Fluorescence Microscopy Image Analysis of Retinal Neurons Using Deep Learning



Mihael Cudic

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2023

Dedication

This thesis is dedicated to my brother, whose humor lightens even the darkest moments.

Acknowledgements

I would first like to express gratitude to my supervisors, Professor Alison Noble and Dr. Jeffrey Diamond, for their invaluable insights and guidance throughout my studies. Their openness to cross-disciplinary learning made this thesis come to fruition. I am also appreciative to the NIH Oxford-Cambridge Scholars Program and the International Biomedical Research Alliance for their financial support, whose funding allowed me to focus entirely on research. In addition, I would like to thank Hua Tian, Will Grimes, Amurta Nath, and Ulisse Bocchero for their patience and willingness to teach me wet lab techniques. In addition to research assistance, the relationships formed throughout my time at the NIH and the University of Oxford have allowed me to grow intellectually and personally. I am grateful for the stimulating discussions and emotional support from my friends Steven Castellano, Mario Shammas, and Adit Sabnis at the NIH. I am also greatly appreciative of my friends at the University of Oxford - Robin Brons, Emile Okada, Lois Ogunlana, and Molly Schoenfeld - for their discussions, comfort, and company. Lastly and most importantly, I am deeply grateful to my parents and brother. Despite my struggles throughout my studies, they have provided unwavering support.

Statement of Originality

The thesis submitted is entirely my own work, except where otherwise indicated.

Abstract

An essential goal of neuroscience is to understand the brain by simultaneously identifying, measuring, and analyzing activity from individual cells within a neural population in live brain tissue. Analyzing fluorescence microscopy (FM) images in real-time with computational algorithms is essential for achieving this goal. Deep learning techniques have shown promise in this area, but face domain-specific challenges due to limited training data, significant amounts of voxel noise in FM images, and thin structures present in large 3D images. In this thesis, I address these issues by introducing a novel deep learning pipeline to analyze static FM images of neurons with minimal data requirements and demonstrate the pipeline's ability to segment neurons from low signal-to-noise ratio FM images with few training samples. The first step of this pipeline employs a Generative Adversarial Network (GAN) equipped to learn imaging properties from a small set of static FM images acquired for a given neuroscientific experiment. Operating like an actual microscope, our fully-trained GAN can then generate realistic static FM images from volumetric reconstructions of neurons with added control over the intensity and noise of the generated images. For the second step in our pipeline, a novel segmentation network is trained on GAN-generated images with reconstructed neurons serving as "gold standard" ground truths. While training on a large dataset of FM images is optimal, a 15% improvement in neuron segmentation accuracy from noisy FM images is shown when architectures are fine-tuned only on a small subsample of real image data. To evaluate the overall feasibility of our pipeline and the utility of generated images, 2 novel synthetic and 3 newly acquired FM image datasets are introduced along with a new evaluation protocol for FM image "realness" that incorporates content, texture, and expert opinion metrics. While this pipeline's primary application is to segment neurons from highly noisy FM images, its utility can be extended to automate other FM tasks such as synapse identification, neuron classification, or super-resolution.

Contents

Abbreviations **viii**

1 Introduction **1**

2 Background **6**

2.1 Introduction 7
2.2 The Retina 7
2.2.1 Retinal Tissue Diversity 8
2.2.2 Retinal Organization 10
2.2.2.1 Photoreceptors 10
2.2.2.2 Horizontal Cells 11
2.2.2.3 Bipolar Cells 11
2.2.2.4 Amacrine Cells 12
2.2.2.5 Ganglion Cells 12
2.3 Fluorescence Microscopy for Retinal Neuroscience 13
2.3.1 Microscopy 13
2.3.1.1 Fluorescence Microscopy 14
2.3.2 Fluorescence Microscopy Imaging 15
2.3.3 Neural Labelling 16
2.4 Supervised Learning 17
2.4.1 Stochastic Gradient Descent 18
2.4.2 Overfitting 19
2.5 Unsupervised Learning 21
2.5.1 Autoencoders 21
2.5.1.1 Variational Autoencoders for Data Generation 23
2.5.1.2 Challenges with Autoencoders 23
2.5.2 Generative Adversarial Networks 24

2.5.2.1 Conditional Generative Adversarial Networks for Corresponding Data . . . 27
2.5.2.2 Cycle Generative Adversarial Networks . . . 28
2.6 Deep Learning for Images . . . 29
2.6.1 Classification . . . 31
2.6.1.1 LeNet-5 . . . 31
2.6.1.2 VGGNet . . . 32
2.6.1.3 Inception-Net . . . 32
2.6.2 Segmentation . . . 33
2.6.2.1 Fully Convolutional Network . . . 33
2.6.2.2 U-Net . . . 34
2.6.3 Style Transfer . . . 35
2.6.4 Image Generation . . . 37
2.7 Deep Learning for Biomedical Images . . . 38
2.8 Classical FM Image Analysis . . . 40
2.8.1 Vessel Enhancement . . . 40
2.8.2 Automated Based Tracers . . . 41
2.9 Conclusion . . . 42

3 Fluorescence Microscopy Image Datasets of Neurons 44
3.1 Introduction . . . 44
3.1.1 Challenges of Deep Learning for Experimental Neuroscience Images . . . 45
3.1.2 Synthetic Datasets of Neurons . . . 48
3.1.3 Publicly Available FM Datasets of Neurons . . . 49
3.1.4 Publicly Available Volumetric Reconstructions of Electron Microscopy Blocks . . . 50
3.2 Novel FM Datasets of Neurons . . . 51
3.2.1 2D Synthetic Images of Bipolar Cells (2DSyn_BC) . . . 52
3.2.2 3D Synthetic Images of Ganglion Cells (3DSyn_GC) . . . 52
3.2.3 3D FM Images of Amacrine Cells (3DFM_AC) . . . 54
3.2.4 3D FM Images of Bipolar Population (3DFM_BCPop) . . . 57
3.2.5 3D FM Images of Overlapping Amacrine Cells (3DFM_overAC) . . . 58
3.3 Discussion . . . 60

4 Unpaired Mesh-to-Image Translation for FM Images of Neurons 63
4.1 Introduction . . . 64
4.2 Related Works . . . 68
4.3 Methodology . . . 70
4.3.1 Voxelized Mesh . . . 70
4.3.2 Slice-to-Slice Translation . . . 70
4.3.3 Generator Architecture . . . 73
4.3.3.1 Topology . . . 73
4.3.3.2 Normalization . . . 73
4.3.3.3 Noise and Acquisition Layer . . . 74
4.3.4 Discriminator Architecture . . . 76
4.3.4.1 Discriminator with Spatial Standard Deviation Layer 76
4.3.4.2 Gramian-based Discriminator . . . 77
4.3.5 Objective . . . 77
4.3.6 Random Sampling of Patches . . . 78
4.4 Implementation . . . 79
4.4.1 Data . . . 79
4.4.1.1 2DSyn_BC . . . 80
4.4.1.2 3DSyn_GC . . . 80
4.4.1.3 3DFM_AC . . . 80
4.4.1.4 3D_BCPop . . . 80
4.4.2 Architecture . . . 81
4.5 Results . . . 82
4.6 Discussion . . . 85

5 Evaluating Realism in FM Images 88
5.1 Introduction . . . 88
5.2 Related Works . . . 90
5.2.1 Classification . . . 90
5.2.2 Sample Distribution Distance . . . 91
5.2.3 Feature Distribution Distance . . . 93
5.2.4 Data Utility . . . 94
5.3 Method . . . 94
5.3.1 Data . . . 94
5.3.2 Evaluation . . . 95
5.3.2.1 Content . . . 95

5.3.2.2 Texture . . . 96
5.3.2.3 Noise . . . 97
5.3.2.4 Utility . . . 97
5.3.2.5 Expert Opinion . . . 99
5.3.3 Comparisons . . . 99
5.4 Results . . . 101
5.5 Conclusion . . . 104

6 Multi-scale Deep Learning for Neural Tracing 111
6.1 Introduction . . . 111
6.2 Related Works . . . 113
6.2.1 Class Imbalance . . . 113
6.2.2 Image Denoisers . . . 115
6.2.3 Segmenting Thin Structures . . . 117
6.2.4 The Mammalian Retina . . . 118
6.3 Methodology . . . 120
6.3.1 Pre-training on GAN-Generated Images . . . 120
6.3.2 Local Residual Denoiser . . . 120
6.3.3 Global Segmentation with Enlarged Ground Truths . . . 121
6.3.4 Multi-Scale Fusion Loss . . . 121
6.4 Implementation . . . 122
6.5 Results . . . 123
6.6 Conclusion . . . 124

7 Conclusion 127

8 Future Works 130

A Expert Opinion 132

Bibliography 134

List of Figures

2.1 A simplified schematic of the mouse retina circuitry 9
2.2 Static fluorescence microscopy image of neurons labeled with three distinct fluorescent proteins 16
2.3 A demonstration of underfitting and overfitting 19
2.4 Schematics of an autoencoder and GAN 21
2.5 An overview of data generation for paired and unpaired datasets . . . 26
2.6 Applications of deep learning for images 29
2.7 A schematic of the U-Net architecture used for image segmentation . 34
2.8 A overview of style transfer 35

3.1 Comparisons between a natural scene and FM images 46
3.2 Sampled images from the 2DSyn_BC and the 3DSyn_GC dataset . . . 53
3.3 Maximum z-projections of all 22 Starburst Amacrine Cells imaged in our new 3DFM_AC dataset. 55
3.4 Patches sampled from the 3DFM_AC dataset with varying imaging conditions 56
3.5 Maximum z-projections from the 3DFM_BCPop dataset. 58
3.6 Samples from our $3DFM_overAC$ dataset and how graound truths were obtained 59

4.1 An overview of our proposed pipeline and a visual comparison of generated synthetic fluorescence microscopy (FM) patches. 67
4.2 An overview of our mesh-to-image translation architecture 71
4.3 Overview of our mesh and FM patch sampling protocol for three datasets 78
4.4 GAN-generated patches on 3DFM_AC dataset with varying imaging conditions 82
4.5 Demonstration of our generator's ability to generate a wide variety of images. 83

4.6 Demonstration of our algorithm's ability to produce images with novel noise characteristics 84
4.7 4 generated synthetic slices with each slice displaying differing morphological variations (rotation, mirror, sheer) of the same neural mesh. 85
4.8 Demonstration of generator generalizability 86

5.1 Real and GAN-generated patches from the 3DFM_AC dataset 97
5.2 Segmentation predictions on never-before-seen FM slices at varying laser powers using generated training datasets from SPADE and our generator. 101
5.3 Average intersection of union scores across varying imaging conditions using generated training datasets from SPADE and our generator . . 102
5.4 Qualitative comparisons between real, SPADE-generated, and Our-generated patches using unpaired and paired training regiments from the 3DFM_BCPop dataset 103
5.5 Comparison of average generated patches with and without `Spatial_stddev` in the discriminator for both synthetic datasets. 104
5.6 Real and generated 128x128 patches sampled from the 2DSyn_BC dataset. While unpaired images were used during training, all real and generated patches were created based on the same voxelized mesh to make visual comparisons easier. 106
5.7 Real and generated 1x128x128 patches sampled from the 3DSyn_GC dataset. While unpaired images were used during training, all real and generated patches were created based on the same voxelized mesh to make visual comparisons easier. 107
5.8 Real and generated 1x128x128 patches sampled from the 3DFM_AC dataset. 108
5.9 Real and generated 1x128x128 patches sampled from the 3DFM_AC dataset. 109
5.10 Real and generated 1x128x128 patches sampled from the 3DFM_BCPop dataset. 110

6.1 An illustration of the challenges involved in segmenting low-SNR 3D FM images of neurons. 112
6.2 An overview of the FM neuron segmentation architecture 119
6.3 Inners workings of our segmentation architecture and segmentation performance on low SNR images. 124

6.4 IoU performance when pre-training on GAN-generated images 125
6.5 Segmentation examples of FM neurons 126

A.1 Email sent out to experts to evaluate the "realism" of GAN-generated images. They were instructed to order 15 sets of 4 6x128x128 GAN-generated images by most similar to least similar to an reference real FM patch. They were also given a brief description of which features they should use to evaluate FM image realism. 132
A.2 Example 4 panels from the expert opinion test. Note that this only shows one slice and experts had the ability to scroll through all 6 slices to better evaluate 3D textures. 133

List of Tables

3.1 Summary of datasets proposed in thesis 45

4.1 Architecture details for our Gramian-based discriminator. 81

5.1 A quantitative ablation study of our proposed unpaired mesh-to-image translation architecture starting from a baseline architecture (SPADE) for three datasets. 98

5.2 A quantitative ablation study on our proposed architecture comparing unpaired and paired training performance on our 3DFM_BCPop dataset. 100

6.1 An ablation study and a quantitative comparison of our proposed architecture for FM neuron segmentation 122

List Of Abbreviations

1-NN 1-Nearest Neighbor Classifier

2D Two-Dimensional

3D Three-Dimensional

AAV Adeno-associated Viruses

AC Amacrine Cells

ADAM Adaptive Moment Estimation

BC Bipolar Cells

BCPop Bipolar Cell Population

BM3D Block Matching and Three-Dimensional filter

CNN Convolutional Neural Network

COOC 3D Co-Occurrence Matrix

D Discriminator

DAE Denoising Autoencoder

EM Electron Microscopy

FC Fully Connected Layer

FID Fréchet Inception Distance

FM Fluorescence Microscopy

FTLoss Focal Tversky Loss

G Generator

H Encoder

I Decoder

IoU Intersection of Union

IS Inception Score

LBP Local Binary Pattern

GAN Generative Adversarial Network

GC Ganglion Cells

GLCM Gray-Level Co-Occurrence Matrix

MIP Maximum Intensity Projection

MRI Magnetic Resonance Imaging

Net Network

NN Neural Network

NMSE Normalized Mean Squared Error

Norm Normalization

PSNR Peak Signal-to-Noise Ratio

ReLU Rectified Linear Unit

SBEM Serial Block-Face Scanning Electron Microscopy

SNR Signal-to-Noise Ratio

SPADE Spatially-Adaptive Normalization

STD Standard deviation

SVM Support Vector Machine

Syn Synthetic

VAE Variational Autoencoder

VGG Visual Geometry Group

XFP Fluorescent Protein

Chapter 1

Introduction

Deep learning models and the human brain, while fundamentally different, have some remarkable similarities in their approach to processing information. These similarities are best seen in visual processing where there is an abundance of neuroscience research and a strong deep learning framework for analysis. For example, edge detectors and color opponency present in the visual cortex have been found to be learned in the first few layers of deep learning models trained on natural scene images [222]. Higher-level visual processing in both biological and fully-trained computational deep learning-based systems are additionally correlated, suggesting that these systems may organize the visual world in similar ways [142]. Most notably, critical periods, first observed by Hubel and Wiesel when studying the visual processing in cats with cataracts [115], were replicated while training deep learning-based systems with various visual deficits [4].

These similarities are in part due to the integration of brain-inspired computational units and architectures into deep learning-based models, as well as the training of these computational models to automate human tasks. For instance, the fundamental building blocks of deep learning systems, known as perceptrons, are directly modeled after spiking neurons as they output non-linear values after integrating inputs [230]. Dense, interconnected networks of these perceptrons then resemble biological neural networks and can compute complex features. By training these artificial neural networks on data collected for the purpose of automating human tasks, such as classification of natural scene images, similar features to those present in biological neural networks have been shown to be learned [222, 142].

As a result, deep learning-based models have been employed as systems models for neuroscience research. In the retina, deep learning-based models have outperformed conventional approaches in modeling retinal responses to natural images [178]. Lindesy et al. additionally used a deep learning-based approach to demonstrate that

the size of the visual cortex directly influences the required complexity of retinal computations, offering a possible reason for why retinas vary across species [161].

Deep learning is, therefore, an undeniably powerful tool for generating new neuroscientific insights; however, its utility for modeling biological systems is limited in several ways. Firstly, biological neural networks represent highly interconnected, dynamic systems functioning in continuous time and space, unlike deep learning models which process digital information [100]. This requires different computational processing units and components for these deep learning models. Secondly, while deep learning models can draw inspiration from neuroscience, we do not yet fully understand biological neural networks and, thus, cannot model them completely from the bottom up. Lastly, even if deep learning architectures could accurately represent the biological network architectures, their objectives still diverge significantly. While deep learning-based models attempt to minimize the prediction error of training data, biological systems have evolved to maximize survival. Consequently, it is challenging for cost functions to account for all ambiguous factors such as energy efficiency, robustness to environmental changes, and heightened sensitivities to improbable events such as imminent danger.

Instead of modeling biological network dynamics, deep learning can be used to expedite neuroscience research by automating large-scale data analysis, a significant bottleneck in the field. Experimental neuroscience generates large volumes of data that only experts can annotate. In certain scenarios, it can even be challenging for experts to accurately annotate data [293]. Although attempts to automate analysis have been made, solutions are primarily confined to a few tasks and underperform in the presence of noise [179]. Deep learning, which has been shown to be robust for a variety of analysis applications, offers a promising alternative in these circumstances.

However, deep learning-based applications to experimental neuroscience present a unique set of challenges. For one, the role of deep learning in experimental neuroscience is fundamentally unique due to the varying nature and objectives of acquired data. Typical applications of deep learning focus on training models on a large sample of representative data points so that the models can then be implemented in real world scenarios. For example, a large dataset of videos taken while driving on streets might be used to train a model for self-driving cars. In contrast, experimental neuroscience research collects new datasets to foster discoveries. Once the dataset is fully analyzed, then novel questions are asked, and new data is collected. Therefore, it is hard to create generalized deep learning-based solutions due the variability of

experimental neuroscience image data and the objective of the analysis is constantly evolving.

Moreover, conventional deep learning-based solutions are ill-suited for experimental neuroscience data. Let's take the example of fluorescence microscopy (FM) for experimental neuroscience, the imaging modality used throughout this thesis. FM is an invaluable tool in neuroscience as it allows for the imaging of neurons deep within tissue with high levels of spatial resolution. However, captured FM images are typically sparse and acquired under uncontrolled conditions. For instance, it's common for individual FM images to contain artifacts from auto-fluorescent particles in tissue, unwanted morphological noise, or accidental human errors [189, 258]. In addition, these images typically contain a significant amount of voxel noise which corrupts an already weak fluorescent signal from neurons [258]. Though this noise can be reduced by increasing the laser power and averaging more frames during image acquisition, this solution isn't always feasible. In live imaging scenarios, the laser power must be minimized to prevent neuronal death and photobleaching caused by excessive laser exposure. As a final limitation, even in high-quality FM images, the neurons present in the images contain thin, branch-like structures that traverse large 3D volumes [179]. Because these structures are defined by their geometries, they do not contain high-spatial redundancies which conventional deep learning-based models make use of for effective image analysis.

To advance neuroscience research, this thesis, therefore, introduces and evaluates a novel deep learning-based pipeline for automated static FM image analysis. The proposed GAN-based solution involves generating realistic static 3D FM images from previously reconstructed neurons using Generative Adversarial Networks (GANs). The GAN, which operates as a synthetic microscope when fully trained, is then used to create a comprehensive dataset to train subsequent deep learning-based models for FM image analysis. While annotated data improves image analysis results, it is not a requirement for our model. We finally demonstrate our pipeline's ability to outperform state-of-the-art deep learning-based models on FM image generation and neuron segmentation tasks from newly acquired FM image datasets.

More specifically, **Chapter 3** introduces five novel static FM image datasets: 2DSyn_BC, 3DSyn_GC, 3DFM_AC, 3DFM_BCPop, and 3DFM_ovAC. Three of these datasets (2DSyn_BC, 3DSyn_GC, 3DFM_AC) consist of individual retinal neurons with varying types of pixels and structured noise. The 3DFM_BCPop dataset and the 3DFM_ovAC dataset contain a population of neurons and overlapping neurons respectively. These datasets differ from those currently publicly available as they

have been specifically designed to be challenging for conventional deep learning architectures due to the significant amounts of voxel and morphological noise present in images and contain reliable ground truths to facilitate thorough deep learning evaluation. All datasets are either already public or will be made public to encourage the community-wide development of deep learning-based applications for FM analysis.

Next, in Chapter 4, we propose an unpaired mesh-to-image translation methodology for generating volumetric FM images of neurons from corresponding ground truths. This is achieved by first learning unique styles efficiently through trainable Gram Matrix computations in the discriminator from a small set of real FM images. Then, we stylize 3D semantic maps of neurons by successively generating 2D slices that depend on 3D context, effectively populating nanoscopic volumetric reconstructions FM characteristics. Finally, we demonstrate that once fully trained, our generator operates as a synthetic microscope that can acquire realistic FM images of neurons with control over the image content and imaging configurations.

The current limitations of GAN metrics for FM image generation are highlighted in **Chapter 5**. We therefore additionally propose a novel evaluation protocol using a mixture of texture-based metrics, unsupervised segmentation accuracy, and an expert opinion test. We demonstrate the feasibility of our GAN-based model and its superior performance compared to state-of-the-art image translation using this novel evaluation protocol.

Finally, we introduce the challenges of applying conventional deep learning for neuron segmentation in **Chapter 6**. In this chapter, we additionally described how to address these limitations through the introduction of a bio-inspired 3-stage CNN that fuses local and global scales. We show that our model more accurately segments neurons than several competing CNN-based models on a 3D FM dataset of neurons and demonstrate the utility of our GAN-generated images to improve analysis performance.

In summary, our pipeline offers a more accurate analysis of various static FM images. Not only does our pipeline produce more data for machine learning algorithms to train on, but it also makes use the nanoscale resolution of the volumetric reconstructions to increase the analysis power for FM images. This opens the door to new forms of image analysis which greatly expand the pipeline's potential impact on the neuroscience community.

For example, a primary objective in neuroscience is to functionally map biological neural networks as such a map could provide novel insights into neurological diseases

and suggest potential pathways for their cures. Recent advances in molecular biology have made it possible to target multiple neurons simultaneously [246, 46]. For instance, GCaMP, a genetically encoded calcium indicator, can be used to monitor the activity of an entire population of neurons, both at the somas and dendrites [188]. However, organizing this information for meaningful analysis requires some understanding of the underlying neural circuitry and structure. Currently, the only method to gain this understanding is through the laborious process of reconstructing electron microscopy blocks which are limited to a few research laboratories [34].

Although the pipeline was originally designed for segmenting FM neurons, its potential applications extend beyond this primary use. For example, the nanoscopic resolution provided by volumetric reconstructions can be used to identify potential synapse locations, synaptic contacts, or dendritic spines in FM images which would otherwise be obscured in FM due to optical resolution limitations. Additionally, individual neurons within a large tissue region can be more efficiently analyzed to determine their type or characteristics. This could potentially be beneficial when surveying the severity of the degeneration or disease in neural tissue. More so, a higher resolution version of the FM images can be created for illustrative purposes and to increase the signal-to-noise for experimental analysis.

Finally, our pipeline can potentially offer an accessible, real-time alternative by enabling the creation of neuron activity maps within these subpopulations through FM image analysis. Because our pipeline automatically generates ground truths from FM image data, deep learning models can be trained on an abundance of realistic FM data to organize the recorded activity of neuronal populations. This can be done through the segmentation of individual neurons from sub-populations of neurons and pairing recorded responses to their parent neurons. While synaptic connections will be lost from FM analysis alone, this new FM analysis could more readily explore neural activity, thereby accelerating the progress in neuroscience.

Chapter 2

Background

Contents

2.1 Introduction 7
2.2 The Retina 7
2.2.1 Retinal Tissue Diversity 8
2.2.2 Retinal Organization 10
2.3 Fluorescence Microscopy for Retinal Neuroscience . . . 13
2.3.1 Microscopy 13
2.3.2 Fluorescence Microscopy Imaging 15
2.3.3 Neural Labelling 16
2.4 Supervised Learning 17
2.4.1 Stochastic Gradient Descent 18
2.4.2 Overfitting 19
2.5 Unsupervised Learning 21
2.5.1 Autoencoders 21
2.5.2 Generative Adversarial Networks 24
2.6 Deep Learning for Images 29
2.6.1 Classification 31
2.6.2 Segmentation 33
2.6.3 Style Transfer 35
2.6.4 Image Generation 37
2.7 Deep Learning for Biomedical Images 38
2.8 Classical FM Image Analysis 40
2.8.1 Vessel Enhancement 40
2.8.2 Automated Based Tracers 41
2.9 Conclusion 42

2.1 Introduction

This thesis is cross-disciplinary by nature as it applies deep learning to experimental neuroscience. Therefore, the proposed work requires an understanding of not only the current techniques and limitations of deep learning, but also the research objectives of and tools available for experimental neuroscience. Like with all truly cross-disciplinary research, it can be hard to convey the significance and entirety of the work outlined in this thesis to those with knowledge predominately in one domain. What might be elementary in one field, can be complex and arcane in another. However, I decided to embark on such a research project precisely because of the challenges in and importance of cross-disciplinary work. While this thesis does not attempt to comprehensively bridge the fields of deep learning and neuroscience, I will discuss how deep learning methods, like convolutional neural networks (CNNs), can be used as a valuable tool in neuroscience and how visual neuroscience can be used to further inspire architectural advancements for CNNs.

In this section, I will provide a concise overview of the background material required to understand the proposed work. This includes a background of the retina, the biological neural system analyzed in this thesis, along with experimental techniques such as fluorescent labeling and fluorescence microscopy (FM). I will then briefly summarize deep learning and state-of-the-art computational techniques to contextualize the necessity of the architectural advances outlined in this thesis. All readers are encouraged to at least skim every section to ensure that a cohesive message is delivered.

2.2 The Retina

Although the deep learning solutions proposed in this thesis are validated using FM images of retinal neurons from mice, they can be adapted to analyze biological neural networks throughout the brain for any species. All that is required to extend this pipeline to other scenarios are a set of experimental FM images of neurons and volumetric reconstructions of the same type of neuron. The mouse retina, however, serves as a foundation for the development of novel deep learning solutions due to the significant amounts of research already detailing its circuitry, abundance of data publicly available, and ease of acquiring new data. Despite differences between the mouse retina and human retina, the fundamental biological neural organization between species is largely the same [125].

To begin, it is, therefore, useful to provide a brief background of the neural circuitry found in the mouse retina and highlight the functional role of different neuron classes. This overview will help motivate the problem trying to be solved as well as the images acquired in this thesis. The organization of the retina additionally serves as inspiration for architectural designs proposed later in Chapter 6.

2.2.1 Retinal Tissue Diversity

The retina is a thin piece of tissue located at the back of the eye. Its primary function is to encode visual stimuli from the environment and transmit this information to the visual cortex of the brain for further processing. Although the main goal of visual neuroscience is to improve our understanding of human visual information processing, studying the human retina presents significant challenges. Non-invasive techniques such as electroretinography [173] or retinal fundus imaging [302, 25] can be employed to observe the overall activity and health of the human retina, but lack the necessary resolution to examine individual neurons and retinal organization. Consequently, the most direct approach to studying the retina is through retinal dissections which constrict the study of retinal tissue to non-humans.

Retinas from other animals are often used as proxy models for human retinas. However, the retinal circuitry varies significantly across species due to evolutionary pressures [13]. For example, while humans typically see in three colors (blue, green, and red) [172], mantis shrimp are thought to see four colors (ultraviolet, blue, green, and red) in order to better detect camouflaged prey deep in the ocean [273, 306]. Similarly, some species, such as primates [60] or birds of prey [118], have a significantly higher density of neurons at the center of the retina to more precisely process the visual environment. In contrast, other species can have more homogeneous distributions of neurons, like in mice [256, 28, 14]. To ensure that retina research translates to humans, it is therefore imperative that the animal model used for experimentation is chosen with consideration for the differences in the retina relative to the human retina.

By far the most common contemporary animal model in retinal research is the mouse. Although mice primarily utilize their whiskers to navigate the world unlike primates [215, 79], there are several benefits to mice retina research. For one, mice are easy to breed and maintain in animal facilities which provide access to cheap and abundant sources of live retinal tissue. The genetics of mice are additionally well studied, enabling researchers to construct complex experiments as they can more easily genetically modify mice. As the final advantage, mice are reported to share

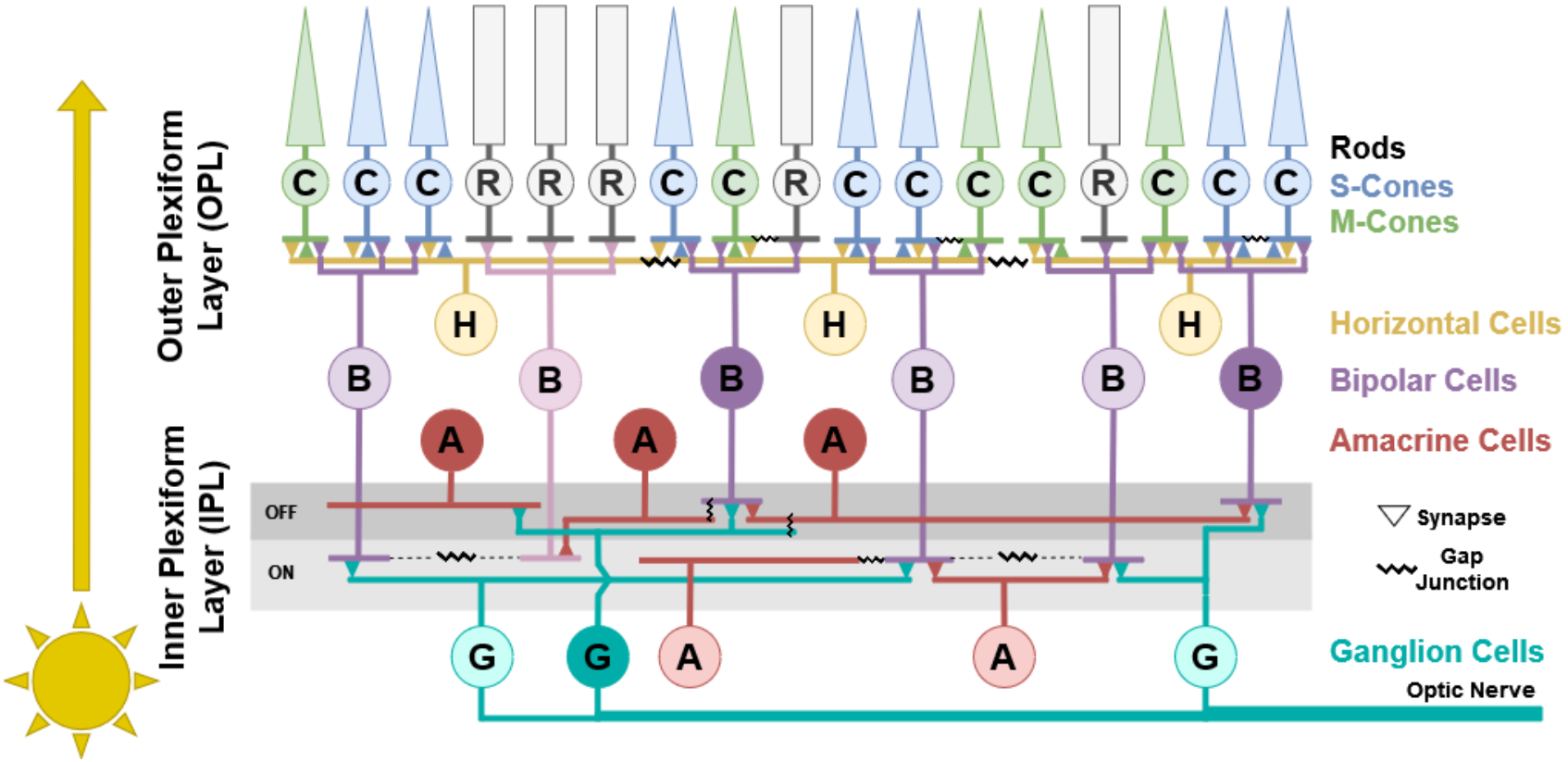


Figure 2.1: A simplified schematic of the mouse retina circuitry. Visual stimuli are detected in the outer retina by three types of photoreceptors: Rods, S-Cones, and M-Cones. Rods primarily function in scotopic conditions, like during nighttime, while cones function during photopic conditions, like during the day. Photoreceptors additionally undergo inhibitory feedback from Horizontal cells which contribute to local contrast enhancement. Then, Bipolar cells relay the photoreceptor signal to either the 'On' or 'Off' layer of the Inner Plexiform Layer (IPL) depending on whether the cells excite at the onset or the cessation of visual stimuli. In the IPL, Amacrine cells integrate Bipolar signals and predominantly provide inhibition to Bipolar, Ganglion, and other Amacrine cells. Finally, Ganglion cells process Bipolar and Amacrine signals to extract various features from the visual world and the output is transmitted to the brain through the optic nerve.

almost all genes with humans, with the average similarity between genes being 85% [69]. This results in many phenomena observed in humans also being observed in mice, such as retinal waves [281].

2.2.2 Retinal Organization

I will now provide a brief overview of the mouse retina circuitry. A simplified schematic of the retina can be found in Fig. 2.1. It should be noted that there are many exceptions to general rules in biology and current research is constantly updating our understanding of retinal circuitry. Thus, the following Sections provide the fundamental principles of the retina sufficient to contextualize this thesis and do not aim to give the most recent account of retinal neuroscience.

2.2.2.1 Photoreceptors

Light is focused onto the retina by the eye's cornea and lense and then passes through most of the retinal tissue before being absorbed by opsin molecules in the outer layer of the retina. Opsins, which are proteins sensitive to a specific range of light wavelengths, reside in sensory neurons known as photoreceptors [81]. When activated, opsins initiate a phototransduction cascade that hyperpolarizes or reduces the activity of, photoreceptors [10, 87]. Therefore, photoreceptors emit the most signal in darkness.

The sensitivity of the phototransduction cascade, and thus the amplitude of the hyperpolarization, is determined by the amount of light in the environment. The more background light present, the more additional light is required to hyperpolarize photoreceptors the same amount. This process, called light adaptation, is an essential part of our ability to see at different times of day [97, 218].

There are two main classes of photoreceptors: rods and cones [87, 55, 31]. They primarily differ in the wavelengths and light levels they are most sensitive to. Rods, known for the cylindrically shaped outer-segment, capture light through the opsin rhodopsin [114, 200]. This protein maximally activates to light at 498 nm and is therefore most sensitive to blue-green light. Furthermore, rods primarily operate in scotopic light levels, such as nighttime, and are saturated at photopic light levels, such as daytime [244].

In contrast, cones have pyramid-like outer segments and are sensitive to photopic, but not scotopic light levels [244]. Individual cone types additionally contain unique opsins sensitive to a specific spectral band. For instance, cones containing L-opsin are most sensitive to 560nm light (red light) while cones containing M-opsin are most

sensitive to 533nm light (green light) [81]. Through operating at photopic light levels and having varying sensitivities, color vision is primarily due to cones.

Finally, rods and cones are also electrically connected through gap junctions, which are intercellular channels that enable ion flow between neurons [11]. The resistance of these gap junctions varies according to light levels and they are believed to provide an alternative pathway for rods to transmit signals to non-rod bipolar cells [248]. However, as this is a brief overview of the retina, a more detailed discussion of gap junctions is beyond the scope of this section.

2.2.2.2 Horizontal Cells

The photoreceptor signal is then transmitted to horizontal cells. These neurons extend laterally across the cell bodies of photoreceptors making synaptic contacts [212, 207]. The size of horizontal cells and the number of their cone contacts vary depending on the species.

The primary function of horizontal cells is to then modify photoreceptor signals by enhancing local contrast [42]. This is achieved through the integration of excitatory photoreceptor signals over a larger area and providing inhibitory feedback to a smaller region of photoreceptors. As a result, photoreceptor response is maximized by the illumination of a small region of the retina and minimized by the illumination of a larger region of the retina. Horizontal cells are additionally strongly coupled through gap junctions which, like photoreceptor gap junctions, are dependant on light levels [177, 295].

2.2.2.3 Bipolar Cells

Bipolar cells additionally contact photoreceptors. Unlike horizontal cells, bipolar cells extend radially throughout the retina and pass information from the outer retina to the inner retina. There are several types of bipolar cells [276, 93, 20, 77]. The exact number and function of these types again depend on the species. Although it is believed that 10-15 bipolar cell types exist [77], they can be organized into three different classes. The first class is On bipolar cells which respond maximally to visual stimuli with a bright center and dimmer surround. The second class of bipolar cells prefers the opposite, with a dimmer center than the surrounding region. Finally, the last class is rod bipolar cells which exclusively make contact with rods and also respond to a bright center spot. While bipolar type 3 and 4 cells are reported to also make contact with rods, rods primarily transit signals through rod bipolar cells [77].

The longevity and intensity of these bipolar responses are dependent on the bipolar cell type [84, 77]. Some spikes are instantaneous and transient, such as those from bipolar type 1 cells, while other spikes are delayed and more sustained, such as those from bipolar type 4 cells. With the wide variety of cell types and classes, the bipolar cell population can then efficiently transmit light-sensitive information with different temporal characteristics.

2.2.2.4 Amacrine Cells

In the inner layer of the retina, amacrine and ganglion cells form synaptic connections with bipolar cell terminals. Amacrine cells exhibit a wide range of morphologies, fields of view, and computational functions [67, 170, 174, 175]. For instance, the mouse retina is estimated to have around 63 different amacrine cell types, with the functions of most currently largely unknown [299]. When excited by bipolar cells, amacrine cells provide inhibitory feedback to other bipolar, ganglion, and even amacrine cells [67].

A unique characteristic of most amacrine cells is their lack of axons, meaning that their computational processing occurs within the neuron's dendrites. For example, Starburst Amacrine Cells (SACs), which will be a primary focus in this thesis, are capable of computing centrifugal motion of a spot within a small range of their dendritic arbor [76].

2.2.2.5 Ganglion Cells

Ganglion cells are the final feature extractor of the retina and integrate input from bipolar and amacrine cells to compute complex features [238, 57]. Unlike other neurons in the retina, ganglion cells transmit spikes through their axons. These axons then merge to create the optic nerve which sends the computed features to the lateral geniculate nucleus and ultimately the visual cortex [66].

There are an estimated 40 different ganglion cell types in the mouse retina each with unique functions and morphologies [57, 238]. As an example, ON Directionally Selective Ganglion Cells (DSGCs) receive input from Starburst Amacrine Cells (SACs) to calculate the cardinal direction of motion for a bright moving spot [269]. In contrast, On-Off DSGCs determine the cardinal directions for either a bright or dark spot moving across the retina [17]. Collectively, the retina operates as a multilayer neural network that discretizes continuous visual information, enabling us to perceive and interpret the world around us.

2.3 Fluorescence Microscopy for Retinal Neuroscience

Microscopes are optical tools used to visualize objects that are typically too small to see with our eyes. In neuroscience, they magnify a region of interest in neural tissue so that individual neurons and networks can be more closely examined. While there are many different types of microscopes utilized in neuroscience, I will now give a brief overview of confocal laser-scanning fluorescence microscopy as that is the primary imaging modality used throughout this thesis. However, the techniques proposed in this thesis can be easily applied to other imaging modalities in neuroscience.

2.3.1 Microscopy

Light microscopy, first invented in the 17th century, remains one of the most important microscopy techniques for studying biological specimens [32, 259]. For this technique, the specimen is first illuminated by visible light focused through a condenser. The light can originate from a variety of sources, usually from a bulb or light-emitting diode. The observed specimen then absorbs the collimated light and subsequently emits a new signal that is passed through an objective lens. This lens magnifies the signal so that the specimen can be examined in detail. The xy optical resolution, r_{xy}, of a light microscope can be succinctly described through the Rayleigh criterion as follows:

$$r_{xy} = \frac{1.22\lambda}{NA_{cond} + NA_{obj}} \tag{2.1}$$

where λ is the wavelength of the light used and NA_{cond} and NA_{obj} refer to the numerical apertures of the condenser and objective lens respectively.

Light microscopy also enables focusing on different focal planes. Since the objective lens bends light differently depending on its position relative to the specimen, various focal planes can be viewed. This concept can be described by the following equation:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i} \tag{2.2}$$

where f is the focal length of the lens, d_o denotes the distance between the lens and the specimen, and d_i denotes the distance between the lens and the image view.

By changing the distance between the lens and the specimen, d_i, different depths of the specimen can be magnified. This is particularly useful in neuroscience, as

neurons reside in layers scattered at various depths throughout the tissue. The axial optical resolution of the microscope, r_z, can be estimated using Abbe's diffraction formula as follows:

$$r_z = \frac{2\lambda}{NA_{obj}^2} \tag{2.3}$$

2.3.1.1 Fluorescence Microscopy

Instead of illuminating a biological sample with visible light as with light microscopy, fluorescence microscopy (FM) employs a high-powered light source that excites fluorophores contained in the sample. This technique allows for more specific and higher sensitivity imaging [155, 237].

Fluorophores are chemical compounds designed to absorb and emit light within specific wavelength ranges [145]. These ranges, also known as excitation and emission spectra, typically span several hundred nanometers and have distinct peaks that span several tens of nanometers. If a fluorophore is exposed to sufficient light within its excitation spectrum, it will subsequently emit light at a slightly longer wavelength through a process called fluorescence.

With FM, there are two main methods of increasing the specificity of the microscope. The first method is to adjust the wavelength of the light source so that it optimally, or near optimally, excites the desired fluorophores while reducing the excitation of undesired fluorophores. Secondly, an emission filter can be employed so that only a certain spectrum of light can be observed, thereby also reducing unwanted signals.

The signal-to-noise ratio of the microscope can also be enhanced by scanning a laser beam across the biological sample. This technique, known as laser-scanning FM, reduces interference from surrounding regions by only illuminating a small region of the sample at a time [219]. The image of the sample is then acquired by systematically scanning the laser across the entire sample.

Image quality can be further improved by combining confocal microscopy with laser-scanning FM. In confocal microscopy, a pinhole is placed before the detector to filter out-of-focus light [199, 245]. As light is emitted from the specimen at various angles, the pinhole only allows light that comes directly from the specimen. This in turn increases the signal-to-noise ratio, as there are no photons from other focal planes hitting the detector.

2.3.2 Fluorescence Microscopy Imaging

To create a digital image of the focal plane, the light emitted from the specimen is captured by a detector. This detector can either be a charge-coupled device or a photomultiplier tube and are responsible for encoding the intensity of the signal with respect to a given spatial location [82, 216]. The fluorescence intensity of signal I excited from a laser with wavelength λ at the location x, y, z can be well described as:

$$I(x, y, z; \lambda) \propto L * C(x, y, z) * T(x, y, z) * \Phi * \epsilon(\lambda) * \gamma \tag{2.4}$$

where L denotes the laser power illuminating the specimen, $C(x, y, z)$ and $T(x, y, z)$ denotes the concentration of fluorophores and specimen thickness at location x, y, z respectively, Φ refers to the quantum yield, $\epsilon(\lambda)$ denotes the absorption coefficient of fluorophores at laser wavelength λ, and γ denotes the fraction of signal that passes through the emission filter [51]. Other factors, like photobleaching [68], can also significantly affect the observed fluorescence intensity but are omitted for simplicity and because the image data acquired in this thesis attempt to minimize the affects of such phenomena.

However, because of resolution limitations described in Section 2.3.1, the fluorescence signal is spread out through space [249]. This blurring of the signal can be described by the Point Spread Function (PSF) of the microscope which can be computed experimentally or theoretically. In most cases, the PSF can be adequately estimated with a 3D Gaussian blur [307] described as follows:

$$PSF(x, y, z; \lambda) = \frac{1}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} \exp^{-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}-\frac{z^2}{2\sigma_z^2}} \tag{2.5}$$

where σ_x, σ_y, and σ_z are the standard deviations describing the anisotropic 3D blurring effect given laser wavelength λ in the x, y, and z direction respectively. Other factors, such as electromagnetic interference, are omitted for simplicity as well.

The resulting fluorescence intensity for a given spatial location is described as follows:

$$I_{PSF}(x, y, z; \lambda) = PSF(x, y, z; \lambda) \circledast I(x, y, z; \lambda) \tag{2.6}$$

where $\circledast$ denotes a 3D convolution.

Additionally, the number of photons n observed by the detector, and thus the resulting observed fluorescence intensity by the sensor, follow a Poisson probability distribution [310]. This can be described as follows:

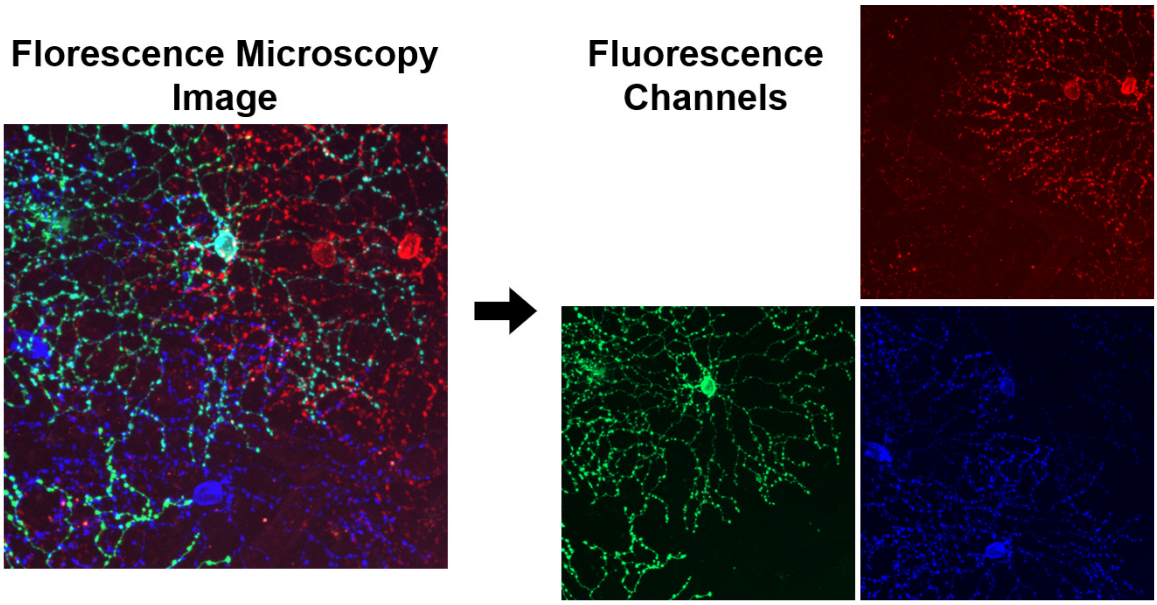


Figure 2.2: Fluorescence Microscopy image of neurons labeled with three distinct fluorescent proteins: EYFP (Yellow-Green), mCherry (Red), and TagBFP (Blue). By imaging the tissue with different laser wavelengths and filter sets, with varying laser wavelengths and employing different filter sets, separate images for each XFP are acquired, allowing researchers to examine the components of the neural tissue.

$$P(N = n; x, y, z, \lambda) = \frac{\mu^n e^{-\mu}}{n!} \tag{2.7}$$

where $P(N = n; \mu)$ is the probability of n photons received given an expectation of μ such that $\mu \propto I_{PSF}(x, y, z; \lambda)$.

2.3.3 Neural Labelling

While microscopy is essential for visualizing biological specimens, control over the emitted signals is equally important when designing experiments. Because neurons are translucent and intricately intertwined, distinguishing between different parts of the biological neural network can be challenging. Therefore, varying emitted signals across the network are required for meaningful analysis.

One approach is to use Golgi staining, a histological process that involves applying silver chromate to neural tissue [225]. The chromate substance is then fully taken up by a random subset of neurons, causing a reaction with lipids and proteins that ultimately turn the neurons black. Because only a subset of neurons undergo color transformation, various neural cell types can be more easily identified. The utility of combining staining with microscopy was first demonstrated by Ramón y Cajal who

used Golgi staining and light microscopy to determine the neuronal morphologies and network organization in the retina, thereby laying the foundation for modern neuroscience research [217].

Light microscopy, however, utilizes visible light to illuminate the specimen which often blends emitted signals together. As an alternative, fluorophores can be used with selective laser lines to achieve targeted light emission like in Fig. 2.2. This enables more refined and controlled visualization of neural components throughout the tissue. For example, fluorescent dyes, such as Alexa 488, can be directly injected into a single neuron so that only that neuron appears during imaging.

Instead of manually introducing fluorophores into individual neurons, the genetics of the neurons can be modified so that they automatically produce fluorescent proteins (XFPs) [275, 246]. This can be achieved by creating and injecting viruses containing XFP-producing DNA directly into neural tissue. The virus then randomly penetrates and introduces its genetic sequence into the neural DNA of a subset of neurons, prompting the neurons to produce the XFPs. Otherwise, animals can be genetically altered as embryos to create mouse lines containing XFP-producing DNA in neurons.

To achieve greater control over which neurons express fluorescence and generate complex fluorescent patterns, advanced genetic techniques are necessary. A common method involves utilizing the interaction between Cre-recombinase proteins and loxP sites in genetic sequences [240, 140, 92]. In essence, Cre-recombinase inverts genetic sequences located between inward-facing loxP sites. Thus, neurons will only express XFPs if inverted XFP-producing genetic sequences between inward-facing loxP sites are introduced to neurons that already express Cre-recombinase. These inverted XFP sequences can be introduced either through a virus [64] or through cross-breeding [1]. This approach enables individual neuron types to be labeled with XFPs. It also allows for individual neurons in a population to be tagged with a unique fluorescent profile.

2.4 Supervised Learning

Machine learning, at its most fundamental level, is the process of learning to transform a set of observed inputs $\mathbf{x}$ to the corresponding observed outputs $\mathbf{y}$ through a mapping function, or model, $F_\theta(\mathbf{x})$.

For linear relationships, model $F_\theta(\mathbf{x})$ can be represented as follow:

$$F_\theta(\mathbf{x}) = \hat{\mathbf{y}} = \theta\mathbf{x} \tag{2.8}$$

where θ is a vector of parameters describing the model and $\hat{\mathbf{y}}$ denotes the estimated observed value. There are several ways to deduce the values for θ that optimally fit the data; however, what is optimal requires a formal definition. In machine learning, this definition is known as a cost function that looks to describe how similar estimated values $\hat{\mathbf{y}}$ are to observed values $\mathbf{y}$. This cost function can be denoted as $J(\theta, \mathbf{x}, \mathbf{y})$. If the cost function is not dependent on the weights of the model, it can be shortened to $J(\hat{\mathbf{y}}, \mathbf{y})$. There are many different forms that the cost functions can take on, some of which will be studied more in-depth in later sections and chapters. The most common cost function is the Mean Squared Error (MSE) [286] and can be described as follows:

$$J(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{n}(\hat{\mathbf{y}} - \mathbf{y})^T(\hat{\mathbf{y}} - \mathbf{y}) \tag{2.9}$$

where n denotes the number of observed samples. For simplicity, we will use MSE as a cost function to continue to build the foundation of machine learning.

Now that the cost function can be formalized into a mathematical equation, we can learn the optimal values of θ, θ_{opt}, which best describes $\mathbf{y}$ by minimizing the cost function. For linear solutions and when using a MSE cost function, an attraction is that this can be achieved analytically:

$$\hat{\theta}_{opt} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{2.10}$$

2.4.1 Stochastic Gradient Descent

If we wish to model more complex relationships where analytical solutions do not exist, then $F_\theta(\mathbf{x})$ can be learned by iteratively tweaking θ so that $J(\hat{\mathbf{y}}, \mathbf{y})$ is gradually minimized over time. This iterative process, called gradient descent [232], is implemented as follows:

$$\theta_{j+1} = \theta_j - \eta\nabla_\theta J(\hat{\mathbf{y}}, \mathbf{y}) \tag{2.11}$$

where θ_j and θ_{j+1} are the parameters in model $F_\theta(\mathbf{x})$ at iteration j and $j+1$, η is the learning rate that determines how much θ changes each iteration, and ∇_θ is the gradient of $J(\hat{\mathbf{y}}, \mathbf{y})$ with respect θ.

In certain cases where $\nabla_\theta J(\hat{\mathbf{y}}, \mathbf{y})$ cannot be computed due to computational constraints or convergence, it can be estimated using a random subset of data points denoted $\hat{y}^{(i)}$ and $y^{(i)}$. Eq. 2.11 can be rewritten as:

$$\theta_{j+1} = \theta_j - \eta\nabla_\theta J(\hat{y}^{(i)}; y^{(i)}) \tag{2.12}$$

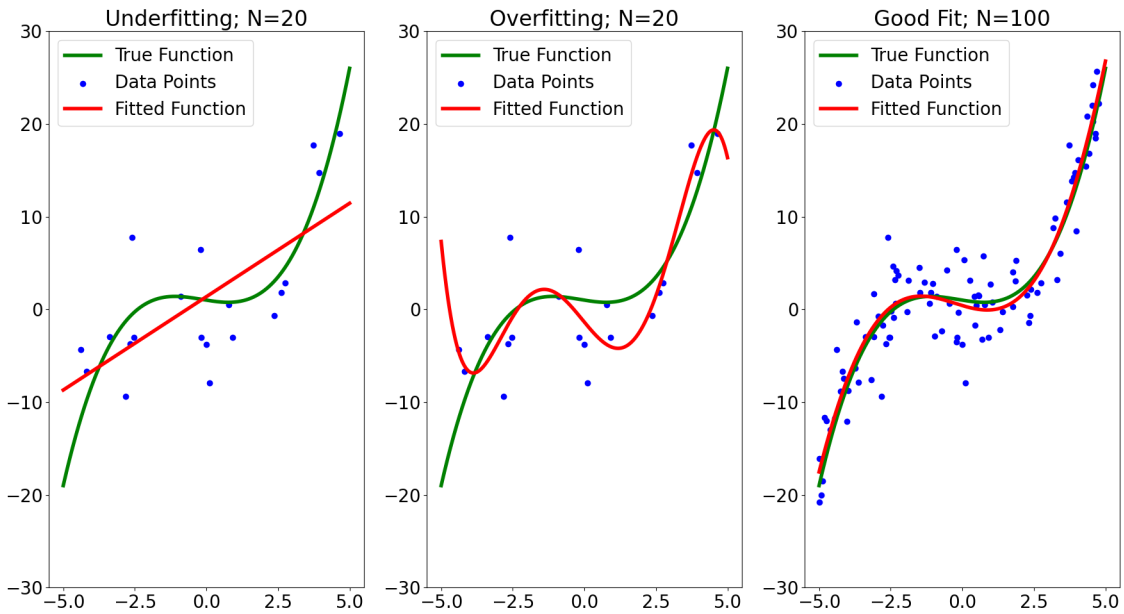


Figure 2.3: A demonstration of a learned model (red) underfitting and overfitting sampled data points (blue) generated from a polynomial with an order of 3 (green). *Left*: A linear model attempts to describe the polynomial relationship of the data, but fails to do so as the model does not contain enough complexity. In this case, the model is said to be underfitting the data. *Middle*: A higher-order polynomial model accurately predicts the sampled data but fails to capture the underlying relationship within the data. This is an example of overfitting. *Right*: When increasing the number of data points from 20 to 100, an accurate estimation of the underlying relationship is learned despite the complexity of the model. This highlights the importance of training deep learning models with large amounts of data.

2.4.2 Overfitting

Although we can learn the relationship between observed inputs $\mathbf{x}$ and outputs $\mathbf{y}$, the data used for training is not always representative of the entire data distribution due to sampling bias [116]. For example, if a machine learning algorithm is tasked with predicting the likelihood of someone developing dementia based on age and other related health factors, and the training data collected only consists of older people, then the trained model may overestimate the prevalence of the disease for the entire population.

Sampling bias can also result from randomly sampling a subset of data from the distribution. If the model has the capacity to learn more complex solutions than the complexity of the training dataset, it can memorize the desired outputs for each value in $\mathbf{x}$ without grasping the true relationship between $\mathbf{x}$ and $\mathbf{y}$. This overfitting needs

to be addressed when training models, particularly deep learning models that possess the ability to memorize vast amounts of data. This is evidenced by deep learning models accurately classifying images even when the pixels within these images are randomized [308].

There are several established methods to minimize overfitting. For all machine learning models, data is split between training, validation, and testing data. Training data will be used to adjust the parameters in the model, while validation data will only be used to determine the generalizability of models in the training process. If the performance on the training and validation data starts to diverge, then the model is said to overfit the training, and training is immediately stopped. Another method is to acquire more training data representative of the testing data distribution as seen in Fig. 2.3. The more data the model is trained on, the harder it is for the model to memorize all data accurately. Acquiring additional data, however, is often difficult, as we will see in Chapter 3. Alternatively, the complexity of the model can be reduced. While effective in some cases, this is typically avoided as larger models are able to learn more complex features.

Other ways to reduce overfitting include augmenting training data by modifying inputs without changing their underlying structure or semantic meaning [251, 278]. Data augmentation techniques are specific to the type of data being trained on. For example, for images, we can apply shearing, cropping, rotating, or translating the image, as these modifications convey the same meaning despite manipulation. Data augmentation prevents the model from memorizing individual samples and promotes learning of more generalizable features by increasing the number of data points trained on [251].

Finally, pre-training models reduce overfitting in deep learning [74, 104]. Instead of randomly initializing weights of a model, weights are initialized from previously trained models on a different dataset. The weights are then fine-tuned on the target dataset to learn task-specific features. This approach works well when the pre-trained network learns features relevant to the desired task [104]. For instance, pre-training on large networks trained on natural scene images can be useful for skin lesion classification tasks with limited data. The medical community has found this technique particularly useful, as creating individual datasets large enough to train deep neural networks is challenging.

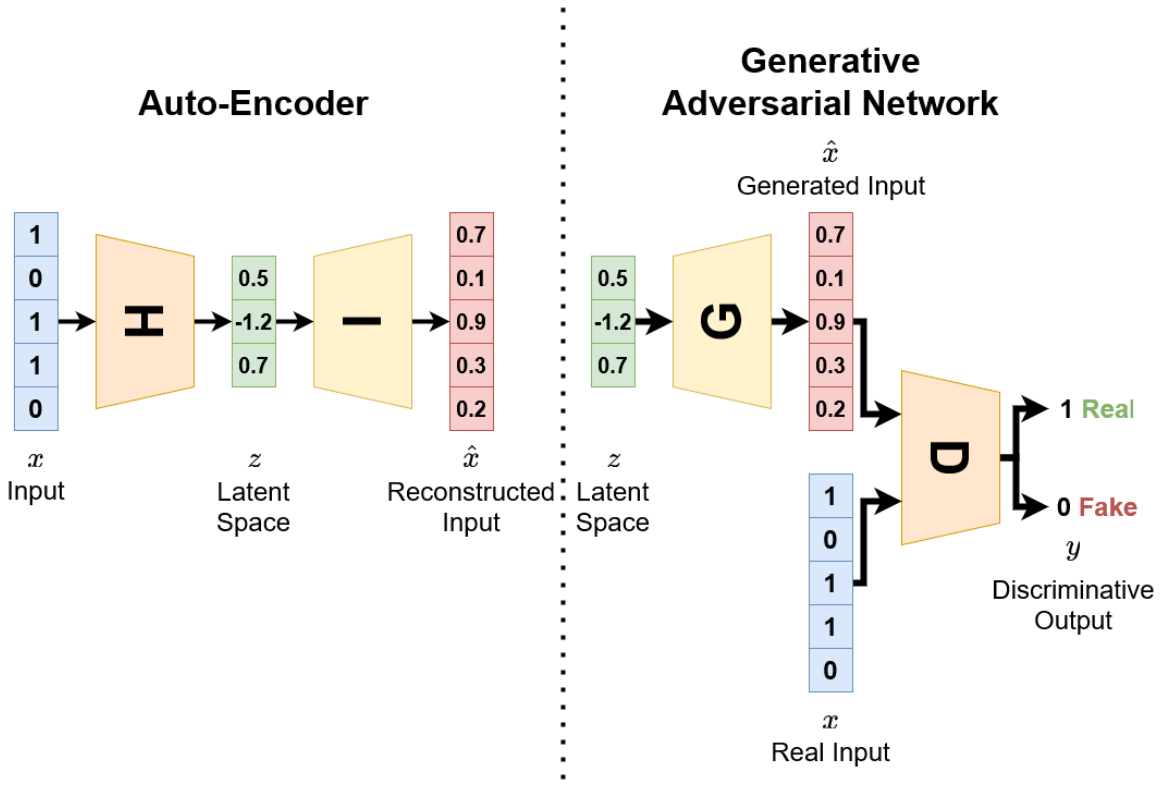


Figure 2.4: A schematic of an autoencoder (*Left*) and a Generative Adversarial Network (GAN) (*Right*). *Left*: The data sample x is funneled through a low-dimensional latent space z through an encoder H. Then a decoder I attempts to reconstruct x from z. After training, H can be used to create new data points. *Right*: On the other hand, a randomly sampled latent space z is passed through a generator G to create an realistic input $\hat{x}$. Then, random samples of x and $\hat{x}$ are passed through the discriminator D which is tasked to discern whether the input is from the real or fake distribution. After training, G can be used to create new data points.

2.5 Unsupervised Learning

All through Section 2.4, the learning process required a set of observed inputs $\mathbf{x}$ and corresponding observed outputs $\mathbf{y}$. However, learning can also be formulated such that $\mathbf{y}$ is not required.

2.5.1 Autoencoders

One way to learn the internal structure of the data without needing to train on observed outputs $\mathbf{y}$ is through an autoencoder [110, 232]. The objective of an autoencoder is to recreate $\mathbf{x}$ after perturbation such that the MSE cost function can then be written as:

$$J(\hat{\mathbf{x}}, \mathbf{x}) = \frac{1}{n}(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) \tag{2.13}$$

where $\hat{\mathbf{x}}$ are the reconstructed inputs after $\mathbf{x}$ is passed through the autoencoder.

To perturb $\hat{\mathbf{x}}$, autoencoders typically consist of two separate models: the encoder H and the decoder I. The objective of the H is to map the set of high dimensional inputs to a lower dimensional latent representation $\mathbf{z}$. This latent representation is then fed to I in order to reconstruct the input. A schematic of an autoencoder can be found in Fig. 2.4.

By funneling $\hat{\mathbf{x}}$ through a smaller feature representation, the network is forced to compress $\hat{\mathbf{x}}$ such that it filters out redundancies in the data while retaining only important features. Once fully-trained, H then functions as a non-linear dimensionality reduction technique or as a pre-trained network, depending on the availability of corresponding observed outputs $\mathbf{y}$ [110]. If no corresponding outputs exist, then clustering methods can be applied on $\mathbf{z}$ to elucidate meaningful structures in the data [254]. However, if corresponding data does exist, then the fully-trained model H can be appended with additional layers and the new network can be retrained for the task at hand. The processes of pre-training a network on one task, in this case an unsupervised task, and then fine-tuning for the desired task is known as transfer learning and it has been shown to improve performance especially when training data is limited [318, 274, 288]. As explained in Section 2.4.2, large networks are prone to overfitting on small datasets as a large network is able to memorize the data. However, if the network is originally pre-trained on another task, then the network is required to learn relevant structures which partially prevents it from overfitting.

Other methods of distortion include adding noise to $\mathbf{x}$, which can be denoted as $\mathbf{x}'$, and forcing the autoencoder to reconstruct $\mathbf{x}$ from $\mathbf{x}'$. In this case, the autoencoder framework is known as a denoising autoencoder (DAE) [280]. By using this training strategy, the DAE is not only forced to learn a compressed representation of $\hat{\mathbf{x}}$, but also learn how to differentiate between the signal and noise.

Additional constraints can also be applied to the compressed latent representation $\mathbf{z}$ so that more meaningful and interpretable features can be extracted, leading to better reconstruction. For example, sparsity constraints can be imposed onto the latent representation by minimizing a weight regularization loss in conjunction with the reconstruction loss [98]. This sparsity loss $\mathcal{L}_{sparse}$ is often defined as:

$$\mathcal{L}_{sparse} = \lambda \|\mathbf{z}\|_1 \tag{2.14}$$

where λ is a scalar to denote the weight of the sparsity constraint when training the autoencoder.

2.5.1.1 Variational Autoencoders for Data Generation

In the current formulation of autoencoders, there is a one-to-one mapping between $\mathbf{x}$ and $\mathbf{z}$. However, a probability distribution over the compressed latent space can be learned instead of a fixed mapping. This can be achieved using a variational autoencoder (VAE) [134]. Unlike traditional autoencoders, the VAE forces H to approximate the posterior distribution of $\mathbf{z}$ given $\mathbf{x}$, denoted as $q_\phi(\mathbf{z}|\mathbf{x})$, to match a normal distribution $\mathcal{N}(0,1)$ using the Kullback-Leibler (KL) divergence D_{KL}. The entire loss function is written as follows:

$$\mathcal{L}_{\text{VAE}} = J(\theta, \mathbf{x}, \mathbf{x}') + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel \mathcal{N}(0,1)) \tag{2.15}$$

where:

$$D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel N(0,1)) = -\frac{1}{2}\sum_{i=0}^{k}(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma) \tag{2.16}$$

such that σ_i and μ_i are the standard deviation and mean of the probability distribution of the ith feature in $\mathbf{z}$ given $\mathbf{x}$, and k denotes the number of features in $\mathbf{z}$.

By imposing the normal distribution constraint onto VAEs, they can learn robust structural representations of the input distribution [134]. The KL divergence also serves as a regularization term, which further helps disentangle features. Once the probability distribution of $\mathbf{z}$ is learned, it can be sampled to generate novel data points that have never been seen before.

2.5.1.2 Challenges with Autoencoders

There are several issues associated with using an autoencoder framework to learn rich latent representations of data. Firstly, the architectures are designed for compression and there is no control over which type of information is lost. Secondly, due to the limitations of data or architectural constraints, the decoder may imperfectly reconstruct the input. This might force an architecture to learn prominent features in order to minimize reconstruction loss, rather than meaningful features of the image.

The loss function can further complicate training as it compares the pixel-wise difference between the original and reconstructed image. This type of loss function is invariant to any translations in the image, even if the translations do not alter the

meaning of the image. For instance, if we reconstruct an image of blood vessels, but the image is shifted several pixels in any direction, the image will still contain the same blood vessels but the reconstruction loss will appear high. As a result, the loss function fails to effectively capture the quality of the reconstruction. Additionally, if too much noise is injected into $\mathbf{x}$, it is possible that the DAE has trouble decoupling the signal from noise which would further decrease reconstruction accuracy.

2.5.2 Generative Adversarial Networks

Similar to autoencoders, GANs consist of two components: a generator G and a discriminator D [96]. However, instead of performing a reconstruction task like autoencoders, G is tasked with generating realistic data from a random vector $\mathbf{z}$, while D evaluates the realness of the generated data. Both G and D then compete against one another so that any improvement in D requires a corresponding improvement in G. More specifically, given a latent representation $\mathbf{z}$ sampled from a predefined probability distribution, G aims to learn the distribution of $\mathbf{x}$ such that $G(\mathbf{z}) \sim p(\mathbf{x})$ where $p(\mathbf{x})$ is the distribution of samples $\mathbf{x}$. On the other hand, D is tasked with learning to differentiate between $G(\mathbf{z})$ and $\mathbf{x}$. This is outlined in Fig. 2.4 and can be summarized as follows:

$$\mathcal{L}_{GAN}(D, G) = J(D(\mathbf{x}), \mathbf{y}_T) + J(D(G(\mathbf{z})), \mathbf{y}_F) \tag{2.17}$$

where $\mathbf{y}_T$ and $\mathbf{y}_F$ are vectors indicating real or fake sampled datapoints respectively. While the formulation of Eq. 2.17 requires a supervised learning strategy, the labels $\mathbf{y}_T$ and $\mathbf{y}_F$ can be imputed for free as we know whether the input to D is generated or real by our experimental design. Therefore, GANs learn the distribution of $\mathbf{x}$ from only the data itself.

In many ways, GANs and VAEs are similar. The GAN generator and VAE decoder, for example, both learn the mapping $\mathbf{z} \mapsto \mathbf{x}$. In GANs, however, $\mathbf{z}$ is predefined while it is learned in VAEs. This framework offers numerous benefits. For one, it ensures that all elements of $\mathbf{z}$ are independent and identically distributed variables which help to disentangle the elements in the latent space. Secondly, replacing the reconstruction loss with an adversarial loss addresses many of the issues outlined in the previous section, enabling a more complex loss function that is invariant to affine transformations to the data. This allows for the generation of high-quality data and enables novel training paradigms, such as domain or style transfer, which cannot be achieved using a reconstruction loss.

GANs, however, have their own unique shortcomings if not properly set up. Because both G and D are competing against each other with their own distinct objectives, it is not guaranteed that both G and D converge to the optimal solution. Rather, it is more common for GANs to diverge [53]. This occurs when one architecture consistently has the upper hand, causing the other architecture to struggle to adapt and improve.

A consequence of sub-optimal GAN training is mode dropping, in which the generator produces only a subset of the distribution of $\mathbf{x}$ instead of the entire distribution [271]. For instance, rather than generating the full range of skin lesion images, a generator might only produce images of a subset of skin lesion types which are easier to generate. In more extreme cases, complete mode collapse can occur [271]. In mode collapse, the trained GANs entirely disregard the distribution of $\mathbf{x}$ and output a single, memorized sample of $\hat{x}^{(i)}$ for all, or a large portion, of $\mathbf{z}$. Because there is no constraint on G to produce a variety of outputs like there was in a VAE, it is reasonable to assume that generating one realistic data point is enough to deceive D.

While GANs have the potential to learn complex data distributions, it is imperative that precautionary steps are taken to stabilize training and mitigate the likelihood of convergence to sub-optimal solutions. One method of stabilizing GANs is through the cost function used in the adversarial loss [201]. Early work featuring GANs utilized a Binary Cross Entropy (BCE) [96], a predominant cost function in classification for other deep learning tasks. When implemented for a GAN, the adversarial loss can be rewritten as:

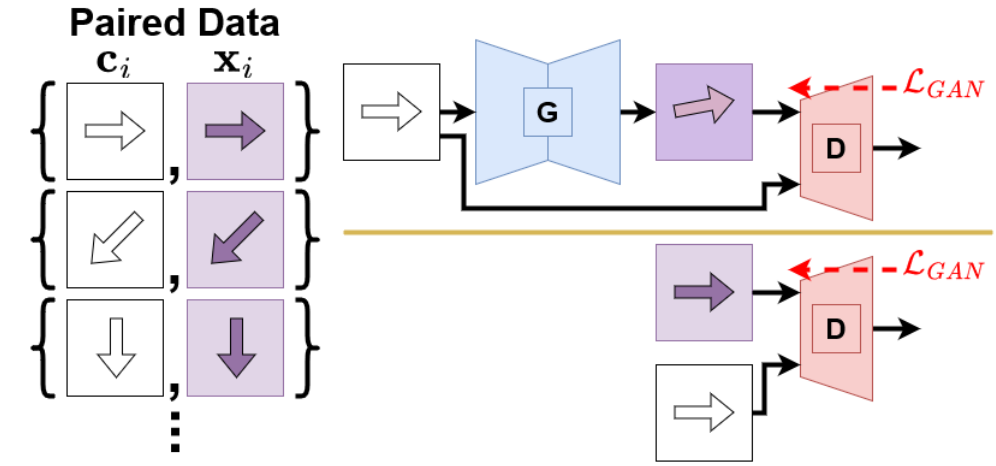
$$\mathcal{L}_{\text{GAN}}(D, G) = \mathbb{E}_{\mathbf{x}} \left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[\log(1 - D(G(\mathbf{z})))\right] \tag{2.18}$$

However, BCE requires $D(\cdot)$ to output a probability value between 0 and 1, which can squash gradients and therefore lead to sub-optimal training [231]. Therefore, alternative cost functions of GANs can be used.

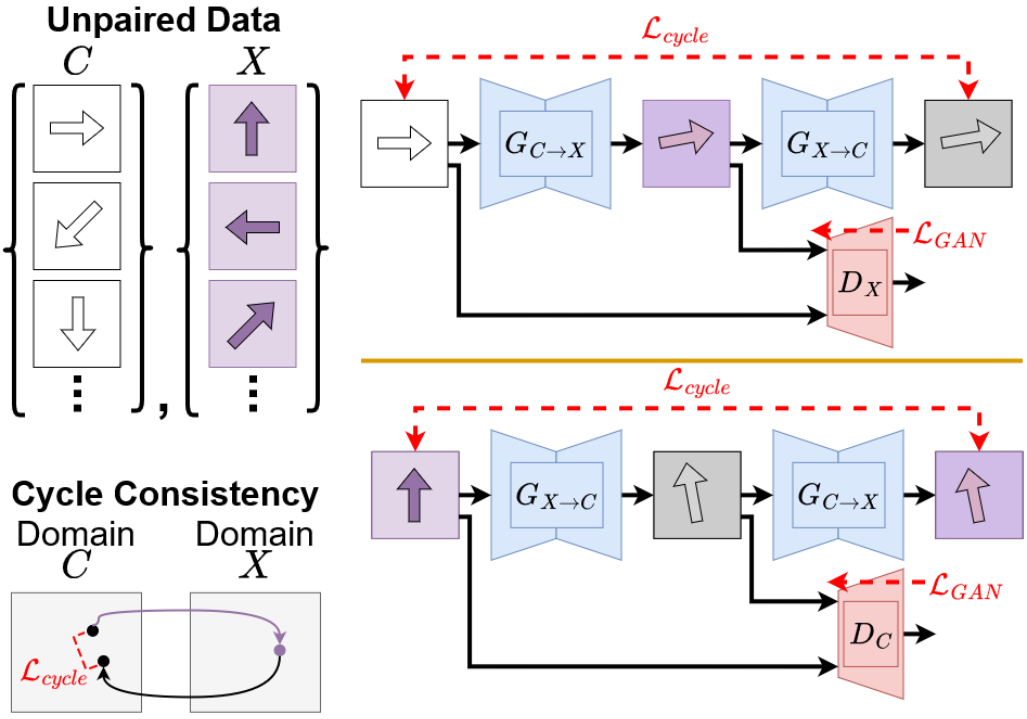
A Wasserstein distance has become one of the most important loss functions in GAN literature [8] and GAN loss Eq. 2.18 can be rewritten as:

$$\mathcal{L}_{\text{WGAN}}(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[D(\mathbf{x})\right] - \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[D(G(\mathbf{z}))\right] \tag{2.19}$$

To ensure that training does not diverge, the weights additionally undergo a Lipschitz constraint such that no one weight value explodes. Unlike Eq. 2.18, this equation attempts to quantify the difficulty of transferring one data distribution to the next, which encourages G to learn a variety of data points. Removing the log component



(a)



(b)

Figure 2.5: An example of data generation used (a) paired and (b) unpaired data. a) For paired data, for every conditional input $\mathbf{c}_i$ there exists one corresponding output $\mathbf{x}_i$ and vice verse. With this, we can ensure correspondence between $\mathbf{c}_i$ and generated output $\hat{\mathbf{x}}_i$ by passing both sets of $\mathbf{x}, \hat{\mathbf{y}}$ and $\mathbf{x}, \mathbf{y}$ to the discriminator. b) For unpaired data, there exists a set of data points in domain C and a differing set of data points in domain X. The relationship between domains C and X is not directly known. Therefore, in addition to a GAN loss $\mathcal{L}_{GAN}$, a cycle consistency loss $\mathcal{L}_{cycle}$ is used to learn a bijective mapping between both domains.

in the loss also allows for larger gradients to move throughout the network, making it easier for the G from the feedback of the D.

2.5.2.1 Conditional Generative Adversarial Networks for Corresponding Data

In the previous description of a GAN, G is tasked to map the latent representation $\mathbf{z}$ to the real distribution $\mathbf{x}$. The more complex $\mathbf{x}$, the likelier it is for GANs to encounter problems while training. Instead of learning the entire distribution of $\mathbf{x}$, subsets of the distribution can be learned to simplify training. This is achieved using a conditional GAN (cGAN), where additional information $\mathbf{c}$ is fed to both G and D so that the GAN is then conditioned $\mathbf{c}$ [183]. For example, $\mathbf{c}$ can be a classification of a skin lesion, and $\mathbf{x}$ can be the corresponding image of that skin lesion. In some cases, $\mathbf{c}$ can itself be an image and $\mathbf{x}$ be another representation of image $\mathbf{c}$. A schematic of cGANs can be found in Fig. 2.5 and the cGAN loss can be written as follows:

$$\mathcal{L}_{\text{cWGAN}}(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}(\mathbf{x}),\mathbf{c}\sim p_{\mathbf{c}}(\mathbf{c})}\left[D(\mathbf{x},\mathbf{c})\right] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\mathbf{c}\sim p_{\mathbf{c}}(\mathbf{c})}\left[D(G(\mathbf{z},\mathbf{c}),\mathbf{c})\right] \quad (2.20)$$

Conditioning the GAN on $\mathbf{c}$ offers several advantages. Firstly, it enables control over the generation of data during test time. A standard GAN relies only on a random sampling of $\mathbf{z}$, leading to a random generation of $\mathbf{x}$. If a data point with specific characteristics is required, such as an image of acne, generating this example image directly would be impossible. However, by conditioning G on $\mathbf{c}$, one element of the latent space is constrained to match the desired condition. As a result, we can randomly sample $\mathbf{x}$ that adheres to the $\mathbf{c}$ condition. In other words, if G is conditioned on all skin lesions, an image of a specific lesion can now be generated. This feature is particularly useful for data augmentation, where we might want to sample data with certain characteristics to improve the training process.

By conditioning D on $\mathbf{c}$, we also force G to generate more realistic data, as D has more context to assess the authenticity of the generated samples [183]. For instance, if a generated skin lesion image appears ambiguous, the discriminator, having additional information about the intended skin lesion, can better determine that the image is fake. This additional context helps to improve the quality of the generated samples, as the generator learns to create more accurate and distinctive outputs based on the conditioned information.

Because a cGAN effectively compartmentalizes the learning of the $\mathbf{x}$ distribution, it can help avoid mode dropping and collapse as there is less of the distribution to

learn. For skin lesions, G is conditioned on a specific lesion type, G will be forced to output results for all skin lesion types.

Although this conditional GAN formulation can yield better results than a standard GAN, it necessitates correspondence between $\mathbf{x}$ and $\mathbf{c}$. This undermines a key advantage of using GANs, which learns structures from unlabelled data.

2.5.2.2 Cycle Generative Adversarial Networks

In cGANs, both G and D are conditioned on $\mathbf{c}$ which necessitates correspondence between $\mathbf{x}$ and $\mathbf{c}$. However, by incorporating a cycle consistency loss [317], the mapping from $G : (\mathbf{c}, \mathbf{z}) \rightarrow \mathbf{x}$ can be learned without requiring correspondence in the data. To explain the cycle consistency loss $\mathcal{L}$cyc, let us first define a generator $G_{C \rightarrow X}$ which is tasked to translate input $\mathbf{c}$ from domain C to output $\mathbf{x}$ from domain X. The second generator $G_{X \rightarrow c}$ is tasked with the inverse domain translation. With this, the cycle consistency loss is shown in Fig. 2.4 and can be expressed as:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G_{C \rightarrow X}, G_{X \rightarrow C}) = \mathbb{E}_{\mathbf{c} \sim p(\mathbf{c})} \left[\| G_{X \rightarrow C}(G_{C \rightarrow X}(\mathbf{c})) - \mathbf{c} \|_1 \right] \\ + \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[\| G_{C \rightarrow X}(G_{X \rightarrow C}(\mathbf{x})) - \mathbf{x} \|_1 \right]\end{aligned}$$

In essence, $\mathcal{L}_{\text{cyc}}$ computes the reconstruction loss after translating input $\mathbf{c}$ to domain $\mathbf{X}$ and then back to domain $\mathbf{C}$. By doing so, it forces the network to learn a one-to-one correspondence between both domains, thereby preventing mode collapse. To illustrate why this is the case, consider that input $\mathbf{c}$ is mapped to a collapsed domain $\mathbf{X}$. It would then be impossible to regain the original data sample $\mathbf{c}$ from $\mathbf{x}$ as all information is lost when translated to domain $\mathbf{X}$.

Simultaneously minimizing both $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{cyc}}$ are necessary to produce realistic data conditioned on another variable. In this GAN formulation, correspondence in data is no longer required as this property is learned during training. Instead, all that is needed are two unpaired sets of unique data points from both domains.

However, there are limitations. For one, $\mathcal{L}_{\text{cyc}}$ utilizes a reconstruction loss that prioritizes the preservation of high-powered, low-frequency features between domains as discussed with autoencoders. Furthermore, this loss function requires an additional generator and discriminator which adds to computational complexity and memory consumption.

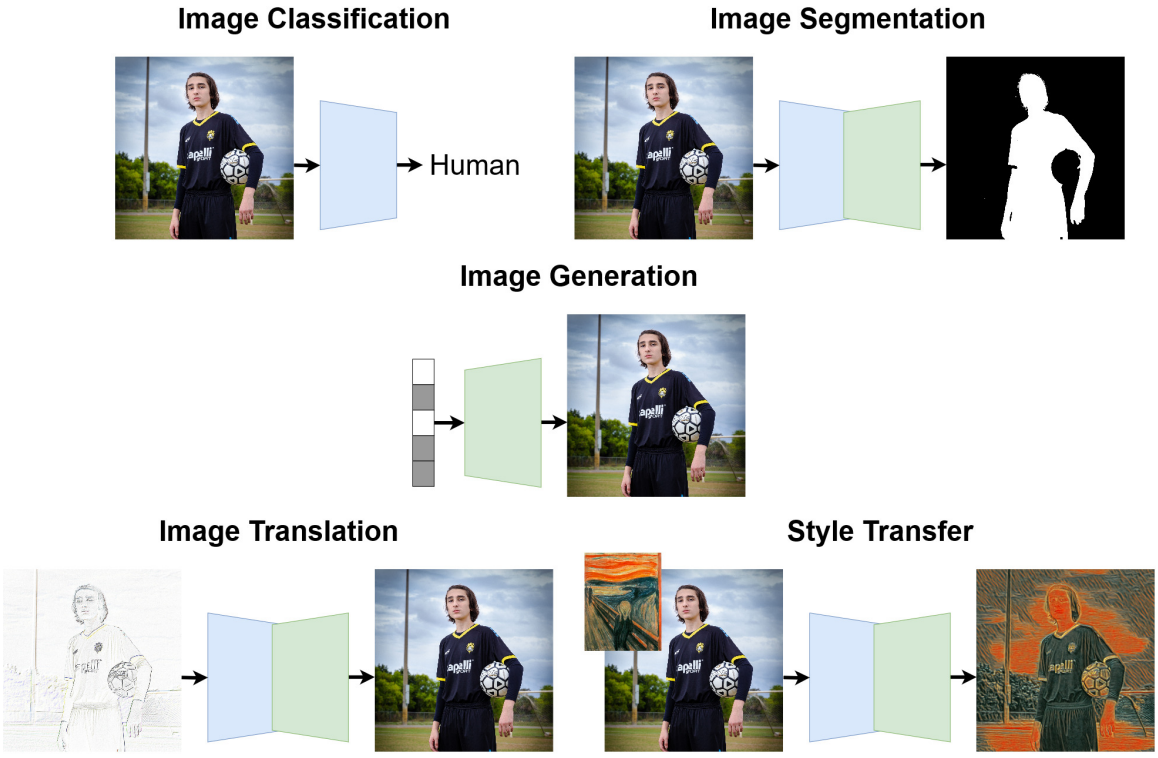


Figure 2.6: Applications of deep learning for images. In image classification, the entire image is categorized into one of the predefined image classes in the dataset. In contrast, image segmentation looks to classify individual pixels of an image that belong to a class. Novel images can additionally be generated either from a latent vector or from another image entirely. Finally, in style transfer, the contents from one image and the style from another image can be mixed to create a novel stylized image.

2.6 Deep Learning for Images

As seen in Fig. 2.6, deep learning has played a crucial role in achieving state-of-the-art performance across a wide range of image-based tasks, such as classification, segmentation, generation, translation, and style transfer. In this section, we will explore key architectures and contributions for each image-based task as this thesis builds upon these works. We begin with a brief overview of the most essential building block in computer vision: the convolutional layer.

Early image analysis architectures employed fully connected layers throughout the network, where each input variable is multiplied by a unique set of weights [146]. The layer can therefore be described as a matrix multiplication of the input $\mathbf{x}$ with the

weight matrix $\mathbf{w}$, as seen in Eq. 2.8. This formulation, however, effectively treats each image pixel as an independent random variable, thereby disregarding spatial relationships between pixels. Additionally, having a unique set of weights for every input variable significantly increases the parameters in the model which increases the likelihood of overfitting.

Convolutional layers were designed to exploit the spatial structures within images [147]. Unlike fully connected layers [230], these layers are comprised of a small group of shared weights that slide across the entirety of the image or feature space, performing element-wise multiplication and summation. We can formalize the convolutional layer as follows:

$$\mathbf{y}_{ij} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathbf{w}_{ab} \cdot \mathbf{x}_{(i+m)(j+n)} + \mathbf{b}$$

where $\mathbf{y}_{ij}$ is the output of the convolution at index (i, j), $\mathbf{w} \in R^{M \times N}$ is the kernel, $\mathbf{x}$ is the input image, and $\mathbf{b}$ is the bias vector. By sliding across the entire image, features can be recognized regardless of their position within the image while retaining spatial structure. Shared weights also reduce the number of parameters, allowing the network to learn more generalizable solutions.

The outputs of a convolutional layer, like that of fully connected layers, are then passed through a non-linear function f_σ to produce an ith feature in the output vector $\mathbf{a}$ such that:

$$\mathbf{a}_i = f_\sigma(\mathbf{y}_i)$$

While there are many non-linear functions that can be employed, the sigmoid function was first used as it constrained inputs between 0 and 1 and was thought to operate similarly to the all-or-nothing spike response of neurons [230]. The sigmoid function can be described as:

$$\sigma(\mathbf{y}_i) = \frac{1}{1 + e^{-\mathbf{y}_i}} \tag{2.21}$$

Convolutional Neural Networks (CNNs) are then formed by stacking several convolutional layers with non-linear activation units. As more layers are stacked, more complex features can be learned for an image task. The final layer of the CNN is dependent on the task at hand. If the desired output is categorical, like in image classification, a Softmax non-linear activation is utilized to compute the approximate probability of each category [147]. The Softmax layer can be described as:

$$\sigma(\mathbf{y}_i) = \frac{e^{\mathbf{y}_i}}{\sum_{j=1}^{n} e^{\mathbf{y}_j}} \tag{2.22}$$

2.6.1 Classification

One of the most essential image analysis tasks that has been addressed with deep learning for images is classification [252, 138, 146, 265]. For this task, images are passed to the model which then predicts the class the image belongs to (see Fig. 2.6). The number of classes and class types are predefined and typically defined for all training data. The earliest image classification architectures were designed for recognizing digits [147]. As these architectures began to surpass human performance, more complex image classification datasets were developed for natural scene images. For instance, ImageNet is a natural image dataset containing over 14 million images across 1,000 classes [138].

Although this thesis does not focus on image-level classification, we will provide an overview of three influential classification architectures that have introduced components advancing image analysis: LetNet-5 [147], VGGNet [252], and Inception Net [266]. These architectures have also been employed to enhance performance on various other image-based deep learning tasks, which will be discussed in subsequent Sections and Chapters.

2.6.1.1 LeNet-5

The first CNN consisted of convolutional, downsampling, and fully connected layers [147]. The downsampling layer employed was max-pooling, a non-parametric layer that takes the maximum value of a predefined region, typically a 2x2 square. The final layers of these CNNs then pass flattened feature maps to stack fully connected layers.

Incorporating max-pooling after every convolutional layer reduces memory usage within the architecture. Additionally, spatially compressing the feature maps enable subsequent feature convolutional layers to have larger receptive fields. While alternative downsampling layers exist, such as average pooling, max-pooling remains the standard pooling technique because it amplifies features and makes models more invariant to minor translations in the image [138]. However, because the network was small, it was largely suited to the classification of digits.

2.6.1.2 VGGNet

One of the seminal architectures in image classification is the VGGNet which achieved a top-5 test accuracy of 88.3% on ImageNet in 2014 [252]. This model is comprised of 16 convolutional layers, making it one of the first deep learning models.

The network was able to achieve success through its simple design by stacking convolutional layers with a 3x3 kernel size followed by max-pooling layers. The reduced kernel size limited the number of parameters needed to be trained, while the max-pooling layer retained the receptive field size of the convolutions. Additionally, Rectified Linear Units (ReLU) activations were employed instead of more traditional sigmoid functions. This non-linear activation function can be summarized as follows:

$$ReLU(\mathbf{x}) = \max(0, \mathbf{x}) \tag{2.23}$$

where x is the into the ReLU function. Because the gradient of ReLU is 1 for positive x and 0 for negative x, initial weights can be tuned more effectively as the gradient can more easily backpropagate through the network.

Although more recent networks outperform VGGNet on classification [265, 105], a pre-trained VGGNet on natural scene images has been employed for alternative applications, such as style transfer [128, 89]. This is because VGGNet utilizes a simple design without layer normalization, enabling the model to learn robust feature representations that are easily transferable across image domains [304]. More discussion on alternative applications of the VGGNet, like style transfer, is found in the future Section 2.6.3. A schematic of a VGGNet architecture can be found in 2.8.

2.6.1.3 Inception-Net

Since the VGGNet, there have been many additional architectures that have performed well on image classification tasks. However, the Inception Net has consistently been a top-performing architecture, achieving a top-5 test accuracy of 93.3% in 2015 [266].

While VGGNet utilized 3x3 convolutions throughout the architecture, the Inception Net introduced inception blocks throughout the network that combined feature maps produced by differing kernel sizes - 1x1, 3x3, 5x5, and 7x7. The varying kernel sizes allowed the network to learn multi-scale representations of the data.

Variations the Inception Net employed batch normalization after each convolution to help train larger networks [119]. This normalization technique can is described below:

$$\mathcal{F}_c^{\texttt{batch}} = \gamma_c \frac{\mathcal{F}_c - \mu_c}{\sigma_c} + \beta_c, \tag{2.24}$$

where $\mathcal{F}_c$ and $\mathcal{F}_c^{\texttt{batch}}$ are the CNN feature maps of the cth channel before and after normalization respectively, the scale and bias parameters are γ_c and β_c respectively, and the mean and standard deviation of the input feature map for a given batch are denoted as μ_c and σ_c respectively. This way, each feature activation in a batch is fit to a normal distribution. In addition to reducing covariate shift of features during training, batch normalization also equalizes the scale of features across network layers. This again makes it easier for gradients to flow throughout the network, improving training and convergence [265].

Further updates to the Inception Net, like residual connections, have maintained its top performance on image classification tasks [267, 264]. As a direct result, it has also been used for validating the performance of architectures designed for other tasks, like image generation. This will be further discussed in Chapter 5.

2.6.2 Segmentation

Another valuable application of deep learning for images is segmentation (see Fig. 2.6). In this task, rather than output a single classification for the entire image, every pixel is assigned a class. Thus, for instance, instead of determining which skin lesion type is in the image, segmentation would attempt to identify which pixels in the image belong to the skin lesion. Neuron segmentation is particularly useful in neuroscience as it allows researchers to study the morphologies and functions of neurons and will be discussed in more detail in Chapter 6. We again present two architectures that are foundational to the work presented in this thesis: FCN [168] and UNet [229].

2.6.2.1 Fully Convolutional Network

Because segmentation requires pixel-by-pixel classification, fully connected layers at the end of CNNs can be replaced with additional convolutional layers. The first architecture to exclusively employ convolutional layers was the FCN (Fully Convolutional Network) [168]. This approach offers several advantages. Firstly, using only convolutions maintains spatial structure throughout the network, thereby maximizing the utilization of spatial information. Moreover, replacing fully connected layers reduces the number of parameters used within the model, thereby also decreasing memory consumption and increasing a network's efficiency. Lastly, since the FCN outputs classes for pixels, while sliding across the image, a network is not constrained to a

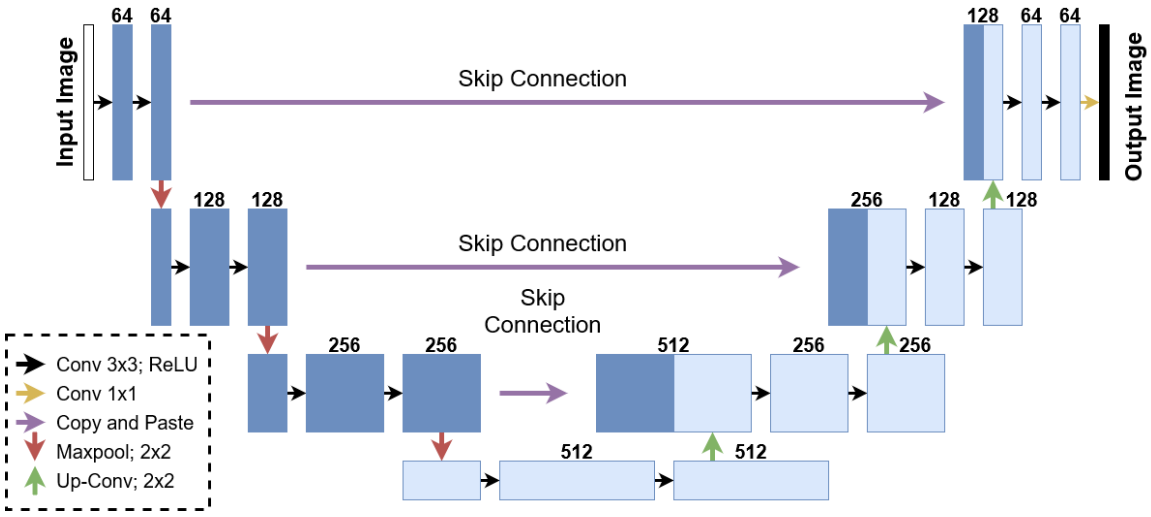


Figure 2.7: A schematic of the UNet architecture used for image segmentation. The input image is passed to 3x3 convolutional layers, then periodically downsampled used 2x2 max-pooling layers. Then, feature maps are upsampled, concatenated with feature maps in previous layers through skip connections, and processed by additional 3x3 convolutional layers. This encoder and decoder stream of information allows the architecture to integrate local and global information more accurately. Note that this schematic only contains 3 downsampling layers while the original UNet contains 4. This was done for simplicity.

specific image size. This flexibility allows a network to be trained on randomly sampled image patches while still being able to output classes for an entire image during testing. Sampling smaller patches additionally enables the use of larger batch sizes which allows the model to converge more quickly to the optimal solution.

2.6.2.2 U-Net

Building upon the FCN, U-Net (seen in Fig. 2.7) has become a seminal architecture for image segmentation [229]. Unlike the FCN, which only employs convolutional and downsampling layers, this architecture additionally utilizes upsampling layers. Resembling an autoencoder, downsampling layers are initially used to spatially compress features and upsampling layers follow to restore the feature space to the same dimensions as the original image.

However, unlike autoencoders, skip connections are employed throughout the network so that features from the encoder stage are shared with the decoder stage. By first funneling information through the downsampled layers, the network is forced to learn global representations of the image. While these global representations provide a good sense of what objects are present in the image, they contain information regard-

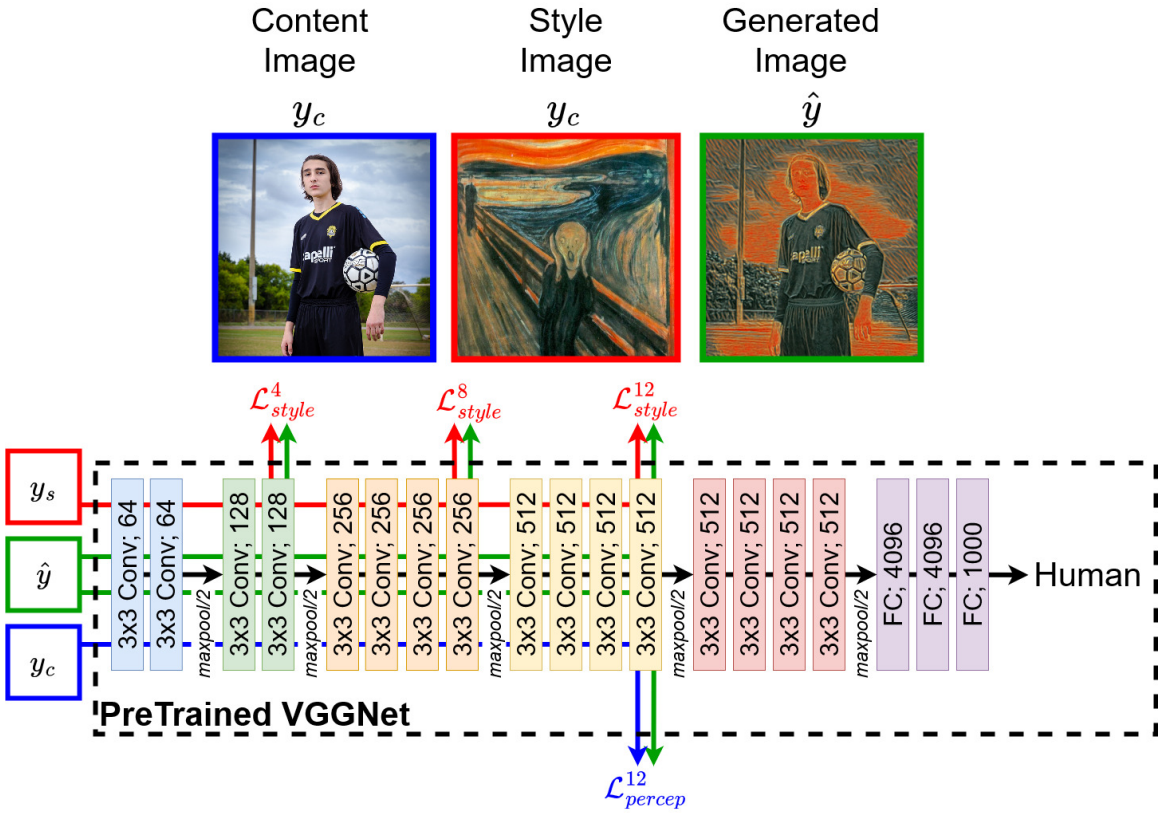


Figure 2.8: An overview of style transfer. A content image y_c and a style image y_s are used to generate an image $\hat{y}$ that is a mixture of the two. This is done by passing all three images through a VGGNet [252] and minimizing the style and perceptual loss at specific layers in the architecture. More details on how these losses are calculated can be found in Section 2.6.3.

ing the precise location of the objects to be segmented within the image. Therefore, skip connections share both local and global information with the decoder so that it can perform accurate segmentation.

Since the introduction of the U-Net, several models have adopted the same encoder-decoder architecture topology, with different forms of combined local and global features being proposed [190]. This approach has led to continued advancements in the field of image segmentation.

2.6.3 Style Transfer

In addition to producing classifications for entire images or individual pixels, CNNs can be used to generate synthetic images. One method of creating such images is through style transfer. The objective of this task is to combine the content of one

image with the style of another image (see Fig. 2.6). As an example, a photo of a human can be generated in the style of Edvard Munch's The Scream.

Gatys et al. first demonstrated the ability to mix the style - i.e. texture, contrast, color scheme, etc. - of one image with the content - i.e. a bridge, house, dog, etc. - of another image [89]. To accomplish this, we start with an image $\mathbf{x}$ containing white noise, a content reference image $\mathbf{c}$, and a style reference image $\mathbf{s}$ such that $\mathbf{x}$, $\mathbf{c}$, and $\mathbf{s}$ are all the same size. Let $\mathcal{F}^j(\mathbf{x})$ then be the jth feature map of the network when processing image $\mathbf{x}$ such that $\mathcal{F}^j(\mathbf{x}) \in R^{Cj \times H_j \times L_j}$ where C_j, H_j, and L_j are the channels, width, and length of the feature map.

The feature maps computed for each input are then utilized to compute the content loss $\mathcal{L}_{\text{content}}$, also known as the perceptual loss $\mathcal{L}_{\text{percep}}$, between $\mathbf{x}$ and $\mathbf{c}$ and style loss $\mathcal{L}_{\text{style}}$ between $\mathbf{x}$ and $\mathbf{s}$. The content loss between $\mathbf{x}$ and $\mathbf{c}$ can be concisely expressed as:

$$\mathcal{L}^j_{\text{content}}(\mathbf{x}, \mathbf{c}) = \frac{1}{C_i H_i L_i} \|\mathcal{F}^j(\mathbf{x}) - \mathcal{F}^j(\mathbf{c})\|_2 \tag{2.25}$$

This loss function computes the L2-norm between feature maps when processing $\mathbf{x}$ and $\mathbf{c}$. By using feature maps instead of pixels, image content can be decoupled from image style. Consider two photos of the same landscape, one taken during the day and another during the night. While the contents of both images are the same, the L2-norm between the images themselves will be high. The feature maps of the VGGNet, however, filter out image style as the network can robustly classify images regardless of image conditions. Therefore, the computed feature maps between images ideally should be similar.

The choice of feature maps significantly determines the objective of the loss function. For example, later layers filter out the most style content but are typically heavily downsampled, leading to less strict penalties in deviations of content. In contrast, earlier layers have more spatial precision but are more sensitive to the style of the image. The feature maps of middle layers are therefore usually utilized for computing the content loss [128].

Similar to content loss, the style loss between $\mathbf{x}$ and $\mathbf{s}$ can be written as:

$$\mathcal{L}^j_{\text{style}}(\mathbf{x}, \mathbf{s}) = \|G^j_{\mathcal{F}}(\mathbf{x}) - G^j_{\mathcal{F}}(\mathbf{s})\|_2 \tag{2.26}$$

$G^j_{\mathcal{F}}(\cdot)$ denotes the Gram Matrix computation of $\mathcal{F}^j(\cdot)$ which can be formalized as:

$$G^j_{\mathcal{F}}(\mathbf{x}) = \frac{1}{C_j H_j L_j} \sum_{l=1}^{L} \sum_{h=1}^{H} \mathcal{F}^j(\mathbf{x})_{l,h,c} \mathcal{F}^j(\mathbf{x})_{l,h,c'} \tag{2.27}$$

The Gram Matrix of feature maps is a useful representation of image style as it strips away all spatial content of the feature maps and encodes the relative combinations of features across the image. Similar to content loss, style loss is also heavily affected by the chosen feature maps. However, since image style can often be determined from a small region of the image, earlier layers of the network are usually employed.

In this training framework proposed by Gatys et al., the individual pixels of $\mathbf{x}$ are then transformed in order to minimize the following objective:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}^{j}_{\text{content}} + \beta \mathcal{L}^{i}_{\text{style}} \tag{2.28}$$

where α and β are both weights that determine the influence of the content and style loss respectively. A schematic overview of style transfer can be found in 2.8.

While this learning framework pioneered the field of style transfer, it requires the training procedure to be repeated for new content and style images. To streamline this process, Johnson et al. proposed training a network to perform real-time stylization of a content image [128]. The same content and style loss formulation was then utilized to adjust the weights of the network rather than the pixels of the output image itself. Additional work has been done to train a generalized network that stylizes images by taking in both a content and style image at test time. This circumvents any need to retrain architectures for new image combinations, however, requires significantly more data.

Furthermore, a pre-trained VGGNet on ImageNet has remained the staple architecture in image stylization. This can be attributed to the simplicity of the architecture and the computed features. For instance, since normalization is not used at any point in the architecture, features are not attenuated or shaped to fit distributions as seen with batch normalization. This allows computed features to be more interpretable and easily used when training other architectures. Because the objective of style transfer is purely aesthetic, there are limited uses of style transfer alone in biomedical images.

2.6.4 Image Generation

Networks can additionally create synthetic images using GANs (see Fig. 2.6). As described in Section 2.5.2, the generators in GANs are designed to generate realistic samples from a latent vector

Another synthetic image creation task is image translation, sometimes called domain adaption. This task attempts to transform an image such that it contains characteristics similar to that of another image. For example, a drawing of a human face can be translated into a colored photo of the human face such that the generated face in the photo contains the same facial features as the face in the drawing as shown in Fig. 2.6.

The most common way to perform image-to-image translation is through conditional GANs which have been described in Section 2.5.2.1. More specifically, this is achieved by conditioning G on image $\mathbf{x}$ so that G applies features on top of the underlying structure of $\mathbf{x}$. This negates the need for G to learn global structures and G can instead model local information. Image-to-image translation was popularized by the pix2pix network and has excelled at denoising and cross-modality domain adaptation for biological and medical applications [120, 157, 226]. As described in Section 2.5.2.2, cGANs can only be applied to paired data which are infeasible or expensive to attain in many contexts. For unpaired image translation, a cycleGAN can be used to ensure a bijective mapping between data [317].

Because the objectives of image translation share similarities with those of other image tasks, advancements in one task are often adopted for another. For example, image translation can be easily framed as an image stylization problem as both try to transform an image to have new properties while maintaining its original structure. Thus, to assist with convergence and improve image realism, style loss can be incorporated into image translation tasks [317]. Moreover, improvements in discriminator architecture or training paradigms in unconditional GANs can be observed in cGANs [183]. For example, rather than employing a single discriminator, as seen in ProGANs, multiple coarse-to-fine discriminators have been used in cGANs to boost performance [131]. Since image translation serves as the foundation of this thesis, a more detailed review of the state-of-the-art will be provided in Chapter 4 to better contextualize the propositions presented herein.

2.7 Deep Learning for Biomedical Images

Deep learning techniques employed for biomedical imaging largely overlap with the architectures used in general image analysis. GANs, such as DCGANs, have been utilized to create a variety of biomedical images, including MRI brain slices [235], skin lesions [27], and CT liver cancer [70]. More sophisticated GAN architectures,

like StyleGAN, have also been adapted for 3D MRI image synthesis by substituting 2D convolutions with their 3D counterparts [112].

Unconditional GANs are useful for biomedical imaging when the goal is to learn the underlying data distribution. However, cGANs offer the additional advantage of augmenting datasets directly as inputs are typically conditioned on a desired output. For instance, cGANs have been used to translate lower-quality or more easily-acquired images, such as CT scans, into higher-quality and less accessible images, like MRI images [63, 43]. This approach allows a set of labeled CT images and unlabeled MRI images to be used for training MRI segmentation models. Additionally, cGANs can be utilized to standardize features across datasets for subsequent analysis [162, 289]. This can be beneficial when attempting to utilize a trained architecture for data with slightly different properties or to prevent the image source from biases training. The Pix2Pix and CycelGAN framework network is commonly used as a domain translation architecture [120, 317]. Again, a more comprehensive analysis of image translation architectures for biomedical images can be found in Section 4.2.

In terms of segmentation, the encoder-decoder framework seen in a U-Net remains the primary architecture for segmenting biomedical images [229]. Variants of the U-Net, like SegNet [15], use pooling indices instead of feature maps in the decoder and have been utilized for tasks such as brain [113, 5] or skin lesion detection [192, 33]. Nested U-Nets add additional skip connections within a single U-Net to form multiple U-Nets, each producing its own segmentation predictions [316]. This allows task-relevant information to transfer more efficiently throughout the architecture, resulting in more accurate segmentation of thin regions typical in biomedical images. The Attention U-Net is another variant of the U-Net architecture that implements attention gates between feature maps to enable the network to focus on specific regions of the image during segmentation [196].

Of course, biomedical images also reside in 3D and the U-Net has therefore been extended to analyze 3D images. The most straightforward extension of the U-Net is to replace 2D convolutions and normalization with 3D convolutions. This was achieved with the 3D U-Net and the V-Net, which mainly differs from the 3D U-Net by using strided convolutions instead of max pooling for the encoder and implements a Dice loss [182]. Architectural advances in 2D encoder-decoders have been additionally translated to 3D encoder-decoders.

Biomedical images also exist in 3D and the U-Net has therefore been extended to analyze 3D images. The most direct adaptation of the U-Net to this context is

the 3D U-Net, which replaces 2D convolutions and normalization with 3D equivalents [54]. The V-Net builds on this by replacing maxpooling in the encoder with strided convolutions and implementing a Dice loss [182]. Architectural advances in 2D encoder-decoders have been additionally translated to 3D encoder-decoders. A more comprehensive discussion on deep learning architectures for the segmentation of FM images of neurons can be found in Section 6.2.

2.8 Classical FM Image Analysis

Automated analysis of FM images of neurons poses a significant challenge due to the complex structure and noise present in the images. As manual analysis of these images requires considerable effort, research has increasingly focused on automating the process. This section briefly outlines a few classical methods for automated tracing of neurons from FM images. While classical neuron reconstruction literature is extensive, it is mostly beyond the scope of this thesis as we are primarily focused on deep learning-based solutions.

2.8.1 Vessel Enhancement

A common method to analyze FM images of neurons is by applying a vessel enhancement filter which amplifies the pixels belonging to vessels while reducing image noise [83, 239, 180]. Most vessel enhancement filters utilize Hessian operations in order to capture the local changes within images. The Hessian for a 3D image I convolved with a Gaussian kernel with a standard deviation of σ can be formalized as follows:

$$H(x,y,z;\sigma) = \begin{bmatrix} \frac{\partial^2 I(x,y,z)}{\partial x^2} & \frac{\partial^2 I(x,y,z)}{\partial x \partial y} & \frac{\partial^2 I(x,y,z)}{\partial x \partial z} \\ \frac{\partial^2 I(x,y,z)}{\partial y \partial x} & \frac{\partial^2 I(x,y,z)}{\partial y^2} & \frac{\partial^2 I(x,y,z)}{\partial y \partial z} \\ \frac{\partial^2 I(x,y,z)}{\partial z \partial x} & \frac{\partial^2 I(x,y,z)}{\partial z \partial y} & \frac{\partial^2 I(x,y,z)}{\partial z^2} \end{bmatrix}. \tag{2.29}$$

This matrix can then be represented through a set of eigenvalues λ_1, λ_2, λ_3 and eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ such that

$$H(x,y,z;\sigma) = \begin{bmatrix} & & \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ & & \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} & & \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ & & \end{bmatrix}^{-1} \tag{2.30}$$

Because the eigenvalues determine the steepness of the signal in a given direction, they can be used to identify vessels.

One of the most common Hessian based vessel filters is the Frangi filter which can be summarized as follows [83]:

$$V(x,y,z) = \begin{cases} \exp\left(-\frac{R_A^2}{2\alpha^2}\right) \cdot \left(1 - \exp\left(-\frac{R_B^2}{2\beta^2}\right)\right) \cdot \exp\left(-\frac{R_S^2}{2\gamma^2}\right) & \text{if } \lambda_2 < 0 \text{ and } \lambda_3 < 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.31}$$

where α, β, γ are scalars determining the characteristics of the vessels to be enhanced and

$$R_A(\lambda_1, \lambda_2, \lambda_3) = \frac{|\lambda_2|}{|\lambda_3|} \tag{2.32}$$

$$R_B(\lambda_1, \lambda_2, \lambda_3) = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}} \tag{2.33}$$

$$R_S(\lambda_1, \lambda_2, \lambda_3) = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \tag{2.34}$$

Minor deviations can be made to Eq. 2.31 to create alternative vessel enhancers, such as the Sato filter [239] and Meijering filter [180]. However, the Frangi filter was used as the primary vessel enhancer in our literature due to its greater flexibility and effectiveness for our requirements. It is worth noting that these vessel enhancers can be estimated by CNNs, as the entire equation is differentiable. To introduce some nonlinearity in the vesselness equation, Eq. 2.31 can be incorporated into the network as demonstrated in Frangi-Net [86]. However, we found that this approach overly constrained the networks throughout our thesis.

2.8.2 Automated Based Tracers

In addition to vessel enhancers, creating binary representations of neurons in the FM image is also useful [166, 3, 179]. The canonical approach consists of four stages: filtering the FM image, generating seed points, representing the seeds as graphs, and determining the optimal graph that represents the neuron. We will now briefly discuss each of these stages.

Images are first filtered in order to amplify the local signal while reducing noise. Techniques employed include vessel enhancement filters [83, 154], anisotropic filters [305, 315], or steerable Gaussian kernels [187], or Laplacian of Gaussian filters [52]. After denoising the image, seeds are generated either manually or automatically throughout the FM image. A simple automated seed generation algorithm consists of thresholding the denoised image but is sensitive to noise [294]. Alternatively, a vector field can be used to identify attractors or local maxima [305, 285]. The vector field can additionally be used to learn an active contour around the dendrites [52, 285].

Once seeds are generated, they are connected to form a graph [305, 210, 294, 56]. The weight between nodes can be determined by the distance between seeds, signal continuity, and signal intensity. One way to compute these weights is by following the local ridges to then connect individual nodes directly [12]. Alternatively, a minimum spanning tree representing the neuron is then constructed [56, 181, 305, 22]. After each seed is connected, edges are slowly pruned so that the resulting graph resembles the branching pattern of a real neuron. A popular method is to use All-Path-Pruning where edges are progressively eliminated which don't abide by certain rules that govern neuron trees [211, 293].

It is important to note that machine learning has been utilized to increase the accuracy of neural tracing. For example, support vector machines (SVMs) have been to segment neurons as a pre-processing step [45, 126] or learn more accurate graph representations [45]. Our deep learning approach for neurons segmentation, however, is entirely data driven and all features are learned. Our main segmentation architecture consists of a shallow CNN with residual connections to filter out local voxel noise and a 3D U-Net to learn the global features contained within the FM image. By combining both local and global scales, we see a marked improvement in neuron segmentation for extremely noisy FM images.

2.9 Conclusion

In conclusion, this thesis will integrate experimental retinal neuroscience and deep learning applications for images. The primary objective of the retina is to be able to extract features from the visual environment and transmit that information to the visual cortex. To accomplish this, the retina is comprised of a three-layer biological neural network with several varieties of neurons, each with a unique function. Despite the retina's complexity, we can explore its components through a combination of fluorescence labeling and imaging techniques. FM Images can then be analyzed using classical automated solutions. However, these approaches often falter when dealing with images that contain significant noise [179]. In contrast, deep learning presents a robust alternative for FM image analysis [228, 296]. Although the primary application of deep learning for image analysis has been in natural scene analysis, these techniques have been repurposed and developed for the biomedical imaging domain and now perform image analysis tasks with high accuracy.

Deep learning, however, requires a significant amount of annotated training data, which is often unavailable in experimental neuroscience. Thus, despite advances in

deep learning, its full analytic power cannot be efficiently employed by neuroscientists for FM images. In this thesis, we hypothesize and demonstrate that overcoming these challenges requires utilizing publicly available data and novel deep learning components designed specifically to handle the noise and sparsity of FM images of neurons. For instance, instead of fully convolutional discriminators, we introduce Gram Matrix computations in the discriminator to encode the styles of FM images more effectively. Furthermore, we introduce a 3-stage CNN that fuses local and global scales so that neurons can be segmented more accurately despite obfuscations of the neuron's signal.

Chapter 3

Fluorescence Microscopy Image Datasets of Neurons

Contents

3.1 Introduction 44

3.1.1 Challenges of Deep Learning for Experimental Neuroscience Images 45

3.1.2 Synthetic Datasets of Neurons 48

3.1.3 Publicly Available FM Datasets of Neurons 49

3.1.4 Publicly Available Volumetric Reconstructions of Electron Microscopy Blocks 50

3.2 Novel FM Datasets of Neurons 51

3.2.1 2D Synthetic Images of Bipolar Cells (2DSyn_BC) 52

3.2.2 3D Synthetic Images of Ganglion Cells (3DSyn_GC) 52

3.2.3 3D FM Images of Amacrine Cells (3DFM_AC) 54

3.2.4 3D FM Images of Bipolar Population (3DFM_BCPop) . . . 57

3.2.5 3D FM Images of Overlapping Amacrine Cells (3DFM_overAC) 58

3.3 Discussion 60

3.1 Introduction

In this chapter, we present 2 synthetic and 3 real FM datasets acquired to advance deep learning applications for FM images of neurons. It is important to emphasize that the data collected is designed to enable testable deep learning experiments, and not necessarily to advance the field of neuroscience. Useful datasets for deep learning experimentation require repeatable and controlled imaging protocols with reliable

Dataset Name	FM?	dim.	#N	#P	#F	xy-cross	brief description	sec.
2DSyn_BC		2D	NA	3	3	512×512	type 5i bipolar cell w/ blobs & voxel noise	3.2.1
3DSyn_GC		3D	NA	2	2	512×512	F-mini ganglion cell w/ structured noise	3.2.2
3DFM_AC	✓	3D	22	5	8	512×512	starburst amacrine cell w/ voxel noise	3.2.3
3DFM_BCPop	✓	3D	2	1	8	2048×2048	type 2 bipolar cell pop. w/ voxel noise	3.2.4
3DFM_overAC	✓	3D	9	1	1	1024×1024	overlapping starburst amacrine cells	3.2.5

Table 3.1: Summary of datasets proposed in thesis. The columns left to right correspond to the dataset name (Dataset Name), whether the dataset consists of real FM images (FM?), the dimensionality of the dataset (dim.), the number of individual stacks (N), the number of laser powers used in each stack (P), the number of unique frames collected (F), the xy-cross section of images (xy-cross), brief description of the data (brief description), and the section that the dataset is described in more detail (sec.).

methods for retrieving ground truths. All data presented in this section was collected solely by me; this includes intravitreal injections, retinal dissections, tissue fixing, and neuron filling.

In brief, our collection of datasets includes both synthetic and real FM image datasets of neurons with paired ground truths. Four of our datasets (2DSyn_BC, 3DSyn_GC, 3DFM_AC, and 3DFM_BCPop) contain images of the same neural content with varying imaging properties and configurations. For instance, our 3DFM_AC dataset features real FM images of 22 Starburst Amacrine Cells imaged with 5 different laser powers and 8 different frame averaging settings, resulting in a total of 880 unique images. The variety of image intensities and noise distributions present within each dataset makes them the first of their kind and enables the evaluation of deep learning models with respect to image quality in a controlled manner. We have also collected FM datasets (3DFM_overAC) featuring overlapping neurons. Unlike previous overlapping neuron data, this dataset contains multiple instances of overlapping neurons, making them more suitable to train machine learning algorithms. Before we begin describing the collected FM datasets in detail, I will first discuss the challenges of deep learning applications for experimental neuroscience images and follow with an overview of neuron datasets currently available to the public. A summary of the data can be found in table 3.1.

3.1.1 Challenges of Deep Learning for Experimental Neuroscience Images

Advances in architectural design and training strategies play an important role in deep learning research; however, recent breakthroughs in the capabilities of deep learning models can be primarily attributed to the increased availability of large and diverse

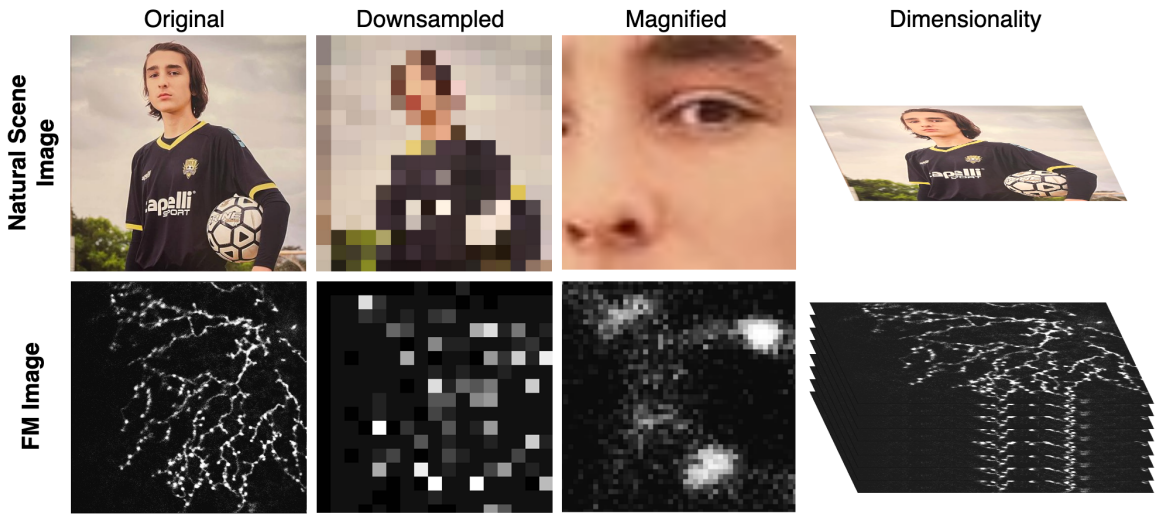


Figure 3.1: Comparisons between a natural scene image of a boy (top) and a FM image of neuronal dendrites (bottom). From left to right are the original, downsampled, magnified versions of the image. The right most column shows the differences between the images' dimensional. Note that the signal in FM images are much more sparse and corrupted than content in the natural scene image.

training datasets [101, 261]. These expansive datasets are essential for deep learning as they increase the precision and utility of trained models. With more training samples, models are less prone to overfitting and therefore can learn complex features that are generalizable to a wide variety of data. Large datasets additionally foster innovation by facilitating thorough evaluations of experimental models.

For image analysis, deep learning methods have been predominately developed for natural scene images due to the abundance of available datasets such as ImageNet [138], CIFAR-10 [137], VGG-FACE [205], IMDB-FACE [282], etc. In particular, the ImageNet dataset, which contains over 14 million natural scene images belonging to over 1,000 categories, has been instrumental for the development of CNNs capable of surpassing human performance on a variety of image-based tasks [252, 265, 106]. Experimental neuroscience images, however, present unique challenges that necessitate tailored solutions despite having many similarities with natural scene images. Therefore, the collection of experimental neuroscience image datasets is imperative not only to train accurate models on new experimental neuroscience tasks but also to promote architecture development designed specifically for experimental neuroscience images.

A key difference between natural scenes and experimental neuroscience images resides in their content as illustrated in Fig. 3.1. In natural scene images, content, such

as a face, is represented by a large number of pixels clustered closely together. Downsampling natural scene images would therefore obscure image details while preserving image semantics. As an example, a downsampled image of a face would still resemble a face despite the identity of the face being unrecognizable. Conventional CNNs exploit this spatial redundancy by compressing feature representations spatially to learn high-level features and conserve memory [146, 147]. The content of experimental neuroscience images, however, is significantly more sparse. For instance, in FM images of neurons, the pixels that represent the neuron often constitute less than 1% of the entire image. More so, the dendrites of neurons are thin vessel-like structures that can have a thickness of a single pixel on occasion. Therefore, downsampling these images results in a significant loss of both the image detail and image semantics.

A second major difference between natural scenes and experimental neuroscience images can be the level of noise present within the image (Fig. 3.1). While both image domains have some form of noise, noise is typically a much more dominant feature in experimental neuroscience images [233]. If the signal in the experimental neuroscience image is already weak, then pixel noise can corrupt the signal such that the signal is indistinguishable from noise. Thus, in addition to the sparse content, experimental neuroscience images can have a sparse signal. Finally, unlike natural scene images which are 2D, experimental neuroscience images are often 3D [253]. Not only does this exacerbate the previous two issues, but it also introduces novel issues like memory constraints during training.

The collection of these biological datasets is typically a laborious process. While general natural scene images can be captured in milliseconds by anyone possessing a smartphone today, experimental neuroscience images often require expensive equipment and trained personnel. This significantly limits the ability to acquire experimental neuroscience images to a handful of institutions with sufficient funding. The content of experimental neuroscience images often complicates the ability to gather large amounts of data. In health care, images of human subjects are protected by law in most cases and can therefore not be readily accessible in centralized, public databases [133]. For scientific experiments, the imaged specimen must be prepared which can take hours, days, and even months depending on the protocol followed. If the sample is compromised, the entire preparation process must be repeated. Finally, imaging acquisition time can substantially slow down the data retrieval process depending on the size of the imaged sample and the instrument used for imaging.

In addition to these difficulties, annotating acquired experimental neuroscience images can be equally, if not more, laborious. Unlike natural scene images which can

be readily analyzed by humans through crowdsourcing, experimental neuroscience images require expertise to annotate. In many cases, teams of experts are brought together to sift through the entirety of the collected data, which can take months [36]. Even with experts annotating data, there can be considerable variability among annotations due to the ambiguity of images [287]. It is therefore common to have multiple experts annotate individual images to create a composite annotation, further increasing the labor required to acquire datasets with robust ground truths.

3.1.2 Synthetic Datasets of Neurons

A straightforward method of acquiring training data for deep learning is to create synthetic FM datasets of neurons. This can be achieved by making use of Eq. 2.7 [279, 221, 260]. First, an image containing a volumetric representation of a neural body is convolved with a point spread function which can be estimated by a 3-dimensional Gaussian blur. Then, noise, typically Poisson noise, is added to the image given the probability of photon emission from the excited fluorescent protein.

Graph representations of neurons, known as neural skeletons, have been used as the content of the synthetic images [279, 221]. These images, however, tend to contain homogeneous neural and background signals which are inconsistent with real FM images. In real FM images, for example, there are regions of the images that might have more noise due to uncontrolled auto-fluorescence of biological material. Furthermore, while neural skeletons represent global 3D morphologies of neurons, they do not contain local volumetric information such as synapses or varicosities as seen in real neural dendrites.

More sophisticated efforts to improve the realism of synthetic images use volumetric reconstructions of neurons [260]. However, that effort focused on modeling color drift along the dendrites of the neurons and not noise patterns expected in images obtained during experiments. Finally, datasets of overlapping neurons have been created by randomly merging two high-quality FM images of individual neurons [151]. While that method effectively augments datasets and assists in training algorithms for segmentation, it could potentially bias the network to unrepresentative textures due to the addition of pixels and unnatural overlapping patterns of dendrites as actual dendrites stratify in structured manners governed by biological rules [251].

3.1.3 Publicly Available FM Datasets of Neurons

While FM images of neurons have always been collected for experimental neuroscience, they are usually in small quantities in uncontrolled image settings, making them unsuitable for deep learning. Research institutions have therefore made significant efforts to collect and share large-scale, standardized microscopy datasets of neurons to advance automated analysis technologies for neuroscience. In this section, I will only describe public datasets collected for the purpose of neuron reconstruction as that is the task set out to be addressed in this thesis and the type of data that is presented in this chapter. More information regarding microscopy datasets of soma bodies or calcium imaging can be found in the following citation [186].

One of the first initiatives to collect microscopy datasets was the Digital Reconstruction of Axonal and Dendritic Morphology (DIADEM) challenge for which a total of 6 datasets collected [38]. These datasets contained a variety of image types, ranging from CA3 hippocampus neurons of rats to olfactory axonal fibers from drosophila. Although the dataset sizes were limited to hundreds or even tens of images, images contained corresponding neural traces, serving as the foundational benchmark to compare state-of-the-art neuron tracing solutions. As a successor to DIADEM, the Allen Institute released BigNeuron, the biggest microscopy image dataset of individual neurons to date [208]. BigNeuron is a collection of 166 unique datasets, each containing thousands of individual neurons with corresponding "gold standard" ground truths. Data is additionally sourced from over 8 species, several brain regions, and different imaging modalities. This greatly expands on the amount and variety of image data available to design and test neuron segmentation. Lastly, the MouseLight project contains over 1,200 complete neurons from the mouse brain [72]. To accomplish this feature, over 20,000 unique, high-resolution FM scans were stitched together to create a composite image of the entire mouse brain. Thus, unlike previous datasets, the whole axonal fibers are captured and traced across several mm. Over 20,000 neuron reconstructions and their neural images have been collected from Drosophila through the FlyCircuits [49] and FlyLight Projects [124]. While there are other neural image datasets, like the National Institutes of Health's Brain Image library [23], they do not contain paired "gold standard" ground truths and therefore cannot be used as a primary training dataset.

However, there are several limitations to current datasets which reduce their utility in everyday laboratory settings. For one, they mostly consist of high signal-to-noise images. While high quality images are desirable, they are not always feasible in experimental neuroscience due to the image modality used or weak fluorescence expression.

Acquiring high quality images additionally requires prolonged laser exposure which can result in photo-bleaching or neuron death. Secondly, the datasets are only of individual neurons without morphological noise. FM images, however, often contain multiple neurons, especially when a population or sub-population of neuron cell types or when neurons are erroneously tagged. A future goal of image analysis is to then segment individual neurons in the presence of morphological noise, which deep learning algorithms have not yet been able to achieve and which these datasets do not address.

3.1.4 Publicly Available Volumetric Reconstructions of Electron Microscopy Blocks

In addition to public databases containing images of neurons, there is a rapidly growing number of publicly available volumetric reconstructions of neurons. Although not directly useful for FM image analysis, these reconstructions provide precise representations of a neuron's structure and, if neighboring neurons are also reconstructed, network connectivity. A central aspect of my thesis utilizes these public volumetric reconstructions to improve FM image analysis of neurons. Therefore, I will now briefly describe the collection process and currently available reconstruction datasets.

Because nanometer resolution is required to resolve small neural structures like synapses and dendritic spines, Serial Block-Face Electron Microscopy (SBEM) is employed instead of light microscopy [35, 65]. In this technique, fixed neural tissue is thinly sliced and imaged individually. Depending on the size of the tissue, this process can take days to weeks. Automated algorithms then align and stitch together slices, creating an Electron Microscopy (EM) block. To generate volumetric reconstructions, experts identify individual neurons and classify each pixel associated with the neuron throughout the entire EM block [34, 36]. This process is repeated for each neuron, sometimes numbering in the thousands, within the EM block. Although automated solutions exist for EM block tracing [123, 169], they are often unreliable, and teams of experts are still required to verify and correct reconstructions [211, 179].

Current publicly available EM block reconstructions include those from the Drosophilia medulla [268], Zebra fish larva brain [109], the mouse primary visual cortex [194] and more. However, the EM block of the retina produced by Helmsteadter, Briggman, and Denk is a seminal dataset in connectomics and instrumental in this thesis [107]. This EM block captures a $132 \times 114 \times 80\mu$m section of a retinal inner layer from a mouse with a resolution of $16.5 \times 16.5 \times 25$nm, resulting in a total block size of $8,192 \times 7,072 \times 3,200$ voxels. Over 20,000 manual labor hours from experts were then

spent analyzing the EM block to create skeleton traces for 950 neurons [107]. These skeletons were then used in combination with the help of crowdsourcing volumetric reconstructions from online volunteers to create reconstructions for over 2,000 neurons in the retina [36]. Reconstructions of most bipolar cell populations and several amacrine and ganglion cell populations can be accessed at `eyewire.com`. Although more retinal EM blocks have been collected and analyzed since then, this EM block is currently one of the most comprehensive EM block analyses in all of neuroscience and has led to significant insights in retinal processing. The time and resources required to prepare, image, and process such large EM volumes are still too great to make this approach of analyzing neural networks universally feasible [36, 34].

As I will describe further in later sections, the volumetric reconstructions of neurons can be used to create synthetic FM datasets and augment real FM training datasets. Briefly, these volumetric reconstructions are represented as 3D meshes - polygons that trace the entire surface of the neural body. Each mesh is therefore encoded by a set of vertices and edges which define each polygon. These meshes can then be voxelized to create a 3D binary image representation of the individual neuron. More specifically, the region enclosing the entire mesh is subdivided into 3D voxels with the space between voxels determined by the mesh's pitch. If a section of the neuronal mesh intersects a voxel, the voxel is marked as 1 in the 3D binary volume. Throughout this thesis, we frequently refer to the voxelized mesh volume as the 'neural content' as it contains contents of the neural body and provides the global structure for generated synthetic FM images as explored in this thesis.

3.2 Novel FM Datasets of Neurons

We will discuss the novel synthetic and real FM datasets aimed at enhancing the generalizability of deep learning for experimental neuroscience. We will first focus on 2 synthetic and 2 real FM datasets containing images of neurons with various optical configurations and image styles. The synthetic datasets can be reproduced using publicly available code (`https://github.com/MihaelCudic/BioSPADE`) and the real FM datasets are can be accessed through an external server (`https://data.mendeley.com/datasets/f6kk4364p4/`). Additionally, we will describe two newly acquired real FM datasets featuring overlapping neurons. Though not yet publicly available, these datasets are expected to become available in the future.

3.2.1 2D Synthetic Images of Bipolar Cells (2DSyn_BC)

In FM images, various sources of noise can obfuscate the signal produced by labeled neurons. One common source of noise is voxel noise as described previously. Although this noise can be mitigated by averaging several acquired frames, this technique cannot be implemented when images need to be acquired as quickly as possible, such as in calcium imaging [24, 99]. Another source of noise present in FM images can be autofluorescence from particles scattered throughout the tissue [242, 6, 185]. This type of structured noise is particularly prominent when imaging retinal tissue, as the vitreous left behind after dissection contains small, highly autofluorescent particles. Therefore, the first dataset created was a synthetic 2D FM image dataset (2DSyn_BC) created from reconstructed mouse Type 5i bipolar cells corrupted by significant amounts of pixel and structured noise as seen in Fig. 3.2.

A total of 41 meshes of previously reconstructed EM data provided by [107] were used to create this synthetic dataset. Synthetic images were constructed by first flattening a randomly sampled mesh along the Z-axis and voxelizing the mesh with a pitch of 200. To mimic the blurring effect seen in FM images as a result of PSF, voxels were then convolved with a Gaussian filter ($\sigma_x = \sigma_y = 1.5$pxs). All non-zero voxels were then set to 0.4 so that there would be background intensity consistent with that of real FM images. As a source of structured noise resembling randomly autofluorescent particles, we randomly inserted 2×2 squares to the image background. Frames were then multiplied by a randomly chosen scalar from the set $\{0.15, 0.30, 0.45\}$. This operation was done to simulate the effect of capturing images under different laser power levels. Additive Gaussian noise was injected by sampling $x \sim \mathcal{N}(0, 0.8\sqrt{(I)}/f)$ where x is the output voxel, I is the mean intensity of the voxel value, and f is the number of frames to be averaged which was randomly chosen from the set $\{1, 2, 4\}$. All values were chosen based on their ability to produce a variety of heavily corrupted images while still maintaining the structure of the neurons.

3.2.2 3D Synthetic Images of Ganglion Cells (3DSyn_GC)

Morphological noise is another source of noise that can make it challenging to accurately analyze FM images. This noise can arise from heterogeneous expression of fluorescent tags in genetically-modified tissue, unwanted channel-bleeding from neurons expressing strong fluorescent markers, or even unintentional injection of fluorescent dyes into neighboring neurons. We model this form of structured noise in a second

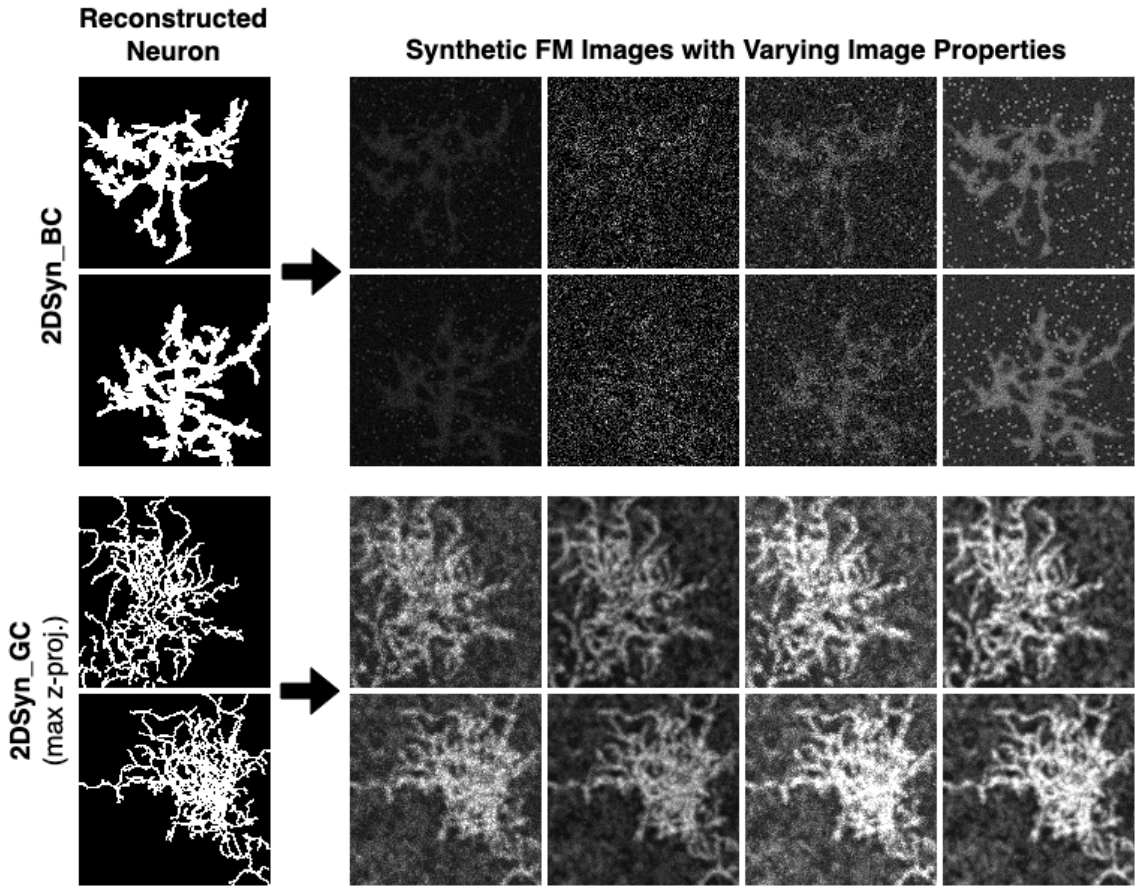


Figure 3.2: (*Right*) Sampled images from the 2DSyn_BC and the 3DSyn_GC dataset with varying image properties derived from (*Left*) volumetric reconstructions of neurons. These synthetic images were created to generate an unlimited number of controlled images containing pixel noise and structured noise to comprehensively test our machine learning architectures on noisy neuron segmentation. Note that the 3DSyn_GC dataset is a 3D dataset, so maximum z-projections of 3D images are displayed.

dataset of synthetic 3D FM images of F-mini Retinal Ganglion Cells (3DSyn_GC) (Fig. 3.2).

A total of 25 meshes of individual 3D-reconstructed ganglion cells along with their relative spatial locations were provided by [107]. All meshes were voxelized with a pitch of 1000 with their relative spatial locations preserved. Each mesh was additionally convolved with a Gaussian filter ($\sigma_z = 1.0, \sigma_x = \sigma_y = 1.5$pxs), and the background was set to 0.1 to again mimic the PSF and background intensities found in real FM images. An anisotropic filter was used as FM typically has higher xy-resolution than z-resolution. Unlike the 2DSyn_BC dataset that contained one neuron per image, all individual neurons were combined so that the entire neural population

would be shown in the image. For each synthetic image, however, an individual neuron was then identified and amplified ($\times 1.5$) so that it was brighter than the background neural population. Images were multiplied by a scalar value randomly sampled from $\{0.4, 0.6\}$. Additive Gaussian noise was again injected into the image by sampling $x \sim \mathcal{N}(0, 0.25\sqrt{(I)/f})$ with the number of frames chosen from the set $\{1, 2\}$. As a result, images resembled individual neurons with a significant amount of dendritic noise in the background. All values were again chosen qualitatively as they were able to present significant amount of voxel and background noise will still preserving the relative signal of the main neuron of the image.

3.2.3 3D FM Images of Amacrine Cells (3DFM_AC)

While synthetic datasets are a convenient tool for training and testing architecture designs because of their ability to generate vast amounts of data, they cannot replace real FM datasets. A new FM dataset of Starburst Amacrine Cells (SACs) expressing EGFP (3DFM_AC) was acquired to address this gap (Figs. 3.3 and 3.4). This dataset is unique in that it is the only dataset that contains the same neural content in different FM styles, offering an opportunity to test architectures in various conditions (Fig. 3.4).

A total of 22 SACs were imaged at 5 varying laser powers (0.81uW, 1.85uW, 3.84uW, 7.71uW, 15.70uW) 8 times as shown in Figs. 5.1 and 3.6. This results in 880 images collected in total, making it a large FM dataset. The XY-pixel resolution of the images acquired were 0.621um and the distance between individual slices of the 3D image was 0.100um. Animal procedures were conducted according to institutional guidelines and approved by the NINDS Animal Care and Use Committee (ASP-1344). For the 3DFM_AC dataset, ChAT-Cre mice ([121]) were intravitreally injected with AAV-7m8-EF1a-BbTagBY virus. The virus was made by NIH's viral core by encasing the EF1a-BbTagBY plasmid (catalog number #45185 from Addgene) in a 7m8 capsid. A 7m8 capsid was used as it was said to have the best penetration in neurons throughout the retina [62]. After testing on a variety of test mice, 2uL of the virus with a $\backsim 0.8 \times 10e12$ vg/mL titration was injected to achieve the desired sparsity of neuron labeling. Retinas were then dissected 3-4 weeks after injection to allow the virus to be fully expressed and to minimize the time between mouse injections. Younger mice less than 1-month-old were used to ensure that the tissue absorbed the virus without killing cells. Once the retinas were injected and dissected, they were fixed with 4% paraformaldehyde for 45 minutes and washed three times

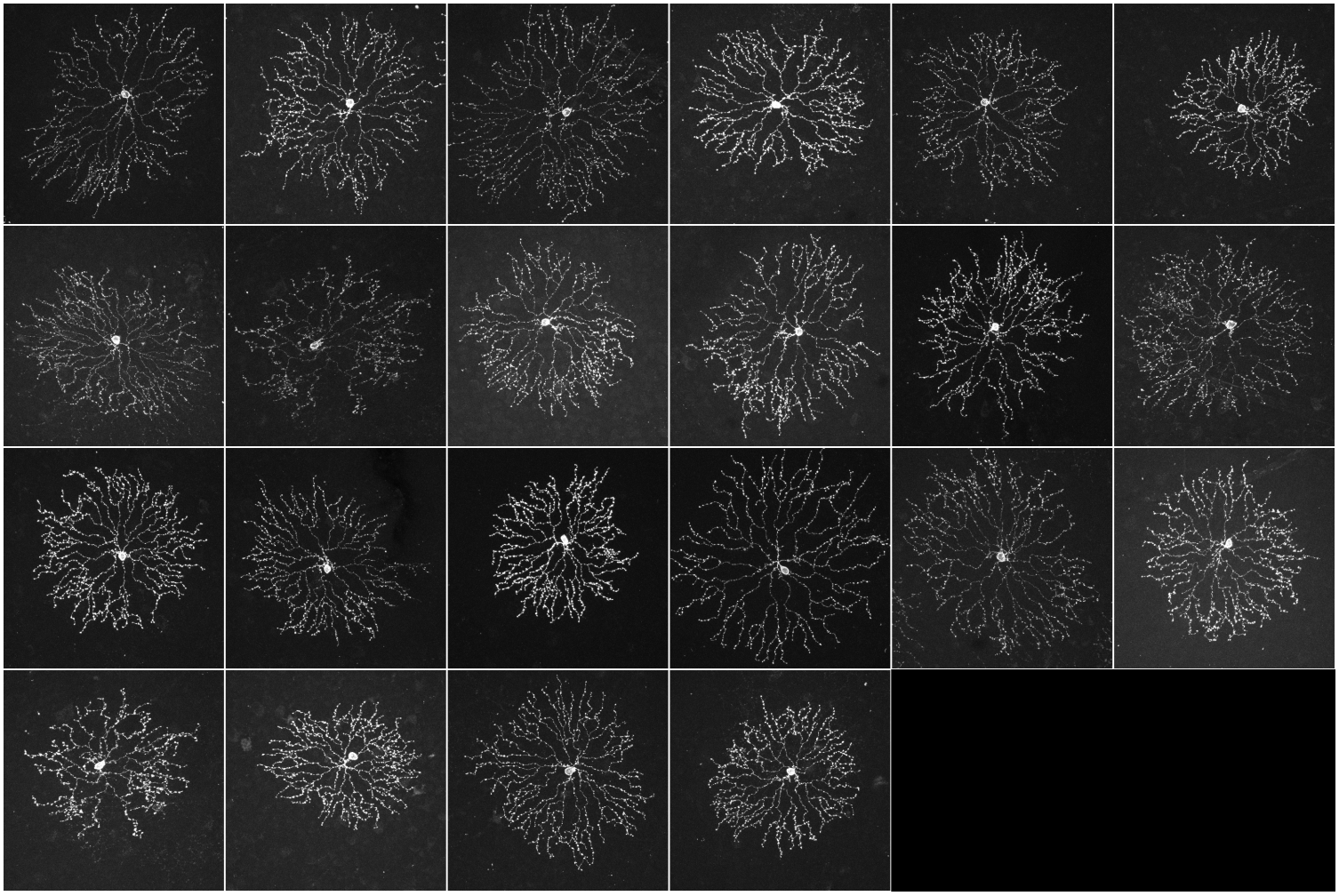


Figure 3.3: Maximum z-projections of all 22 Starburst Amacrine Cells imaged in our new 3DFM_AC dataset.

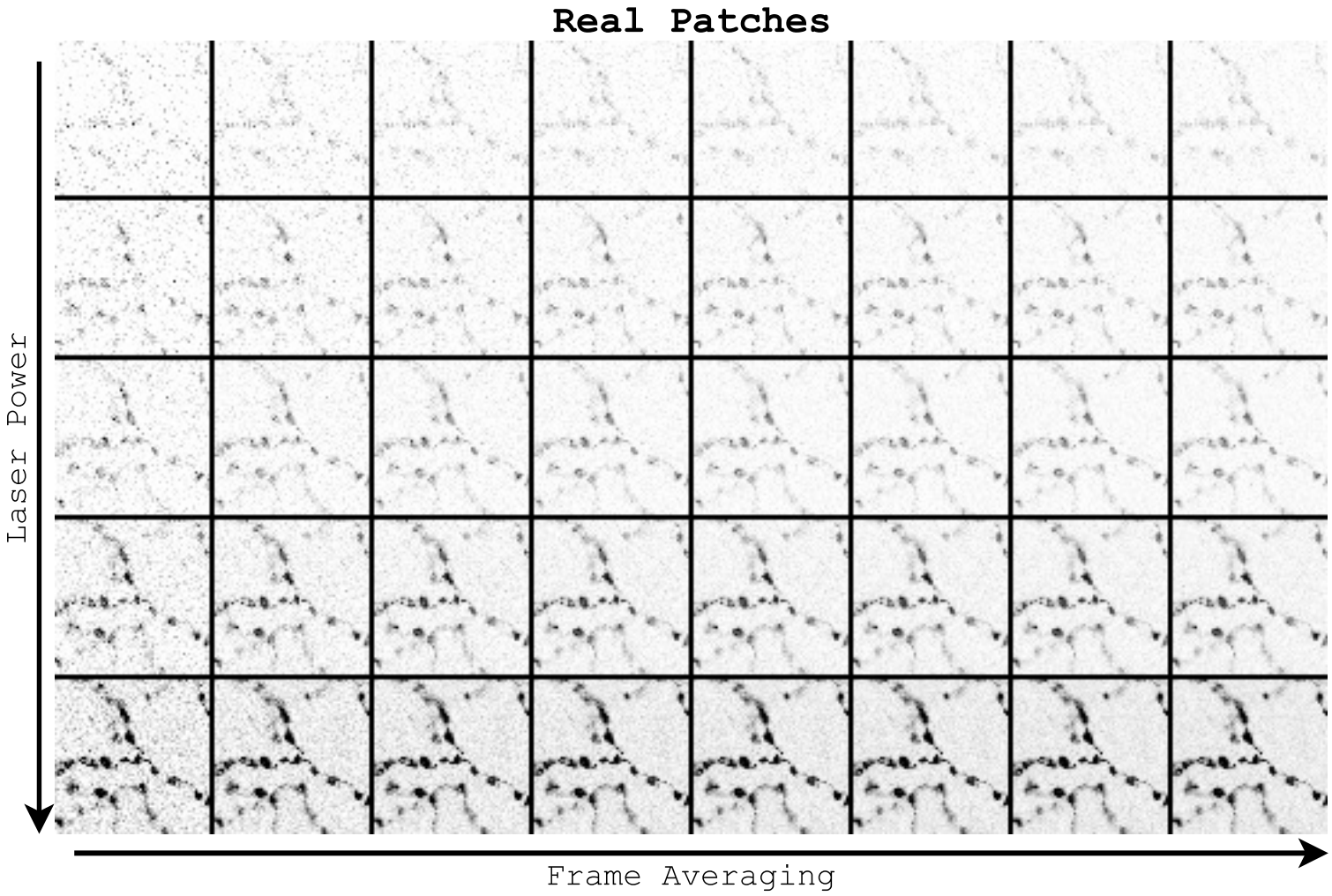


Figure 3.4: 64x64 patches (inverted for visual effect) sampled from the 3DFM_AC dataset at varying powers and frame averaging. A total of 5 different laser powers and 8 frame averaging combinations were acquired.

with phosphate buffered saline. A confocal microscope (Zeiss LSM 510) with a Plan-Neofluar 40x/1.3 Oil objective, a 488 laser line, and a BP505-530 filter was then used to acquire stacks of the fixed retinas. The complete spectrum of laser power and frame averaging configurations can be found in Fig. 3.4.

To minimize photobleaching and efficiently collect images, the laser power was set to the lowest level and gradually increased as regions of the tissue were fully scanned. Moreso, the microscope was programmed to successively capture time series, each consisting of 8 samples at a single focal plane, before moving on to the next focal plane. FM stacks with varying noise levels could then be produced by averaging any permutation of samples collected in the time series. This approach significantly reduced imaging time and laser exposure as there were only 8 sweeps per focal plane instead of 36 sweeps for a given laser power.

After completing a full scan of each region, slight translations between stacks were observed, with the most significant translation occurring in the z direction due to stage slipping. As this issue could not be resolved during acquisition, the stacks were computationally aligned post-acquisition. To accomplish this, we first generated the highest quality stack from each scan by averaging all eight frames. Next, the averaged stacks were aligned in the xy direction by maximizing the cross-correlation between the z-projections of each stack. After determining the xy alignment, the same process was applied to align the stacks along the z-axis. The cross-correlation between the stacks utilized for alignment can be summarized as follows:

$$(f \star g)(u, v) = \sum_{y=0}^{N-1} \sum_{z=0}^{P-1} f(x, y) \cdot g(x - u, y - v) \tag{3.1}$$

where f and g denote two separate images, x and y denote the location of a pixel, u and v are the displacement values, and N and P are the sizes of the window used to compute the cross-correlation.

Because all images were aligned, ground truths can be easily retrieved by performing a 3D Frangi vessel enhancement filter [83] on the image stack obtained with the highest laser power (i.e. the highest quality stack). This ground truth, while patchy, is sufficient for our purposes.

3.2.4 3D FM Images of Bipolar Population (3DFM_BCPop)

A second real dataset of 2 FM stacks (3DFM_BCPop) was acquired of GFP-expressing Bipolar Type 2 Cell populations from Syt2-GFP mouse retinas (Fig. 3.5). Regions of 318.2um×318.2um were imaged with an xy-resolution of 0.155um and a z-resolution

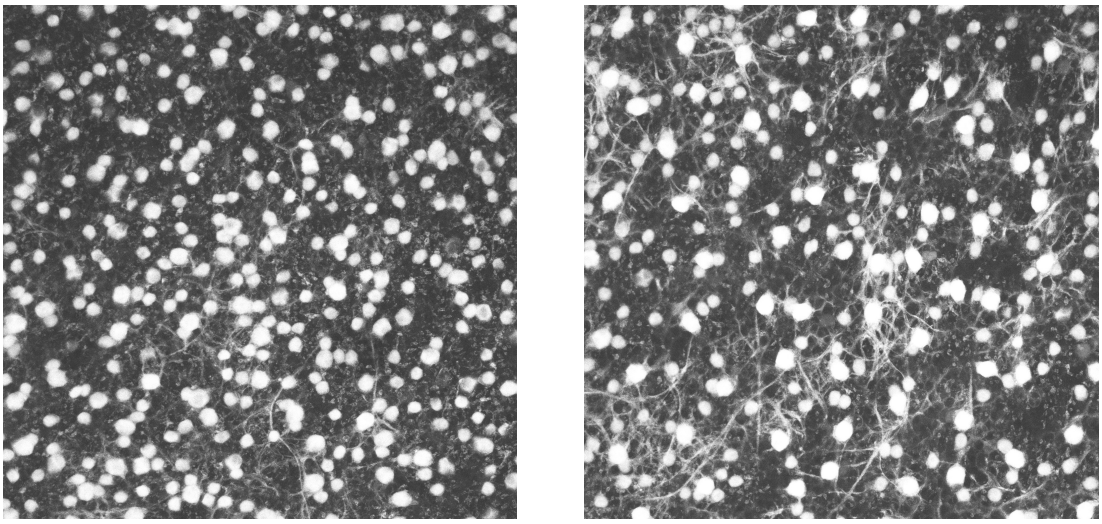


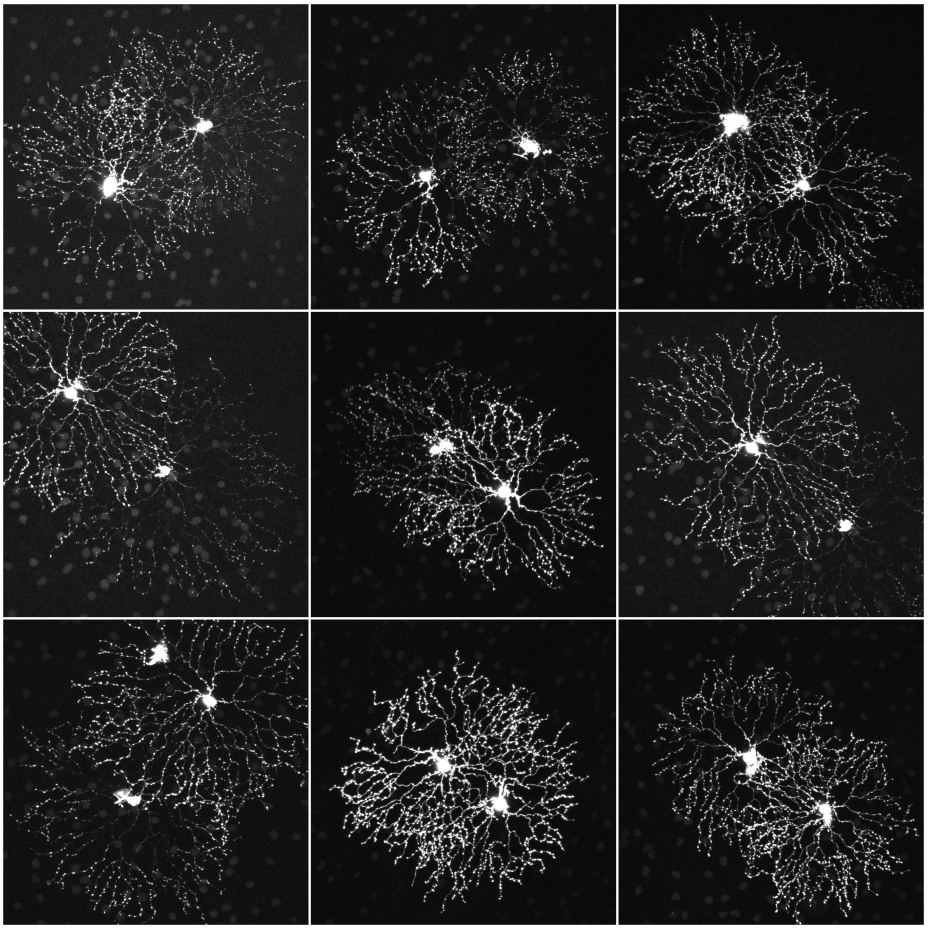
Figure 3.5: Maximum z-projections of all 2 Bipolar Cell populations imaged in our new 3DFM_BCPop dataset.

of 0.100um, resulting in stacks having an xy-cross section of 2048×2048 pixels. Both stacks were collected with the same process outlined in Section 3.2.3 such that 8 unique frames acquired. However, only a single laser power of of 15.70uW was used. Ground truths were then created using two 3D Frangi filters on fully averaged stacks, one for the somatic layer ($\alpha = 0.01, \beta = 10, \gamma = 15$) and one for the axon terminal layer ($\alpha = 0.01, \beta = 10, \gamma = 8$) (see Section 2.8.1 for more details on the filter implementation).

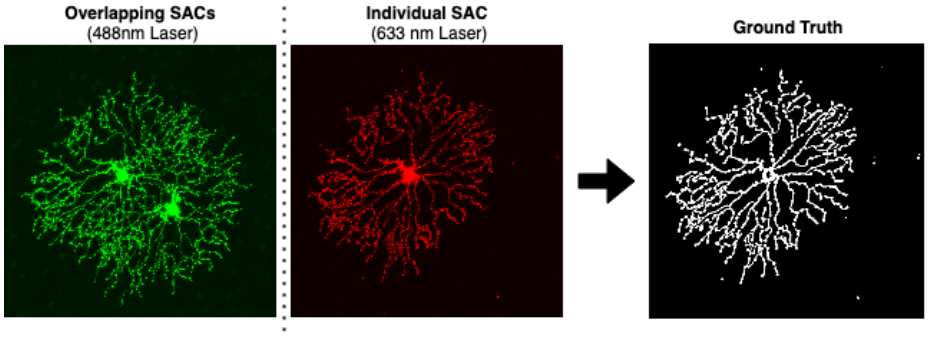
3.2.5 3D FM Images of Overlapping Amacrine Cells (3DFM_overAC)

For the first overlapping neuron dataset, a total of 9 FM stacks were acquired of overlapping SAC pairs from live retinal tissue as seen in Fig. 3.6a. Both SACs were filled with Alexa 488, a green fluorescent dye, and one of the SACs was filled with Alexa 647, a far-red fluorescent dye. By utilizing dyes with differing excitation and emission spectra, the microscope was configured so that two separate images were produced: one of the overlapping neurons and one of just a single neuron (Fig. 3.6b. To correct for any spatial translations when acquiring these two sets of images, alignment was performed using the automated cross-correlation protocol outline in Section 3.2.3. This alignment ensured that every image of overlapping SACs had a corresponding image of a single, labeled SAC.

More specifically, ChAT-tdTomato+ mice (Jackson Laboratory), which labels all SACs with tdTomato, were used for our retinal dissections. While tdTomato is a



(a)



(b)

Figure 3.6: a) A collection of all 9 overlapping SACs included in the $3DFM_overAC$ dataset. b) An example image showing overlapping SACs captured in the green channel with a 488 nm laser, alongside a single SAC in the red channel obtained with a 633 nm laser. The individual SAC was then processed through a Frangi filter to determine the ground truth.

strong red fluorescent protein with an emission peak of 581 nm, both Alexa dyes (Alexa 488 and Alexa 647) used to fill neurons were chosen to ensure that channel bleeding was minimal. Dissected tissues were then whole-mounted, placed in the microscope stage for visualization, and perfused with Ames media (Sigma) at room temperature. Electrodes with a resistance of 20-50 $M\Omega$ were filled to the tip with the 10-50 µM of Alexa dye and used to puncture the somas of targeted SACs. The dye was then delivered to the neuron using iontophoretic voltage pulses (10-kHz square wave, 50 ms, 15 V peak-to-peak). The electrode was promptly withdrawn from the SAC directly after the dye filled the cell body and replaced with an electrode containing the other combination of Alexa dyes. This process was then repeated on a neighboring SAC. The region of interest was first imaged using a 488 laser line and a BP505-530 filter to isolate the Alexa 488 signal. Images were acquired averaging 8 frames with a resolution of $0.31 \times 0.31 \times 0.5 \mu m^3$, thereby creating a high quality image of overlapping SACs. For the ground truth, a laser line of 561 and a filter of BP650-710 was used to isolate the Alexa 647 signal. Because this image only serves as ground truth, we only imaged with 4 frame averaging and with a resolution of $0.62 \times 0.62 \times 0.5 \mu m^3$. These ground truth images were then ran through a Frangi Vessel enhancement filter to acquire a binarized ground truth. A confocal microscope (Zeiss LSM 510) with a W Plan-Apochromat 40X/1.0 NA objective (Olympus) was used for both images.

3.3 Discussion

As discussed in the previous sections, several large FM datasets of neurons are available to the public, but their utility for deep learning applications in experimental neuroscience remains limited. One reason is that these datasets are mostly comprised of high-quality images of neurons, which can be difficult to acquire due to optical and experimental constraints. Secondly, while FM images acquired in real world experiments often feature morphological noise, images in these datasets are of single neurons. To address these limitations, we provide both synthetic and real FM datasets that feature unwanted image artifacts in a controlled manner, enabling deep learning algorithms to be trained and evaluated for broader neuroscientific contexts.

Our first two datasets, 2DSyn_BC and 3DSyn_GC, are synthetic FM datasets created from volumetric reconstructions of neurons. While we also offer real FM datasets, these synthetic datasets serve as easy benchmarks for deep learning models as we have complete control over image construction. This allows us to create

an arbitrary number of images corrupted in ways similar to those of real FM experiments, while automatically producing corresponding ground truths. For example, the 2DSyn_BC dataset consists of images with significant amounts of pixel noise, making it difficult even for experts to identify neural structures. Images in this dataset are also in 2D so that state-of-the-art models previously designed for 2D natural scene images can be compared to newly designed models. Our second synthetic dataset, 3DSyn_GC, contains images of individual neurons with weakly labelled neuropil spanning the entire background. Consequently, semi-global and global fields of view are required to properly isolate the main neuron, making it challenging for conventional deep learning models.

Although synthetic datasets offer some advantages, they cannot fully replace evaluation on real FM datasets. For this reason, we provide three FM datasets, each focusing on a specific type of image noise. The first dataset, 3DFM_AC, is a comprehensive dataset of 22 individual neurons imaged with 40 different optical configurations. Some image configurations produce significant amounts of voxel noise which further obfuscates the neurons already thin dendrites. However, because the images are taken of the same content, the ground truths can be recovered from high-quality images of the same neural content. This allows for the unique evaluation of deep learning models across a variety of noise conditions, therefore providing insights on the tradeoff between image analysis accuracy and laser exposure.

The 3DFM_BCPop dataset, a more complex dataset, was collected from neuron populations with varying imaging properties. These images are less sparse compared to those in the 3DFM_AC dataset but contain more uncontrolled structures such as erroneous dendrites and auto-fluorescence from nearby cell bodies. Nevertheless, similar insights regarding the trade-off between analysis performance and image quality can be obtained through this dataset. Although the ground truths encompass all the neurons captured in the image, additional work is needed to isolate individual neurons. This would then enable machine learning algorithms to learn instance segmentation of neurons within a population.

Our final real FM dataset, 3DFM_OverAC, consists of pairs of overlapping SACs, making it the first dataset of overlapping neurons. The ability to segment overlapping neurons in FM would greatly increase the utility of deep learning in experimental neuroscience as multiple neurons could then be studied simultaneously. While this dataset is of only 8 images, it is a necessary starting point for model development and will serve as inspiration for the collection of other overlapping neuron datasets. Successfully tackling this task will require models to learn the neuron's morphology

while still having the precision to classify individual pixels accurately. Because mouse SACs have unique outward and radial branching patterns, this dataset is an ideal testing environment to learn stereotyped, global morphologies.

Chapter 4

Unpaired Mesh-to-Image Translation for FM Images of Neurons

Contents

4.1 Introduction 64
4.2 Related Works 68
4.3 Methodology 70
4.3.1 Voxelized Mesh 70
4.3.2 Slice-to-Slice Translation 70
4.3.3 Generator Architecture 73
4.3.4 Discriminator Architecture 76
4.3.5 Objective 77
4.3.6 Random Sampling of Patches 78
4.4 Implementation 79
4.4.1 Data 79
4.4.2 Architecture 81
4.5 Results 82
4.6 Discussion 85

The following two Chapters are primarily from the work "Unpaired mesh-to-image translation for 3D fluorescent microscopy images of neurons" published in Medical Image Analysis where I am the first and only author except for my supervisors [58]. Therefore, all work was conducted by me alone.

4.1 Introduction

As discussed in previous chapters, fluorescence microscopy (FM) is an essential tool in neuroscience due to its ability to image deep within fixed or live neural tissue with high sensitivity and spatial resolution [176, 263, 290]. Image data collected through FM can then be used to study neuron activity, network organization, or tissue composition [99, 167, 214, 227]. However, extracting the relevant information from these images to test scientific hypotheses remains challenging as images are hard to annotate and limited. Efforts in the neuroscience community have been made to streamline image analysis using a mixture of crowdsourcing large FM datasets [39, 209, 24], automated neural analysis solutions [294, 291, 80, 301, 153, 314, 7, 313, 47, 153], and synthetic training datasets [247, 262, 255, 152]. While these approaches have met occasional success, they are limited in their ability to impact the general neuroscience community.

Public FM datasets, as seen in Section 3.1.3, are not fully representative of the broad range of images acquired in experimental neuroscience and cannot be solely relied upon to automate specific tasks. Even if more datasets are to be collected like the ones we proposed in Chapter 3, the collection of all datasets will still likely never capture the full variety of image characteristics conferred by different biological samples, experimental configurations, and optical calibrations.

To understand why this is the case, let's consider a research lab studying retinitis pigmentosa that wishes to automate the classification and counting of abnormal horizontal cells in the bird retina. While the research goal has widespread appeal due to its clinical relevance, the data collected and the computational task are niche. Firstly, horizontal cells, which stratify laterally across the retina, have morphologies that are different from other neurons in the retina and brain [207]. Secondly, the morphologies of horizontal cells differ across species. Bird horizontal cells, for instance, lack axons, while the horizontal cells of mice and primates have axons [207]. Lastly, horizontal cells in diseased retinas often appear in smaller quantities, exhibit reduced dendritic arborization, and are more distorted than their healthy counterparts [59, 171]. These three factors alone – neuron cell type, species type, and disease type – contribute to images with distinct yet consistent features that are not represented in centralized datasets. The most significant reason why current centralized datasets provide limited utility for this example, however, is because they are designed for neuron reconstruction and not neuron classification, which are entirely different tasks.

Current off-the-shelf algorithms for FM image analysis are additionally limited. Automated solutions must be applied to data they were explicitly designed to an-

alyze, and their generalizability usually compromises accuracy. For example, automated neural tracing solutions use local patterns that are common features among all FM images of neurons to differentiate dendrites from noise. However, because these algorithms ignore global information about the neuron's morphology, the algorithm can erroneously segment dendrites in the background of the image in addition to the desired dendrites [3]. Any significant voxel noise can further reduce the accuracy of analysis, as it can be hard to reconstruct obfuscated dendrites only with local information.

Even if models are developed for specific tasks, applying them to real-world data often leads to failure cases due to subtle differences in image content and quality between training and testing data. It is therefore common practice to use large public datasets for pre-training networks and then collect a smaller, task-specific dataset for fine-tuning [74, 104, 224, 75]. While this approach can be effective for natural scene images, it is less so for FM datasets due to their greater variability [319, 223]. For example, pretraining networks on pyramidal neurons, which are known to have dendrites spanning pyramid-like volumes, would learn global priors that would not be useful when analyzing other neuron types like horizontal cells, which stratify laterally. Correcting these biases would then require significantly larger fine-tuning datasets.

Fine-tuning on small FM image datasets also has practical challenges. Images for fine-tuning must be annotated, but obtaining FM images with sufficient quality and clarity for ground truths can be difficult or impossible due to experimental limitations. Most importantly, the collection of these datasets can render most applications of automation in experimental neuroscience redundant. If a fine-tuning dataset contains annotated data of sufficient quantity, then the fine-tuning dataset alone can be used for the analysis of scientific experiments to derive the statistical power needed to disprove an experiment's null hypothesis.

Synthetic datasets can be a viable alternative to real datasets, as they offer control over how the images are created and are able to produce an infinite number of training images [247, 262, 255, 279, 221, 260, 270]. An established approach for generating synthetic FM images of biological content is to use manually-designed shapes, features, and noise distributions [247, 262, 255]. For FM images of neurons, neural skeletons or volumetric traces are typically convolved with manually-designed point spread functions, and varying levels of Poisson noise are injected into the 3D stack [279, 221, 260]. Determining the optimal feature set to model FM imaging statistics, however, is not straightforward. While the optical parameters used to acquire images

are recorded during experimentation, there are always differences between theoretical and experimental optical imaging statistics due to mirror or lens misalignment, accumulation of impurities, and varying light refraction of imaged samples [48, 94]. Estimating these statistics experimentally with fluorescent beads is also problematic as it requires additional equipment and is highly susceptible to noise [202]. Even in the ideal case of being able to accurately model local image statistics, more sophisticated techniques are needed to generate structured noise in FM images often seen in the form of auto-fluorescence of nearby cells or unwanted labeled morphologies.

We overcome these challenges by learning local and global FM imaging characteristics directly from a few unlabeled images acquired from an experiment and then stylizing meshes of previously reconstructed neurons (Fig. 6.1). This way, large numbers of representative training data can be generated with "gold standard" ground truths from the images already acquired for a given experiment, and thereby minimizing the amount of additional work required by experimentalists and the costs of experiments. In addition, the ground truth reconstructions provide nanoscale resolution to increase the precision of local FM image analysis.

Unlike other microscopy image generators [16], our unpaired mesh-to-image translation method eliminates the need for any manual annotation as neural meshes can be sourced from rapidly growing, publicly available datasets despite not having corresponding FM images [107]. Neural meshes provide the added benefit of representing the volume of neurons through nodes and edges which can be readily manipulated to substantially increase the variety of synthetic images produced. Furthermore, we explicitly model FM images at differing laser powers and frame averaging, mimicking the parameter space encountered during actual FM experiments. Our model, as a result, effectively operates as a synthetic microscope able to acquire realistic FM images of neurons with various structured noise and optical configurations (Fig. 6.1).

The primary contributions of this work are an unpaired training methodology for generating large synthetic 3D fluorescent images of neurons by successively stylizing 2D slices of voxelized meshes of neurons with 3D context. This is achieved by the introduction of a discriminator with trainable Gram matrix operations to learn stochastic textures at multiple scales. All code is made publicly available at `https://github.com/MihaelCudic/BioSPADE.git`.

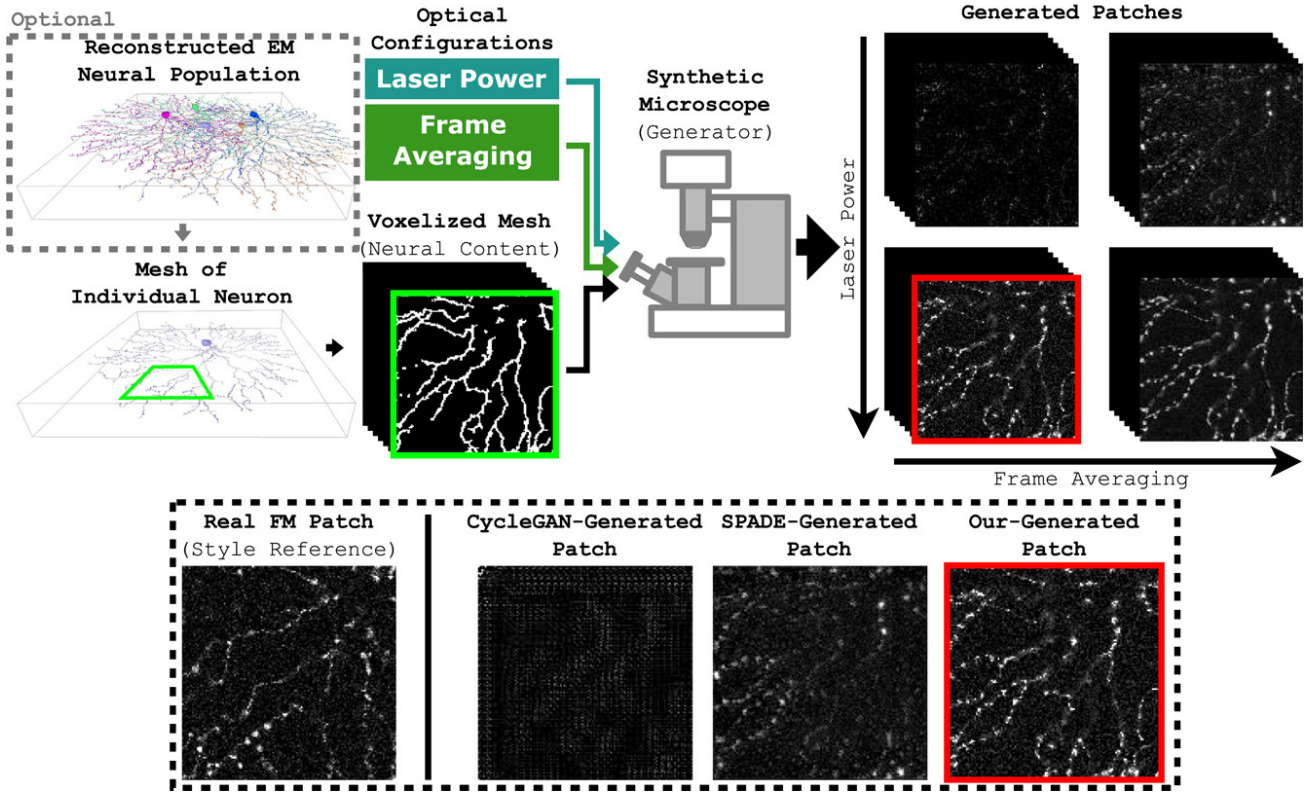


Figure 4.1: An overview of our proposed pipeline and a visual comparison of generated synthetic fluorescence microscopy (FM) patches. We start with a neural mesh that can be sourced from publicly available neural population reconstructions of Electron Microscopy (EM) blocks. Meshes are then voxelized and inputted into our generator. Using real FM stacks as a style reference, the generator then learns to stylize the content of the voxelized mesh to produce realistic synthetic FM images of neurons. Note that our generated patch more realistically models background noise and stochastic texture along the neuron's dendrites compared to a standard unpaired image translation architecture (CycleGAN) and a state-of-the-art image translation architecture (SPADE). Meshes in this figure were provided by [107].

4.2 Related Works

GANs ([96]) can be used to learn both the imaging properties and structured noise and therefore generate fluorescent images. This is accomplished by having a discriminator D that learns to discern real from fake images and a generator G that attempts to generate fake images that fool D (see Section 2.5.2 for a more in-depth overview).

Deep convolutional GANs, for example, have generated a variety of medical and biological images [112, 198, 95]. However, they continue to falter when generating images that vary considerably at local and global scales [21, 19]. Incorporating information from multiple scales has long been a challenge in computer vision, making it difficult for GANs to converge to optimal solutions. 3D fluorescence images of neurons pose a particularly difficult challenge due to their intricate geometric patterns spread across large 3D volumes as spatial compression of these images leads to a loss of substantial semantic information (see Fig. 6.1 and 4.3).

An approach to capture information at different scales is to implement a multi-objective discrimination task [71, 284, 184, 191]. To be considered computationally realistic, generated and real images must occupy similar regions in any number of computed spaces. This can be achieved through multiple discriminative outputs in a single architecture or by using multiple discriminator architectures. The patchGAN discriminator, for instance, is a single discriminator that outputs a discriminatory evaluation for patches across an entire image [120]. This allows the discriminator to focus on semi-global details and forces the generator to output realistic sections across the entire image. Multiple discriminatory computations additionally increase the rigor of realism by compartmentalizing the discrimination task. Now, different architectures can focus on unique aspects of the image making the standard for realism much higher. Wang et al. and Durugkar et al. utilized this idea by employing multiple discriminators to learn coarse and fine image textures [71, 284], compartmentalizing local and global discrimination tasks. Projecting images into different spaces was also explored by Mok et al. and Neyshabur et al. [184, 191].

Another effective approach is for GANs to alternatively perform image-to-image translation by conditioning G on image x so that G applies features on top of the underlying structure of x. This negates the need for G to learn global structures and G can instead model local information. Image-to-image translation was popularized by the pix2pix network and has excelled at denoising and cross-modality domain adaptation for biomedical images [120, 310, 73]. Semantic maps, such as ground truths for segmentation, can condition GANs to learn realistic images with known semantics

as well [250]. Such an approach has frequently been used for geometrically structured images like retinal fundus images [136]. This makes it easier on the generator as it no longer needs to invent the main content in the image as that is already supplied. All that is required is to texturize the main content and background structures, thereby simplifying training.

If not careful, G can ignore x and lead to mode collapse. The pix2pix network overcomes this problem by also conditioning D on $G(x)$ so that the input and generated images are forced to have the same structure. This, however, requires paired image datasets which are often infeasible or expensive to attain in the biomedical field. Another approach is to use the cycleGAN framework to compute the cycle consistency and ensure bijective mapping [317]. Although this framework allows for learning on unpaired images, additional architectures are required making it impractical for large 3D biomedical imaging. The differences between unpaired and paired translations are focused on in Chapter 2.

Style transfer, which looks to stylize the content of an image with local image statistics similar to that of another image, has also been instrumental in generating synthetic biomedical images ([9, 122, 50, 111]). Gatys et al. first demonstrated the ability to mix the style - i.e. texture, contrast, color scheme, etc. - of one image with the content - i.e. a bridge, house, dog, etc. - of another image, now known as style transfer [89]. The main contribution of this work is that the style of an image can be represented by passing an image through a pre-trained VGG-net and computing the Gram Matrix at a specific layer. By matching the Gram matrices of generated and styled images, the desired style can be learned. This idea was extended to standard image translation networks by incorporating style loss as an additional constraint on G [128].

Current approaches of style transfer, however, typically use a pre-trained VGGNet which limits the range of possible style representations to be learned ([89, 128, 284, 204]). For one, VGGNet only looks at 2D images and does not encode any 3D styles. Secondly, VGGNet was trained on natural images and therefore introduces a bias toward certain styles seen in nature. Instead of a pre-trained network, we propose integrating Gram matrix operations directly into a discriminator so that styles are optimized for the domain at hand.

4.3 Methodology

4.3.1 Voxelized Mesh

We start with a mesh of a previously reconstructed neuron that we wish to represent as an FM stack. Because this mesh is represented by a set of polygons connected by common edges and vertices, it can be easily transformed to take on varying configurations while maintaining the underlying morphological structures unique to neuronal cell type. Changes in configurations are imperative as it augments the mesh dataset which help build generalizable solutions. One way to transform the neural mesh is through affine transformations, including rotation, sheering, flip, and translation, which can be performed by multiplying the set of vertices with a transformation matrix as shown below:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.1}$$

where $[X, Y, Z, 1]$ denote the original set of vertices belonging to the mesh, $[X', Y', Z', 1]$ denote the set of transformed vertices, a_{ij} denote the elements of a 3×3 rotation matrix and t_x, t_y, t_z denote the translation along the x, y, and z axes. More sophisticated nonlinear transformations or manipulations of individual segments of the neural mesh are also feasible; however, only affine transformations were used throughout this thesis for simplicity.

The mesh is then voxelized to create a binarized 3D matrix $M \in R^{H \times L \times W}$ where H, L, and W are the height, length, and width of the voxelized mesh. More specifically, individual voxels within M that contain any vertices are designated as 1, while all other voxels are set to 0. The pitch, or voxel size, determines the resolution of the voxelized mesh, as a smaller pitch would increase the size of M. This value was set according to the resolution of the acquired real FM stacks, ensuring that the geometries and 3D scales of neurons are consistent across voxel and real domains.

4.3.2 Slice-to-Slice Translation

FM stacks contain consecutive 2D slices of a specimen at differing focal planes. The distance between each focal plane is constant, but the number of slices acquired in each stack varies depending on the dimensions of the imaged region. We mimic this acquisition process by successively generating 2D slices of neurons that then

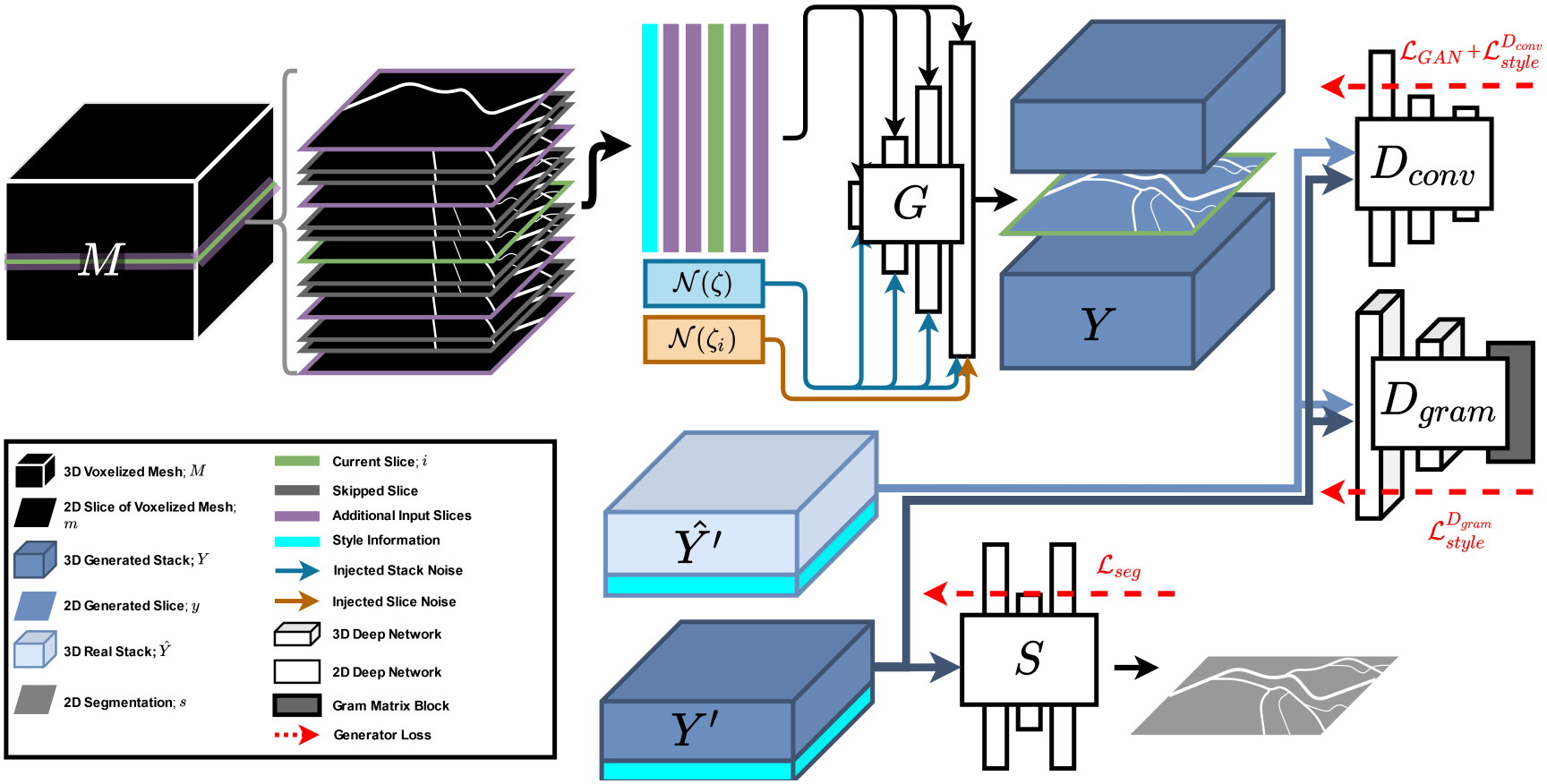


Figure 4.2: An overview of our architecture. A 2D slice m_i from a 3D voxelized mesh $\mathbf{M}$ is input to a generator G along with noise and neighboring slices above and m_i. By using G to map $m_i \rightarrow y_i$ for all slices in $\mathbf{M}$, we generate a synthetic 3D image $\mathbf{Y}$. The discriminator D then samples slices from $\mathbf{Y}$ and real 3D image$\hat{\mathbf{Y}}$ (denoted by $\mathbf{Y}'$ and $\hat{\mathbf{Y}}'$ respectively). A Gram matrix operation is additionally added to D_{gram} for style encoding. The generator loss is then computed as the summation of the GAN loss ($\mathcal{L}_{GAN}$), the D_{conv} style loss ($\mathcal{L}_{style}^{D_{conv}}$), the D_{gram} style loss ($\mathcal{L}_{style}^{D_{gram}}$), and segmentation loss ($\mathcal{L}_{seg}$).

compose an entire 3D image stack. As a result, our generator G is comprised of only 2D convolutions instead of 3D convolutions, taking advantage of the redundancy of information across the z-axis and considerably reducing the number of parameters needing to be trained. This memory-efficient architecture also helps prevent mode collapse and preserves thin structures seen in FM images which are often missed by 3D convolutions. Numerous factors additionally affect the quality of the acquired slice, but for the sake of simplicity, we focus on 3 variables: laser power p, frames averaged f, and z-depth of the focal plane z. Laser power is known to increase the intensity of both the signal and noise in an image, as more energy is used to excite fluorescent protein molecules. The number of frames averaged also affects the signal-to-noise ratio of an image, with more frames averaged leading to a reduction in voxel noise. Lastly, z-depth is crucial, as different types of morphological background noise appear at various depths within the tissue. For example, an image may contain extraneous somas or dendrites if the focal plane is positioned at the somatic or dendritic layer of the tissue respectively.

We condition G on both the 2D slice m_i of our voxelized mesh and the set s_i of styles $\{p, f, z_i\}$, such that G at slice i takes on the mapping $G_i : m_i, s_i \rightarrow y_i$. By successively generating y_i, we are able to then generate a synthetic volume Y. Patterns that appear in y_i, however, may depend on the presence of the neuron in neighboring slices. To provide more context to our generator, we add n_G slices spaced at intervals ΔG_z above and below m_i as extra input channels. These additional slices also condition G so that 3D styles can be modeled while still using 2D convolutions. Various types of noise are added to the generator as discussed below. The generator learns the mapping $G_i : \{x_i, s_i, \zeta, \zeta_i\} \rightarrow y_i$, where $x_i = m_{i,i\pm\Delta G_z,\ldots,i\pm n_p\times\Delta G_z}$ and where ζ and ζ_i represent injected stack and slice noise respectively. For simplicity, let $G(M, S, \zeta)$ also designate Y.

Furthermore, two discriminators are used to encode 3D FM image statistics. One is a fully convolutional discriminator (D^{conv}) and the other is a Gramian-based discriminator outlined below (D^{gram}). Both discriminators are presented with a downsampled 3D volume with n_D slices sampled at intervals of ΔD_z.

The discriminator losses $\mathcal{L}_{D^{conv}}$ and $\mathcal{L}_{D^{gram}}$ are then computed through a Hinge loss as outlined in [158] and summarized below:

$$\mathcal{L}_{GAN}(D) = \frac{1}{2}\left(\max(0, 1 - D(\hat{Y})) + \max(0, 1 + D(Y))\right) \tag{4.2}$$

where $\hat{Y}$ is the downsampled real FM image and the predicted label of $\hat{Y}$ or Y is between -1 and 1, with -1 denoting fake and 1 denoting real.

Since our algorithm is intended to be used on unpaired image datasets, steps must be taken to ensure that the semantic content of M is preserved. We therefore utilized a segmentation loss $\mathcal{L}_{seg}$ by training a subsequent network S to reconstruct M from Y. As a way to preserve memory, S is again a 2D CNN. A complete summary of our methodology can be found in Fig. 6.2. Details of all architectures are described in the following sections.

4.3.3 Generator Architecture

The architecture of G is based on SPADE, a state-of-the-art image translation architecture [204]. However, adjustments were made to the generator so that it was better suited for FM images of neurons. The details of G are as follows.

4.3.3.1 Topology

Most image translation generators are based on an encoder-decoder framework. In the encoder pathway, a high-resolution image is first fed to G and passed through several convolutional and downsampling layers. The downsampling layers create a spatial bottleneck that forces the learning of compressed, high-level features while also saving computational memory. These compressed features are then again passed through convolutional and upsampling layers to output the generated image.

If the input into G is a semantic map rather than an image, the encoder pathway then becomes redundant as high-level features are already present in the semantic map. Therefore, a more efficient G consists of only a decoder with varying scales of the semantic map injected throughout the architecture as seen in [204]. We use this generator topology as our M can be viewed as a semantic map outlining the differences between the background and neural signal.

4.3.3.2 Normalization

While most GANs use batch normalization (Eq. 2.24) throughout the generator, this normalization technique struggles to generate realistic textures of uniform space. This is because the same normalizing function is applied to all image content. Building upon batch normalization, Park et al. [204] introduced a Spatially-Adaptive Denormalization block (SPADE) which allows for realistic stylization of uniform empty space like that seen in our voxelized meshes. This normalization block amends Eq. 2.24 and can be summarized as follows:

$$\mathcal{F}_c^{\texttt{spade}} = \gamma_c(\mathbf{x}) \odot \frac{\mathcal{F}_c - \mu_c}{\sigma_c} + \beta_c(\mathbf{x}), \tag{4.3}$$

where $\mathcal{F}_c^{\texttt{spade}}$ is the CNNs cth feature map after normalization, $\gamma_c(\cdot)$ and $\beta_c(\cdot)$ denote the learned scale and bias parameters generated by a separate convolution layer respectively, and $\odot$ denotes element-wise multiplication. Note that the key difference between SPADE and batch normalization is that the scale and bias terms are tensors conditioned on $\mathbf{x}$ instead of a static term. As a result, dendrites are treated differently than background noise during normalization.

While this normalization is useful when generating a distribution of unique features, it is particularly sensitive to the content and size of FM image patches as it tries to force every feature map activation into a normal distribution despite the content in the batches. To illustrate this, consider the scenario where the size of M is expanded to include more background voxels. This would significantly change the μ_i^c and σ_i^c values and thereby the feature map activations, even though the content of the image remains unchanged. Similarly, if a batch of FM patches contains few dendrites while other batches have many, the normalization would again force the feature map activations to appear similarly despite different content. To establish a more robust normalization method for our types of images, we modify SPADE so that the feature maps are normalized by the power of the feature map and not the variance, which is less sensitive to differing image content as seen here:

$$\mathcal{F}_c^{\texttt{spade}} = \gamma_c(\mathbf{x}) \odot \frac{\mathcal{F}_c}{\|\mathcal{F}_c\|_2} + \beta_c(\mathbf{x}), \tag{4.4}$$

This approach makes the normalization less reliant on the ratio between the signal and background noise, thus stabilizing training. Ultimately, we employ a combination of both batch normalization and pixel normalization in order to take advantage of both of their strong points.

4.3.3.3 Noise and Acquisition Layer

Noise is a defining property of biological images and needs to be explicitly modeled in generator G. As highlighted throughout this thesis, there are two primary types of noise in FM images: structured and voxel noise. Both need to be modeled by our GAN to accurately generate realistic FM images. For structured noise spanning multiple slices, Gaussian noise was added as an extra feature channel at varying spatial scales and kept constant for each slice of a 3D image. The Gaussian noise channel serves as a latent space representation of varying structured noise that can

be injected into the image. If new Gaussian noise were to be sampled, the image would exhibit a completely different set of structured noise. The GAN should learn the mapping between the noise channels and structured noise. Finally, by keeping the noise constant for each slice, we ensure that the injected structured noise remains consistent throughout the entire FM stack. Otherwise, each new slice would have a completely different injected noise distribution.

Because a GAN uses convolutions to output its final layer, it struggles to introduce voxel noise where noise is independently sampled in each voxel as the convolutions spread out information. So instead, similar to noise-robust GANs [130, 29], we introduce an `acquisition` layer as the output layer of G in order to inject voxel noise directly into each generated slice. However, our `acquisition` layer takes extra steps to mimic image acquisition in fluorescence microscopes. More specifically, the `acquisition` layer receives the voxel offset value $\mu_c \in R^{L \times W}$, a set of n feature maps $\zeta_c \in R^{n \times L \times W}$ containing the probability distribution parameters for $P(x)$ where $x \in R^{L \times W}$, a channel offset scalar β_c, and the number of frames averaged F. We then sample $\mathcal{X}^c \sim P(x)$ F times such that $\mathcal{X}^c \in R^{F \times L \times W}$ and obtain the acquired image by:

$$\mathcal{F}_c^{\texttt{acq}} = \frac{1}{F} \sum_{f=1}^{F} clip(\mu_c + \mathcal{X}_f^c + \beta, 0, 1) \tag{4.5}$$

where $clip(\cdot, 0, 1)$ denotes the voxel-wise clipping of values between 0 and 1. Ideally, we would be able to introduce Poisson noise directly into this equation. However, Poisson noise is a discrete noise distribution, while the images and the noise we can introduce are decimal values. Moreover, Poisson noise cannot be sampled in a differentiable manner, which means that the machine learning architecture cannot learn to sample the Poisson noise. Instead, we can only estimate the noise that would be introduced from Poisson noise. To model Gaussian noise, we simply set $\zeta_c \in R^{1 \times L \times W}$, denoted as σ_c, and sample voxel noise $\mathcal{X}^c \sim \sigma_c \mathcal{N}(0, 1)^{F \times L \times W}$. By averaging frames within the layer, we minimize the number of pixel distributions that the GAN needs to learn, as it instead uses the pixel distribution of a single frame as a basis for the remaining frames that are averaged.

However, non-Gaussian distributions are required to model real FM noise with Poisson noise being the most common type [312]. We, therefore, decided to explicitly model a Fréchet probability function due to its flexibility in producing a wide range of tailed distributions. This is accomplished by having ζ_c denote the set of feature maps $\{\alpha_c, s_c\}$ where α_c is the inverse of the Fréchet shape parameter and s_c the Fréchet

scale parameter [85]. We then sample voxel noise such that $\mathcal{X}^c \sim s_c(-1 \log u)^{\alpha_c}$ where $u \sim U(0,1)^{F \times L \times W}$.

The implementation of our acquisition layer additionally allows us to have control over the final image output. Not only do we have the spatially dependent noise characteristics which can be used to artificially dampen or amplify noise, but we can also estimate a denoised version of the image as:

$$\mathcal{F}_c^{\hat{\texttt{acq}}} = clip\Big(\mu_c + \beta + \Big(\frac{1}{\alpha_c + 1}\Big)^{\alpha_c}, 0, 1\Big) \tag{4.6}$$

4.3.4 Discriminator Architecture

Like the generator, most of the discriminator architecture follows that of SPADE. Here we discuss the additions made so that FM styles could be represented and learned more accurately.

4.3.4.1 Discriminator with Spatial Standard Deviation Layer

Learning the image characteristics of FM stacks is a complex task as local features are stochastic, but greatly depend on global context. Convolutions alone are ill-suited for this task as they have a limited field of view and typically reduce high-frequency information after every layer. We address this concern by first introducing a `Spatial_stddev` layer before the second to last layer in D^{conv}. This layer, which is a derivative of `Minibatch_stddev` as outlined in [131], computes the average standard deviation of activations across all spatial locations for each feature map, replicates the values, and concatenates those values with the original feature maps. For the 2D case, the output of this layer can be summarized by a tuple with elements:

$$\mathcal{F}_{l,w}^{\texttt{spatial,c}} = \Big(\mathcal{F}_{l,w}^{c}, \sqrt{\frac{1}{L * W} \sum_{l \in L, w \in W} (\mathcal{F}_{l,w}^{c} - \overline{\mathcal{F}}_{L,W}^{c})^2 + \epsilon}\Big) \tag{4.7}$$

such that $\mathcal{F}^{\texttt{spatial,c}} \in R^{2xLxW}$, the cth feature map $\mathcal{F}^c \in R^{1xLxW}$, $(\cdot,\cdot)$ denotes the concatenation of feature maps, ϵ is 10^{-6}, and L and W are the length and width of the feature map respectively.

This layer serves two primary purposes. Firstly, the variance of each feature map allows D^{conv} to encode textures while maintaining spatial information. Secondly, computing the variance ensures that generated structures are spatially variable, helping prevent mode collapse.

4.3.4.2 Gramian-based Discriminator

Finally, we want to be able to compute the style of an image in a way that is learned for the specific dataset. In order to do this, we do not want to use pre-trained networks.

So, let us consider $D_J^{gram}(y, s)$ to be the feature maps of the Jth layer of D^{gram} when processing input y with style encoding s. Although our methodology can be naturally extended to 3D images, for simplicity we assume y to be a 2D image and $D_J^{gram}(y, s)$ to be of shape $C_J \times L_J \times W_J$ where C is the number of output channels. The Gram matrix $\phi_J^{D_J}(y, s)$ is then defined by a $C_J \times C_J$ symmetric matrix with elements:

$$\phi_J^D(y)_{c,c'} = \frac{1}{C_J L_J W_J} \sum_{l=1}^{L} \sum_{w=1}^{W} D_J^{gram}(y, s)_{l,w,c} D_J^{gram}(y, s)_{l,w,c'} \tag{4.8}$$

The information contained in $\phi_J^D(y, s)$ is further reduced to a $(1 + C_j)C_j/2$ vector $\psi_J^D(y, s)$ by flattening the lower triangle of the matrix. Meaningful style representations can now be learned by feeding $\psi_J^D(y, s)$ through a fully-connected network trained with an adversarial loss. Therefore, D^{gram} is summarized by initial convolutional layers, computation of $\psi_J^D(y, s)$, and final fully-connected layers.

Now that we have a network learning to encode styles relevant to the task at hand, we compute the style loss at various layers throughout the network:

$$\mathcal{L}_{style}^{D,j}(\hat{y}, y, s) = \sum_{j=1}^{J} \mathbb{E}_{\hat{y},y} \|(\phi_j^D(\hat{y}, s) - \phi_j^D(y, s))\|_1 \tag{4.9}$$

4.3.5 Objective

After combining all the loss terms, we obtain the following multi-objective equation:

$$G^* = \arg \min_G \max_D \lambda_{seg} \mathcal{L}_{seg}(G) + \sum_{k=1}^{K} \mathcal{L}_{GAN}(G, D_{conv}^k) + \lambda_{style}^{D_{conv}} \sum_{j \in J} \mathcal{L}_{style}^{D_{conv},j}(G) + \lambda_{style}^{D_{gram}} \sum_{j \in J} \mathcal{L}_{style}^{D_{gram},j}(G) \tag{4.10}$$

where λ_{seg} is the weight for the segmentation loss, $\lambda_{style}^{D_{conv}}$ and $\lambda_{style}^{D_{gram}}$ are the weights for the style loss computed from D_{conv} and D_{gram} respectively, and K is the number of D_{conv}s used to compute the multi-scale GAN loss.

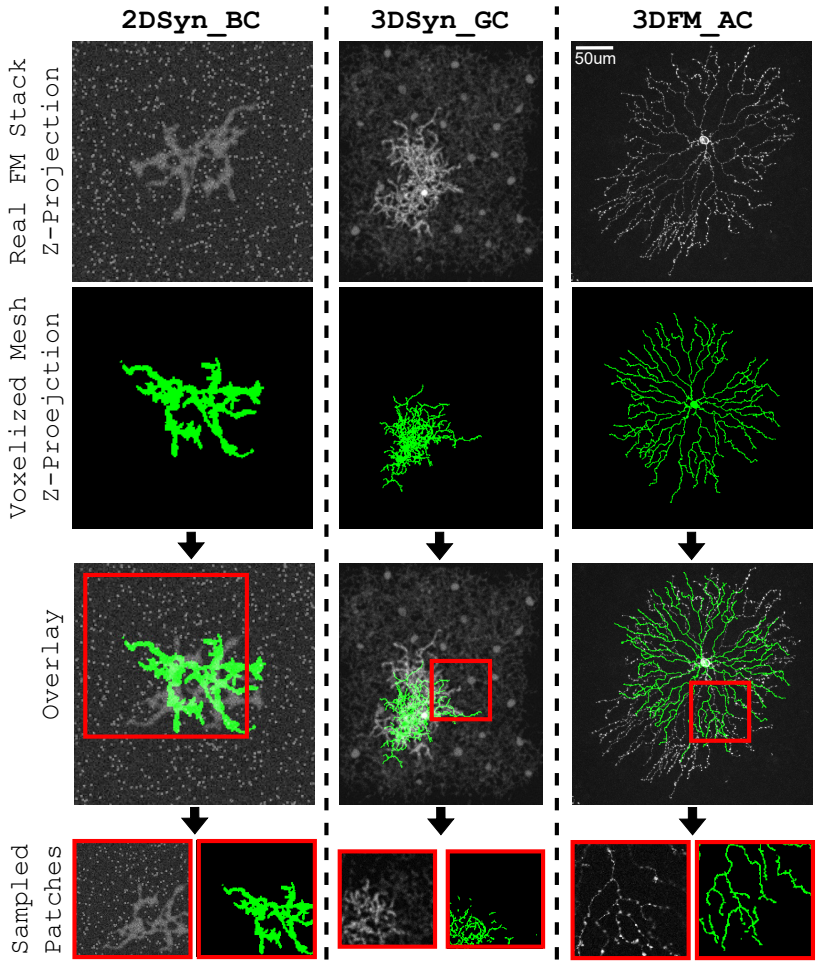


Figure 4.3: Overview of our mesh and FM patch sampling protocol for three datasets (columns). We first overlay a real stack (shown as the maximum z-projection in grey) with a voxelized mesh of a previously reconstructed neuron (shown in green) so that features are spatially correlated. Patches are then sampled from the same spatial regions in the real and voxelized mesh domains, ensuring that sampled patches contain similar features. Datasets include a synthetic 2D Bipolar Cell dataset (2DSyn_BC), a synthetic 3D Ganglion Cell dataset (3DSyn_GC), and a newly acquired 3D FM Amacrine Cell dataset (3DFM_AC). Note: the fourth 3D FM Bipolar Cell Population dataset (3DFM_BCPop) seen in Fig. 3.5 was excluded from this figure as binarized semantic maps were used as the input into our generator instead of meshes in order to directly compare unpaired and paired training regiments.

4.3.6 Random Sampling of Patches

Now that we have the 3D voxelized mesh M which contains the neural content of the desired generated images, we randomly sample a real FM stack denoted as $\hat{Y}$ to serve as our style reference. While image translation can theoretically be performed using a set of Ms and $\hat{Y}$s, GPU memory constraints make this impossible even when using

a batch size of 1. Instead of the entire image, patches from both M and $\hat{Y}$ need to be sampled during training. This presents additional challenges when training unpaired image translation architectures as there is no guarantee that sampled patches across domains have similar features or weak spatial correspondence. While content similarity across domains is not a theoretical requirement for unpaired image translation, it is in practice. When first randomly sampling patches independently from both voxelized mesh and FM image domains, we encountered failure cases where the architecture was unable to generate anything resembling a real FM image. Failure cases were additionally more likely when smaller batch sizes were used.

To illustrate this issue, let us examine Eqs. 4.8 and 4.9 more closely. While the Gramian Matrix computation attempts to decouple the image style from the image content by aggregating features independent of spatial locations, features used to compute the Gram Matrix are still conditioned on the image content as $D_J^{gram}(., s)$ is the feature map of the final convolutional layer in D^{gram}. Consequentially, $\psi_J^D(., s)$ will result in different embeddings if computed for an image of somas compared to an image of dendrites despite both images having the same FM imaging characteristics. Given that Eq. 4.9 looks to minimize the Gram Matrices between both generated and real FM images, the generator could conceivably attempt to fabricate unwanted content in addition to learning the style of real FM images. This effect, however, can be mitigated if both $\psi_J^D(y, s)$ and $\psi_J^D(\hat{y}, s)$ have similar neural content as the latent embedding associated with the image content will cancel each other out, leaving only the desired style difference.

To help stabilize the learning procedure, we do not randomly sample patches across domains. We instead randomly sample the same spatial regions across domains by first overlaying the neuron in M with the neuron in the real stack as shown in Fig. 4.3. Depending on the dataset, this was achieved in several ways which will be further outlined in Section 4.4.

4.4 Implementation

4.4.1 Data

For the evaluation of our unpaired mesh-to-image translation model, we trained our generator on 4 different FM image datasets: 2DSyn_BC, 3DSyn_GC, 3DFM_AC, and 3D_BCPop. These datasets were chosen as they each exhibited a wide range of textures that the generator had to model. Specifics about each dataset are described in more detail in Chapter 3. Additional manipulations to the data and tailored

sampling protocols were required so that we could train our generator on unpaired data. These manipulations are described as follows:

4.4.1.1 2DSyn_BC

To create an unpaired voxel dataset, the same 41 meshes used for the synthetic images were flattened and flipped along the x-axis. Flipping was required to ensure that there was no one-to-one mapping of geometries, but that features of the sampled real image had similar geometries to that of the voxelized mesh. 128x128 patches were then continuously sampled during training.

4.4.1.2 3DSyn_GC

Similar to the 2DSyn_BC, an unpaired voxel dataset was generated by flipping all neurons belonging to the Ganglion cell population along the x-axis. To ensure a weak correspondence between the geometries, unpaired voxelized neurons were randomly sampled based on their overlap with the neuron present in the original image as illustrated in Fig. 4.3. The random selection was additionally weighted by the amount of overlap between neurons such that the voxelized neuron with the greatest overlap had the highest probability of being chosen. During the training process, 4x128x128 patches were further sampled to augment the dataset.

4.4.1.3 3DFM_AC

To condition the GAN, 19 SAC meshes were voxelized with a pitch of 645 provided by [34]. To overlay the voxelized mesh with the image, we employ the same procedure used for aligning images of SACs, as described in Section 3.2.3 and shown in Fig. 4.3. However, instead of finding the maximum cross-correlation xy-correlation from maximum-z-projections, axes were completely collapsed, ensuring that cross-correlation overlap was maximized for each dimension separately. This was done to improve computational efficiency. Patches of 4x128x128 were sampled after the meshes were overlayed with a real stack for training.

4.4.1.4 3D_BCPop

When training was with unpaired data, patches of 4x128x128 at the same retinal depth were randomly sampled from both the FM stack and the binarized semantic map such that there was no correspondence between the two. This was done as the entire xy-cross section of the stack was 2048x2048 and the probability that sampled

Discriminator	Norm.	Act.	Output Shape
Input Image	-	-	$1 \times 4 \times 128 \times 128$
Conv3 $\times$ 3 $\times$ 3	instance	ReLU	$32 \times 4 \times 128 \times 128$
Conv4 $\times$ 4 $\times$ 4	batch	ReLU	$32 \times 4 \times 128 \times 128$
Conv3 $\times$ 3 $\times$ 3	batch	ReLU	$64 \times 4 \times 64 \times 64$
Conv4 $\times$ 4 $\times$ 4	batch	ReLU	$64 \times 4 \times 64 \times 64$
Gram	-	-	1×2112
Dense	batch	ReLU	1×1056
Dense	-	-	1×1

Table 4.1: Architecture details for our Gramian-based discriminator.

real and voxelized mesh patches containing the same pixel was trivial. Other forms of manipulation of the unpaired voxelized dataset, like rotation or flipping, would likely contain irrelevant style information that the GAN would use for discrimination.

4.4.2 Architecture

Some minor adjustments to SPADE's generator were required to train on our data. For one, the network was made smaller by eliminating one of the upsampling layers and a base of 16 filters was used rather than 128. Additionally, m_i was concatenated to the second to last feature map of G to ensure that the thin processes present in m_i were not lost. We found that batch normalization was helpful to generate realistic background structured noise, while pixel normalization was better suited to learning pixel noise distribution. As a result, a combined approach was used with batch normalization implemented in the first 3 layers and pixel normalization thereafter. The final layer was our `acquisition` layer. Gaussian noise was explicitly modeled for our synthetic datasets and Fréchet noise for our real FM dataset. Moreover, 2, 3, and 4 neighboring slices of the voxelized mesh were concatenated with m_i and input to the generator for the 3DSyn_GC, 3DFM_AC, and 3DFM_BCPop datasets respectively. Finally, when training on 3DFM_BCpop, our generator had an additional upsampling layer which was required to model some of the larger objects present in the images.

The multi-scale discriminators were also closely modeled after SPADE. The base filters were again downsized from 128 to 32 to shrink the network. Instance normalization was implemented at the first layer to ensure that each feature map was normalized across all styles. Our `Spatial_stddev` layer was also incorporated in the second to last layer of D_{conv}.

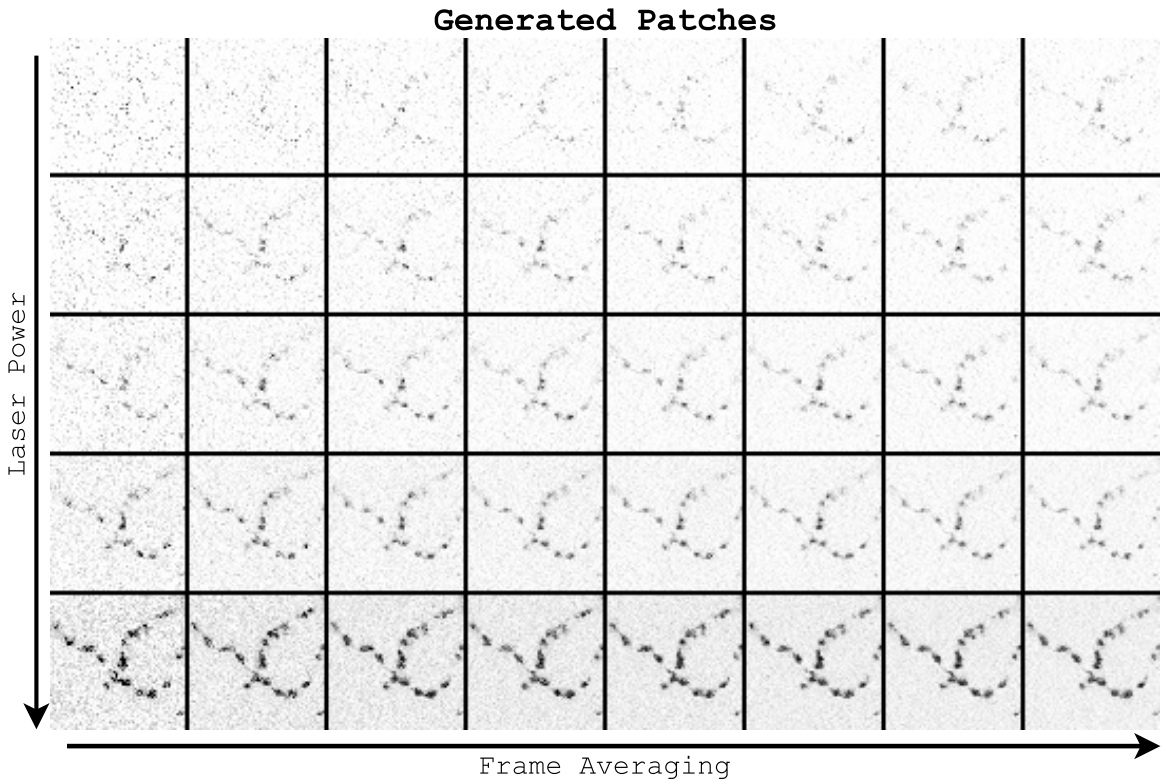


Figure 4.4: 64x64 generated patches (inverted for visual effect) from a GAN trained on 3DFM_AC dataset. Patches sampled illustrate powers and frame averaging. A total of 5 different laser powers and 8 frame averaging combinations were acquired.

An additional 2D U-Net was used as our segmentation network. The segmentation network used instance normalization for the first layer and the remaining layers were batch normalization layers. The U-Net only had one downsampling layer to minimize the network size. A categorical cross-entropy loss was used along with ADAM optimization ($\beta_1 = 0.9, \beta_2 = 0.999$) with a learning rate of 0.01.

Finally, our D_{gram} consisted of a shallow 4-layer CNN, a Gram matrix operation, and 2 fully-connected layers. The topology for the 3D network is shown in Table 1.

Training of G occurred over 200 epochs with an additional 100 epochs of linearly decaying step size. All additional training details are outlined in [204]. All code and implementation details can be accessed at `https://github.com/MihaelCudic/BioSPADE.git`.

4.5 Results

As shown in Figs. 6.1 and 5.1, our architecture generates realistic FM images across a variety of styles and content. It is important to note that each style is unique, suggesting that mode collapse is at least partially averted. Similar to an actual

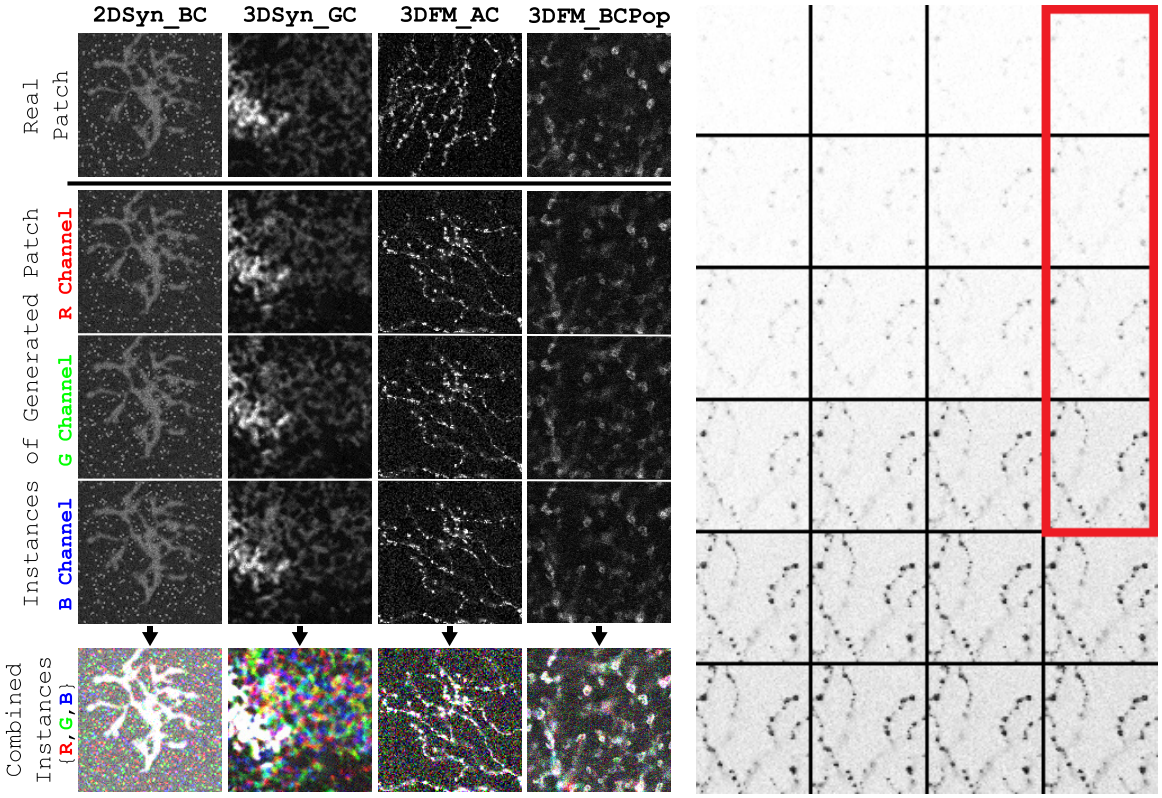


Figure 4.5: Demonstration of our generator's ability to generate a wide variety of images. a) 3 instances of generated patches (middle rows) using the same voxelized mesh and their combined RGB image where every instance corresponds to a different color channel. The combined RGB image shows that our generator learns the spatial distribution of structured noise as each instance is unique. b) Generated patches (inverted for visual effect) with linearly increasing laser powers. Patches outlined in red are generated from laser powers seen during training. The laser powers for the remaining patches are realistically interpolated by the network.

microscope, each new acquisition of a frame preserves the structure of the voxelized mesh while individual voxel noise varies (Fig. 4.5a). Moreover, our model effectively generates differing manifestations of morphological noise as seen with the 3DSyn_GC dataset.

Optical imaging parameters can be additionally adjusted to create novel images. For instance, our generator is able to interpolate novel laser powers like a real microscope despite only training on 5 unique laser powers (Fig. 4.5b). Our `Aquisition` layer also allows us to decouple the signal from the noise so that we can generate images with new noise profiles and images that are theoretically impossible to obtain, such as denoised FM images (Fig. 4.6). While voxel noise is still present in denoised

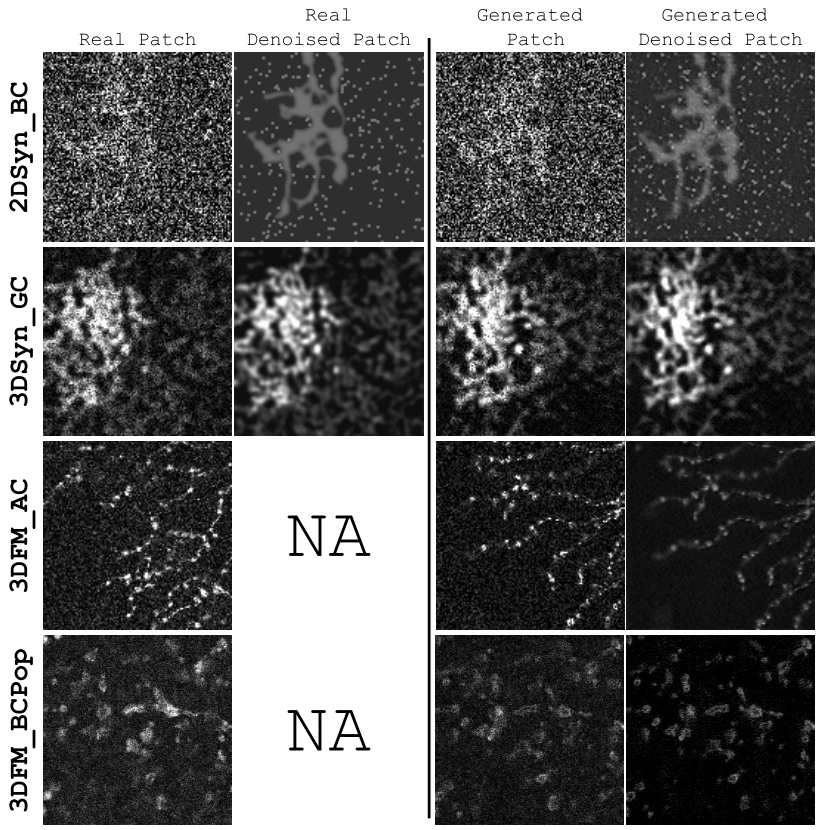


Figure 4.6: Demonstration of our algorithm's ability to produce images with novel noise characteristics for all three datasets (rows). Shown are paired noisy and denoised patches from real and generated data. NOTE: there exists no real denoised patch as every frame acquired contains noise.

images upon close inspection, it is important to stress that these images are a valuable by-product of our training approach and can provide scientists with higher-quality FM image visualizations.

The use of meshes as the basis of the input to our generator grants unique control over the content of generated images since morphological manipulations can be performed easily on the mesh vertices and edges. Although we only explore simple linear transformations (rotations, mirror, and sheer) shown in Fig. 4.7, non-linear transformations and manipulations to individual branches could be used to increase the content variety in generated images. Most significantly, once the transfer function from the voxelized mesh to FM images is learned, our generator is then able to generate realistic synthetic FM images with arbitrary content. As demonstrated by Fig. 4.8, we are able to produce realistic FM images of novel geometries, multiple overlapping neurons, and larger patch sizes despite never having trained on these types of voxelized meshes. The most notable example can be seen when stylizing concentric

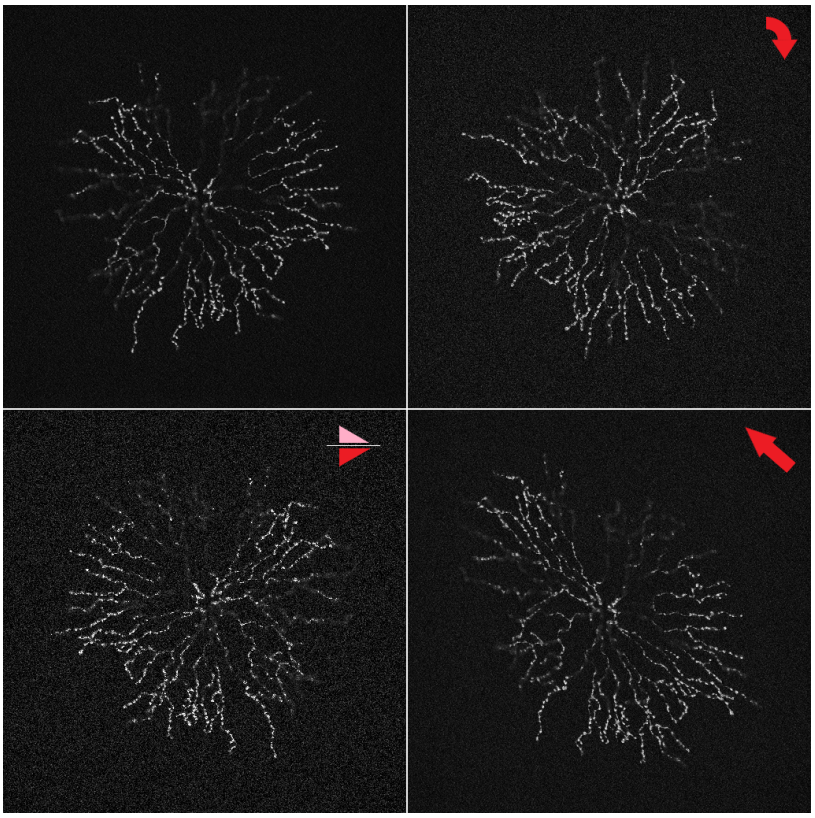


Figure 4.7: 4 generated synthetic slices with each slice displaying differing morphological variations (rotation, mirror, sheer) of the same neural mesh.

squares as if they were imaged with the same microscope used for the 2DSyn_BC dataset.

4.6 Discussion

The conventional paradigm for deep learning requires a significant amount of labeled training data. If unavailable, it is typical to use public or synthetic datasets for pretraining. In this chapter, we propose an alternative pipeline for training deep learning methods with limited data. Instead of acquiring additional training data, we leverage publicly available volumetric reconstructions of neurons to create realistic data.

There are many benefits to this approach. Firstly, this approach requires minimal adjustment to the way neuroscience experiments are conducted. Rather than collecting and labeling data specifically for training purposes, our pipeline enables the direct utilization of unannotated data. As a result, experimental neuroscientists can use data already gathered for a given experiment to generate more similar training

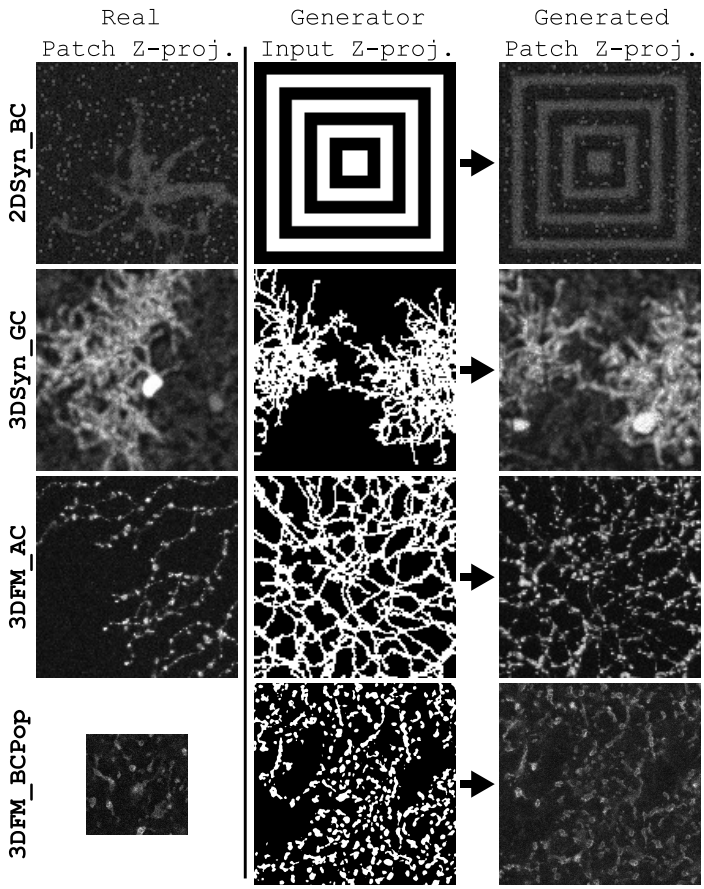


Figure 4.8: Demonstration of generator generalizability. The left column shows Z-projections of real patches sampled as style references for all datasets (rows). In the middle column, panels top to down show a generator input of novel geometries, a voxelized mesh Z-projection of a neuron with an additional adjacent neuron, a voxelized mesh Z-projection of several overlapping neurons, and a binarized semantic map double the size of the real patch sampled during training. Each voxelized mesh was therefore never seen in training. The final row shows the Z-projection of the generated synthetic FM image conditioned on the novel generator inputs.

data. Secondly, volumetric neuron datasets are very detailed and precise representations of neurons. Because they are used to condition synthetic images, they can also be used to create "gold standard" ground truths. This eliminates the need for manual annotation of data, which is prone to variations as even experts cannot agree on the correct ground truths all the time. Volumetric reconstructions can additionally are easily manipulated to create novel configurations.

Most significantly, once our model is fully trained, it can operate similarly to an actual fluorescence microscope and produce realistic FM images of novel biological content with optical configurations never seen before in training. This greatly expands the impact of our pipeline to the neuroscience community as our model can learn

imaging characteristics from a small dataset of easily acquired FM images and then generate realistic synthetic FM training datasets of images that are much harder to acquire. For example, not only do we show that multiple styles can be learned explicitly during training, but the learned spatially-dependent noise parameters in the final layer can also be adjusted after training to account for more styles. This enables us to compute paired noisy and denoised images which do not exist in real-world FM applications and subsequently train a denoising algorithm to improve the quality of FM images acquired during an experiment. Finally, additional neurons, along with other structures, can be inserted into the voxelized mesh while preserving realistic FM imaging characteristics so that more complex image segmentation tasks can be learned. This is especially important when trying to analyze neural network activity as individual dendritic responses need to be associated with their parent neuron and ground truths are challenging to acquire.

A limitation of this approach is that it requires the existence of volumetric reconstructions that are of the same neuron type, brain region, and species as the neurons contained in the acquired FM image dataset. While all neurons imaged and modeled in this paper are from the mouse retina, as more EM reconstruction datasets become publicly available, our pipeline could be applied to any part of the brain.

Chapter 5

Evaluating Realism in FM Images

Contents

5.1 Introduction 88
5.2 Related Works 90
5.2.1 Classification 90
5.2.2 Sample Distribution Distance 91
5.2.3 Feature Distribution Distance 93
5.2.4 Data Utility 94
5.3 Method 94
5.3.1 Data 94
5.3.2 Evaluation 95
5.3.3 Comparisons 99
5.4 Results 101
5.5 Conclusion 104

5.1 Introduction

When evaluating newly designed GANs, a common question asked is: "Is this image real or fake?". Although the question appears straightforward, it is fundamentally flawed. For one, the definition of "real" and "fake" is ambiguous. Is an image of a landscape that is distorted and heavily blurred less "real" than a high-quality image of the same landscape but with blue grass and a green sky? The answer to this question would depend on your definition of "real". If so, how is it possible to assign the subjective quality of "realness" objectively? This philosophical conundrum has no clear answer.

Secondly, determining the "realness" of a single image is impossible without multiple examples of "real" images. Because the initial question makes an explicit distinction between a "real" and "fake" image, we assume that there are relative differences between the two. Therefore, we can reword the question to "Is there anything different in this image from a real image?" which would be meaningless without a reference to "real" images.

While we might not be able to create a universal definition of "real" and "fake" based on the images themselves outright, we can first create useful categories of images by how they are created. For example, we know that some images are acquired from an imaging device, like a digital optical camera, while others can be generated through a computational algorithm, like a GAN. We can then define a "real" image as camera-generated and a "fake" image as GAN-generated. Given a collection of both sets of images, we can then come up with a definition to describe the relative differences between "real" and "fake" images from the images themselves. As an example, "real" images of human faces might tend to have more impurities, like wrinkles or blemishes, than that of "fake" images. More complex relative definitions can be learned by deep learning architectures, which is the function of a GAN discriminator.

Humans, however, can assess an image's realism without needing separate sets of "real" and "fake" images as we can instead use our prior experiences to guide judgments. Throughout interactions with the world, our minds attempt to make sense of our experiences by learning governing rules and properties necessary for our social function and survival. This allows humans to learn that the structure of a human face typically has two eyes, a nose, and a mouth. However, because what we learn must also be relevant to our lives, we filter our information that is irrelevant [90]. For example, while we may use the shape and color of someone's hair to help identify them, we may not pay attention to the individual strands of hair and how they interlock with one another. This is a common feature in visual processing, as humans know to attend to low-frequency information, like face shape, much more than high-frequency, like hair strand width [197, 117, 243]. As a result, human intuition for realism is also biased toward a subset of features. For example, an image of a human face with improbably thick strands of hair or fragmented hair strands might be deemed "fake" by a viewer.

In biomedical images, the human intuition of a "real" image becomes less reliable, as our life-long experience with the natural world may not be a suitable reference. As mentioned in Chapter 3, there are significant differences between natural scene images and biomedical images. While humans can learn to better attend to features unique

to biomedical images, there are inherent limitations. Similar to how pre-training an unrepresentative network biases the network toward irrelevant features, humans will always be biased toward features in natural scene images. Deep learning models trained on x-ray images, for example, can outperform radiologists [224, 165]. This is because deep learning models can learn to be more sensitive to varying shades of grey present in x-ray images, a skill that is less relevant for radiologists in the natural world [144]. Furthermore, noise, which tends to be filtered out by the human visual system [197, 243], can be a dominant feature in biomedical images, especially in FM images during live imaging [26].

Up until now, we have discussed realism through the lens of human opinion. This is largely because the success of GANs is determined by how easily they generate images that fool us into thinking they are real. However, it is important to be aware that this evaluation method, like all of evaluation methods, is biased. Why should human evaluation, even as experts, be seen as the most rigorous of tests when our visual system is not designed to analyze features in biomedical images?

What is important is not necessarily that the image looks real, but rather how well it achieves an objective. In this chapter, we explore various ways to evaluate the realness of FM images of neurons that focus on content, local texture, noise distribution, and utility of the image. While each with its own limitations, together the metrics do a sufficient job of determining the better-performing architecture.

In the previous chapter, we proposed a novel GAN to generate realistic FM images. In this chapter, we will explore the performance of our proposed GAN. We compare our unpaired mesh-to-image architecture with 2 other state-of-the-art image translation architectures on 2 synthetic FM datasets and 2 acquired FM datasets of retinal neurons that were in Chapter 3.

5.2 Related Works

5.2.1 Classification

One method of quantifying image realism is to perform a classification task [193]. The most intuitive way of evaluating image realism is to ask humans directly, as mentioned in the introduction [309]. For this method, humans are usually given a set of randomly sampled real and generated images without labels. They are then tasked with classifying which set the images belong to. A classification accuracy or confusion rate of 50% would imply that real and generated images are indistinguishable. Several humans usually partake in this analysis, as the outcomes can be subjective. If the

images belong to a specific domain, like FM images, then experts in that field are required for the test. It is important to be aware that the more tests an individual partakes in, the more training they have at discerning between real and generated images, which can bias results.

Because human evaluation of GAN performance is time-consuming and labor-intensive, especially when expert input is required, there has been significant effort toward developing computationally-derived metrics. The Classifier Two-Sample Test, for example, aims to determine whether two samples appear to be from the same distribution [257]. A deep learning model can be used as a classifier, but will likely easily distinguish between any two sets of images if not constrained. Instead, [297] used a 1-Nearest Neighbors classifier (1-NN) due to its simplicity. This algorithm works by taking a novel sample, finding the most similar image in the training dataset, and assigning it the same label. The similarity of images is typically determined using a Euclidean distance, also known as the L2 distance. The similarity metric can be useful for natural scene images but is a less effective metric for FM images of neurons. This is because this metric is sensitive to noise as two images sampled of the same content and noise distribution would be considered less similar than a sampled image and the average of samples. More importantly, because this metric only computes the similarity of voxels, any spatial translation of content will drastically change the computed similarity as neuronal content is thin.

5.2.2 Sample Distribution Distance

In addition to the classification of individual samples, we can evaluate the distance between the distribution of samples [96, 292]. Because there exists only a finite number of samples for both real and fake distributions, they must first be estimated. This can be done using Kernel Density Estimation (KDE), also known as Parzen windowing [206]. For KDE, a kernel $K(\cdot)$ is placed at the center of every sample as follows:

$$\hat{f}(\mathbf{x}) = \frac{1}{N\left|\mathbf{H}\right|^{1/2}} \sum_{i=1}^{n} K\left(\mathbf{H}^{-1/2}(\mathbf{x}-\mathbf{x}_i)\right) \tag{5.1}$$

where $\hat{f}(\mathbf{x})$ is the estimated continuous distribution of $\mathbf{x}$, N is the number of samples, and $\mathbf{H}$ is bandwidth of the kernel. In most cases, the kernel can be used as a Gaussian kernel which can be written as:

$$K(\mathbf{u}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{u}^\top \mathbf{\Sigma}^{-1}\mathbf{u}\right) \tag{5.2}$$

where $\mathbf{\Sigma}$ is the covariance matrix of $\mathbf{u}$.

Once the real estimated distribution, denoted as $\hat{P}$, and the fake estimated distribution, denoted as $\hat{Q}$, are derived, then the distance between the distributions can be calculated using a variety of metrics. The most common distance metric is the Kullback-Leibler divergence (D_{KL}) which attempts to calculate the relative entropy between distributions [141]. It can be more formally described as follows:

$$D_{KL}(\hat{P} \parallel \hat{Q}) = \sum_{\mathbf{x}} \hat{P}(\mathbf{x}) \log \frac{\hat{P}(\mathbf{x})}{\hat{Q}(\mathbf{x})} \tag{5.3}$$

Because D_{KL} employs a log function, this distance metric is asymmetric, meaning $D_{KL}(\hat{P} \parallel \hat{Q}) \neq D_{KL}(\hat{Q} \parallel \hat{P})$. Therefore, the symmetric equivalent to D_{KL} is the Jensen-Shannon divergence (JSD) [159] which can be written as:

$$\text{JSD}(P, Q) = \frac{1}{2}D_{KL}(\hat{P} \parallel \hat{Q}) + \frac{1}{2}D_{KL}(\hat{Q} \parallel \hat{P}) \tag{5.4}$$

For KDE to be effective, a substantial number of data must be collected throughout the data space [272]. This means that as the dimensions increase, the amount of data required increases exponentially, thereby often limiting KDE to low-dimensional data.

Instead of estimating the distance between distributions directly from samples, deep learning can be utilized to learn a distance. Similar to a classification task, a neural network can be employed to assign a critic score for both real and generated images. A critic score of 1 indicates the image appears real while a score of -1 indicates the image appears generated [8]. The training of the network then uses a Wasserstein Distance Loss with Lipschitz constraint on the weights to learn the effort it would require for one distribution of data to be transformed into another [8]. In practice, this function is similar to the discriminator of a WGAN (refer to Section 2.5.2 for further details). However, is highly sensitive to the architecture implemented, as shallow networks may struggle to effectively distinguish between real and generated images.

The primary limitation of all distribution distance measures to evaluate GANs is that they do not account for the quality of data generated. The mean data sample, which in itself might be an unrealistic sample, might appear closer to another distribution of data especially if the pixel and spatial variance are high.

5.2.3 Feature Distribution Distance

Instead of performing classification or computing the distance of distributions in the image space, GAN performance can be evaluated by first projecting images to a pre-defined feature space. In this feature space, real and fake images would ideally be further separated and represented as a rich, low-dimensional vector. If these conditions are met, the distances between the distributions of real and fake features can be computed more reliably and provide better insight into GAN performance.

The most prevalent GAN metric utilizes features from an Inception network trained on ImageNet [138]. As this network can outperform humans on natural scene image classification, the learned features share similarities with that of the human visual processing system. Although features of a pre-trained Inception network are biased toward natural scenes and the network itself is restricted to only 2D images, we review two Inception-based GAN metrics as they have become standard in literature [236, 108]. Additionally, it is imperative to further explain why these metrics are not suitable for FM images of neurons.

The first metric we will discuss is the Inception Score (IS) [236]. At its core, this metric quantifies the entropy of predicted class labels given a distribution of generated images. It can be formalized as follows:

$$IS(G) = \exp\left(\mathbb{E}_{\hat{\mathbf{x}}\sim p_g}\left[D_{KL}\left(p(\mathbf{y}_i|\hat{\mathbf{x}}) \parallel p(\mathbf{y}_i)\right)\right]\right) \tag{5.5}$$

where $\hat{\mathbf{x}}$ is a image $\hat{\mathbf{x}}$ sampled from the generated distribution p_g, $p(y|\hat{\mathbf{x}})$ is the conditional probability of a class y given $\hat{\mathbf{x}}$, and $p(\mathbf{y}_i)$ is the marginal probability of class $\mathbf{y}_i$. If, for example, generated images are amorphous and lack clear structure, then the Inception network will be unable to definitively classify images and therefore result in high classification entropy. Consequently, the assumption of this metric is that low-quality images have less class fidelity.

While IS has been shown to be highly correlated with human evaluation of image realness [18], it is limited. For one, it does not directly penalize GANs for mode collapse, as a single realistic image memorized from the training dataset for each mode would perform well. Additionally, because it is trained to predict natural scene categories, only images containing objects found in the real world, and not FM images of neurons, should be used.

Another common Inception-based network is the Fréchet Inception Distance (FID) [108]. Instead of computing the image classifications, this metric computes the dis-

tance between higher-order features for both real and generated images, which can be summarized as:

$$FID(P,Q) = |\mu_P - \mu_Q|^2 + T_r(\Sigma_P + \Sigma_Q - 2(\Sigma_P\Sigma_Q)^{\frac{1}{2}}) \tag{5.6}$$

where P and Q are the distributions of real and generated image features respectively, μ_P and μ_Q are the mean feature representations for real and generated images respectively, and Σ_P and Σ_Q are the covariance matrices of the feature representations for real and generated images respectively. However, the mean and covariance assume that the distribution of features is Gaussian, which is not always the case especially if images are of a distribution different from that of the training data. Additionally, the features used are from the convolutional layer just before the final downsampling layer. Therefore, while not a classifier, it also computes features that are heavily correlated to natural scene categories.

In the ideal case, networks pre-trained on data representative of the tasks at hand should be used. Because this does not exist for 3D FM images of neurons, we can instead compute a feature set that encapsulates a variety of FM imaging properties directly.

5.2.4 Data Utility

A more direct way to assess the usefulness of generated images is to evaluate their overall utility in subsequent tasks [303, 150]. For unconditional GANs, this is challenging due to the absence of labels. However, for conditional GANs, labels used to condition images and the images themselves can be used as training data. Networks can then be either trained exclusively on GAN-generated images or a combination of GAN-generated and real images. This approach is particularly beneficial for image translation tasks, where the subsequent task may involve segmentation. A limitation of quantifying utility is that it is not a direct measure of image quality or image diversity. Additionally, the performance of subsequent tasks is heavily dependent on the architectures used [30].

5.3 Method

5.3.1 Data

While all training was posed as an unpaired image translation task, 200 128×128 paired patches and 100 $4 \times 128 \times 128$ paired patches were synthesized for testing on

2DSyn_BC and 3DSyn_GC datasets respectively so that generated synthetic images and real synthetic images were conditioned on the same voxelized mesh. This allows us to directly evaluate generated and real image similarity. For our 3DFM_AC dataset, evaluation was done on 104 $4 \times 128 \times 128$ unpaired patches as no paired patches exist. Patch sizes and the remaining sampling protocols were conserved between training and testing to ensure the same sampling of features and prevent metrics from biasing toward the null space of FM images containing individual neurons. Unlike the other three datasets that contained individual neurons, our 3DFM_BCpop dataset was of a neural population and therefore contained significantly less null space. Testing on this dataset was done over 8 $36 \times 512 \times 512$ randomly sampled patches with paired semantic maps as input to determine our architecture's ability to generate images larger than training patches. Patches were used instead of the entire $38 \times 2048 \times 2048$ stack due to memory constraints.

5.3.2 Evaluation

We evaluated GAN performance based on 5 main criteria: content, local texture, noise, utility, and expert opinion.

5.3.2.1 Content

To quantify the content realism of our synthetic datasets, we first averaged 25 instantiations for both real and corresponding generated images to reduce pixel noise. The Normalized Mean Squared Error (NMSE) was then calculated between the averaged real and generated images across the neuron body. NMSE is a commonly used metric for comparing the difference between a predicted sample y and a desired sample $\hat{y}$, particularly when the data contains varying intensities [286]. By normalizing the MSE with the power of the desired sample, the metric ensures that the comparison is less sensitive to signal intensity. The NMSE can be described in more detail as follows:

$$NMSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{(\hat{\mathbf{y}} - \mathbf{y})^T(\hat{\mathbf{y}} - \mathbf{y})}{\mathbf{y}^T\mathbf{y}} \tag{5.7}$$

Furthermore, given that background pixels are a significant proportion of our FM images, they can bias the NSME. Therefore, to ensure the computations are focused primarily on the neuronal dendrites, these background pixels were masked out.

Because stacks in the 3D_BCPop dataset were acquired once, the NMSE was computed between real and corresponding generated images for only a single instantiation

when evaluating content realism on 3D_BCPop data. However, the same direct comparison of image content could not be done when evaluating performance on the unpaired 3D_AC dataset as there was no correspondence between meshes and FM images. We, therefore, computed the NMSE of the auto-correlation matrix with size $3 \times 11 \times 11$ of both real and generated images as a way to estimate if the correct point spread function was learned. The auto-correlation matrix is computed as follows:

$$R_{XX}(i,j,k) = \sum_{x=-m}^{m} \sum_{y=-n}^{n} \sum_{z=p}^{p} X(x,y,z) \cdot X(x-i-1, y-j-1, z-k-1) \quad (5.8)$$

where $R_{XX} \in R^{2m \times 2n \times 2p}$ is the auto-correlation matrix, $R_{XX}(i,j,k)$ is the value of the auto-correlation matrix at position i, j, k, and X is a 3D image.

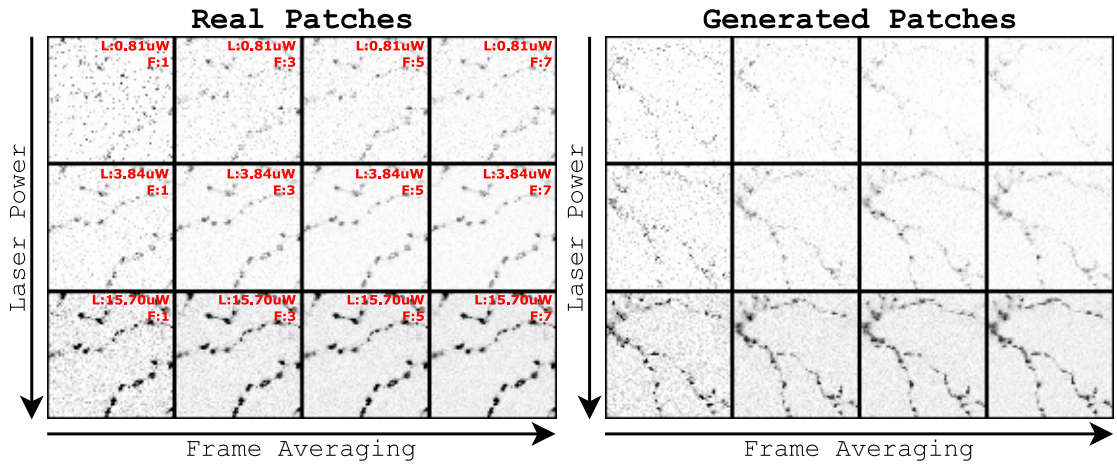
5.3.2.2 Texture

Instead of using a pre-trained neural network to project images into a feature space, we computed local texture-based features that were more descriptive of FM images. The distribution of computed values for both real and generated images was then created and compared. The first texture-based metric employed was the Local Binary Pattern (LBP) [102]. For this method, neighboring pixels to a central pixel were identified. If a neighboring pixel had a greater value than the central pixel, the value was set to 1; if lower, then 0. Concatenating the assigned values in a clockwise direction forms a binary string that represents the pattern surrounding the central pixel. The distribution of all binary strings is calculated which then quantifies the image texture. This process was conducted for both real and generated images and the distribution of patterns was compared using JSD.

As a second texture metric, Gray Level Co-Occurance Matrix (GLCM) was employed [195]. This metric measures the frequency of pixel intensity pairs separated by Δx and Δy in an image. This can be represented in a matrix C as follows:

$$C(i,j;\Delta x, \Delta y) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \begin{cases} 1, & \text{if } I(x,y) = i \text{ and } I(x+\Delta x, y+\Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (5.9)$$

where i and j are integer pixel values between 0 and 255 and $I(x,y)$ is the intensity value of the pixel at locations x and y. Once the GLCM was calculated for each pixel in real and generated images, the MSE between the two matrices was calculated to determine the relative difference between textures.



(a) (b)

Figure 5.1: a) Real and b) generated 64x64 patches (inverted for visual effect) sampled from the 3DFM_AC dataset at varying laser powers and frame averaging (labeled L and F respectively in red). For visual purposes, only a subset of styles are portrayed.

Both LBP and GLCM are limited to 2D images. However, GLCM can be easily extended to 3D using the multi-sort Co-occurence Matrix (COOC) [135]. Because all three metrics are sensitive to voxel noise distributions, a Gaussian filter was additionally employed to blur images prior to the computation of these metrics.

5.3.2.3 Noise

The realism of the noise was additionally computed by taking the MSE of the average Peak Signal-to-Noise Ratio (PSNR) and the JSD of the voxel intensity values for both real and generated images.

The PSNR measures the amount of corruption a signal undergoes given an original signal [41]. It can be summarized as follows:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}_I} \right) \tag{5.10}$$

where MAX_I is the maximum value of the signal and MSE_I is MSE between the corrupted image and the original image. Because the original image must be known, the MSE of the average PSNR was only computed for 2DSyn_BC and 2DSyn_GC datasets.

5.3.2.4 Utility

On real FM datasets, we also evaluated the utility of generated images by calculating the Intersection over Union (IoU) [78] score from predicted segmentation of unseen

			2DSyn_BC					3DSyn_GC						3DFM_AC						
ID	Training Config.	$\mathcal{L}_{style}$	NMSE $\times 10^1$	LBP $\times 10^3$	GLCM $\times 10^8$	PSNR $\times 10^0$	JSD $\times 10^2$	NMSE $\times 10^1$	LBP $\times 10^3$	GLCM $\times 10^8$	COOC $\times 10^8$	PSNR $\times 10^0$	JSD $\times 10^2$	NMSE_{auto} $\times 10^{-1}$	LBP $\times 10^3$	GLCM $\times 10^8$	COOC $\times 10^8$	JSD $\times 10^2$	IoU $\times 10^0$	Avg. Rank
CycleGAN			4.691	9.119	4.115	16735.152	12.709	8.047	96.143	17.062	9.884	13841.816	3.536	49.685	45.440	30.005	19.533	11.577	0.142∓0.201	
SPADE			0.872	0.861	0.371	0.688	1.057	2.978	0.407	3.618	0.927	118.636	4.157	17519510.059	2.707	14.172	2.028	1.967	52.242∓3.278	2.880
A	SPADE + $\mathcal{L}_{seg}$		0.955	1.098	0.739	0.895	1.511	1.892	0.381	**1.521**	0.593	147.718	3.930	33988.546	1.637	16.445	2.108	2.359	58.973∓0.118	
B	A+Instance_Norm		0.434	0.410	0.108	0.083	0.137	2.058	0.708	4.467	0.871	200.654	3.187	1.854	1.494	5.576	0.903	0.823	64.480∓1.38	2.867
C	B+Spatial_stddev		0.930	0.300	0.078	0.191	0.095	1.934	0.833	2.350	0.857	816.951	4.539	4.721	1.209	6.631	2.269	0.876	70.121∓2.618	
D	C+Acquisition		0.457	0.240	0.079	0.482	0.087	1.810	**0.236**	4.386	1.191	63.312	2.167	div*	div*	div*	div*	div*	div*	
E	D+Pixel_Norm		0.490	0.346	0.115	1.109	0.081	1.869	0.262	4.116	1.121	36.106	**2.036**	1.581	3.021	4.621	0.741	1.138	68.990∓3.845	
F	E−$\mathcal{L}_{percep}^{VGG}$ **		0.443	0.202	0.099	0.043	0.048	2.023	0.667	3.220	0.620	10.597	2.101	3.476	3.406	7.334	3.079	2.038	72.216∓1.210	
G	F	$\mathcal{L}_{style}^{VGG}$	0.442	0.168	0.262	0.014	0.054	1.568	0.693	6.122	1.655	9.175	3.715	1.327	1.682	**1.830**	0.523	**0.519**	75.206∓0.670	
H	F	$\mathcal{L}_{style}^{D_{conv}}$	0.434	**0.166**	**0.016**	0.015	0.027	1.604	0.399	2.905	0.398	4.229	2.401	1.673	1.177	6.219	**0.299**	0.711	59.871∓1.111	
Ours	F	$\mathcal{L}_{style}^{D_{conv}} + \mathcal{L}_{style}^{D_{gram}}$	**0.401**	0.204	0.032	**0.001**	**0.011**	1.824	0.500	1.962	0.409	1.932	2.354	1.055	1.637	3.912	0.552	0.646	76.008∓0.877	2.267
I	F	$\mathcal{L}_{style}^{D_{gram}}$	0.678	0.387	0.072	22.750	0.056	1.683	0.491	4.336	1.217	8.369	2.230	1.622	1.311	5.631	0.327	0.987	**77.938∓0.568**	
Alt.	F−Spatial_stddev	$\mathcal{L}_{style}^{D_{conv}} + \mathcal{L}_{style}^{D_{gram}}$	0.557	0.314	0.185	6.997	0.131	**1.279**	0.419	2.034	**0.373**	**0.287**	2.657	**1.007**	**1.269**	3.044	0.348	0.777	75.252∓3.076	**1.987**

Table 5.1: A quantitative ablation study of our proposed unpaired mesh-to-image translation architecture starting from a baseline architecture (SPADE) for three datasets. CycleGAN was additionally used as a baseline comparison. Various style losses (column $\mathcal{L}_{style}$) were additionally tested to isolate their impact on training. Metrics used to evaluate our architectures include content metrics (NMSE and $NMSE_{auto}$), local texture metrics (LBP, GLCM, and COOC), noise distribution metrics (PSNR and JSD), a utility metric (IoU), and an expert opinion test (Avg. Rank). More details about individual metrics can be found in the Evaluation section. Darker green and darker red signify better and worse performing architectures for a given metric and the best performing architecture was bold. 'div*' designates when an algorithm diverged and $\mathcal{L}_{percep}^{VGG}$ ** designates the perceptual loss used in [204] and defined in [128].

images. Specifically, we trained an additional network only on generated images to perform segmentation for each FM style. The segmentation network was a U-Net with 3 downsampling layers. Instance normalization was used at the first layer to ensure images with less signal could be segmented. The segmentation network was trained 3 separate times and the average IoU score was reported. For the 3D_FM dataset, ground truths for the real data were acquired using the Frangi filter on the highest-quality FM stack and a total of 4 never-before-seen stacks were used for testing. Likewise, a never-before-seen stack was used for testing on the 3D_BCPop dataset.

The IoU score is preferred over the accuracy, as a large portion of the image consists of background noise. Predicting only the background would yield an accuracy in the high 90%, resulting in a highly biased outcome. IoU is better suited for thinner objects [78] and can be described as follows :

$$IoU(\hat{Y}, Y) = \frac{Area(\hat{Y} \cap Y)}{Area(\hat{Y} \cup Y)} \tag{5.11}$$

where $\hat{Y}$ is the predicted segmentation and Y is the ground truth segmentation.

5.3.2.5 Expert Opinion

Finally, we evaluated selected algorithms trained on the 3DFM_AC dataset through a blind ranking test with 6 expert microscopists working in neuroscience at the National Institute of Neurological Disorders and Stroke. Specifically, each expert was given 15 sets of randomly sampled $6 \times 128 \times 128$ patches containing a real patch and 4 generated patches from every selected algorithm. The real patch served as a reference style and expert microscopists were tasked to rank the generated images in the set by their realness (1 being most real and 4 being least real). No ties were allowed in the ranking. Experts were asked to focus on the point spread function, the change in intensity along the dendrites, and the background noise. The average rank was reported for the expert opinion. The contents of the email describing the task and some examples can be found in the Appendix Figs. A.1 and A.2.

5.3.3 Comparisons

We compare our proposed architecture to two state-of-the-art image translation architectures. The first comparison is with CycleGAN, which remains the standard unpaired image translation architecture and is described further in Section 2.6.3. The second architecture we compare to is SPADE, which served as the foundation for our

		3DFM_BCPop					
ID	Unpaired	NMSE x10^1	LBP x10^3	GLCM x10^8	COOC x10^8	JSD x10^2	IoU x10^1
CycleGAN	✓	2.228	2.557	0.055	0.077	1.212	49.249±1.811
SPADE	✓	2.244	0.246	0.379	0.007	0.590	54.401±1.447
A	✓	3.192	1.066	3.158	0.062	1.755	50.340±1.125
B	✓	1.541	0.276	0.015	0.010	0.107	55.357±0.656
C	✓	1.524	0.111	0.064	0.031	0.185	57.930±0.423
D	✓	2.013	0.112	0.009	0.013	0.088	58.177±0.744
E	✓	1.570	0.085	0.016	0.008	0.077	56.477±1.006
F	✓	1.622	0.100	0.214	0.170	0.530	54.455±1.585
G	✓	1.641	**0.046**	0.114	0.084	0.162	61.351±0.487
H	✓	1.535	0.077	0.049	0.028	0.052	58.959±2.770
Ours	✓	1.535	0.094	0.027	0.013	0.067	59.652±0.708
I	✓	1.819	0.176	0.124	0.078	0.881	61.770±0.871
Alt	✓	1.560	0.086	**0.008**	0.004	**0.026**	58.006±1.869
SPADE		1.353	0.229	0.035	**0.003**	0.500	62.948±0.760
Ours		**1.317**	0.231	0.014	0.011	0.199	**66.609±0.363**

Table 5.2: A quantitative ablation study on our proposed architecture comparing unpaired and paired training performance on our 3DFM_BCPop dataset. Metrics were the same as those in Table 2. Darker green and darker red signify better and worse performing architectures for a given metric and the best performing architecture was bold. Note that the 3DFM_BCPop dataset uses a semantic map instead of a mesh to condition our generator so that both unpaired and paired training regiments can be tested.

architecture design. As both CycleGAN and SPADE are primarily intended for 2D image translation, we deliberately evaluated all architecture with 2D images from our 2DSyn_BC dataset. Any performance improvement in 2D would therefore highlight the theoretical advances proposed in Chapter 4. For the remaining 3D experiments, both CycleGAN and SPADE were adapted by replacing 2D convolutions with 3D convolutions. Because SPADE utilizes a pre-trained VGG-net to compute perceptual and style loss, individual 2D slices from the 3D FM image patch were passed instead. More so, because SPADE is designed for paired image translation, we additionally compared performance on paired translation tasks using our 3DFM_BCPop dataset. Similarly, any performance improvement on paired translation would further validate the architectural advances proposed. To adjust SPADE for unpaired image translation, we then simply the SPADE generator using an unconditional discriminator. Finally, we conducted a thorough ablation study of our novel architecture. This enabled us to evaluate the effectiveness of each introduced architectural component.

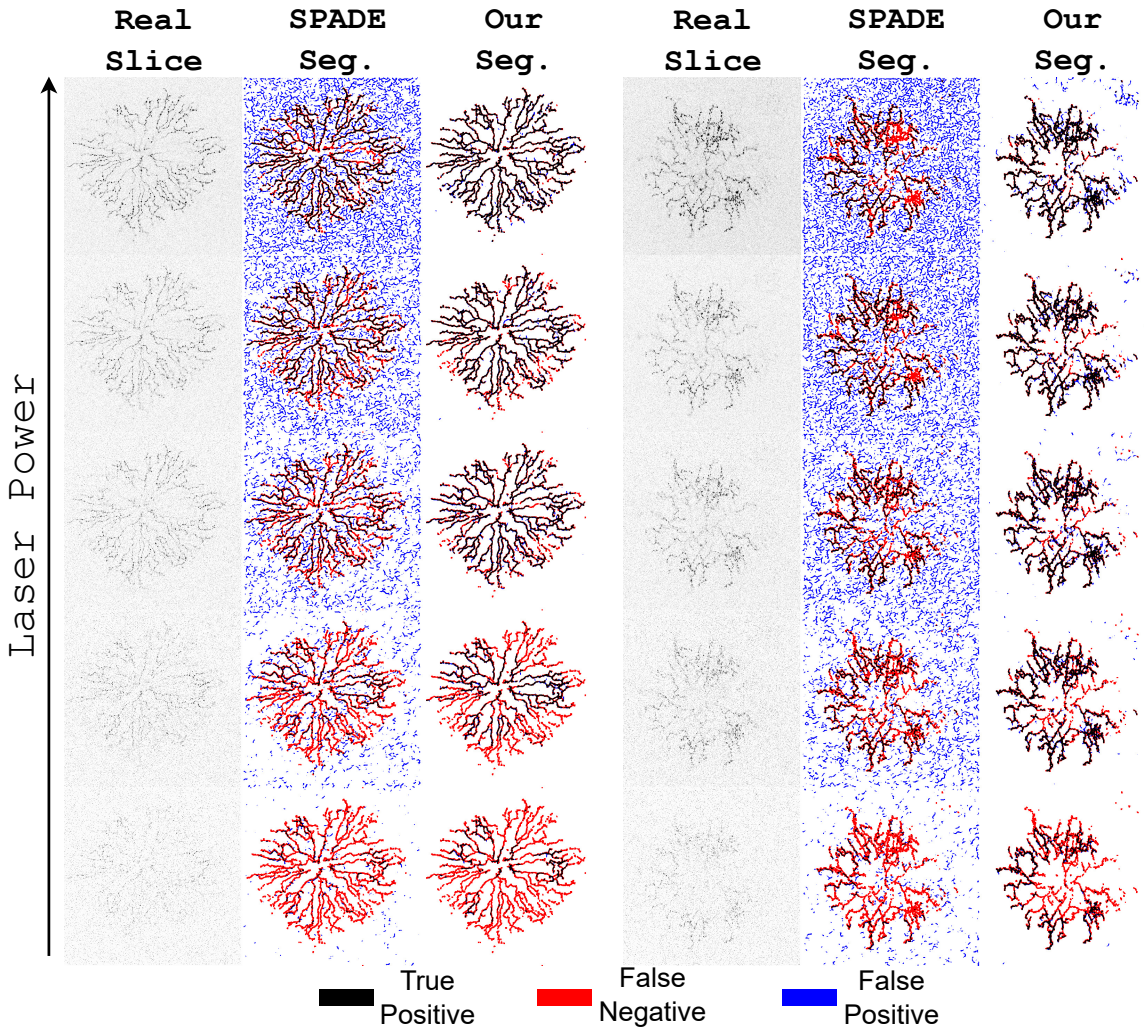


Figure 5.2: Segmentation predictions on never-before-seen FM slices (inverted for visual effect) at varying laser powers using generated training datasets from SPADE and our generator.

5.4 Results

Quantitative and qualitative comparisons show that our proposed architecture more accurately generates realistic FM images across several datasets compared to CycleGAN, SPADE, and SPADE's various iterations. While it is challenging to visually distinguish algorithms on the synthetic datasets (Figs. 5.6 and 5.7), we see clearly see that the CycleGAN fails on both. Quantitatively, our architecture is a top performer by almost every computed metric (Table 5.1 and 5.2). Specifically on the 2DSyn_BC, our architecture outperforms both CycleGAN and SPADE architectures on almost every metric, improving on some metrics by an order of magnitude. This result is important as both CycleGAN and SPADE were designed for 2D image translation.

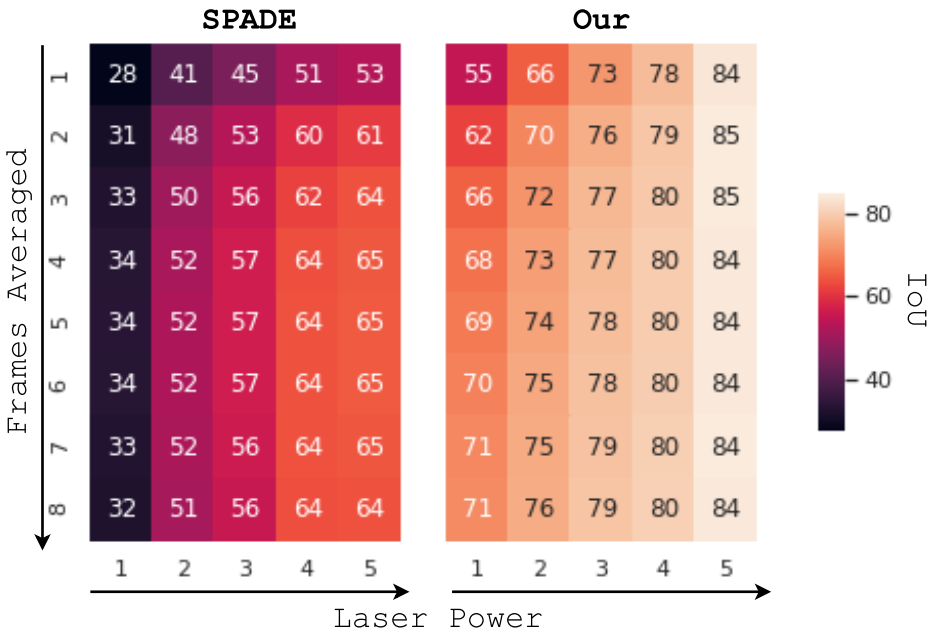


Figure 5.3: Average intersection of union (IoU) scores across multiple frame averaging and laser powers using generated training datasets from SPADE and our generator. Note that our generated training datasets improve IoU scores by nearly 40 points for low signal-to-noise images.

More drastic differences, therefore, can be expected and are also seen in the real 3D FM datasets.

Qualitatively, our architecture demonstrates improved modeling of the reference style while preserving the dendritic structure, as evident in Figures 5.8 to 5.10. CycleGAN performs the worst, particularly failing on the 3DFM_AC dataset. Although SPADE manages to retain the dendritic structure for the 3DFM_AC dataset, there are instances where the signal is entirely lost or extraneous blobs appear around the intended dendrite locations. These findings are corroborated by quantitative results where we again improve on nearly every metric (Table 5.1 and 5.2) relative to the CycleGAN and SPADE architectures. Most notable is the nearly 24 and 5-point improvement on average IoU score when training a separate segmentation network only on our generated images from the 3DFM_AC and 3DFM_BCPop datasets respectively. On closer inspection, we see that our generated synthetic data is able to train a segmentation algorithm to accurately segment neurons from noisy FM images (Fig. 5.3), sometimes with nearly a 40-point improvement in IoU. Results from our expert opinion test additionally verify the noticeable improvement in image realism when generating patches with our model as our generated patches were consistently ranked higher than the two baseline models. When comparing paired and unpaired train-

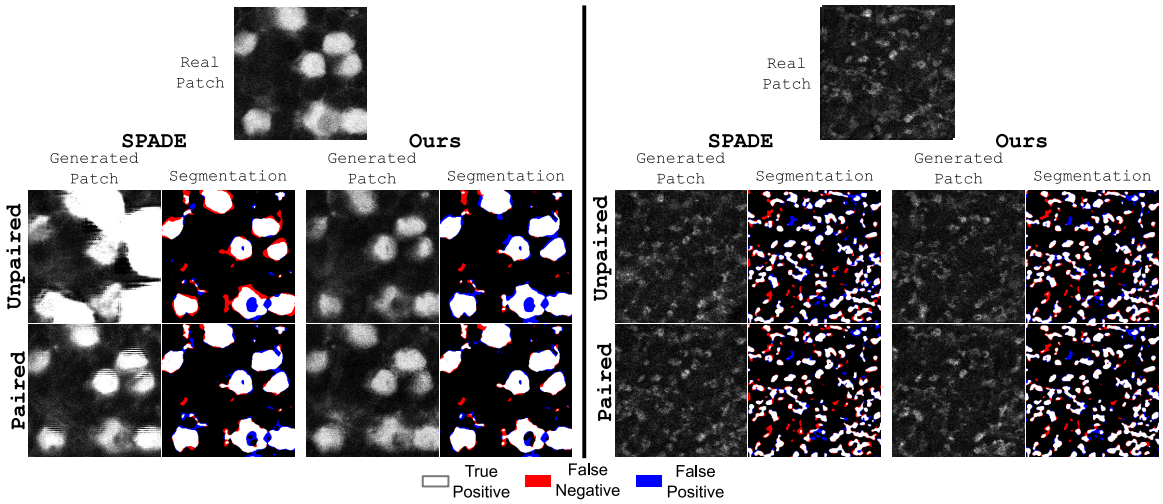


Figure 5.4: Qualitative comparisons between real, SPADE-generated, and Our-generated patches using unpaired and paired training regiments from the 3DFM_BCPop dataset. Generated patches were then used to train a separate machine learning algorithm to perform segmentation. To the right of every generated patch is the predicted segmentation of the real patch from the trained segmentation network.

ing paradigms (Table 5.2), we also see that paired training increases performance as expected. However, our architecture when trained with unpaired data remains competitive with SPADE when trained with paired data, highlighting the importance of our architectural advances. These results are further illustrated qualitatively (Fig. 5.4) where noticeable artifacts are seen in paired training of SPADE and not in unpaired training of our architecture.

Finally, our results demonstrate the utility of the architectural components introduced in this paper. Replacing $\mathcal{L}_{style}^{VGG}$ with $\mathcal{L}_{style}^{D_{conv}}$ and $\mathcal{L}_{style}^{D_{gram}}$ improves the architecture's performance on most computed metrics for all datasets as the VGGNet is pre-trained on natural scene images devoid of the stochastic textures seen in FM images. More so, $\mathcal{L}_{style}^{VGG}$ is limited to 2D images, while $\mathcal{L}_{style}^{D_{gram}}$ is custom to the domain at hand. Our `Aquisition` layer additionally allows us to decouple the signal from the noise so that we can generate images with new noise profiles and images that are theoretically impossible to obtain, such as denoised FM images (Fig. 4.6). As for our `Spatial_stddev` layer, it allows our architecture to better learn the spatial distribution of structured noise. Figure 5.5 shows that the `Spatial_stddev` layer forces the blobs to be randomly spread throughout the background of generated bipolar cell images and minimizes uniform regions and erroneous structures from the generated Ganglion Cell images. However, for images that do not contain structured noise, like

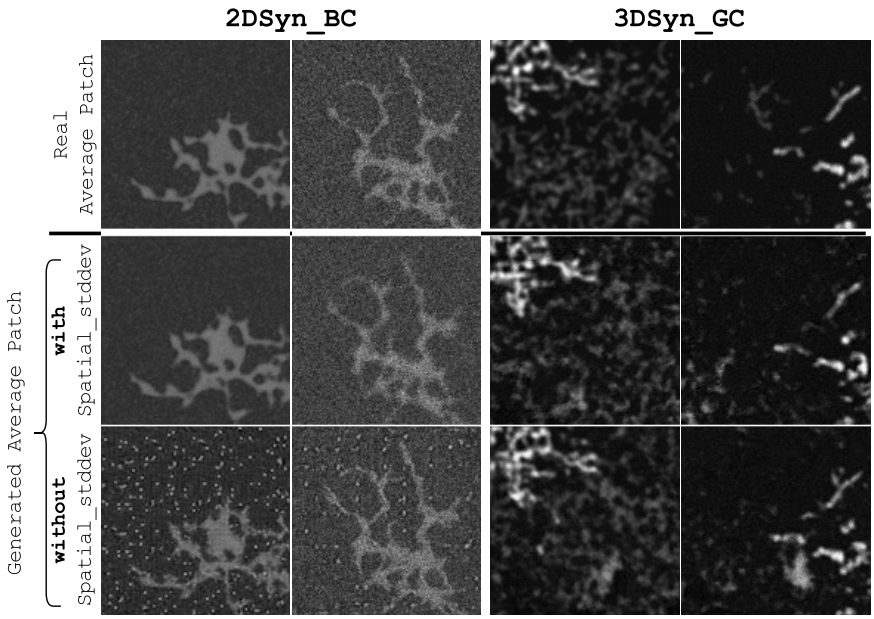


Figure 5.5: Comparison of average generated patches with and without `Spatial_stddev` in the discriminator for both synthetic datasets. 20 samples were generated and then averaged together. For the Ganglion Cell dataset, stack noise ζ was frozen so that the structure of the dendrites were preserved when frames were averaged. We see that the inclusion of `Spatial_stddev` increases the spatial variability of each generated instance, while the exclusion of `Spatial_stddev` leads to mode collapse and erroneous background textures.

our FM datasets, the `Spatial_stddev` layer can hurt performance as evidenced by the increase in most content and texture-based metrics when it is left out of the discriminator. While the performance decrease is not significant, we suggest additional training with `Spatial_stddev` when testing different architecture designs that have limited structured noise.

5.5 Conclusion

Our unpaired mesh-to-image translation architecture was evaluated on four FM image datasets, two of which were synthetic datasets and two of which were newly acquired 3D FM datasets that are available for public use. Using a variety of evaluation metrics, we show that our generator is able to learn stochastic textures and structured noise more accurately than two state-of-the-art architectures across all datasets. Among the architectural advances, we show that learning styles lost with a trainable Gram Matrix operation enable the generator to learn more realistic FM textures than using a VGG-net trained on natural scene images.

The utility of our pipeline is further demonstrated by its ability to generate a training dataset for a segmentation network such that FM images of neurons can be accurately segmented with no manual annotation required. While segmentation was our only measure of utility, additional features of FM images can be predicted with the same pipeline. For instance, soma size can be automatically determined from volumetric meshes and serve as labels for generated images. Both generated images and labels can then be used to train a network to identify the soma size again without the need for manual annotation. Therefore, instead of collecting extensive annotated datasets of FM images that have limited utility, we suggest consolidating datasets of reconstructed neurons which offer significantly greater potential for analysis.

More work, however, needs to be done to test the complexity of features our architecture can produce as our generator is conditioned only on the neuron of a specific type and species. Knowledge of the lower limit of real data points required to sufficiently model FM styles should also be explored in future works as it would give neuroscientists a sense of how much image data is required to automate a task.

In this chapter, we also introduce an evaluation protocol for GAN-generated FM images of neurons. Understanding biases present in GAN metrics is essential for effectively evaluating GAN performance, though it may not always be intuitive. To reduce potential bias, we chose to use metrics from feature extractors that are well-understood, allowing us to have a better grasp of what we were measuring. Because every used metric measures a unique property of FM images, it is therefore important that the GAN perform well on all metrics rather than be a top performer on a select few metrics. Although GAN metrics have been tested for natural scenes, future research should focus on developing more generalizable metrics to evaluate GAN performance.

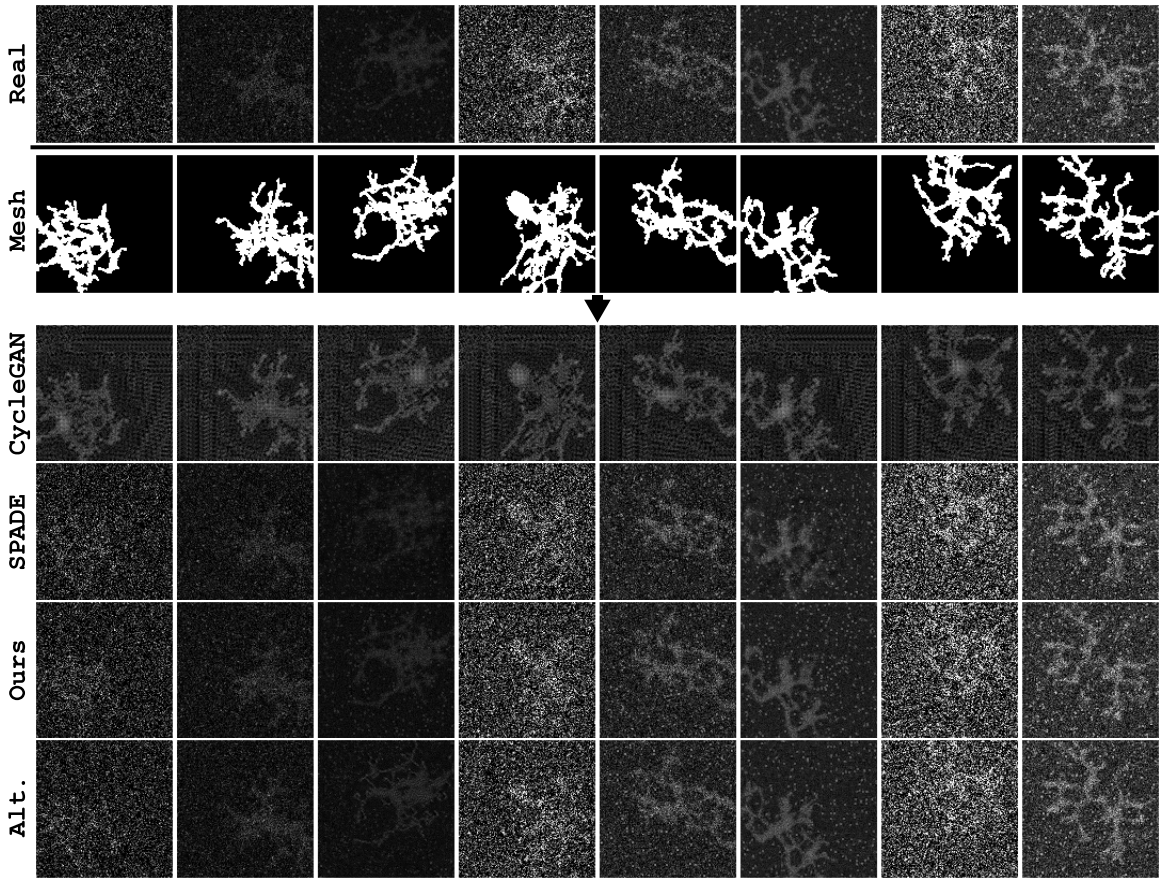


Figure 5.6: Real and generated 128x128 patches sampled from the 2DSyn_BC dataset. While unpaired images were used during training, all real and generated patches were created based on the same voxelized mesh to make visual comparisons easier.

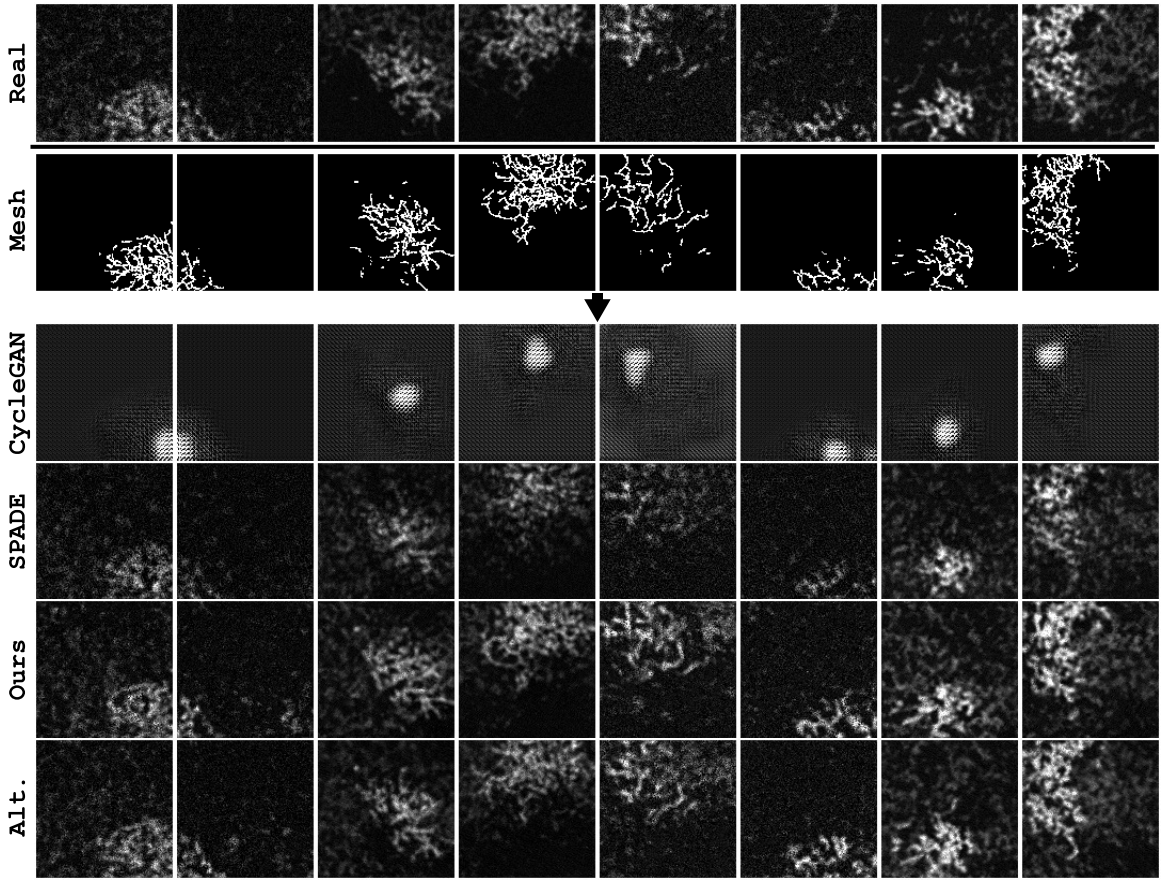


Figure 5.7: Real and generated 1x128x128 patches sampled from the 3DSyn_GC dataset. While unpaired images were used during training, all real and generated patches were created based on the same voxelized mesh to make visual comparisons easier.

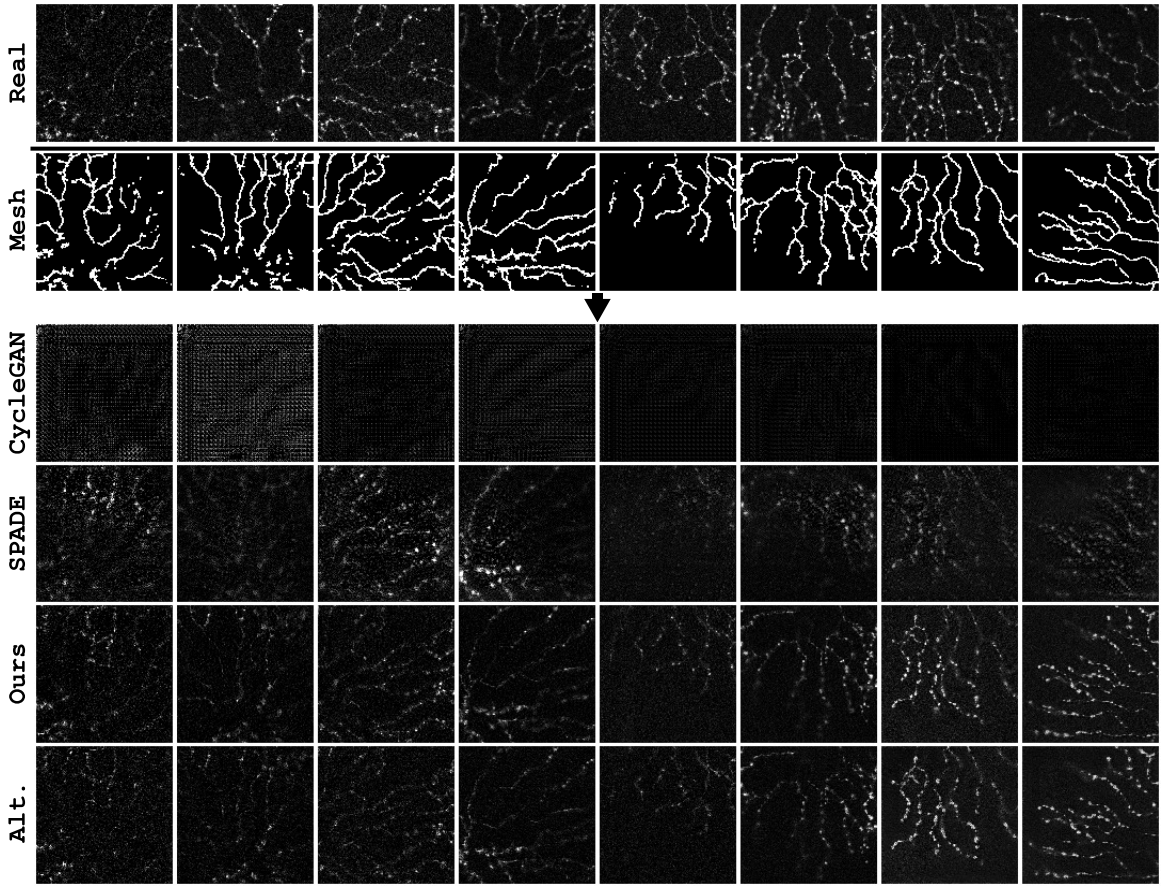


Figure 5.8: Real and generated 1x128x128 patches sampled from the 3DFM_AC dataset.

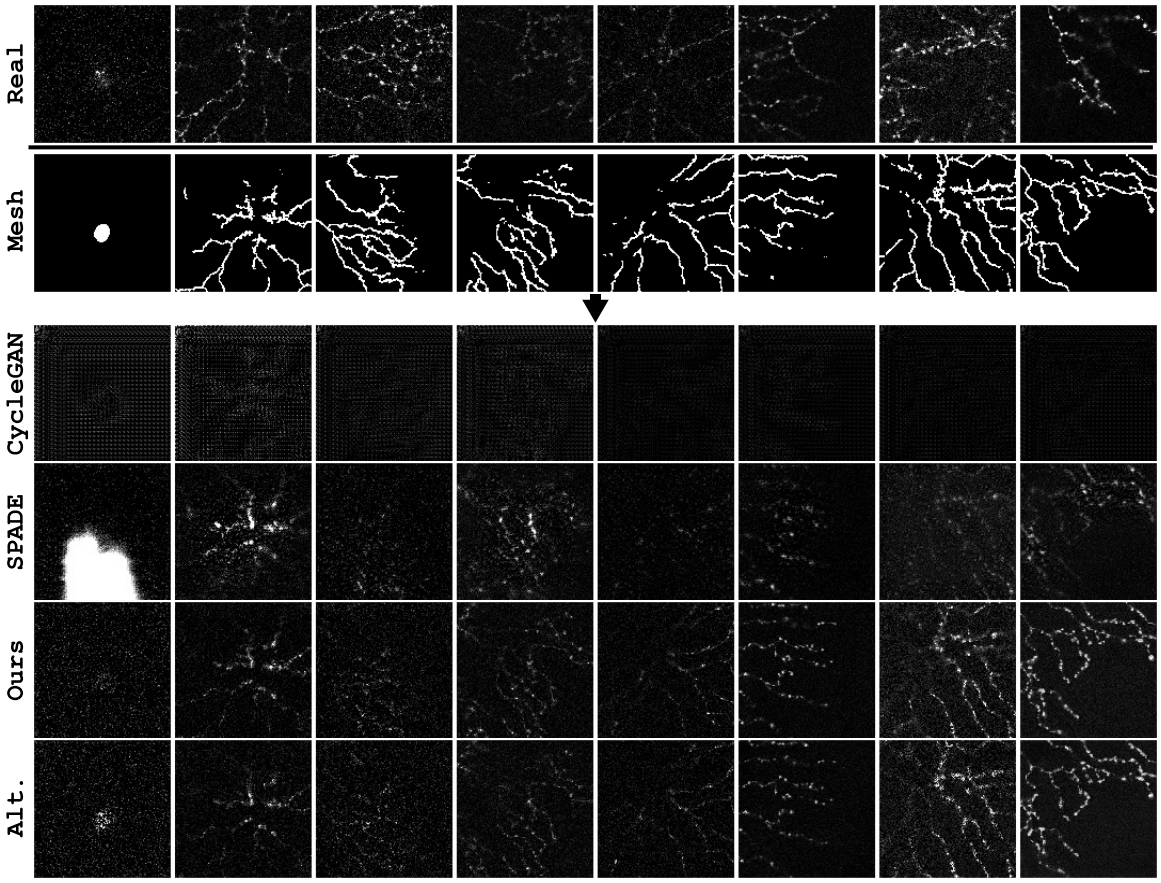


Figure 5.9: Real and generated 1x128x128 patches sampled from the 3DFM_AC dataset.

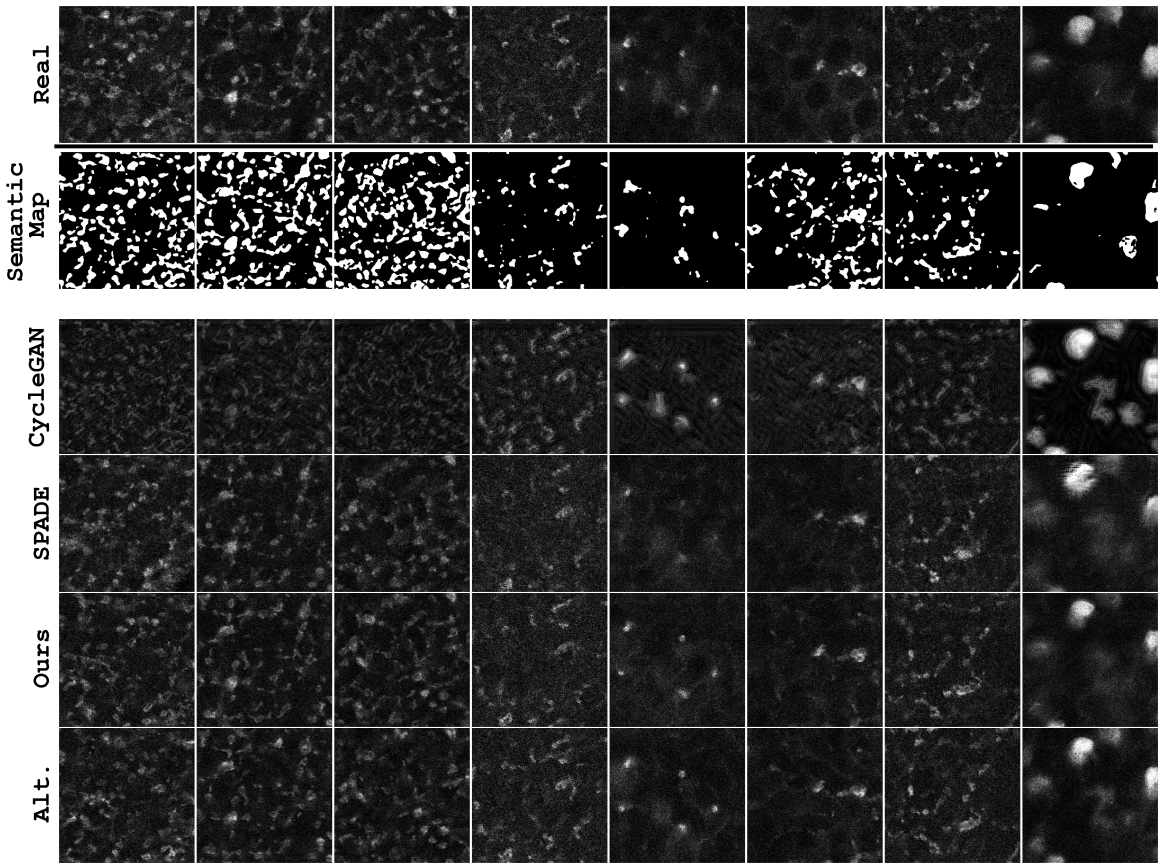


Figure 5.10: Real and generated 1x128x128 patches sampled from the 3DFM_BCPop dataset.

Chapter 6

Multi-scale Deep Learning for Neural Tracing

Contents

6.1 Introduction 111
6.2 Related Works 113
6.2.1 Class Imbalance 113
6.2.2 Image Denoisers 115
6.2.3 Segmenting Thin Structures 117
6.2.4 The Mammalian Retina 118
6.3 Methodology 120
6.3.1 Pre-training on GAN-Generated Images 120
6.3.2 Local Residual Denoiser 120
6.3.3 Global Segmentation with Enlarged Ground Truths 121
6.3.4 Multi-Scale Fusion Loss 121
6.4 Implementation 122
6.5 Results 123
6.6 Conclusion 124

6.1 Introduction

Automated neural tracing is a fundamental tool in neuroscience, enabling researchers to investigate the morphologies of individual neurons and the connectivity of biological neural networks [179, 3]. While digital traces of neurons can be generated from various imaging modalities [123, 143, 313], they are particularly valuable in the analysis of fluorescence microscopy (FM) images [166]. FM is widely accessible, and unlike other

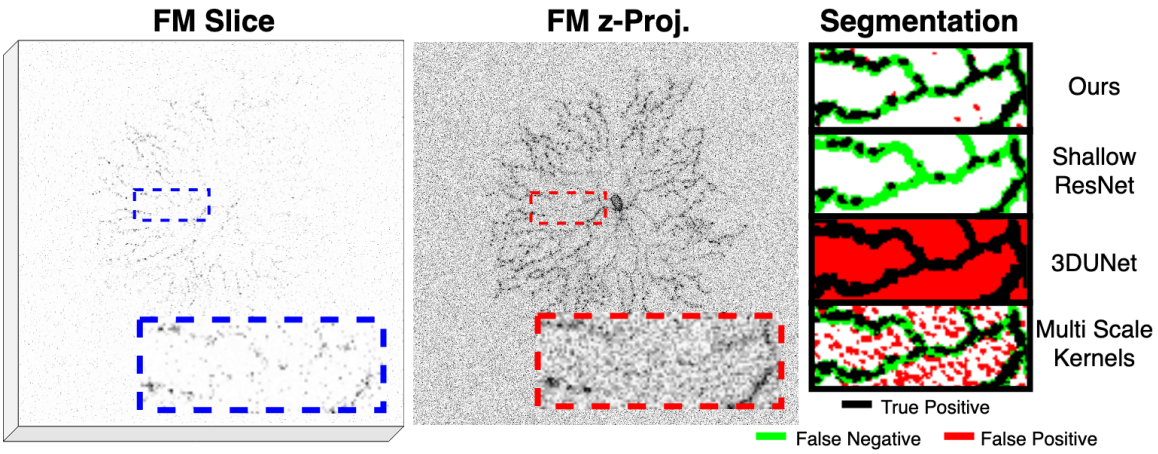


Figure 6.1: An illustration of the challenges involved in segmenting low-SNR 3D FM images of neurons. *Left Panel*: A slice of an example low-SNR FM image provided by the $3DFM_AC$ dataset. *Middle Panel*: A maximum z-projection of the same example FM image. At the bottom right of both the slice and maximum z-projection, we can see a magnified region highlighting the obfuscation of the signal through significant amounts of Poisson noise. *Right Panel*: From top to bottom, we show the segmentation predictions for the region of interest from our proposed architecture, a Shallow ResNet, a 3DU-Net, and MultiScaleKernel [283].

microscopy techniques, it can image neurons deep within fixed or live tissue with high sensitivity and spatial resolution. However, acquiring high-quality FM images is not always feasible due to weak expression of fluorescent markers, nonspecific or heterogeneous labeling of neurons, and potential neuron damage caused by prolonged laser exposure. As a result, significant amounts of morphological and voxel noise are often present in FM images.

Classical automated neural tracing algorithms typically rely on connecting generated seed points along the neuron's dendrites through ridge detection [12, 127], active contour analysis [241, 285], or graph theory [181, 210, 47]. However, these methods are especially sensitive to obfuscations of the neural signal and therefore struggle to trace neurons from low Signal-to-Noise Ratio (SNR) FM images [179]. Convolutional neural networks (CNNs) have shown promise at handling a wide variety of noise and have been used for neural tracing as a pre-processing step by segmenting or enhancing neurons present in FM images [153]. However, segmenting low-SNR 3D FM images with CNNs poses additional challenges. For one, voxels containing neural signals make up a small minority of 3D FM image voxels, creating extreme class imbalances that bias networks toward non-neural voxels. Additionally, the variability of signal expression across FM images decreases performance in the presence of noise as CNNs

are further biased toward high-SNR images. Finally, thin dendrites imaged by FM can appear as single voxels or partial volume signals which are then missed by convolutions. While current approaches attempt to overcome these issues by utilizing a single encoder-decoder architecture framework [54], resulting segmentation predictions can be patchy and discontinuous as seen in Fig. 6.1.

In this chapter, we propose a retina-inspired 3-stage CNN-based segmentation pipeline that fuses local and global scales for low-SNR 3D FM images. We first handle the signal variability of FM images by passing images through a local denoiser with instance normalization employed after the first convolutional layer. Then, to preserve thin structures while learning global information, the remaining signal is amplified by a second architecture trained on a ground truth that is enlarged by a Maximum Intensity Projection (MIP). Both local and global information are fused to produce a final segmentation prediction. We alleviate the need for large amounts of training data by pretraining our network on GAN-generated images and then fine-tune on real data. Overall, our proposed architecture outperformed 6 competing architectures on neuron segmentation from a 3D FM dataset of retinal neurons.

6.2 Related Works

6.2.1 Class Imbalance

Deep learning models often converge to sub-optimal solutions during optimization. One example of this occurs during the training of a network in the presence of high-class imbalance [103, 40]. Ideally, a network should learn the underlying principles governing classification; however, the overall performance typically decreases with class imbalance as the network becomes biased toward the more frequent class. In cases where the class imbalance is extreme, like for FM images of neurons where there are significantly more voxels of the background than the neural signal, the network might only output the most frequent classes.

Several strategies can be employed to address class imbalance for deep learning. The most common approach is to weigh the loss from a specific class. To illustrate this strategy, consider the example of Cross-Entropy loss (CE) which is the most popular loss function used for classification. For binary classification, CE can be written as:

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases} \tag{6.1}$$

where $y \in -1, 1$ denotes the ground truth class and p denotes the estimated probability for class $y = 1$. This can be further simplified to:

$$CE(p_t) = -\log(p_t) \tag{6.2}$$

such that

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{6.3}$$

Then, a class weight can be added to the loss function as follows:

$$CE(p_t) = -A(y)\log(p_t) \tag{6.4}$$

such that

$$A(y) = \begin{cases} \alpha & \text{if } y = 1 \\ 1 - \alpha & \text{otherwise} \end{cases} \tag{6.5}$$

where α is the loss weight for class $y = 1$ and is typically set to the inverse frequency of class $y = 1$. Although this method can improve performance, it can also bias the network toward the minority class [40]. This can cause the model to become overly sensitive and learn features that are not generalizable to the overall dataset, resulting in poor performance on unseen data.

A more effective solution is to use cost-sensitive learning. By changing the cost function so that well-predicted values have a significantly smaller loss, the network will focus on harder-to-predict examples. This allows the network to focus on the infrequent classes while still having an understanding of the probability of their occurrence. This enables the network to prioritize infrequent classes while still maintaining an understanding of the probability of their occurrence. A common cost-sensitive learning technique is the Focal Loss [160] as described here:

$$FL(p_t) = -(1 - p_t)^{\gamma}\log(p_t) \tag{6.6}$$

where $\gamma \geq 0$ serves as a tunable parameter. An alternative cost is the Tversky Loss [234]:

$$TL(\epsilon, \beta) = \frac{TP}{TP + \alpha \cdot FN + \beta \cdot FP} \tag{6.7}$$

where TP is the true positive rate, FN is the false negative rate, FP is the false positive rate, ϵ and β are the FN and FP weights respectively. If both $\epsilon = 0.5$ and

$\beta = 0.5$, then we can achieve the dice coefficient [234]. It is important to note that this loss function does not rely on the more frequent true negatives, making it less sensitive to extreme class imbalance. However, due to its emphasis on the overlap between ground truths and predicted segmentations, it tends to identify blob-like shapes. Although this is beneficial in many biomedical imaging contexts, blob-like segmentations can be problematic for thin structures, like those found in FM images of neurons.

Both the Focal loss and the Tversky loss have also been combined to create the Focal Tversky loss [2] which combines the benefits of both loss functions. The governing equation in this case is:

$$\mathrm{FTL}(\epsilon, \beta, \gamma) = (1 - \mathrm{TI}(\epsilon, \beta))^{\gamma} \tag{6.8}$$

This was the loss function used when training our model.

6.2.2 Image Denoisers

Although image denoising is typically considered a separate task from image segmentation, noise presents a significant challenge for any type of image analysis. Not only does noise obscure the signal in the image, but it also makes it more difficult to converge to an optimal segmentation solution as the noise can propagate forward to feature maps upstream. We, therefore, turn to literature for automated image denoising as these architectures are designed to specifically handle significant amounts of noise.

Images can be pre-processed for analysis by applying smoothing techniques, such as an averaging or Gaussian filter. However, these methods spread information spatially, thereby further weakening the signal to be segmented. Anisotropic filters [213] and the BM3D [61] denoiser have been proposed, but often require strict assumptions about the noise in the data or are computationally inefficient.

Deep learning has been proposed to implicitly learn the underlying image noise distributions in order to denoise the image. If multiple instances of an image with the same noise properties are available, then the Noise2Noise training paradigm can be used where given one instance of the image, the network attempts to predict another instance of the image [149]. Because the noise in images is sampled independently of each other, it is impossible for one noise sample to predict another. Consequently, when Mean Squared Error (MSE) is used as a loss function, the model learns the average pixel value of the distribution.

If only one instance of the noisy image exists, then a Noise2Void training paradigm can be employed [139]. In this approach, a CNN with a blind spot at the center of the receptive field is tasked with predicting the center pixel, forcing the network to utilize surrounding regions for denoising. Although this self-supervised training paradigm enables models to learn a non-linear denoising function, the results are limited because the center pixel, which usually contains the most relevant information for the denoised pixel, is not considered.

If a corresponding denoised image is available, the network can be trained to perform image reconstruction in a supervised training framework [311]. Instead of predicting the denoised image directly, however, the most common approach is to predict the residual image noise [311]. This denoising objective can be formalized for additive noise as follows:

$$\mathbf{y} = \mathbf{x} + \mathbf{v} \tag{6.9}$$

where $\mathbf{y}$ and $\mathbf{x}$ are the noisy and clean images respectively, and $\mathbf{v}$ denotes the image noise. Since $\mathbf{x}$ is not observed, the objective can be reformulated as:

$$\mathbf{x} = \mathbf{y} - \mathcal{R}(\mathbf{y}) \tag{6.10}$$

where $\mathcal{R}(\mathbf{y})$ is image residual equal to $\mathbf{v}$. In this supervised learning framework, the network can focus on learning the lost details in the image rather than the original image itself. Additionally, because the mean of $\mathcal{R}(\mathbf{y})$ would be centered around 0, the bias of the output layer is effectively eliminated. This enables the network to converge faster, leading to improved performance and more efficient training than if the CNN were to output the denoised image directly.

Since the introduction of residual learning for image restoration, several additional training paradigms have been developed that build upon this concept. These typically involve modifications to the network architecture, such as adding more layers, using dilated kernels, or adopting an encoder-decoder topology [44, 203]. To enhance the quality of the image and make it appear more natural, a pre-trained, such as a VGGNet, can be used to minimize the perceptual loss between the original and denoised image [91]. Furthermore, a GAN framework with a discriminator trained on high-quality images can be implemented[298].

6.2.3 Segmenting Thin Structures

Several CNN-based architectures have been proposed to segment neurons in 3D FM images. To reduce the number of parameters needed to be determined in training, 2D CNNs have been used on several occasions by projecting 3D images to 2D [47, 314]. This can be done through a maximum intensity projection (MIP) of the entire 3D image or by treating each slice as a unique 2D image. Liu et al. [164] alternatively concatenated cross-sections of a 3D patch at 9 different orientations to integrate depth information into 2D analysis. The effectiveness of using 2D CNNs on these images, however, is entirely dependent on the orientation and density of the neuron's dendrites.

For more generalized neuron segmentation, variations of 3D CNNs based on an encoder-decoder architecture with skip connections have been proposed [54]. Li et al. [153] introduced the first 3D segmentation architecture using inception layers instead of regular convolutional layers. Wang et al. [283] extended this architecture by adding multi-scale fusion blocks throughout a 3D U-Net. Efforts have also been made to incorporate global information into the network. Liu et al. [163], for example, implemented anisotropic kernels throughout a V-Net, a encoder-decoder CNN for 3D biomedical images, to increase the xy-cross section receptive fields of convolutions. Alternatively, Li and Shen [152] used a 3D U-Net with dilated kernels and spatial pyramid pooling at the network's deepest layer. Despite success using encoder-decoder architectures for FM images, they still struggle to accurately analyze images when the SNR is low as thin dendrites are commonly overlooked and mixed with global information in unwanted ways.

Thin object detection has additionally been explored for natural scenes and biomedical images. For example, TO-Net integrates local, global, and fusion streams to segment natural scene images [156]. Similarly, several 2D vessel segmentation techniques attempt to fuse different scales by combining features learned by multiple architectures. Yan et al. [300], for example, trained a network to segment thick and thin vessels separately and then fused the different segmentation predictions together. Furthermore, Galdran et al. [88] stacked two U-Nets together, achieving state-of-the-art performance on retinal fundus image segmentation. Similar techniques have also been explored for 3D vessel segmentation. We draw inspiration from this literature to train multiple architectures on local and global scales for 3D FM image segmentation of neurons.

Finally, because FM images of neurons with paired ground truths are scarce, CNNs are increasingly pre-trained on synthetic images. The generation of synthetic training

data is often achieved through manually designed transformations and adding Poisson noise to texturize previous volumetric reconstructions of neurons [47, 152]. However, these approaches are limited as they can only make local transformations and are unable to learn more complex forms of noise. Alternatively, Generative Adversarial Networks (GANs) can be used to synthesize microscopy images, including FM images of neurons as described in Chapters 4 and 5. By having a set of real images, GANs can learn and generate structures and noise patterns with a realistic visual appearance. We, therefore, decided to use GAN-generated images as the initial data to pre-train our network and to fine-tune on real data.

6.2.4 The Mammalian Retina

Because the mammalian retina is equipped to handle large amounts of noise in the visual world, its architecture provides additional insight into handling noisy FM images. While a more in-depth analysis of the retina was given in Section 2.2, we will now provide a brief overview of the retinal function to better contextualize our architecture design.

Noisy signals from the visual world are first captured by photoreceptors. The strength of the photoreceptor output is relative to the light in the environment, serving as a form of normalization for the signal output [97]. As a result, small amounts of signal in a dark environment will elicit a greater response than the same signal in a dim environment. The photoreceptor output is then further modified by horizontal cells which perform local contrast normalization [129].

The outputs of the photoreceptors are transferred by bipolar cells to amacrine and ganglion cells [77]. Amacrine cells are responsible for computing a variety of complex functions and therefore have receptive fields that can sample from a select few bipolar cells in a small region to many bipolar cells across large portions of the retina [174, 175].

The outputs of both bipolar and amacrine cells are then fed to ganglion cells, which serve as the final feature layer of the retina. Because bipolar cells make direct contact with ganglion cells, effectively "skipping" amacrine cells, photoreceptor output has a direct pathway to ganglion cells. Together, this architectural organization, along with many more mechanisms not discussed in this thesis, enable the retina to extract relevant information from noisy signals. We use the retina organization as inspiration for the creation of our local denoiser (see Section 6.3.2 and multi-scale fusion (see 6.3.4).

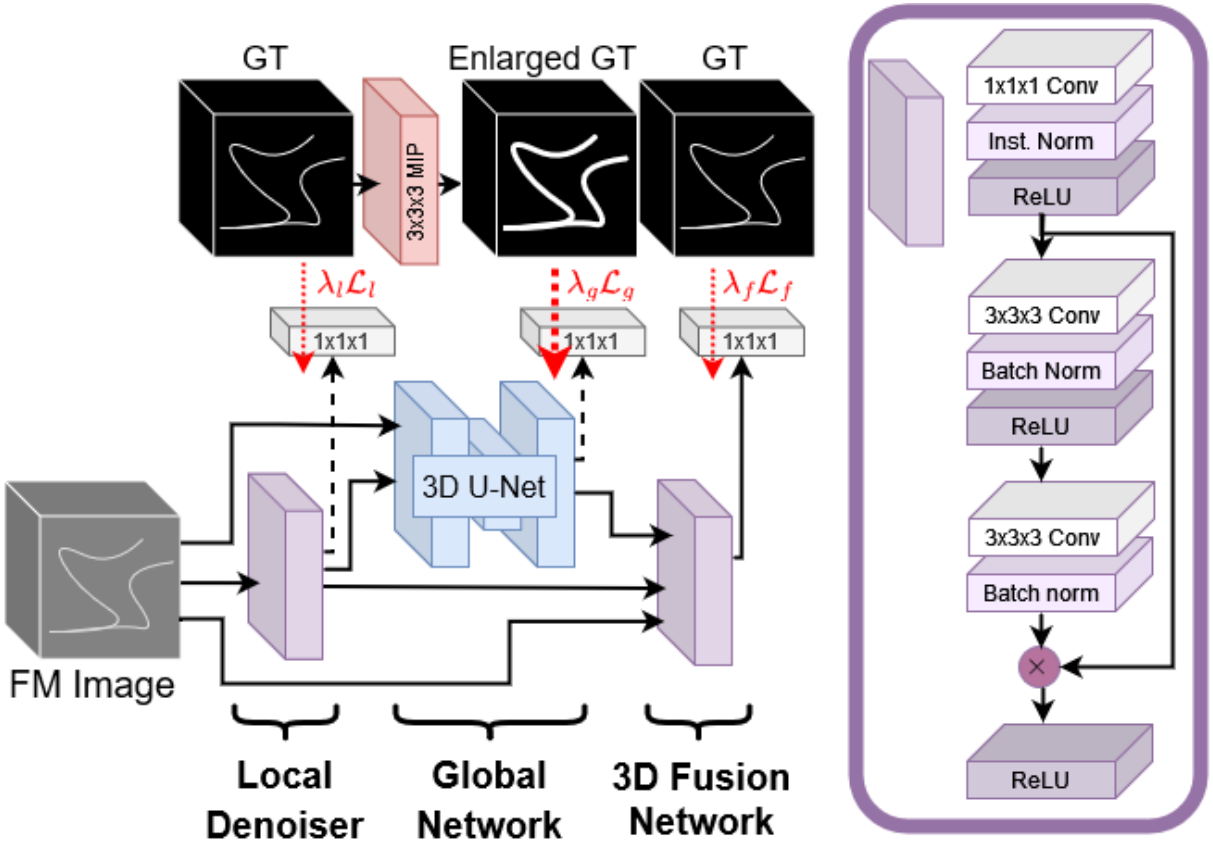


Figure 6.2: An overview of our architecture. The FM image is first passed through a local denoiser that is comprised of a 3D Residual Block with a kernel of 1x1x1 and instance normalization in the first layer (see purple box). Then, the features from the local denoiser are fed to the global network, a 3D U-Net in our case, which is trained on an enlarged ground truth (GT). Both local denoiser and global network features are finally fed to the fusion network. All stages are trained with their own loss function.

6.3 Methodology

6.3.1 Pre-training on GAN-Generated Images

To create synthetic 3D FM training data, we implement the conditional GAN outlined in the previous two chapters which can stylize volumetric reconstructions of neurons as if they were imaged by FM. Instead of producing a predetermined amount of data, we pre-train our network by continuously generating novel images which reduce the probability of over-fitting the segmentation architecture to the synthetic data. More so, we increase the variability of generated images shown to our network by first applying affine geometric transformations - such as rotation, sheer, and flipping - to the volumetric reconstructions. We additionally alter the noise in generated images by randomly sampling laser powers within the ranges used during training.

6.3.2 Local Residual Denoiser

The intensity and variability of noise across FM images limit a CNN's ability for optimal segmentation. What might be noise in a high-SNR FM image can be signal in a low-SNR one. We therefore first pass our FM image through a local denoiser trained to handle various noise patterns as shown in Fig. 6.2.

While normalizing input images between 0 and 1 is standard practice to equalize the power of the input signal, this is insufficient power when significant amounts of noise obscure the signal as with FM images. We, therefore, implement instance normalization in the first layer of the denoiser as this technique can normalize features irrespective of image style [277]. Instance normalization, which is a derivation of batch normalization (see Eq. 2.24), can be summarized as follows:

$$\mathcal{F}_i^{\texttt{instance,c}} = \gamma_i^c \frac{\mathcal{F}_i^c - \mu_i^s}{\sigma_i^s} + \beta_i^c \tag{6.11}$$

where μ_i^s and σ_i^s represent the mean and standard deviation computed for each sample instead of the entire batch. This ensures that every feature can be represented by a Gaussian distribution, effectively normalizing the feature map regardless of signal power in the image, similar to the light adaptation of photoreceptors.

The local denoiser's topology consists of a single residual block followed by a segmentation layer. Residual blocks, which are a precursor to residual learning, have become essential building blocks in deep learning and can be represented as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x} \tag{6.12}$$

Here, $\mathbf{y}$ is the output of the residual block, $\mathcal{F}(\cdot)$ denotes a two-layer network, and $\mathbf{x}$ represents the input. Residual blocks facilitate gradient flow throughout the network, thanks to their skip connections. The presence of skip connections enables the network to learn identity mappings more efficiently, leading to improved model performance and faster convergence. Furthermore, this formulation shares a similar structure with the residual denoiser.

In our residual block, the first convolutional layer has a kernel size of 1x1x1 which additionally enables the final segmentation layer to have access to individual pixels, thereby preserving thin structures. Like horizontal cells, this network then modifies the individual pixel values based on local information.

6.3.3 Global Segmentation with Enlarged Ground Truths

The penultimate feature layer of the denoiser is passed through a deep CNN to incorporate global context into the segmentation pipeline (Fig. 6.2). Because this architecture prioritizes regions of interest over accurate pixel segmentation, we enlarge the ground truths by passing it through a 3x3x3 local MIP, thereby amplifying the importance of thin structures. MIPs are particularly advantageous for fluorescence microscopy (FM) images of neurons as the branching structure of neurons reduces the possibility of a MIP irreversibly blending information. Utilizing a MIP on the ground truth rather than the input additionally ensures that only the signal, and not the noise, is amplified by the network. Our global segmentor is a 3D U-Net due to its known general use case and previous implementation for neuron segmentation. More details about U-Net can be found in Section 2.6.2.2. However, we acknowledge that this architecture could be replaced by better-performing ones as they emerge in literature.

6.3.4 Multi-Scale Fusion Loss

Similar to the ganglion cell layer in the retina, local and global scales are passed to a final CNN with the same architecture build as our local denoiser. Because all three architectures are trained simultaneously, the total loss denoted as $\mathcal{L}_t$ can be summarized as the weighted sum of three loss terms:

$$\mathcal{L}_t = \lambda_l \mathcal{L}_l + \lambda_g \mathcal{L}_g + \lambda_f \mathcal{L}_f \tag{6.13}$$

where $\mathcal{L}_l$, $\mathcal{L}_g$, $\mathcal{L}_f$ are losses from the local denoiser, global segmentor, and fusion model respectively and λ_l, λ_g, λ_f are their corresponding scalar weights. For our

Architecture	$\mathcal{L}_t$	# Params.	3DFM_AC FT Loss (↓) L-SNR	FT Loss (↓) All	Dice Coef. (↑) L-SNR	Dice Coef. (↑) All	IoU (↑) L-SNR	IoU (↑) All
Ours (O)	$\lambda_l\mathcal{L}_l + \lambda_g\mathcal{L}_g + \lambda_f\mathcal{L}_f$	4.3 mil	**0.499**	**0.249**	**0.597**	**0.863**	**0.452**	**0.781**
O - GAN Pretraining	$\lambda_l\mathcal{L}_l + \lambda_g\mathcal{L}_g + \lambda_f\mathcal{L}_f$	4.3 mil	0.593	0.294	0.512	0.856	0.362	0.773
O - Enlarged Ground Truth (EGT)	$\lambda_l\mathcal{L}_l + \lambda_g\mathcal{L}_g + \lambda_f\mathcal{L}_f$	4.3 mil	0.628	0.348	0.505	0.844	0.360	0.759
O - EGT	$\lambda_f\mathcal{L}_f$	4.3 mil	0.590	0.270	0.490	0.838	0.339	0.746
O - Fusion Net. (N_F) - EGT	$\lambda_g\mathcal{L}_g$	4.2 mil	0.786	0.617	0.476	0.848	0.332	0.764
O - Denoiser Net. - N_F - EGT	$\lambda_g\mathcal{L}_g$	4.1 mil	0.732	0.701	0.363	0.537	0.248	0.416
MultiScaleKernel [283]	-	2.8 mil	0.731	0.662	0.436	0.591	0.303	0.466
VNet	-	12.2 mil	0.759	0.691	0.361	0.554	0.250	0.439
3DResUNet	-	8.8 mil	0.731	0.514	0.333	0.578	0.226	0.485
3DUNet	-	4.1 mil	0.864	0.720	0.333	0.532	0.226	0.420
DeepVessel-FCN [270]	-	0.5 mil	0.626	0.280	0.448	0.854	0.302	0.770
Shallow 3D ResBlocks	-	0.1 mil	0.618	0.440	0.420	0.725	0.277	0.622

Table 6.1: An ablation study and a quantitative comparison of our proposed architecture. Performance on lowest SNR (L-SNR) testing images and all testing images contained in the $3DFM_AC$ are reported. L-SNR images are created by using the lowest laser power and frame averaging for each testing image. Metrics computed are the focal Tversky loss (FT loss), Dice coefficient, and intersection of union score (IoU). Details regarding the competing architectures can be found in Sections. 2.7 and 6.2.3.

purposes, we set $\lambda_l = \lambda_g = \lambda_f = 1$ during pre-training on synthetic data and $\lambda_l = \lambda_g = 0.1; \lambda_f = 1$ during fine tuning on real data. To deal with class imbalance, we used a focal Tversky loss which has been shown to be effective [2]. We additionally weight loss values for each voxel so that background voxels carry less importance than voxels closer to the dendrites. To further promote the segmentation of thin objects, we do not compute the loss at the boundaries of the ground truth where correct segmentations are more ambiguous in practice (Fig. 6.3a).

6.4 Implementation

All CNNs start with 32 initial filters and were trained using ADAM optimization with a step size of 0.001, a β_1 of 0.9, and a β_2 of 0.999. Early stopping was implemented after 8 successive epochs of no improvement. We compare performance of several architectures. As a baseline, we implement a shallow CNN comprised of 2 3D residual blocks. We then compare standard encoder-decoder architectures - 3D U-Net [54], 3D ResU-Net [148], V-Net [182] - as they are the backbones to most current neural segmentation architectures. Finally, a state-of-the-art vessel segmentor (DeepVessel-FCN) [270] and neuron segmentation architecture (MultiScaleKernels) are used as upper bounds on performance [283]. Note that all competing architectures were additionally pre-trained on GAN-generated data to ensure fair comparisons.

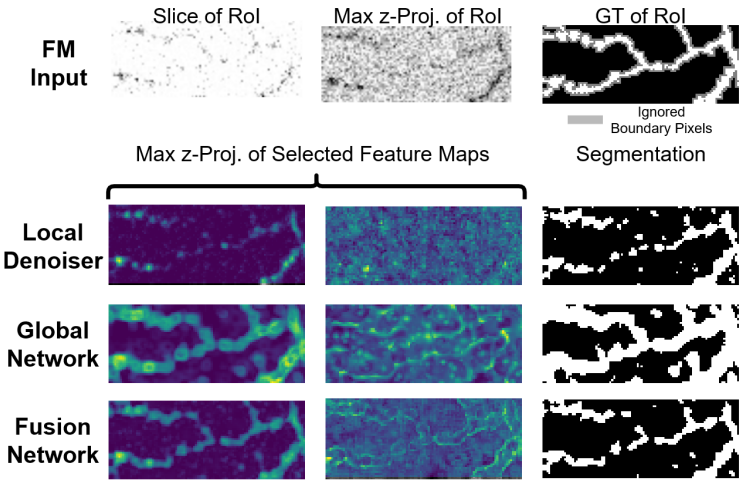
We evaluate our architectures on the $3DFM_AC$ dataset which is described in detail in Section 3.2.3. A total of 12 stacks each with 5 different laser powers and 8 varying frame averaging were used for the training and validating of the GAN and segmentation networks. An additional 8 never-before-seen stacks were used for testing. Stacks had a xy-cross section of 512x512 voxels and the number of slices for stacks varied between 19-34. Patch sizes of 160x160x12 and a batch size of 4 were used for pre-training. We increase the patch size to 256x256x12 using a batch size of 1 for fine-tuning so that more neural content can be seen.

To assess the utility of GAN-generated images, we further compared performance with and without pre-training. Training was done on a variable number of real stacks, 2 real stacks were used for validation, and an additional 10 real stacks were used for testing. Due to the variability of performance on a low number of training samples, results are reported as the mean and standard deviation over 3 runs with randomly chosen training data. It should be noted that the validation and testing data were the same across experiments.

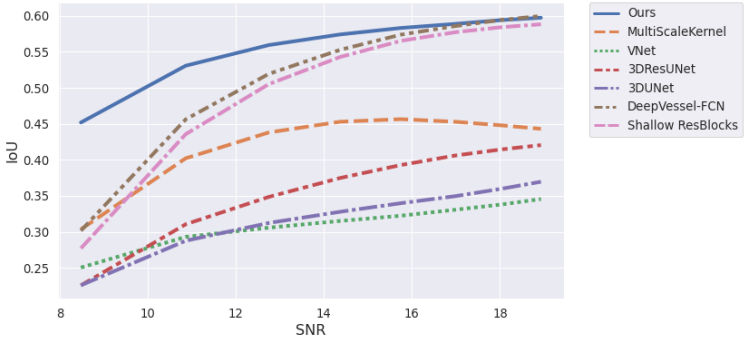
6.5 Results

As we can see from Table 6.1, our proposed architecture is best performing for all metrics computed, increasing the intersection of union (IoU) score by nearly 0.315 when compared to a state-of-the-art neuron segmentation model. Most significant improvements are seen on the lowest SNR images (Table 6.1 and Fig. 6.3b) as the IoU is increased by at least 0.15 compared to all other models. While encoder-decoder based models (middle section in Table 6.1) are standard for neuron segmentation, they performed worse than models based on shallow architectures with no down-sampling (bottom section in Table 6.1). This is because a single architecture mixing multiple scales easily overlooks weak signal expression and results in complete failures (Fig. 6.5).

However, the ablation study (top section in Table 6.1) highlights how this can be overcome by fusing local and global scales as just the inclusion of a local denoiser significantly improves the performance of a stand-alone 3D U-Net. This is further demonstrated in Fig. 6.3a where we can see that most noise is eradicated after the local denoiser and each network successively predicts more accurate neural traces. GAN-generated images for pre-training further increased IoU by 0.09 on the lowest SNR images, proving the utility of incorporating GANs into neural segmentation pipelines. More specifically, as we can see in Fig. 6.4, pre-training on GAN-generated



(a)



(b)

Figure 6.3: a) Maximum z-projections of 2 sampled feature maps and segmentation predictions for each fully trained architecture given the region of interest (RoI) from Fig. 6.1. b) IoU score relative to the images' SNR for all architectures. While our network consistently performs best, the greatest improvements can be found on lower SNR images.

images consistently performed equal to or better than if only trained on real images. The results were most significant on the lowest SNR images which is to be expected as our GAN-generated images helped prevent overfitting. Finally, Figs. 6.1 and 6.5 demonstrate that our contributions better preserve continuity along the dendrite than competing architectures while also avoiding complete failure cases.

6.6 Conclusion

Low-SNR 3D FM images of neurons pose significant challenges to CNNs as the signal is often obscured and sparse. While local scales are necessary to accurately segment thin dendrites, global scales are also necessary as neurons span large spatial volumes.



(a)



(b)

Figure 6.4: IoU performance when pre-training on GAN-generated images compared to no pre-training for a) the lowest SNR images and b) all images.

Instead of using one network to learn both scales which can confuse learned features, we instead fuse local and global scales using 3 separate CNNs thereby preserving information flow. We show that doing so significantly improves segmentation performance on low-SNR FM images compared to standard architectures and those made specifically for neuron segmentation. This is also accomplished with small amounts of data by using GAN-generated training images to pre-train our networks. While we focus mostly on images corrupted by voxel noise, further work should be done to explore segmentation in the presence of other noise sources, like morphological noise.

Additionally, this study emphasizes the importance of developing architectures tailored for a specific imaging domain. By understanding the features present in FM images of neurons, we can customize architectures to more efficiently tackle the task at hand. This is evidenced by our model's significant improvement in performance when

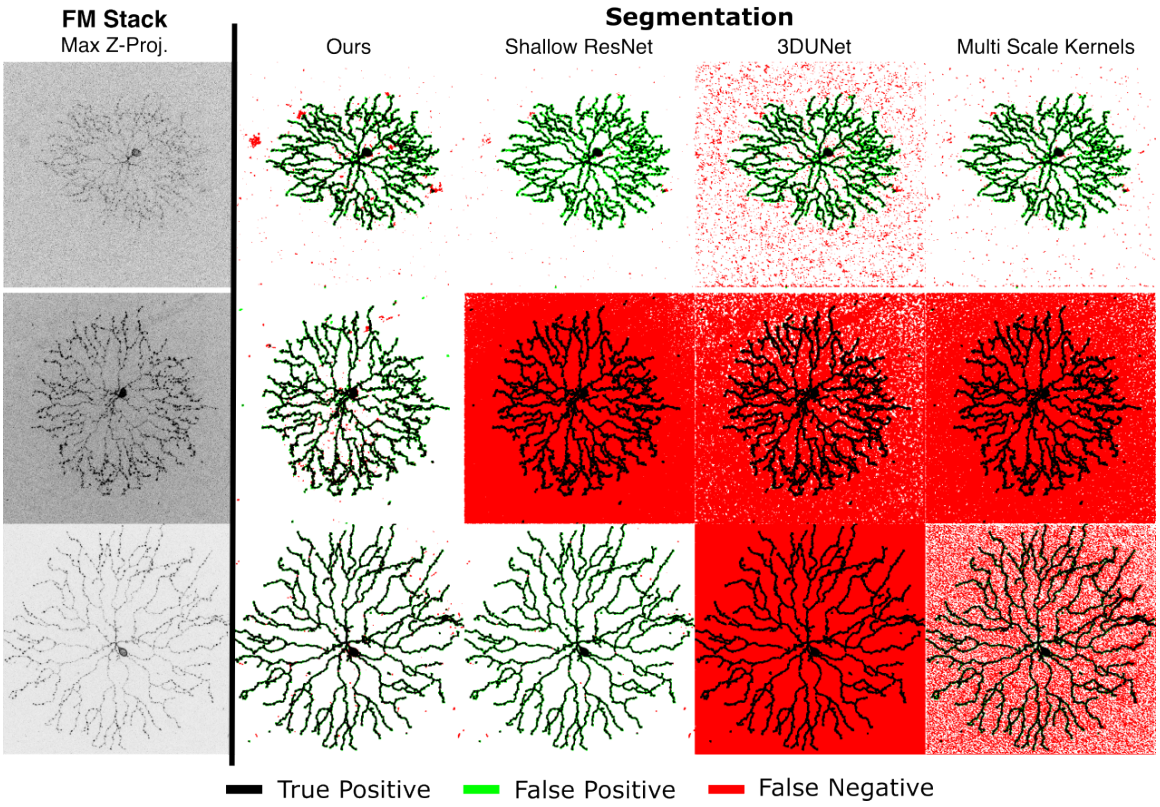


Figure 6.5: Maximum z-projections of 3 sampled FM Stacks (rows) along with segmentation predictions from several architectures (columns). Note segmentation predictions that are mostly red signify a failure to identify any neuron structure.

compared to generic deep learning architectures like U-Net. Finally, we showcase the advantages of pre-training on GAN-generated images, as it enhances performance in nearly every conducted experiment. The most significant improvement is observed on the noisiest images, where the network is more prone to overfitting due to the difficulty in accurately segmenting the image. This result is important as it demonstrates that performance is not compromised when the amount of labeled data is reduced, thereby further validating the utility of the pipeline produced in this thesis.

Chapter 7

Conclusion

This thesis aims to advance neuroscience through the introduction of deep learning-based techniques suited for experimental FM image data. The proposed pipeline we have described in Chapters 4-6 is designed to be easy-to-use and robust so that individual laboratories can analyze data from a wide variety of imaging conditions without requiring changes to the existing workflow of researchers. To start, we applied this approach to the segmentation of individual retinal neurons. We demonstrated how our pipeline achieves state-of-the-art performance in analyzing extremely noisy FM images, increasing unsupervised IoU segmentation performance by upwards of 40 percentage points (Fig. 5.3) and supervised segmentation by nearly 15 percentage points (Fig. 6.1). This is additionally accomplished while also significantly reducing the amount of manual labor traditionally required in these processes. As a result, we demonstrate how volumetric reconstructions of neurons can be utilized to train deep learning models. The full reconstruction volume allows models to learn global neuron morphologies, while the nanoscopic resolution of the reconstruction allows for more accurate local segmentation despite signal obfuscation in the FM image. Although segmentation of neurons served as a proof of concept for our pipeline, it can be adapted for other FM image analysis tasks such as automated localization of neural structures and neuron classification.

Specifically, in Chapter 4, we first proposed five unique datasets intended to foster deep learning advancements in experimental neuroscience. Two of these datasets (2DSyn_BC, 3DFM_GC) are synthetic to provide complete control over the image generation process for more effective evaluation of deep learning architectures. The remaining three datasets (3DFM_AC, 3DFM_BCPop, 3DFM_overAC) consist of real FM images. The first dataset, 3DFM_AC, contains images of individual SACs imaged at 5 different laser powers and 8 differing frame averaging. 3DFM_BCPop is the second proposed real FM dataset consisting of bipolar cell populations imaged with

8 different frame averaging. The final dataset proposed is the 3D FM data which features pairs of overlapping SACs. Each dataset poses unique challenges in experimental neuroscience from images corrupted with extreme amounts of pixel noise to images containing large amounts of morphological overlap. All datasets come with paired ground truths and are either already available or are in the process of being made available to the public. While more datasets are needed to thoroughly evaluate deep learning architectures on a variety of experimental neuroscience conditions, these datasets serve as foundational datasets to inspire the collection of more complex, yet controlled, datasets that are required for the field to progress.

The proposed deep learning pipeline employs a GAN to generate realistic 3D FM images of neurons. State-of-the-art FM image generation is achieved by first learning unique FM imaging characteristics from a set of images collected during experimentation through trainable Gram Matrix computations in the GAN discriminator. Subsequently, we stylize previously reconstructed neurons by sequentially generating 2D slices that account for 3D context. This unpaired mesh-to-image translation methodology allows us to learn a synthetic microscope and acquire realistic FM images of neurons with control over the image content and imaging configurations. Because the GAN generator is conditioned on volumetric reconstructions of neurons, generated images automatically have corresponding "gold standard" ground truths which can be used to train subsequent deep learning networks for image analysis, thereby significantly reducing the amount of manual annotation required to employ deep learning.

A novel evaluation protocol was designed to evaluate GANs on FM images of neurons. This protocol focused on 6 different criteria: neural content, local texture, voxel noise, utility, and expert opinion. Because these metrics did not require deep learning architectures, the results from the evaluation protocol are more interpretable and better suited for FM images. The results from our evaluation demonstrated the ability of the architectural advances in our GAN to better model FM imaging characteristics compared to two competing state-of-the-art image translation architectures. The most important result of our evaluation protocol demonstrated the ability of our GAN-generated images alone to be used to train a network for neuron segmentation from real FM images. While our evaluation protocol is tailored for the FM imaging domain, it still contains biases that need to be addressed. The development of standardized and generalizable GAN metrics is needed for biomedical and FM images so that GAN performance can be reliably evaluated.

Finally, in Chapter 6, we introduced a novel three-stage CNN-based model capable of more accurately segmenting neurons from low SNR 3D FM images than several

state-of-the-art neuron segmentation model. By fusing local and global scales, this method addresses partial volume effects and extreme class imbalance that hinders traditional CNNs when analyzing 3D FM images of neurons. While a 3D UNet is utilized to learn global information, more advanced techniques can be employed to better learn morphological priors. Transformers, for example, have been shown to effectively learn local and global information, but are heavily dependent on large amounts of data. Because we can generate as much realistic data as we want, these advanced techniques can now be applied to experimental neuroscience context and should be thoroughly explored. The code for the mesh-to-image translation pipeline is accessible to the public through `https://github.com/MihaelCudic/BioSPADE.git`. Instructions on how to download and utilize the pipeline are outlined within the GitHub repository and in comments throughout the code. To implement the pipeline, all that is required are experimental FM images, a set of corresponding volumetric reconstructions of the content in the experimental images, and an updated options file to describe the training details (i.e. project name, path locations to image file, dimensions of images, etc.). Furthermore, the code was created so that it can be employed by those with expertise in computer science and deep learning. Although architecture and training configurations are largely conserved across implementations on four diverse datasets, users with expertise have the added ability to adapt the pipeline to their specific requirements to help obtain optimal results. This control is accessed through the options file where each configuration detail is clearly described. The implementation for our novel segmentation architecture proposed in Chapter 6 will also be made available such that it can be easily ported to the existing mesh-to-image translation repository. All FM images and corresponding ground truths are, or will be, made available to the public. Currently, four out of five datasets can be found here: `https://data.mendeley.com/datasets/f6kk4364p4/`. The diversity of datasets used throughout this thesis demonstrates the proposed pipeline's ability to generalize to a variety of FM images. While the morphologies of neurons vary throughout the brain and across species, the differences in the global semantics of FM images are modelled by the volumetric reconstructions of those neurons so that our translation pipeline is tasked only learn more local and general FM textures.

Chapter 8

Future Works

A significant question that remains unresolved in this thesis is "How much unlabeled and labeled data does this pipeline require for successful execution?" In general, the answer to this question depends on the complexity of the task trying to be automated, with more complex tasks requiring more data. However, additional experimentation, similar to that of Fig. 6.4, on differing datasets can provide more intuition into this question. Such comprehensive analysis may help neuroscientists, particularly those without a background in computation, in understanding what is required to automate their analysis.

Another strong requirement of this pipeline is volumetric reconstructions of neurons from EM blocks which are the basis of our generated 3D FM images. Although this resource has been growing rapidly over the last few years, its collection has largely been restricted to a few brain regions. Even within these image areas, not all neurons are reconstructed. For instance, the seminal retina EM block used throughout this thesis does not contain volumetric reconstructions of Mueller glia [36], a neuron largely implicated in regulating the inflammatory response of the neural tissue. Additionally, many EM blocks contain skeleton traces as they are easier to collect than volumetric reconstructions. Therefore, further research is needed to extend this pipeline to other neuron representations and in extreme cases where no representations of the neurons exist. This would require the GAN to learn morphological priors so that it can add realistic branching patterns or local content like varicosities. If a more generalized solution exists, then it would pave the way for a much wider application and impact of this pipeline to the neuroscience community.

Additionally, the scope of the analysis in this thesis is currently limited to static images of individual neurons; however, the work could also extend to real-time analysis of neural network activity. Genetically encoded fluorescent indicators have made it possible to record individual dendritic and somatic physiological responses from

hundreds of neurons at once with fluorescent microscopy (FM). However, properly analyzing population activity requires additional knowledge of the underlying circuitry so that each response can be paired with its parent neuron. Previous work has therefore focused on pairing recordings with reconstructed circuits from 3D EM blocks [36]. In practice, the time and resources necessary to prepare, image, and process such large EM volumes remain too great to make this approach generally feasible [35, 36]. An alternative would be to pair recordings to reconstructed circuits from FM data. Although the synaptic connectivity of the circuit will be lost due to resolution limitations of FM, this approach would allow for real-time population analysis using only one imaging modality. The main limitation to using FM data for circuit reconstruction, however, is the lack of sufficient ground truth data of fluorescent neurons and neural populations. Because any combination of neuron pairs can be used to condition our FM image generator for a variety of imaging conditions, we can now train a deep learning model for the segmentation of overlapping neurons with sufficient data.

More generally, by utilizing 3D models of biological content as input to GANs and ground truths for segmentation tasks, we should be able to develop segmentation models with significantly less annotated data. For example, instead of collecting a large MRI dataset for brain tumor segmentation, 3D models of the brain can be consolidated and passed to the pipeline discussed in this thesis. Like the advantages demonstrated for FM image generation and segmentation, our pipeline applied to MRI images would allow GANs to model local textures which could potentially result in more realistic GAN-generated MRI datasets and thereby better unsupervised pretraining on subsequent brain tumor segmentation models. As a result, this pipeline could be used to help maximize deep learning full potential on a variety of biomedical imaging tasks.

Appendix A

Expert Opinion

Subject: Expert Opinion Test

Thank you for volunteering to evaluate my machine learning algorithm!

You will be given 15 sets of 5 6x128x128 (~.6umx80umx80um) fluorescent microscopy (FM) images of Starburst Amacrine Cell (SAC) dendrites. Each set contains a real FM image (the left most image in the panel) which will serve as a reference to gauge image realism. The remaining 4 patches will be of simulated FM SAC dendrites using various machine learning algorithms.

So here is the task:

1. Download and unzip the zip file (ExpertOpinionTest.zip) attached to this email. You should see a tiff file (test.tif) and an excel file (Answer_Sheet.xlsx).
2. Open ImageJ or Fiji (or any other software used to scroll through image stacks). If you do not have this software, you can download it here: https://imagej.net/Fiji/Downloads
3. Open the Answer_Sheet.xlsx file
4. Write your first and last name
5. Open test.tif in the software and zoom in to the first set (the topmost set) so that the images are as large as possible while still being able to see all 5 images in the set at once.
6. Scroll through the image slices for each image in the set.
7. Using the left most image as reference (labelled 'real'), rank the remaining images (to the right of the gray line) according to how realistic they appear (I will help define this later). Let 1 represent the most realistic image and 4 the least. Ties are not allowed. Report these ranks in the excel file under the correct set number (i.e. row number) and image ID (i.e. columns labelled 'A', 'B', 'C', and 'D').
8. Do the same for the other 14 image sets.
9. Save the excel file and return it under the same file name.

What is a Real Image:

Determining the realism can be tricky. The morphology of the dendrites in each synthetic stack are the same. What you should focus on instead of the morphology is the texture of the image. This includes the **point spread function**, **background noise**, **intensity across the dendrites**, the **shape of varicosities**, etc. The primary focus should be the pixels around the dendrites. The secondary focus is the null space. REMEMBER, we are not looking for the best quality image, but rather the most real (which includes the noise) based on the reference real image.

FINAL NOTES:

Don't over think it, don't use tool boxes, go with your gut instinct! It would be best to have the results on Thursday, the 10th of March.

If you have any questions, please don't hesitate to ask.

Thank you again,
Mihael Cudic

Figure A.1: Email sent out to experts to evaluate the "realism" of GAN-generated images. They were instructed to order 15 sets of 4 6x128x128 GAN-generated images by most similar to least similar to an reference real FM patch. They were also given a brief description of which features they should use to evaluate FM image realism.

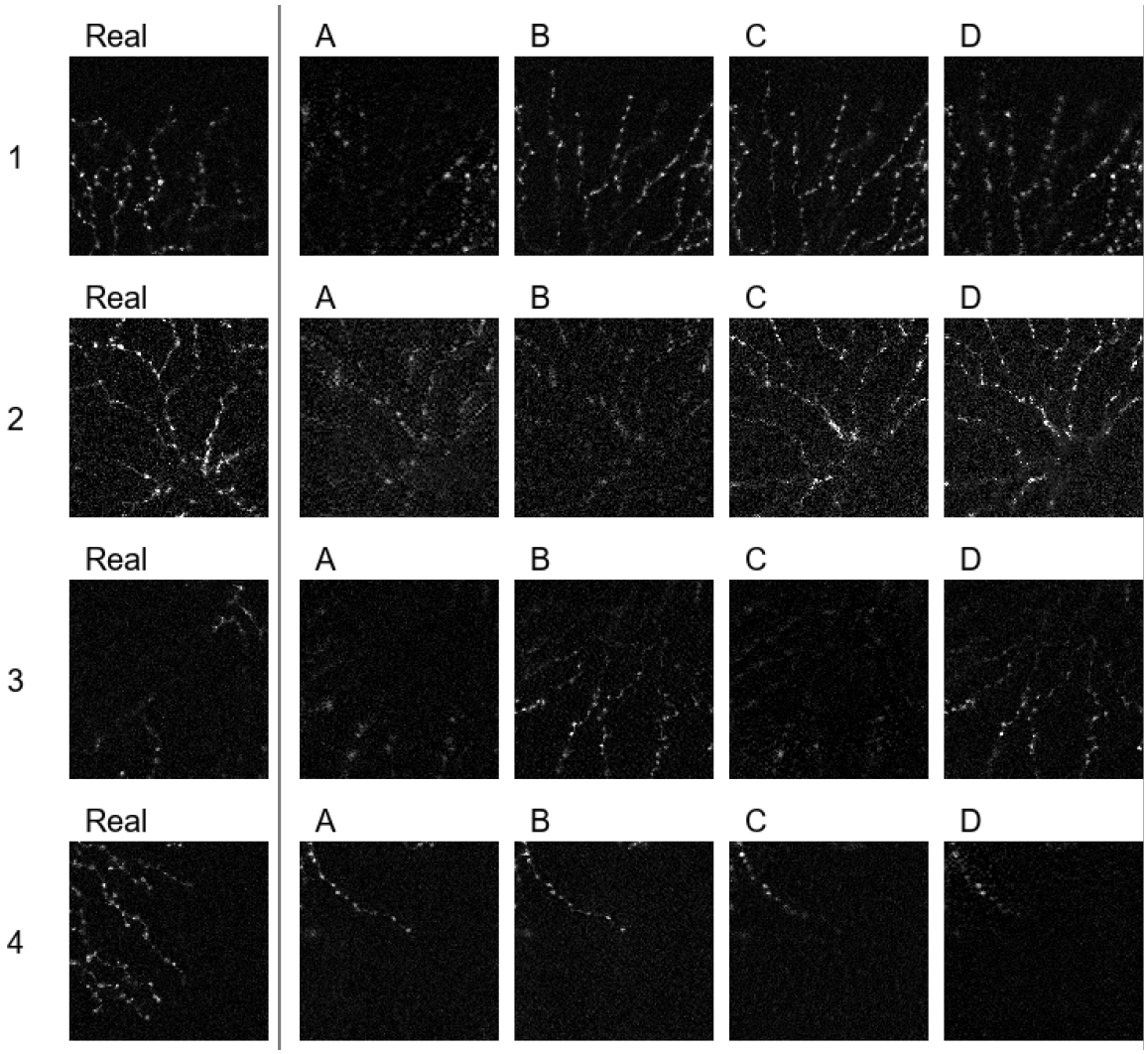


Figure A.2: Example 4 panels from the expert opinion test. Note that this only shows one slice and experts had the ability to scroll through all 6 slices to better evaluate 3D textures.

Bibliography

[1] Takaya Abe and Toshihiko Fujimori. Reporter mouse lines for fluorescence imaging. *Development, growth & differentiation*, 55(4):390–405, 2013.

[2] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, pages 683–687. IEEE, 2019.

[3] Ludovica Acciai, Paolo Soda, and Giulio Iannello. Automated neuron tracing methods: an updated account. *Neuroinformatics*, 14:353–367, 2016.

[4] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.

[5] Salma Alqazzaz, Xianfang Sun, Xin Yang, and Len Nokes. Automated brain tumor segmentation on multi-modal mr image using segnet. *Computational Visual Media*, 5:209–219, 2019.

[6] H Andersson, T Baechi, M Hoechl, and C Richter. Autofluorescence of living cells. *Journal of microscopy*, 191(Pt 1):1–7, 1998.

[7] Noah Apthorpe, Alexander Riordan, Robert Aguilar, Jan Homann, Yi Gu, David Tank, and H Sebastian Seung. Automatic neuron detection in calcium imaging data using convolutional networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[9] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. Medgan: Medical image translation using gans. *arXiv preprint arXiv:1806.06397*, 2018.

[10] Vadim Y Arshavsky, Trevor D Lamb, and Edward N Pugh Jr. G proteins and phototransduction. *Annual review of physiology*, 64(1):153–187, 2002.

[11] Sabrina Asteriti, Claudia Gargini, and Lorenzo Cangiano. Mouse rods signal through gap junctions with cones. *Elife*, 3:e01386, 2014.

[12] Stephen R Aylward and Elizabeth Bullitt. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE transactions on medical imaging*, 21(2):61–75, 2002.

[13] Tom Baden, Thomas Euler, and Philipp Berens. Understanding the retinal basis of vision across species. *Nature Reviews Neuroscience*, 21(1):5–20, 2020.

[14] Tom Baden, Timm Schubert, Le Chang, Tao Wei, Mariana Zaichuk, Bernd Wissinger, and Thomas Euler. A tale of two retinal domains: near-optimal sampling of achromatic contrasts in natural scenes through asymmetric photoreceptor distribution. *Neuron*, 80(5):1206–1217, 2013.

[15] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[16] Piotr Baniukiewicz, E Josiah Lutton, Sharon Collier, and Till Bretschneider. Generative adversarial networks for augmenting training data of microscopic cell images. *Frontiers in Computer Science*, page 10, 2019.

[17] HB Barlow and William R Levick. The mechanism of directionally selective units in rabbit's retina. *The Journal of physiology*, 178(3):477, 1965.

[18] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[19] Andrew Beers, James Brown, Ken Chang, J Peter Campbell, Susan Ostmo, Michael F Chiang, and Jayashree Kalpathy-Cramer. High-resolution medical image synthesis using progressively grown generative adversarial networks. *arXiv preprint arXiv:1805.03144*, 2018.

[20] Christian Behrens, Timm Schubert, Silke Haverkamp, Thomas Euler, and Philipp Berens. Connectivity map of bipolar cells and photoreceptors in the mouse retina. *Elife*, 5:e20041, 2016.

[21] Valentina Bellemo, Philippe Burlina, Liu Yong, Tien Yin Wong, and Daniel Shu Wei Ting. Generative adversarial networks (gans) for retinal fundus image synthesis. In *Asian Conference on Computer Vision*, pages 289–302. Springer, 2018.

[22] Fethallah Benmansour and Laurent D Cohen. Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision*, 92:192–210, 2011.

[23] Kathy Benninger, Greg Hood, Derek Simmel, Luke Tuite, Arthur Wetzel, Alexander Ropelewski, Simon Watkins, Alan Watson, and Marcel Bruchez. Cyberinfrastructure of a multi-petabyte microscopy resource for neuroscience research. In *Practice and Experience in Advanced Research Computing*, pages 1–7. 2020.

[24] P Berens, L Theis, J Stone, N Sofroniew, A Tolias, M Bethge, and J Freeman. Standardizing and benchmarking data analysis for calcium imaging. In *Computational and Systems Neuroscience Meeting (COSYNE 2017)*, pages 66–67, 2017.

[25] Rui Bernardes, Pedro Serranho, and Conceição Lobo. Digital ocular fundus imaging: a review. *Ophthalmologica*, 226(4):161–181, 2011.

[26] Eric Betzig, George H Patterson, Rachid Sougrat, O Wolf Lindwasser, Scott Olenych, Juan S Bonifacino, Michael W Davidson, Jennifer Lippincott-Schwartz, and Harald F Hess. Imaging intracellular fluorescent proteins at nanometer resolution. *science*, 313(5793):1642–1645, 2006.

[27] Alceu Bissoto, Fábio Perez, Eduardo Valle, and Sandra Avila. Skin lesion synthesis with generative adversarial networks. In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis: First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings 5*, pages 294–302. Springer, 2018.

[28] Adam Bleckert, Gregory W Schwartz, Maxwell H Turner, Fred Rieke, and Rachel OL Wong. Visual space is represented by nonmatching topographies

of distinct mouse retinal ganglion cell types. *Current Biology*, 24(3):310–315, 2014.

[29] Ashish Bora, Eric Price, and Alexandros G Dimakis. Ambientgan: Generative models from lossy measurements. In *International Conference on Learning Representations*, 2018.

[30] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[31] James K Bowmaker and HJk Dartnall. Visual pigments of rods and cones in a human retina. *The Journal of physiology*, 298(1):501–511, 1980.

[32] Savile Bradbury, Brian Bracegirdle, Savile Bradbury, and B Bracegirdle. *Introduction to light microscopy*. Bios Scientific Publishers Oxford, 1998.

[33] Prashant Brahmbhatt and Siddhi Nath Rajan. Skin lesion segmentation using segnet with binary crossentropy. In *Proceedings of the International Conference on Artificial Intelligence and Speech Technology (AIST2019), Delhi, India*, pages 14–15, 2019.

[34] Kevin L Briggman and Davi D Bock. Volume electron microscopy for neuronal circuit reconstruction. *Current opinion in neurobiology*, 22(1):154–161, 2012.

[35] Kevin L Briggman and Winfried Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology*, 16(5):562–570, 2006.

[36] Kevin L Briggman, Moritz Helmstaedter, and Winfried Denk. Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471(7337):183, 2011.

[37] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[38] Kerry M Brown, Germán Barrionuevo, Alison J Canty, Vincenzo De Paola, Judith A Hirsch, Gregory SXE Jefferis, Ju Lu, Marjolein Snippe, Izumi Sugihara, and Giorgio A Ascoli. The diadem data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9:143–157, 2011.

[39] Kerry M Brown, Germán Barrionuevo, Alison J Canty, Vincenzo De Paola, Judith A Hirsch, Gregory SXE Jefferis, Ju Lu, Marjolein Snippe, Izumi Sugihara, and Giorgio A Ascoli. The diadem data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9(2):143–157, 2011.

[40] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

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

[42] Camille A Chapot, Christian Behrens, Luke E Rogerson, Tom Baden, Sinziana Pop, Philipp Berens, Thomas Euler, and Timm Schubert. Local signals in mouse horizontal cell dendrites. *Current Biology*, 27(23):3603–3615, 2017.

[43] Agisilaos Chartsias, Thomas Joyce, Mario Valerio Giuffrida, and Sotirios A Tsaftaris. Multimodal mr synthesis via modality-invariant latent representation. *IEEE transactions on medical imaging*, 37(3):803–814, 2017.

[44] Chang Chen, Zhiwei Xiong, Xinmei Tian, and Feng Wu. Deep boosting for image denoising. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018.

[45] Hanbo Chen, Hang Xiao, Tianming Liu, and Hanchuan Peng. Smarttracing: self-learning-based neuron reconstruction. *Brain informatics*, 2(3):135–144, 2015.

[46] Tsai-Wen Chen, Trevor J Wardill, Yi Sun, Stefan R Pulver, Sabine L Renninger, Amy Baohan, Eric R Schreiter, Rex A Kerr, Michael B Orger, Vivek Jayaraman, et al. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300, 2013.

[47] Weixun Chen, Min Liu, Hao Du, Miroslav Radojević, Yaonan Wang, and Erik Meijering. Deep-learning-based automated neuron reconstruction from 3d microscopy images using synthetic training images. *IEEE Transactions on Medical Imaging*, 41(5):1031–1042, 2021.

[48] O Chernyavskiy, L Kubínová, and XW Mao. Analysis of point spread function degradation in thick tissues. *Microscopy and Microanalysis*, 16(S2):286–287, 2010.

[49] Ann-Shyn Chiang, Chih-Yung Lin, Chao-Chun Chuang, Hsiu-Ming Chang, Chang-Huain Hsieh, Chang-Wei Yeh, Chi-Tin Shih, Jian-Jheng Wu, Guo-Tzau Wang, Yung-Chang Chen, et al. Three-dimensional reconstruction of brain-wide wiring networks in drosophila at single-cell resolution. *Current biology*, 21(1):1–11, 2011.

[50] Hyungjoo Cho, Sungbin Lim, Gunho Choi, and Hyunseok Min. Neural stain-style transfer learning using gan for histopathological images. *arXiv preprint arXiv:1710.08543*, 2017.

[51] Yong-Joon Choi and Kazuaki Sawada. Physical sensors: Fluorescence sensors. In Roger Narayan, editor, *Encyclopedia of Sensors and Biosensors (First Edition)*, pages 1–19. Elsevier, Oxford, first edition edition, 2023.

[52] Paarth Chothani, Vivek Mehta, and Armen Stepanyants. Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, 9:263–278, 2011.

[53] Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. *arXiv preprint arXiv:2002.04185*, 2020.

[54] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016.

[55] ADOLPH I Cohen. Rods and cones. *Physiology of photoreceptor organs*, 7:2, 1972.

[56] AR Cohen, B Roysam, and JN Turner. Automated tracing and volume measurements of neurons from 3-d confocal fluorescence microscopy data. *Journal of microscopy*, 173(2):103–114, 1994.

[57] J Coombs, D Van Der List, G-Y Wang, and LM Chalupa. Morphological properties of mouse retinal ganglion cells. *Neuroscience*, 140(1):123–136, 2006.

[58] Mihael Cudic, Jeffrey S Diamond, and J Alison Noble. Unpaired mesh-to-image translation for 3d fluorescent microscopy images of neurons. *Medical Image Analysis*, 86:102768, 2023.

[59] Nicolás Cuenca, I Pinilla, Yves Sauvé, Bin Lu, Shaomei Wang, and Raymond D Lund. Regressive and reactive changes in the connectivity patterns of rod and cone pathways of p23h transgenic rat retina. *Neuroscience*, 127(2):301–317, 2004.

[60] Christine A Curcio and Kimberly A Allen. Topography of ganglion cells in human retina. *Journal of comparative Neurology*, 300(1):5–25, 1990.

[61] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[62] Deniz Dalkara, Leah C Byrne, Ryan R Klimczak, Meike Visel, Lu Yin, William H Merigan, John G Flannery, and David V Schaffer. In vivo–directed evolution of a new adeno-associated virus for therapeutic outer retinal gene delivery from the vitreous. *Science translational medicine*, 5(189):189ra76–189ra76, 2013.

[63] Salman UH Dar, Mahmut Yurt, Levent Karacan, Aykut Erdem, Erkut Erdem, and Tolga Cukur. Image synthesis in multi-contrast mri with conditional generative adversarial networks. *IEEE transactions on medical imaging*, 38(10):2375–2388, 2019.

[64] R De Martin, M Raidl, E Hofer, and BR Binder. Adenovirus-mediated expression of green fluorescent protein. *Gene therapy*, 4(5):493–495, 1997.

[65] Winfried Denk and Heinz Horstmann. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS biology*, 2(11):e329, 2004.

[66] Onkar S Dhande, Benjamin K Stafford, Jung-Hwan A Lim, and Andrew D Huberman. Contributions of retinal ganglion cells to subcortical visual processing and behaviors. *Annual review of vision science*, 1:291–328, 2015.

[67] Jeffrey S Diamond. Inhibitory interneurons in the retina: types, circuitry, and function. *Annual review of vision science*, 3:1–24, 2017.

[68] Alberto Diaspro, Giuseppe Chirico, Cesare Usai, Paola Ramoino, and Jurek Dobrucki. Photobleaching. *Handbook of biological confocal microscopy*, pages 690–702, 2006.

[69] William F Dietrich, Joyce Miller, Robert Steen, Mark A Merchant, Deborah Damron-Boles, Zeeshan Husain, Robert Dredge, Mark J Daly, Kimberly A Ingalls, Tara J O'Connor, et al. A comprehensive genetic map of the mouse genome. *Nature*, 380(6570):149–152, 1996.

[70] Keisuke Doman, Takaaki Konishi, and Yoshito Mekada. Lesion image synthesis using dcgans for metastatic liver cancer detection. *Deep Learning in Medical Image Analysis: Challenges and Applications*, pages 95–106, 2020.

[71] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

[72] Michael N Economo, Nathan G Clack, Luke D Lavis, Charles R Gerfen, Karel Svoboda, Eugene W Myers, and Jayaram Chandrashekar. A platform for brain-wide imaging and reconstruction of individual neurons. *elife*, 5:e10566, 2016.

[73] Hajar Emami, Ming Dong, Siamak P Nejad-Davarani, and Carri K Glide-Hurst. Generating synthetic cts from magnetic resonance images using generative adversarial networks. *Medical physics*, 45(8):3627–3636, 2018.

[74] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings, 2010.

[75] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[76] Thomas Euler, Peter B Detwiler, and Winfried Denk. Directionally selective calcium signals in dendrites of starburst amacrine cells. *Nature*, 418(6900):845–852, 2002.

[77] Thomas Euler, Silke Haverkamp, Timm Schubert, and Tom Baden. Retinal bipolar cells: elementary building blocks of vision. *Nature Reviews Neuroscience*, 15(8):507–519, 2014.

[78] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[79] Dirk Feldmeyer, Michael Brecht, Fritjof Helmchen, Carl CH Petersen, James FA Poulet, Jochen F Staiger, Heiko J Luhmann, and Cornelius Schwarz. Barrel cortex function. *Progress in neurobiology*, 103:3–27, 2013.

[80] Linqing Feng, Ting Zhao, and Jinhyun Kim. neutube 1.0: a new design for efficient neuron reconstruction software based on the swc format. *eneuro*, 2(1), 2015.

[81] JB Findlay and DJ Pappin. The opsin family of proteins. *Biochemical Journal*, 238(3):625, 1986.

[82] R Foord, R Jones, CJ Oliver, and ER Pike. The use of photomultiplier tubes for photon counting. *Applied optics*, 8(10):1975–1989, 1969.

[83] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI'98: First International Conference Cambridge, MA, USA, October 11–13, 1998 Proceedings 1*, pages 130–137. Springer, 1998.

[84] Katrin Franke, Philipp Berens, Timm Schubert, Matthias Bethge, Thomas Euler, and Tom Baden. Inhibition decorrelates visual feature representations in the inner retina. *Nature*, 542(7642):439–444, 2017.

[85] M Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[86] Weilin Fu, Katharina Breininger, Roman Schaffert, Nishant Ravikumar, Tobias Würfl, Jim Fujimoto, Eric Moult, and Andreas Maier. Frangi-net. In *Bildverarbeitung für die Medizin 2018: Algorithmen-Systeme-Anwendungen. Proceedings des Workshops vom 11. bis 13. März 2018 in Erlangen*, pages 341–346. Springer, 2018.

[87] Yingbin Fu and King-Wai Yau. Phototransduction in mouse rods and cones. *Pflügers Archiv-European Journal of Physiology*, 454:805–819, 2007.

[88] Adrian Galdran, André Anjos, José Dolz, Hadi Chakor, Hervé Lombaert, and Ismail Ben Ayed. State-of-the-art retinal vessel segmentation with minimalistic models. *Scientific Reports*, 12(1):6174, 2022.

[89] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[90] Wilson S Geisler. Sequential ideal-observer analysis of visual discriminations. *Psychological review*, 96(2):267, 1989.

[91] Maryam Gholizadeh-Ansari, Javad Alirezaie, and Paul Babyn. Deep learning for low-dose ct denoising using perceptual loss and edge detection layer. *Journal of digital imaging*, 33:504–515, 2020.

[92] Kaushik Ghosh and Gregory D Van Duyne. Cre–loxp biochemistry. *Methods*, 28(3):374–383, 2002.

[93] Krishna K Ghosh, Sascha Bujan, Silke Haverkamp, Andreas Feigenspan, and Heinz Wässle. Types of bipolar cells in the mouse retina. *Journal of Comparative Neurology*, 469(1):70–82, 2004.

[94] Sreya Ghosh and Chrysanthe Preza. Fluorescence microscopy point spread function model accounting for aberrations due to refractive index variability within a specimen. *Journal of biomedical optics*, 20(7):075003, 2015.

[95] Peter Goldsborough, Nick Pawlowski, Juan C Caicedo, Shantanu Singh, and Anne Carpenter. Cytogan: generative modeling of cell images. *bioRxiv*, page 227645, 2017.

[96] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[97] Viktor I Govardovskii, Peter D Calvert, and Vadim Y Arshavsky. Photoreceptor light adaptation: untangling desensitization and sensitization. *The Journal of General Physiology*, 116(6):791–794, 2000.

[98] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[99] Christine Grienberger and Arthur Konnerth. Imaging calcium in neurons. *Neuron*, 73(5):862–885, 2012.

[100] Wilfried Haensch, Tayfun Gokmen, and Ruchir Puri. The next generation of deep learning hardware: Analog computing. *Proceedings of the IEEE*, 107(1):108–122, 2018.

[101] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12, 2009.

[102] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, (6):610–621, 1973.

[103] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[104] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019.

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

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

[107] Moritz Helmstaedter, Kevin L Briggman, Srinivas C Turaga, Viren Jain, H Sebastian Seung, and Winfried Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168, 2013.

[108] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[109] David Grant Colburn Hildebrand, Marcelo Cicconet, Russel Miguel Torres, Woohyuk Choi, Tran Minh Quan, Jungmin Moon, Arthur Willis Wetzel, Andrew Scott Champion, Brett Jesse Graham, Owen Randlett, et al. Whole-brain serial-section electron microscopy in larval zebrafish. *Nature*, 545(7654):345–349, 2017.

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

[111] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, et al. nucleaizer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Systems*, 10(5):453–458, 2020.

[112] Sungmin Hong, Razvan Marinescu, Adrian V Dalca, Anna K Bonkhoff, Martin Bretzner, Natalia S Rost, and Polina Golland. 3d-stylegan: A style-based generative adversarial network for generative modeling of three-dimensional medical images. In *Deep Generative Models, and Data Augmentation, Labelling, and Imperfections: First Workshop, DGM4MICCAI 2021, and First Workshop, DALI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 1*, pages 24–34. Springer, 2021.

[113] Xiaojun Hu, Weijian Luo, Jiliang Hu, Sheng Guo, Weilin Huang, Matthew R Scott, Roland Wiest, Michael Dahlweid, and Mauricio Reyes. Brain segnet: 3d local refinement network for brain lesion segmentation. *BMC medical imaging*, 20:1–10, 2020.

[114] Ruth Hubbard and Allen Kropf. The action of light on rhodopsin. *Proceedings of the National Academy of Sciences*, 44(2):130–139, 1958.

[115] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[116] Alice C Hughes, Michael C Orr, Keping Ma, Mark J Costello, John Waller, Pieter Provoost, Qinmin Yang, Chaodong Zhu, and Huijie Qiao. Sampling biases shape our view of the natural world. *Ecography*, 44(9):1259–1269, 2021.

[117] Howard C Hughes, George Nozawa, and Frederick Kitterle. Global precedence, spatial frequency channels, and the statistics of natural images. *Journal of cognitive neuroscience*, 8(3):197–230, 1996.

[118] Oscar Inzunza, Hermes Bravo, Ricardo L Smith, and Manuel Angel. Topography and morphology of retinal ganglion cells in falconiforms: A study on predatory and carrion-eating birds. *The Anatomical Record*, 229(2):271–277, 1991.

[119] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[120] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[121] Elena Ivanova, G-S Hwang, and Z-H Pan. Characterization of transgenic mouse lines expressing cre recombinase in the retina. *Neuroscience*, 165(1):233–243, 2010.

[122] Mohammadhassan Izadyyazdanabadi, Evgenii Belykh, Xiaochun Zhao, Leandro Borba Moreira, Sirin Gandhi, Claudio Cavallo, Jennifer Eschbacher, Peter Nakaji, Mark C Preul, and Yezhou Yang. Fluorescence image histology pattern transformation using image style transfer. *Frontiers in Oncology*, page 519, 2019.

[123] Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature methods*, 15(8):605–610, 2018.

[124] Arnim Jenett, Gerald M Rubin, Teri-TB Ngo, David Shepherd, Christine Murphy, Heather Dionne, Barret D Pfeiffer, Amanda Cavallaro, Donald Hall, Jennifer Jeter, et al. A gal4-driver line resource for drosophila neurobiology. *Cell reports*, 2(4):991–1001, 2012.

[125] Chang-Jin Jeon, Enrica Strettoi, and Richard H Masland. The major cell populations of the mouse retina. *Journal of Neuroscience*, 18(21):8936–8946, 1998.

[126] David Jiménez, Demetrio Labate, Ioannis A Kakadiaris, and Manos Papadakis. Improved automatic centerline tracing for dendritic and axonal structures. *Neuroinformatics*, 13:227–244, 2015.

[127] Dezhe Z Jin, Ting Zhao, David L Hunt, Rachel P Tillage, Ching-Lung Hsu, and Nelson Spruston. Shutu: open-source software for efficient and accurate reconstruction of dendritic morphology. *Frontiers in Neuroinformatics*, 13:68, 2019.

[128] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[129] M Kamermans and H Spekreijse. The feedback pathway from horizontal cells to cones: A mini review with a look ahead. *Vision research*, 39(15):2449–2468, 1999.

[130] Takuhiro Kaneko and Tatsuya Harada. Noise robust generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8404–8414, 2020.

[131] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[132] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[133] Mehmet Kayaalp. Patient privacy in the era of big data. *Balkan medical journal*, 35(1):8–17, 2018.

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

[135] Vassili A Kovalev, Frithjof Kruggel, H-J Gertz, and D Yves von Cramon. Three-dimensional texture analysis of mri brain datasets. *IEEE Transactions on Medical Imaging*, 20(5):424–433, 2001.

[136] Oren Z Kraus, Jimmy Lei Ba, and Brendan J Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016.

[137] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[138] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[139] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2129–2137, 2019.

[140] Ralf Kühn and Raul M. Torres. Cre/loxp recombination system and gene targeting. *Transgenesis Techniques: Principles and Protocols*, pages 175–204, 2002.

[141] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[142] Ilya Kuzovkin, Raul Vicente, Mathilde Petton, Jean-Philippe Lachaux, Monica Baciu, Philippe Kahane, Sylvain Rheims, Juan R Vidal, and Jaan Aru. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications biology*, 1(1):107, 2018.

[143] Melike Lakadamyali, Hazen Babcock, Mark Bates, Xiaowei Zhuang, and Jeff Lichtman. 3d multicolor super-resolution imaging offers improved accuracy in neuron tracing. *PloS one*, 7(1):e30826, 2012.

[144] Paras Lakhani and Baskaran Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017.

[145] Joseph R Lakowicz and Joseph R Lakowicz. Fluorophores. *Principles of fluorescence spectroscopy*, pages 63–93, 1999.

[146] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[147] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[148] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebastian Seung. Superhuman accuracy on the snemi3d connectomics challenge. *arXiv preprint arXiv:1706.00120*, 2017.

[149] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

[150] Timothée Lesort, Florian Bordes, Jean-Francois Goudou, and David Filliat. Evaluation of generative networks through their data augmentation capacity. 2018.

[151] Qiufu Li and Linlin Shen. 3d neuron reconstruction in tangled neuronal image with deep networks. *IEEE transactions on medical imaging*, 39(2):425–435, 2019.

[152] Qiufu Li and Linlin Shen. 3d neuron reconstruction in tangled neuronal image with deep networks. *IEEE transactions on medical imaging*, 39(2):425–435, 2019.

[153] Rongjian Li, Tao Zeng, Hanchuan Peng, and Shuiwang Ji. Deep learning segmentation of optical microscopy images improves 3-d neuron reconstruction. *IEEE transactions on medical imaging*, 36(7):1533–1541, 2017.

[154] Haoyi Liang, Scott T Acton, and Daniel S Weller. Content-aware neuron image enhancement. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3510–3514. IEEE, 2017.

[155] Jeff W Lichtman and José-Angel Conchello. Fluorescence microscopy. *Nature methods*, 2(12):910–919, 2005.

[156] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 305–314, 2021.

[157] Kaisa Liimatainen, Lauri Kananen, Leena Latonen, and Pekka Ruusuvuori. Iterative unsupervised domain adaptation for generalized cell detection from brightfield z-stacks. *BMC Bioinformatics*, 20(1):80, 2019.

[158] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[159] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[160] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[161] Jack Lindsey, Samuel A Ocko, Surya Ganguli, and Stephane Deny. A unified theory of early visual representations from retina to cortex through anatomically constrained deep cnns. *arXiv preprint arXiv:1901.00945*, 2019.

[162] Mengting Liu, Piyush Maiti, Sophia Thomopoulos, Alyssa Zhu, Yaqiong Chai, Hosung Kim, and Neda Jahanshad. Style transfer using generative adversarial networks for multi-site mri harmonization. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*, pages 313–322. Springer, 2021.

[163] Min Liu, Huiqiong Luo, Yinghui Tan, Xueping Wang, and Weixun Chen. Improved v-net based image segmentation for 3d neuron reconstruction. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 443–448. IEEE, 2018.

[164] Siqi Liu, Donghao Zhang, Yang Song, Hanchuan Peng, and Weidong Cai. Triple-crossing 2.5 d convolutional neural network for detecting neuronal arbours in 3d microscopic images. In *Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8*, pages 185–193. Springer, 2017.

[165] Xiaoxuan Liu, Livia Faes, Aditya U Kale, Siegfried K Wagner, Dun Jack Fu, Alice Bruynseels, Thushika Mahendiran, Gabriella Moraes, Mohith Shamdas, Christoph Kern, et al. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The lancet digital health*, 1(6):e271–e297, 2019.

[166] Yufeng Liu, Gaoyu Wang, Giorgio A Ascoli, Jiangning Zhou, and Lijuan Liu. Neuron tracing from light microscopy images: automation, deep learning and bench testing. *Bioinformatics*, 38(24):5329–5339, 2022.

[167] Jean Livet, Tamily A Weissman, Hyuno Kang, Ryan W Draft, Ju Lu, Robyn A Bennis, Joshua R Sanes, and Jeff W Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450(7166):56–62, 2007.

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

[169] Ju Lu. Neuronal tracing for connectomic studies. *Neuroinformatics*, 9:159–166, 2011.

[170] Margaret A MacNeil and Richard H Masland. Extreme diversity among amacrine cells: implications for function. *Neuron*, 20(5):971–982, 1998.

[171] Robert E Marc, Bryan W Jones, James R Anderson, Krista Kinard, David W Marshak, John H Wilson, Theodore Wensel, and Robert J Lucas. Neural reprogramming in retinal degeneration. *Investigative ophthalmology & visual science*, 48(7):3364–3371, 2007.

[172] WB Marks, William H Dobelle, and Edward F MacNichol Jr. Visual pigments of single primate cones. *Science*, 143(3611):1181–1183, 1964.

[173] Michael F Marmor, Graham E Holder, Mathias W Seeliger, and Shuichi Yamamoto. Standard for clinical electroretinography (2004 update). *Documenta ophthalmologica*, 108:107–114, 2004.

[174] Richard H Masland. Amacrine cells. *Trends in neurosciences*, 11(9):405–410, 1988.

[175] Richard H Masland. The tasks of amacrine cells. *Visual neuroscience*, 29(1):3–9, 2012.

[176] Brian Matsumoto. *Cell biological applications of confocal microscopy*. Elsevier, 2003.

[177] DG McHahon, AG Knapp, and JE Dowling. Horizontal cell gap junctions: single-channel conductance and modulation by dopamine. *Proceedings of the National Academy of Sciences*, 86(19):7639–7643, 1989.

[178] Lane McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen Baccus. Deep learning models of the retinal response to natural scenes. *Advances in neural information processing systems*, 29, 2016.

[179] Erik Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77(7):693–704, 2010.

[180] Erik Meijering, M Jacob, J-CF Sarria, Pl Steiner, H Hirling, and e M Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 58(2):167–176, 2004.

[181] Erik Meijering, Mathews Jacob, J-C Floyd Sarria, and Michael Unser. A novel approach to neurite tracing in fluorescence microscopy images. In *SIP*, pages 491–495, 2003.

[182] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[183] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[184] Tony CW Mok and Albert CS Chung. Learning data augmentation for brain tumor segmentation with coarse-to-fine generative adversarial networks. In *International MICCAI Brainlesion Workshop*, pages 70–80. Springer, 2018.

[185] Monica Monici. Cell and tissue autofluorescence research and diagnostic applications. *Biotechnology annual review*, 11:227–256, 2005.

[186] Roberto Morelli, Luca Clissa, Roberto Amici, Matteo Cerri, Timna Hitrec, Marco Luppi, Lorenzo Rinaldi, Fabio Squarcio, and Antonio Zoccoli. Automating cell counting in fluorescent microscopy through deep learning with c-resunet. *Scientific Reports*, 11(1):22920, 2021.

[187] Darren R Myatt, Tye Hadlington, Giorgio A Ascoli, and Slawomir J Nasuto. Neuromantic–from semi-manual to semi-automatic reconstruction of neuron morphology. *Frontiers in neuroinformatics*, 6:4, 2012.

[188] Junichi Nakai, Masamichi Ohkura, and Keiji Imoto. A high signal-to-noise ca2+ probe composed of a single green fluorescent protein. *Nature biotechnology*, 19(2):137–141, 2001.

[189] Michael Neumann and Detlef Gabel. Simple method for reduction of autofluorescence in fluorescence microscopy. *Journal of Histochemistry & Cytochemistry*, 50(3):437–439, 2002.

[190] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016.

[191] Behnam Neyshabur, Srinadh Bhojanapalli, and Ayan Chakrabarti. Stabilizing gan training with multiple random projections. *arXiv preprint arXiv:1705.07831*, 2017.

[192] Quoc Cuong Ninh, Thi-Thao Tran, Tien Thanh Tran, Thi Anh Xuan Tran, and Van-Truong Pham. Skin lesion segmentation based on modification of segnet neural networks. In *2019 6th NAFOSTED conference on information and computer science (NICS)*, pages 575–578. IEEE, 2019.

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

[194] Seung Wook Oh, Julie A Harris, Lydia Ng, Brent Winslow, Nicholas Cain, Stefan Mihalas, Quanxin Wang, Chris Lau, Leonard Kuan, Alex M Henry, et al. A mesoscale connectome of the mouse brain. *Nature*, 508(7495):207–214, 2014.

[195] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[196] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[197] Aude Oliva and Philippe G Schyns. Coarse blobs or fine edges? evidence that information diagnosticity changes the perception of complex visual stimuli. *Cognitive psychology*, 34(1):72–107, 1997.

[198] Anton Osokin, Anatole Chessel, Rafael E Carazo Salas, and Federico Vaggi. Gans for biological image synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2233–2242, 2017.

[199] Stephen W Paddock. Confocal laser scanning microscopy. *Biotechniques*, 27(5):992–1004, 1999.

[200] Krzysztof Palczewski. G protein–coupled receptor rhodopsin. *Annu. Rev. Biochem.*, 75:743–767, 2006.

[201] Zhaoqing Pan, Weijie Yu, Bosi Wang, Haoran Xie, Victor S Sheng, Jianjun Lei, and Sam Kwong. Loss functions of generative adversarial networks (gans): opportunities and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4):500–522, 2020.

[202] Praveen Pankajakshan, Laure Blanc-Féraud, Zvi Kam, and Josiane Zerubia. Point-spread function retrieval for fluorescence microscopy. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1095–1098. IEEE, 2009.

[203] Bumjun Park, Songhyun Yu, and Jechang Jeong. Densely connected hierarchical network for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[204] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[205] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[206] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[207] Leo Peichl and Juncal González-Soriano. Morphological types of horizontal cell in rodent retinae: a comparison of rat, mouse, gerbil, and guinea pig. *Visual neuroscience*, 11(3):501–517, 1994.

[208] Hanchuan Peng, Michael Hawrylycz, Jane Roskams, Sean Hill, Nelson Spruston, Erik Meijering, and Giorgio A Ascoli. Bigneuron: large-scale 3d neuron reconstruction from optical microscopy images. *Neuron*, 87(2):252–256, 2015.

[209] Hanchuan Peng, Michael Hawrylycz, Jane Roskams, Sean Hill, Nelson Spruston, Erik Meijering, and Giorgio A Ascoli. Bigneuron: large-scale 3d neuron reconstruction from optical microscopy images. *Neuron*, 87(2):252–256, 2015.

[210] Hanchuan Peng, Fuhui Long, and Gene Myers. Automatic 3d neuron tracing using all-path pruning. *Bioinformatics*, 27(13):i239–i247, 2011.

[211] Hanchuan Peng, Zongcai Ruan, Fuhui Long, Julie H Simpson, and Eugene W Myers. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology*, 28(4):348–353, 2010.

[212] Ido Perlman, Helga Kolb, and Ralph Nelson. S-potentials and horizontal cells. *Webvision: the organization of the retina and visual system [Internet]*, 2011.

[213] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[214] Darcy S Peterka, Hiroto Takahashi, and Rafael Yuste. Imaging voltage in neurons. *Neuron*, 69(1):9–21, 2011.

[215] Carl CH Petersen. The functional organization of the barrel cortex. *Neuron*, 56(2):339–355, 2007.

[216] Courtney Peterson. How it works: the charged-coupled device, or ccd. *Journal of young investigators*, 3(1), 2001.

[217] Marco Piccolino. Cajal and the retina: a 100-year retrospective. *Trends in neurosciences*, 11(12):521–525, 1988.

[218] MH Pirenne. Dark-adaptation and night vision. In *The Visual Process*, pages 93–122. Elsevier, 1962.

[219] Johan S Ploem. Laser scanning fluorescence microscopy. *Applied optics*, 26(16):3226–3231, 1987.

[220] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[221] Miroslav Radojević and Erik Meijering. Automated neuron reconstruction from 3d fluorescence microscopy images using sequential monte carlo estimation. *Neuroinformatics*, 17(3):423–442, 2019.

[222] Ivet Rafegas and Maria Vanrell. Color representation in cnns: parallelisms with biological vision. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2697–2705, 2017.

[223] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[224] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[225] Enrique Ramón-Moliner. The golgi-cox technique, 1970.

[226] Jian Ren, Ilker Hacihaliloglu, Eric A Singer, David J Foran, and Xin Qi. Unsupervised domain adaptation for classification of histopathology whole-slide images. *Frontiers in Bioengineering and Biotechnology*, 7, 2019.

[227] Kenneth J Rhodes and James S Trimmer. Antibodies as valuable neuroscience research tools versus reagents of mass distraction. *Journal of Neuroscience*, 26(31):8017–8020, 2006.

[228] Yair Rivenson, Zoltán Göröcs, Harun Günaydin, Yibo Zhang, Hongda Wang, and Aydogan Ozcan. Deep learning microscopy. *Optica*, 4(11):1437–1443, 2017.

[229] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[230] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[231] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020.

[232] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[233] Sameera V Mohd Sagheer and Sudhish N George. A review on medical image denoising algorithms. *Biomedical signal processing and control*, 61:102036, 2020.

[234] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8*, pages 379–387. Springer, 2017.

[235] Hojjat Salehinejad, Shahrokh Valaee, Tim Dowdell, Errol Colak, and Joseph Barfett. Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 990–994. IEEE, 2018.

[236] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[237] Michael J Sanderson, Ian Smith, Ian Parker, and Martin D Bootman. Fluorescence microscopy. *Cold Spring Harbor Protocols*, 2014(10):pdb–top071795, 2014.

[238] Joshua R Sanes and Richard H Masland. The types of retinal ganglion cells: current status and implications for neuronal classification. *Annual review of neuroscience*, 38:221–246, 2015.

[239] Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Hideki Atsumi, Shigeyuki Yoshida, Thomas Koller, Guido Gerig, and Ron Kikinis. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical image analysis*, 2(2):143–168, 1998.

[240] Brian Sauer and Nancy Henderson. Site-specific dna recombination in mammalian cells by the cre recombinase of bacteriophage p1. *Proceedings of the National Academy of Sciences*, 85(14):5166–5170, 1988.

[241] Stephan Schmitt, Jan Felix Evers, Carsten Duch, Michael Scholz, and Klaus Obermayer. New methods for the computer-assisted 3-d reconstruction of neurons from confocal image stacks. *Neuroimage*, 23(4):1283–1298, 2004.

[242] Steffen Schmitz-Valckenberg, Frank G Holz, Alan C Bird, and Richard F Spaide. Fundus autofluorescence imaging: review and perspectives. *Retina*, 28(3):385–409, 2008.

[243] Philippe G Schyns and Aude Oliva. From blobs to boundary edges: Evidence for time-and spatial-scale-dependent scene recognition. *Psychological science*, 5(4):195–200, 1994.

[244] ML Sears. The eye. vegetative physiology and biochemistry—, hugh davson (ed.) the eye. vegetative physiology and biochemistry, vol. 1, academic press, new york (1969), p. 679, 1971.

[245] Denis Semwogerere and Eric R Weeks. Confocal microscopy. *Encyclopedia of biomaterials and biomedical engineering*, 23:1–10, 2005.

[246] Nathan C Shaner, George H Patterson, and Michael W Davidson. Advances in fluorescent protein technology. *Journal of cell science*, 120(24):4247–4260, 2007.

[247] Aabid Shariff, Robert F Murphy, and Gustavo K Rohde. A generative model of microtubule distributions, and indirect estimation of its parameters from fluorescence microscopy images. *Cytometry Part A: The Journal of the International Society for Advancement of Cytometry*, 77(5):457–466, 2010.

[248] Lindsay T Sharpe and Andrew Stockman. Rod pathways: the importance of seeing nothing. *Trends in neurosciences*, 22(11):497–504, 1999.

[249] Peter J Shaw and David J Rawlins. The point-spread function of a confocal microscope: its measurement and use in deconvolution of 3-d data. *Journal of Microscopy*, 163(2):151–165, 1991.

[250] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 1–11. Springer, 2018.

[251] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[252] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[253] Satya P Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan, and Balázs Gulyás. 3d deep learning on medical images: a review. *Sensors*, 20(18):5097, 2020.

[254] Chunfeng Song, Feng Liu, Yongzhen Huang, Liang Wang, and Tieniu Tan. Auto-encoder based data clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 18th Iberoamerican Congress, CIARP 2013, Havana, Cuba, November 20-23, 2013, Proceedings, Part I 18*, pages 117–124. Springer, 2013.

[255] Dmitry V Sorokin, Igor Peterlík, Vladimír Ulman, David Svoboda, Tereza Nečasová, Katsiarina Morgaenko, Lívia Eiselleová, Lenka Tesařová, and Martin Maška. Filogen: a model-based generator of synthetic 3-d time-lapse sequences of single motile cells with growing and branching filopodia. *IEEE Transactions on Medical Imaging*, 37(12):2630–2641, 2018.

[256] Maureen E Stabio, Katelyn B Sondereker, Sean D Haghgou, Brittany L Day, Berrien Chidsey, Shai Sabbah, and Jordan M Renna. A novel map of the mouse eye for orienting retinal topography in anatomical space. *Journal of Comparative Neurology*, 526(11):1749–1759, 2018.

[257] J Steinebach. El lehmann, jp romano: Testing statistical hypotheses. *Metrika*, 64(2):255–256, 2006.

[258] Ernst HK Stelzer. Contrast, resolution, pixelation, dynamic range and signal-to-noise ratio: fundamental limits to resolution in fluorescence light microscopy. *Journal of Microscopy*, 189(1):15–24, 1998.

[259] David J Stephens and Victoria J Allan. Light microscopy techniques for live cell imaging. *science*, 300(5616):82–86, 2003.

[260] Uygar Sümbül, Douglas Roossien, Dawen Cai, Fei Chen, Nicholas Barry, John P Cunningham, Edward Boyden, and Liam Paninski. Automated scalable segmentation of neurons from multispectral images. In *Advances in Neural Information Processing Systems*, pages 1912–1920, 2016.

[261] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[262] David Svoboda and Vladimir Ulman. Mitogen: a framework for generating 3d synthetic time-lapse sequences of cell populations in fluorescence microscopy. *IEEE Transactions on Medical Imaging*, 36(1):310–321, 2016.

[263] Karel Svoboda and Ryohei Yasuda. Principles of two-photon excitation microscopy and its applications to neuroscience. *Neuron*, 50(6):823–839, 2006.

[264] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

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

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

[267] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[268] Shin-ya Takemura, Yoshinori Aso, Toshihide Hige, Allan Wong, Zhiyuan Lu, C Shan Xu, Patricia K Rivlin, Harald Hess, Ting Zhao, Toufiq Parag, et al. A connectome of a learning and memory center in the adult drosophila brain. *Elife*, 6:e26975, 2017.

[269] W Rowland Taylor, Shigang He, William R Levick, and David I Vaney. Dendritic computation of direction selectivity by retinal ganglion cells. *Science*, 289(5488):2347–2350, 2000.

[270] Giles Tetteh, Velizar Efremov, Nils D Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud, and Bjoern H Menze. Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *Frontiers in Neuroscience*, page 1285, 2020.

[271] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020.

[272] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[273] Hanne H Thoen, Martin J How, Tsyr-Huei Chiou, and Justin Marshall. A different form of color vision in mantis shrimp. *Science*, 343(6169):411–413, 2014.

[274] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[275] Roger Y Tsien. The green fluorescent protein. *Annual review of biochemistry*, 67(1):509–544, 1998.

[276] Yoshihiko Tsukamoto and Naoko Omi. Classification of mouse retinal bipolar cells: type-specific connectivity with special reference to rod-driven aii amacrine pathways. *Frontiers in neuroanatomy*, 11:92, 2017.

[277] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[278] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[279] Zlatko Vasilkoski and Armen Stepanyants. Detection of the optimal neuron traces in confocal microscopy images. *Journal of Neuroscience Methods*, 178(1):197–204, 2009.

[280] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[281] Christiane Voufo, Andy Quaen Chen, Benjamin E Smith, Rongshan Yan, Marla B Feller, and Alexandre Tiriac. Circuit mechanisms underlying embryonic retinal waves. *Elife*, 12:e81983, 2023.

[282] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 765–780, 2018.

[283] Heng Wang, Donghao Zhang, Yang Song, Siqi Liu, Heng Huang, Mei Chen, Hanchuan Peng, and Weidong Cai. Multiscale kernels for enhanced u-shaped network to improve 3d neuron tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[284] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[285] Yu Wang, Arunachalam Narayanaswamy, Chia-Ling Tsai, and Badrinath Roysam. A broadly applicable 3-d neuron tracing method based on open-curve snake. *Neuroinformatics*, 9:193–217, 2011.

[286] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[287] Simon K Warfield, Kelly H Zou, and William M Wells. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7):903–921, 2004.

[288] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[289] Per Welander, Simon Karlsson, and Anders Eklund. Generative adversarial networks for image-to-image translation on multi-contrast mr images-a comparison of cyclegan and unit. *arXiv preprint arXiv:1806.07777*, 2018.

[290] Brian A Wilt, Laurie D Burns, Eric Tatt Wei Ho, Kunal K Ghosh, Eran A Mukamel, and Mark J Schnitzer. Advances in light microscopy for neuroscience. *Annual Review of Neuroscience*, 32:435–506, 2009.

[291] Jingpeng Wu, Yong He, Zhongqin Yang, Congdi Guo, Qingming Luo, Wei Zhou, Shangbin Chen, Anan Li, Benyi Xiong, Tao Jiang, et al. 3d braincv: simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution. *Neuroimage*, 87:199–208, 2014.

[292] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

[293] Fei Xia and Meliha Yetisgen-Yildiz. Clinical corpus annotation: challenges and strategies. In *Proceedings of the third workshop on building and evaluating resources for biomedical text mining (BioTxtM'2012) in conjunction with the international conference on language resources and evaluation (LREC), Istanbul, Turkey*, pages 21–27, 2012.

[294] Hang Xiao and Hanchuan Peng. App2: automatic tracing of 3d neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11):1448–1454, 2013.

[295] Daiyan Xin and Stewart A Bloomfield. Dark-and light-induced changes in coupling between horizontal cells in mammalian retina. *Journal of Comparative Neurology*, 405(1):75–87, 1999.

[296] Fuyong Xing, Yuanpu Xie, Hai Su, Fujun Liu, and Lin Yang. Deep learning in microscopy image analysis: A survey. *IEEE transactions on neural networks and learning systems*, 29(10):4550–4568, 2017.

[297] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.

[298] Qiaojing Yan and Wei Wang. Dcgans for image super-resolution, denoising and debluring. *Advances in neural information processing systems*, pages 487–495, 2017.

[299] Wenjun Yan, Mallory A Laboulaye, Nicholas M Tran, Irene E Whitney, Inbal Benhar, and Joshua R Sanes. Mouse retinal cell atlas: molecular identification of over sixty amacrine cell types. *Journal of Neuroscience*, 40(27):5177–5195, 2020.

[300] Zengqiang Yan, Xin Yang, and Kwang-Ting Cheng. A three-stage deep learning model for accurate retinal vessel segmentation. *IEEE journal of Biomedical and Health Informatics*, 23(4):1427–1436, 2018.

[301] Jian Yang, Ming Hao, Xiaoyang Liu, Zhijiang Wan, Ning Zhong, and Hanchuan Peng. Fmst: an automatic neuron tracing method based on fast marching and minimum spanning tree. *Neuroinformatics*, 17(2):185–196, 2019.

[302] Lawrence A Yannuzzi, Michael D Ober, Jason S Slakter, Richard F Spaide, Yale L Fisher, Robert W Flower, and Richard Rosen. Ophthalmic fundus imaging: today and beyond. *American journal of ophthalmology*, 137(3):511–524, 2004.

[303] Yuancheng Ye, Lijuan Wang, Yue Wu, Yinpeng Chen, Yingli Tian, Zicheng Liu, and Zhengyou Zhang. Gan quality index (gqi) by gan-induced classifier. 2018.

[304] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[305] Xiaosong Yuan, Joshua T Trachtenberg, Steve M Potter, and Badrinath Roysam. Mdl constrained 3-d grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7:213–232, 2009.

[306] Qasim Zaidi, Justin Marshall, Hanne Thoen, and Bevil R Conway. Evolution of neural computations: Mantis shrimp and human color decoding. *i-Perception*, 5(6):492–496, 2014.

[307] Bo Zhang, Josiane Zerubia, and Jean-Christophe Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied optics*, 46(10):1819–1829, 2007.

[308] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. understanding deep learning requires rethinking generalization ", iclr 2017. *arXiv preprint arXiv:1611.03530*, 2017.

[309] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[310] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[311] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[312] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. A poisson-gaussian denoising dataset with real fluorescence microscopy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11710–11718, 2019.

[313] Jie Zhao, Xuejin Chen, Zhiwei Xiong, Dong Liu, Junjie Zeng, Chaoyu Xie, Yueyi Zhang, Zheng-Jun Zha, Guoqiang Bi, and Feng Wu. Neuronal population reconstruction from ultra-scale optical microscopy images via progressive learning. *IEEE Transactions on Medical Imaging*, 39(12):4034–4046, 2020.

[314] Zhi Zhou, Hsien-Chi Kuo, Hanchuan Peng, and Fuhui Long. Deepneuron: an open deep learning toolbox for neuron tracing. *Brain informatics*, 5(2):1–9, 2018.

[315] Zhi Zhou, Staci Sorensen, Hongkui Zeng, Michael Hawrylycz, and Hanchuan Peng. Adaptive image enhancement for tracing 3d morphologies of neurons and brain vasculatures. *Neuroinformatics*, 13:153–166, 2015.

[316] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.

[317] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[318] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[319] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.