Medical Image Analysis 86 (2023) 102768

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Unpaired mesh-to-image translation for 3D fluorescent microscopy images of neurons

Mihael Cudic [a,b,c], Jeffrey S. Diamond [b], J. Alison Noble [c,*]

[a] National Institutes of Health Oxford-Cambridge Scholars Program, USA
[b] National Institutes of Neurological Diseases and Disorders, Bethesda, MD 20814, USA
[c] Department of Engineering Science, University of Oxford, Oxford OX3 7DQ, UK

ARTICLE INFO

Dataset link: https://github.com/MihaelCudic/BioSPADE.git, https://data.mendeley.com/datasets/f6kk4364p4

MSC:
41A05
41A10
65D05
65D17

Keywords:
Unpaired image translation
Style transfer
Synthetic fluorescent microscopy images
Synthetic neurons

ABSTRACT

While Generative Adversarial Networks (GANs) can now reliably produce realistic images in a multitude of imaging domains, they are ill-equipped to model thin, stochastic textures present in many large 3D fluorescent microscopy (FM) images acquired in biological research. This is especially problematic in neuroscience where the lack of ground truth data impedes the development of automated image analysis algorithms for neurons and neural populations. We therefore propose an unpaired mesh-to-image translation methodology for generating volumetric FM images of neurons from paired ground truths. We start by learning unique FM styles efficiently through a Gramian-based discriminator. Then, we stylize 3D voxelized meshes of previously reconstructed neurons by successively generating slices. As a result, we effectively create a synthetic microscope and can acquire realistic FM images of neurons with control over the image content and imaging configurations. We demonstrate the feasibility of our architecture and its superior performance compared to state-of-the-art image translation architectures through a variety of texture-based metrics, unsupervised segmentation accuracy, and an expert opinion test. In this study, we use 2 synthetic FM datasets and 2 newly acquired FM datasets of retinal neurons.

1. Introduction

Fluorescent 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 (Matsumoto, 2003; Svoboda and Yasuda, 2006; Wilt et al., 2009). Image data collected through FM can then be used to study neuron activity, network organization, or tissue composition (Grienberger and Konnerth, 2012; Livet et al., 2007; Peterka et al., 2011; Rhodes and Trimmer, 2006). However, extracting the relevant information from these images to test scientific hypotheses remains challenging as images are large and plentiful. Efforts in the neuroscience community have been made to streamline image analysis using a mixture of crowd sourcing large FM datasets (Brown et al., 2011; Peng et al., 2015; Berens et al., 2017), automated neural tracing solutions (Xiao and Peng, 2013; Wu et al., 2014; Feng et al., 2015; Yang et al., 2019; Li et al., 2017; Zhou et al., 2018; Apthorpe et al., 2016; Zhao et al., 2020; Chen et al., 2021; Li et al., 2017), and synthetic training datasets (Shariff et al., 2010; Svoboda and Ulman, 2016; Sorokin et al., 2018; Li and Shen, 2019).

While some approaches have met success, they are limited in their ability to impact the general neuroscience community. For one, public FM datasets do not contain the full variety of image characteristics conferred by different biological samples, experimental configurations, and optical calibrations. Moreover, off-the-shelf neural tracing algorithms focus on local connectivity patterns to provide generalizable tracing solutions at the expense of accuracy as global neuron morphology for a specific cell type is ignored (Acciai et al., 2016). Additional data representative of the task at hand is still required for optimal fine-tuning. Collecting FM images of sufficient quality and clarity to serve as ground truths is typically impractical and, when exposure to excitation light must be limited for various experimental reasons, often impossible. Synthetic datasets can be generated as an alternative, but techniques are unable to model both realistic local textures and global morphological noise.

We show in this paper that we overcome these challenges by learning local and global FM imaging characteristics directly from few unlabeled images acquired from an experiment and then stylizing meshes of previously reconstructed neurons (Fig. 1). This way, large numbers of representative training data can be generated with "gold standard" ground truths. Unlike other microscopy image generators (Baniukiewicz et al., 2019), our unpaired mesh-to-image translation paradigm eliminates the need for any manual annotation as

* Corresponding author.
E-mail address: alison.noble@eng.ox.ac.uk (J.A. Noble).

https://doi.org/10.1016/j.media.2023.102768
Received 19 April 2022; Received in revised form 18 January 2023; Accepted 8 February 2023
Available online 11 February 2023
1361-8415/Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

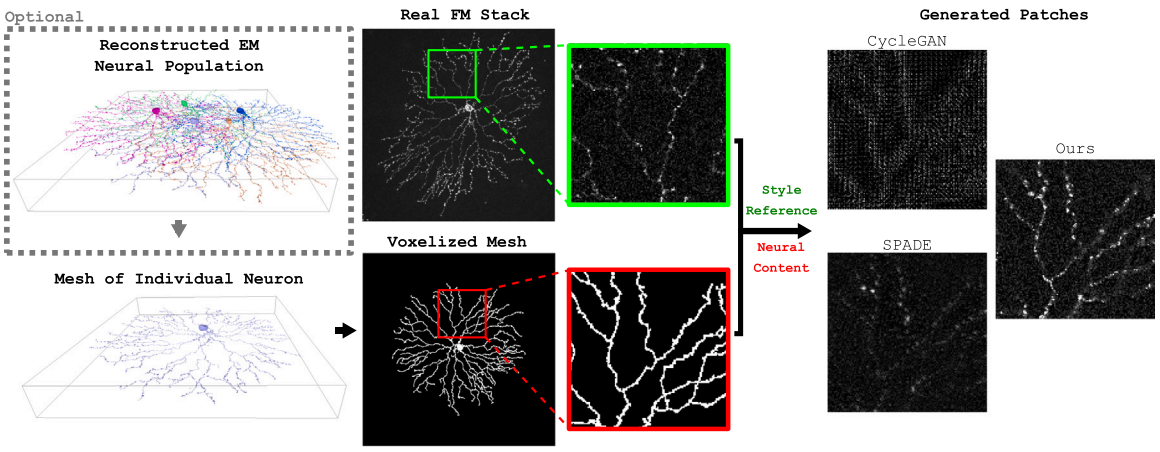

Fig. 1. An overview of our proposed pipeline and a visual comparison of generated synthetic fluorescent microscopy (FM) patches. We start with a neural mesh which can be sourced from publicly available neural population reconstructions of Electron Microscopy (EM) blocks. Meshes are then voxelized and 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 Helmstaedter et al. (2013).

neural meshes can be sourced from rapidly growing, publicly available datasets despite not having corresponding FM images (Helmstaedter et al., 2013). Neural meshes provide the added benefit of representing the volume of neurons through nodes and edges which can be easily 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. 1).

In this paper, we propose an unpaired mesh-to-image translation methodology which is shown to reliably and accurately model realistic FM images of neurons. 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.
- The introduction of a discriminator with trainable Gram matrix operations to learn stochastic textures at multiple scales.
- An experimental evaluation of our methodology on 2 synthetic and 2 real FM image datasets of neurons, demonstrating our architecture's superior performance to state-of-the-art image translation architectures.
- All code and our newly acquired 3D FM image datasets are publicly available.

2. Related works

An established approach for generating synthetic FM images of biological content is to use manually-designed shapes, features, and noise distributions (Shariff et al., 2010; Svoboda and Ulman, 2016; Sorokin et al., 2018). For FM images of neurons, neural skeletons are typically convolved with manually-designed point spread functions and varying levels of Poisson noise is injected into the 3D stack (Vasilkoski and Stepanyants, 2009; Radojević and Meijering, 2019). Sümbül et al. modify this approach by using volumetric electron microscopy (EM) reconstructions of neurons to model incongruities along a neuron's dendrites (Sümbül et al., 2016). Determining the optimal feature set to model FM imaging statistics, however, is not straight forward. 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 (Chernyavskiy et al., 2010; Ghosh and Preza, 2015). Estimating these statistics experimentally is also problematic as it requires additional equipment and is highly susceptible to noise (Pankajakshan et al., 2009). Even in the ideal case of being able to accurately model local 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. Instead, GANs (Goodfellow et al., 2014) 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.

Deep convolutional GANs have generated a variety of medical and biological images (Hong et al., 2021; Osokin et al., 2017; Goldsborough et al., 2017), but continue to falter when generating images that vary considerably at local and global scales (Bellemo et al., 2018; Beers et al., 2018). Generating 3D images of neurons is particularly challenging as neurons have intricate geometric patterns spanning large 3D volumes and cannot be spatially compressed without losing substantial semantic information (see Figs. 1 and 2). One way to capture information at different scales is to implement a multi-objective discrimination task (Durugkar et al., 2016; Wang et al., 2018; Mok and Chung, 2018; Neyshabur et al., 2017). Another useful technique to more easily learn global semantics is to condition GANs on an abstract representation of the object trying to be generated, denoted as x (Isola et al., 2017). In the case of biological images, x is typically a binarized image of the biological content or corresponding ground truths (Baniukiewicz et al., 2019; Liimatainen et al., 2019; Ren et al., 2019; Bailo et al., 2019; Han et al., 2018; Shin et al., 2018; Kraus et al., 2016; Costa et al., 2017). However, this image-to-image translation paradigm is typically limited to only paired image datasets as both the generator G and discriminator D are conditioned on x to aid convergence during training (Isola et al., 2017). Unpaired image-to-image translation algorithms, namely cycleGANs, have been proposed, but require several additional architectures to ensure correspondence between x and generated images and are therefore excessively computationally intensive for large 3D images (Zhu et al., 2017; Yang et al., 2018). We differ from these methods by only incorporating a shallow segmentation network and segmentation loss during training to ensure that generated images preserve the content of x in an unpaired manner.

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 (Armanious et al., 2018; Izadyyazdanabadi et al., 2019; Cho et al., 2017; Hollandi et al., 2020). Current approaches of style transfer, however, typically

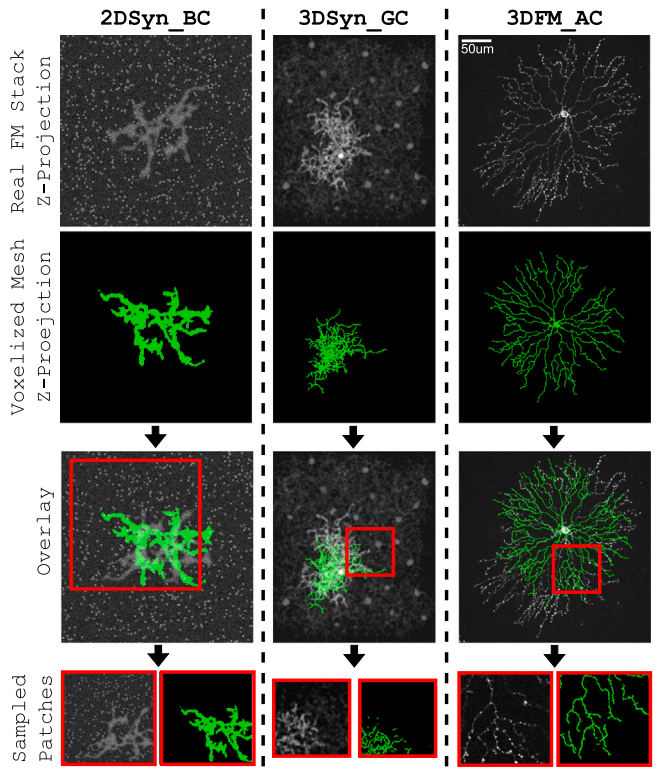

Fig. 2. Overview of our mesh and FM patch sampling protocol for three datasets (columns). We first overlay a real stack (shown as the maximum z-projection in gray) with a voxelized mesh of a previously reconstructed neuron (shown in green) so that features are spatially correlated. Patches are then sampled 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. B.11 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.

use a pre-trained VGGNet which limits the range of possible style representations to be learned (Gatys et al., 2016; Johnson et al., 2016; Wang et al., 2018; Park et al., 2019). 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.

3. Methods

3.1. Sampling the voxelized mesh

We start with a mesh of a previously reconstructed neuron that we wish to represent as an FM stack. This 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. The voxel size, or pitch, used to voxelize the mesh is determined by the resolution of the acquired real FM stacks, ensuring that the geometries and 3D scales of the neurons are consistent across voxel and real domains.

Patches are sampled from matrix M to further reduce GPU memory consumption. We found that randomly sampling patches independently from both FM image and voxelized mesh domains led to significantly worse results as it was not guaranteed that the same features were sampled across domains for a given batch. For example, somas from real FM stacks could be over-sampled in a batch causing a generator to attempt to artificially create soma-like features. This effect is exacerbated when small batch sizes are used. To help stabilize the learning procedure, we instead randomly sample the same spatial regions across domains by overlaying the neuron in M with the neuron in the real stack as shown in Fig. 2.

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 generating 2D slices of neurons that then 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.

As a starting point for our architecture, we decided to use the SPADE as it provides state-of-the-art performance on image translation tasks conditioned on semantic maps and is optimally suited to texturize the uniform empty space seen in our voxelized meshes (Park et al., 2019). We then 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 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 Lim and Ye (2017).

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 a 2D CNN. A summary of our methodology can be found in Fig. 3.

3.3. 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 Karras et al. (2017), computes the average standard deviation of activations across all spatial locations for each feature map, replicates the values, and concatenates those

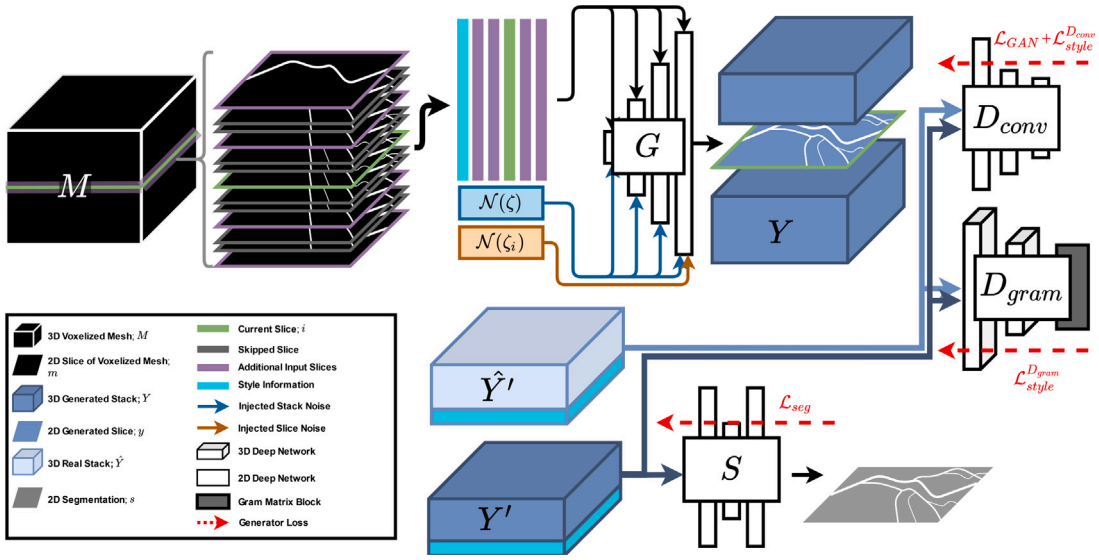

Fig. 3. An overview of our architecture. A 2D slice m_i from a 3D voxelized mesh **M** is input to a generator G along with noise and neighboring slices above and m_i. By using G to map $m_i \rightarrow y_i$ for all slices in **M**, we generate a synthetic 3D image **Y**. The discriminator D then samples slices from **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 ($\mathcal{L}_{seg}$).

values with the original feature maps. For the 2D case, the output of this layer can be summarized by a tuple with elements:

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

such that $\mathcal{F}^{\text{spatial,c}} \in R^{2xLxW}$, 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.

3.4. Noise and acquisition layer

Noise is a defining property of biological images and needs to be explicitly modeled in generator G. For structured noise spanning multiple slices, Gaussian noise is added as an extra feature channel at varying spatial scales and kept constant for each slice of a 3D image.

Similar to noise robust GANs (Kaneko and Harada, 2020; Bora et al., 2018), we also 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 more similarly mimic image acquisition in fluorescent microscopes. More specifically, the `acquisition` layer receives the voxel offset value $\mu_c \in R^{LxW}$, a set of n feature maps $\zeta_c \in R^{nxLxW}$ containing the probability distribution parameters for $P(x)$ where $x \in R^{LxW}$, 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 $\mathcal{X}^c \in R^{FxLxW}$ and obtain the acquired image by:

$$\mathcal{F}^{\text{acq,c}} = \frac{1}{F}\sum_{f=1}^{F} clip(\mu_c + \mathcal{X}_f^c + \beta, 0, 1) \tag{2}$$

where $clip(\cdot,0,1)$ denotes the voxel-wise clipping of values between 0 and 1. To model Gaussian noise, we simply set $\zeta_c \in R^{1xLxW}$, which we denote as σ_c, and sample voxel noise $\mathcal{X}^c \sim \sigma_c \mathcal{N}(0,1)^{FxLxW}$.

However, non-Gaussian distributions are required to model real FM noise. 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. We then sample voxel noise such that $\mathcal{X}^c \sim s_c(-1\log u)^{\alpha_c}$ where $u \sim U(0,1)^{FxLxW}$.

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:

$$\hat{\mathcal{F}^{\text{acq,c}}} = clip\left(\mu_c + \beta + \left(\frac{1}{\alpha_c + 1}\right)^{\alpha_c}, 0, 1\right) \tag{3}$$

Table 1
Architecture details for our Gramian-based discriminator.

Discriminator	Norm.	Act.	Output shape
Input image	–	–	1 × 4 × 128 × 128
Conv3 × 3 × 3	Instance	ReLU	32 × 4 × 128 × 128
Conv4 × 4 × 4	Batch	ReLU	32 × 4 × 128 × 128
Conv3 × 3 × 3	Batch	ReLU	64 × 4 × 64 × 64
Conv4 × 4 × 4	Batch	ReLU	64 × 4 × 64 × 64
Gram	–	–	1 × 2112
Dense	Batch	ReLU	1 × 1056
Dense	–	–	1 × 1

3.5. Gramian-based discriminator

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 $\phi_{J_J}^{D}(y,s)$ is then defined by a $C_J \times C_J$ symmetric matrix with elements:

$$\phi_J^D(y)_{c,c'} = \frac{1}{C_J L_J W_J}\sum_{l=1}^{L}\sum_{w=1}^{W} D_J^{gram}(y,s)_{l,w,c} D_J^{gram}(y,s)_{l,w,c'} \tag{4}$$

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:

$$\mathcal{L}_{style}^{D,j}(\hat{y}, y, s) = \sum_{j=1}^{J} \mathbb{E}_{\hat{y},y} \|(\phi_j^D(\hat{y}, s) - \phi_j^D(y, s))\|_1 \tag{5}$$

3.6. Objective

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

$$G^* = \arg\min_G \max_D \lambda_{seg}\mathcal{L}_{seg}(G) + \sum_{k=1}^{K} \mathcal{L}_{GAN}(G, D_{conv}^k) + \lambda_{style}^{D_{conv}} \sum_{j\in J} \mathcal{L}_{style}^{D_{conv},j}(G) + \lambda_{style}^{D_{gram}} \sum_{j\in J} \mathcal{L}_{style}^{D_{gram},j}(G) \tag{6}$$

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.

4. Datasets

4.1. Synthetic 2D Bipolar Cells (2DSyn_BC)

The first dataset modeled is a synthetic 2D FM dataset of Type 5i bipolar cells (2DSyn_BC). A total of 41 meshes of previously reconstructed EM data provided by Helmstaedter et al. (2013) 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. Voxels were then convolved with a Gaussian filter ($\sigma_x = \sigma_y = 1.5$) and all non-zero voxels were set to 0.4. As a source of structured noise, we randomly inserted 2×2 squares to the image background. Frames were then multiplied by a randomly chosen scalar from the set $\{0.15, 0.30, 0.45\}$ to mimic the varying laser powers. 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 $\{1, 2, 4\}$. To create a second unpaired voxel dataset, the same 41 meshes were flattened and then flipped along the x-axis ensuring that there was no one-to-one mapping of geometries as shown in 2. 128×128 patches were continuously sampled during training.

4.2. Synthetic 3D Ganglion Cells (3DSyn_GC)

To see how well our algorithm extends to 3D FM stacks, we additionally created a synthetic 3D dataset of F-mini Retinal Ganglion Cells (3DSyn_GC). A total of 25 meshes along with their relative spatial locations were provided by Helmstaedter et al. (2013). All meshes were voxelized with a pitch of 1000 with their relative spatial locations preserved. Each mesh was convolved with a Gaussian filter ($\sigma_z = 1.0, \sigma_x = \sigma_y = 1.5$) and the background was set to 0.1. All individual neurons were combined into one image, recreating the neural population. An individual neuron was then identified and amplified (×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 $\{1, 2\}$. The unpaired second voxel dataset was created by flipping all neurons along the x-axis again as shown in Fig. 2. $4 \times 128 \times 128$ patches were additionally sampled during training.

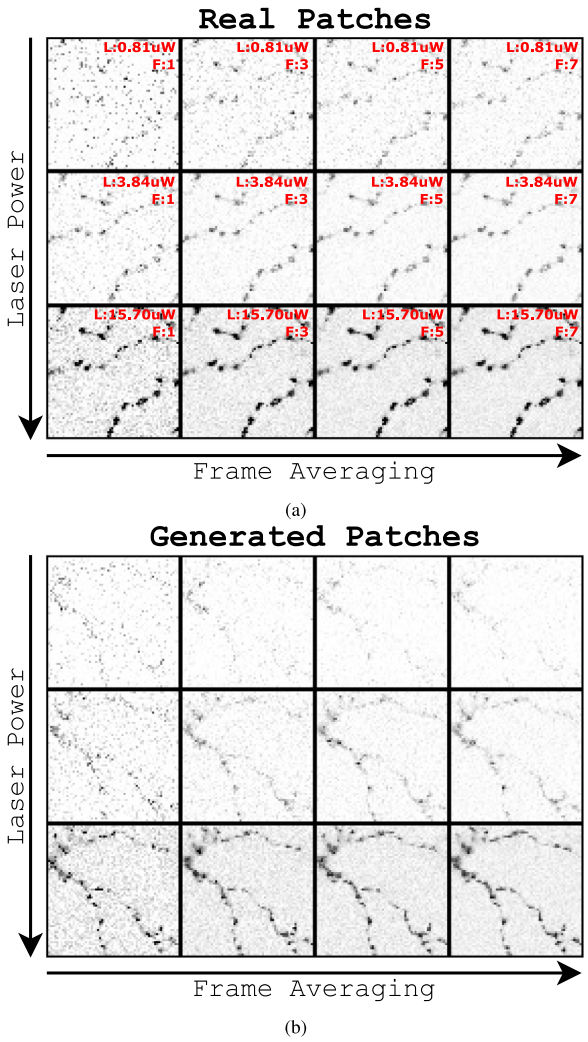

Fig. 4. (a) Real and (b) generated 64×64 patches (inverted for visual effect) sampled from the 3DFM_AC dataset at varying laser powers and frame averaging (labeled L and F respectively in red). For visual purposes, only a subset of styles are portrayed. A complete panel of real and generated patches of varying styles can be found in Fig. B.12.

4.3. FM 3D Amacrine Cells (3DFM_AC)

A novel dataset of FM stacks (3DFM_AC) was acquired of individual EGFP-expressing Starburst Amacrine Cells (SACs) (Fig. B.10). A total of 22 SACs were imaged at 5 varying laser powers (0.81 μW, 1.85 μW, 3.84 μW, 7.71 μW, 15.70 μW) 8 times as shown Figs. 4 and B.12. The xy-resolution and z-resolution of the images acquired were 0.621 um and 0.100 um respectively. More details regarding the protocol to acquire this dataset can be found in Appendix A. To condition the GAN, 19 SAC meshes were voxelized with a pitch of 645 provided by Helmstaedter et al. (2013). Patches of $4 \times 128 \times 128$ were sampled after the meshes were overlaid with real stack for training as shown in Fig. 2. Our entire dataset can be directly accessed from https://data.mendeley.com/datasets/f6kk4364p4.

4.4. FM 3D Bipolar Cell Population (3DFM_BCPop)

An additional novel dataset of 2 FM stacks (3DFM_BCPop) was acquired of GFP-expressing Bipolar Type 2 Cell populations from their somas to their axon terminals (Fig. B.11). Regions of 318.2 um × 318.2 um were imaged with an xy-resolution of 0.155 um and a z-resolution

of 0.100 um, resulting in stacks having an xy-cross section of 2048 × 2048 pixels. Both stacks were collected by averaging 8 frames acquired with a laser power of 15.70 μW. Instead of voxelized meshes, binarized semantic maps were created using two 3D Frangi filters, one for the somatic layer ($\alpha = 0.01, \beta = 10, \sigma_y = 15$) and one for the axon terminal layer ($\alpha = 0.01, \beta = 10, \sigma_y = 8$). This was done so that we could directly compare results between paired and unpaired image translation training regiments. When training was with unpaired data, patches of 4 × 18 × 128 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. Our entire dataset can also be directly accessed from link listed in the previous sub-section.

5. Implementation

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 were 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 learn 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. More over, 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 maps were normalized across all styles. Our `Spatial_stddev` layer was also incorporated in the second to last layer of D_{conv}.

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 networks 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 Park et al. (2019). All code and implementation details can be accessed at https://github.com/MihaelCudic/BioSPADE.git.

6. Evaluation

Quantitatively determining the realism of images produced by our architecture is challenging as conventional metrics like inception score are not suited for the FM domain. It is even more challenging to evaluate performance when using real unpaired data.

To provide meaningful metrics, we thus evaluated on 3 main criteria: content, local texture, and noise distribution. While all training was posed as an unpaired image translation task, 200 128 × 128 paired patches and 100 4 × 128 × 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 × 128 × 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 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 × 512 × 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 × 2048 × 2048 stack due to memory constraints.

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. Because stacks in the 3D_BCPop dataset were acquired once, the NMSE was computed between real and corresponding generated images for a only 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 × 11 × 11 of both real and generated images as a way to estimate if the correct point spread function was learned.

Standard texture similarity metrics were used to determine the local texture realism. The MSE of the Gray Level Co-Occurrence Matrix (GLCM) and the John Shannon Divergence (JSD) of the Local Binary Patterns (LBP) were used on 2D slices (Haralick et al., 1973; Ojala et al., 2002). For the 3D case, we computed MSE of the multi-sort Co-occurrence Matrix (COOC) (Kovalev et al., 2001). Because these metrics are sensitive to voxel noise distributions, a Gaussian filter was used to blur the image prior to the calculation of these metrics.

The realism of the noise was additionally computed by taking the MSE of the Peak Signal-to-Noise Ratio (PSNR) and the JSD of the voxel intensity values for both real and generated images. The MSE of the PSNR was not calculated for the real FM datasets as the theoretical denoised image does not exist.

For our real FM datasets, we also estimated the utility of generated images by computing the Intersection of Union (IoU) score from predicted segmentations of never-before-seen 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.

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. More specifically, each expert was given 15 sets of randomly sampled 6 × 128 × 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.

Table 2
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 Park et al. (2019) and defined in Johnson et al. (2016).

ID	Training Config.	$\mathcal{L}_{style}$	2DSyn_BC					3DSyn_GC						3DFM_AC						
			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**

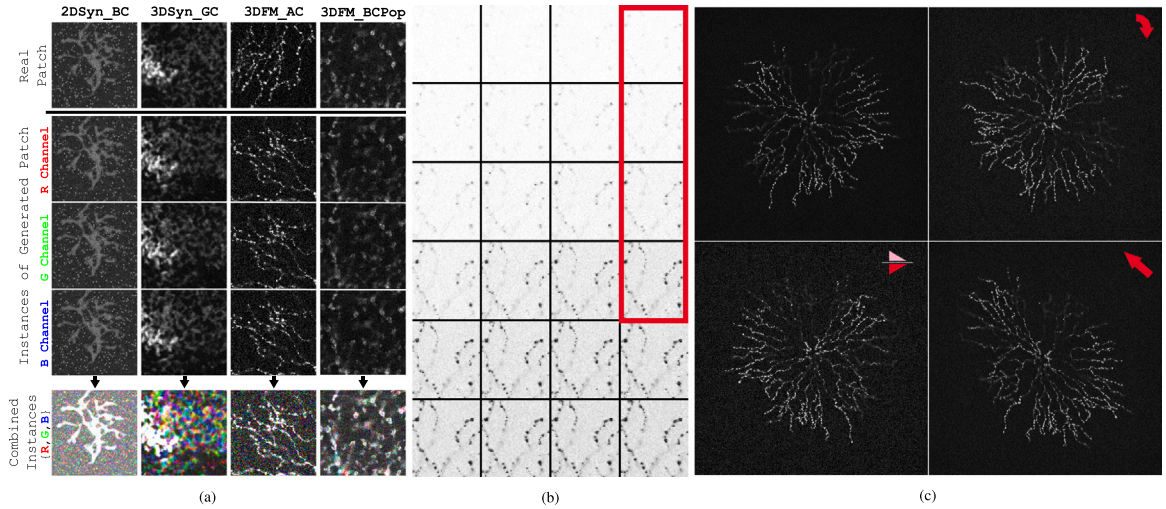

Fig. 5. Demonstration of our generator's ability to generate a wide variety of images. (a) 3 instances of generated patches (middle rows) using the same voxelized mesh and their combined RGB image where every instance corresponds to a different color channel. The combined RGB image shows that our generator 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. (c) 4 generated synthetic slices with each slice displaying differing morphological variations (rotation, mirror, sheer) of the same neural mesh. .

Table 3
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.

ID	Unpaired	3DFM_BCPop					
		NMSE $\times 10^1$	LBP $\times 10^3$	GLCM $\times 10^8$	COOC $\times 10^8$	JSD $\times 10^2$	IoU $\times 10^1$
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**

7. Results

As shown in Figs. 4 and B.12, our architecture generates realistic FM images across a variety of styles and content. Similar to an actual microscope, each new acquisition of a frame preserves the structure of the voxelized mesh while individual voxel noise varies (Fig. 5(a)). Likewise, optical imaging parameters, such as laser power, can be adjusted to create a spectrum of images (Fig. 5(b)).

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. 5(c), 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. 6, 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.

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. B.13 and B.14), we see quantitatively that our architecture is a top performer by almost every computed metric (Tables 2 and 3). 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. More drastic differences, however, can be seen on the real FM datasets. Qualitatively, we see that our architecture better models the reference style while maintaining dendritic structure (Figs. B.15–B.17). These findings are corroborated by quantitative results where we again improve on nearly

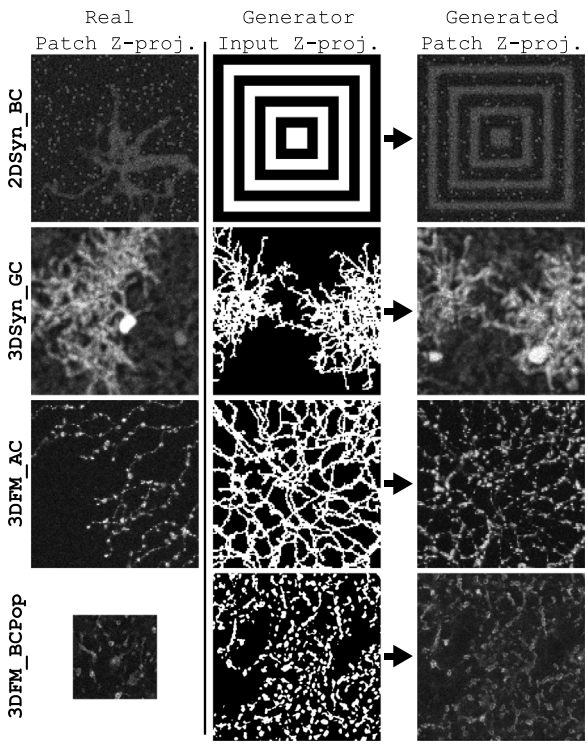

Fig. 6. Demonstration of generator generalizability. The left column shows Z-projections of real patches sampled as style references for all datasets (rows). In the middle column, panels top to down show a generator input of novel geometries, a voxelized mesh Z-projection of a neuron with an additional 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.

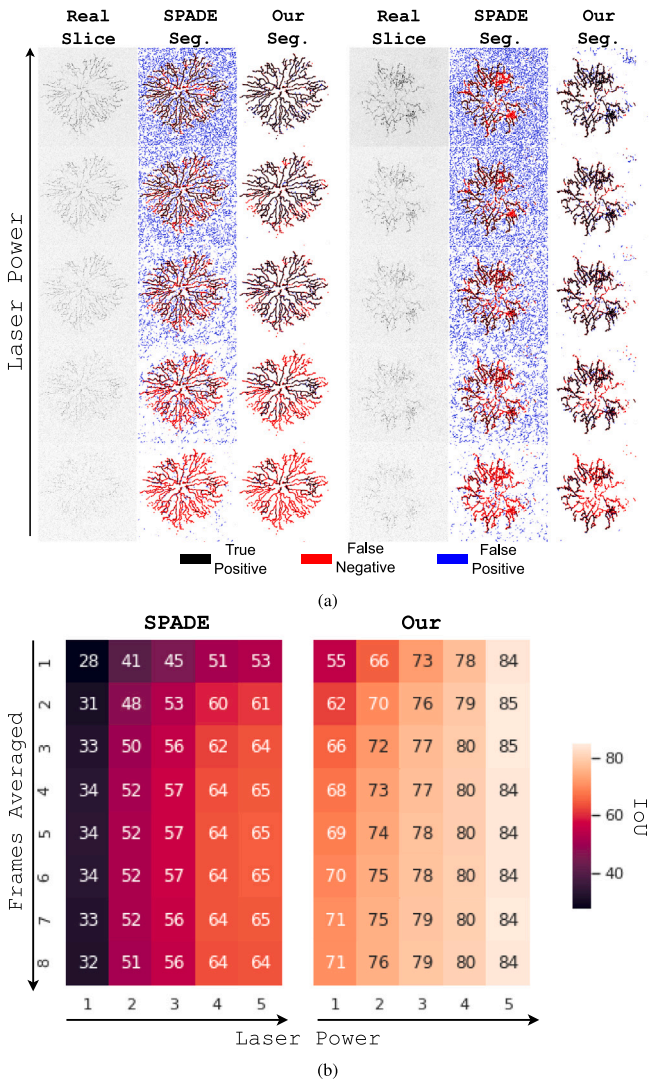

Fig. 7. (a) Segmentation predictions on never-before-seen FM slices (inverted for visual effect) at varying laser powers using generated training datasets from SPADE and our generator. (b) Average intersection of union (IoU) scores across multiple frame averaging and laser powers using generated training datasets from SPADE and our generator. Note that our generated training datasets improve IoU scores nearly 40 points for low signal-to-noise images.

every metric (Tables 2 and 3) 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. 7), 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 between paired and unpaired training paradigms (Table 3), 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. B.18) 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 `Acquisition` 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. 8). As for our `Spatial_stddev` layer, it allows our architecture to better learn the spatial distribution of structured noise. Fig. 9 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 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 additionally training with `Spatial_stddev` when testing different architecture designs that have limited structured noise.

8. Conclusion

In this paper, we have presented an unpaired image translation methodology to generate realistic FM stacks from meshes of previously reconstructed neurons. Our algorithm trained 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 alternative architectures across all datasets. The utility of our pipeline

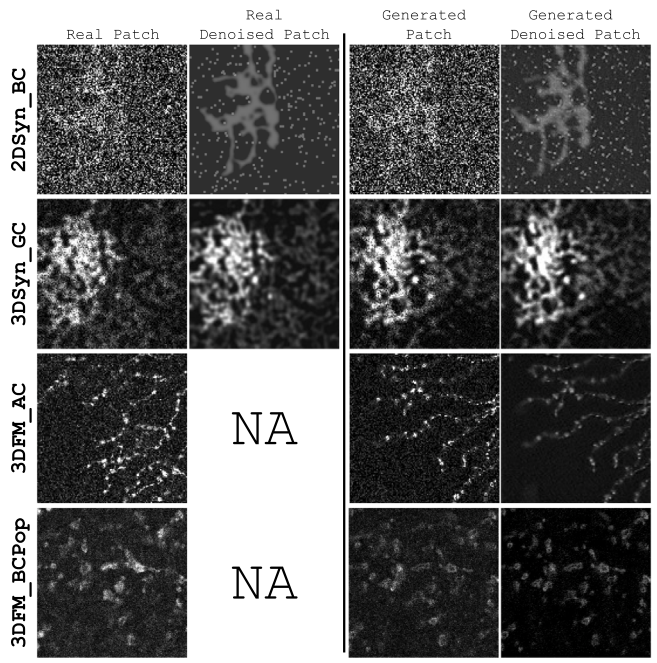

Fig. 8. Demonstration of our algorithm's ability to produce images with novel noise characteristics for all three datasets (rows). Shown are paired noisy and denoised patches from real and generated data. NOTE: there exists no real denoised patch as every frame acquired contains noise.

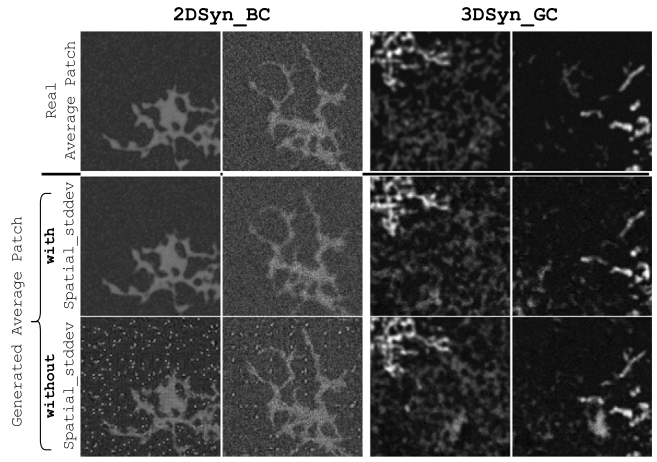

Fig. 9. Comparison of average generated patches with and without `Spatial_stddev` in the discriminator for both synthetic datasets. 20 samples were generated then averaged together. For the Ganglion Cell dataset, stack noise ζ was frozen so that the structure of the dendrites were preserved when frames 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.

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. 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. 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.

Most significantly, once our model is fully trained, it can operate similar to an actual fluorescent 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. While all neurons imaged and modeled in this paper are from the retina, as more EM reconstruction datasets become publicly available, our pipeline can be used for any part of the brain.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Alison Noble, Editorial Board of Medical Image Analysis.

Code and data

Code (https://github.com/MihaelCudic/BioSPADE.git) and data (https://data.mendeley.com/datasets/f6kk4364p4) are made public.

Acknowledgment

This work was supported by the NINDS, USA Intramural Research Program.

Appendix A. 3DFM_AC and 3DFM_BCPop dataset protocol

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 (Ivanova et al., 2010) were intravitreally injected with 2 μL of AAV-7m8-EF1a-BbTagBY virus with a $\backsim 0.8 \times 10e12$ vg/mL titration (Cai et al., 2013). Retinas were then dissected 3–4 weeks after injection, fixed with 4% paraformaldehyde for 45 min, and washed three times. 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 Z-projections for all 22 SACs can be found in Fig. B.10 and an example of the complete spectrum of laser power and frame averaging configurations can be found in Fig. B.12(a). To acquire the 3DFM_BCPop dataset, the retina of a Syt2 mouse was dissect, fixed, and imaged using the same protocol with only a laser power of 15.70 μW and a magnification of 0.25. Samples of the data collected can be seen in Fig. B.11.

Appendix B. Qualitative comparisons

Due to space constraints, a comprehensive qualitative comparison between patches generated by our and competing models are shown here. Figs. B.13 and B.14 show randomly selected patches generated from models trained on 2DSyn_BC and 3DSyn_GC datasets respectively. Then, Figs. B.15, B.16, and B.17 show randomly selected patches generated from models trained on our newly acquired 3DFM_AC and 3DFM_BCPop datasets. The complete spectrum of laser power and frame averaging configurations produced by our generator for a given voxelized mesh patch is shown in Fig. B.12(b).

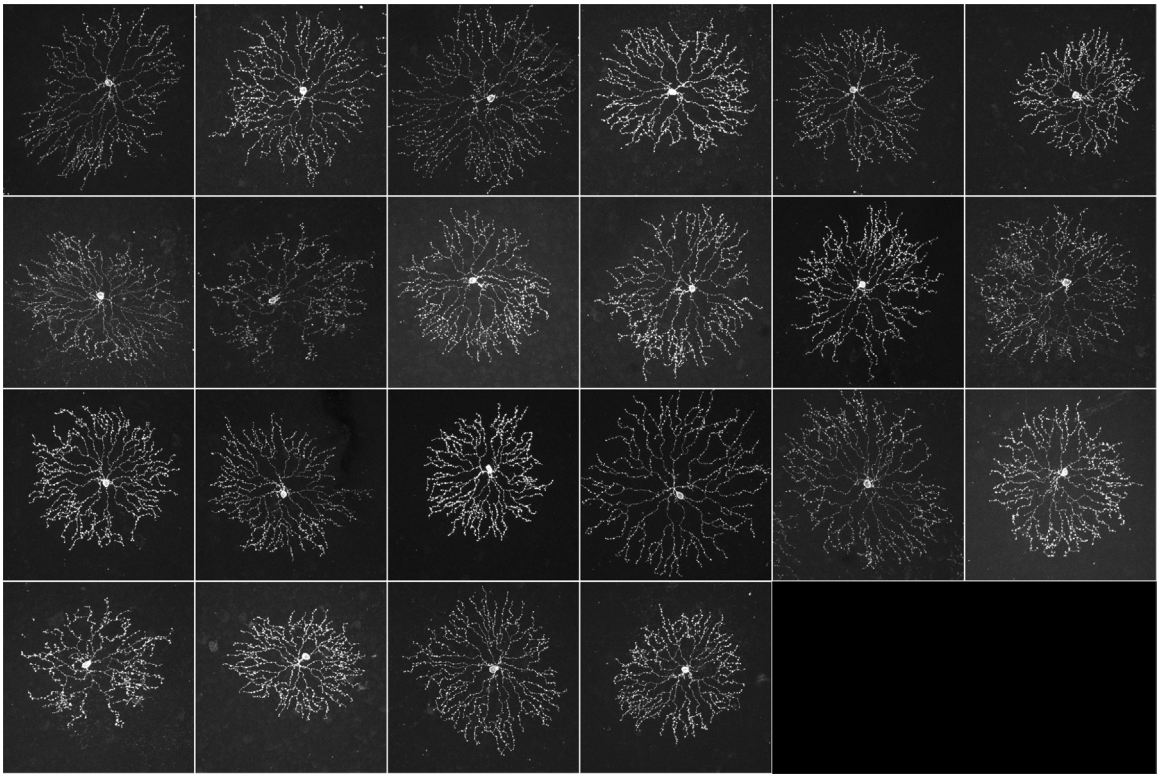

Fig. B.10. Maximum z-projections of all 22 Starburst Amacrine Cells imaged in our new 3DFM_AC dataset.

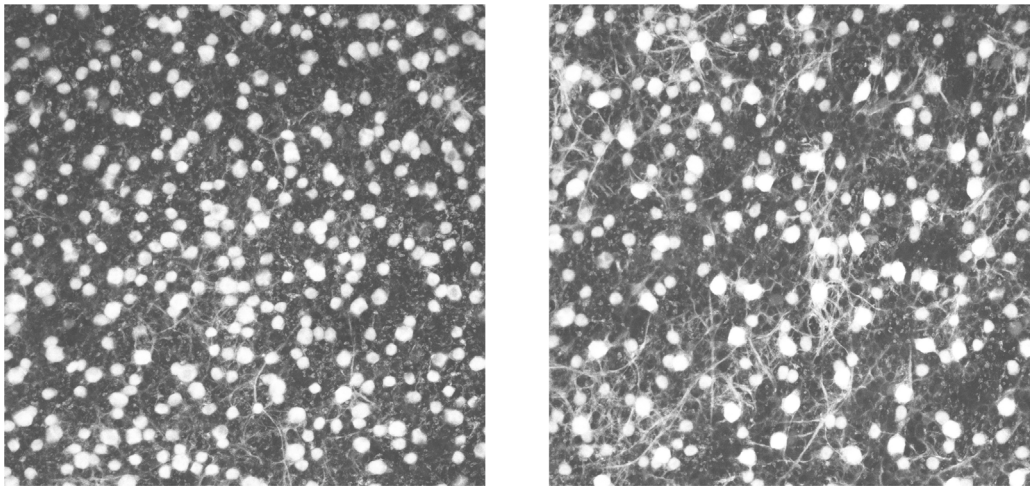

Fig. B.11. Maximum z-projections of all 2 Bipolar Cell populations imaged in our new 3DFM_BCPop dataset.

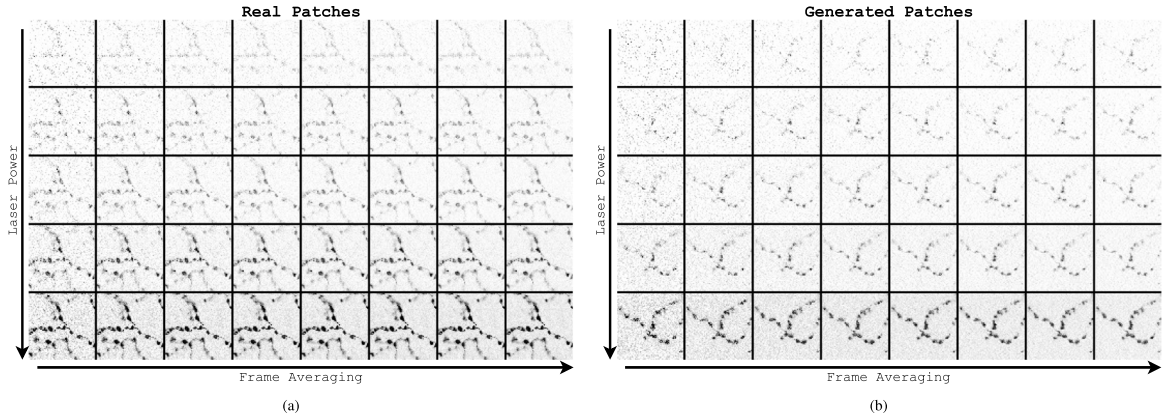

Fig. B.12. (a) Real and (b) generated 64 × 64 patches (inverted for visual effect) sampled from the 3DFM_AC dataset at varying powers and frame averaging. A total of 5 different laser powers and 8 frame averaging combinations were acquired.

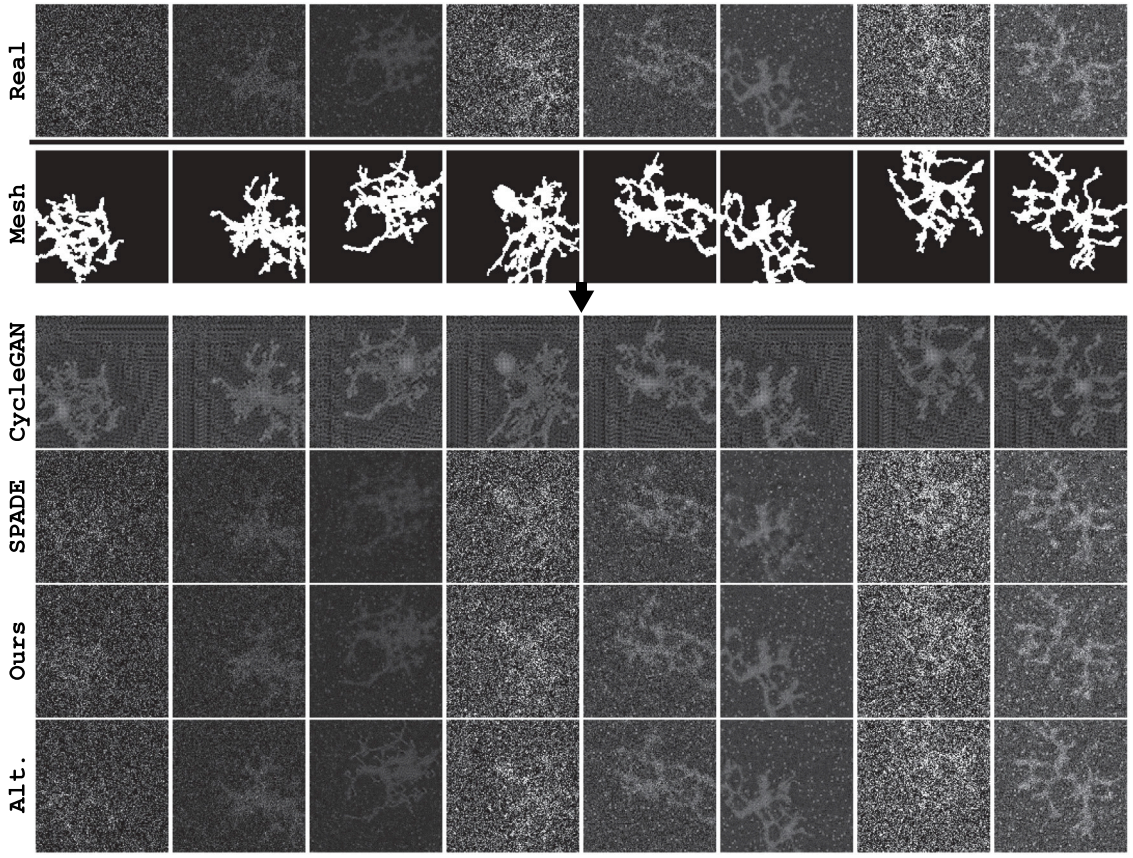

Fig. B.13. Real and generated 128 × 128 patches sampled from the 2DSyn_BC dataset. While unpaired images were used during training, all real and generated patches were created based on the same voxelized mesh to make visual comparisons easier.

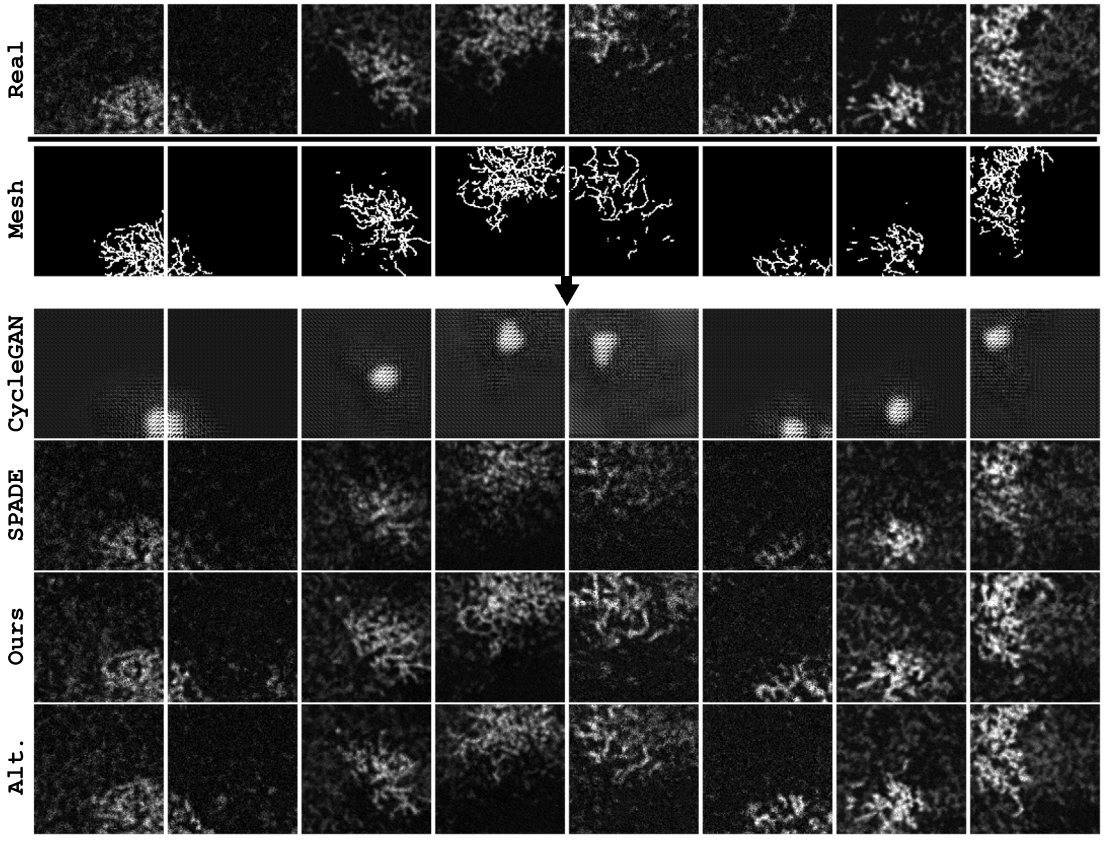

Fig. B.14. Real and generated 1 × 128 × 128 patches sampled from the 3DSyn_GC dataset. While unpaired images were used during training, all real and generated patches were created based on the same voxelized mesh to make visual comparisons easier.

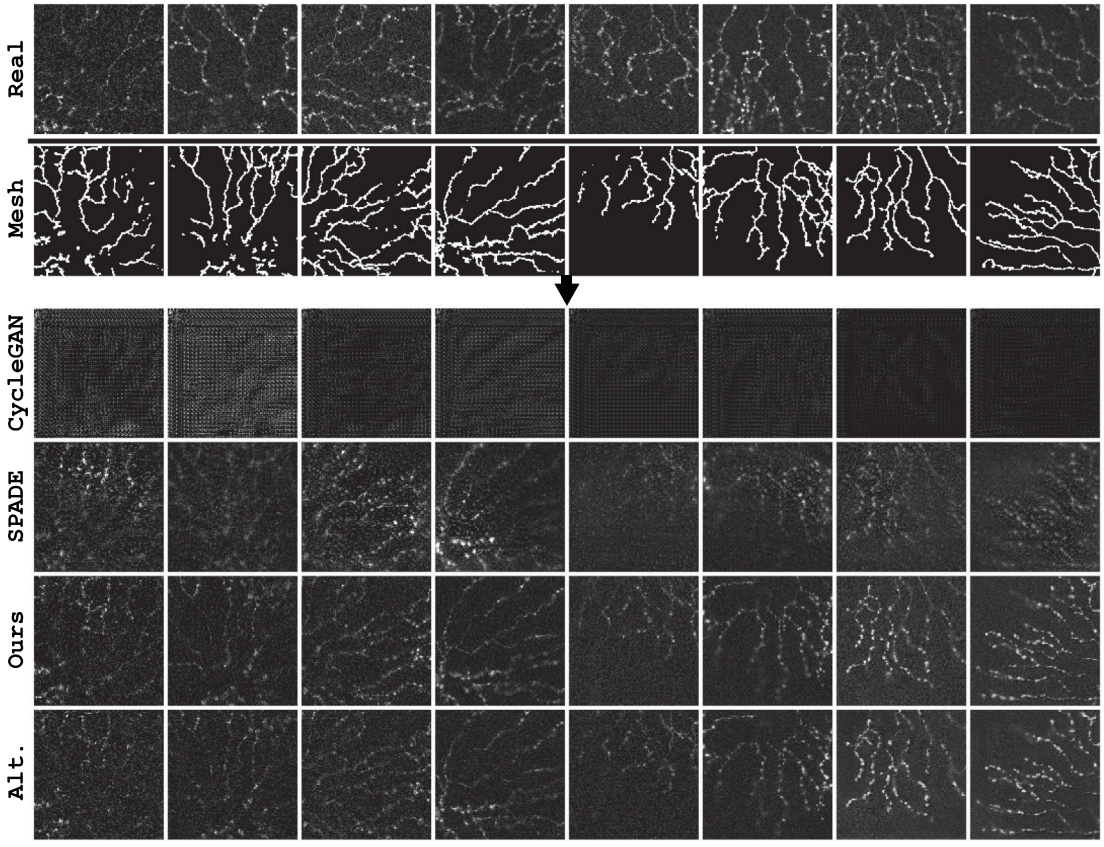

Fig. B.15. Real and generated 1 × 128 × 128 patches sampled from the 3DFM_AC dataset.

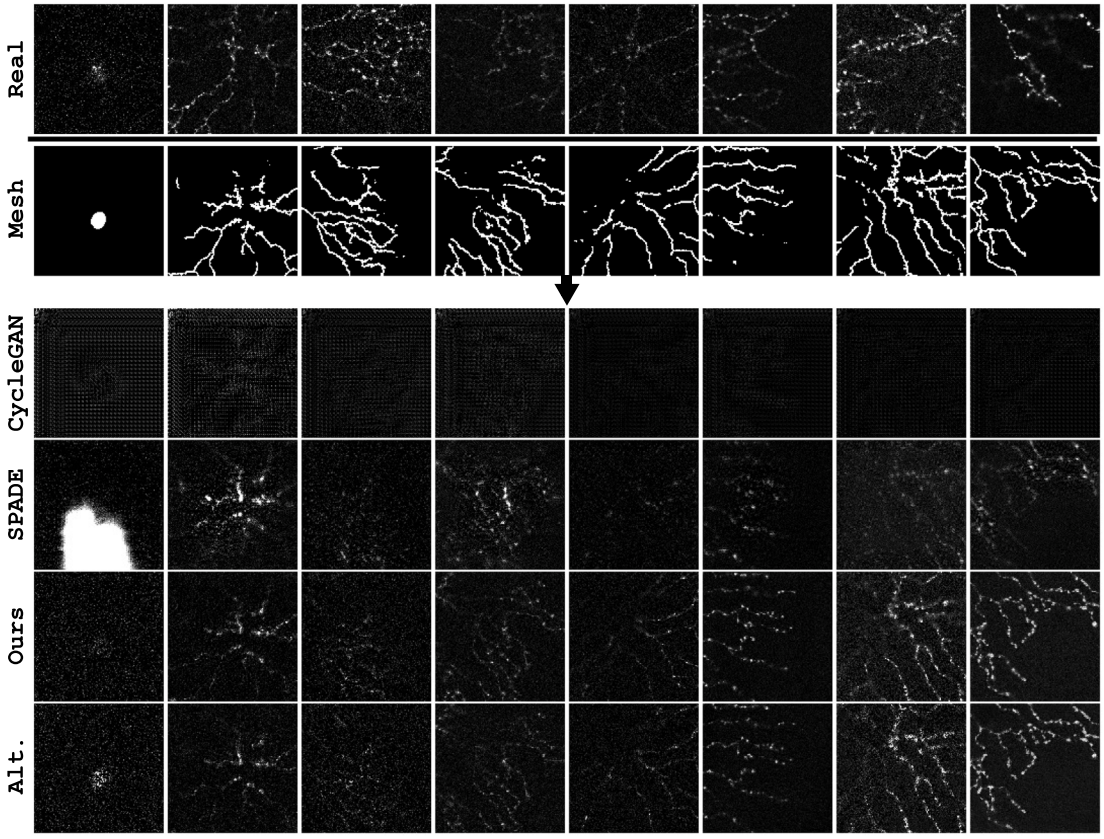

Fig. B.16. Real and generated 1 × 128 × 128 patches sampled from the 3DFM_AC dataset.

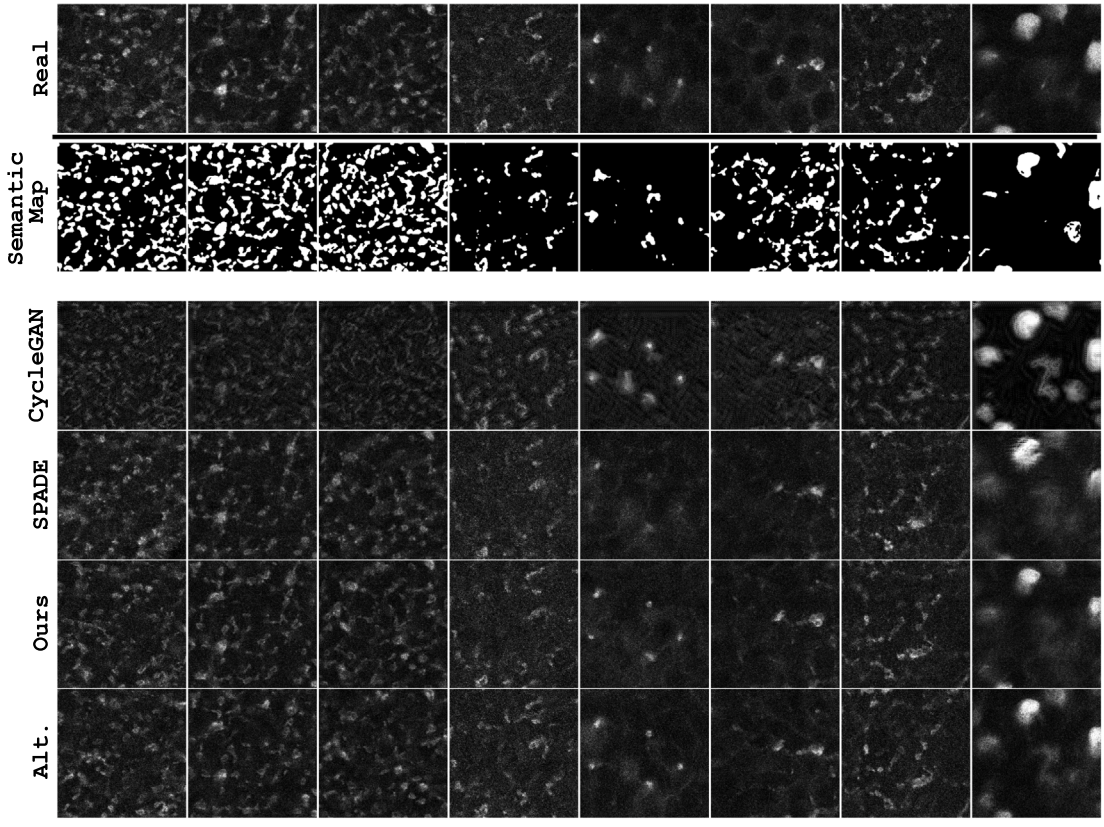

Fig. B.17. Real and generated 1 × 128 × 128 patches sampled from the 3DFM_BCPop dataset.

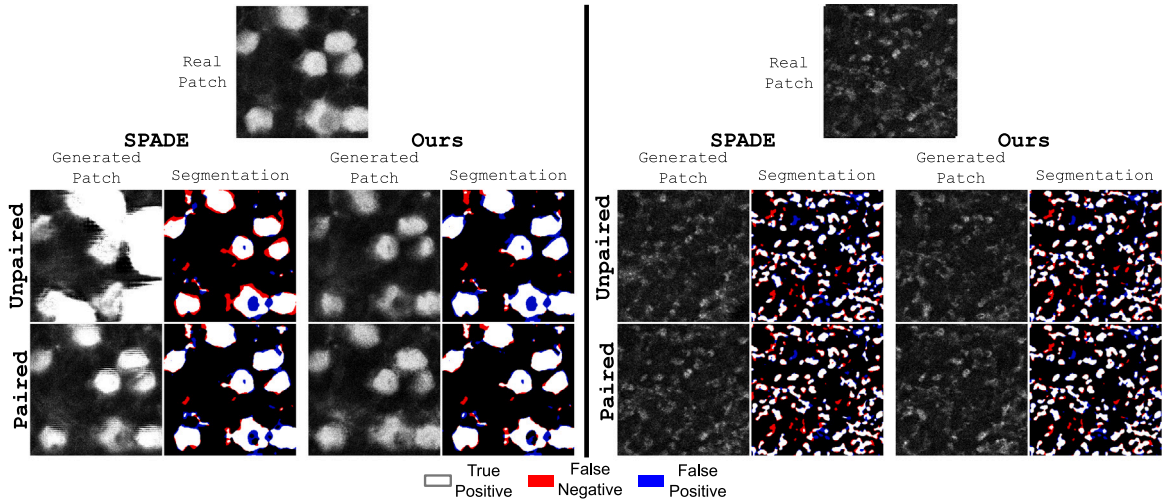

Fig. B.18. Qualitative comparisons between real, SPADE-generated, and Our-generated patches using unpaired and paired training regiments from the 3DFM_BCPop dataset. Generated patches were then used to train a separate machine learning algorithm to perform segmentation. To the right of every generated patch is the predicted segmentation of the real patch from the trained segmentation network.

References

Acciai, L., Soda, P., Iannello, G., 2016. Automated neuron tracing methods: an updated account. Neuroinformatics 14 (4), 353–367.

Apthorpe, N., Riordan, A., Aguilar, R., Homann, J., Gu, Y., Tank, D., Seung, H.S., 2016. Automatic neuron detection in calcium imaging data using convolutional networks. Adv. Neural Inf. Process. Syst. 29.

Armanious, K., Jiang, C., Fischer, M., Küstner, T., Nikolaou, K., Gatidis, S., Yang, B., 2018. MedGAN: Medical image translation using GANs. arXiv preprint arXiv:1806.06397.

Bailo, O., Ham, D., Min Shin, Y., 2019. Red blood cell image generation for data augmentation using conditional generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.

Baniukiewicz, P., Lutton, E.J., Collier, S., Bretschneider, T., 2019. Generative adversarial networks for augmenting training data of microscopic cell images. Front. Comput. Sci. 10.

Beers, A., Brown, J., Chang, K., Campbell, J.P., Ostmo, S., Chiang, M.F., Kalpathy-Cramer, J., 2018. High-resolution medical image synthesis using progressively grown generative adversarial networks. arXiv preprint arXiv:1805.03144.

Bellemo, V., Burlina, P., Yong, L., Wong, T.Y., Ting, D.S.W., 2018. Generative adversarial networks (GANs) for retinal fundus image synthesis. In: Asian Conference on Computer Vision. Springer, pp. 289–302.

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

Bora, A., Price, E., Dimakis, A.G., 2018. AmbientGAN: Generative models from lossy measurements. In: International Conference on Learning Representations.

Brown, K.M., Barrionuevo, G., Canty, A.J., De Paola, V., Hirsch, J.A., Jefferis, G.S., Lu, J., Snippe, M., Sugihara, I., Ascoli, G.A., 2011. The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. Neuroinformatics 9 (2), 143–157.

Cai, D., Cohen, K.B., Luo, T., Lichtman, J.W., Sanes, J.R., 2013. Improved tools for the Brainbow toolbox. Nature Methods 10 (6), 540–547.

Chen, W., Liu, M., Du, H., Radojević, M., Wang, Y., Meijering, E., 2021. Deep-learning-based automated neuron reconstruction from 3D microscopy images using synthetic training images. IEEE Trans. Med. Imaging 41 (5), 1031–1042.

Chernyavskiy, O., Kubínová, L., Mao, X., 2010. Analysis of point spread function degradation in thick tissues. Microsc. Microanal. 16 (S2), 286–287.

Cho, H., Lim, S., Choi, G., Min, H., 2017. Neural stain-style transfer learning using GAN for histopathological images. arXiv preprint arXiv:1710.08543.

Costa, P., Galdran, A., Meyer, M.I., Niemeijer, M., Abràmoff, M., Mendonça, A.M., Campilho, A., 2017. End-to-end adversarial retinal image synthesis. IEEE Trans. Med. Imaging 37 (3), 781–791.

Durugkar, I., Gemp, I., Mahadevan, S., 2016. Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673.

Feng, L., Zhao, T., Kim, J., 2015. neuTube 1.0: a new design for efficient neuron reconstruction software based on the SWC format. Eneuro 2 (1).

Gatys, L.A., Ecker, A.S., Bethge, M., 2016. Image style transfer using convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. CVPR.

Ghosh, S., Preza, C., 2015. Fluorescence microscopy point spread function model accounting for aberrations due to refractive index variability within a specimen. J. Biomed. Opt. 20 (7), 075003.

Goldsborough, P., Pawlowski, N., Caicedo, J.C., Singh, S., Carpenter, A., 2017. CytoGAN: generative modeling of cell images. BioRxiv 227645.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680.

Grienberger, C., Konnerth, A., 2012. Imaging calcium in neurons. Neuron 73 (5), 862–885.

Han, L., Murphy, R.F., Ramanan, D., 2018. Learning generative models of tissue organization with supervised GANs. In: 2018 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE, pp. 682–690.

Haralick, R.M., Shanmugam, K., Dinstein, I.H., 1973. Textural features for image classification. IEEE Trans. Syst. Man Cybern. (6), 610–621.

Helmstaedter, M., Briggman, K.L., Turaga, S.C., Jain, V., Seung, H.S., Denk, W., 2013. Connectomic reconstruction of the inner plexiform layer in the mouse retina. Nature 500 (7461), 168.

Hollandi, R., Szkalisity, A., Toth, T., Tasnadi, E., Molnar, C., Mathe, B., Grexa, I., Molnar, J., Balind, A., Gorbe, M., et al., 2020. nucleAIzer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. Cell Syst. 10 (5), 453–458.

Hong, S., Marinescu, R., Dalca, A.V., Bonkhoff, A.K., Bretzner, M., Rost, N.S., Golland, P., 2021. 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. Springer, pp. 24–34.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1125–1134.

Ivanova, E., Hwang, G.-S., Pan, Z.-H., 2010. Characterization of transgenic mouse lines expressing Cre recombinase in the retina. Neuroscience 165 (1), 233–243.

Izadyyazdanabadi, M., Belykh, E., Zhao, X., Moreira, L.B., Gandhi, S., Cavallo, C., Eschbacher, J., Nakaji, P., Preul, M.C., Yang, Y., 2019. Fluorescence image histology pattern transformation using image style transfer. Front. Oncol. 519.

Johnson, J., Alahi, A., Fei-Fei, L., 2016. Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision.

Kaneko, T., Harada, T., 2020. Noise robust generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8404–8414.

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

Kovalev, V.A., Kruggel, F., Gertz, H.-J., von Cramon, D.Y., 2001. Three-dimensional texture analysis of MRI brain datasets. IEEE Trans. Med. Imaging 20 (5), 424–433.

Kraus, O.Z., Ba, J.L., Frey, B.J., 2016. Classifying and segmenting microscopy images with deep multiple instance learning. Bioinformatics 32 (12), i52–i59.

Li, Q., Shen, L., 2019. 3D neuron reconstruction in tangled neuronal image with deep networks. IEEE Trans. Med. Imaging 39 (2), 425–435.

Li, R., Zeng, T., Peng, H., Ji, S., 2017. Deep learning segmentation of optical microscopy images improves 3-D neuron reconstruction. IEEE Trans. Med. Imaging 36 (7), 1533–1541.

Liimatainen, K., Kananen, L., Latonen, L., Ruusuvuori, P., 2019. Iterative unsupervised domain adaptation for generalized cell detection from brightfield z-stacks. BMC Bioinformatics 20 (1), 80.

Lim, J.H., Ye, J.C., 2017. Geometric gan. arXiv preprint arXiv:1705.02894.

Livet, J., Weissman, T.A., Kang, H., Draft, R.W., Lu, J., Bennis, R.A., Sanes, J.R., Lichtman, J.W., 2007. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. Nature 450 (7166), 56–62.

Matsumoto, B., 2003. Cell Biological Applications of Confocal Microscopy. Elsevier.

Mok, T.C., Chung, A.C., 2018. Learning data augmentation for brain tumor segmentation with coarse-to-fine generative adversarial networks. In: International MICCAI Brainlesion Workshop. Springer, pp. 70–80.

Neyshabur, B., Bhojanapalli, S., Chakrabarti, A., 2017. Stabilizing gan training with multiple random projections. arXiv preprint arXiv:1705.07831.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987.

Osokin, A., Chessel, A., Carazo Salas, R.E., Vaggi, F., 2017. GANs for biological image synthesis. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2233–2242.

Pankajakshan, P., Blanc-Féraud, L., Kam, Z., Zerubia, J., 2009. Point-spread function retrieval for fluorescence microscopy. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE, pp. 1095–1098.

Park, T., Liu, M.-Y., Wang, T.-C., Zhu, J.-Y., 2019. Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2337–2346.

Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E., Ascoli, G.A., 2015. BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images. Neuron 87 (2), 252–256.

Peterka, D.S., Takahashi, H., Yuste, R., 2011. Imaging voltage in neurons. Neuron 69 (1), 9–21.

Radojević, M., Meijering, E., 2019. Automated neuron reconstruction from 3D fluorescence microscopy images using sequential Monte Carlo estimation. Neuroinformatics 17 (3), 423–442.

Ren, J., Hacihaliloglu, I., Singer, E.A., Foran, D.J., Qi, X., 2019. Unsupervised domain adaptation for classification of histopathology whole-slide images. Front. Bioeng. Biotechnol. 7.

Rhodes, K.J., Trimmer, J.S., 2006. Antibodies as valuable neuroscience research tools versus reagents of mass distraction. J. Neurosci. 26 (31), 8017–8020.

Shariff, A., Murphy, R.F., Rohde, G.K., 2010. A generative model of microtubule distributions, and indirect estimation of its parameters from fluorescence microscopy images. Cytometry A: J. Int. Soc. Adv. Cytom. 77 (5), 457–466.

Shin, H.-C., Tenenholtz, N.A., Rogers, J.K., Schwarz, C.G., Senjem, M.L., Gunter, J.L., Andriole, K.P., Michalski, M., 2018. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In: International Workshop on Simulation and Synthesis in Medical Imaging. Springer, pp. 1–11.

Sorokin, D.V., Peterlík, I., Ulman, V., Svoboda, D., Nečasová, T., Morgaenko, K., Eiselleová, L., Tesařová, L., Maška, M., 2018. FiloGen: a model-based generator of synthetic 3-D time-lapse sequences of single motile cells with growing and branching filopodia. IEEE Trans. Med. Imaging 37 (12), 2630–2641.

Sümbül, U., Roossien, D., Cai, D., Chen, F., Barry, N., Cunningham, J.P., Boyden, E., Paninski, L., 2016. Automated scalable segmentation of neurons from multispectral images. In: Advances in Neural Information Processing Systems. pp. 1912–1920.

Svoboda, D., Ulman, V., 2016. MitoGen: a framework for generating 3D synthetic time-lapse sequences of cell populations in fluorescence microscopy. IEEE Trans. Med. Imaging 36 (1), 310–321.

Svoboda, K., Yasuda, R., 2006. Principles of two-photon excitation microscopy and its applications to neuroscience. Neuron 50 (6), 823–839.

Vasilkoski, Z., Stepanyants, A., 2009. Detection of the optimal neuron traces in confocal microscopy images. J. Neurosci. Methods 178 (1), 197–204.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B., 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8798–8807.

Wilt, B.A., Burns, L.D., Wei Ho, E.T., Ghosh, K.K., Mukamel, E.A., Schnitzer, M.J., 2009. Advances in light microscopy for neuroscience. Annu. Rev. Neurosci. 32, 435–506.

Wu, J., He, Y., Yang, Z., Guo, C., Luo, Q., Zhou, W., Chen, S., Li, A., Xiong, B., Jiang, T., et al., 2014. 3D BrainCV: simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution. Neuroimage 87, 199–208.

Xiao, H., Peng, H., 2013. APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. Bioinformatics 29 (11), 1448–1454.

Yang, J., Hao, M., Liu, X., Wan, Z., Zhong, N., Peng, H., 2019. FMST: an automatic neuron tracing method based on fast marching and minimum spanning tree. Neuroinformatics 17 (2), 185–196.

Yang, H., Sun, J., Carass, A., Zhao, C., Lee, J., Xu, Z., Prince, J., 2018. Unpaired brain MR-to-CT synthesis using a structure-constrained CycleGAN. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 174–182.

Zhao, J., Chen, X., Xiong, Z., Liu, D., Zeng, J., Xie, C., Zhang, Y., Zha, Z.-J., Bi, G., Wu, F., 2020. Neuronal population reconstruction from ultra-scale optical microscopy images via progressive learning. IEEE Trans. Med. Imaging 39 (12), 4034–4046.

Zhou, Z., Kuo, H.-C., Peng, H., Long, F., 2018. DeepNeuron: an open deep learning toolbox for neuron tracing. Brain Inform. 5 (2), 1–9.

Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232.