

**NEURAL NETWORK-BASED MICROSCOPE DEFOCUS
CORRECTION WITH POINT SPREAD FUNCTION RECOVERY**

Sheila Werth, Alex Matlock



Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

August 30, 2018

Technical Report No. ECE-2018-2

Contents

1	Introduction and Problem Statement	1
2	Inverse Problems, Analytical Reconstruction Methods, and Neural Networks	1
3	Project Implementation	2
3.1	Optical Imaging Model	2
3.2	Training Datasets	4
3.3	Neural Network	4
3.4	Inverse Problem Formulation	5
3.5	Notation	5
3.6	Operator \mathcal{B} : Imposing Circular Symmetry	6
3.7	Direct Estimation of \mathbf{x}	7
3.8	Estimation of $\hat{\mathbf{k}}(\omega)$ in the Fourier Space	7
4	Experimental Results	8
4.1	Neural Network Results	8
4.2	Network Failure Cases	11
4.3	PSF Recovery	12
5	Discussion and Conclusion	14

List of Figures

1	Imaging System Overview, Forward Model, and Fluorescent Microscope Images Under In-Focus and Defocused Imaging Conditions. Images of the microscope and microscope Point Spread Function (PSF) adapted from [15].	3
2	Modified U-Net Architecture Employed for Training on both Synthetic and Experimental Datasets.	5
3	Heat Map Illustrating each Pixel Value's Distance from the Image Center.	6
4	Linear Interpolation Illustration used for Weight Computation in Matrix \mathcal{B} . A pixel d distance from the image center will be the weighted sum of the two values in \mathbf{x} at distances d_i and d_{i+1}	7
5	Example neural network output images from networks A-F training schemes trained on synthetic data (left) and experimental data (right). The top row shows the in-focus ground truth image at $0\mu m$ defocus and input defocused images. The subsequent image rows show the network outputs for each trained network. All images are shown in the same scaling.	9
6	SSIM (Left) and PSNR (Right) Results Comparing the Network outputs and In-Focus Ground Truth Images. These values were averaged over 25 images from both the experimentally-trained (Top) and synthetically-trained (Bottom) neural networks.	10
7	Example Network Outputs and Evaluation Metrics from the Synthetic Data-Trained Network F on Images Outside The Defocus Training Region.	11
8	A Blurry Stock Photo Deblurred Using the Multi-Focus Synthetic Data-Trained Network F [19].	12
9	Synthetically Defocused Microscope Images (Left), the estimated PSF between these images and the in-focus images (Middle), compared with the ground truth PSF (Right).	12
10	Real Defocused Microscope Images (Left) and their Corresponding PSF Estimates (Right).	13

1 Introduction and Problem Statement

A common problem inherent to optical microscopy is that of image blur from a defocused microscope. When an object exists outside a microscope's depth-of-field (DOF), or the region over which an imaging system perceives an object with minimal aberrations, the additional light propagation from the object to the microscope image plane introduces blurring and a loss of resolution to the final image. This blur effectively results from the spatially-variant nature of the microscope's Point Spread Function (PSF) along the axial direction as shown in Figure 1. This PSF can be predicted and removed with a well-characterized microscope and analytically-based deconvolution methods [11, 15], but this approach is not feasible for commercial microscopes with proprietary design parameters. When considering time-sensitive or non-repeatable biological measurements with such systems, the inability to deconvolve defocus blurred images can impair biology researchers and diagnosticians from evaluating their specimens. This limitation thus creates a need for a fast, simple tool that corrects image defocus without requiring prior knowledge of the imaging system [16]. We propose using convolutional neural networks for deblurring images taken under these defocus conditions and utilizing analytical-based minimization techniques for recovering the imaging system's unknown physical parameters.

2 Inverse Problems, Analytical Reconstruction Methods, and Neural Networks

The general problem of image deblurring is well-characterized and traditionally solved using analytical methods leveraging the physical understanding of the underlying system and various regularization mechanisms. Prior knowledge of the system's physical constraints enables a simulation of the forward model to be used during the image restoration process that restricts the overall solution space to physically relevant results with a minimized risk of overfitting. These methods can have closed-form or iterative solutions using simple methods such as the moore-penrose pseudoinverse and Tikhonov regularization or modern optimization algorithms implementing FISTA or ADMM, respectively [2, 1]. The latter methods are advantageous as they allow additional denoising priors such as sparsity or total variation to be implemented for further constraining the inverse problem. These methods are not immediately available for this project, however, due to our assumption that the system's forward model is an unknown parameter unavailable to the user.

Deep Neural Networks (DNNs) have been used as an alternative approach to solving such inverse deblurring problems. Unlike traditional analytical methods, DNNs use large training datasets to learn optimal weights over many 'deep' convolution layers for achieving a specific task. When trained properly, the DNN acts as the inverse operator for correcting the specific undesirable aberration in the input image. DNNs have been implemented for numerous deblurring applications including motion and defocus correction [18, 6]. The basic structural elements and breadth of architectures that have already been applied to biological applications are reviewed in [13].

While DNNs exhibit impressive visual results, purely network based reconstructions are inherently difficult to trust. The "black box" learning approach of DNNs can introduce artificial features to the output image that could cause misdiagnoses when used in biology for diagnostics or cell morphology applications. These artifacts can be mitigated through large training datasets, but this is computationally expensive and not feasible in many biological situations with sparse datasets. For these reasons, groups have begun investigating hybrid inverse problems combining neural networks with well-established analytical methods and optimization-based techniques.

In these combined approaches, the use of DNNs depends on the research groups' trust in the network output. In minimal trust cases, neural networks are used purely for denoising purposes, as shown by Jin *et al.* where the network is trained to correct artifacts introduced from the object reconstruction [10]. Work by Dong *et al.* and Gupta *et al.* places more trust in the network and use CNNs as a denoising prior when solving the inverse problem with ADMM [8, 5]. These works train the network for projecting the estimate from gradient descent and other iterative methods onto a

convex set, thereby utilizing the network’s denoising capabilities for improving the object estimate when solving the inverse problem. Others have explored this as well in other imaging applications such as Magnetic Resonance Imaging and for generating a universal network-based inverse problem solver [17, 3].

Very recently, work by Yang *et al.* and Jiang *et al.* have investigated microscope defocus correction addressing the same need investigated by this project [9, 20]. These purely DNN-based techniques apply a network for image classification rather than image restoration to determine a microscope’s defocus position during the image acquisition process. This classification enables real-time updating of the microscope’s physical position for correcting the defocused image without possibly introducing artifacts from network-based image restoration methods [9, 20]. We note these works provide more trust-worthy images by correcting the imaging system during acquisition, but we differentiate our work by enabling the user to deblur out-of-focus images taken retro-actively using DNN-based image restoration.

3 Project Implementation

Our project uses a hybrid DNN and optimization-based approach for recovering in-focus microscope images from defocused images taken by a microscope with unknown physical parameters. We place significant trust on an encoder-decoder neural network for directly refocusing the microscope images using an open source dataset of microscope images discussed in further detail below. We use a modified U-Net architecture, shown in Figure 2, for the refocusing process and use the outputs from the network combined with the input defocused image to estimate the unknown system PSF using optimization-based methods.

In the first portion of this section we provide a brief overview of the optics behind microscopy describing the forward model, followed by a description of the dataset used for training the neural network, the network itself, and finally the inverse problem formulations used for estimating the imaging system’s PSF. We subsequently discuss the results from this project in the next section on both the network and PSF estimation process and make final comments on the project’s limitations and future steps in the conclusion.

3.1 Optical Imaging Model

A diagram of the optical imaging system and forward model can be seen in Figure 1. We assume that we have a commercial, diffraction-limited microscope with circular optical elements and minimal optical aberrations imaging fluorescent cell samples based on the chosen BBBC006v1 fluorescent microscopy dataset from the Broad Bioimage Benchmark Collection [12]. Without optical aberrations, we assume the microscope acts as a linear shift-invariant imaging system in the transverse direction and is shift-variant with a changing PSF at different defocus planes along the axial direction. This shift-variant PSF for different focal planes can be seen in Figure 1, where the sidelobes from the PSF increase in energy with increasing defocus distance and result in resolution loss and image blur. From evaluating fluorescent microscope images, we may also assume the system follows an incoherent imaging model. The level of coherence of an imaging system refers to the phase behavior of the observed electromagnetic fields captured by the imaging system. Because CCDs, CMOS, and other optical sensors are too slow for measuring the phase and amplitude of an optical field directly, the resulting image at the camera plane is the time-averaged intensity of the observed optical fields filtered by the microscope PSF. When the observed fields are coherent and in-phase with each other, the resulting intensity from a single plane in the object is a direct convolution of the complex fields convolved with the system’s coherent PSF:

$$g(x, y) = |f(x, y) \otimes PSF(x, y)|^2, \quad (1)$$

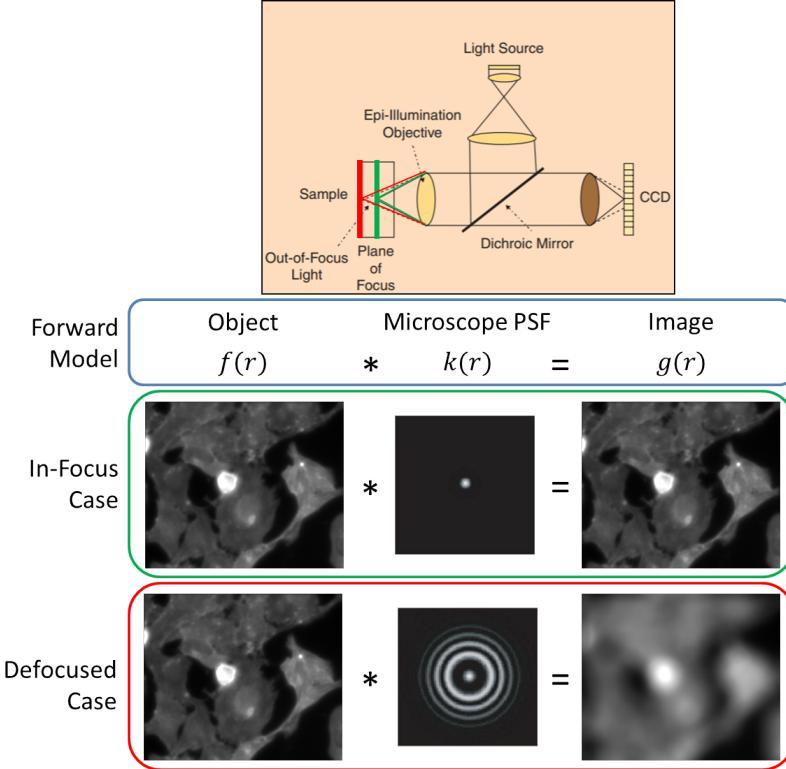


Figure 1: Imaging System Overview, Forward Model, and Fluorescent Microscope Images Under In-Focus and Defocused Imaging Conditions. Images of the microscope and microscope Point Spread Function (PSF) adapted from [15].

where f is the observed field from the object, PSF is the system PSF, and g is the observed intensity. The PSF for coherent imaging directly relates to the maximum acceptance angle, or numerical aperture (NA), of the imaging optics. Given circular optical elements with no aberrations, the PSF at the in-focus plane is the inverse Fourier transform of a circular low-pass filter with a cutoff spatial frequency based on the microscope NA:

$$PSF(x, y) = \mathcal{F}^{-1}\{P(\nu_x, \nu_y)\}, \quad (2)$$

$$P(\nu_x, \nu_y) = \begin{cases} 1 & \sqrt{\nu_x^2 + \nu_y^2} \leq \frac{NA}{\lambda} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where λ is the optical wavelength, P denotes the filter function in the Fourier space, and ν_x, ν_y denote the transverse spatial frequencies. When defocused, a propagation factor describing the light diffraction over the defocus distance is convolved with the system PSF, shown here as a product in the Fourier space:

$$PSF(x, y) = \mathcal{F}^{-1}\{P(\nu_x, \nu_y)W(\nu_x, \nu_y, z_d)\}, \quad (4)$$

$$W(\nu_x, \nu_y, z_d) = e^{j2\pi\lambda z_d[1 + \frac{1}{4\pi}(\nu_x^2 + \nu_y^2)]}, \quad (5)$$

where we use d to denote the defocus condition. In the coherent case, this PSF becomes complex with defocus due to the propagation term W and is difficult to recover without further approximations. When the observed fields from the object are out-of-phase and incoherent with each other, however, the time-averaged intensity at the image plane results in a removal of any interference between the

observed fields and results in an intensity-based convolution of the microscope PSF with the object field:

$$g(x, y) = |f(x, y)|^2 \otimes |PSF(x, y)|^2. \quad (6)$$

A full derivation behind this incoherent imaging system model can be found elsewhere [7]. The imaging system considered here assumes the model shown above, as the fluorescing particles at each point in the image dataset emit light independently of one another and thus have no phase relation. We thus aim to recover this intensity-based incoherent PSF, which we define with simplified notation as the blurring kernel $k(x, y)$:

$$k(x, y) = |PSF(x, y)|^2, \quad (7)$$

matching our forward model displayed in Figure 1. Since this kernel is fundamentally a convolution of two circular filters in the Fourier domain, we can assume this kernel has compact support in the Fourier space, is isotropic in the spatial and Fourier domains, is non-negative, and will resemble a bessel function in the spatial domain. This model and these assumptions were our basis for simulating our own blurred dataset and recovering an estimate of the blurring kernel $k(x, y)$ in this project.

3.2 Training Datasets

As previously mentioned, we utilized the BBBC006v1 dataset for training our network to refocus images captured at a given defocus plane [12]. This dataset contains images of human U2OS osteosarcoma epithelial cells labeled with Hoechst 33342 and Phalloidin fluorescent dyes taken at in-focus and out-of-focus positions using a fluorescent microscope. A total of 768 fields of view containing different U2OS cells are available with 32 image sets taken at defocus planes spanning $\pm 32\mu m$ with $2\mu m$ axial stepsize between each defocus image set. For the current investigation, we use images obtained only from the phalloidin fluorophores staining the cell membrane. The phalloidin fluorescent images provided fine detail and high-frequency features from the cell over the Hoeschst stained images and were selected for adequately testing the network's refocusing capabilities. We had a total of 768 image pairs for our training network, and the networks discussed below used an approximate 80 : 20 split of training to testing data, respectively.

For evaluating our recovery of $k(x, y)$, we required a secondary dataset for training the neural network with a known blurring kernel so that we had a ground truth kernel when evaluating our estimated PSF. Based on the previously described forward model, we simulated a new dataset of blurred images over a range of defocus positions from the in-focus images in the experimental dataset. This system was assumed to have 20x magnification, a 0.25 NA maximum acceptance angle, and a pixel spacing of $6\mu m$ at the camera plane based on common biological imaging microscope parameters. We also filtered the in-focus images from the experimental dataset with the incoherent PSF without defocus blur to generate new in-focus images for the simulated microscope system. This step was taken to make sure the network was not possibly learning higher resolution features from the experimental dataset that would not be present in the simulated system with a maximum frequency bandwidth of 0.25 NA. Simulated defocus images were generated for 2-10um in stepsizes of $2\mu m$, $15\mu m$ - $30\mu m$ in steps of $5\mu m$, and a $40\mu m$ defocus with significant blurring where network recovery of an in-focus image was expected to fail.

3.3 Neural Network

The neural network used for image deblurring is a modified version of the U-Net DNN proposed by Ronneberger *et al.* (Fig. 2) [14]. The original U-Net architecture was an image classification network for segmenting a biological image of cells into cell and non-cell regions. Since its original publication, this architecture has been used extensively in other classification and image restoration tasks and was chosen based on these other uses. This architecture features an encoder-decoder network with ten

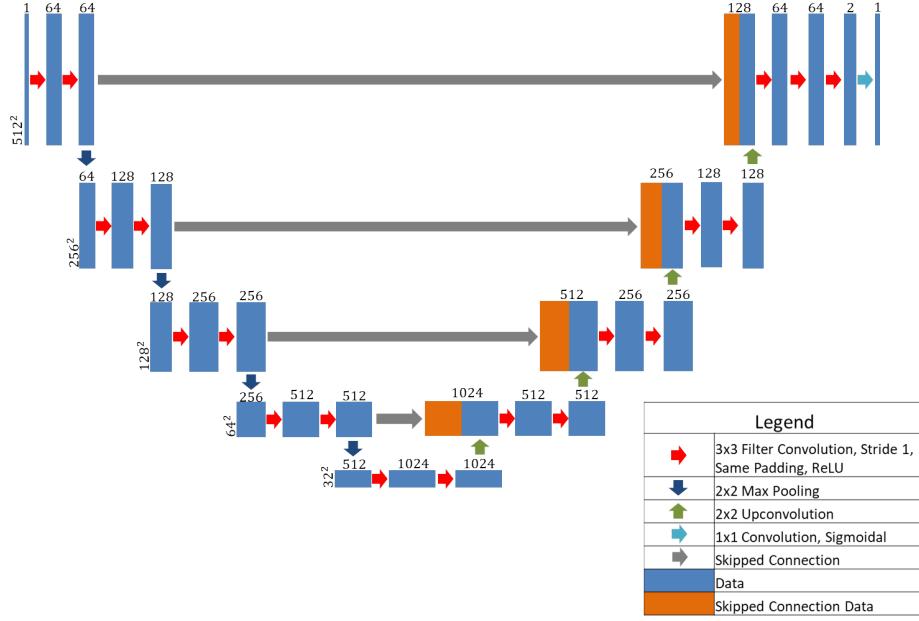


Figure 2: Modified U-Net Architecture Employed for Training on both Synthetic and Experimental Datasets.

convolution layers using rectified linear unit (ReLU) activation on the encoding and decoding sides with 2x2 max pooling for reducing the image size during the encoding process and 2x2 upconvolution for restoring image size in the decoding process. This encoder-decoder scheme enables the initial shallow convolutional layers to learn localized feature maps while the deep layers following max pooling should correspond to global features in the downsampled image. We hoped that this design would learn a generalized version of the blurring kernel at these deep convolutional layers in the U-Net structure and provide edge-preservation from the network’s upper layers. From the original U-Net, we modified the architecture by preserving the image size following each convolution step and introduced an additional convolution layer at the end of the decoding process for removing the pixel-wise segmentation process of the original U-Net.

3.4 Inverse Problem Formulation

A stated objective of this project is to recover the unknown blurring kernel from the microscope using the network refocused and blurred microscope image. For this task, we utilized the assumption that the circular optical elements generate an isotropic PSF to recover a one-dimensional PSF estimate, \mathbf{x} . This choice was made for easing the computational complexity of the inverse problem. We design a linear operator \mathbf{B} to map this one-dimensional PSF representation into the two dimensional isotropic space. Initially, we attempted to directly estimate the one-dimensional representation of the PSF. However, to reduce computational burden, we eventually opted for two-dimensional PSF recovery in the Fourier space. In this way, we estimate the two-dimensional PSF and later use the operator \mathbf{B} to project the estimate into the space of isotropic images. This section will introduce notation for our problem, the design of operator \mathbf{B} , and our two different reconstruction methods.

3.5 Notation

Figure 1 shows the blurring forward model of the system. In vector format, the in-focus image \mathbf{f} is convolved with the linear shift invariant PSF \mathbf{k} to produce defocused image \mathbf{g} . This may be written

as follows:

$$\mathbf{g} = \mathbf{f} * \mathbf{k}, \quad (8)$$

where $\mathbf{k} \in \mathbb{R}^{N^2}$, $\mathbf{f} \in \mathbb{R}^{M^2}$, and $\mathbf{g} \in \mathbb{R}^{(M+N-1)^2}$. This may also be written in matrix vector form:

$$\mathbf{g} = \mathbf{A}\mathbf{k} \quad (9)$$

Implementing this convolution in the spatial domain involves forming matrix $\mathbf{A} \in \mathbb{R}^{(M+N-1)^2 \times M^2}$, which becomes prohibitively large with relatively small sizes of \mathbf{k} and \mathbf{g} . For a faster implementation, a circulant assumption may be made. Under this assumption, \mathbf{A} becomes square and \mathbf{k}, \mathbf{f} , and \mathbf{g} become the same length. In this case, the convolution may be written as pointwise multiplication in the frequency domain:

$$\hat{\mathbf{g}}(\omega) = \hat{\mathbf{f}}(\omega)\hat{\mathbf{k}}(\omega) \quad (10)$$

3.6 Operator \mathbf{B} : Imposing Circular Symmetry

Fortunately, we have prior knowledge about \mathbf{k} to help in the estimation process. One of the strongest pieces of information available is the fact that both \mathbf{k} and $\hat{\mathbf{k}}(\omega)$ are isotropic images. This means that two-dimensional \mathbf{k} and $\hat{\mathbf{k}}(\omega)$ may be completely described by one-dimension. Each pixel value only depends on its distance from the center of the image. We introduce an operator \mathbf{B} to map one-dimensional vector $\mathbf{x} \in \mathbb{R}^{M/2}$ into the space of isotropic images:

$$\mathbf{k} = \mathbf{B}\mathbf{x} \quad (11)$$

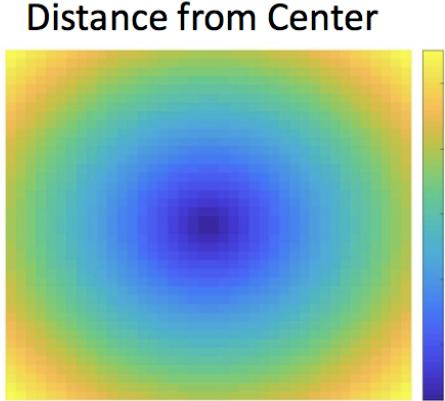


Figure 3: Heat Map Illustrating each Pixel Value's Distance from the Image Center.

Linear interpolation was used to calculate the weights in \mathbf{B} . First, the distance of each pixel from the center d in the isotropic image \mathbf{k} was computed, as shown in Figure 3. If this distance d is greater than the vector length \mathbf{x} , as is the case for image corners, then the corresponding row in \mathbf{B} is set to zero. For the remaining pixels, the output pixel value is the weighted sum of the nearest two elements in vector \mathbf{x} at distances d_i and d_{i+1} , as shown in Figure 4. If $\Delta d = d - d_i$, the weights in \mathbf{B} are $w_1 = (1 - \Delta d)$ and $w_2 = \Delta d$. These weights are placed in the columns of \mathbf{B} corresponding to the elements of \mathbf{x} at distances d_i and d_{i+1} .

We note here that multiplying any two-dimensional \mathbf{k} by the projection operator $(\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T$ projects \mathbf{k} into an isotropic image space.

3.7 Direct Estimation of \mathbf{x}

Originally, we hoped to estimate and impose priors directly on one-dimensional \mathbf{x} . Using \mathbf{B} , we write the following least squares formulation in space and frequency:

$$\mathbf{x}_g = \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{g} - \mathbf{A}\mathbf{B}\mathbf{x}\|_2^2 \quad (12)$$

and frequency:

$$\hat{\mathbf{x}}_g(\omega) = \min_{\hat{\mathbf{x}}(\omega)} \frac{1}{2} \|\hat{\mathbf{g}}(\omega) - \hat{\mathbf{A}}(\omega)\mathbf{B}\hat{\mathbf{x}}(\omega)\|_2^2 \quad (13)$$

The corresponding closed form solutions are:

$$\mathbf{x}_g = (\mathbf{B}^H \mathbf{A}^H \mathbf{A}\mathbf{B})^{-1} \mathbf{B}^H \mathbf{A}^H \mathbf{g} \quad (14)$$

and

$$\hat{\mathbf{x}}_g(\omega) = (\mathbf{B}^H \hat{\mathbf{A}}(\omega)^H \hat{\mathbf{A}}(\omega)\mathbf{B})^{-1} \mathbf{B}^H \hat{\mathbf{A}}(\omega)^H \hat{\mathbf{g}}(\omega) \quad (15)$$

While a direct implementation of the one-dimensional \mathbf{x} or $\hat{\mathbf{x}}(\omega)$ is ideal, this method is too computationally expensive for applying to the full microscope image size. For an efficient implementation, we instead implemented the convolution in the Fourier space where the circular convolution can be performed easily with point-wise multiplication of the matrices. In this approach, a two-dimensional estimate is formed initially, projected onto the space of isotropic images using operator \mathbf{B} , and then expanded to a two-dimensional estimate again. This process averages out image noise and results in a cleaner PSF estimate.

3.8 Estimation of $\hat{k}(\omega)$ in the Fourier Space

Considering the circular convolution in \mathbf{A} with $\mathbf{B}\hat{\mathbf{x}}$, we implement convolution in the Fourier domain. For this purpose, a two-dimensional estimate of $\hat{\mathbf{C}}(\omega) = \mathbf{B}\hat{\mathbf{x}}(\omega)$ is formed first using the least squares formulation:

$$\hat{\mathbf{C}}_g(\omega) = \min_{\hat{\mathbf{C}}(\omega)} \frac{1}{2} \|\hat{\mathbf{g}}(\omega) - \hat{\mathbf{A}}(\omega)\hat{\mathbf{C}}(\omega)\|_2^2 = \frac{\hat{\mathbf{f}}^*(\omega)\hat{\mathbf{g}}(\omega)}{|\hat{\mathbf{f}}(\omega)|^2} \quad (16)$$

This method effectively finds the generalized solution of the PSF estimate and is subject to noise amplification for cases where $|\hat{\mathbf{f}}(\omega)|^2$ is close to zero. To prevent this condition, we added a prior

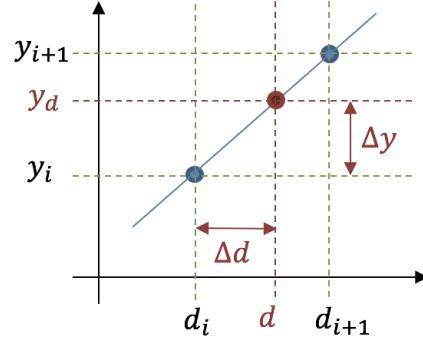


Figure 4: Linear Interpolation Illustration used for Weight Computation in Matrix \mathbf{B} . A pixel d distance from the image center will be the weighted sum of the two values in \mathbf{x} at distances d_i and d_{i+1} .

to the loss function enforcing a minimization of the energy in the PSF estimate. This prior follows Tikhonov regularization and allows for controlling this removing the noise amplification:

$$\hat{\mathbf{C}}_g(\omega) = \min_{\hat{\mathbf{C}}(\omega)} \frac{1}{2} \|\hat{\mathbf{g}}(\omega) - \hat{\mathbf{A}}(\omega)\hat{\mathbf{C}}(\omega)\|_2^2 + \mu \|\hat{\mathbf{C}}(\omega)\|_2^2 = \frac{\hat{\mathbf{f}}^*(\omega)}{|\hat{\mathbf{f}}(\omega)|^2 + \mu} \hat{\mathbf{g}}(\omega) \quad (17)$$

In both cases, we then project the estimate $\hat{\mathbf{C}}_g(\omega)$ onto the isotropic image space:

$$\hat{\mathbf{x}}_g(\omega) = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \hat{\mathbf{C}}_g(\omega) \quad (18)$$

This is the implementation under which we recovered the PSF estimate in the Fourier space, and our results are shown in the following section.

4 Experimental Results

4.1 Neural Network Results

The primary goal for the neural network was to find a robust implementation that recovers an in-focus image from a blurred input image taken at any arbitrary microscope defocus position. Our steps were threefold for achieving this task: 1) Validate that the network could recover an in-focus image from a single defocus plane, 2) Find an implementation scheme generalizing the network for arbitrary defocus correction, and 3) Validate network on images within the trained defocus regions, outside the defocus regions, and with arbitrary blurred images outside the dataset. These factors were chosen to evaluate the strengths and limitations of this neural network refocusing process. The code for this network was made using Keras and Tensorflow in the Python programming environment and training was performed on an NVIDIA Tesla P100 GPU using the Boston University Shared Computing Cluster. While varying based on dataset size and the network parameters, the training took approximately four to eight hours to complete.

Table 1: Training Dataset Image Distribution

Defocus Dataset/Network	$2\mu m$	$6\mu m$	$10\mu m$	$20\mu m$	$30\mu m$	Dataset Size
A	600	0	0	0	0	600
B	0	600	0	0	0	600
C	0	0	600	0	0	600
D	0	0	0	600	0	600
E	0	0	0	0	600	600
F	125	125	125	125	125	625

For the first validation step, multiple networks were trained using in-focus and defocused image pairs set at a single defocus plane. A total of 600 image pairs were used for each network training process at the defocus positions of $2\mu m$, $6\mu m$, $10\mu m$, $20\mu m$, and $30\mu m$. These network training schemes were labeled A through E and can be found in the Table 1. Each scheme was tested on both the experimental and synthetic dataset so that a separate synthetically-trained and experimentally-trained network existed for each training condition. After testing different training parameters, the optimal network parameters were found using 300 epochs or iterations through the training dataset, a batch size of 5 image pairs, with a Mean Squared Error (MSE) loss function and 10^{-4} learning rate. The batch size defines the number of images to use for each loss function calculation within a given epoch while the learning rate sets the rate at which the an optimal solution is found for the network's trained weights. These parameters and the MSE loss function were empirically determined based on where the network's outputs achieved the most visually similar images to the ground truth in-focus images. This process was aided by the fact that using larger batch sizes generated memory issues with the computer's GPU and smaller or larger learning rates quickly generated equivalently blurry output images to the original input defocused image or minimized the output image to contain the

same value at every pixel, respectively. Using these same training parameters, the network F in Table 1 was also trained using equal weighting of 125 image pairs from each defocus position. This network addressed the second step for generalizing the neural network for correcting any arbitrary defocus.

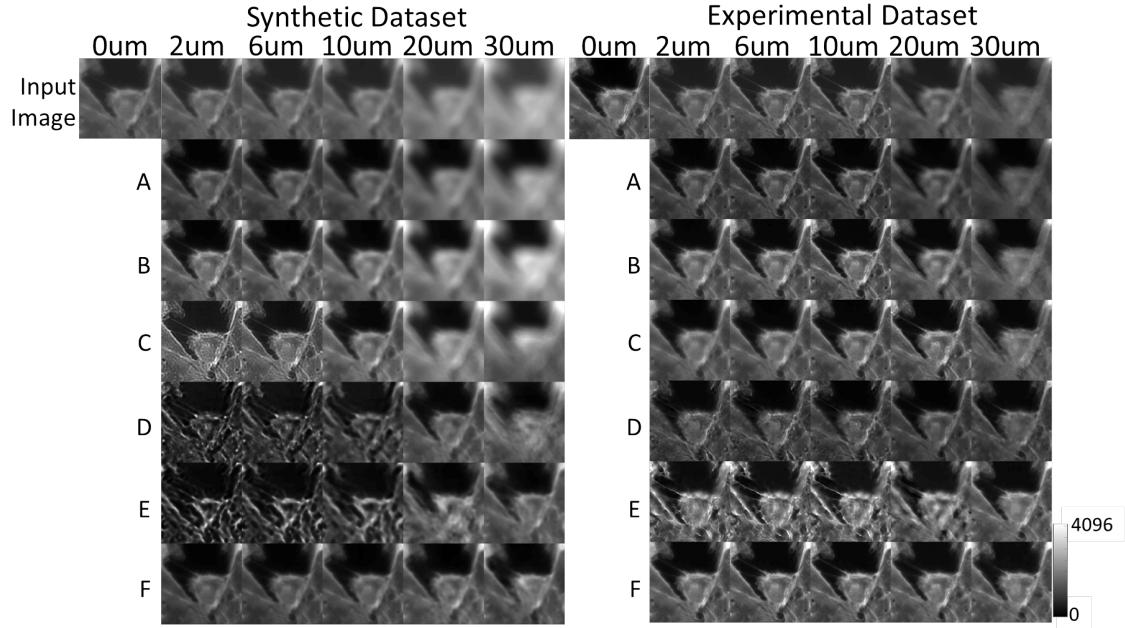


Figure 5: Example neural network output images from networks A-F training schemes trained on synthetic data (left) and experimental data (right). The top row shows the in-focus ground truth image at $0\mu m$ defocus and input defocused images. The subsequent image rows show the network outputs for each trained network. All images are shown in the same scaling.

The results from these networks A through F can be seen in Figure 5. Each network was tested on defocus images both within and outside the defocus positions where the network was trained. The single defocus plane networks show qualitatively good image refocusing when the input image has a small defocus up to $10\mu m$ in the experimental case and $6\mu m$ in the synthetic case. This result is particularly evident in the experimentally-trained case where networks A through C contain features at nearly the same resolution as the in-focus image. This trend breaks down for networks trained in both the synthetic and experimental cases with larger defocus positions at $20\mu m$ and $30\mu m$. These cases contain significant image blur networks A through C did not likely learn, which explains the blurred outputs from these networks at these defocus positions. Networks D and E only recover a reasonable in-focus image at their trained defocus position and introduce significant image artifacts at other defocus planes. This result suggests the networks have learned specific levels of image blur and are not generalizable to other blurring conditions. Due to the larger blur present in the synthetic dataset, this trend also occurs at the $10\mu m$ defocus plane for the synthetically-trained networks. These results suggest that while we satisfy our first validation step of recovering refocused images from a defocused input, these networks primarily work over their trained defocus regions and are not generalizable to arbitrary defocus positions.

The multi-focus network F shows much better results, however, for learning a general deblurring method in both the synthetic and experimental dataset cases. These networks recover high-resolution features such as the thin filopodia extensions from the cell and the ridges in the cellular membrane regardless of the original defocus position. These images suggest that, qualitatively, we can create a generalized neural network that will recover an in-focus image regardless of the input image's

defocus position.

We quantified the refocusing results from these networks by evaluating the average Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM) from 25 input test images on both the synthetic and experimentally-trained network images using ground-truth in-focus images as reference. The results from these metrics are shown in Figure 6 and were implemented using MATLAB 2017b with the following formulations:

$$PSNR(f_{est}(x, y), f(x, y)) = \log_{10} \frac{4096^2}{\|f_{est}(x, y) - f(x, y)\|^2}, \quad (19)$$

and

$$SSIM(f_{est}(x, y), f(x, y)) = \frac{(2\mu_{f_{est}}\mu_f + C_1)(2\sigma_{f_{est}, f} + C_2)}{(\mu_{f_{est}}^2 + \mu_f^2 + C_1)(\sigma_{f_{est}}^2 + \sigma_f^2 + C_2)}. \quad (20)$$

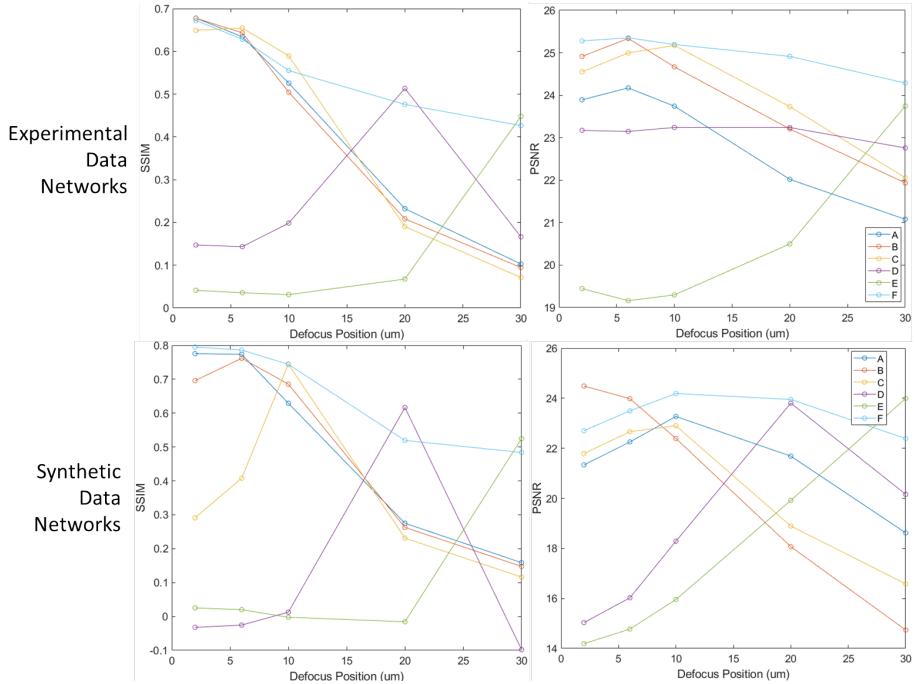


Figure 6: SSIM (Left) and PSNR (Right) Results Comparing the Network outputs and In-Focus Ground Truth Images. These values were averaged over 25 images from both the experimentally-trained (Top) and synthetically-trained (Bottom) neural networks.

Here, $f_{est}(x, y)$ is the estimated in-focus image from the network, $f(x, y)$ is the ground-truth in-focus image, μ and σ denotes mean and standard deviation respectively, and C_1 and C_2 define constants set for preventing divide by zero errors. The value 4096 was chosen for the PSNR calculation given the 12-bit dataset images. In the experimental dataset, we observe networks A through C have similar PSNR and SSIM values for the small defocus values and quickly degrade with increasing defocus, matching our qualitative expectations from Figure 5. Networks D and E also show only the largest SSIM values at their respective trained defocus plane as expected from qualitative evaluations of the image outputs. An interesting result is the PSNR for the network D trained on $20\mu m$ defocused data from the experimental dataset. The PSNR is almost flat at a value of 23 while the SSIM exhibits a peak at 0.53 for the $20\mu m$ defocused images. Since the PSNR depends primarily on the pixel-wise image differences and not on the image statistics like SSIM, this flat PSNR likely results from the overall decrease in pixel value observed in Figure 5 for network D on the experimentally-trained

networks. The pixel values vary across all the network outputs due to the stochastic nature of the network training process and is observed to be lower from network D compared to the other network outputs. This difference likely dominates any blurring effects in the image and does not effect the SSIM result as significantly due to that metric's reliance on image variance and statistics. In addition, this pixel variation was observed to occur when testing the same test image on multiple networks trained with the exact same dataset and training parameters. This variability indicates the networks are finding different minima between training sessions and that further work may be required for finding the optimal training parameters to achieve consistent network outputs.

From these image metrics, we also observe the network F trained on multiple defocus positions maintains large PSNR and SSIM values across all tested defocus positions. This agrees with our observations on Figure 5 and further confirms our second task of finding a generalizable neural network that refocuses images taken at arbitrary defocus positions. To further evaluate this condition and find where this network fails, we tested network F with additional cases listed below.

4.2 Network Failure Cases

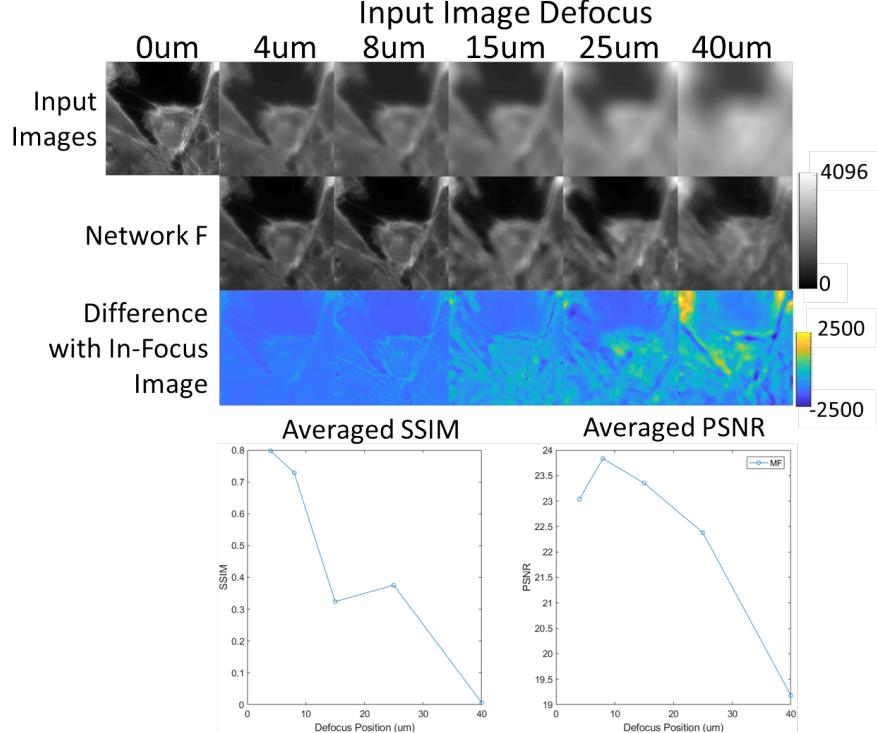


Figure 7: Example Network Outputs and Evaluation Metrics from the Synthetic Data-Trained Network F on Images Outside The Defocus Training Region.

For evaluating the multi-focus network F's limitations, we evaluated the cases where defocused images outside the trained defocus positions and blurred images outside the dataset were used as inputs for network F from the synthetic data trained case. These cases were tested for evaluating the robustness of this multi-focus network and determining whether the network is overfit to the human U2OS cells from the BBBC006v1 dataset. The results from these tests can be seen in Figure 7 and Figure 8, respectively. In Figure 7, the network's in-focus image recovery for $4\mu m$, $8\mu m$, $15\mu m$, $25\mu m$, and $40\mu m$ is shown along with images displaying pixel-wise differences between the network output and ground truth in-focus image and the average SSIM and PSNR over 25 test images.

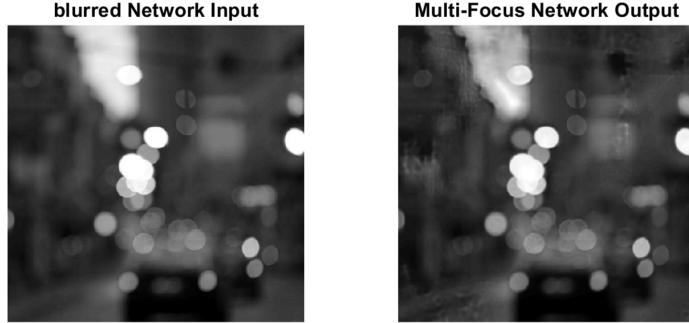


Figure 8: A Blurry Stock Photo Deblurred Using the Multi-Focus Synthetic Data-Trained Network F [19].

These results show worse performance than on the test images from the defocus positions where the network was trained, with substantial errors present in the difference images beyond $15\mu m$. This is evident in the differences images where large yellow regions indicating large pixel differences are clearly visible. The averaged SSIM and PSNR metrics also show a significant decrease in value beyond $8\mu m$, indicating the multi-focus network is not as robust for generalized image refocusing as previously mentioned. In Figure 8 where a random blurred image was submitted to the network, we observe negligible deblurring to the image but notice that additional features resembling the U2OS cell shape are present in the blurred streetlights of the image. These results indicate the network does learn some of the cell features from the dataset and that the cells are being overfit. Further work is therefore required to fully generalize this neural network for microscope deblurring applications.

4.3 PSF Recovery

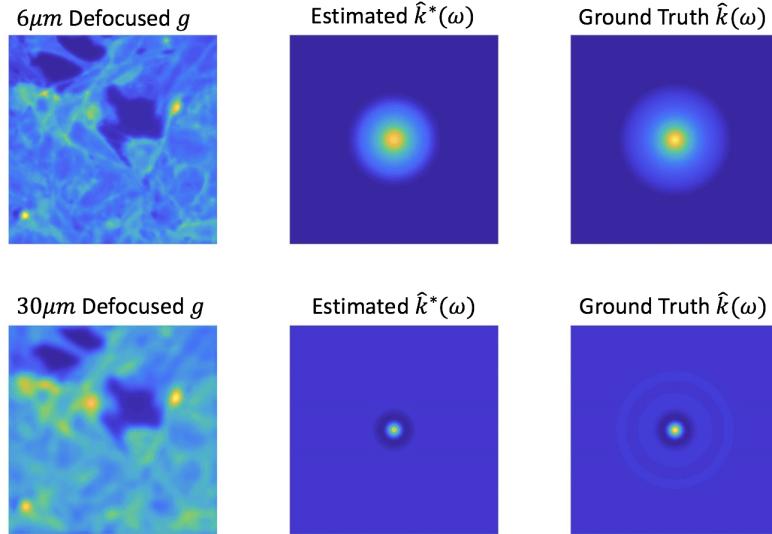


Figure 9: Synthetically Defocused Microscope Images (Left), the estimated PSF between these images and the in-focus images (Middle), compared with the ground truth PSF (Right).

Our PSF estimation algorithm was tested on synthetic data, where the PSF was known, and on real data, where the microscope properties were unknown. Results with synthetic data are shown in Figure 9. Here, two different defocus conditions were evaluated and their estimated PSFs were compared with the ground truth PSFs in the Fourier space. The first row of plots corresponds to $6\mu m$ defocus depth; the second row corresponds to $30\mu m$ defocus depth and is substantially more blurry. The blurred image \mathbf{g} is shown in the left column, with known PSF $\hat{\mathbf{k}}(\omega)$ shown on the right. Using the \mathbf{g} and sharpened neural network output \mathbf{f} , our algorithm estimates PSF $\hat{\mathbf{k}}_g(\omega)$ shown in the middle column. There is close agreement between true and estimated shape, indicating we can recover the PSF geometry from the in-focus and defocused image pair. We note that the PSF values are dissimilar, however, with the recovered PSF have a magnitude close to 1 at the image center while the original PSF is closer to 0.06 in the Fourier space. We were unable to reconcile this issue before the project's end, but we observe that this could be from the use of an in-focus and defocused image for PSF recovery rather than the true object and defocused image. This will recover the relative PSF difference between the in-focus and defocused image and will have similar shape to the original PSF but will not have the same magnitudes. This will be a factor to investigate going forward. Between the two defocus images shown in Figure 9, we observe the PSF's frequency spectrum bandwidth is larger in the $6\mu m$ condition compared to the $30\mu m$ case. This agrees with our expectations because the higher spatial frequencies are still relatively preserved in the $6\mu m$ image while they are blurred out in the $30\mu m$, so the corresponding blurring kernel should have greater energy in the higher frequencies.

Results for real microscope data, where the true PSF is not known, are shown in Figure 10. Although no ground truth PSF exists for the experimental dataset, we observe the recovered PSF frequency spectrum has a larger bandwidth than the synthetic dataset. This agrees with expectations because the experimental dataset contains higher resolution features than the synthetic dataset and thus will have a larger bandwidth. These results suggest the PSF is being recovered but additional tests, preferably from experimental data obtained with a known system design, are required for validating this claim.

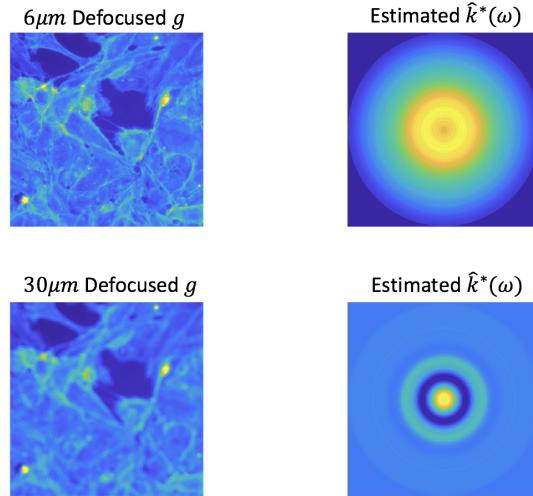


Figure 10: Real Defocused Microscope Images (Left) and their Corresponding PSF Estimates (Right).

5 Discussion and Conclusion

The results of this project suggest that neural networks can be used for refocusing blurred microscope images with the recovery of the imaging system PSF, but a number of steps are still required for developing this project into a usable software tool for biologists to deblur defocused microscope images. One of the most critical steps in this project is finding a robust neural network implementation with truly generalized image refocusing capabilities for an arbitrary defocus position. Our network training results indicate the network learned how to deblur specific defocus planes rather than learning a generic deblurring method for any defocus position. Combined with the image artifacts observed in Figure 8, these results suggest our dataset is too small and that better results could be obtained from training the network across more defocus image positions with additional cell types and different imaging systems. Recent work by Christiansen *et al.* suggests this is possible, as this group taught a neural network to add fluorescent labels to label-free images after training on a large dataset from multiple imaging systems on different cell types and different fluorescent labels [4]. In addition, the network architecture could be improved through the addition of more layers and residual blocks that have been shown to improve image deblurring capabilities [18]. These options would necessitate more training data as well due to the addition of more weights in the network. Prior to increasing network complexity or training size, however, simple network modifications could be utilized as well for improving the network's stability. The pixel value variability mentioned previously suggests the stochastic gradient descent-based method used for finding the optimal weights is not reaching the same minimum for each network training case. The use of additional epochs during training or an adaptive learning rate that decreases after a certain number of epochs could help the network find an optimal solution every time. Work by Zhao *et al.* also shows that SSIM could be used instead of MSE for the network loss function [21]. As shown with our results, SSIM provides better differentiation for in-focus and defocused images based on visual characteristics and thus could provide a better gradient calculation during the network training process for generating better refocused images.

On the PSF estimation, our project showed decent qualitative recovery of the blurring kernel's geometry but has poor quantitative PSF recovery from the substantial difference in the true and estimated PSF magnitudes. We acknowledge here that our method will not recover the true blurring kernel, but rather will recover the difference between the in-focus and defocused blurring kernel since we use an in-focus image for the ground truth. We can correct this going forward by training the neural network with synthetic data using the simulated objects and an array of blurred images from different defocus positions and simulated microscope designs. With a sufficiently large dataset, this could generalize the neural network for recovering the object rather than the in-focus image. This would enable the inverse problem to recover the true blurring kernel of the imaging system instead of the estimate recovered currently. Furthermore, the PSF estimation could be improved by applying the minimization simultaneously on multiple images from the same defocus position. Since each image should have the same blurring kernel for a given axial position, using multiple images could reduce noise present in the PSF recovery and provide a better estimate. This minimization process could also be improved with additional priors in the minimization process. The most immediate prior useful for the PSF estimation would be the use of an indicator sequence restricting the PSF's frequency spectrum onto a circular region. This would utilize the compact support of the PSF's bandwidth for minimizing the PSF estimate to a more optimal solution.

Overall, the results from our project act as a proof-of-concept test showing that neural networks can correct defocused microscope images and that an unknown imaging system's PSF can be recovered from the refocused and blurred image pair. Going forward, we hope to continue exploring the aforementioned improvements we can apply to our current work and eliminate the PSF magnitude error, overfitting artifacts, and other issues in our network and PSF recovery algorithms. Based on the recent publications on neural networks for defocus correction found in the literature, we believe this research is relevant to the optical imaging field and could be useful for biological research in the future.

References

- [1] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [2] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [3] J. R. Chang, C.-L. Li, B. Poczos, B. V. Kumar, and A. C. Sankaranarayanan, “One network to solve them all: solving linear inverse problems using deep projection models,” *arXiv preprint arXiv:1706.05160*, 2017.
- [4] E. M. Christiansen, S. J. Yang, D. M. Ando, A. Javaherian, G. Skibinski, S. Lipnick, E. Mount, A. O’Neil, K. Shah, A. K. Lee, *et al.*, “In silico labeling: Predicting fluorescent labels in unlabeled images,” *Cell*, 2018.
- [5] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, “Denoising prior driven deep neural network for image restoration,” *arXiv preprint arXiv:1801.06756*, 2018.
- [6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2758–2766, 2015.
- [7] J. W. Goodman, *Introduction to Fourier Optics*. McGraw-Hill, 2 ed., 1996.
- [8] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, “CNN-based projected gradient descent for consistent image reconstruction,” *arXiv preprint arXiv:1709.01809*, 2017.
- [9] S. Jiang, J. Liao, Z. Bian, K. Guo, Y. Zhang, and G. Zheng, “Transform-and multi-domain deep learning for single-frame rapid autofocusing in whole slide imaging,” *Biomedical optics express*, vol. 9, no. 4, pp. 1601–1612, 2018.
- [10] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.
- [11] Z. Jingshan, R. A. Claus, J. Dauwels, L. Tian, and L. Waller, “Transport of intensity phase imaging by intensity spectrum fitting of exponentially spaced defocus planes,” *Opt. Express*, vol. 22, pp. 10661–10674, May 2014.
- [12] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, “Annotated high-throughput microscopy image sets for validation,” *Nat Methods*, vol. 9, no. 7, p. 637, 2012.
- [13] M. T. McCann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: A review,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017.
- [14] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [15] P. Sarder and A. Nehorai, “Deconvolution methods for 3-d fluorescence microscopy images,” *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 32–45, 2006.
- [16] A. S. Stender, K. Marchuk, C. Liu, S. Sander, M. W. Meyer, E. A. Smith, B. Neupane, G. Wang, J. Li, J.-X. Cheng, B. Huang, and N. Fang, “Single cell optical imaging and spectroscopy,” *Chemical Reviews*, vol. 113, no. 4, pp. 2469–2527, 2013. PMID: 23410134.

- [17] J. Sun, H. Li, Z. Xu, *et al.*, “Deep ADMM-Net for compressive sensing MRI,” in *Advances in Neural Information Processing Systems*, pp. 10–18, 2016.
- [18] X. Tao, H. Gao, Y. Wang, X. Shen, J. Wang, and J. Jia, “Scale-recurrent network for deep image deblurring,” *ArxiV preprint arXiv:1802.01770*, 2018.
- [19] Viktor Bezrukov, “Blurred traffic jam,” 2012. [Online; accessed May 2nd, 2018].
- [20] S. J. Yang, M. Berndl, D. M. Ando, M. Barch, A. Narayanaswamy, E. Christiansen, S. Hoyer, C. Roat, J. Hung, C. T. Rueden, *et al.*, “Assessing microscope image focus quality with deep learning,” *BMC bioinformatics*, vol. 19, no. 1, p. 77, 2018.
- [21] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.