3D Image Restoration using Machine Learning

Master Thesis Defense

Sönke Ziemer

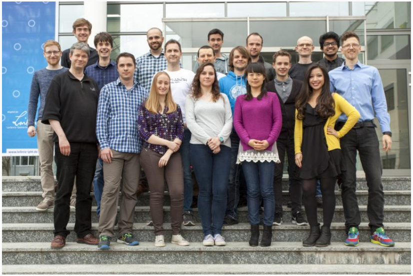
5.11.2018

Thank You

Supervisors: Rainer Heintzmann, Christian Karras, Benedict Diederich

Proofreaders: Dina Ratsimandresy, Elsie Quansah

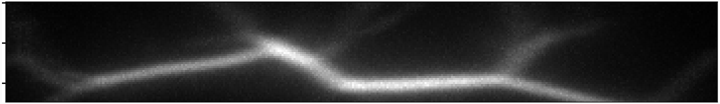
The Bio-Nanoimaging Group



3D Microscopic Imaging: $\mathrm{obj} \mapsto \mathrm{img}$

Investigated ways to recover obj

img



$\downarrow$?

obj

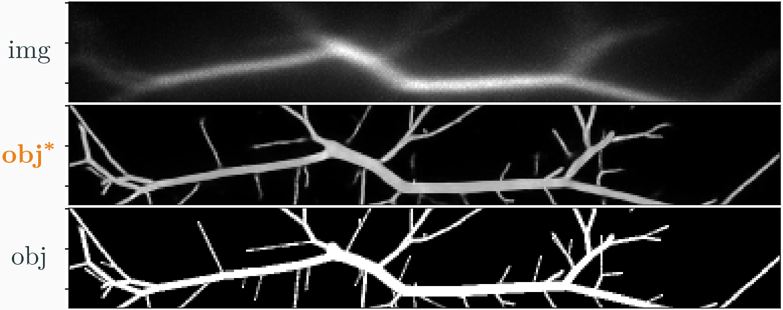
Shown are Maximum Projections

Introduction

3D Microscopic Imaging: $\mathrm{obj} \mapsto \mathrm{img}$

Investigated ways to recover obj

Using a Convolutional Neural Network:

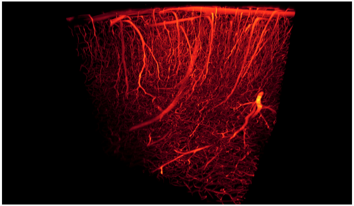


Contents

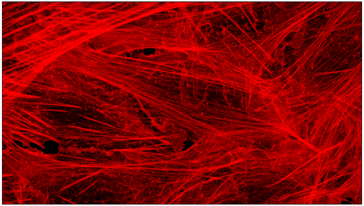
Dataset Generation: `VascuSynth`

Software to simulate blood vessels

Assume similarity to actin or microtubules



Blood Vessels, By Antonino Paolo Di Giovanna [CC BY-SA 4.0], from Wikimedia Commons



Actin, adapted from Howard Vindin [CC BY-SA 4.0], from Wikimedia Commons

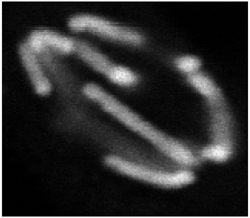
Related: [1, 2]

Dataset Generation: Imaging

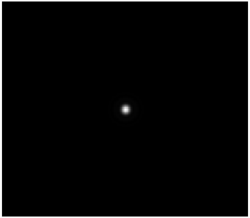
$$\mathrm{img} = \mathsf{N}_{\mathsf{Poisson}}(\mathrm{obj} \otimes \mathrm{PSF} + \mathrm{b})$$

Dataset Generation: Imaging

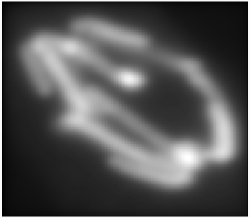
$$\mathrm{img} = \mathsf{N}_{\mathsf{Poisson}}(\mathbf{obj} \otimes \mathbf{PSF} + \mathrm{b})$$



$\otimes_{\mathsf{3D}}$

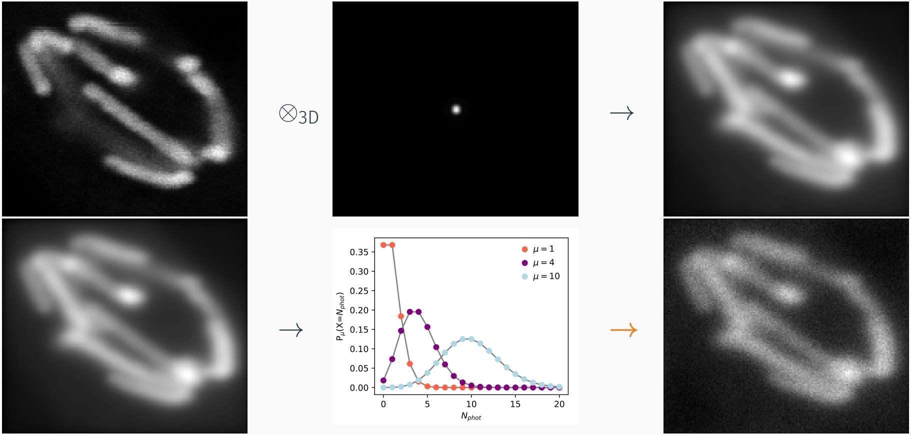


$\rightarrow$



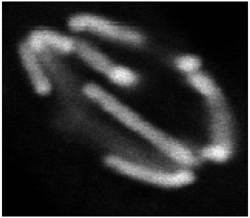
Dataset Generation: Imaging

$$\mathrm{img} = \mathrm{N_{Poisson}}(\mathrm{obj} \otimes \mathrm{PSF} + \mathrm{b})$$

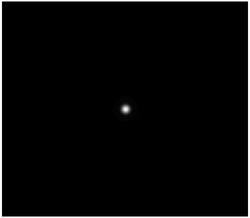


Dataset Generation: Scaling blur and noise

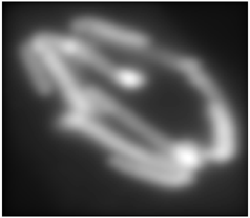
Blur: Vary wavelength $\lambda = 520\,\text{nm}$, $N_{\text{phot}} = 1000$

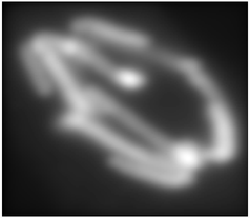


$\otimes_{3D}$

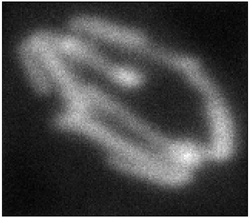


$\rightarrow$



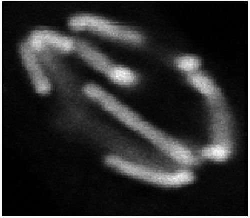


$\rightarrow$

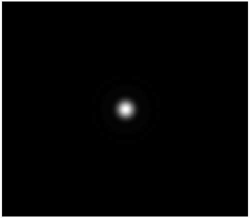


Dataset Generation: Scaling blur and noise

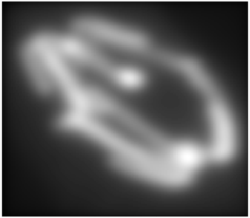
Blur: Vary wavelength $\lambda = 1040\,\mathrm{nm}$, $N_{\mathrm{phot}} = 1000$

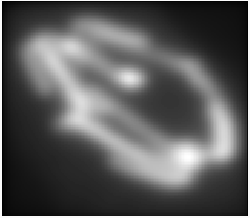


$\otimes_{3D}$

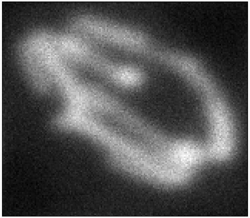


$\rightarrow$



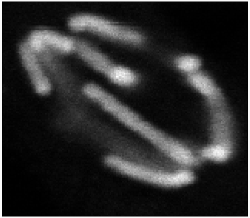


$\rightarrow$

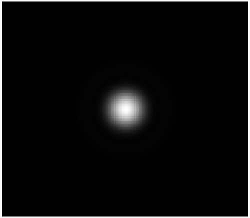


Dataset Generation: Scaling blur and noise

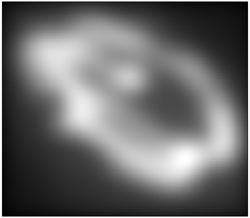
Blur: Vary wavelength $\lambda = 2080\,\mathrm{nm}$, $\quad N_{\mathrm{phot}} = 1000$

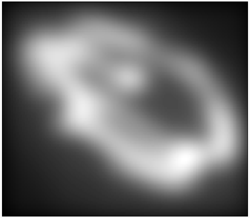


$\otimes_{3D}$

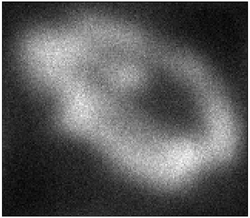


$\rightarrow$



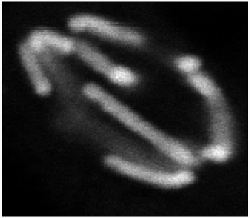


$\rightarrow$

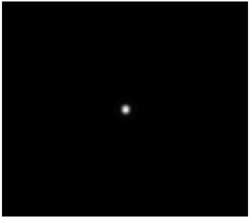


Dataset Generation: Scaling blur and noise

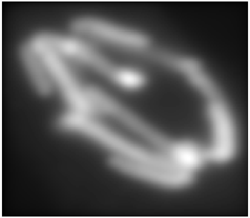
Blur: Vary number of photons $\lambda = 520\,\text{nm}$, $N_{\text{phot}} = 10\,000$

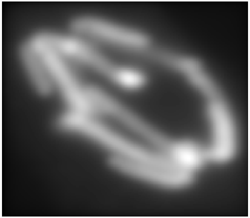


$\otimes_{3D}$

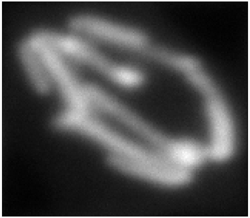


$\rightarrow$



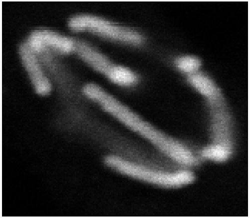


$\rightarrow$

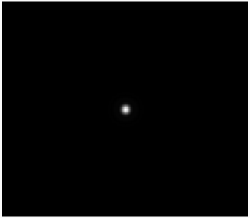


Dataset Generation: Scaling blur and noise

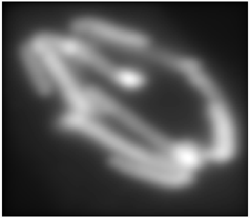
Blur: Vary number of photons $\lambda = 520\,\text{nm}$, $N_{\text{phot}} = 1000$

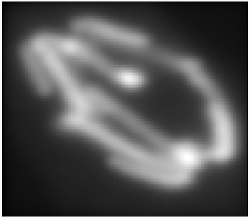


$\otimes_{3D}$

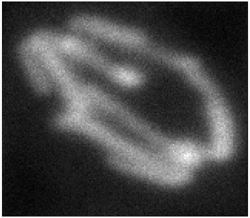


$\rightarrow$



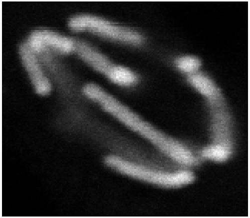


$\rightarrow$

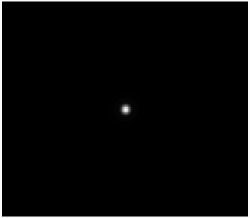


Dataset Generation: Scaling blur and noise

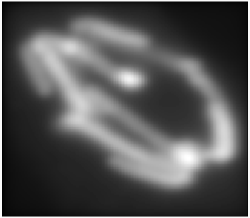
Blur: Vary number of photons $\lambda = 520\,\mathrm{nm}$, $N_{\mathrm{phot}} = 100$

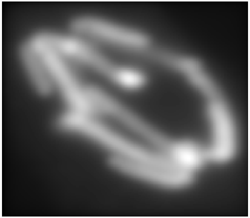


$\otimes_{3D}$

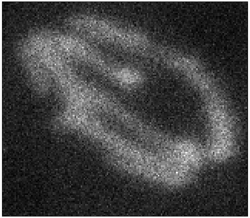


$\rightarrow$





$\rightarrow$



Deconvolution: Approaches

Direct inversion / Wiener Deconvolution:

$\rightarrow$ one step, but limited results

Cost Function Approach:

$\rightarrow$ minimize cost

$$\mathrm{obj}^* = \operatorname*{argmin}_{\mathrm{est}} \mathrm{cost}_{\mathrm{img}}(\mathrm{est})$$

$\rightarrow$ gradient descent:

$$\mathrm{est}_i^{(n+1)} = \mathrm{est}_i^{(n)} + \alpha^{(n)} \frac{\mathrm{dcost}_{\mathrm{img}}}{\mathrm{dest}_i}(\mathrm{est}_i^{(n)})$$

Related: [3]

$$\mathrm{cost}_{\mathrm{img}}(\mathrm{est}) = \mathsf{fid}_{\mathrm{img}}(\mathrm{est}) + \alpha \cdot \mathsf{Reg}(\mathrm{est})$$

Cost Function Approach: Data Fidelity

$$\mathrm{cost}_{\mathrm{img}}(\mathrm{est}) = \mathsf{fid}_{\mathrm{img}}(\mathbf{est}) + \alpha \cdot \mathsf{Reg}(\mathrm{est})$$

Reason for data fidelity:
→ estimate should be related to image

Examples:

$$\mathsf{MSE}_{\mathrm{img}}[\mathsf{blur}(\mathrm{est})] = \frac{1}{N} \sum_{i=0}^{N-1} [\mathsf{blur}(\mathrm{est}_i) - \mathrm{img}_i]^2$$

$$\mathsf{Poi}_{\mathrm{img}}[\mathsf{blur}(\mathrm{est})] = \sum_{i=0}^{N-1} [\mathsf{blur}(\mathrm{est}_i) + \ln \mathrm{img}_i! - \mathrm{img}_i \cdot \ln \mathsf{blur}(\mathrm{est}_i)]$$

Cost Function Approach: Regularization

$$\mathrm{cost}_{\mathrm{img}}(\mathrm{est}) = \mathsf{fid}_{\mathrm{img}}(\mathrm{est}) + \alpha \cdot \mathrm{Reg}(\mathbf{est})$$

Reasons for Regularization:
$\rightarrow$ Statistical Estimate from a single measurement
$\rightarrow$ Break ambiguity of estimate

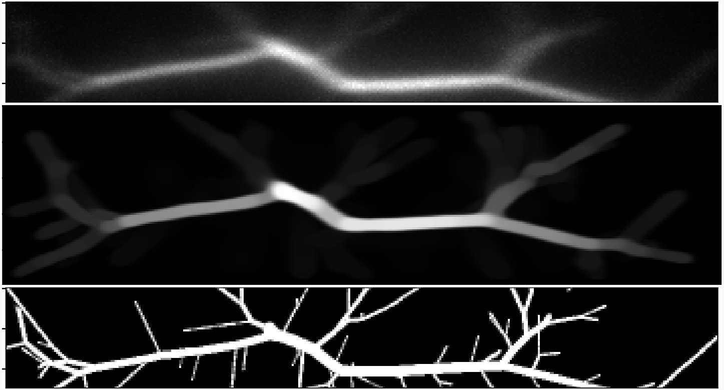
Example:

$$\mathsf{TV}(\mathrm{est}) = \sum_{i=0}^{N-1} \sqrt{(\partial_{\mathsf{x}}\mathrm{est}_i)^2 + (\partial_{\mathsf{y}}\mathrm{est}_i)^2 + (\partial_{\mathsf{z}}\mathrm{est}_i)^2}$$

Related: [4]

Result: Cost Function Approach

$$\mathrm{cost}_{\mathrm{img}}(\mathrm{est}) = \sum_{i=0}^{N-1} \left[\mathsf{blur}(\mathrm{est}_i) + \ln \mathrm{img}_i! - \mathrm{img}_i \cdot \ln \mathsf{blur}(\mathrm{est}_i)\right] + \mathsf{TV}(\mathrm{est})$$



img

obj*

obj

Deconvolution using a Convolutional Neural Network

Before: Cost was function of est with respect to **one** img

Now: Define a cost as a function of CNN-parameters $\vec{\theta}$ with respect to **all** images in a dataset $\mathcal{D}$

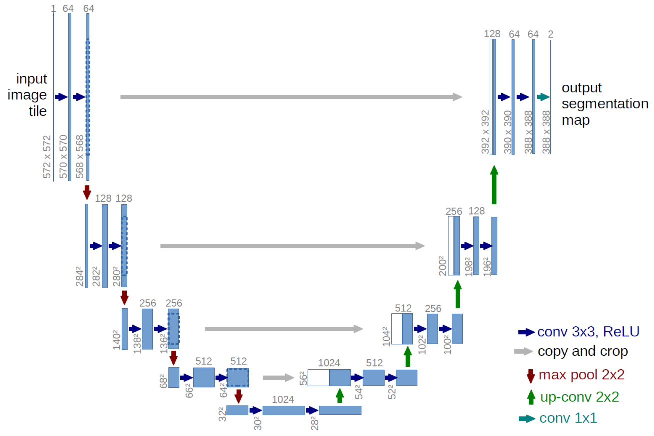
$\rightarrow$ 2 Steps:

1. Train CNN on dataset: $\vec{\theta}^* = \mathsf{argmin}_{\vec{\theta}}\, \mathrm{cost}_{\mathcal{D}}(\vec{\theta})$
2. Test CNN on unknown images: $\mathrm{obj}^* = \mathsf{CNN}_{\vec{\theta}^*}(\mathrm{img})$

Related: [3]

Deconvolution using a CNN

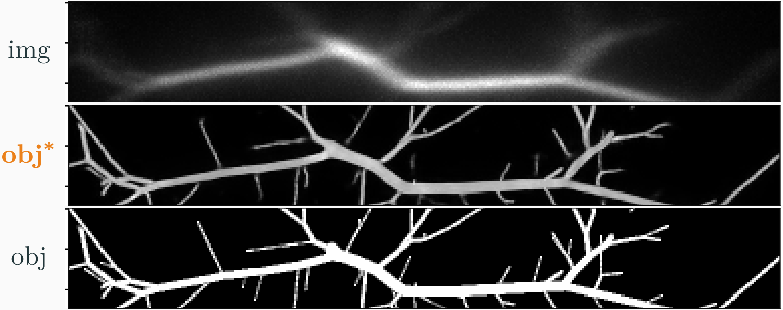
network architecture: U-Net



Related: [5, 6]

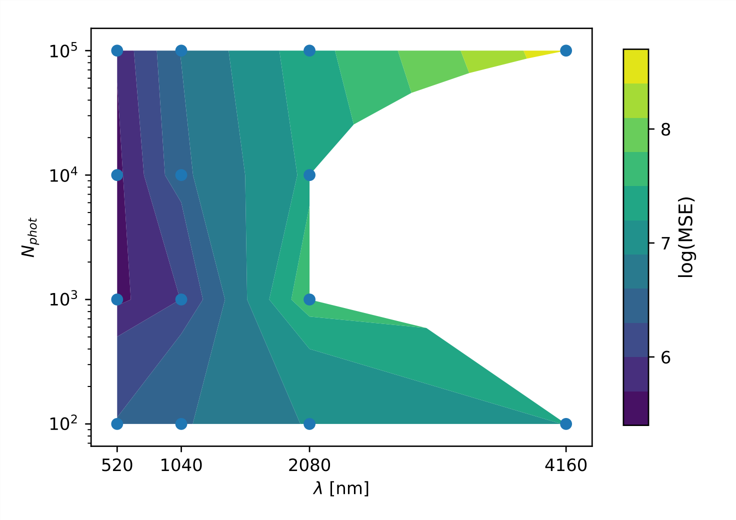
Results: CNN - Proof of Principle

$$\mathrm{cost}_{\mathcal{D}}(\vec{\theta}) = \frac{1}{N_{\mathcal{D}}} \sum_{(\mathrm{obj},\mathrm{img})\in\mathcal{D}} \mathsf{MSE}_{\mathrm{obj},\mathrm{img}}(\mathsf{CNN}_{\vec{\theta}}(\mathrm{img}_i), \mathrm{obj}_i)$$



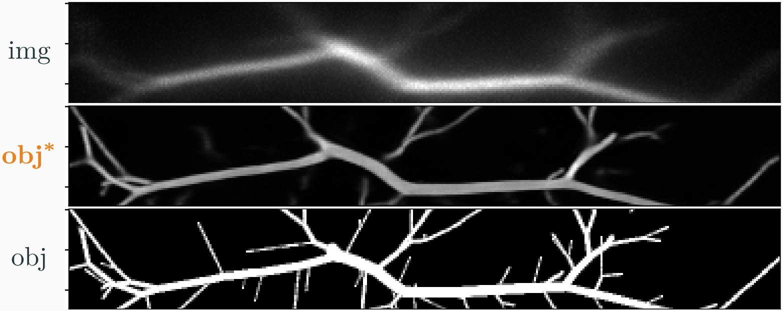
$\lambda = 520\,\mathrm{nm}, N_{\mathrm{phot}} = 10\,000$

Results: CNN - Application Limits



Results: CNN - Application Limits

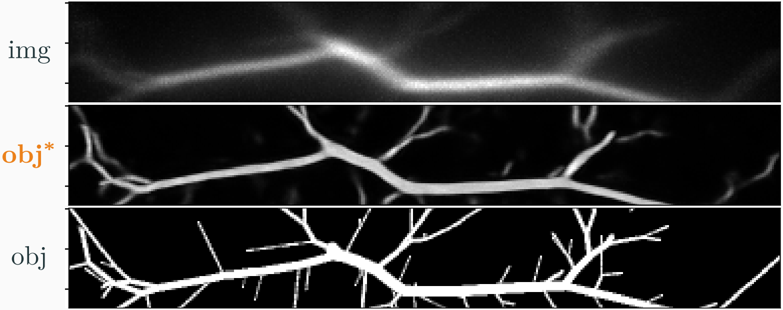
$$\mathrm{cost}_{\mathcal{D}}(\vec{\theta}) = \frac{1}{N_{\mathcal{D}}} \sum_{(\mathrm{obj},\mathrm{img})\in\mathcal{D}} \mathsf{MSE}_{\mathrm{obj},\mathrm{img}}(\mathsf{CNN}_{\vec{\theta}}(\mathrm{img}_i), \mathrm{obj}_i)$$



$\lambda = 1040\,\mathrm{nm}, N_{\mathrm{phot}} = 10\,000$

Results: CNN - Application Limits

$$\mathrm{cost}_{\mathcal{D}}(\vec{\theta}) = \frac{1}{N_{\mathcal{D}}} \sum_{(\mathrm{obj},\mathrm{img})\in\mathcal{D}} \mathsf{MSE}_{\mathrm{obj},\mathrm{img}}(\mathsf{CNN}_{\vec{\theta}}(\mathrm{img}_i), \mathrm{obj}_i)$$

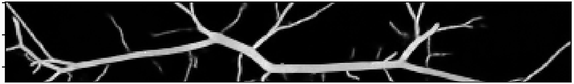
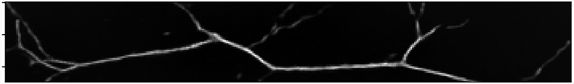


$\lambda = 520\,\mathrm{nm}, N_{\mathrm{phot}} = 100$

Results: CNN - Transferability

Transfer to a different blur level did not work well.

Transfer to a different noise level:

training	testing	Result
1000	1000	
100 000	1000	
100	1000	

$\lambda = 520\,\mathrm{nm}$, N_{phot}varies

Conclusion and Outlook

Proof of Principle ✓

Applicability of models (✓)

Transfer of models $\times$

Conclusion and Outlook

Proof of Principle ✓

$\rightarrow$ Poisson cost for CNN
$\rightarrow$ Unsupervised Training
$\rightarrow$ Architecture and Hyperparameter

Applicability of models (✓)

Transfer of models $\times$

Conclusion and Outlook

Proof of Principle ✓	→ Poisson cost for CNN → Unsupervised Training → Architecture and Hyperparameter
Applicability of models (✓)	→ Collect more data points → Quantify image quality
Transfer of models ×	

Conclusion and Outlook

Proof of Principle ✓	→ Poisson cost for CNN → Unsupervised Training → Architecture and Hyperparameter
Applicability of models (✓)	→ Collect more data points → Quantify image quality
Transfer of models ×	→ Identify reasons → Countermeasures

Thank you for your attention

References

[1] G. Hamarneh and P. Jassi. "VascuSynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis." In: **Computerized Medical Imaging and Graphics** 34.8 (Dec. 2010), p. 605. DOI: 10.1016/j.compmedimag.2010.06.002.

[2] P. Jassi and G. Hamarneh. "VascuSynth: Vascular Tree Synthesis Software." In: **Insight Journal** January-June (Apr. 2011), p. 1. URL: http://hdl.handle.net/10380/3260.

[3] M. T. McCann, K. H. Jin, and M. Unser. "Convolutional Neural Networks for Inverse Problems in Imaging: A Review." In: **IEEE Signal Processing Magazine** 34.6 (Nov. 2017), p. 85. DOI: 10.1109/msp.2017.2739299.

[4] U. S. Kamilov et al. "Optical Tomographic Image Reconstruction Based on Beam Propagation and Sparse Regularization." In: **IEEE Transactions on Computational Imaging** 2.1 (Mar. 2016), p. 59. DOI: 10.1109/tci.2016.2519261.

[5] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: **Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015**. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Cham: Springer International Publishing, May 18, 2015, p. 234. DOI: 10.1007/978-3-319-24574-4_28.

[6] M. Weigert et al. "Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy." In: **bioRxiv (pre-print)** (Dec. 2017). DOI: 10.1101/236463.

Appendix