Finding Ultrasound Sub-apertures for Liver Vessel Segmentation

2019 Spring BME590 Project

Felix Q. Jin*
Department of Biomedical Engineering
Duke University
Durham, NC 27708
fqj@duke.edu

Abstract

In conventional B-mode ultrasound imaging, the signals received by each transducer element are coherently summed after proper time delays are applied. Images produced using a small fraction of the receive channels have reduced image quality, but structures of interest are still often visible. In this work, we use deep learning to search for combinations of receive channels, termed a sub-aperture, that maintain enough imaging information to perform a certain task. In particular, we jointly trained a deep convolutional neural network (CNN) to select an aperture configuration and accurately segment hepatic blood vessels in ultrasound images of the liver. Using an annealing factor, we were able to force the network to select a limited number of channels. In some cases, the network selected a sub-aperture that produced better imaging metrics than a naive equally-spaced configuration. Here our results did not outperform manually designed sub-apertures for spatial compounding, most likely due to the larger search space and inefficient annealing methods. The learned solution were not stable, but basic principles such as lateral symmetry were reproducible. Our results demonstrate that ultrasound aperture weights can be incorporated into a neural network model and trained simultaneously with a task such as segmentation.

1 Introduction

Ultrasound transducers are composed of a series of small transducer elements, usually laid out in a line. In conventional B-mode imaging, these elements produce a focused sound wave and each listen for returning echos, which is coherent and contains phase information. The simplest beamforming method is delay and sum, where received signals are coherently summed after compensating for geometric focal delays [1]. Better beamforming methods such as spatial compounding involve incoherently summing images produced by the coherent sum of different sub-apertures [2].

Due to the spatial coherence of received echos, there is built-in redundancy in ultrasound. Images may be formed by summing only a small fraction of channels. These images will have reduced lateral resolution and increased electronic noise. However, making images from a limited number of channels offers faster frame rate and less hardware requirements, with potential applications to cardiac and fetal ultrasound [3].

In recent years, deep convolutional neural networks (CNN) have been applied to a variety of tasks including classification [4] and segmentation [5]. Physical imaging parameters can be incorporated into a neural network model and simultaneously optimized along with the usual weight parameters [6].

*MD/PhD Candidate, Medical Scientist Training Program, Duke School of Medicine

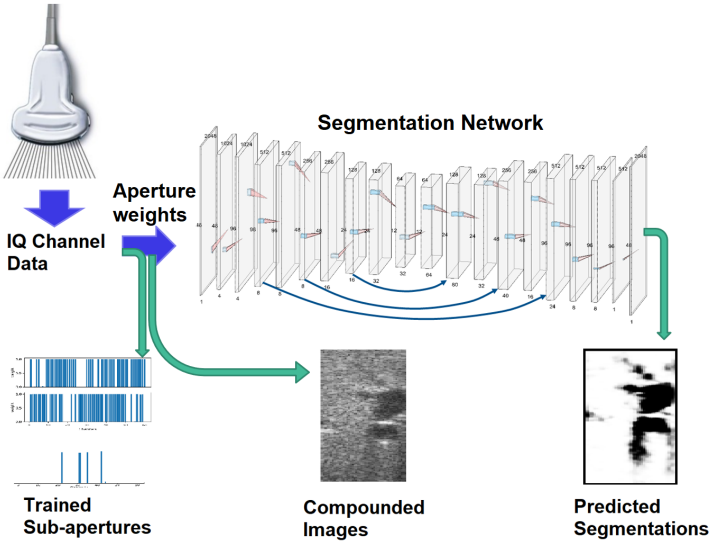

Figure 1: Summary of the aperture learning framework

In this work, we design and train a deep neural network to segment hepatic vasculature in liver ultrasound images. Our end-to-end deep neural network takes complex (IQ) channel data as input and outputs a predicted segmentation map. Each receive channel is associated with a trainable weight. We trained the networks to use a limited number of elements and to use spatial compounding. In some cases, the learned sub-aperture was superior to a simply-designed sub-aperture. The trained configurations were consistent with basic principles of ultrasound imaging. The entire framework is summarized in Figure 1.

2 Related Work

Following deep learning's recent successes in a variety of computer vision tasks, significant interest has been generated in applying deep learning methods to medical imaging. In the ultrasound field, deep learning has been applied both to image processing and image formation. Neural networks have been used to segment the left ventricle in cardiac ultrasound [7] and to classify of breast tumors [8].

However, image formation from the recorded channel data is equally important. The authors of [9] designed a proof-of-concept CNN that directly converts RF channel data to a segmented image of a lesion. They were able to train an end-to-end network that bypasses the typical delay, summing, enveloping detection, and segmentation steps. Because the authors input RF channel data directly into their convolutional network, the learned weights cannot be interpreted as selecting any particular subset of channels. Additionally, this work was limited to simulated and phantom cyst images.

Another group [10] investigated whether a deep neural network could reconstruct images from scanline subsampled data. They increase ultrasound frame rate by only acquiring every other scan line and using a CNN to fill in the missing information. Their network produced similar images from full and subsampled data, but both had poorer resolution compared to standard beamforming.

Physical imaging parameters can be optimized using deep learning by incorporating the physical imaging model into the network's computational graph. In [6], the authors trained a network that selected an ideal illumination arrangement for a specific optical microscope image classification task. The illumination parameters can be directly extracted from the learned weights of the network and then implemented into the physical microscope. The idea of learning physical parameters has applications for a vast number of imaging modalities, including ultrasound.

3 Methods

3.1 Data

Our dataset consisted of 72 *in vivo* acquisitions of the liver, originally collected for studying ultrasonic image quality metrics [11]. As described in that original paper, this dataset was acquired using a Vantage 256 Verasonics research scanner with a C5-2v curvilinear array transmitting at 2.4 MHz and receiving at 4.8 MHz, using pulse inversion harmonic imaging. The focal depth was set at 6 cm, with $f/2$ aperture diameter. These images of the liver and hepatic vasculature were acquired from 11 volunteers.

Each acquisition was stored as complex (IQ) channel data, consisting of 60 individual receive channels with the correct focal delays applied. The original images spanned 15.6 cm in range and 30 degrees in azimuth, but were cropped to the range of 2.2 - 11.0 cm for computational efficiency. The liver blood vessels were manually segmented in each image using the ImageJ software. These segmentation masks were used as ground truth for training. The dataset was randomly divided into 60 training, 10 validation, and 2 testing images. In addition, a single acquisition of a tissue mimicking phantom with cylindrical lesions was added to the testing set. Data augmentation consisted of random horizontal flips and shifts only.

3.2 Network

Our neural network for segmenting the liver vessels uses an encoder-decoder style architecture based on the popular UNet [5] for medical image segmentation. Major differences include a set of initial layers that resize the image, and the use of depthwise separable convolutions [12] in most blocks. Ultrasound imaging systems have remarkable axial resolution compared to lateral resolution, due to the carrier frequency of the acoustic radiation. We hypothesized that data processing by the symmetrically-shaped 3x3 convolutional kernels in the model will benefit from resizing the input image to an aspect ratio that is closer to the physical aspect ratio of the images. Therefore, the input image is first passed through a set of 3 convolutional layers with varying stride to produce an output scaled down 4x axially and scaled up 2x laterally.

The remaining network consists of 3 downsampling blocks and 4 blocks upsampling blocks. Each block consists of 2 residual-connected convolutional layers. Skip connections were used between appropriately matched down and up blocks. BatchNorm and ReLU6 activation were used after every layer except the output layer. The segmentation network architecture is detailed in the supplemental table 1. A binary cross-entropy loss function was used as the objective function between the network's output and the ground truth masks.

An aperture block was used to sum the 60 channel IQ data into a single channel image. Each channel had an associated trainable weight w_i, which forms a weight vector $\mathbf{w}$. The weights were transformed according to:

$$\mathbf{w}' \leftarrow \sigma(\alpha \mathbf{w}) \tag{1}$$

where σ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ and α is a scalar annealing factor. As α becomes larger, the transformed weights are pushed closer to 0 or 1. Thus, the annealing factor can gradually force a binary selection of channels. The final image was created by a weighted sum of the input channels according to the transformed weights, taking the amplitude of the resulting complex-valued image, applying a 0.2 power compression, and scaling to the range [0, 1].

To fix the number of channels selected, the sum of all transformed weights $\Sigma w_i'$ was constrained to equal a integer n using a squared-error loss function. The entire network's loss function is show in equation 2, where Y^* is the predicted segmentation mask and Y is the ground truth. λ was set to 0.5 in our experiments.

$$\mathcal{L} = \mathcal{L}_{BCE}(Y^*, Y) + \lambda \mathcal{L}_{SE}(\Sigma w_i', n) \tag{2}$$

3.3 Training

Our neural networks were implemented and trained using PyTorch on an NVIDIA Tesla V100 GPU. Stochastic gradient descent with Nesterov momentum 0.9 was used. When training only the segmentation network, we used a learning rate of 1.0, batch size of 10, and trained for 50 epochs. When jointly training the aperture weights and segmentation network, we used a learning rate of 0.01,

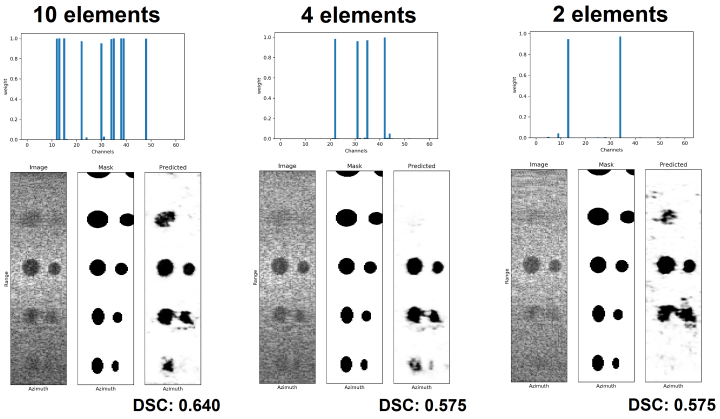

Figure 2: Aperture configurations and segmentation results for 10, 4, and 2 elements. For each case, the selected channels are shown above and shown below is a panel of the computed image, the ground truth mask, and the predicted segmentation. The DSC for each segmentation is stated below each case.

batch size of 60, and trained for 200 epochs. The annealing factor α was increased linearly from 1 to 6 for small element numbers (n<10) or from 1 to 21 for large element numbers (n>10). No other regularization methods were used. These hyperparameters were all tuned using the validation set.

3.4 Metrics

The primary metric for our segmentation results is the Dice Similarity Coefficient (DSC) defined as:

$$DSC(Y^*, Y) = \frac{2|Y^* \cap Y|}{|Y^*| \cup |Y|} \tag{3}$$

which is a number between 0 and 1 that measures how close two masks are to each other. In addition to this metric, we also calculated the common ultrasound metrics speckle signal-to-noise-ratio (SNR) and contrast. These two metrics are summarized in following equations:

$$\text{SNR} = \frac{\mu}{\sigma} \qquad \text{Contrast} = \frac{\mu_{out} - \mu_{in}}{\mu_{out}} \tag{4}$$

In conventional delay-and-sum ultrasound, the speckle SNR is typically close to 1.9 [2]. A higher SNR indicates more "smooth" appearing speckle with lower brightness variance.

4 Results

4.1 Limited element number

We trained networks to jointly perform segmentation and channel selection by fixing the number of allowed elements to either 10, 4, or 2 and then annealed the weight transformation function as described above. Show in figure 2 are the test results of theses networks on the phantom image. As element number is decreased, the visibility of lesions decreases, especially those away from the focus. The DSC of the predicted segmentation also decreased from 10 to 4 elements.

We compared the learned 4-element sub-aperture configuration to a simple manually-designed one consisting of 4 equally spaced elements. As seen in figure 3, the learned configuration produced a better image for the phantom data, with higher SNR, central lesion contrast, and segmentation DSC.

Our learned configurations were not stable over multiple trials. Each time a network is trained from scratch, it will converge to a slightly different sub-aperture design. However, we did find that sub-apertures would consistently bracket the center. For example, figure 4 shows three independent runs with the 2-element constraint. In each case, the network selected an element on the left and right sides.

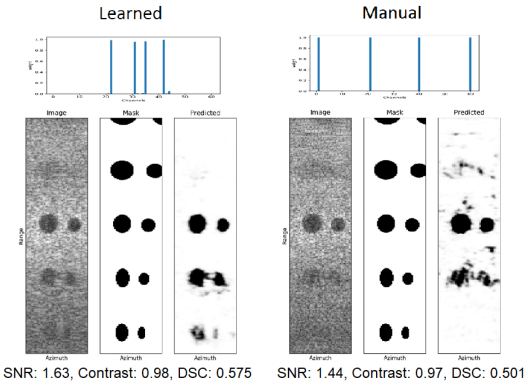

Figure 3: Learned vs manually designed sub-apertures with 4 elements each. The learned configuration produces slightly better images with higher SNR, contrast, and DSC.

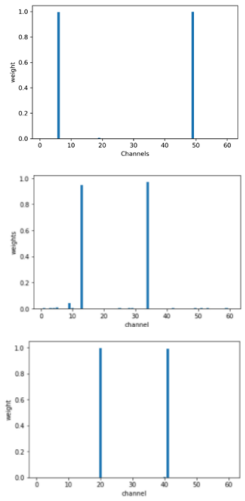

Figure 4: Three different sub-apertures learned for the 2-element case from independent runs

4.2 Spatial compounding

We also trained the network for two-aperture spatial compounding with 45 elements in each sub-aperture. We compared these results to a manually-designed configuration which consisted of the 45 leftmost and 45 rightmost elements. As shown in figure 5, the learned sub-apertures have no obvious symmetry. The imaging metrics and Dice coefficient results indicate that this learned configuration does not perform as well as the manually-designed one.

5 Discussion

In this work, we designed an end-to-end deep learning model for converting IQ channel data of liver blood vessels to vessel segmentation masks. Using channel weights, annealing, and an constraining loss, we forced the network to select certain aperture configurations with a limited element number. These configurations were jointly optimized with the segmentation network, a traditional encoder-decoder.

In the case of 4 elements, the learned sub-aperture performed better than the equally spaced design, as measured by SNR, contrast, and Dice similarity coefficient. Lateral resolution is increased by including more lateral elements because they probe higher frequencies in lateral k-space. However, the image data from lateral elements is less spatially coherent due to the distance between them. Thus

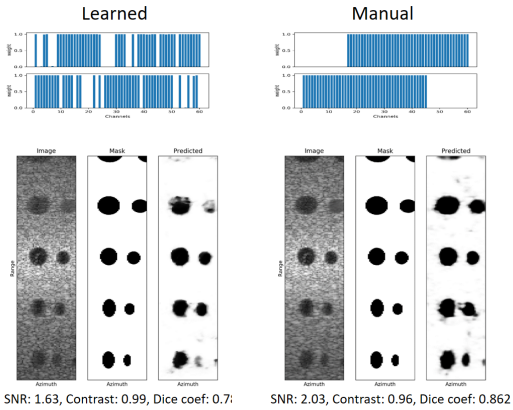

Figure 5: Two-aperture spatial compounding with 45 elements each: learned vs manually-designed. The learned configuration is difficult to interpret, and its imaging metrics were inferior to that of the manually-designed sub-apertures.

there is a trade off between wider and narrower spacing. The learned configuration took these factors into account and chose a narrower sub-aperture.

The stability of solutions is an issue with the current annealing method, as different training runs will produce different sub-aperture designs. The large redundancy between elements is part of the problem. Switching between neighboring elements does not generate a significant change in the image quality due to the high spatial coherence between neighboring elements. Another source of redundancy is the robustness of the segmentation network itself. The neural network is trained to accurately segment both easy and difficult cases. Thus, even if an aperture configuration produces slightly worse image quality, the network may compensate and output an equally accurate prediction. The optimization process is not directly sensitive to imaging metrics because the objective function is an accurate segmentation. Lastly, the sigmoid transformation function may not be the most effective way to anneal the channel weights. As the annealing factor increases, the weight values tend to have a negative gradient that fixed sum term cannot balance out. If a poor annealing strength or timing is used, the channel weights will tend towards zero and only a single random element is selected.

Despite the instability of solutions, basic trends were still observed. In particular, we found that the best 2-element apertures all selected an element from each side of the transducer. This conforms to our fundamental understanding about aperture physics, namely that an element's lateral position determines which lateral frequencies it can probe. When only 2-elements are available, it is advantageous to choose a pair that can probe both positive and negative lateral frequencies.

Our results with spatial compounding are harder to interpret. The increased degree of freedom in this experiment exacerbate solution instability. The current annealing algorithm likely cannot find the best solutions. However, we did notice that areas of relative sparsity were staggered between the two learned sub-apertures, which may reflect the general principle of summing uncorrelated information for speckle reduction.

We hope future research can improve upon the annealing methods presented in this work. Our neural network model can be extended to a variety of related tasks, such as heart segmentation or cirrhosis staging. With more difficult tasks, the ability of the loss function to discriminate sub-aperture designs increases. Ultrasound manufacturers currently use much more advanced beamforming methods than presented in this work. In the future, we plan to explore how deep learning methods can find the best configurations for these advanced imaging methods.

Acknowledgments

I would like to thank Will Long for providing me the data, and Dr. Gregg Trahey and Dr. Roarke Horstmeyer for their guidance.

References

[1] K. E. Thomenius, "Evolution of ultrasound beamformers," in *1996 IEEE Ultrasonics Symposium. Proceedings*, vol. 2, pp. 1615–1622, IEEE, 1996.

[2] M. E. Anderson and G. E. Trahey, "A seminar on k-space applied to medical ultrasound," 2000.

[3] O. Senouf, S. Vedula, G. Zurakhov, A. Bronstein, M. Zibulevsky, O. Michailovich, D. Adam, and D. Blondheim, "High frame-rate cardiac ultrasound imaging with deep learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 126–134, Springer, 2018.

[4] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[5] 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*, pp. 234–241, Springer, 2015.

[6] R. Horstmeyer, R. Y. Chen, B. Kappes, and B. Judkewitz, "Convolutional neural networks that teach microscopes how to image," *arXiv preprint arXiv:1709.07223*, 2017.

[7] G. Carneiro, J. C. Nascimento, and A. 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.

[8] Q. Zhang, Y. Xiao, W. Dai, J. Suo, C. Wang, J. Shi, and H. Zheng, "Deep learning based classification of breast tumors with shear-wave elastography," *Ultrasonics*, vol. 72, pp. 150–157, 2016.

[9] A. A. Nair, M. R. Gubbi, T. D. Tran, A. Reiter, and M. A. L. Bell, "A fully convolutional neural network for beamforming ultrasound images," in *2018 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, IEEE, 2018.

[10] W. Simson, M. Paschali, N. Navab, and G. Zahnd, "Deep learning beamforming for sub-sampled ultrasound data," in *2018 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, IEEE, 2018.

[11] W. Long, N. Bottenus, and G. E. Trahey, "Lag-one coherence as a metric for ultrasonic image quality," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 65, no. 10, pp. 1768–1780, 2018.

[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Table 1: Segmentation network architecture

Layer name	Description	Output Shape (C x H x W)
Input	Input image	1 x 2048 x 48
Ax_down_0	5x3 convolution with stride (2, 1)	4 x 1024 x 48
Lat_up	3x3 transposed convolution with stride (1, 2)	4 x 1024 x 96
Ax_down_1	5x3 convolution with stride (2,1)	8 x 512 x 96
Down_0	3x3 depthwise separable conv.	8 x 512 x 96
	3x3 depthwise separable conv. with stride 2	8 x 256 x 48
Down_1	3x3 depthwise separable conv.	16 x 256 x 48
	3x3 depthwise separable conv. with stride 2	16 x 128 x 24
Down_2	3x3 depthwise separable conv.	32 x 128 x 24
	3x3 depthwise separable conv. with stride 2	32 x 64 x 12
Up_0	3x3 depthwise separable conv.	64 x 64 x 12
	3x3 depthwise separable conv.	64 x 64 x 12
	2x bilinear interpolation	64 x 128 x 24
	Concatenate with output of Down_1	80 x 128 x 24
Up_1	3x3 depthwise separable conv.	32 x 128 x 24
	3x3 depthwise separable conv.	32 x 128 x 24
	2x bilinear interpolation	32 x 256 x 48
	Concatenate with output of Down_0	40 x 256 x 48
Up_2	3x3 depthwise separable conv.	16 x 256 x 48
	3x3 depthwise separable conv.	16 x 256 x 48
	2x bilinear interpolation	16 x 512 x 96
	Concatenate with output of Ax_down_1	24 x 512 x 96
Up_3	3x3 depthwise separable conv.	8 x 512 x 96
	3x3 depthwise separable conv.	8 x 512 x 96
	1x1 convolution	1 x 512 x 96
	(4x, 0.5x) bilinear interpolation	1 x 2048 x 48