

# CohereNet: A deep learning approach to coherence-based beamforming

Alycen Wiacek\*, Eduardo Gonzalez†, Najim Dehak\*, and Muyinatu A. Lediju Bell\*†‡

\*Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD

†Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD

‡Department of Computer Science, Johns Hopkins University, Baltimore, MD

**Abstract**—Short-lag spatial coherence (SLSC) beamforming has the potential to improve the diagnostic power of a multitude of ultrasound imaging techniques. One challenge for advanced real-time implementation is repeated correlation calculations. To address this challenge, this paper introduces CohereNet – a novel deep neural network architecture that estimates the coherence function in efforts to bypass the repeated correlation calculations required for SLSC imaging. The network was trained and evaluated using *in vivo* breast data, demonstrating similar contrast, CNR, SNR, and GCNR with an average correlation between the original image and the DNN image of 0.93, and improved computational speed (i.e., a factor of 3.4 improvement) when compared to the offline implementations. In addition, the model is generalizable across multiple tissue types, probe geometries, and ultrasound systems. These results are promising for the use of deep learning architectures as a replacement for correlation estimation in multiple areas of coherence-based ultrasound imaging.

## I. INTRODUCTION

A multitude of ultrasound image formation and signal processing techniques rely on fundamental cross-correlation calculations, including speckle tracking, elastography, and advanced beamforming techniques. These methods are advantageous because of their improvements in image quality or their additional insight into the disease status of tissue. For example, speckle tracking can be used to characterize cardiac function by assessing the deformation of the ventricles [1], elastography relies on temporal correlation measurements of tissue after the application of a radiation force impulse in order to measure elastic properties of tissue [2], and advanced beamforming techniques such as minimum variance and short-lag spatial coherence (SLSC) beamforming improve image resolution and contrast, respectively [3], [4]. Each of these techniques require repeated correlation calculations, which are often considered as the time-consuming bottleneck to real-time processing and image display.

In order to bypass the repeated correlation steps required for many of these techniques, we propose a custom Deep Neural Network (DNN) to be used as a universal approximator in order to estimate the coherence function for applications in coherence-based beamforming, specifically SLSC [4]. The network is constructed to model the computational structure of the coherence function, and consists of a custom fully connected (FC) architecture with the goal of estimating the coherence function, while preserving the quality of SLSC images. Because our goal is to apply this neural network to *in vivo* data from different patients and regions of the body,

generalization is critical for successful application. In this paper, we introduce CohereNet — a custom DNN trained with *in vivo* breast data to estimate the coherence functions required to create SLSC images. We also explore the generalizability of CohereNet to other types of phantom and *in vivo* tissue, as well as a range of probe geometries.

## II. METHODS

### A. Short-lag Spatial Coherence (SLSC) Implementation

SLSC relies on the spatial coherence of backscattered pressure waves that are received across the ultrasound transducer. Once the data is delayed, normalized correlation measurements are calculated between equally spaced elements, or lags, resulting in the normalized spatial correlation:

$$\hat{R}(m) = \frac{1}{N-m} \sum_{i=1}^{N-m} \frac{\sum_{n=n_1}^{n_2} s_i(n) s_{i+m}(n)}{\sqrt{\sum_{n=n_1}^{n_2} s_i^2(n) \sum_{n=n_1}^{n_2} s_{i+m}^2(n)}} \quad (1)$$

where  $N$  is the number of receive elements in the transducer,  $m$  is the lag,  $s_i(n)$  is a time-delayed, zero-mean signal received at element  $i$  from depth  $n$ .

This resulting spatial coherence function is then summed up to a specific short-lag value,  $M$ , yielding the value of the SLSC pixel. This process is repeated for each lateral and axial position in the image, with an axial correlation kernel,  $k$ , of 7 samples. For the purposes of explanation,  $s_k(n) \in \mathbb{R}^{k \times N}$  will be referred to as one axial correlation kernel of ultrasound channel data across the receive aperture.

The SLSC algorithm is implemented through a Matlab mex function on a central processing unit (CPU). In addition, an optimized implementation exists and is detailed in [5] which runs on a graphical processing unit (GPU). The main difference in computation between the original CPU implementation of SLSC and the GPU implementation is in the correlation calculation, which uses mathematical simplifications in order to perform pixel-wise correlations and allow for parallel processing on a GPU. Additional details can be found in [5]. The GPU implementation was used in this study for comparison of image quality and speed with the DNN SLSC implementation.

### B. Deep Neural Network

A custom DNN was implemented using Keras [6] with Tensorflow [7] with an architecture shown in Fig. 1. The input to the network was one axial kernel of delayed channel data,  $s_k(n)$ , and the output was the associated coherence function,

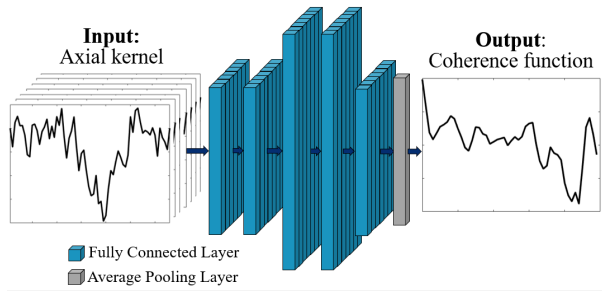


Fig. 1. Graphic depiction of the network architecture.

$\hat{R}$ , over the entire receive aperture. Due to the geometry of the receive array, the dimensionality of the input data is  $7 \times 64$  (i.e., one axial kernel of the entire receive aperture) and the output is  $1 \times 64$  (i.e., the coherence function for all lags). In order to include data without zeros associated with the receive aperture, the channel data was delayed and filtered accordingly prior to being used as training data. The ground truth coherence functions were generated using a MATLAB (MathWorks Inc., Natick, MA, USA) implemented mex function for each lateral and axial position, as described in Section II-A.

The network was trained using a modified mean squared error (MSE) loss function:

$$MSE = \frac{1}{m} \sum_{i=1}^m w_i (y_i - \hat{y}_i) \quad (2)$$

where  $m = 64$  is the number of lags being computed,  $y_i$  is the computed coherence function,  $\hat{y}_i$  is the ground truth coherence function, and  $w$  is a vector of Gaussian weights with  $\mu = 0$  and  $\sigma = 0.025$ . The custom weighting scheme was implemented in order to place higher penalty on the shorter lags, which are typically used to create an image, and therefore more important to image quality.

The network was trained using a batch size of 128 using an ‘adam’ optimizer for 5 epochs with a learning rate of 0.001. The training and validation data were loaded using a custom Keras class in order to ensure data was not repeated and was randomized between each epoch. The PC used for this process was an Intel Core i5-6600k CPU with 32GB of RAM alongside an Nvidia GTX Titan X (Pascal) with 12GB of VRAM and a core clock speed of 1531MHz.

### C. Dataset

The training data were taken from a dataset of 24 unique ultrasound scans of *in vivo* breast masses, obtained after informed consent and approval from the Johns Hopkins Medicine Institutional Review Board [8]. Data were collected using an Alpinion ECUBE-12R research ultrasound scanner connected to an Alpinion L8-17 linear array ultrasound transducer (Alpinion, Seoul, South Korea). The transducer has 128 elements, with  $N = 64$  allowed to receive at one time. The data were split into training, validation, and testing sets, with 18 patients used for training, 3 for validation, and 3 for testing to ensure generalization across patients. Each patient

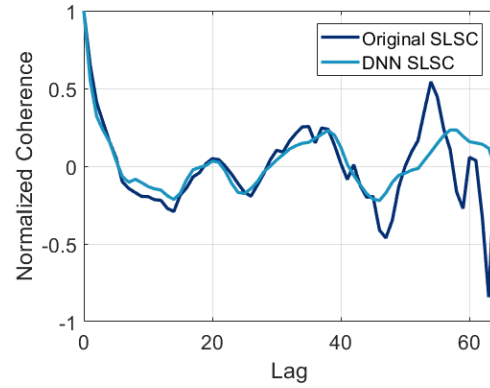


Fig. 2. Example coherence function where the original coherence function is shown in dark blue, and the DNN computed coherence function is in light blue.

contains 2 orthogonal acquisitions (radial and anti-radial), with 10 frames per acquisition. When deconstructed into training examples, this results in 92.2 million training examples and 15.4 million for each validation and testing.

In order to test the ability of the network to generalize across different types of data, testing was also performed using *in vivo* liver data (which was described in a previous publication [9]), data from a CIRS Model 054GS phantom data (CIRS, Norfolk, VA), phased and curvilinear arrays, and the Alpinion and Verasonics ultrasound systems (Verasonics, Kirkland, WA).

### D. Quantitative metrics

Contrast, signal-to-noise ratio (SNR), contrast-to-noise ratio (CNR), and generalized contrast-to-noise ratio (GCNR) [10] were measured and compared across matched coherence-based images formed using each of the CPU, GPU, and DNN algorithms created with the same channel data according to the following equations:

$$\text{Contrast} = 20 \log_{10} \left( \frac{S_i}{S_o} \right) \quad (3)$$

$$\text{SNR} = \frac{S_o}{\sigma_o} \quad (4)$$

$$\text{CNR} = \frac{|S_i - S_o|}{\sqrt{\sigma_i^2 + \sigma_o^2}} \quad (5)$$

$$\text{GCNR} = 1 - \sum_{x=0}^1 \min_x \{p_i(x), p_o(x)\} \quad (6)$$

where  $S_i$  and  $\sigma_i$  are the mean and standard deviation, respectively, within a region of interest (ROI) inside of the target,  $S_o$  and  $\sigma_o$  are the mean and standard deviation, respectively, of a region of interest outside of the target, and  $p_i$  and  $p_o$  are the probability density functions of the signal inside and outside the target, respectively.

### III. RESULTS

Fig. 2 shows an example coherence function using the deep learning model. Overall, the DNN predicted the spatial coherence with a MSE of 0.037. The DNN SLSC coherence function (shown in light blue) fits the shorter lags ( $<40$ ) better than the larger lags, due to the weighted loss function used to train the network (as mentioned in Section II-B).

Fig. 3 shows one example from the test set of *in vivo* breast data formed from the same channel data using the original CPU algorithm, the GPU algorithm, and the DNN. Fig. 3(b) shows the GPU SLSC image which appears brighter and more smoothed than the original SLSC image of Fig. 3(a). The DNN SLSC image (i.e., Fig. 3(c)) is qualitatively more similar to the original SLSC image both in texture and in brightness.

To quantify the image quality differences, Fig. 4 shows the contrast, SNR, CNR, and GCNR for each of the images from Fig. 3. First, confirming the qualitative observations above, the contrast, SNR, CNR, and GCNR were all similar when the original SLSC results are compared to the DNN SLSC results (and both have less similarity with the GPU SLSC results). The correlation between the original SLSC image and the GPU SLSC image was 0.81, and the correlation between the original SLSC image and the DNN SLSC image was 0.94. The contrast of the original, GPU, and DNN SLSC was -5.2 dB, -3.2 dB and -5.5 dB, respectively. The SNR for the original and DNN SLSC were 2.7 and 2.5, respectively, while the GPU SLSC was 3.1. The CNR for both the original and DNN were approximately 1.0 and the CNR for the GPU SLSC was 0.7. Finally, the GCNR of the original SLSC, and DNN SLSC were both approximately 0.6, and the GCNR of the GPU SLSC was 0.4. Over the entire test set (i.e. 60 frames from 3 patients not included during training or validation), the mean contrast difference between the original SLSC image and the DNN SLSC image was 0.6dB, and the mean SNR, CNR, and GCNR differences were 0.07, 0.09, and 0.04, respectively and the average correlation was 0.93.

The computation time for the original SLSC implemented on CPU was 2.6 s. On GPU, this processing time was 0.4 s, corresponding to a 6.4x improvement. The improvement compared to the original SLSC for the DNN SLSC was 3.4x.

In order to test generalizability, Fig. 5 shows different test examples including *in vivo* liver data [9] as well as the CIRS phantom. Figs. 5(a)-(d) show the results from a linear array, demonstrating that the DNN SLSC images are generally qualitatively similar to the original SLSC image. Figs. 5(e)-(f) and 5(g)-(h) show results from the curvilinear and phased arrays, respectively. Both pairs of images are qualitatively very similar. Finally, Figs. 5(i)-(j) show results from the Verasonics phased array. Overall, on the entire additional test set, there was an average correlation of 0.96 between the original SLSC image and the DNN SLSC image.

### IV. DISCUSSION

There are three main advantages to a deep learning approach to coherence-based beamforming. First, by training a deep

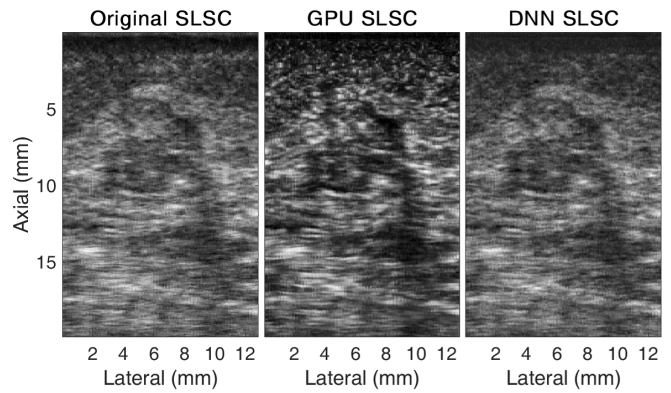


Fig. 3. Example ultrasound images generated from (a) the original SLSC algorithm on CPU, (b) GPU SLSC, and (c) DNN SLSC.

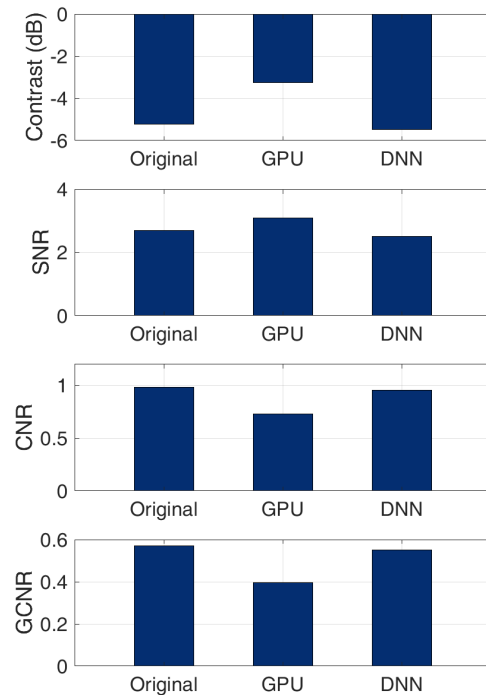


Fig. 4. The contrast (a), SNR (b), CNR (c), and GCNR (d) of each type of image shown in Fig. 3.

neural network to estimate the coherence function, the repeated lateral and axial calculations can be bypassed, therefore creating images faster than the original CPU-based approach. This opens up the possibility of integrating additional, more complex algorithms into the processing pipeline.

Second, the DNN SLSC image is more similar both qualitatively (as shown in Fig. 3) and quantitatively (as shown in Fig. 4) to the original SLSC algorithm when compared to the GPU SLSC algorithm. This is likely due to the simplifications required to compute the average correlation for a specific lag value on the GPU. Because the neural network architecture is modeled after the mathematical correlation calculation, the

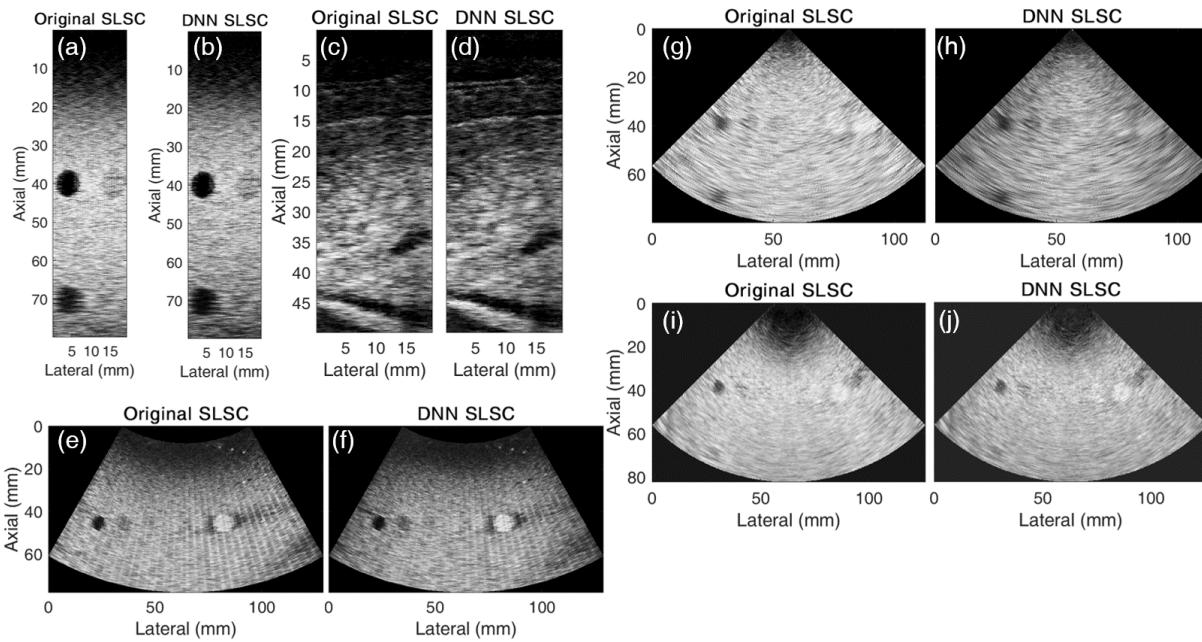


Fig. 5. Example ultrasound images on the left using the original SLSC algorithm and on the right the DNN of the CIRS phantom using (a) and (b) a linear array, (e) and (f) curvilinear array, (g) and (h) an Alpinion phased array, (i) and (j) a Verasonics phased array and (c) and (d) in *in vivo* liver using a linear array.

network is learning the coherence function itself.

Third, the learned features allow the network to be generalizable across multiple ultrasound imaging platforms and multiple transducer types as shown in Fig. 5, as these differences introduce changes only to the raw ultrasound data and not the deep learning architecture. This generalizability is promising, considering that the coherence information is used in many other ultrasound applications, making this network useful for beamforming algorithms other than SLSC and its many derivatives including minimum variance beamforming and coherence-weighted imaging.

While the GPU algorithm offers better speed improvements than the DNN SLSC implementation, there is additional room for improvement.

## V. CONCLUSION

This paper demonstrates the first use of deep learning to estimate spatial coherence functions in order to form coherence-based ultrasound images. The trained, custom DNN (i.e., CohereNet) resulted in images with an average correlation of 0.93 between the calculated image and the DNN image, as well as similar image quality metrics including contrast, SNR, CNR and GCNR. The DNN approach improves computational speed by 3.4x compared to the offline implementation. The generalizability of this network across different tissue and transducer types is promising for the use of deep learning for coherence estimation in multiple areas of ultrasound imaging where coherence information is required.

## VI. ACKNOWLEDGEMENT

The authors thank Dongwoon Hyun for sharing his Verasonics GPU SLSC code.

## REFERENCES

- [1] H. Geyer, G. Caracciolo, H. Abe, S. Wilansky, S. Carerj, F. Gentile *et al.*, "Assessment of myocardial mechanics using speckle tracking echocardiography: fundamentals and clinical applications," *Journal of the American Society of Echocardiography*, vol. 23, no. 4, pp. 351–369, 2010.
- [2] J.-L. Gennisson, T. Defieux, M. Fink, and M. Tanter, "Ultrasound elastography: principles and techniques," *Diagnostic and interventional imaging*, vol. 94, no. 5, pp. 487–495, 2013.
- [3] J.-F. Synnevag, A. Austeng, and S. Holm, "Benefits of minimum-variance beamforming in medical ultrasound imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 56, no. 9, pp. 1868–1879, 2009.
- [4] M. A. Lediju, G. E. Trahey, B. C. Byram, and J. J. Dahl, "Short-lag spatial coherence of backscattered echoes: Imaging characteristics," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 58, no. 7, pp. 1377–1388, 2011.
- [5] D. Hyun, A. L. C. Crowley, M. LeFevre, J. Cleve, J. Rosenberg, and J. J. Dahl, "Improved visualization in difficult-to-image stress echocardiography patients using real-time harmonic spatial coherence imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 66, no. 3, pp. 433–441, 2018.
- [6] F. Chollet *et al.*, "Keras," <https://keras.io>, 2015.
- [7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <http://tensorflow.org/>
- [8] A. Wiacek, O. M. H. Rindal, E. Falomo, K. Myers, K. Fabrega-Foster, S. Harvey *et al.*, "Robust short-lag spatial coherence imaging of breast ultrasound data: Initial clinical results," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 66, no. 3, pp. 527–540, 2018.
- [9] A. A. Nair, T. D. Tran, and M. A. L. Bell, "Robust short-lag spatial coherence imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2017.
- [10] A. Rodriguez-Molares, O. M. H. Rindal, J. D'hooge, S.-E. Måsåøy, A. Austeng, and H. Torp, "The generalized contrast-to-noise ratio," in *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2018, pp. 1–4.