/2153-3539.186902

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). *Perceptual Losses for Real-Time Style Transfer and Super-Resolution* (pp. 694–711). https://doi.org/10.1007/978-3-319-46475-6_43

Kamnitsas, K., Ledig, C., Newcombe, V. F. J., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., & Glocker, B. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, *36*, 61–78. https://doi.org/10.1016/j.media.2016.10.004

Kashif, M. N., Raza, S. E. A., Sirinukunwattana, K., Arif, M., & Rajpoot, N. (2016). Handcrafted features with convolutional neural networks for detection of tumor cells in histology images. *Proceedings - International Symposium on Biomedical Imaging*. https://doi.org/10.1109/ISBI.2016.7493441

Khoshdeli, M., & Parvin, B. (2018). Feature-Based Representation Improves Color Decomposition and Nuclear Detection Using a Convolutional Neural Network. *IEEE Transactions on Biomedical Engineering*. https://doi.org/10.1109/TBME.2017.2711529

Koenderink, J. J. (1984). The structure of images. *Biological Cybernetics*. https://doi.org/10.1007/BF00336961

Komatsu, R., & Gonsalves, T. (2019). *EFFECTIVENESS OF U-NET IN DENOISING RGB IMAGES*. https://doi.org/10.5121/csit.2019.90201

Kong, H., Akakin, H. C., & Sarma, S. E. (2013a). A generalized laplacian of gaussian filter for blob detection and its applications. *IEEE Transactions on Cybernetics*. https://doi.org/10.1109/TSMCB.2012.2228639

Kong, H., Akakin, H. C., & Sarma, S. E. (2013b). A generalized laplacian of gaussian filter for blob detection and its applications. *IEEE Transactions on Cybernetics*. https://doi.org/10.1109/TSMCB.2012.2228639

Lee, G., Nho, K., Kang, B., Sohn, K. A., Kim, D., Weiner, M. W., Aisen, P., Petersen, R., Jack, C. R., Jagust, W., Trojanowki, J. Q., Toga, A. W., Beckett, L., Green, R. C., Saykin, A. J., Morris, J., Shaw, L. M., Khachaturian, Z., Sorensen, G., … Fargher, K. (2019). Predicting Alzheimer's disease progression using multi-modal deep learning approach. *Scientific Reports*. https://doi.org/10.1038/s41598-018-37769-z

Lee, H. G., & Lee, S. C. (2017). Nucleus segmentation using Gaussian mixture based shape models. IEEE journal of biomedical and health informatics, 22(1), 235-243.

Lempitsky, V., & Zisserman, A. (2010). Learning To Count Objects in Images. *Neural Information Processing Systems (NIPS)*, 1–9. https://doi.org/10.1111/1467-9280.03439

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2011.6126542

Li, G., Liu, T., Tarokh, A., Nie, J., Guo, L., Mara, A., Holley, S., & Wong, S. T. C. (2007). 3D cell nuclei segmentation based on gradient flow tracking. *BMC Cell Biology*. https://doi.org/10.1186/1471-2121-8-40

Li, W., Li, J., Sarma, K. v., Ho, K. C., Shen, S., Knudsen, B. S., Gertych, A., & Arnold, C. W. (2019). Path R-CNN for Prostate Cancer Diagnosis and Gleason Grading of Histological Images. *IEEE Transactions on Medical Imaging*, *38*(4), 945–954. https://doi.org/10.1109/TMI.2018.2875868

Lindeberg, T. (1993). Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*. https://doi.org/10.1007/BF01469346

Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*. https://doi.org/10.1023/A:1008045108935

Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., & Huisman, H. (2014). Computer-aided detection of prostate cancer in MRI. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2014.2303821

Liu, M., Cheng, D., & Yan, W. (2018). Classification of alzheimer's disease by combination of convolutional and recurrent neural networks using FDG-PET images. *Frontiers in Neuroinformatics*. https://doi.org/10.3389/fninf.2018.00035

Liu, S., Liu, S., Cai, W., Che, H., Pujol, S., Kikinis, R., Feng, D., & Fulham, M. J. (2015). Multimodal Neuroimaging Feature Learning for Multiclass Diagnosis of Alzheimer's Disease. *IEEE Transactions on Biomedical Engineering*. https://doi.org/10.1109/TBME.2014.2372011

Lord, A., Ehrlich, S., Borchardt, V., Geisler, D., Seidel, M., Huber, S., Murr, J., & Walter, M. (2016). Brain parcellation choice affects disease-related topology differences increasingly from global to local network levels. *Psychiatry Research - Neuroimaging*. https://doi.org/10.1016/j.pscychresns.2016.02.001

Lowe, D. G. (2004a). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Lowe, D. G. (2004b). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Loy, G., & Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2003.1217601

Ma, Y., Liu, J., Liu, Y., Fu, H., Hu, Y., Cheng, J., Qi, H., Wu, Y., Zhang, J., & Zhao, Y. (2021). Structure and Illumination Constrained GAN for Medical Image Enhancement. *IEEE Transactions on Medical Imaging*, *40*(12), 3955–3967. https://doi.org/10.1109/TMI.2021.3101937

Mahmood, F., Borders, D., Chen, R., McKay, G. N., Salimian, K. J., Baras, A., & Durr, N. J. (2019). Deep Adversarial Training for Multi-Organ Nuclei Segmentation in Histopathology Images. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/tmi.2019.2927182

Malpica, N., De Solórzano, C. O., Vaquero, J. J., Santos, A., Vallcorba, I., García-Sagredo, J. M., & Del Pozo, F. (1997). Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*. https://doi.org/10.1002/(SICI)1097-0320(19970801)28:4<289::AID-CYTO3>3.0.CO;2-7

Meijering, E., Dzyubachyk, O., Smal, I., & van Cappellen, W. A. (2009). Tracking in cell and developmental biology. In *Seminars in Cell and Developmental Biology*. https://doi.org/10.1016/j.semcdb.2009.07.004

Mikolajczyk, K., & Schmid, C. (2004a). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*. https://doi.org/10.1023/B:VISI.0000027790.02288.f2

Mikolajczyk, K., & Schmid, C. (2004b). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*. https://doi.org/10.1023/B:VISI.0000027790.02288.f2

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2005.188

Moon, W. K., Shen, Y. W., Bae, M. S., Huang, C. S., Chen, J. H., & Chang, R. F. (2013). Computer-aided tumor detection based on multi-scale blob detection algorithm in automated breast ultrasound images. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2012.2230403

Nuha, F. U., & Afiahayati. (2018). Training dataset reduction on generative adversarial network. *Procedia Computer Science*, *144*, 133–139. https://doi.org/10.1016/j.procs.2018.10.513

Otsu, & N. (1996). A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*.

Oulbacha, R., & Kadoury, S. (2020). MRI to CT Synthesis of the Lumbar Spine from a Pseudo-3D Cycle GAN. *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 1784–1787. https://doi.org/10.1109/ISBI45749.2020.9098421

Phan, H. T. H., Kumar, A., Feng, D., Fulham, M., & Kim, J. (2019). Optimizing contextual feature learning for mitosis detection with convolutional recurrent neural networks. *Proceedings - International Symposium on Biomedical Imaging*. https://doi.org/10.1109/ISBI.2019.8759224

Phansalkar, N., More, S., Sabale, A., & Joshi, M. (2011). Adaptive local thresholding for detection of nuclei in diversity stained cytology images. *ICCSP 2011 - 2011 International Conference on Communications and Signal Processing*. https://doi.org/10.1109/ICCSP.2011.5739305

Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., & Klambauer, G. (2018). Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. Journal of chemical information and modeling, 58(9), 1736-1741.

Raza, S. E. A., Abduljabbar, K., Jamal-Hanjani, M., Veeriah, S., Quesne, J. Le, Swanton, C., & Yuan, Y. (2019). Deconvolving convolutional neural network for cell detection. *Proceedings - International Symposium on Biomedical Imaging*. https://doi.org/10.1109/ISBI.2019.8759333

Raza, S. E. A., Cheung, L., Epstein, D., Pelengaris, S., Khan, M., & Rajpoot, N. M. (2017). MIMO-Net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images. *Proceedings - International*

Symposium on Biomedical Imaging, 337–340. https://doi.org/10.1109/ISBI.2017.7950532

Ronneberger, O., Fischer, P., & Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-24574-4_28

Ronneberger, O., Fischer, P., & Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-24574-4_28

Rouhi, R., Jafari, M., Kasaei, S., & Keshavarzian, P. (2015). Benign and malignant breast tumors classification based on region growing and CNN segmentation. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2014.09.020

Samala, R. K., Chan, H.-P., Hadjiiski, L., Helvie, M. A., Richter, C. D., & Cha, K. H. (2019). Breast Cancer Diagnosis in Digital Breast Tomosynthesis: Effects of Training Sample Size on Multi-Stage Transfer Learning Using Deep Neural Nets. *IEEE Transactions on Medical Imaging*, *38*(3), 686–696. https://doi.org/10.1109/TMI.2018.2870343

Sánchez, C. I., Niemeijer, M., Išgum, I., Dumitrescu, A., Suttorp-Schulten, M. S. A., Abràmoff, M. D., & van Ginneken, B. (2012). Contextual computer-aided detection: Improving bright lesion detection in retinal images and coronary calcification identification in CT scans. *Medical Image Analysis*. https://doi.org/10.1016/j.media.2011.05.004

Sandfort, V., Yan, K., Pickhardt, P. J., & Summers, R. M. (2019). Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific Reports*, *9*(1), 16884. https://doi.org/10.1038/s41598-019-52737-x

Sener, O., & Savarese, S. (2017). *Active Learning for Convolutional Neural Networks: A Core-Set Approach*.

Shelhamer, E., Long, J., & Darrell, T. (2017a). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2016.2572683

Shelhamer, E., Long, J., & Darrell, T. (2017b). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2016.2572683

Shin, S. Y., Lee, S., Yun, I. D., Kim, S. M., & Lee, K. M. (2019). Joint Weakly and Semi-Supervised Deep Learning for Localization and Classification of Masses in Breast Ultrasound Images. *IEEE Transactions on Medical Imaging*, *38*(3), 762–774. https://doi.org/10.1109/TMI.2018.2872031

Sinha, S., Zhang, H., Goyal, A., Bengio, Y., Larochelle, H., & Odena, A. (2019). *Small-GAN: Speeding Up GAN Training Using Core-sets*.

Song, Y., Zhu, Y., & Du, X. (2019). Dynamic residual dense network for image denoising. *Sensors (Switzerland)*. https://doi.org/10.3390/s19173809

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, *35*(5), 1299–1312. https://doi.org/10.1109/TMI.2016.2535302

Tan, J., Gao, Y., Liang, Z., Cao, W., Pomeroy, M. J., Huo, Y., Li, L., Barish, M. A., Abbasi, A. F., & Pickhardt, P. J. (2020). 3D-GLCM CNN: A 3-Dimensional Gray-Level Co-Occurrence Matrix-Based CNN Model for Polyp Classification via CT Colonography. *IEEE Transactions on Medical Imaging*, *39*(6), 2013–2024. https://doi.org/10.1109/TMI.2019.2963177

Tangri, N., Hougen, I., Alam, A., Perrone, R., McFarlane, P., & Pei, Y. (2017). Total kidney volume as a biomarker of disease progression in autosomal dominant polycystic kidney disease. *Canadian Journal of Kidney Health and Disease*. https://doi.org/10.1177/2054358117693355

Toolbox, I. P. (2004). Image Processing Toolbox. *Image Processing*.

Tuytelaars, T., & Mikolajczyk, K. (2007). Local Invariant Feature Detectors: A Survey. *Foundations and Trends® in Computer Graphics and Vision*. https://doi.org/10.1561/0600000017

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). *Instance Normalization: The Missing Ingredient for Fast Stylization*.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., & Gelly, S. (2018). Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717.

Wahab, N., Khan, A., & Lee, Y. S. (2019). Transfer learning based deep CNN for segmentation and detection of mitoses in breast cancer histopathological images. *Microscopy*. https://doi.org/10.1093/jmicro/dfz002

Wang, G., Lopez-Molina, C., & de Baets, B. (2017). Blob Reconstruction Using Unilateral Second Order Gaussian Kernels with Application to High-ISO Long-

Exposure Image Denoising. *2017 IEEE International Conference on Computer Vision (ICCV)*, 4827–4835. https://doi.org/10.1109/ICCV.2017.516

Witkin, A. P. (1983). Scale-space Filtering. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. https://doi.org/10.1109/ICASSP.1984.1172729

Wolf, G. W. (2011). Facility location: concepts, models, algorithms and case studies. Series: Contributions to Management Science. *International Journal of Geographical Information Science*, *25*(2), 331–333. https://doi.org/10.1080/13658816.2010.528422

Wu, Q., Merchant, F. A., & Castleman, K. R. (2007). Microscope Image Processing. In *Microscope Image Processing*. https://doi.org/10.1016/B978-0-12-372578-3.X0001-3

Wu, T., Gaw, N., Xu, Y., Li, J., Wang, L., Fu, Y., Silva, A., Zwart, C., Borad, M., DeLeon, T., & Patel, B. (2017). Quantitative Imaging System for Cancer Diagnosis and Treatment Planning: An Interdisciplinary Approach. In *The Operations Research Revolution*. https://doi.org/10.1287/educ.2017.0173

Xing, F., Cornish, T. C., Bennett, T., Ghosh, D., & Yang, L. (2019). Pixel-to-Pixel Learning with Weak Supervision for Single-Stage Nucleus Recognition in Ki67 Images. *IEEE Transactions on Biomedical Engineering*. https://doi.org/10.1109/TBME.2019.2900378

Xing, F., Xie, Y., & Yang, L. (2016). An automatic learning-based framework for robust nucleus segmentation. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2015.2481436

Xu, Y., Gao, F., Wu, T., Bennett, K. M., Charlton, J. R., & Sarkar, S. (2019). U-net with optimal thresholding for small blob detection in medical images. *IEEE International Conference on Automation Science and Engineering*. https://doi.org/10.1109/COASE.2019.8843234

Xu, Y., Wu, T., Charlton, J. R., Gao, F., & Bennett, K. M. (2021). Small Blob Detector Using Bi-Threshold Constrained Adaptive Scales. *IEEE Transactions on Biomedical Engineering*, *68*(9), 2654–2665. https://doi.org/10.1109/TBME.2020.3046252

Xu, Y., Wu, T., Gao, F., Charlton, J. R., & Bennett, K. M. (2020). Improved small blob detection in 3D images using jointly constrained deep learning and Hessian analysis. *Scientific Reports*. https://doi.org/10.1038/s41598-019-57223-y

Xue, Y., Bigras, G., Hugh, J., & Ray, N. (2019). Training Convolutional Neural Networks and Compressed Sensing End-to-End for Microscopy Cell Detection. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2019.2907093

Yang, S., Kim, E. Y., & Ye, J. C. (2021). Continuous Conversion of CT Kernel Using Switchable CycleGAN With AdaIN. *IEEE Transactions on Medical Imaging*, *40*(11), 3015–3029. https://doi.org/10.1109/TMI.2021.3077615

Yap, M. H., Pons, G., Martí, J., Ganau, S., Sentís, M., Zwiggelaar, R., Davison, A. K., & Martí, R. (2018). Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Networks. *IEEE Journal of Biomedical and Health Informatics*, *22*(4), 1218–1226. https://doi.org/10.1109/JBHI.2017.2731873

Zhang, M., Wu, T., Beeman, S. C., Cullen-McEwen, L., Bertram, J. F., Charlton, J. R., Baldelomar, E., & Bennett, K. M. (2016). Efficient Small Blob Detection Based on Local Convexity, Intensity and Shape Information. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2015.2509463

Zhang, M., Wu, T., & Bennett, K. M. (2015). Small Blob Identification in Medical Images Using Regional Features From Optimum Scale. *IEEE Transactions on Biomedical Engineering*. https://doi.org/10.1109/TBME.2014.2360154

Zhang, Z., Yang, L., & Zheng, Y. (2018). Translating and Segmenting Multimodal Medical Volumes with Cycle- and Shape-Consistency Generative Adversarial Network. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9242–9251. https://doi.org/10.1109/CVPR.2018.00963

Zhou, L., Schaefferkoetter, J. D., Tham, I. W. K., Huang, G., & Yan, J. (2020). Supervised learning with cyclegan for low-dose FDG PET image denoising. *Medical Image Analysis*, *65*, 101770. https://doi.org/10.1016/j.media.2020.101770

Zhu, H., Fang, Q., Huang, Y., & Xu, K. (2020). Semi-supervised method for image texture classification of pituitary tumors via CycleGAN and optimized feature extraction. *BMC Medical Informatics and Decision Making*, *20*(1), 215. https://doi.org/10.1186/s12911-020-01230-x

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2242–2251. https://doi.org/10.1109/ICCV.2017.244

Zreik, M., van Hamersvelt, R. W., Wolterink, J. M., Leiner, T., Viergever, M. A., & Isgum, I. (2019). A Recurrent CNN for Automatic Detection and Classification of Coronary Artery Plaque and Stenosis in Coronary CT Angiography. *IEEE Transactions on Medical Imaging*, *38*(7), 1588–1598. https://doi.org/10.1109/TMI.2018.2883807

APPENDIX A

PROOF OF MONOTONICITY OF UNET PROBABILITY MAP

Proposition 1. For any blob, let the normalized intensity distribution of the blob be $I_b(x,y,z) \in [0,1]^{I_1 \times I_2 \times I_3}$, and the centroid of the blob be (μ_x, μ_y, μ_z), assuming that the blob (after denoising) follows a rotationally symmetric Gaussian distribution,

$$I_b(x,y,z) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-\mu_x)^2+(y-\mu_y)^2+(z-\mu_z)^2}{2\sigma^2}\right). \quad (6.1)$$

The probability predicted by U-Net increases or decreases monotonically from the centroid to the boundary of the dark or bright blob.

[Proof]

Here we focus on bright blobs. Let the input intensity distributions of a blob with noise be $I_N \in [0,1]^{I_1 \times I_2 \times I_3}$. We have

$$I_N = I_b + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I). \quad (6.2)$$

We define the probability map from U-Net as $U(x,y,z) \in [0,1]^{I_1 \times I_2 \times I_3}$, which indicates the probability of each voxel belonging to any blob. The probability of blob U_b can approximate the intensity distribution of the blob based on equation (3.4):

$$U_b(x,y,z) = \mathcal{F}_b(I_N; \Theta) = \mathcal{F}_b(I_b + \varepsilon; \Theta) \approx I_b(x,y,z). \quad (6.3)$$

The probabilities from $U_b(x,y,z)$ thus follow a Gaussian distribution and the probabilities monotonically decrease from the centroid to the boundary of a blob, with $U_b(\mu_x, \mu_y, \mu_z)$ reaching maximum probability.

[End of Proof]

Proposition 2. Given a binarized probability map, a blob can be identified with a radius r. With $B_L(x,y,z)$ and $B_H(x,y,z)$, we obtain r_{δ_L}, r_{δ_H} respectively. $B_L(x,y,z)$

marks a larger blob region extending to the boundaries with low probability and $B_H(x, y, z)$ marks a smaller blob region extending the boundary with high probability, that is $r_{\delta_L} > r_{\delta_H}$.

[Proof]

From equation (6.1) and equation (6.3), we get:

$$U_b(x, y, z) \approx I_b(x, y, z) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-\mu_x)^2+(y-\mu_y)^2+(z-\mu_z)^2}{2\sigma^2}\right). \quad (6.4)$$

Let the radius of a blob be $r(\delta) \in R$. The distance between the thresholding pixel $(x_\delta, y_\delta, z_\delta)$ and the centroid of blob can be approximated by the radius of the blob:

$$r(\delta) \approx \sqrt{(x_\delta - \mu_x)^2 + (y_\delta - \mu_y)^2 + (z_\delta - \mu_z)^2}. \quad (6.5)$$

Given high probability threshold δ_H and low probability threshold δ_L,

$$U_b(x_{\delta_H}, y_{\delta_H}, z_{\delta_H}) = \delta_H, \quad (6.6)$$

and

$$U_b(x_{\delta_L}, y_{\delta_L}, z_{\delta_L}) = \delta_{L.} \quad (6.6)$$

From ***Proposition 1***, the blob centroid has the maximum probability and the probability monotonically decreases from the centroid to the boundary:

$$U_b(x_{\delta_L}, y_{\delta_L}, z_{\delta_L}) < U_b(x_{\delta_H}, y_{\delta_H}, z_{\delta_H}) < U_b(\mu_x, \mu_y, \mu_z), \quad (6.7)$$

and

$$r(\delta_L) > r(\delta_H) > r\left(U_b(\mu_x, \mu_y, \mu_z)\right) = 0. \quad (6.8)$$

[End of Proof]

APPENDIX B

PROOF OF 3D ELLIPTICAL GAUSSIAN FUNCTION

Proposition 1. Let $F(x, y, z)$ be the 3D elliptical Gaussian function with the following general form:

$$F(x, y, z) = \mathcal{A} \cdot e^{-(a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 + d(x-x_0)(y-y_0) + e(y-y_0)(z-z_0) + f(x-x_0)(z-z_0))},$$

where $\mathcal{A}$ is a normalization factor, and x_0, y_0, z_0 are the coordinates of the center of $F(x, y, z)$.

Assume the coefficients a, b, c, d, e, f explicitly control the shape and orientation of $F(x, y, z)$ by means of θ, φ, σ_x, σ_y, σ_z. i.e., we want to show that:

$$a = \frac{sin^2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin^2\theta sin^2\varphi}{\sigma_y^2} + \frac{cos^2\theta}{\sigma_z^2}$$

$$b = \frac{cos^2\theta cos^2\varphi}{\sigma_x^2} + \frac{cos^2\theta sin^2\varphi}{\sigma_y^2} + \frac{sin^2\theta}{\sigma_z^2}$$

$$c = \frac{sin^2\varphi}{\sigma_x^2} + \frac{cos^2\varphi}{\sigma_y^2}$$

$$d = \frac{sin2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin2\theta sin^2\varphi}{\sigma_y^2} - \frac{sin2\theta}{\sigma_z^2}$$

$$e = -\frac{cos\theta sin2\varphi}{\sigma_x^2} + \frac{cos\theta sin2\varphi}{\sigma_y^2}$$

$$f = -\frac{sin\theta sin2\varphi}{\sigma_x^2} + \frac{sin\theta sin2\varphi}{\sigma_y^2}$$

[Proof]

Assume the probability density for two normally distributed variables is given by:

$$F(\boldsymbol{x}) = \mathcal{A} \cdot e^{-(\boldsymbol{u}-\boldsymbol{a})^T B (\boldsymbol{u}-\boldsymbol{a})}, \tag{6.9}$$

where $\mathcal{A}$ is a normalization factor, $\boldsymbol{u} = (u_1, u_2, u_3)^{\boldsymbol{T}}$ and $\boldsymbol{a} = (a_1, a_2, a_3)^{\boldsymbol{T}}$. B is the inverse of the covariance matrix

$$C = \begin{pmatrix} \sigma_1^2 & cov(u_1, u_2) & cov(u_1, u_3) \\ cov(u_1, u_2) & \sigma_2^2 & cov(u_2, u_3) \\ cov(u_1, u_3) & cov(u_2, u_3) & \sigma_3^2 \end{pmatrix}, \tag{6.10}$$

where σ_1, σ_2, σ_3 are the standard deviation of u_1, u_2, u_3. So B is given by

$$B = C^{-1} = \frac{1}{D}\begin{pmatrix} \sigma_2^2\sigma_3^2 - cov(u_2, u_3)^2 & cov(u_1, u_3)cov(u_2, u_3) - \sigma_3^2 cov(u_1, u_2) & cov(u_1, u_2)cov(u_2, u_3) - \sigma_2^2 cov(u_1, u_3) \\ cov(u_1, u_3)cov(u_2, u_3) - \sigma_3^2 cov(u_1, u_2) & \sigma_1^2\sigma_3^2 - cov(u_1, u_3)^2 & cov(u_1, u_2)cov(u_1, u_3) - \sigma_1^2 cov(u_2, u_3) \\ cov(u_1, u_2)cov(u_2, u_3) - \sigma_2^2 cov(u_1, u_3) & cov(u_1, u_2)cov(u_1, u_3) - \sigma_1^2 cov(u_2, u_3) & \sigma_1^2\sigma_2^2 - cov(u_1, u_2)^2 \end{pmatrix},$$

(6.11)

where $D = \sigma_1^2\sigma_2^2\sigma_3^2 - \sigma_1^2 cov(u_2, u_3)^2 - \sigma_2^2 cov(u_1, u_3)^2 - \sigma_3^2 cov(u_1, u_2)^2 + 2cov(u_1, u_2)cov(u_2, u_3)cov(u_1, u_3)$

With the correlation coefficient

$$\rho_{12} = \frac{cov(u_1, u_2)}{\sigma_1\sigma_2}$$

$$\rho_{23} = \frac{cov(u_2, u_3)}{\sigma_2\sigma_3}$$

$$\rho_{13} = \frac{cov(u_1, u_3)}{\sigma_1\sigma_3},$$

(6.12)

B can be rewritten as

$$B = \frac{1}{E}\begin{pmatrix} \sigma_1^{-2}(1 - \rho_{23}^2) & \sigma_1^{-1}\sigma_2^{-1}(\rho_{13}\rho_{23} - \rho_{12}) & \sigma_1^{-1}\sigma_3^{-1}(\rho_{12}\rho_{23} - \rho_{13}) \\ \sigma_1^{-1}\sigma_2^{-1}(\rho_{13}\rho_{23} - \rho_{12}) & \sigma_2^{-2}(1 - \rho_{13}^2) & \sigma_2^{-1}\sigma_3^{-1}(\rho_{12}\rho_{13} - \rho_{23}) \\ \sigma_1^{-1}\sigma_3^{-1}(\rho_{12}\rho_{23} - \rho_{13}) & \sigma_2^{-1}\sigma_3^{-1}(\rho_{12}\rho_{13} - \rho_{23}) & \sigma_3^{-2}(1 - \rho_{12}^2) \end{pmatrix}, \tag{6.13}$$

where $E = 1 - \rho_{12}^2 - \rho_{13}^2 - \rho_{23}^2 + 2\rho_{12}\rho_{13}\rho_{23}$, so

$$(\boldsymbol{u}-\boldsymbol{v})^{\boldsymbol{T}}\boldsymbol{B}(\boldsymbol{u}-\boldsymbol{v}) = \frac{1}{E}[(u_1-a_1)^2\sigma_1^{-2}(1-\rho_{23}^2) + (u_1-a_1)(u_2-$$

$$a_2)\sigma_1^{-1}\sigma_2^{-1}(\rho_{13}\rho_{23}-\rho_{12}) + (u_1-a_1)(u_3-a_3)\sigma_1^{-1}\sigma_3^{-1}(\rho_{12}\rho_{23}-\rho_{13}) +$$

$$(u_1-a_1)(u_2-a_2)\sigma_1^{-1}\sigma_2^{-1}(\rho_{13}\rho_{23}-\rho_{12}) + (u_2-a_2)^2\sigma_2^{-2}(1-\rho_{13}^2) +$$

$$(u_2-a_2)(u_3-a_3)\sigma_2^{-1}\sigma_3^{-1}(\rho_{12}\rho_{13}-\rho_{23}) + (u_1-a_1)(u_3-a_3)\sigma_1^{-1}\sigma_3^{-1}(\rho_{12}\rho_{23}-$$

$$\rho_{13}) + (u_2-a_2)(u_3-a_3)\sigma_2^{-1}\sigma_3^{-1}(\rho_{12}\rho_{13}-\rho_{23}) + (u_3-a_3)^2\sigma_3^{-2}(1-\rho_{12}^2)\,] = C, \tag{6.14}$$

where $E = 1-\rho_{12}^2-\rho_{13}^2-\rho_{23}^2+2\rho_{12}\rho_{13}\rho_{23}$.

Let $u_1' = u_1-a_1$, $u_2' = u_2-a_2$, $u_3' = u_3-a_3$, let C be a constant C = 1, then equation (6.14) can be rewritten as：

$$(\boldsymbol{u}-\boldsymbol{v})^{\boldsymbol{T}}\boldsymbol{B}(\boldsymbol{u}-\boldsymbol{v}) = \frac{1}{E}\left[\frac{u_1'^2(1-\rho_{23}^2)}{\sigma_1^2} + \frac{2u_1'u_2'(\rho_{13}\rho_{23}-\rho_{12})}{\sigma_1\sigma_2} + \frac{u_2'^2(1-\rho_{13}^2)}{\sigma_2^2} + \frac{2u_1'u_3'(\rho_{12}\rho_{23}-\rho_{13})}{\sigma_1\sigma_3} +\right.$$

$$\left.\frac{u_3'^2(1-\rho_{12}^2)}{\sigma_3^2} + \frac{2u_2'u_3'(\rho_{12}\rho_{13}-\rho_{23})}{\sigma_2\sigma_3}\right] = 1, \tag{6.15}$$

where $E = 1-\rho_{12}^2-\rho_{13}^2-\rho_{23}^2+2\rho_{12}\rho_{13}\rho_{23}$.

In this problem, since we consider u_1, u_2, u_3 as three orthogonal directions in the 3D coordinate systems, to simply the model, we can consider $\rho_{12} = \rho_{23} = \rho_{13} = 0$, then

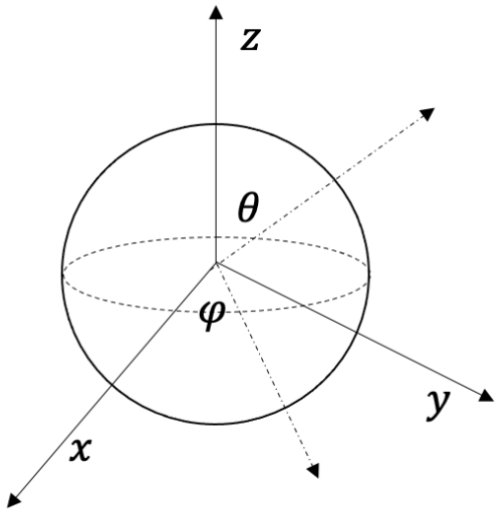
$$(\boldsymbol{u}-\boldsymbol{v})^{\boldsymbol{T}}\boldsymbol{B}(\boldsymbol{u}-\boldsymbol{v}) = \frac{u_1'^2}{\sigma_1^2} + \frac{u_2'^2}{\sigma_2^2} + \frac{u_3'^2}{\sigma_3^2} = 1, \tag{6.16}$$

To consider the orientations of blobs, we need to transfer equation (6.16) into Spherical coordinate systems, so units can be transferred as:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\varphi & \sin\theta\sin\varphi & \cos\theta \\ \cos\theta\cos\varphi & \cos\theta\sin\varphi & -\sin\theta \\ -\sin\varphi & \cos\varphi & 0 \end{pmatrix}\begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix}, \tag{6.17}$$

Then

$$\begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}, \tag{6.18}$$



Then

$$(\boldsymbol{u}-\boldsymbol{v})^{\boldsymbol{T}}\boldsymbol{B}(\boldsymbol{u}-\boldsymbol{v}) = \frac{(\sin\theta\cos\varphi x'+\cos\theta\cos\varphi y'-\sin\varphi z')^2}{\sigma_1^2} +$$

$$\frac{(\sin\theta\sin\varphi x'+\cos\theta\sin\varphi y'+\cos\varphi z')^2}{\sigma_2^2} + \frac{(\cos\theta x'-\sin\theta y')^2}{\sigma_3^2} = 1, \tag{6.19}$$

Since $\sigma_1^2 = \sigma_x^2, \sigma_2^2 = \sigma_y^2, \sigma_3^2 = \sigma_z^2$, let $x' = x - x_0, y' = y - y_0, z' = z - z_0$

$$(\boldsymbol{u}-\boldsymbol{v})^{\boldsymbol{T}}\boldsymbol{B}(\boldsymbol{u}-\boldsymbol{v}) = (x-x_0)^2\left(\frac{sin^2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin^2\theta sin^2\varphi}{\sigma_y^2} + \frac{cos^2\theta}{\sigma_z^2}\right) + (y -$$

$$y_0)^2\left(\frac{cos^2\theta cos^2\varphi}{\sigma_x^2} + \frac{cos^2\theta sin^2\varphi}{\sigma_y^2} + \frac{sin^2\theta}{\sigma_z^2}\right) + (z-z_0)^2\left(\frac{sin^2\varphi}{\sigma_x^2} + \frac{cos^2\varphi}{\sigma_y^2}\right) + (x-x_0)(y -$$

$$y_0)\left(\frac{sin2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin2\theta sin^2\varphi}{\sigma_y^2} - \frac{sin2\theta}{\sigma_z^2}\right) + (y-y_0)(z-z_0)\left(-\frac{cos\theta sin2\varphi}{\sigma_x^2} + \frac{cos\theta sin2\varphi}{\sigma_y^2}\right) +$$

$$(x-x_0)(z-z_0)\left(-\frac{sin\theta sin2\varphi}{\sigma_x^2} + \frac{sin\theta sin2\varphi}{\sigma_y^2}\right). \tag{6.20}$$

Let $F(x,y,z)$ be the 3D elliptical Gaussian function with the following general form:

$$F(x,y,z) = \mathcal{A} \cdot e^{-(a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 + d(x-x_0)(y-y_0) + e(y-y_0)(z-z_0) + f(x-x_0)(z-z_0))},$$

where $\mathcal{A}$ is a normalization factor,

then

$$a = \frac{sin^2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin^2\theta sin^2\varphi}{\sigma_y^2} + \frac{cos^2\theta}{\sigma_z^2}$$

$$b = \frac{cos^2\theta cos^2\varphi}{\sigma_x^2} + \frac{cos^2\theta sin^2\varphi}{\sigma_y^2} + \frac{sin^2\theta}{\sigma_z^2}$$

$$c = \frac{sin^2\varphi}{\sigma_x^2} + \frac{cos^2\varphi}{\sigma_y^2}$$

$$d = \frac{sin2\theta cos^2\varphi}{\sigma_x^2} + \frac{sin2\theta sin^2\varphi}{\sigma_y^2} - \frac{sin2\theta}{\sigma_z^2}$$

$$e = -\frac{cos\theta sin2\varphi}{\sigma_x^2} + \frac{cos\theta sin2\varphi}{\sigma_y^2}$$

$$f = -\frac{sin\theta sin2\varphi}{\sigma_x^2} + \frac{sin\theta sin2\varphi}{\sigma_y^2},$$

(6.21)

[End of Proof]