DEEP CONVOLUTIONAL-DECONVOLUTIONAL NEURAL NETWORK FOR ULTRASONIC TOMOGRAPHY

Wenbo Zhao, Zhun Chen, Yuanwei Jin, José M. F. Moura, Ming Li, Jimmy Zhu

Author Affiliation(s)

ABSTRACT

Ultrasonic tomography usually uses iterative methods to reconstruct images, which suffer heavy computation burden. Inspired by the beamforming technique, this paper presents an end-to-end neural network framework for ultrasonic tomography, which employs a convolutional-deconvolutional structure for fast image inferring from received signals. Simulations validate the effectiveness of the proposed framework with high inference accuracy and acceptable image quality.

***Index Terms*—**

Ultrasonic tomography, Neural networks, Image reconstruction, Deep learning

1. INTRODUCTION

Ultrasonic tomography involves reconstructing the image by slices from ultrasound signals received by ultrasonic transducers after the signals probing through some medium. Conventional image reconstruction methods for ultrasonic tomography are iterative in nature, such as the algebraic reconstruction techniques (ART)[1], the simultaneous algebraic reconstruction technique (SART)[2], the simultaneous iterative reconstruction technique (SIRT)[3], the propagation and backpropagation method (PBP)[4]. They are computationally expensive, and the iterations are repeated in each imaging process. We propose an end-to-end neural network approach for ultrasonic tomography, which allows for fast inference of images from received signals given the trained network. We adopt a deep convolutional-deconvolutional network structure with skip layers to fully capture the characteristics of the underlying wave propagation model, in which the stacked convolutional network performs multi-scale weighting to encode the input, and the stacked deconvolutional network decodes these higher layer representations into images. We train the proposed conv-deconv network on large dataset that takes the cross spectral matrix of the received signals as input, and the ground truth images as output. We validate the effectiveness of the conv-deconv network by computing the error of predicted images on test input.

Thanks to XYZ agency for funding.

1.1. Related Work

Literature shows that artificial neural networks (ANNs) have been used in solving the inverse problem of ultrasonic tomography for target detection[5, 6]. These approaches usually assume a simplified mathematical model for wave propagation, have low image resolution and quality, and use simple multilayer perception structure. More common applications are using neural networks for classifying and segmenting ultrasonic images[7, 8, 9, 10], in which deep learning structures, such as deep belief network[11], are employed. These approaches do not generalize a wave propagation model, and are applications specific. Other applications involves using neural networks for ultrasound image enhancement and compression[12]. We present an end-to-end, deep conv-deconv structured network that generalizes the wave propagation model, provide fast inference of high quality images over the input signals.

1.2. Contributions

The contributions of this work are two-folded. Firstly, it provides an end-to-end neural network framework for fast inference of high quality images from ultrasonic signals with high accuracy. Secondly, it promotes a deep conv-deconv network structure that is able to fully generalize the characteristics of the underlying wave propagation model.

2. APPROACH

In this section, we derive the mathematical formulation of the generalized ultrasonic wave propagation and image reconstruction process, and present the conv-deconv network for solving the problem.

2.1. Problem Formulation

Consider using a linear array of N ultrasonic transducers to transmit ultrasonic pulses to the medium under inspection and receive the responses. We assume the imaging region in the medium is formed by $\sqrt{L} \times \sqrt{L}$ rectangular grids, where L is the total number of pixels in the image. The received signal

$S_n(t)$ at time t at the $n^{\mathbf{th}}$ transducer is represented by

$$S_n(t) = \sum_{i}^{M} q_i(t) g_n(x_i) + b(t), \tag{1}$$

where q_i is one of the M sources, $g_n(x_i)$ is the Green's function specified at source location x_i and for the receiving channel n, and b is the additive noise. We consider the signals' frequency domain representations, and let $S_n(f)$ denote the Fourier transform of $S_n(t)$. In order to retrieve the source signal q_i at the location x_i, we correlate S_n by g_n at an arbitrary location x for each receivers, which yields[13]

$$U_x(f) = \alpha \sum_{n}^{N} g_n(x) S_n(f), \tag{2}$$

where α is a multiplicative coefficient, and the value for $U_x(f)$ peaks when x coincides with the source location x_i. This follows the power of U_x

$$A_x = |U_x|^2 = \mathbf{g}(x)^* \langle \mathbf{S}, \mathbf{S}^* \rangle \mathbf{g}(x), \tag{3}$$

where $\langle \cdot \rangle$ is the average operator, and $\mathbf{g}(x)^*$ denotes the complex conjugate of $\mathbf{g}(x)$. We further collect the power A_x for all possible L locations in the image into a $\sqrt{L} \times \sqrt{L}$ matrix

$$\mathbf{A} = \mathbf{W}^* \mathbf{C} \mathbf{W}, \tag{4}$$

where $\mathbf{W} = \alpha \mathbf{g}$ is the unknown $N \times \sqrt{L}$ weight matrix, and $\mathbf{C} = \langle \mathbf{S}, \mathbf{S}^* \rangle$ is the $N \times N$ cross spectral matrix. We generalize (4) into

$$\mathbf{A} = \mathcal{R}_{\mathbf{w}}(\mathbf{C}), \tag{5}$$

where the nonlinear operator $\mathcal{R}_{\mathbf{w}} : H \rightarrow H_j$ maps from Hilbert space H to H_j. Our objective is to find the optimal $\mathbf{W}$ that

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} |\mathcal{R}_{\mathbf{w}}(\mathbf{C}) - \mathbf{y}|^2, \tag{6}$$

where $\mathbf{y}$ is the ground truth of $\mathbf{A}$. We propose a neural network approach to solve (6), as shown in next subsection.

2.2. Deep Convolution-Deconvolution Network

We propose a convolutional-deconvolutional network structure that takes the cross spectral matrix (CSM) $\mathbf{C}$ of the received signals as input, and outputs image $\mathbf{A}$. The structure of the network is illustrated in Fig. 1. The former part of the network is constructed by a $N \times N \times R$ input layer followed by a sequence of convolutional layers, where N and R denote the number of channels in the receiver array and the number of angles in the scan. The configuration of these convolutional layers is shown in Tab. 1. We use the rectified linear unit (ReLU) as our activation function[14]. We note that in replacement of using max-pooling layer as the down-sampling layer, which is adopted by many state-of-art convolutional architectures[15, 16, 17, 18], we use convolutional

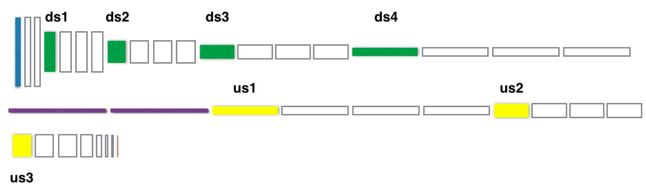

Fig. 1. Illustration of the conv-deconv network structure. The blue and red rectangles represent input and output. The green boxes represent down-sampling layers, and the yellow ones represent up-sampling layers. The rest are convolutional layers.

Table 1. Configuration of convolutional layers

layer no.	1	2	3	4	5	6	7	8	9	10
filter size F	5		4	3			4	3		
no. of filters D	64		128				256			
padding size P	2		1							
stride S	1		2	1			2	1		
activation δ	ReLU									

layer no.	11	12	13	14	15
filter size F	4	3			4
no. of filters D	512				1024
padding size P	1				
stride S	2	1			2
activation δ	ReLU				

layers with stride $S = 2$ to perform down-sampling. This allows the network to learn the down-sampling parameters by itself and leads to comparable results as using max-pooling layers[19].

2.2.1. *Deconvolution*

The latter part of the conv-deconv network is constructud by a sequence of deconvolutional networks. Contrary to the convolutional layer that takes input of size $I \times I$ and yields output of size $O \times O$, where $O = (I - F + P)/2S + 1$, the deconvolution operation reverses what is done in the convolution via what is called *transposed convolution*[20] and outputs the shape of size $I \times I$ from input size $O \times O$[21], and hence the deconvolutional network up-samples the output of the convolutional network. The configuration of the deconvolution layers is shown in Fig. 2. We note that this structure closely

Table 2. Configuration of deconvolutional layers

layer no.	16	17	18	19	20	21	22	23
filter size F	3							
no. of filters D	1024	512				256		
padding size P	1	3	1			5	1	
stride S	1							
activation δ	ReLU							
layer no.	24	25	26	27	28	29	30	31
filter size F	3							
no. of filters D	256	128			64	32	16	4
padding size P	1	9	1					
stride S	1							
activation δ	ReLU							

resembles the stacked autoencoders[22, 23], with a primary difference that autoencoders have their output the same with their input, and learn the compact representation of the input in the bottleneck layers (e.g. the layer 15 and 16 in Fig. 1), while our proposed structure takes in the CSM and outputs an image. In another sense, we can treat the convolutional layers as stacked encoders that encode the input to its abstract representation, and the deconvolutional layer as stacked decoders that decode the abstract representation to image.

2.2.2. *Skip-Layer Structure*

We also employ the skip-layer structure to directly combine a deep, coarse layer with the information from a shallow, fine layer to produce accurate and detailed image reconstruction[17]. Specifically, we directly sum the output of the down-sampling layer (layer *ds4* in Fig. 1) with the output of the up-sampling layer (layer *us1*), and feed the summation layer to the convolutional layer after layer *us1*. We then sum the output of layer *ds3* and the output of layer *us2*, and feed the summation layer to the layer after layer *us2*.

3. SIMULATIONS AND RESULTS

In the simulations, we use the specialized ultrasound simulation software package Field II[], and perform the network training and inference on Keras[24] framework with Theano[25] backend.

3.1. Simulation Setting

In the simulation, we collect $T = 7000$ images, and transform each of them to binary image by forcing pixel intensities greater than the mean intensity plus two times the standard deviation of the intensities to one, and the rest pixel intensities to zero. We use this collection of binary images $\mathbf{B} = \bigcup_{t=1}^{T} \{\mathbf{x}_t : x_i = 1 \,\text{or}\, 0, i = 1, 2, \ldots, L\}$ as ground truth images. A sample of the images is shown in Fig. **??**. We use an array of 192 elements with the kerf[] $0.0025\,\text{mm}$. To generate the signal responses, we transmit a sine wave with the the center frequency $f_c = 5\,\text{MHz}$ and the sampling frequency $f_s = 100\,\text{MHz}$ to the medium $\mathbf{B}$. For each $\mathbf{x}_t \subset \mathbf{B}$, we impose a set of scatters with uniformly distributed positions and normally distributed amplitudes with zero mean and variance scaled by the variance of $\mathbf{x}_t$. We simulate the behavior of phased array by repeatedly steer the transmitted signal to the image plane from -48 degrees to 48 degrees with 1.45 degrees increment, total 66 lines, in each transmission only 64 elements being active. This generates the signal responses $\mathbf{S}$ of dimension $64 \times K \times 66$ for each $\mathbf{x}_t$, with K being the length of the signal responses. An illustration of the received signals for 64 channels at steering angle ?? is shown in Fig. **??**.

3.2. Training Network and Inference

The input for training the network is the CSM $\mathbf{C}$ computed from the generated signal responses $\mathbf{S}$ over a total 7000 samples, which is a 4D tensor of dimension $7000 \times 66 \times 64 \times 64$. The output for training the network is the corresponding image collection $\mathbf{B}$. In the training, we randomly shuffle the training set with 10 percent of the training set as validation set. The specific training configuration is shown in Tab. 3. We use the mean square error (MSE) to measure the error between the predicted images and the ground truth images. The plot of training error ϵ, cross validation error r with regard to the number of iterations is illustrated in Fig. **??**. It shows that ... Using the trained network, we now predict the images

Table 3. Training configuration

batch size B_s	100
no. epoch E_p	5000
learning rate γ	0.01
learning rate decay η	0.0005
momentum m	0.9
loss L_e	MSE

over a test set containing 1000 CSMs computed from 1000 additional signal responses generated from the images not in $\mathbf{B}$. Fig. **??** shows the plot of error with regard to the number of iterations. It shows that ... Tab. **??** shows the metrics of accuracy.

4. CONCLUSIONS

5. REFERENCES

[1] Richard Gordon, "A tutorial on art (algebraic reconstruction techniques)," *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 78–93, 1974.

[2] Anders H Andersen and Avinash C Kak, "Simultaneous algebraic reconstruction technique (sart): a superior implementation of the art algorithm," *Ultrasonic imaging*, vol. 6, no. 1, pp. 81–94, 1984.

[3] Jeannot Trampert and Jean-Jacques Leveque, "Simultaneous iterative reconstruction technique: physical interpretation based on the generalized least squares solution," *J. geophys. Res*, vol. 95, no. 12, pp. 553–9, 1990.

[4] Chengdong Dong and Yuanwei Jin, "Mimo nonlinear ultrasonic tomography by propagation and backpropagation method," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1056–1069, 2013.

[5] Denis M Anthony, Evor L Hines, David A Hutchins, and J Toby Mottram, "Ultrasound tomography imaging of defects using neural networks," *Neural Computation*, vol. 4, no. 5, pp. 758–771, 1992.

[6] AC Pardoe, David A. Hutchins, J. Toby Mottram, and Evor L. Hines, "Neural networks applied to ultrasonic tomographic image reconstruction," *Neural Computing & Applications*, vol. 5, no. 2, pp. 106–123, 1997.

[7] Zümray Dokur and Tamer Ölmez, "Segmentation of ultrasound images by using a hybrid neural network," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1825–1836, 2002.

[8] H Sujana, S Swarnamani, and S Suresh, "Application of artificial neural networks for the classification of liver lesions by image texture parameters," *Ultrasound in medicine & biology*, vol. 22, no. 9, pp. 1177–1181, 1996.

[9] James S Prater and William D Richard, "Segmenting ultrasound images of the prostate using neural networks," *Ultrasonic Imaging*, vol. 14, no. 2, pp. 159–185, 1992.

[10] Ernest J Feleppa, Ronald D Ennis, Peter B Schiff, Cheng-Shie Wuu, Andrew Kalisz, Jeffery Ketterling, Stella Urban, Tian Liu, William R Fair, Christopher R Porter, et al., "Ultrasonic spectrum-analysis and neural-network classification as a basis for ultrasonic imaging to target brachytherapy of prostate cancer," *Brachytherapy*, vol. 1, no. 1, pp. 48–53, 2002.

[11] Gustavo Carneiro, Jacinto C Nascimento, and António Freitas, "The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods," *IEEE*

Transactions on Image Processing, vol. 21, no. 3, pp. 968–982, 2012.

[12] AS Miller, BH Blott, et al., "Review of neural network applications in medical imaging and signal processing," *Medical and Biological Engineering and Computing*, vol. 30, no. 5, pp. 449–464, 1992.

[13] Robert P Dougherty, "What is beamforming?," in *Berlin Beamforming Conference (BeBeC), Germany Berlin*, 2008.

[14] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[19] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[20] Vincent Dumoulin and Francesco Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[21] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[22] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

[23] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[24] François Chollet, "keras," *GitHub repository*, 2015.

[25] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.