DeepLSR: a deep learning approach for laser speckle reduction

Taylor L. Bobrow¹, Faisal Mahmood¹, Miguel Inserni¹, & Nicholas J. Durr¹,²

¹Department of Biomedical Engineering, Johns Hopkins University (JHU), Baltimore, MD 21218
²Corresponding author: ndurr@jhu.edu

Speckle artifacts degrade image quality in virtually all modalities that utilize coherent energy, including optical coherence tomography, reflectance confocal microscopy, and ultrasound. We present a deep learning framework for laser speckle reduction, DeepLSR (https://durr.jhu.edu/DeepLSR), that transforms coherently-illuminated images to a speckle-free domain. We apply this method to widefield images of objects and tissues illuminated with a multi-wavelength laser, reconstructing speckle-reduced images with greater fidelity than conventional methods.

Laser illumination offers many advantages for imaging over incoherent light, including high power density, efficient light generation, narrow spectral bandwidth, robust stability, long lifetime, and fast triggering capability. Unfortunately, coherent illumination also introduces speckle artifacts which are caused by constructive and destructive interference between emitted wavefronts. The poor image quality resulting from speckle noise prohibits lasers from being used in many widefield imaging applications. For example, commercial endoscopes utilize arc lamps or light-emitting diodes (LEDs) as illumination sources and consequently require large-diameter light guides to transmit sufficient illumination power. Speckle noise also corrupts image quality in optical coherence tomography (OCT), reflectance confocal microscopy, and ultrasound imaging. To mitigate laser speckle noise, several optical methods and image processing algorithms have been explored. In general, optical approaches add cost and complexity, reduce power throughput, and place fundamental limitations on imaging speed. Image processing techniques, on the other hand, are computationally complex, require parameter tuning, and degrade resolution as they reduce speckle.

Here, we present a deep convolutional neural network for laser speckle reduction (DeepLSR) on widefield images formed from multi-wavelength, red-green-blue laser illumination. We describe a method for effectively learning the distribution of speckle artifacts to target and reduce noise in images not previously seen by the network. This technique relies on pairs of coherent- and incoherent-illuminated images of a variety of objects to learn a transformation from speckled images to speckle-free approximations. Previous work in OCT has explored shallow neural networks for estimating filter parameters in a speckle reduction model, and deep networks for speckle reduction using a set of registered and averaged volumes of retinal tissue as ground truth. In widefield imaging, deep learning networks have been applied for general image denoising, but not specifically for speckle reduction. Our approach is novel in its use of a true incoherent source as a target ground truth, the use of a diverse set of objects for training, and in its applica-
Figure 1: **DeepLSR Architecture.** a) Training architecture for image-to-image translation-based laser speckle reduction using a conditional Generative Adversarial Network. A generator learns to transform between pairs of images acquired with coherent and incoherent illumination while a discriminator learns to classify input images as real or fake. b) Once training is complete, the discriminator is discarded and the trained generator (DeepLSR) reduces laser speckle noise in images not previously seen by the generator.

DeepLSR utilizes a conditional Generative Adversarial Network (cGAN) to reduce laser speckle by posing the problem as an image-to-image translation task. In this way, a structured loss is learned during the training process. This is contrary to unstructured approaches which utilize a per-pixel classification or regression, where each pixel is treated as independent from the others, inhibiting the networks ability to learn from spatial relationships in the image. The overall architecture involves simultaneously training a speckle-free image generator and a real-versus-fake image discriminator, given a conditional input (Fig. 1). While the generator learns to generate a realistic mapping from an input speckled image to an output speckle-free image, the discriminator learns to classify pairs of input and generated output images as either real or fake. During this adversarial training, the discriminator provides feedback on the quality of the image pairs to the generator. The resulting trained generator is then capable of reducing speckle noise in images it has never seen.

We trained and tested DeepLSR using a total of 2,895 images acquired from up to 9 different positions of: (1) 113 assorted household and laboratory objects picked to represent a wide range
of textures, shapes, and bidirectional reflectance distribution functions and (2) ex-vivo porcine esophagus, intestine, and stomach from three animals. These samples were illuminated using a red-green-blue laser for coherent illumination, the same laser with added optical speckle reduction (oLSR) from an oscillating diffuser, and an LED for incoherent illumination. All images acquired from six objects and one porcine were excluded from the training set, and these 30 images were used for testing the trained network (Fig. 2 and Fig. 3). In addition to DeepLSR, which learned a transformation from laser illumination to LED (DeepLSR), we also trained networks to learn transformations from images with optical speckle reduction to LED (DeepLSR+oLSR) and images with laser illumination to optical speckle reduction (Laser → oLSR).

To quantify the performance of DeepLSR, the trained networks were used to despeckle the reserved test images. Output images from DeepLSR were compared to the speckle-free, incoherent images by measuring peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). PSNR assesses the relative noise of an image, while SSIM is a valuable metric for image comparison using quantities that are important for human perception (image contrast, luminance, and structure)\(^{14}\). We also compared DeepLSR to two standard image processing denoising techniques for speckle reduction: median filtering and non-local means. The input parameters for these algorithms were determined by multi-objective optimization, where PSNR and SSIM were used as the objective functions, using the same training data set that was used for DeepLSR. To assess the effect of the network on image resolution, a modulation transfer function (MTF) was measured.

Our validation tests on assorted objects imaged with laser illumination demonstrate that DeepLSR reduces speckle noise by 5.3 dB, compared to a 3.0 dB reduction by non-local means filtering, and a 4.4 dB reduction by oLSR (Supplementary Table 1). We also found that oLSR can be used in combination with deep learning to provide enhanced speckle reduction compared to DeepLSR or oLSR alone (DeepLSR+oLSR). The DeepLSR method has a minor effect on resolution measured by a slanted edge test, as demonstrated from the modulation transfer functions of LED illumination compared to laser illumination with DeepLSR (Fig. 2c).

In applications involving tissue imaging, the object of interest is often a turbid medium that naturally blurs speckle artifacts. To assess the applicability of DeepLSR in this scenario, we applied our model to images of gastrointestinal tissue illuminated with laser light. In these tissue validation tests, DeepLSR reduced speckle noise by 6.3 dB, compared to 2.6 dB reduction by non-local means filtering, and a 3.7 dB reduction by oLSR. Fig. 3 shows representative images of test samples with laser illumination, conventional noise reduction methods, DeepLSR, and LED illumination. As in the assorted objects, DeepLSR removed speckle artifacts while retaining structural features with greater fidelity than conventional imaging processing approaches (Supplementary Table 2).
Figure 2: **DeepLSR compared to conventional speckle reduction methods.** DeepLSR was trained on an assortment of images that represent a variety of textures, shapes, and bidirectional reflectance distribution functions. a) Images of two test objects illuminated with laser illumination, laser illumination with optical speckle reduction (oLSR), median filtering, non-local means, DeepLSR applied to the laser illuminated image, and the target speckle-free image illuminated with a light-emitting diode (LED). b) Speckle artifacts removed from the laser illuminated images by DeepLSR. c) Modulation transfer functions for LED illumination and laser illumination with DeepLSR found using a slanted edge. d) Images of a 1951 United States Air Force Target with each illumination strategy and laser illumination with DeepLSR.
Figure 3: DeepLSR applied to images of laser-illuminated \textit{ex-vivo} porcine gastrointestinal tissues not previously seen by the network.
DeepLSR may be particularly useful in endoscopy applications that require bright illumination or small-diameter endoscopes. Incoherent light sources for endoscopy, such as arc lamps and LEDs, require large-diameter light guides to deliver sufficient optical power through an endoscope. Laser illumination enables the delivery of greater illumination power through fiber optics, and can generate incoherent-like images after DeepLSR or DeepLSR+oLSR is applied. Moreover, in widefield applications that require coherent light, such as laser speckle contrast imaging for mapping flow [13], DeepLSR allows both a computational image and a conventional image to be acquired simultaneously. DeepLSR may also be useful in OCT, ultrasound, and industry applications, such as part inspection. As a data-driven approach, DeepLSR should be trained on images that span the target domain. We have made the DeepLSR model and source code for widefield laser illumination available here: https://durr.jhu.edu/DeepLSR and provide step-by-step instructions for installing and applying this framework to new data sets in the Supplementary Materials.

**Code availability:** The source code and trained models for DeepLSR are available on GitHub (http://durr.jhu.edu/DeepLSR). A tutorial describing the use of DeepLSR and a protocol for training new speckle reduction networks is provided as a supplementary note.

**Data availability:** All data used for training DeepLSR and images used for testing are available through GitHub (http://durr.jhu.edu/DeepLSR).

**Author contributions:** T.L.B., F.M., and N.J.D. designed the experimentation. T.L.B. and M.I. conducted experiments and collected images. F.M. wrote the network code. T.L.B. analyzed data and computed metrics. T.L.B, F.M., and N.J.D. produced the manuscript with input from M.I.
References

1. Goodman, J. W. Some fundamental properties of speckle. *J. Opt. Soc. Am.* **66**, 1145–1150 (1976).

2. Bashkansky, M. & Reintjes, J. Statistics and reduction of speckle in optical coherence tomography. *Opt. Lett.* **25**, 545–547 (2000).

3. Glazowski, C. & Rajadhyaksha, M. Optimal detection pinhole for lowering speckle noise while maintaining adequate optical sectioning in confocal reflectance microscopes. *Journal of Biomedical Optics* **17**, 085001 (2012).

4. Abbott, J. G. & Thurstone, F. Acoustic speckle: Theory and experimental analysis. *Ultrasonic Imaging* **1**, 303–324 (1979).

5. Goodman, J. *Speckle phenomena in optics: Theory and applications* (W. H. Freeman, 2007).

6. Liba, O. *et al.* Speckle-modulating optical coherence tomography in living mice and humans. *Nature Communications* **8**, 15845 (2017).

7. Huang, T., Yang, G. & Tang, G. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **27**, 13–18 (1979).

8. Buades, A., Coll, B. & Morel, J.-M. A Non-Local Algorithm for Image Denoising. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)* 2, 60–65.

9. Adler, D. C., Ko, T. H. & Fujimoto, J. G. Speckle reduction in optical coherence tomography images by use of a spatially adaptive wavelet filter. *Optics Letters* **29**, 2878 (2004).

10. Avanaki, M. R. N., Laissue, P. P., Eom, T. J., Podoleanu, A. G. & Hojjatoleslami, A. Speckle reduction using an artificial neural network algorithm. *Applied Optics* **52**, 5050 (2013).

11. Ma, Y. *et al.* Speckle noise reduction in optical coherence tomography images based on edgesensitive cGAN. *Biomedical Optics Express* **9**, 5129 (2018).

12. Agostinelli, F., Anderson, M. R. & Lee, H. Adaptive multi-column deep neural networks with application to robust image denoising. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1* 1493–1501 (2013).

13. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. Image-to-Image Translation with Conditional Adversarial Networks (2016). [1611.07004](https://arxiv.org/abs/1611.07004)

14. Wang, Z., Bovik, A., Sheikh, H. & Simoncelli, E. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* **13**, 600–612 (2004).
15. Dunn, A. K., Bolay, H., Moskowitz, M. A. & Boas, D. A. Dynamic Imaging of Cerebral Blood Flow Using Laser Speckle. *Journal of Cerebral Blood Flow & Metabolism* **21**, 195–201 (2001).
Methods

Imaging Setup: Paired images for adversarial training were captured using laser illumination, laser illumination with optical speckle reduction (oLSR), and light-emitting diode (LED) illumination. A laser unit with 445nm, 520nm, and 638nm diodes (Optlasers microRGB) and a white-light LED (Luxeon Star Tri-Star) with Rebel 448nm, 530nm, and 655nm diodes were used to illuminate samples. Diodes were selected so that wavelengths emitted by the laser and LED were similar. The laser unit’s beam was positioned normal to the aperture of an Optical Laser Speckle Reducer (oLSR, Optotune LSR-3005-24D), and the oLSR was toggled on and off for imaging with and without speckle reduction. A frosted poly-carbonate triple lens (Luxeon Star #10508) and a 600 grit diffusion lens were used to match the full-width at half maximum of the LEDs illumination intensity profile to that of the laser. For profile-matching, the laser and LED were both aimed at a Teflon imaging target placed below a color, 8-bit CMOS detector (ThorLabs #DCC3240C, 12mm/F1.8) with an integration time set to 50 ms. A linear polarizing sheet (Thorlabs, #LPVISE2X3) was placed in front of the light sources and a linear polarizer (Edmund Optics #47316) was mounted to the detector and adjusted to minimize specular reflection by cross polarization. The illumination intensities of the laser and LED red-green-blue channels were matched to one another by modulating the individual diodes to achieve the same average pixel value and periodic adjustments were made as the diodes’ power drifted over time.

Data Acquisition and Preprocessing: Data was acquired from: (1) 113 assorted household and laboratory objects imaged at up to 9 different positions, with each illumination source, resulting in 1497 images, and (2) 466 images of ex-vivo porcine esophagus, intestine, and stomach from three animals, with each illumination source, resulting in 1398 images. Before training, the histograms of each laser and oLSR image were adjusted to match the corresponding LED image using uniform histogram matching to correct for any white-balancing discrepancies. The 2895 images, were resized from 1280x1024 to 1024x1024 pixels using bicubic interpolation. Images were then divided into sets of laser, oLSR, and LED images to be paired with their corresponding ground truth images for network training. 90 images (39 images from 6 objects, and 51 images acquired from porcine tissue) were removed from the dataset for final network testing. The objects imaged for testing were not seen in the training set and similarly the porcine tissue images for testing came from a different animal than the images used for training.

Network Training: An adversarial deep learning paradigm was used to train the networks. To that effect, two deep networks, a generator and a discriminator, were iteratively trained. The generator was tasked with generating target images and the discriminator was tasked with classifying the generator output as real or fake and giving feedback to the loss function of the generator. This paradigm enables the network to use non-local information when making determinations.

Ground truth similarity was enforced by introducing an $\ell_1$ minimization term to the objective function between the predicted image and the ground truth. By inserting this term, the objective of the generator is not only to fool the discriminator but also to be close to the ground truth output. To prevent mode collapse, images were pooled and fed to the discriminator in batches rather than
individual images in each iteration. Spectral normalization was used to stabilize GAN training when learning simultaneously from assorted objects and tissue\textsuperscript{12}. The problem was solved using ADAM for stochastic optimization\textsuperscript{18}. Further details about the generator and discriminator architectures can be found in\textsuperscript{19} A plot of generator and discriminator loss versus epoch are reported in Supplementary Fig. 1.

The network was trained for 400 epochs. The learning rate was set to 0.0002 for the first 200 epochs and linearly decayed to a learning rate of zero over the remaining 200 epochs. The size of the image buffer that stores generated images was set to 64. The networks were implemented using PyTorch 0.4 and the training was run on Nvidia P100 GPUs using Google Cloud. The average training time for each epoch was 303 seconds and the entire network was trained in approximately 33.66 hours. Once the training process is complete, the trained network computes speckle-reduced images at 6 frames per second on a virtual workstation with 4 CPUs on a 2.6 GHz Intel Xeon E5 processor and at 27 frames per second when using a P100 GPU.

**Evaluation Metrics**: To quantify the performance of DeepLSR, we measured the peak signal-to-noise (PSNR) ratio and structural similarity index (SSIM) using the incoherent, speckle-free image as the target. PSNR assesses the relative noise of an image and was computed using,

\[
PSNR = 10 \log_{10}(\frac{R^2}{MSE})
\]

where \( R \) is the detector’s bit depth of 255 and \( MSE \) is the mean squared error between images. SSIM was computed using,

\[
SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}
\]

where \( x \) and \( y \) are images for comparison, the image mean is \( \mu \), variance is \( \sigma \), covariance between \( x \) and \( y \) is \( \sigma_{xy} \), and the constant \( C \) is added for avoiding instability when the denominator is close to 0. Average image SSIM was calculated using windows of 11x11 pixels. The resultant SSIM index is a value between -1 and 1, where an index of 1 indicates equivalent image inputs. Images of a slanted edge and of a 1951 United States Airforce Resolution Target were used to assess the of effect of the DeepLSR method on image resolution. The Slanted Edge MTF plugin available for ImageJ\textsuperscript{20} was used to formulate a modulation transfer functions.

**Performance**: The three trained networks were tested on the reserved test images of assorted objects and porcine tissue. The average PSNR and SSIM between the network-estimated images and the ground truth images are reported in Supplementary Table 1 and Supplementary Table 2. For performance benchmarks, we report PSNR and SSIM comparisons for laser vs. LED, oLSR vs. LED, and laser vs. oLSR. We also compared DeepLSR to standard image processing denoising techniques: median filtering and non-local means\textsuperscript{7,8} The input parameters for these algorithms
were optimized using the same training data set as was used for DeepLSR. A median filter kernel size of 7 and an NLM kernel size of 4, window size of 5, and a filter strength of 0.283 were utilized.

Additional references
16. Graetzel, C., Suter, M. & Aschwanden, M. Reducing laser speckle with electroactive polymer actuators 943004 (2015).
17. Miyato, T., Kataoka, T., Koyama, M. & Yoshida, Y. Spectral Normalization for Generative Adversarial Networks (2018). [1802.05957]
18. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization (2014). [1412.6980]
19. Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR abs/1511.06434 (2015). [1511.06434]
20. Fiji: an open-source platform for biological-image analysis. Nature Methods 9, 676–682 (2012).
Supplementary Fig. 1: The generator and discriminator loss from training the network to transform laser illuminated images to images illuminated with a light-emitting diode. The networks loss function aims to minimize the generator loss while maximizing the discriminator loss. In this way, the generator learns to produce accurate approximations of output images that frequently fool the discriminator into classifying the images as real target-domain images.
### Table 1: Evaluation with images (n=13) of six assorted objects.

| Method                  | Ground Truth | PSNR (dB) | SSIM    |
|-------------------------|--------------|-----------|---------|
| Laser*                  | LED*         | 25.20±4.36| 0.64±0.16|
| oLSR*                   | LED*         | 29.60±5.08| 0.80±0.10|
| Median Filtering        | LED*         | 26.34±4.35| 0.68±0.13|
| Non-Local Means         | LED*         | 28.19±4.08| 0.82±0.08|
| DeepLSR                 | LED*         | 30.52±4.78| 0.83±0.07|
| DeepLSR+oLSR            | LED*         | 32.76±5.20| 0.86±0.04|
| Laser*                  | oLSR*        | 25.75±3.83| 0.67±0.15|
| Laser → oLSR            | oLSR*        | 29.02±4.37| 0.71±0.15|

*Raw Data*
| Method                        | Ground Truth | PSNR (dB)  | SSIM     |
|-------------------------------|--------------|------------|----------|
| Laser*                        | LED*         | 31.04±1.21 | 0.74±0.03|
| oLSR*                         | LED*         | 34.76±1.10 | 0.93±0.01|
| Median Filtering              | LED*         | 32.58±1.33 | 0.86±0.02|
| Non-Local Means               | LED*         | 33.65±1.40 | 0.94±0.01|
| DeepLSR                       | LED*         | 37.29±1.60 | 0.93±0.01|
| DeepLSR+oLSR                  | LED*         | 39.51±1.98 | 0.95±0.00|
| Laser*                        | oLSR*        | 32.29±1.47 | 0.73±0.05|
| Laser → oLSR                  | oLSR*        | 38.21±1.26 | 0.91±0.02|

* Raw Data

Supplementary Table 2: Evaluation with images (n=17) of porcine gastrointestinal tissues.
Supplementary Note

Step-by-Step Instructions for running DeepLSR

Setting up cloud computing resources to run DeepLSR
DeepLSR requires performance-capable graphics processing units (GPUs) as training is computationally intensive. We utilized Google Cloud computing to train the models reported in this publication.

• In order to run DeepLSR, setup a Google Cloud instance with the necessary dependencies (Ubuntu, PyTorch, and CUDA) by following the instructions provided here: https://cloud.google.com/deep-learning-vm/docs/pytorch_start_instance

★ The code released with this publication was tested using Ubuntu 16.04. We recommend utilizing the same operating system to avoid complications.
★ For best performance, we suggest selecting multiple Nvidia P100 GPUs.

• With the Google Cloud instance setup, a few dependencies must be installed. Install torchvision, dominate (v2.3.1+), visdom (v0.1.8.3+) and scipy using the following commands.

★ For users utilizing Anaconda:
  conda install torchvision -c soumith
  conda install -c conda-forge dominate
  conda install -c conda-forge visdom
  conda install -c anaconda scipy

★ For users utilizing pip:
  pip install torchvision
  pip install dominate
  pip install visdom
  pip install scipy

Training DeepLSR for Laser Speckle Reduction

• Begin by cloning the DeepLSR GitHub repository found at [(https://durr.jhu.edu/DeepLSR)](https://durr.jhu.edu/DeepLSR)
• The directory structure for the dataset should be organized as follows:
  
  ```
  SOMEPATH  # Some arbitrary path
  \|-- Datasets  #Datasets folder
          \|-- XYZ_Dataset  #Active dataset
                  \|-- test
                  \|-- train
  ```

• To utilize the training data referenced in our publication, visit our GitHub repository for instructions.

• If training using your own dataset:
  
  * All test and train data should be in either .jpeg, .jpg or .png formats. .tiff and other raw formats can result in extremely slow training.
  * All data has to be paired side-by-side i.e. the input and output should be concatenated end-to-end in a single image.
  * The size of each individual image should be 2nx2n.
  * Once the dataset is setup use the following command for training:
    ```
    python train.py --dataroot <datapath> --name DeepLSR --gpu_ids 0 --display_id 0 --lambda_L1 70 --niter 200 --niter_decay 200 --pool_size 64 --loadSize <image_size> --fineSize <image_size>
    ```

• Training Parameters:
  --niter is the number of epochs with constant learning rate --niter_decay is the number of epochs with linearly decaying learning rate
  --lr adjusts the learning rate, default = 0.002
  --gpu_ids is the number of GPUs used, 0 means one GPU, 1 means 2 GPUs and -1 means no GPU.
  --lambda_L1 is the \( \ell_1 \) regularization parameter used for the training. This is set to 70 by default because on our data the tuned range of this parameter was [67,74].
  --load_size is the size of the input/output image.
  --fine_size is the size of the random crop from within the image to introduce jitter.

• To view training losses and results, run python -m visdom.server and click the URL [http://localhost:8097](http://localhost:8097) For cloud servers replace localhost with your IP.

• To view epoch-wise intermediate training results, ./checkpoints/DeepLSR/web/index.html

**Testing the Trained Network**
• To test with our pre-trained models, visit our GitHub repository for download instructions.

• Once the model has been uploaded to Google Cloud and configured, run the following code:
  python test.py --dataroot <datapath> --name DeepLSR --gpu_ids 0 --display_id 0 --loadSize <image_size> --fineSize <image_size>

  ⋆ To test with our data <image_size> should be set to 1024.
  ⋆ The test results will be saved to a html file here: ./results/DeepLSR/test_latest/index.html