

Deep Image Prior

for noise reduction in optical coherence tomography images

Kristen Hagan, David Li, and Jessica Loo

Introduction

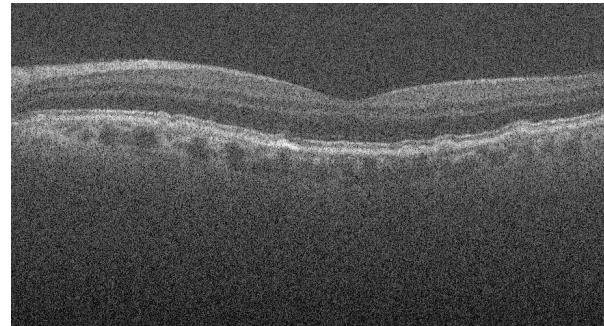
- Optical coherence tomography (OCT): 3D optical imaging technique using interferometry to acquire reflectance profiles
- Variety of clinical uses - but most dominant in ophthalmology for screening and diagnosis
- OCT imaging is corrupted by shot noise and speckle artifacts



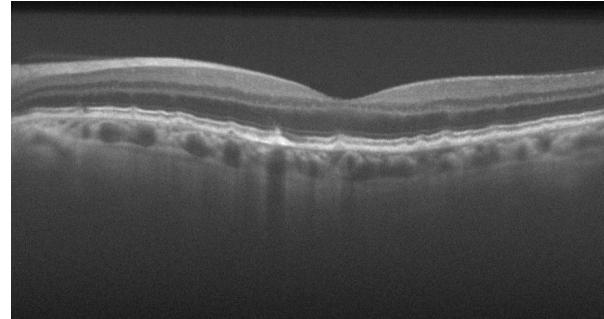
Leica Microsystems

Data Set

- 22 sets of OCT scans (raw and averaged) [1]
- 5 scans selected for optimization
- Cropped to uniform size and normalized
- Averaged scans are used as a “ground truth”

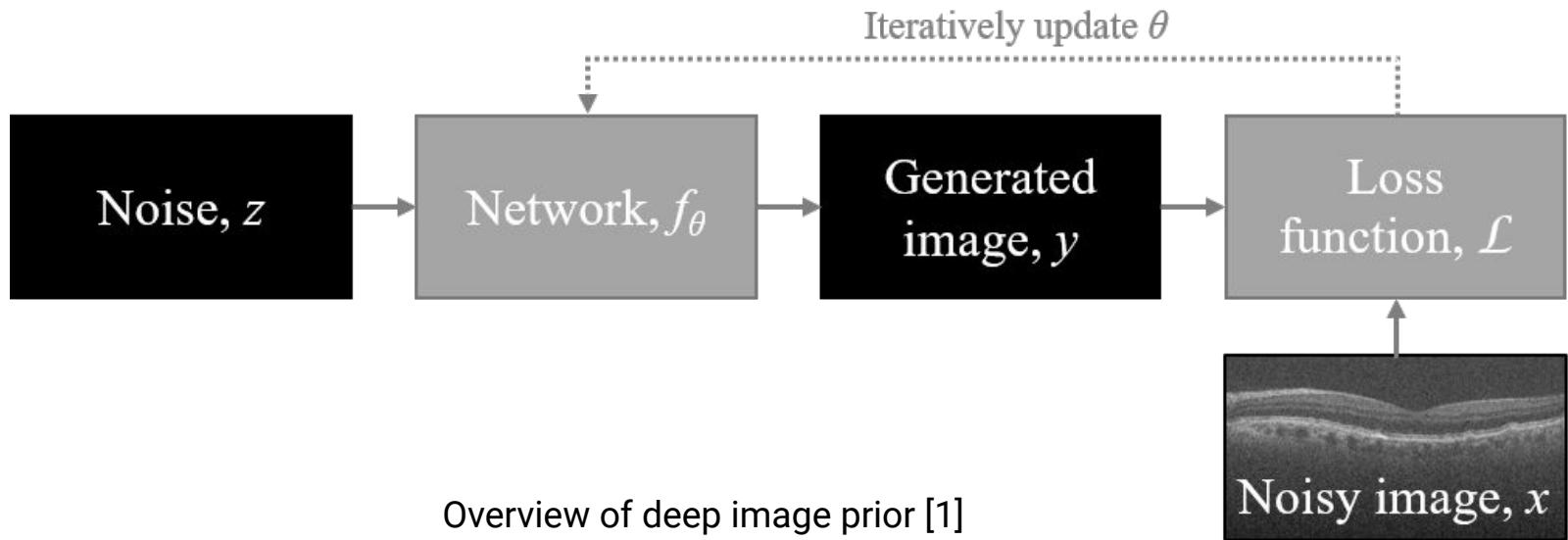


Raw



Averaged

Method

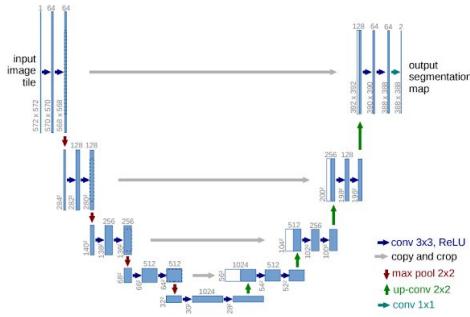


[1] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.

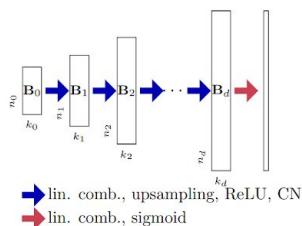
Method

Networks:

- U-Net [1]



- Deep Decoder [2]



Loss functions:

- $\mathcal{L}_{L_2} = |y - x|^2$
- $\mathcal{L}_{L_1} = |y - x|$
- $\mathcal{L}_{L_2-L_1} = w_0 \mathcal{L}_{L_1} + w_1 \mathcal{L}_{L_2}$
- $\mathcal{L}_{L_2-tv} = w_1 \mathcal{L}_{L_2} + w_2 tv(y)$
- $\mathcal{L}_{L_2-edge} = w_1 \mathcal{L}_{L_2} + w_2 edge_h(y) + w_3 edge_v(y)$

[1] 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*, 2015, pp. 234–241.

[2] R. Heckel, and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," *arXiv preprint arXiv:1810.03982*, 2018.

Results

$$CNR = \frac{|\mu_f - \mu_b|}{0.5\sqrt{\sigma_f^2 + \sigma_b^2}}$$

$$SSIM(g, y) = \frac{(2\mu_g\mu_y + C_1)(2\sigma_{gy} + C_2)}{(\mu_g^+\mu_y^2 + C_1)(\sigma_g^+\sigma_y^2 + C_2)}$$

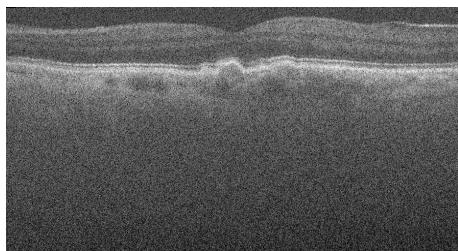
TABLE I
AVERAGE QUANTITATIVE METRICS

Network	Loss	CNR	SSIM
None	None	1.780	0.079
U-Net	\mathcal{L}_{L_2}	7.347	0.519
U-Net	\mathcal{L}_{L_1}	5.985	0.435
U-Net	$\mathcal{L}_{L_2-L_1}$	6.426	0.507
U-Net	\mathcal{L}_{L_2-tv}	8.295	0.526
U-Net	\mathcal{L}_{L_2-edge}	10.254	0.510
Deep Decoder	\mathcal{L}_{L_2-tv}	4.783	0.393
Deep Decoder	\mathcal{L}_{L_2-edge}	7.807	0.492

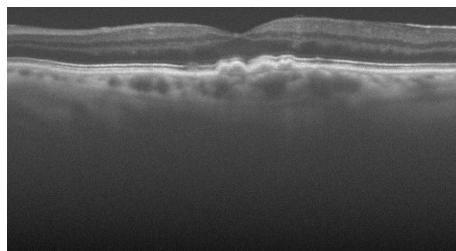
Network = None and **Loss** = None indicate the metrics of the original noisy image. The best metrics are shown in **bold**.

Results – 1

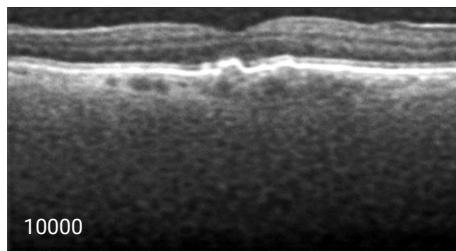
Noisy image:



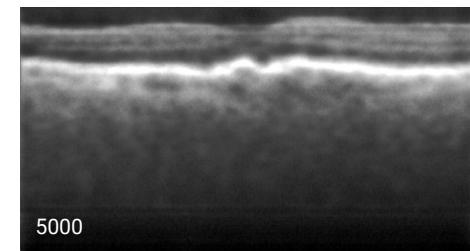
Averaged image:



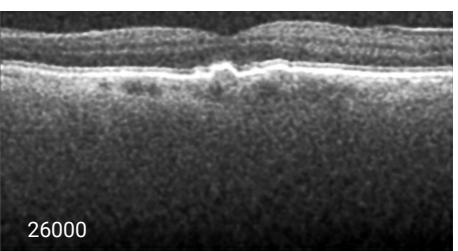
U-Net (L2):



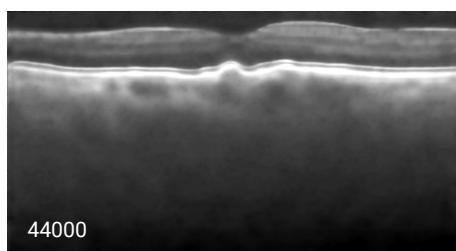
U-Net (L1):



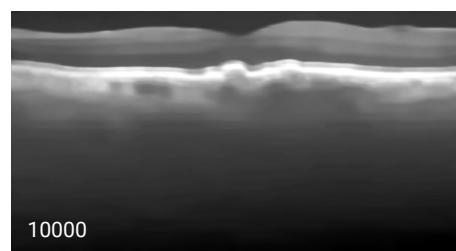
U-Net (L2_L1):



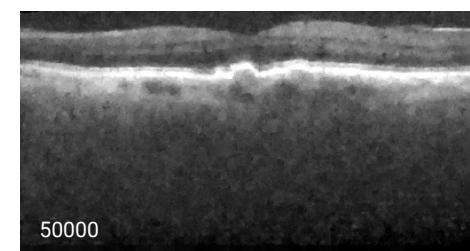
U-Net (L2_tv):



U-Net (L2_edge):

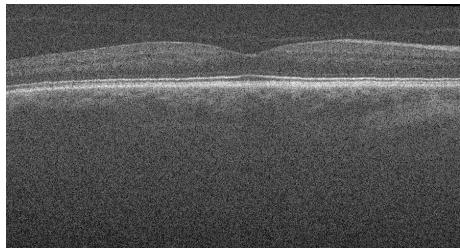


Deep Decoder (L2_edge):

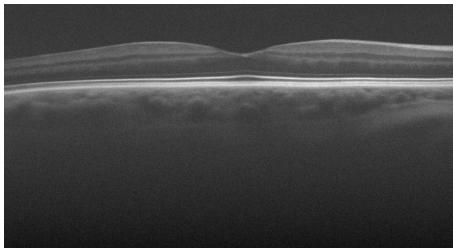


Results - 2

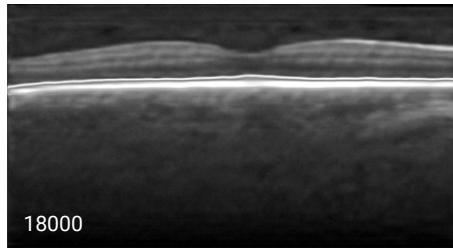
Noisy image:



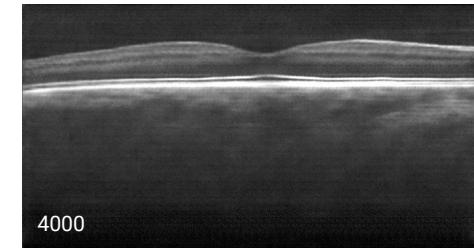
Averaged image:



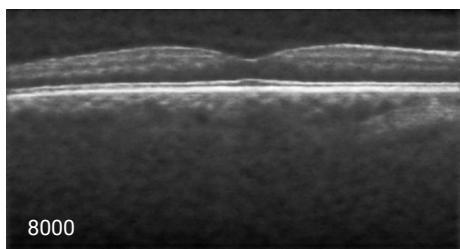
U-Net (L2):



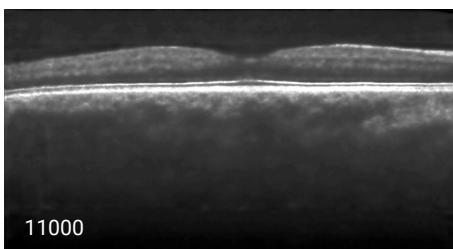
U-Net (L1):



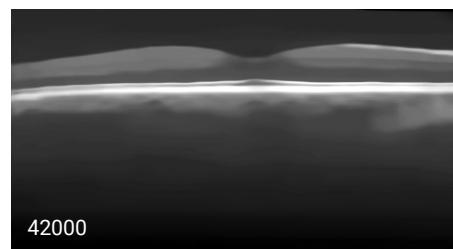
U-Net (L2_L1):



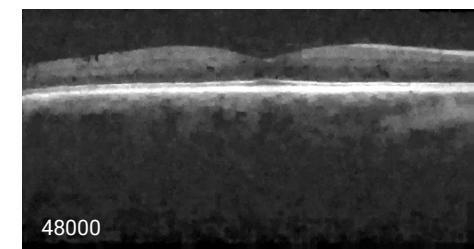
U-Net (L2_tv):



U-Net (L2_edge):

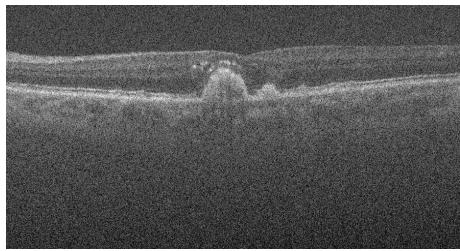


Deep Decoder (L2_edge):

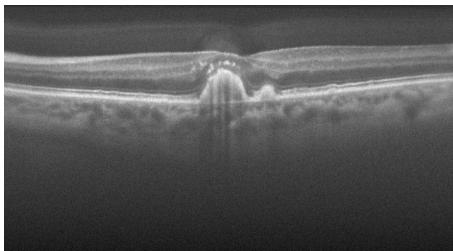


Results – 3

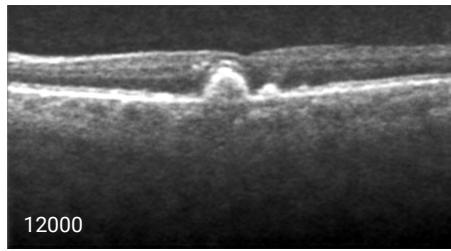
Noisy image:



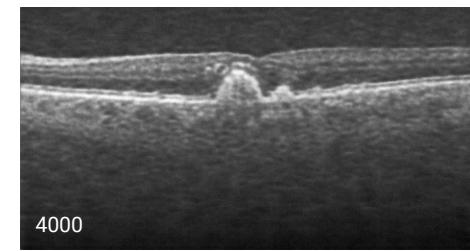
Averaged image:



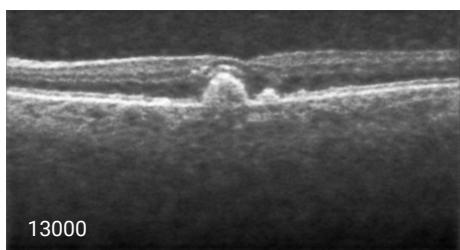
U-Net (L2):



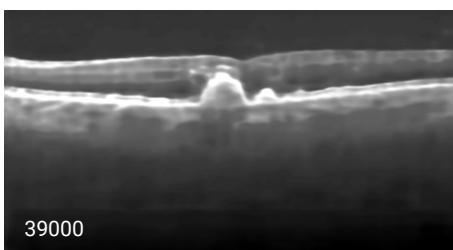
U-Net (L1):



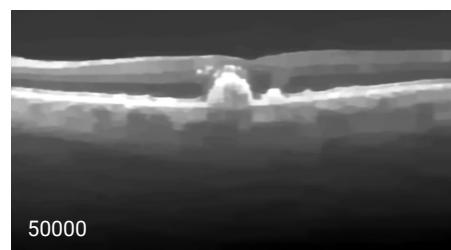
U-Net (L2_L1):



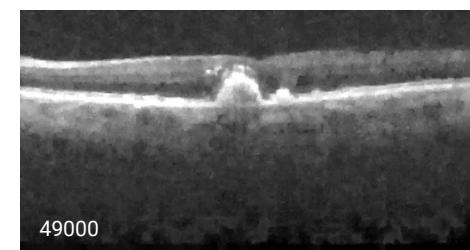
U-Net (L2_tv):



U-Net (L2_edge):

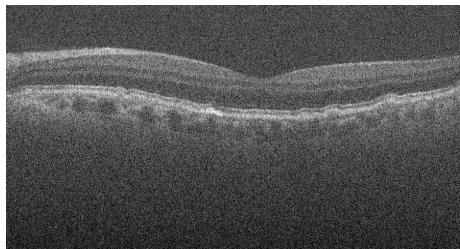


Deep Decoder (L2_edge):

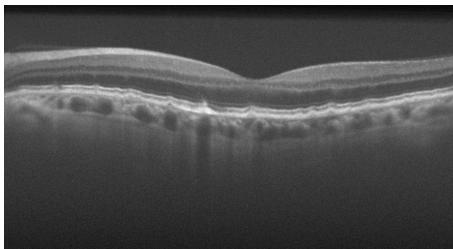


Results - 4

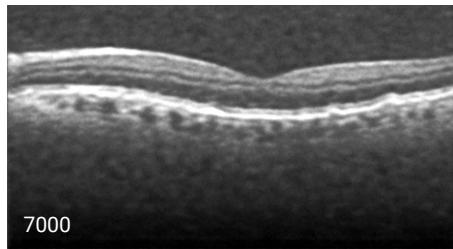
Noisy image:



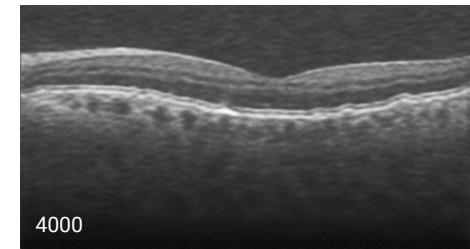
Averaged image:



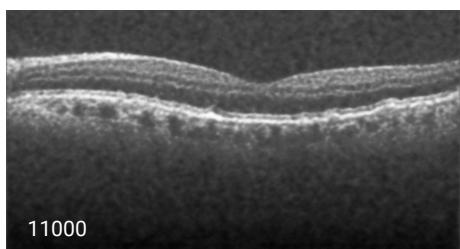
U-Net (L2):



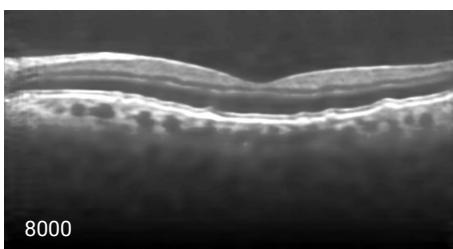
U-Net (L1):



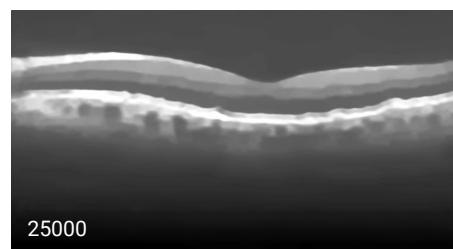
U-Net (L2_L1):



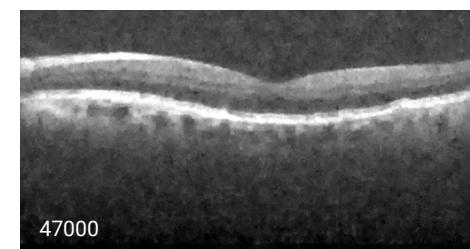
U-Net (L2_tv):



U-Net (L2_edge):



Deep Decoder (L2_edge):

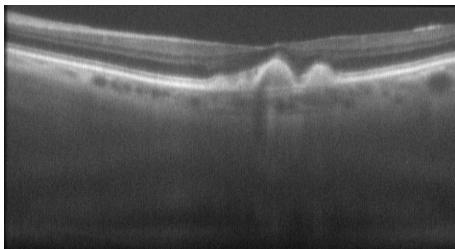


Results - 5

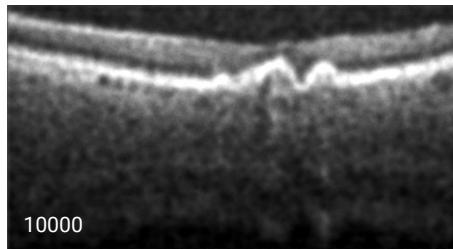
Noisy image:



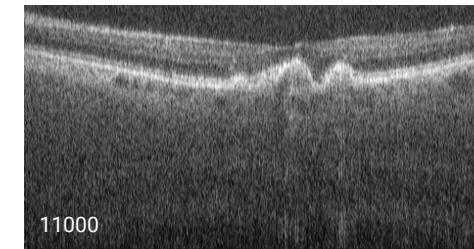
Averaged image:



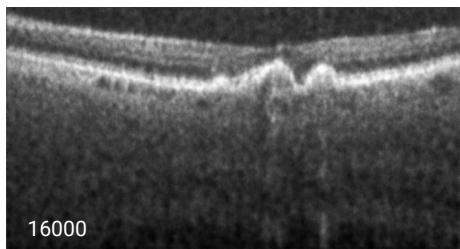
U-Net (L2):



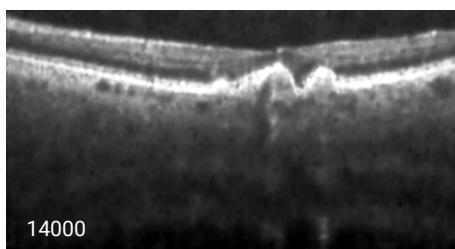
U-Net (L1):



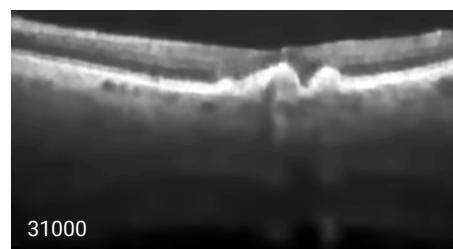
U-Net (L2_L1):



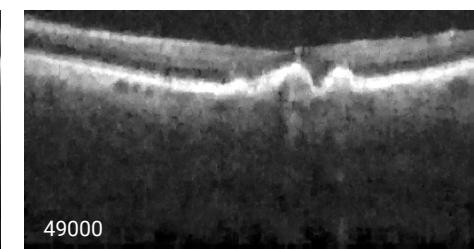
U-Net (L2_tv):



U-Net (L2_edge):



Deep Decoder (L2_edge):



16000

14000

31000

49000

Conclusion

- Combination of U-Net with $\mathcal{L}_{L_2_tv}$ and $\mathcal{L}_{L_2_edge}$ provided the best noise reduction
- Accuracy of downstream tasks (e.g. segmentation) as alternative evaluation method
- OCT-specific loss functions (e.g. noise model, retinal structures)