Generalizing Imaging Through Scattering Media With Uncertainty Estimates

Jared M. Cochrane Matthew Beveridge Iddo Drori
jaredmco@mit.edu mattbev@mit.edu idrori@mit.edu
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139

Abstract

Imaging through scattering media is challenging: object features are hidden under highly-scattered photons. Conventional methods that characterize scattering properties, such as the media input-output transmission matrix, are susceptible to environmental disturbance that is not ideal for many imaging scenarios, especially in biomedical imaging. Learning from examples is ideal for imaging in highly scattered regimes because it is adaptable and accurate even when the microstructures of the scattering media change. In current approaches, network output on unseen scattering media contain artifacts that inhibit meaningful object recognition. We present a network architecture that is able to generate high quality images over a range of different scattering media and image sizes with minimal artifacts. Our network learns the statistical information within highly scattered speckle intensity patterns. This allows us to compute an accurate mapping from different speckle patterns to their corresponding objects given scattering media with varying microstructures. Our network demonstrates superior performance compared to similar models, especially when trained on a single scattering medium and then tested on unseen scattering media. We estimate the uncertainty of our approach and use the available data efficiently, increasing the generalizability of predicting objects from unseen scattering media with multiple different diffusers.

1. Introduction

In biological imaging, tissues act as scattering media that induce aberration and background noise in the captured image, where the true object is faded out. Retrieving the hidden object from the image thus becomes a challenging inverse problem in computational optics. Normally, the properties of the random scattering media are not known and are difficult to fully characterize. Traditional techniques formulate this problem as an optimization based on a transmission matrix or forward operator, with a regularization term derived from the object prior knowledge: $\hat{x} = \underset{x}{\operatorname{argmin}} \|y - Ax\|^2 + \lambda\Phi(x)$, where x is the unknown object with $\hat{x}$ being its estimation, and y is the observed image, A the forward matrix, and a regularization function $\Phi(x)$ with a weighting parameter λ. However, many practical instances of imaging arise when such formulations and methods fail. The nonlinearity that exists in the forward imaging process, especially under heavy light scattering conditions, means that learning from examples is an ideal solution due to the ability to handle nonlinearities.

Real world applications. Motivating this work are real-world applications including (i) Imaging through tissue with visible light, which allows for non-invasive sensing inside the body without exposure to excess radiation, while potentially allowing for better functional imaging than standards today such as the MRI; (ii) Privacy preserving use cases, e.g. human-computer interaction systems, where the agent must observe characteristics of the human but the image of them is obscured to preserve their privacy. Thus, the agent is able to capture essential information without capturing identifying information; (iii) Sensing through dense fog for autonomous navigation (driving, flight, etc.) allows for safe movement in inclement weather; and (iv) Underwater imaging, where turbulence and particulate matter obscure the line of sight.

Instead of solving for the data-fidelity and regularizer by optimization, learning-based methods alternatively model the forward operator and regularizer simultaneously through known objects and their images through random media. A first implementation of this approach [1] used support vector regression learning and successfully learns to reconstruct face objects. However, the fully-connected two-layer architecture fails to effectively generalize from trained face objects to other non-facial object classes. A better network architecture is necessary for more generalizable learning and accurate performance. A U-Net was first proposed for biomedical image segmentation [5]. The skip-

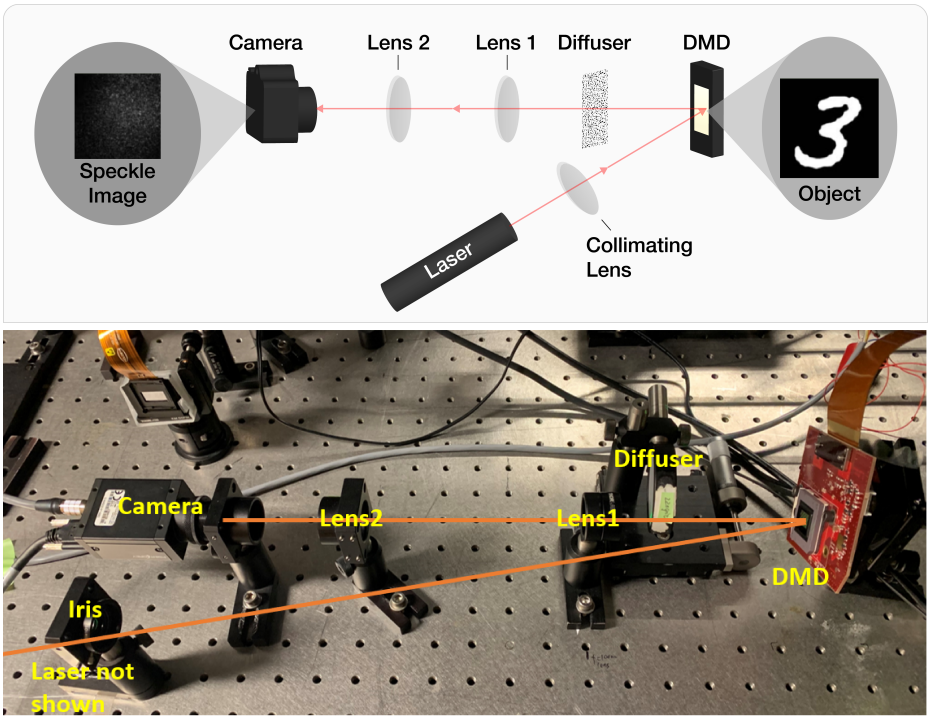


Figure 1: Experimental setup of the scattering media imaging system. Top: schematic of the optical configuration, with an example of a speckle pattern (right) that is mapped to the corresponding ground truth object (left). Bottom: the physical configuration corresponding to the schematic diagram.

connection in the U-Net architecture enables its superiority in extracting image features over other CNN architectures.

Such a U-Net model has been applied to this problem [3] taking a speckle pattern as input and using an encoder-decoder structure to generate high-resolution images. In order to account for the data sparsity that often accompanies computational imaging, the negative Pearson correlation coefficient (NPCC) is used [3] rather than cross entropy as the neural network loss function. The resulting network, called IDiffNet, adapts to different scattering media for sparse inputs, with the NPCC used to learn sparsity as a strong prior in the ground truth values. While IDifNet works well on training and testing data from the same database and distribution, it does not generalize well among different databases, and suffers from overfitting [3].

In a similar fashion, a U-Net is used [4] to map speckle patterns to two output images: the predicted object and background, for a set of different diffusers. Instead of implementing computational imagining as an inverse problem, recent work learns the statistical properties of speckle intensity patterns in a way that generalizes to various different scattering media. Data augmentation may be used to increase the training set size, for example by simulation [7]. In this work we use a new experimental setup, using a digital micromirror device (DMD) instead of a spatial light modulator (SLM) as the pixel-wise intensity object, resulting in speckles size of 10 micrometers instead of 16 micrometers as seen in previous work [4]. When testing on speckles from previously unseen objects through unseen diffusers (types of scattering media), neural networks trained on image sets with multiple diffusers perform better than ones trained on a single diffuser [4]. While previous work may be generalized to multiple different diffusers within the same class, a limitation are obvious artifacts, such as discontinuities, which are observed in the object prediction.

In this work, we use a U-Net architecture to learn the forward operator and regularizer that generate high-quality imaging to show the statistical properties of speckle patterns[1]. We accomplish this in a manner that allows our model to adapt to different diffuser microstructures. We also explore how to make better use of limited experimental data by comparing the performance, output object accuracy, and generalizability among different loss functions as well as data compositions.

[1]Code available at https://www.dropbox.com/s/rpioiafjxdupz4k/Scattering-main.zip?dl=0

2. Methods

2.1. Experimental Setup and Data Acquisition

Our experimental setup is illustrated and shown in Figure 1. Light from a laser source is first collimated and then illuminates onto a digital micro-mirror device (DMD, DLP LightCrafter 6500, pixel size 7.6 micron). The DMD, placed at a certain angle relative to the illumination beam, acts as a pixel-wise intensity object. After modulation by DMD, the beam passes through a thin glass diffuser (Thorlabs, 220 grits, DG10-220) and gets scattered. The resulting image is then relayed by a two-lens telescope imaging system onto a camera (FLIR, Grasshopper 3, pixel size 3.45 micron). The focal lengths of lens 1 and lens 2 are 150mm and 100mm, respectively, which provide a magnification of 0.67. The speckle size of the system is calculated by taking the autocorrelation of a speckle pattern through a diffuser and measuring the full width at half-maximum, which is $\sim 20\mu m$.

The central 512×512 DMD pixels are used as the object; the corresponding central 750×750 camera pixels are used as the speckle intensity for training and testing. The objects displayed on the DMD are binary images adopted from MNIST. In total, we use 600 objects and collect the speckle images for 8 different diffusers. The data from diffusers 1-4 are used in training, where the images of the first 550 objects are training data and the remaining 50 are validation data. This means that 2200 data pairs are used in total for training. The data from diffusers 5-8 are used in testing, where we characterize the output accuracy of the network by testing on seen objects (object 1-550) and unseen objects (object 551-600) from unseen diffusers.

As a pre-processing step, the input and output images are down-sampled to 128×128 and 256×256 for computational efficiency. Next, the input speckle images are transformed to grayscale. The output is a single channel produced by a sigmoid layer that represents the object. The input and ground truth pairs are converted to tensors. For each diffuser, our network is trained on 550 images and validated on 50 images. Data loaders are constructed with a batch size of 32.

2.2. Neural Network Implementation

A major difference between our U-Net and previous work [4] is that the later uses a two-channel network that splits each input image into two tensors: one for the object itself and another for the background. In contrast, our U-Net considers only a single channel outputted through a sigmoid activation layer. This produces a clearer reconstruction as shown by comparing Figure 5 to the output of [4]. In our U-Net, each convolutional layer is replaced with a dense block. Our U-Net model is separated into an encoder and decoder. The encoder uses five layers, each consisting of 2D Convolution-ReLU-Dense Blocks followed by max pooling, to reduce the lateral size of the image while increasing the number of tensors in the channel dimension. The convolutional kernel is size 3×3 and the dense kernel is 5×5. The decoder uses a similar series of operations joined by up-sampling and concatenation in the channel dimension with the corresponding encoder layer. This re-expands the image lateral size and results in the number of channel-dimension tensors to be one output image.

Each dense block consists of several subsequent convolutional blocks. During encoding, this convolutional block series is repeated four times, while decoding has this series repeated only three times. The basic structure of a convolutional block consists of batch normalization, ReLU, convolutional layer, and conditional drop-out with probability of 0.5. The resulting feature maps from these subsequent convolutional blocks are concatenated in the channel dimension. The up-sampling function consists of three layers: nearest-neighbor up-sampling, 2D convolution, and ReLU. The up-sampling is used in the decoding part of the network, which is iteratively followed by concatenation with the previous dense block outputs in the channel dimension.

Our network is trained using stochastic gradient descent with momentum. During training, the batch is forward propagated through the model, the loss is computed and back propagated, the tracked gradients for the modules are zeroed, and the step function applied to the optimizer. During evaluation, the model is validated using previously unseen validation data. The training loss and validation loss are computed for each epoch. Commonly used loss functions including mean squared error (MSE) and mean absolute error (MAE) do not promote sparsity since they assume the underlying signals follow Gaussian and Laplace statistics, respectively. Considering the high sparsity in the MNIST database, we consider two more appropriate candidates for the loss function: the negative Pearson correlation coefficient (NPCC) and average binary cross entropy (BCE):

$$L_{NPCC} = -\frac{\sum_i (x-\tilde{x})(p-\tilde{p})}{\sqrt{\sum_i (x-\tilde{x})^2}\sqrt{\sum_i (p-\tilde{p})^2}}, \quad (1)$$

$$L = -\frac{1}{2N}\sum_i \left(x\log(p) + (1-x)\log(1-p)\right), \quad (2)$$

where $\tilde{x}$ and $\tilde{p}$ are the average ground truth x and network output p, and i indexes each of the N pixels of the image.

2.3. Uncertainty Estimates using Dropout

Dropout is commonly used for neural network regularization during training. In this work, we would also like to learn the posterior over the network weights $p(\theta|x,y) = \frac{p(y|x,\theta)p(\theta)}{p(y|x)}$. However, this in intractable. We therefore estimate uncertainty using dropout. Specifically, we approxi-

Loss	D_1	$D_1 \sim D_2$	$D_1 \sim D_4$
BCE	0.773	0.802	0.815
NPCC	0.746	0.752	0.750

Table 1: Pearson correlation coefficient (PCC) for validation data sets using different loss functions. Binary cross entropy (BCE) and negative Pearson correlation coefficient (NPCC) are two candidate loss functions, while the PCC measures the output image quality.

mate the posterior by sampling using dropout [2]. We train the network using dropout and then test each example x by running multiple forward passes with dropout weights. For $i = 1 \dots n$ where $n = 100$, we sample n binary masks from a Bernoulli distribution with probability p, such that $m_i \sim \mathit{Ber}(p)$. In this case, we use $p = 0.1$ to generate the Bernoulli masks. We then use $\theta_i = \theta \odot m_i$, where $\odot$ denotes point-wise multiplication, to compute the mean:

$$\mathbb{E}(\hat{y}|x) = \frac{1}{n}\sum_{i=1}^{n} f(x|\theta_i), \tag{3}$$

and use the mean to compute the variance:

$$v(\hat{y}|x) = \frac{1}{n}\sum_{i=1}^{n} f(x)^2 - \mathbb{E}(\hat{y}|x)^2, \tag{4}$$

as an approximation of uncertainty.

3. Results

3.1. Different Loss Functions

First, we compare two loss functions: Binary Cross Entropy (BCE) and the Negative Pearson Correlation Coefficient (NPCC). The reconstruction losses of the validation data using these two functions are shown in Table 1. For each loss function, we train with one, two, and four different diffusers.

As seen in Table 1, BCE loss achieves higher validation accuracy regardless of the number of different diffusers used in training.

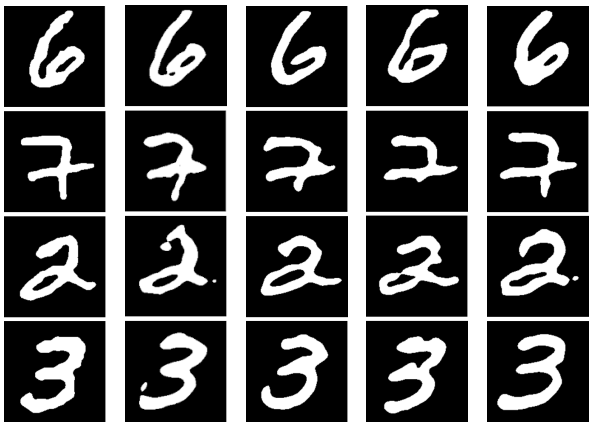
We interpret this to be the intrinsic power of the BCE loss function, where both false positive and false negative outputs are penalized. In contrast, NPCC only rewards the true positives, resulting in inevitable false positives. Thus, for the rest of this work we use the BCE loss for training. Note that the results in the below table apply to a network that is trained and tested on the same set of diffusers.

3.2. Predict Unseen Objects From Unseen Diffusers

Our second task is to predict unseen objects from unseen diffusers, where the set of objects has never been used for training. Representative examples are shown in Figure 2, demonstrating that our network is able to make high-quality predictions of these unseen objects through unseen diffusers. It is not surprising to see this result since the unseen object and training object belong to the same handwritten digits. The network is able to make predictions of the same object from different diffusers with equal quality.

Samples	D_1	$D_1 \sim D_2$	$D_1 \sim D_4$
550	0.517	0.568	0.633
1100	-	0.607	0.658
2200	-	-	0.676

Table 2: Pearson correlation coefficient (PCC) for unseen objects through unseen diffusers using binary cross entropy (BCE) as the loss function.



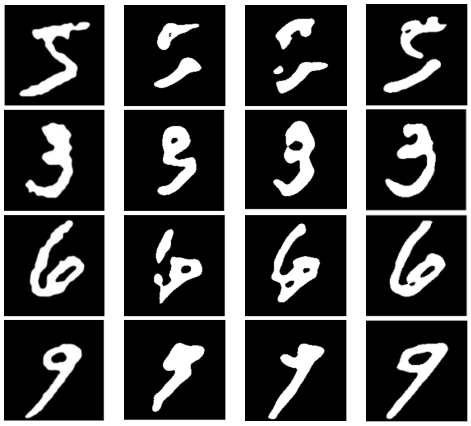
(a) (b) (c) (d) (e)

Figure 2: Multiple diffusers: Training on diffusers $D_1 \sim D_4$ and testing results of reconstructing unseen objects through unseen diffusers $D_5 \sim D_8$. Despite the differences across the speckle patterns, the reconstructions are of high quality. From left to right: (a) Ground truth, (b) D_5 test output, (c) D_6 test output, (d) D_7 test output, (e) D_8 output.

3.3. Generalizability

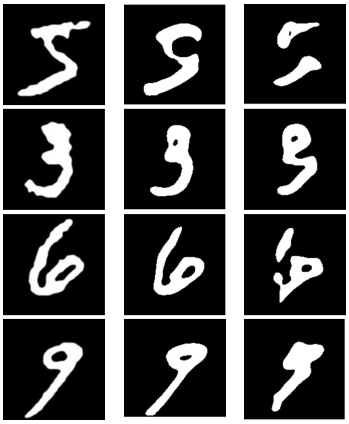
To evaluate the improvement of network performance, we use a different training data composition and compute the average PCC of unseen objects from all four unseen diffusers.

Similar to previous work [1], the network is able to reconstruct unseen objects when trained and tested on the same diffuser. In contrast with previous work, using a different experimental setup and network architecture, our design also successfully performs the same task as shown in Figure 4. A U-Net trained on only a single diffuser cannot



(a) (b) (c) (d)

Figure 3: Increasing the number of training diffusers: Training on data from one, two, and four diffusers, respectively. Testing results of unseen objects through unseen diffusers. Reconstruction quality increases with the number of diffusers used in training. From left to right: (a) ground truth, (b) test output from network trained on D_1, (c) test output from network trained on D_1 and D_2, (d) test output from network trained on D_1-D_4.

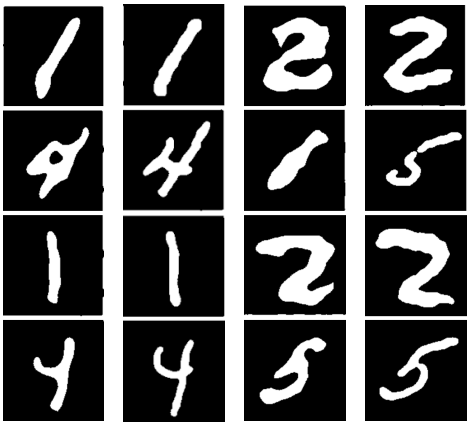


(a) (b) (c)

Figure 4: Single diffuser training: Testing results of unseen objects with the network trained on data from a single diffuser. While the network reconstructs an unseen objects using the same diffuser that it is trained on, it fails on speckles from a different unseen diffuser. From left to right: (a) ground truth, (b) test results from seen diffuser, (c) test results from unseen diffuser.

be reliably generalized to other diffusers, since it is tuned to fit only to the model of a specific diffuser.

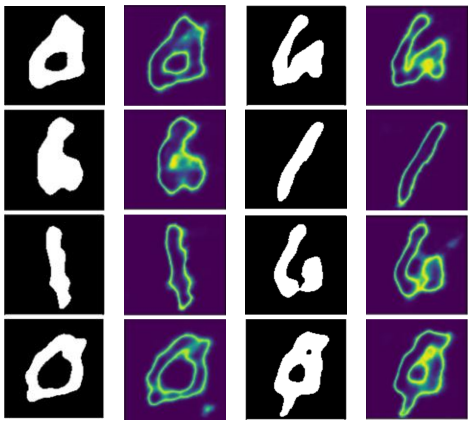
Next, we split the training data into different compositions. When training with one diffuser, we use the full size of the 550 data; when training with two diffusers, we use data sizes of 550 and 1100; and when training with four diffusers, we use data sizes of 550, 1100 and 2200. Table 2 shows the average PCC of unseen objects in each case. In general, the performance of the network improves as more diffusers are used. Performance also improves by increasing the size of the training data, as seen by a comparison within each column. In addition, using four diffusers with 550 data has better performance than using two diffusers with 1100 data, which further verifies the effectiveness of training with multiple diffusers. Additional representative sample output images are shown in Figure 3.



(a) (b) (c) (d)

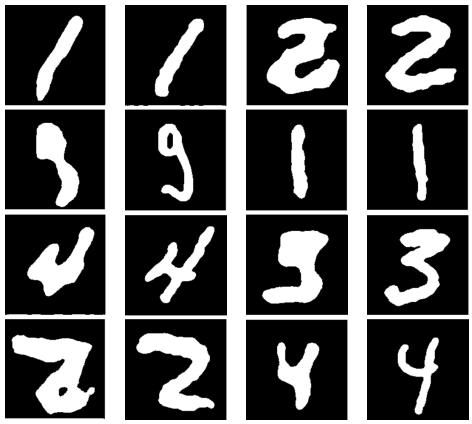
Figure 5: Testing results of unseen objects through unseen diffusers from the network trained with data from D_1 and tested on D_2 and D_3. From left to right: (a) ground truth for D_2, (b) corresponding network output, (c) ground truth for D_3, (d) corresponding network output.

We also trained our network on an image size of 256×256. In this case, for efficiency, our network is trained on a single diffuser, D_1, with batch size 32 and image size 256×256 and tested on two unseen diffusers (D_2 and D_3). While the network in previous work [4] exhibits exceptionally poor imaging quality on unseen diffusers, our network is able to output images of much higher quality: generally high enough to be meaningfully identified as shown by several representative output samples in Figure 5 for both D_2 and D_3. We compare the results to those in [4]. The continuous nature of our network output, as compared to the previous model, is be explained by the fact that previous work uses a two-channel output from a softmax layer. In contrast, our U-Net architecture uses only a single channel of output with a sigmoid activation.



(a) (b) (c) (d)

Figure 6: Uncertainty results from model output. The difference images in (b) and (d) are computed by subtracting the model output minus the output variance. From left to right: (a) model output, (b) corresponding difference between output and variance (c) model output, (d) corresponding difference between output and variance.



(a) (b) (c) (d)

Figure 7: Sample output from a network trained on D_1 and tested on D_2 with dropout rate $p = 0.1$. From left to right: (a) model output, (b) corresponding ground truth, (c) model output, (d) corresponding ground truth.

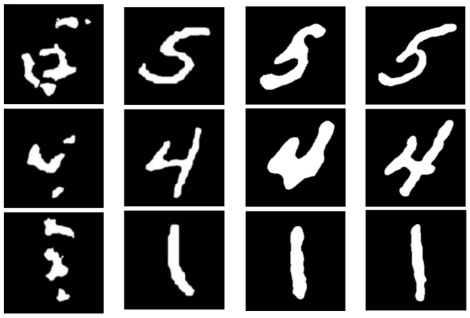
3.4. Uncertainty Estimates

In order to visualize the uncertainty of the network's output, we perform the computations detailed in Section 2.3. We train the U-Net with a dropout rate of $p = 0.1$ to calculate $n = 100$ forward passes through the network. The U-Net's parameter matrix is point-wise multiplied with a Bernoulli mask according to $m_i \sim \mathit{Ber}(p)$. These forward passes are then averaged and subtracted from the average square of the forward propagation in the original model to yield a matrix of variances. Figure 6 depicts several examples of network outputs followed by the differences between the output and its corresponding variance.

Following the notation in Section 2.3, $f(x)$ is the network output, x is the input image, and $v(\hat{y}|x)$ is the model variance. In Figure 6, columns (a) and (c) are the network's output, $f(x)$, while columns (b) and (d) are the difference between the model output and variance $f(x) - v(\hat{y}|x)$.

3.5. Changing the Dropout Rate

It is important to note that the results up until this point have all assumed a dropout rate of $p = 0.5$. However, this high of a dropout rate gives rise to significant noise in the network output and thus the model was retrained with a lower dropout rate in order to improve its accuracy, as measured by the PCC. Given the updated dropout of $p = 0.1$, the figure below shows sample outputs from a network trained on D_1 and tested on D_2. Furthermore, lowering the dropout rate raised the average PCC for the model output from 0.517, as given in Table 2, to 0.749.



(a) (b) (c) (d)

Figure 8: Comparison of predictions on an unseen difuser between previous work and our work. (a) Previous work results by Li et al. [4] prediction with an unseen difuser and ground truth (b). (c) Our prediction result with an unseen difuser and ground truth (d).

3.6. Comparison with Previous Work

Figure 8 compares our work with previous work, specifically the generalization quality when training on one diffuser and testing on another. The Figure shows a side-by-side comparison. Our work demonstrates the ability to accurately reconstruct speckle patterns from diffusers that the network was not trained on. The network trained on one diffuser accurately reconstructs speckle patterns from another diffuser. Figure 8 shows that this represents a significant improvement over previous work.

3.7. Limitations

Inherent limitations of our setup are threefold: our model (i) assumes a linear imaging system, (ii) only applies within the context of planar imaging, and (iii) requires an active laser source [6]. Next, we elaborate on each of these limitations. First, our imaging model assumes that the model output is determined by convolving an object with a point spread function that both vary across the image's horizontal and vertical dimensions with added noise. The object is determined by how the media interacts with incoming light, and the point spread function is determined by the properties of the imaging system. In this case, the object is the speckle pattern and the point spread function is represented by the U-Net weight matrix. While this offers a relatively simplified model of the imaging process, it is commonly assumed in practice [6]. Second, our model only applies to imaging in planar regimes. Finally, our model makes use of an active laser source. Acquiring, calibrating, and correctly positioning such a laser source may be expensive and complicated.

4. Conclusions

Current methods of imaging through scattering media have a difficult time generating consistently accurate results across multiple unseen scattering media. Instead of conceptualizing highly-scattered computational imaging as a linear inverse problem, this work creates and trains a network that is able to generalize and perform accurately even when the scattering medium changes. Using an encoder-decoder U-Net architecture with a binary cross entropy loss function, our implementation is able to generate accurate outputs for seen objects through unseen diffusers. Perhaps more importantly, our network is also able to generate accurate outputs for unseen objects through unseen diffusers given training on multiple different diffusers. In addition, our network trained on a single diffuser D_1 is able to generalize to unseen diffusers D_2 and D_3 with relatively high image quality, which demonstrates the robustness of the encoder-decoder U-Net architecture for imaging through highly scattered media.

References

[1] Ryoichi Horisaki, Ryosuke Takagi, and Jun Tanida. Learning-based imaging through scattering media. *Optics express*, 24(13):13738–13743, 2016.

[2] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[3] Shuai Li, Mo Deng, Justin Lee, Ayan Sinha, and George Barbastathis. Imaging through glass diffusers using densely connected convolutional networks. *Optica*, 5(7):803–813, 2018.

[4] Yunzhe Li, Yujia Xue, and Lei Tian. Deep speckle correlation: A deep learning approach toward scalable imaging through scattering media. *Optica*, 5(10):1181–1190, 2018.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[6] Guy Satat. *All photons imaging: Time-resolved computational imaging through scattering for vehicles and medical applications with probabilistic and data-driven algorithms*. PhD thesis, Massachusetts Institute of Technology, 2019.

[7] Fei Wang, Hao Wang, Haichao Wang, Guowei Li, and Guohai Situ. Learning from simulation: An end-to-end deep-learning approach for computational ghost imaging. *Optics express*, 27(18):25560–25572, 2019.