Noise Suppression in Ultrasound Beamforming Using Convolutional Neural Networks

By

Zhanwen Chen

Thesis

Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE
in
Computer Science

December 14, 2019

Nashville, Tennessee

Approved:

Matthew Berger, Ph.D.

Maithilee Kunda, Ph.D.

To my parents, Cheng Yuehong and Chen Feng. Thank you for years of unconditional support and encouragement.

ACKNOWLEDGMENTS

I want to thank my PI, Dr. Brett Byram for giving me my first research opportunity through which I have grown as a researcher for the last year and a half. I want to acknowledge our post-doc, Dr. Adam Luchies whose thorough explanations of the data and methods made everything possible. I also want to thank my other colleagues in the BEAM lab, Chris Khan, Katie Ozgun, Dr. Jaime Tierney, Dr. Kazuyuki Dei, Siegfried Schlunk, Emelina Vienneau, and Abbie Weeks for tirelessly explaining the basics of ultrasound and a variety of related work.

I want to thank my thesis advisor, Dr. Matthew Berger for teasing apart problems, giving insight, and posing critical questions that inspire scientific thinking in a computer scientist. In addition, it would be difficult to conduct the various important experiments without the serious computing power he has generously shared.

I also want to thank my Computer Science PI, Dr. Maithilee Kunda, for advising me since the very beginning of my master's career. She has drilled in me important lessons on how to be a researcher.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

Introduction

1.1 Introduction to Ultrasound Beamforming

Diagnostic medical ultrasound has its roots in sonar and ultrasonic metal flaw detectors. It is a noninvasive, affordable, portable, and real-time method to characterize the cross-sectional view of soft tissues compared with other imaging modalities such as computed tomography (CT) and magnetic resonance imaging (MRI). The underlying principle of ultrasound is the measurement of time elapsed between sending a signal and receiving its echo; given the sound speed a priori, we can thus calculate the distance to an object based on this duration.

Ultrasound imaging consists of three steps: emitting sound waves (transmit), receiving echoes (receive), and interpreting those responses to form an image. The transmit step is achieved with ultrasonic transducers - devices that convert electricity into ultrasound waves or vice versa. These same transducers are utilized to receive ultrasonic echoes.

In practice, ultrasound scans are acquired with an array of transducer elements and each transducer element's pulse transmission is preciously timed by a computer. The most basic case of ultrasound imaging is plane wave imaging, where all elements in an array transducer emit the same acoustic pulses at the same time, forming a flat wavefront. After the transmit event, the waves propagate the field of view and are scattered back to the transducers as pulse-echo responses.

The distance to an object is then calculated as

$$depth = \frac{sound\,speed * time}{2}$$

accounting for both directions of travel. With this relationship, the time dimension can be translated to the distance or the depth dimension during the processing step. Suppose there are a total of N time samples and M_{active} active transducer elements. Thus, for each time sample $n \in N$, each transducer $m \in M$ receives a raw electric signal $y_m(n)$ measured in volts, resulting in an N by M data matrix. The acoustic power measured in decibels (dB) is the log-compressed and normalized envelope of the raw data after adjusting for time delay, given by

$$L_V = 20\, log_{10}(V)\, dB.$$

This produces an image of N by M resolution, where each pixel represents the relative intensity in the dynamic range of measured acoustic energy. A typical dynamic range is 60dB. This grayscale image modality is called brightness mode, or B-mode.

Ultrasound relies on scattered waves (echoes) which only occur at surfaces between varying acoustic impedance. Conversely, if the field of view contains a single, even medium such as air, no signal is returned to the transducers.

Now consider the nontrivial example of plane wave imaging of a phantom, which is an artificial composite of materials of various shapes and acoustic impedance. As is the case previously, all elements emit the same pulse at the same timesteps. However, as each pulse wave travels through the composite, it encounters varying impedance, and some of the wave energy gets scattered at various points in depth and at various degrees, depending on the location and the impedance of the component materials. The returning signals that result from this scattering are used to form a tomographic view of the phantom.

Compared with plane wave imaging, focused transmit provides better signal-to-noise ratios. The rationale for focused imaging is that responses from adjacent transducer elements are more relevant than those farther away. In practice, focus in ultrasound requires a subgroup of the total transducer array to form a single 1D depth-signal series, as opposed to one series for each element. A set of time-delayed (focused) transducer waves is called a beam. Focused imaging maximizes the signal, minimizes the noise, and results in a higher signal-to-noise ratio. Beamforming can also be viewed as a method for spatial filtering, it gives us spatial selection over where the energy returns from.

To achieve focusing, we first select a subset of transducers (called an aperture) and slide the selection by one element for num_beams times, where num_beams is the configurable number of beams in the overall array. This results in a new channel/aperture dimension to our data, in addition to the depth and lateral ones. We call this new type of data matrix channel data.

Within each subset, we need to send out a focused wave of a curved wavefront by taking advantage of wave interference. The superimposition of waves can cause constructive or destructive interferences, depending on their relative phases and amplitudes upon contact. We preset a focus (typically controlled by a knob on an ultrasound machine), from which we then use a Pythagorean-like calculation to compute the delays we need.

The processing of channel data in order to form an image is called beamforming. The most basic method of beamforming is delay and sum (DAS). We time-delay the data after receiving in order to adjust for the path length differences between the returning wavefront and the transducer elements. After applying delays, we finally operate on the channel dimension. The dimension of our post-delayed channel data is $[depth, elements\_per\_channel, num\_beams]$. To form each beam (vertical slice in the final image), we collapse its channel dimension by summing all 1D transducers responses in its aperture group, resulting in a new data matrix of size $[depth, num\_beams]$, called beamformed radio frequency (RF) data. The log compressed envelope (amplitude profile) of the normalized beamformed RF data is the resulting image.

1.2 Challenges in Ultrasound Beamforming

Although widely accepted, DAS beamforming is not an ideal method for clinical application due to the presence of many noises or artifacts, of which we present two: off-axis scattering and reverberation.

Underlying these two artifacts is the basic mechanism of ultrasound - scattering. Scattering describes the reflection of an acoustic wave as it encounters the boundary between matters of differing impedance. There are many scatters (boundaries) in the field of view. We assume that a wave scatters once before returning to the transducer array and that a transducer subarray emits a straight wave aimed linearly down. However, these assumptions are not always true. The unintended behaviors of scattering lead to artifacts such as off-axis scattering and reverberation clutter.

1.2.1 Off-Axis Scattering

The first such artifact is off-axis scattering. We typically assume that pulse-waves propagate downward, but in reality, pulse waves exhibit diffraction when emitted by a transducer beam. The diffraction resembles the way sound travels. Using an analogy, when person A shouts a secret message facing person B, a bystander C can often hear the message as well (if less clearly). In ultrasound, this phenomenon can be illustrated by beam plots - the normalized far-field magnitude of the transmit pressure versus observation angle. The acoustic pressure we want to focus on is the main lobe, and the side lobes are the diffractions that dilute the energy and cause off-axis echoes that degrade the image.

1.2.2 Reverberation

Another cause of image degradation is reverberation or multipath scattering. We assume that when a wave encounters a boundary, it is reflected back to and only to the transducers. However, in reality, the scattered wave can travel in all directions. In addition, the divergent scattered waves can be further scattered by boundaries outside the ROI of the emitting beam. As a result of bouncing around in the field of view, the returning signal gets registered to deeper depths because it takes longer to return to the transducer.

1.3 Noise Suppression Algorithms

Many algorithms have been developed to address ultrasound artifacts. Earlier methods include Tissue Harmonic Imaging (THI) [1, 2, 3, 4, 5] which suppresses noise but decreases the axial resolution, Time-Reversal Technique [6] which achieves noise suppression only in limited ultrasound applications, Minimum-Variance Beamforming (MV) [7, 8] whose contrast improvements in phantoms do not translate to in vivo images, Coherence Factor (CF) [9, 10] which improves the contrast ratio (CR) in images but at a cost of lower contrast-to-noise ratios (CNRs) and speckle signal-to-noise ratios (sSNR), and Short-Lag Spatial Coherence (SLSC) [11] which improves contrast but decreases lateral resolution and has a higher computational cost.

Machine learning methods have also been studied. They learn to separate signal from noise in received channel data rather than altering the transmit phase as in most classic approaches discussed above. One such method is Aperture Domain Model Image REconstruction (ADMIRE) developed by Byram and Dei [12]. ADMIRE has proven highly effective in suppressing noises and improving image contrast without loss in resolution. However, it is too computationally expensive, having a low frame rate that precludes real-time clinical imaging.

In addition, studies have shown that deep neural networks are effective in suppressing noise sources. Of particular relevance is the application of multilayer perceptrons (MLPs) in the aperture domain [13, 14]. Like ADMIRE, MLPs produce substantial CNR improvements without loss in image resolution while preserving the speckle pattern. In addition, its fast inference allows for real-time clinical usage.

Lastly, convolutional neural networks (CNNs) have also been used in biomedical imaging in general [15] and various ultrasound imaging modalities [16]. In terms of ultrasound beamforming, work has been done using CNNs to learn the entire beamforming process [17]. However, CNNs have not been used to suppress noise in the aperture domain. In this study, a convolutional neural network is trained to output the signal component of the input, which has three modes: accept, reject, and mixed. In the accept mode, the input should be preserved because it only has signals; in the reject mode, the input which consists entirely of noise should be zeroed; in the mixed mode, the signal component is preserved while the noise component is zeroed.

There are two inherent challenges to deep learning in ultrasound beamforming. One is in an ill-defined learning objective. Both classic and machine learning-based adaptive beamforming methods discussed above attempt to improve the contrast and the contrast-to-noise ratios while preserving the speckle pattern in in vivo images. However, training objectives in deep learning such as minimizing regression loss do not directly translate into the above-mentioned goals for adaptive beamforming.

Another challenge in the training-test domain mismatch. It is impractical to acquire in vivo training data due to the lack of ground truth. Doctors cannot know for sure the exact composition of a body part in each human subject without a biopsy or an invasive probe. Instead, we must rely on simulated or phantom scans whose compositions are known a priori. Even with different datasets, we still need to manually define the lesion and background regions required in the calculation of image metrics (discussed in the Methods section). These challenges are further discussed in the Discussions section.

1.4 Contribution

This thesis studies the effectiveness of convolutional neural networks on suppressing off-axis scattering in the frequency domain. I first show that models with both convolutional layers and fully-connected layers can approximate the performance of MLPs in suppressing noise and improving CNR. I also study the performance of fully-convolutional models (FCNs) and find that they do not perform as well as MLPs but outperform the DAS benchmark. By investigating the effect of kernel size and the number of convolutional layers on CNR, I find that convolutions do not solve the noise suppression problem better than MLPs; in fact, FCNs approximate MLPs by having a large receptive field to cover either the complex component or the entire input space. In terms of the real-time application, models with convolutional layers may be better suited than MLPs due to having fewer weights. For example, a typical FCN, LeNet, and MLP have 10^4, 10^5, and 10^6 weights respectively.

1.5 Organization

Chapter 2 discusses background on deep neural networks and related work on noise suppression with deep learning and CNN-based beamforming. Chapter 3 describes the training data generation process and signal grouping. Chapter 4 explains the CNN architectures and the training pipeline, including a random hyperparameter search technique. Chapter 5 details the beamforming pipeline for evaluation. Chapter 6 addresses the limitations of this work, discusses the results, and concludes the thesis with potential future work.

Chapter 2

Background

Many related methods address the problem of off-axis scattering including classic, machine learning-based, and deep learning approaches. Some classic approaches such as Tissue Harmonic Imaging and Time-Reversal Technique alter the transmit scheme. Others such as Short-Lag Spatial Coherence, Coherence Factor, and Minimum-Variance Beamforming focus on the postprocessing of the receive data; so do machine learning- and deep learning-based methods. In addition, we introduce convolutional neural networks (CNNs) and discuss related CNN-based applications even though they do not directly apply to ultrasound beamforming.

2.1 Classic Acoustic Clutter Suppression Algorithms

2.1.1 Tissue Harmonic Imaging

One widely-used approach to suppress cluttered reverberation signals is tissue harmonic imaging (THI). THI circumvents the inherent reverberation in the commonly used fundamental frequency (f_c) by adopting a higher frequency - the second harmonic frequency (f_{hc}). Because reverberation clutter primarily occurs at the fundamental frequency, reflected signals received at a second harmonic frequency are not subject to the same clutter [18, 19, 20]. As a result, harmonic B-mode images have better quality with higher contrast, improved resolution, and less near-field artifact. However, the tradeoffs of higher frequency are higher attenuation and lower amplitude which cause a loss in axial resolution [1, 2, 3, 4, 5]. Attenuation is the loss of power (amplitude) as a wave travels through depth. In soft tissue, higher frequency exacerbates attenuation. In addition, the narrowed bandwidth (fewer frequencies) reduces axial resolution [21], which is measures how close two scatters are along the depth dimension. Axial resolution is a function of pulse length as well as transducer frequency.

2.1.2 Time-Reversal Technique

Time-reversal is another method for suppressing reverberation clutter. In this method, ultrasound waves are transmitted and received twice. After the initial transmit and receive, the signals are reversed and re-transmitted into the field of view. The re-transmitted signals propagate back and refocus on the original source throughout the same medium, subject to the same reverberation. While clutter noises present differently for each transmit (incoherent), non-clutter signals follow the similar frequency patterns (coherent). This approach sums the original and the re-transmitted signals, amplifying the signal and reduces reverberation clutter thanks to the constructive and destructive interference of waves depending on coherence. The limitation for this approach is its requirement for a point-like source (e.g., a kidney stone) in the medium as the focal region is difficult to determine [22, 6].

2.1.3 Short-Lag Spatial Coherence

Short-lag spatial coherence (SLSC) is a beamforming technique that takes advantage of the spatial similarity among the response waves across the aperture [11]. Instead of summing across the channels as is the case in delay-and-sum (DAS) beamforming, SLSC measures - for each beam - the average correlation between all pairs of channels separated by l (”lag”) elements, for a given set of lags. The rationale behind this approach is that adjacent channel signals (short lags) are coherent (similar) spatially, but noises are incoherent. Coherence is a form of correlation or covariance between waves. Coherent waves have the same shape but are separated by a time delay. Weighted multiplication of waves would amplify the coherent components (the desired signals) and suppress the incoherent ones (the noises). As a result, the beamformed images show higher contrast, improved contrast-to-noise ratios, and better image texture [23]. The tradeoffs for these improvements include more computational complexity from additional matrix-based correlation derivation and loss in image resolution from only utilizing partial aperture information [24]. Moreover, the values in the image matrix are correlation measures instead of dB. Therefore, SLSC images are not directly comparable to B-mode images.

2.1.4 Coherence Factor

Coherence factor (CF) is a post-processing technique that computes a weight for each beam and each depth and applies these weights to the delay-and-sum (DAS) beamformed RF data [9, 10]. Mathematically, the CF is the ratio of the sum of coherent signals over all signals in each beam. Similar to SLSC, CF takes advantage of the high-coherence property of non-clutter signals to suppress cluttered signals. Compared with SLSC, this method improves image contrast while having a low computational complexity [22]. However, CF images tend to have a poor dynamic range because it suppresses signals in hypoechoic (mostly black) regions, making them look anechoic (completely black). It also destroys the speckle pattern and thus reduces the contrast-to-noise ratio (CNR) in images.

2.1.5 Minimum-Variance Beamforming

Minimum variance (MV) beamforming is an approach to suppress off-axis scattering. It does so by minimizing the power (the zero-mean variance) of off-axis regions while preserving the power of the target region (a point location) [7, 8]. MV has proven effective in improving contrast in phantom targets. The drawback of MV is its sensitivity to the dB variation from inside the focal region. In addition, the image quality improvements do not translate to in vivo images.

2.2 Machine Learning-Based Acoustic Clutter Suppression Algorithms

2.2.1 ADMIRE

Aperture Domain Model Image Reconstruction (ADMIRE) is a model-based approach to suppress both off-axis scattering and reverberation clutter. It operates on frequency-domain channel data, decomposes the cluttered signal, selects the scatterer in the region of interest (ROI), and reconstructs the decluttered signal. It then uses regression to determine the coefficient for regularizing each component signal. ADMIRE proves highly effective in suppressing both off-axis scattering and reverberation clutter. However, the computational complexity inherent in this approach precludes real-time applications until further optimization can increase the frame rate [22, 12].

2.2.2 Regularized Inverse

Another machine learning-based approach is regularized inverse or least squares (LS) beamforming. Given the steering angles - the directions of transducer elements in a beam to form a curved wavefront for focused imaging - a priori, this approach modeled each lateral scanline in the DAS beamformed RF data as a function of the scatterer's peak signal (desired), the given steering matrix, and a Gaussian error term. Stacking the per-depth least-squares solutions to the model, the LS approach produces images with improved contrast-to-noise ratios (CNR) [25].

2.3 Deep Learning-Based Acoustic Clutter Suppression Algorithm

2.3.1 Introduction to Neural Networks

Neural networks are machine learning models that learn by backpropagating loss. They are theoretically able to learn a broad set of complex functions by using non-linear activations [26]. Convolutional Neural Networks (CNNs) are special neural networks that take advantage of localized parameter sharing, requiring fewer parameters and thus having a reduced risk of overfitting. CNNs have seen widespread applications in Computer Vision, Natural Language Processing, and Medical Imaging alike.

LeNet is the first influential CNN architecture for classifying images. It consists of two convolutional layers followed by two fully-connected ones. Each convolutional layer is followed by a pooling layer for down-sampling. This small architecture was successfully used to recognize handwritten digits [27]. Another early architecture is AlexNet with 5 convolutional layers followed by fully-connected ones [28], which became the first neural-network winner for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012.

2.3.2 Random Hyperparameter Search and Neural Architecture Search

Training neural networks involves selecting training and model hyperparameters such as learning rate, dropout rate, and the width of fully-connected layers. Recent studies show that random search is more effective than grid search in finding the optimal neural network [29]. Furthermore, as there are no established models for ultrasound beamforming, a random search for hyperparameters in model arhitecture, such as the kernel dimensions, the number of kernels, and the padding/stride dimensions for a convolutional layer may be necessary.

A related technique for model selection is neural architecture search (NAS), which uses search heuristics such as reinforcement learning [30, 31] or evolution [32] to choose a model architecture that minimizes a separate validation loss. A common strategy is having a controller - itself a recurrent neural network model - select and evaluate types of layers in a child model (the optimal model). However, at the time of writing, popular automated machine learning (AutoML) frameworks such as Auto-Keras [33] and Neural Network Intelligence (NNI) do not support the automated search of convolutional neural networks for regression tasks. To address this problem, I implement a hybrid search that manually defines the types and order of layers, only to vary their sizes and the number of layers. This approach is discussed in detail in the Methods section.

2.3.3 CNN Architectures

Most well-known CNN architectures have fully-connected layers after convolutional ones. However, it is not always clear that full connections are necessary as they flatten the spatial features detected by the convolutional layers. There are two notable architectures that avoid fully-connected layers: the Fully Convolutional Network (FCN) [34] and the U-Net [35]. Both were proposed to solve the problem of semantic segmentation or pixel-wise image classification, while U-Net focuses on biomedical images. Although they differ in architecture, both feature bottlenecks/encoder-decoders and upsampling layers in order to bring the shrunk convolutional outputs back to the size of the input.

The bottleneck layer is an intermediate layer that reduces the size of data coming from its previous layer. Bottleneck layers are used to obtain a (nonlinear) representation of the input with reduced dimensionality, i.e., performing dimension reduction. An example bottleneck layer is an autoencoder, which is used to reduce the previous output and is in turn used to generate a larger encoding that approximates the original input (hence ”auto”) [36]. For example, the GoogLeNet architecture that won the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC2014) features 1 by 1 convolutional blocks (termed ”network-in-networks” or NiNs) that reduce the number of features before the computationally expensive parallel blocks. [37]. This bottleneck-upsampling approach is a promising network design element that could also be applied to regression tasks because it enables output sizes to match the input.

2.3.4 Convolutional Neural Networks for General Regression

One example CNN solution to a regression task is image orientation prediction. Fischer et al. proposed an approach to train a modified AlexNet to output the orientation of an image. The input is an image while the output is either a single value in degrees or two values - the sine y and cosine x of an input image. They use significant amount of data augmentation where they generate new training examples by rotating existing ones. The results suggest that predicting a single rotation value is difficult. They found that a single CNN performed poorly compared with a hybrid classification-regression method that performed better than non-neural network approaches, albeit with a nontrivial mean absolute error of $21°$ for rotations in the range of $[-180°, 180°]$ [38]. Although the problem addressed in this study is different from mine in terms of the output resolution, both tasks have an implicit classification component. For their hybrid approach, a rough orthogonal rotation is classified at first before a finer rotation is finally predicted. For the aperture-domain denoising problem that I consider, a neural network is trained to recognize three signal modes: accept, reject, and mixed. It must then extract the signal component. My problem definition is discussed in the Methods section.

2.3.5 Multi-Layer Perceptrons for Suppressing Off-Axis Scattering

Recently, Luchies and Byram proposed a neural-network approach to suppress off-axis scattering [13, 14]. They trained multi-layer perceptrons (MLPs) that operated in the short-time Fourier transform (STFT) or the frequency domain to suppress the off-axis signals based on simulated point targets. These beamformers are convolutional in nature insofar as the networks, including their weights, are reused through depth; however, fully-connected layers are used to span the aperture dimension. This method proved effective in improving contrast while preserving speckle patterns. This work motivates further exploration of CNNs on the same STFT-domain data, as there may be spatial features in the frequency domain that could be more effectively learned by CNNs, such as aperture shapes. In addition, training in the STFT domain helps avoid having to train for experimental parameters such as different pulse shapes and depth-dependent attenuation.

2.3.6 Convolutional Neural Networks for Beamforming

There are two notable related CNN-based approaches for reducing off-axis scattering, learning ultrasound reconstruction, and speckle reduction. For example, Yoon et al. proposed a method that effectively interpolates missing sub-sampled RF data in 3D ultrasound [39].

Hyun et al. showed that fully-convolutional neural networks (FCNs) have the potential to learn a speckle-reducing beamformer which also suppresses off-axis scattering. Their learning task is to beamform a B-mode image from raw RF data. Their networks used between 2 and 16 convolutional layers with same padding in order to preserve the input dimensionality throughout the network. Notably, they explored different loss functions including L1, L2, SSIM, and MS-SSIM losses. They were able to show SNR improvements in both phantom and in vivo targets. However, the images did not indicate clear CNR improvements for all targets [40].

Chapter 3

Methods

3.0.1 Frequency-Domain Learning

Even though it would be intuitive to simply denoise our raw channel data (in the time domain) along the aperture dimension because off-axis scattering and reverberation are defined across the channel, time-domain ultrasound signals are subject to the issue of depth-dependent attenuation, which describes the loss of acoustic energy as the signal travels through a medium. Depth-dependent attenuation is not only a function of time/distance but also of frequency. If we train on time-domain signals, it is possible that we will need to learn this function in addition to denoising. Our hypothesis is that because depth-dependent attenuation is a function of both time and frequency, it would be intractable to learn given only time-domain data. Therefore, we can circumvent this issue by using processing the signal amplitude across the channel per frequency. Because the 4th, 5th, and 6th frequencies are most affected, we can save our efforts by only training one model for each of these three frequencies.

In order to extract the frequency-domain data from time-domain channel data, we use the short-time Fourier transform (STFT) on the time-domain channel data to generate a time-frequency spectrogram that we call STFT-domain data. The original time dimension is consumed to produce the frequency dimension and the time segments dimension. As a result, the data dimensionality increases to 4D: the number of frequencies, the number of segments, the number of channels per beam, and the number of beams. The value of each data point changes from voltage to complex amplitude data consisting of the real in-phase (I) and the imaginary quadrature (Q) components.

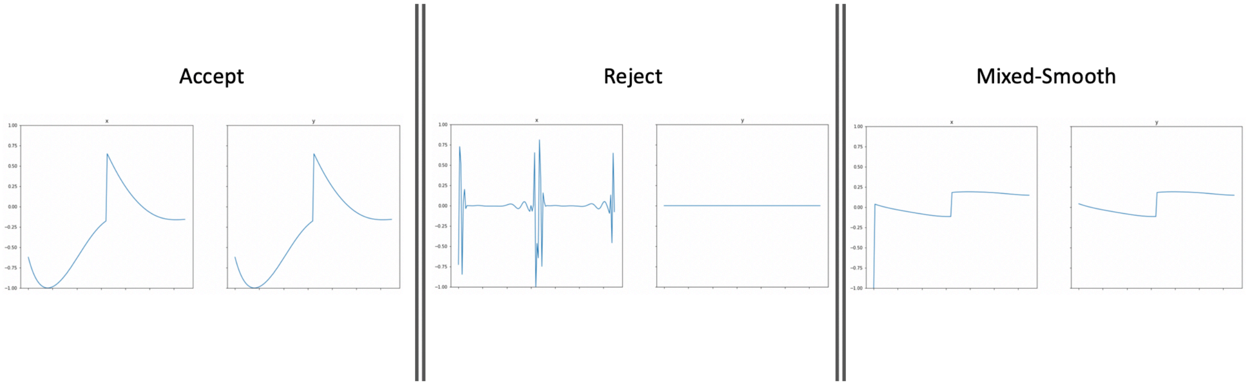

Figure 3.1: The simulated training data has three modes: accept, reject, and discriminate.

Additional benefits to training in the STFT domain include resilience to changes in pulse shape which only affects the time domain. Lastly, being able to use complex data instead of relying on the Hilbert transform is another benefit of training in the STFT domain.

The learning task is preserving the signal component of frequency-domain input data while discarding (zeroing out) the noise component. More specifically, there are three cases of input signals: accept, reject, and mixed. If a frequency-domain input signal falls inside the main lobe of a beam (accept case), the target signal to learn is the same as the input. In other words, the mapping from inputs to targets in the accept case is the identity function. In the reject case, the targets are set to 0s as they are all from the off-axis region. In the mixed case, however, there are signals from both inside and outside the main lobe. The targets, in this case, is the inside component of the mixed signal, which is accessible from the simulation. The learning task here is to discriminate between the signal and the noise and only preserve the signal component. Figure 3.1 shows these three modes.

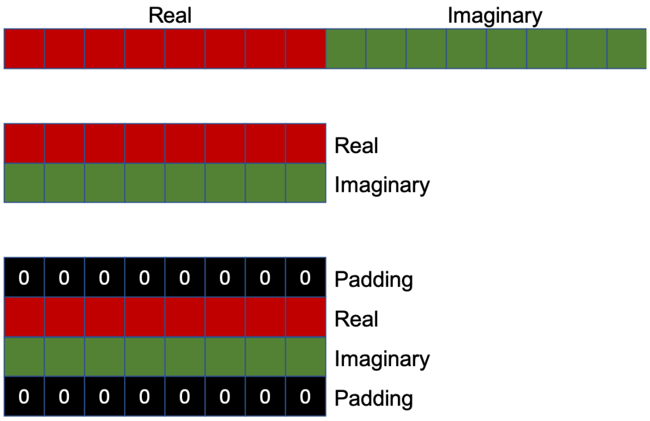

Figure 3.2: Top to bottom: 1D, single-channel convolution. 1D, two-channel convolution. 2D, 1-channel convolution.

3.1 Training Data Generation

The training dataset in this thesis was generated by Luchies and Byram for their MLP denoising studies [13]. Training data was generated from Field II [41, 42] simulated point target responses. The simulated ultrasonic array was based on the L7-4 (38 mm) linear array transducer. Point targets were randomly placed in an annular sector centered at the focal depth of the transducer array [14]. Both inputs and targets have been delayed and have gone through the STFT step. The number of elements per beam is 65. We used 10^5 examples for training and 10^4 for validation.

3.2 Grouping of Real and Imaginary Input Componenets

The MLP beamformers developed by Luchies and Byram concatenate the real and imaginary components of the STFT data. However, as spatial order matters for CNNs, it is not clear how they should be grouped as inputs to the neural networks. As a result, we studied three different groupings: concatenation, channel-stacking, and height-stacking. The concatenation is simply appending the imaginary component to the real component. This is the same as in Luchies and Byram and would be a 1D input of length 130. Channel-stacking would have two 1D signals of length 65 on each of the two input channels. The first two cases use 1D convolution. The third case, height-stacking, is to form a 2D image whose height is 2, width is 65, and on a single channel. This case can be understood as a narrow gray image. By zero-padding in the height dimension both above the real component and below the complex component, a convolution of a height of two thus convolves three times: once on only the reals, once on both the real and the complex, and once only on the complex. The rationale behind this case is that it forces the network to learn the interactions between the real and the complex. Figure 3.2 illustrates these three groupings.

3.3 CNN Architectures

3.3.1 LeNet-Like

Our first CNN architectures were similar to that of LeNet. It has two or five convolutional layers followed by fully-connected layers. In our study, each convolutional layer has a randomly selected set of kernel size, padding size, stride size, and the number of kernels. An additional constraint is that each convolutional layer should have more kernels than the previous one. The model optionally uses a pooling layer after each convolutional layer and the number of fully-connected layers at the end was randomly chosen between one and three. The activation function for each non-output layer was the rectified linear unit (ReLU). The optimizer was randomly chosen between Adam and stochastic gradient descent (SGD).

3.3.2 Fully-Convolutional Nets (FCNs)

In order to further study the role of convolutional layers in learned denoising functions independent of fully-connected layers, we implemented a fully-convolutional architecture that features four convolutional layers. Because the first two convolutional layers reduce the input resolution, we place two upsampling layers after the second and third convolutional layers in order to bring the output resolution back to the input resolution. We constrain the number of kernels for the second convolutional layer should be greater than that of the first and smaller than that of the third. The number of kernels for the last convolutional layer needs to equal the number of input channels (either one or two, depending on input data groupings).

After a broad model search, we conducted another study on the performance of models as a function of the kernel size of each convolutional layers. If performance increases with kernel size, it would imply that convolution is not effective because a convolutional layer approximates a fully-connected one as the convolution kernel size increases. Asymptotically, if the kernel size equals the input size, the convolutional layer becomes a fully-connected one because no stride is possible. To do this, we constrained the kernel size for all convolutional layers to be the same and varied only the singular kernel size and the padding for each layer (to match input and output resolutions).

We further investigated the effectiveness of the convolution operation by increasing the number of convolutional layers in FCNs. If model performance is a function of the number of layers, there would be two possible conclusions. One is that a deeper model is more effective due to an increase in learning capacity. The other possible explanation is that there may be few spatial features to learn from the data and that the convolutional model is approaching a fully-connected one. As the number of convolution layers increases, each cell in the last convolutional layer depends on more input cells, increasing the receptive field. An oversized receptive field is less conducive to learning local spatial structures because the mechanism of localized parameter sharing in CNNs is undermined by becoming more global (looking at the entire input) than local (looking at a small section of the input). If every output element depends on every input element, then the CNN is effectively approximating an MLP.

3.4 MLP with Bottlenecks (MLPB)

Our last study is to investigate why MLPs perform well as indicated by Luchies and Byram. We created an MLP similar to their best-performing one with 5 layers with 1040 hidden nodes in each layer. We varied the number of nodes in the 3rd fully-connected layer to investigate the effect of bottlenecking the neural network. If a narrower intermediate layer can achieve a similar level of performance, then it would imply that there is a more generalizable representation than the full MLP.

3.5 Implementation Details

The architecture and training hyperparameter search was implemented with logical constraint satisfaction with Prolog. A neural network layer is represented by a dictionary. Each type of layer has a unique formula for output resolution. For 1D convolutional layers, the output length is

$$L_{out} = (L_{in} - L_{kernel} + 2 * L_p adding) // L_{stride} + 1$$

. The output resolution of each layer becomes the input resolution of the next one. By recursion, the output resolution of the entire network is a function of the input resolution and the model hyperparameter set of each layer. If we constrain the output resolution to equal the input resolution and the output resolution of each layer to be greater or equal to 1, the program will solve for an entire architecture.

3.6 Evaluation Methods

3.6.1 The Evaluation Scan Datasets

An important distinction from a typical machine learning problem is that the evaluation dataset in our study differs from the total training dataset which includes both training and validation data. This divergence presents a challenge to the learning task and is discussed in Chapter 6. The same ATL L7-4 (38 mm) linear array transducer was operated using a Verasonics Vantage 128 system (Verasonics, Kirkland, WA) to scan a physical phantom. A cylindrical cyst having 5mm in diameter located at 7cm depth was scanned using a cross-sectional view at five different positions (”targets”) along the axial dimension. The phantom dataset is used for model selection.

Yet another evaluation dataset is the in vivo dataset. We used the same equipment and scanning parameters for acquiring phantom data to scan the liver of a 38-year-old healthy male to look for vessels in the liver in a study approved by the Vanderbilt University Institutional Review Board. We use beamformed images in this scan as an auxiliary validation as in vivo scans are difficult to compare due to practical limitations.

3.6.2 Beamforming

We first create a benchmark for each of our evaluation scan targets using delay-and-sum (DAS) beamforming. Delays were applied to the acquired channel data to adjust for true signal depths. Because DAS simply sums the time-domain signals across the channel, no STFT and inverse STFT are applied. After summation across the channel dimension, an envelope detection algorithm is applied to the beamformed RF data to extract the absolute amplitude. The log-10 compression of the envelope multiplied by negative 20 (negative in order to invert color) converts the units from volts (V) to decibels (dB). The resulting data matrix in dB is the final beamformed image whose grayscale mapping corresponds to the dynamic range of the data.

In contrast, the application of trained neural networks to denoise the data in the STFT domain requires an extra step after applying the delays and before the summation across the aperture compared with DAS. Because the models are trained with inputs and outputs both in the STFT domain, we apply STFT to the postdelayed channel data (time by channels per beam by beams) along the time axis. The resulting matrix has four dimensions: frequencies, time segments, channels per beam, and beams. As previously described, we only train and apply models on the 4th, 5th, and 6th frequencies along the channel dimension and leave data in the remaining frequencies as-is. After this in-place operation, we must convert our 4D matrix back to the time domain with an inverse STFT operation along the frequency and segment dimensions to return our data to channel data. The remaining channel-axis summation, envelope detection, and the multiplication of log compressed envelope are the same as in DAS.

3.6.3 Image Quality Metrics

Now that we have both the DAS and the neural-network beamformed images, we need quantitative metrics to compare each model with the DAS benchmark, other CNNs, and the MLPs by Luchies and Byram. The image quality metrics used for evaluation purposes include the contrast-to-noise ratio (CNR), the contrast ratio (CR), and the speckle SNR.

Because we want to increase the contrast of the region of interest (ROI) without significantly altering the background speckle, the primary metric we rely on is the contrast-to-noise ratio (CNR). For each target, we pre-define the region of interest (ROI) and the background speckle region. The CNR is a measure of the normalized difference between average signal intensity. The formal definition for CNR is

$$\mathrm{CNR} = 20\log_{10}\left(\frac{|\mu_{\mathrm{background}} - \mu_{\mathrm{lesion}}|}{\sqrt{\sigma^2_{\mathrm{background}} + \sigma^2_{\mathrm{lesion}}}}\right), \tag{4}$$

Another metric is the contrast ratio (CR), defined as

$$\mathrm{CR} = -20\log_{10}\left(\frac{\mu_{\mathrm{lesion}}}{\mu_{\mathrm{background}}}\right), \tag{3}$$

.

The last metric of interest is the speckle signal-to-noise ratio (SNR), expressed as

$$\mathrm{SNRs} = \frac{\mu_{\mathrm{background}}}{\sigma_{\mathrm{background}}}, \tag{5}$$

where μ is the mean and σ is the standard deviation of the uncompressed envelope.

Because each model has one image for each of the five phantom scan targets and each of the two in vivo targets, an average is also calculated for all metrics.

Chapter 4

Results

4.1 LeNet-Like CNNs

I trained and beamformed over 1000 LeNet-like models generated with our constraint-satisfaction random search. The best performing LeNet has an average CNR of 5.46dB with a standard deviation of 0.45dB. The architecture and hyperparameters of the best-performing model are shown in Table 4.1.

I found that LeNet-like CNNs featuring two convolutional layers and two or three fully-connected layers were effective in suppressing off-axis noise. I also found that adding Gaussian noise in training was useful. Furthermore, for LeNets, it was better to concatenate the I and Q components as a single-channel 1D array or stack them as a two-channel 1D array than to stack them as a one-channel 2D array. Furthermore, almost all top-performing models used Adam instead of SGD as their optimizers. In terms of loss functions, L1 was ineffective compared with MSE and Smooth L1.

For phantom targets, the CNR was 5.46±0.45 dB, 5.57±0.20 dB, and 4.24±0.38 dB for the best LeNet, the best MLP, and DAS, respectively. A qualitative assessment of the speckle pattern at the top of the phantom images suggests that LeNets introduce less interference farther away from the focus, suggesting a potential benefit of having a larger depth of field.

In addition, like the best MLP, the best LeNet is able to translate its phantom CNR improvements to in vivo scans.

A t-test was used to compare the phantom CNR value for all scan targets between the best LeNet and the best MLP. The difference was not statistically significant, with a p-value of 0.45. This finding suggests that LeNets produce equivalent results

Hyperparameter	Value
Architecture Type	LeNet-like
Input Formulation	1x130x1
Using Max Pooling	True
Using Batch Normalization	True
Adding Gaussian Noise	True
Conv1 Kernel Size	17
Conv1 Number of Kernels	45
Conv1 Stride	1
Conv1 Dropout	0
Pool1 Kernel Size	2
Pool1 Stride	2
Conv2 Kernel Size	12
Conv2 Number of Kernels	35
Conv2 Stride	1
Conv2 Dropout	0.5149
Pool2 Kernel Size	2
Pool2 Stride	2
Fully-Connected (FC) Layers	2
FC Layers Width	109
Batch Size	32
Activation Function	ReLU
Loss Function	Smooth Mean Absolute Error
Optimizer	Adam
Learning Rate	1.803e-4

Table 4.1: Architecture and hyperparameters for the best LeNet model

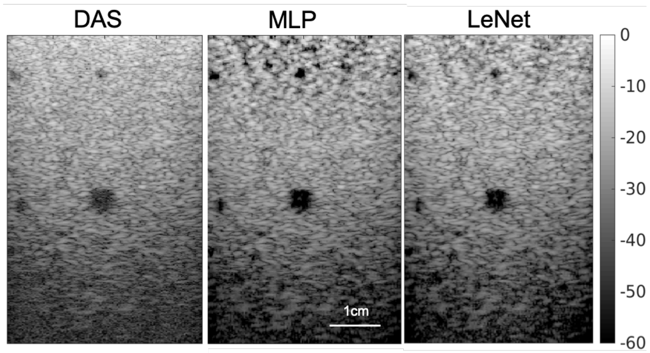

Figure 4.1: For a phantom target, DAS has a CNR of 4.40, MLP 5.54, LeNet 5.29

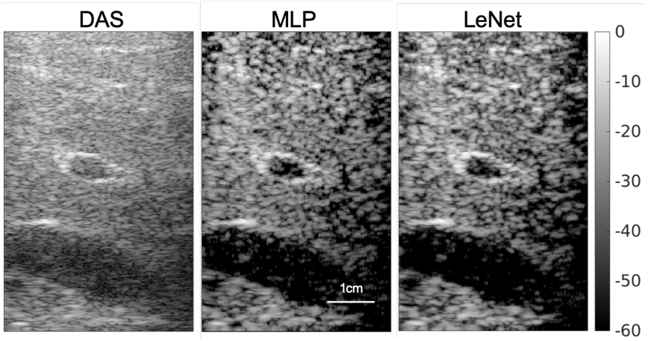

Figure 4.2: For an in vivo target, DAS has a CNR of -14.98, MLP -0.80, LeNet -2.84

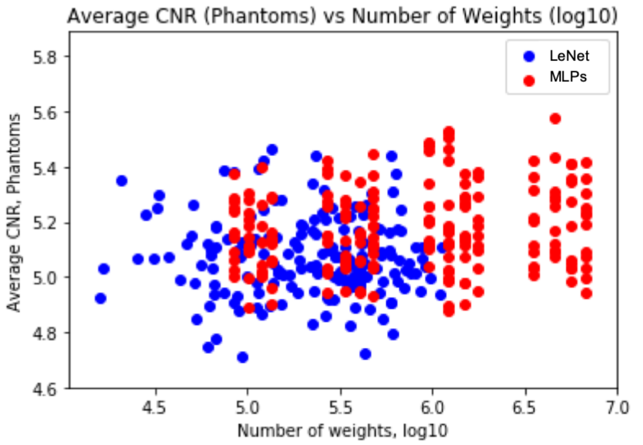

Figure 4.3: The best LeNets tend to have fewer weights than the best MLPs

to MLPs. A plot of CNR values as a function of the total number of model weights also shows that LeNets approximate the performance of MLPs with two magnitudes fewer weights. This could potentially mean that LeNets are easier to train and faster to deploy compared with MLPs.

4.2 FCNs

4.2.1 Unconstrained FCN-4

I also trained and beamformed 450 FCNs with the best-performing model having an average CNR of 4.93dB and a standard deviation of 0.20dB. The architecture and hyperparameters of the best-performing model are shown in Table 4.2.

Hyperparameter	Value
Architecture Type	FCN-4
Input Formulation	1x65x2
Using Batch Normalization	False
Adding Gaussian Noise	True
Conv1 Kernel Size	13
Conv1 Number of Kernels	36
Conv1 Padding	3
Conv1 Stride	2
Conv2 Kernel Size	4
Conv2 Number of Kernels	59
Conv2 Padding	2
Conv2 Stride	2
Conv3 Kernel Size	6
Conv3 Number of Kernels	19
Conv3 Padding	3
Conv3 Stride	1
Conv4 Kernel Size	6
Conv4 Number of Kernels	2
Conv4 Padding	2
Conv4 Stride	1
Batch Size	32
Activation Function	LeakyReLU
Loss Function	Smooth Mean Absolute Error
Optimizer	Adam
Learning Rate	1.0e-5

Table 4.2: Architecture and hyperparameters for the best FCN model

For phantom targets, the CNR was 4.93±0.20 dB for the best FCN model, offering a modest improvement over DAS (4.24±0.38 dB). Visually, the FCN is unable to improve the contrast of the structures at the top of the phantom.

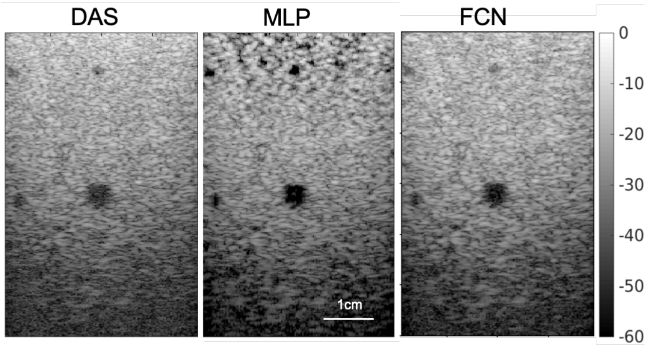

Figure 4.4: For a phantom target, DAS has a CNR of 4.39, MLP 5.54, FCN 4.87

For in vivo targets, the FCN beamformed image has a modest contrast improvement over the DAS image. A qualitative assessment indicates that the smaller vascular features are not as clear in the FCN image as in the MLP beamformed image.

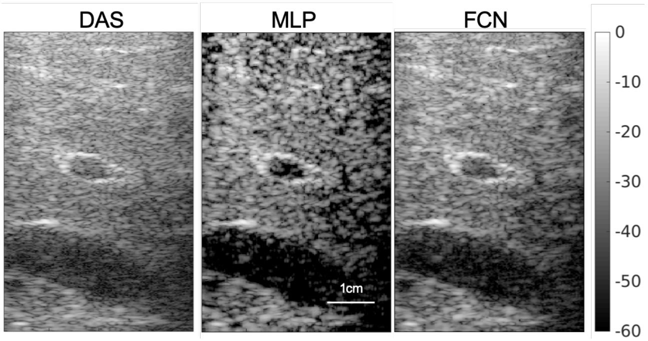

Figure 4.5: For an in vivo target, DAS has a CNR of -14.98, MLP -0.80, FCN -5.92

The full range of models performance for FCNs with outliers, as a function of the log number of weights, is shown in Figure 4.6. The same distribution without outliers is shown in Figure 4.7.

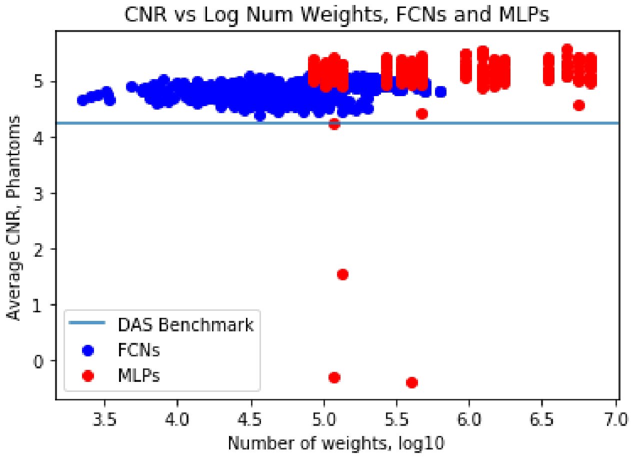

Figure 4.6: Phantom average CNR as a function of the log number of weights (with outliers), FCNs and MLPs

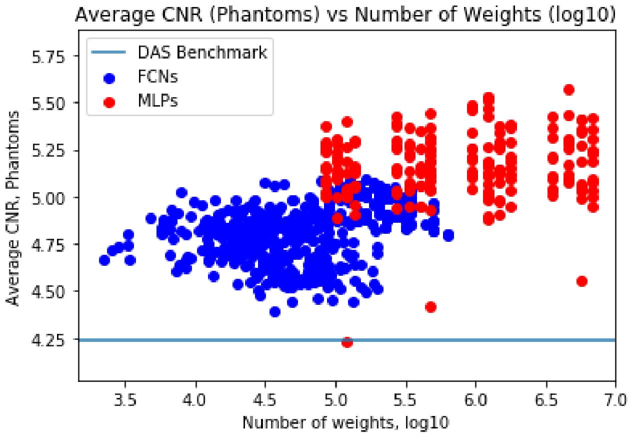

Figure 4.7: Phantom average CNR as a function of the log number of weights (without outliers), FCNs and MLPs (excluding outliers)

Hyperparameter	Value
Architecture Type	FCN-4
Input Formulation	1x130x1
Using Batch Normalization	True
Adding Gaussian Noise	True
Conv1 Kernel Size	3 to 65
Conv1 Number of Kernels	18
Conv1 Padding	0 to kernel size divided by 2
Conv1 Stride	2
Conv2 Kernel Size	Same as Conv1
Conv2 Number of Kernels	91
Conv2 Padding	0 to kernel size divided by 2
Conv2 Stride	2
Conv3 Kernel Size	Same as Conv1
Conv3 Number of Kernels	88
Conv3 Padding	0 to kernel size divided by 2
Conv3 Stride	1
Conv4 Kernel Size	Same as Conv1
Conv4 Number of Kernels	1
Conv4 Padding	0 to kernel size divided by 2
Conv4 Stride	1
Batch Size	32
Activation Function	LeakyReLU
Loss Function	Smooth Mean Absolute Error
Optimizer	Adam
Learning Rate	1.0e-5

Table 4.3: Architecture and hyperparameters search space for the constrained FCN-4 to study the effect of varying kernel size on CNR

4.2.2 Constrained FCN-4, Varying Kernel Size

For the study on CNR as a function of kernel size, I use the network configuration shown in Table 4.3.

4.2.3 Constrained FCN-N, Varying Number of Layers

For the study on CNR as a function of the number of layers, I use the network configuration shown in Table 4.4.

Hyperparameter	Value
Architecture Type	FCN-N
Input Formulation	1x130x1
Using Batch Normalization	True
Adding Gaussian Noise	True
Conv Kernel Size	5 or 7
Conv Number of Kernels	30 or 50
Conv Padding	kernel size divided by 2
Conv Stride	1
Number of Conv Layers	5, 10, 15, or 20
Batch Size	32
Activation Function	LeakyReLU
Loss Function	Smooth Mean Absolute Error
Optimizer	Adam
Learning Rate	1.0e-5

Table 4.4: Architecture and hyperparameters search space for the constrained FCN-N to study the effect of varying the number of layers on CNR

4.3 Analyzing Convolution

4.3.1 The Receptive Field of Convolution

The convolution operation is inherently recursive. For an output element of a convolutional layer, not only does it depend on the input to its own layer, but it is also an indirect product of any previous convolutional layer. The theoretical receptive field (RF) describes the number of input elements that ultimately map to the final output. Conceptually, if the size of the RF approximates the size of the input, the network can be thought of as approximating an MLP because every output element can depend on each input element in both networks. In other words, controlling for model size, if FCNs with full convolutional support significantly outperform FCNs without it on the same learning task, then the former FCNs are likely approximating MLPs and there may be few spatial features to learn from the data.

The size is the RF is recursively defined for each convolutional layer as [43]

$$rf_0 = 1$$

$$rf_i = rf_{i-1} + (kernel\_size - 1) * jump_{i-1}$$

where

$$jump_0 = 1$$

$$jump_i = jump_{i-1} * stride$$

In other words, the size of the receptive field with respect to the input is a function of the kernel size and the stride of each layer as well as the number of (convolutional) layers. Furthermore, the upsampling operation used in FCNs does not affect the size of the RF because the recursive dependency remains unaffected.

The fact that FCNs do not outperform MLPs but improves upon the DAS benchmark could be because that convolutions may be approximating full connections by having a large receptive field to cover most of or all of the input. To test this hypothesis, we designed two studies.

4.3.1.1 CNR as a Function of Convolutional Kernel Size

The first mechanism for enlarging the RF is by increasing the kernel size. To investigate this potential effect, we used the existing FCN-4 architecture, only to vary the size of the kernel between 3 and 65 (and padding). All four convolutional layers share the same kernel size (but not the same padding size so as to ensure output resolution). In order to create a comparison between FCNs and MLPs, we concatenated the real and imaginary data as a flat list. The model search process undersamples larger kernel sizes due to an increased difficulty of ensuring output

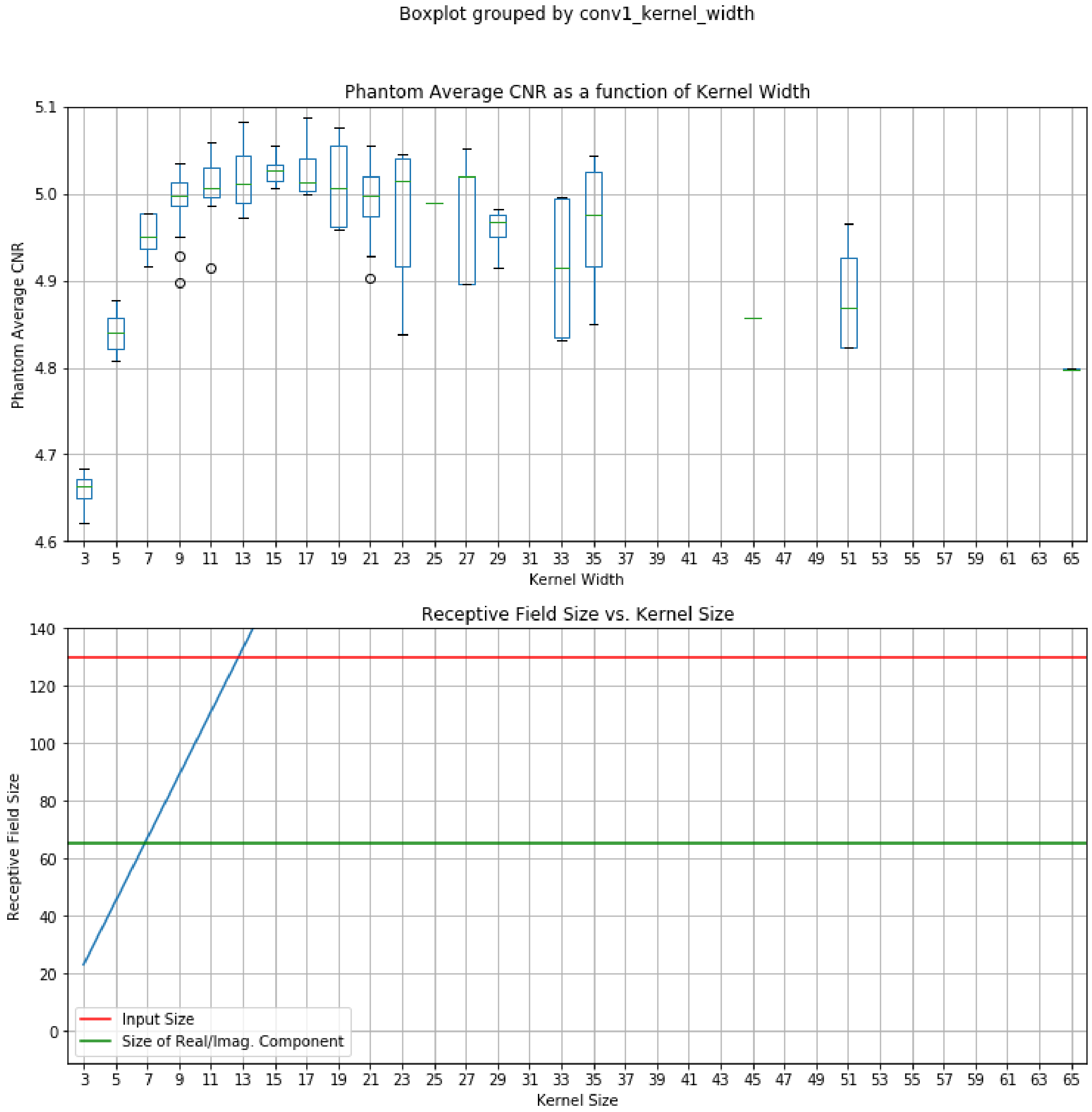

Figure 4.8: Top: Phantom average CNR as a function of convolutional kernel size, FCN-4. Bottom: Receptive field size as a function of convolutional kernel size, FCN-4

resolution at larger kernel sizes. However, Figure 4.8 suggests that models with a kernel size of 7 or above dramatically outperform those under 7, with 17 producing the top-performing model.

An examination of the RF's growth pattern as a function of kernel size for the FCN-4 architecture (Figure 4.8 shows that kernel sizes 7 and 17 are above the intersections with the real/imaginary component size and the input size, respectively. In other words, for the model whose kernel size is 7, all real outputs map to all real inputs, and all imaginary outputs map to all imaginary inputs; for the model with a kernel size of 17, all outputs map to all inputs. In both cases, the complex-component/full convolution support (that is, the full coverage of the RF over the input) is achieved before the final layer. It appears that having component support dramatically improves training, gains from having all input support are relatively modest. Further increases in kernel size do not offer additional benefits.

4.3.2 CNR as a Function of The Number of Convolutional Layers

Because the RF also depends on the number of convolutional layers, I further investigate my hypothesis that the convolution is approximating full connections with a large RF. For this experiment, we implemented a special architecture, FCN-N, with same padding. In other words, the input and output resolutions of each convolutional layer stay the same. In order to isolate the number of convolutional layers as the sole variable for the RF size, we only use kernel sizes smaller than 9 such as 7. As before, I constrain all layers to share the same kernel size. I use a stride of 1 and same padding (which is 3 for 7). Figure 4.9 indicates that 10 layers is the approximate peak, given an incomplete sampling range over the number of layers.

As is shown in Figure 4.9, 10 is just below the number of layers needed for the RF to fully support each complex component, while more layers would be needed (22 or above) for full input support. Thus far, CNR performance appears to depend on an

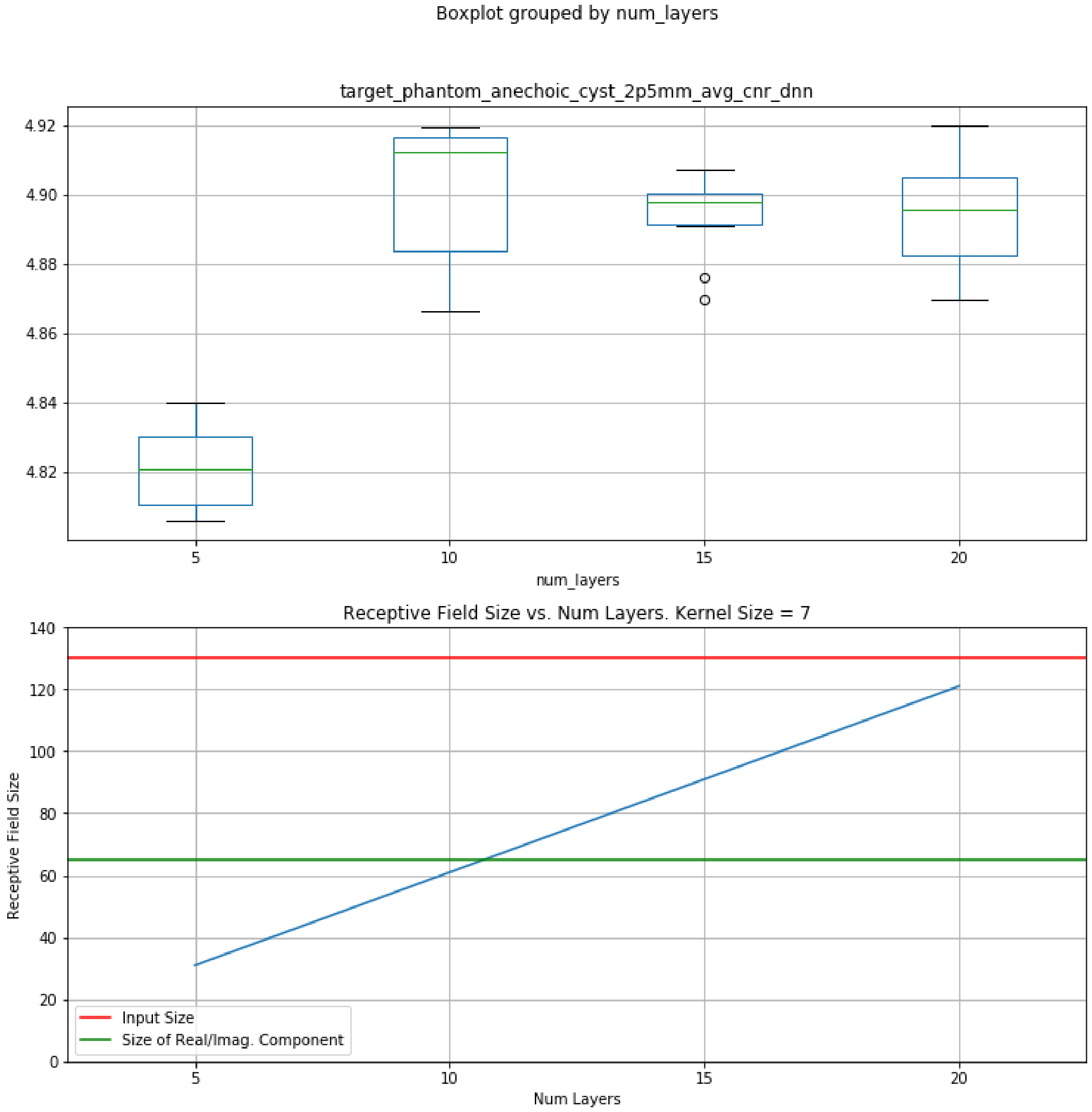

Figure 4.9: Phantom average CNR as a function of the number of convolutional layers, FCN-N. Bottom: Receptive field size as a function of the number of convolutional layers, FCN-N

RF offering full component support and corroborate our hypothesis that convolution alone does not extract unique features not readily learnable by MLPs.

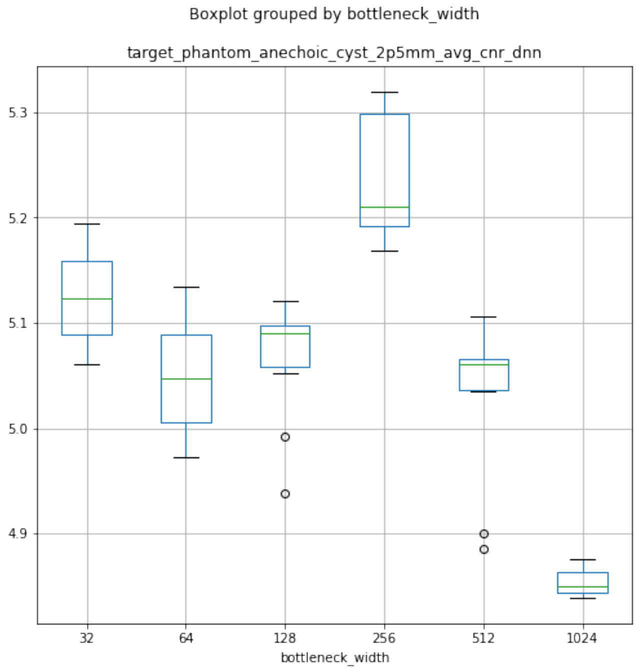

Figure 4.10: Phantom average CNR as a function of bottleneck width, MLPB-5

4.4 Bottleneck and Feature Representation with MLP

Another question remains as to why MLPs work to start with. One hypothesis is that they are overfitting by using bigger models. To test my hypothesis, I implemented bottlenecked MLPs (MLPBs) that restrict the best performing MLP (5 layers, each with 1040 nodes) to significantly reduce its middle layer (to 32, 64, 128, 256, and 512 nodes). For the 50 MLPBs trained, the phantom CNR is significantly lower than that of the best-performing MLP (Figure 4.10). Furthermore, none of the MLPBs outperform the reference MLP in in vivo results. These results suggest that MLPs are not simply overfitting to the training data.

Chapter 5

Discussion

5.1 Challenges to Learning

5.1.1 Learning and Inference Domain Mismatch

One challenge present in this study is the mismatch between the training data and the test data. The training data is inherently not beamformable into images as they are simulated point target responses rather than complete scan or simulated targets. Our evaluation dataset, including the phantom dataset and the in vivo dataset, are of specific targets and are beamformable into images.

5.1.2 Training Objective Mismatch

Another challenge to solving the denoising task is a mismatch between the machine learning domain and the inference domain. In our task, the training and validation in the machine learning process are in the STFT domain. The machine learning objective is minimizing the mean-squared-error (MSE) or Smooth Mean Absolute Error (SmoothL1) loss between the simulated STFT input and the simulated STFT target signals. However, the inference step is in the image domain. We conduct model selection not by their loss curves, but by the beamformed image metrics and the qualitative assessment of resulting images. In other words, the training/validation loss is not necessarily correlated with model performance.

5.2 The Role of Convolution

My earlier study on LeNet-like beamformers indicates that a hybrid architecture consisting of both convolutional and fully-connected layers can achieve a CNR improvement similar to that of MLPs with potential benefits including a larger field of view and fewer network weights. However, models without any fully-connected layer do worse than LeNets. This suggests that convolution by itself may be unable to learn frequency-domain noise suppression in ultrasound beamforming.

Further studies on the effect of increasing the kernel size and the number of layers in FCNs indicate that full convolutional support over a complex data component improves performance. However, this performance increase drops off once the full component RF threshold is reached, as a further increase in either the kernel size or the number of layers no longer has a significant impact on CNR. The fact that FCNs with less than component support significantly underperform suggests that FCNs with large RFs may be approximating MLPs. In fact, considering that larger FCNs with 20 layers also compare unfavorably to MLPs, it is likely that few spatial features can be learned from the frequency-domain input. In addition, this dropoff at full component support may indicate that the real and imaginary components do not depend on each other.

5.3 Trust and Ethics

The inherent lack of interpretability of deep neural networks can call into question the trustworthiness of deep learning-based beamformers. For the end-user (for example, a doctor), a critical issue may be an unpredictable artifact pattern in the resulting ultrasound images. Whereas the benefits and limitations of classic beamforming techniques are well understood, it is unclear how a neural network model arrives at an image or what the failure modes may be. In medical imaging, both false positives and false negatives can have serious health implications for the patient. In addition, this lack of interpretability introduces a responsibility issue between the medical practitioner and the AI practitioner, with the potential to result in moral hazard where one blames the other.

Two approaches could address these issues. One is to relegate deep learning-based beamformers to a more assistive role as an additional beamforming modality in addition to those produced by trusted classic techniques. Another is conducting more extensive patient studies to better understand the performance of neural network beamformers in practice.

Chapter 6

Conclusion

The main contribution of this thesis is the study of convolutional neural networks in suppressing off-axis scattering noise in ultrasound beamforming. I showed that models with convolutional layers can improve the CNR in beamformed images by denoising the channel data in the STFT domain relative to the benchmark DAS beamformed images. Models with both convolutional and fully-connected layers are able to match the performance of MLPs with the additional benefit of having fewer total weights in the network. However, models with only convolutional layers do not perform as well as MLPs. I studied the effect on CNR of the convolutional kernel size and number of convolutional layers and determined that convolution does not contribute to learning any more than it is approximating full connections by having a large enough receptive field to cover either the complex component or the entire input space.

Opportunities for future work include the creation of a new unified training and evaluation dataset from physical phantoms to solve the training and test domain mismatch problem inherent in our study. Another direction is to develop a differentiable loss function from our model selection metric in the beamformed image domain - the CNR, for which I have contributed some initial work. This will solve the learning objective mismatch problem also previously described.

BIBLIOGRAPHY

[1] TG Muir and EL Carstensen. “Prediction of nonlinear acoustic effects at biomedical frequencies and intensities”. In: Ultrasound in medicine & biology 6.4 (1980), pp. 345–357.

[2] HC Starritt et al. “The development of harmonic distortion in pulsed finite-amplitude ultrasound passing through liver”. In: Physics in Medicine & Biology 31.12 (1986), p. 1401.

[3] Victor F Humphrey. “Nonlinear propagation in ultrasonic fields: measurements, modelling and harmonic imaging”. In: Ultrasonics 38.1-8 (2000), pp. 267–272.

[4] Richard SC Cobbold. Foundations of biomedical ultrasound. Oxford university press, 2006.

[5] Arash Anvari, Flemming Forsberg, and Anthony E Samir. “A primer on the physical principles of tissue harmonic imaging”. In: Radiographics 35.7 (2015), pp. 1955–1964.

[6] Mathias Fink. “Time reversal of ultrasonic fields. I. Basic principles”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 39.5 (1992), pp. 555–566.

[7] Johan Fredrik Synnevag, Andreas Austeng, and Sverre Holm. “Adaptive beamforming applied to medical ultrasound imaging”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 54.8 (2007), pp. 1606–1613.

[8] Iben Kraglund Holfort, Fredrik Gran, and Jorgen Arendt Jensen. “Broadband minimum variance beamforming for ultrasound imaging”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 56.2 (2009), pp. 314–325.

[9] Raoul Mallart and Mathias Fink. “Adaptive focusing in scattering media through sound-speed inhomogeneities: The van Cittert Zernike approach and focusing criterion”. In: The Journal of the Acoustical Society of America 96.6 (1994), pp. 3721–3732.

[10] KW Hollman, KW Rigby, and M O’donnell. “Coherence factor of speckle from a multi-row probe”. In: 1999 IEEE Ultrasonics Symposium. Proceedings. International Symposium (Cat. No. 99CH37027). Vol. 2. IEEE. 1999, pp. 1257–1260.

[11] Muyinatu A Lediju et al. “Short-lag spatial coherence of backscattered echoes: Imaging characteristics”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 58.7 (2011), pp. 1377–1388.

[12] Brett Byram et al. “A model and regularization scheme for ultrasonic beamforming clutter reduction”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 62.11 (2015), pp. 1913–1927.

[13] A. C. Luchies and B. C. Byram. “Deep Neural Networks for Ultrasound Beamforming”. In: IEEE Transactions on Medical Imaging 37.9 (Sept. 2018), pp. 2010–2021. DOI: 10.1109/TMI.2018.2809641.

[14] Adam C Luchies and Brett C Byram. “Training improvements for ultrasound beamforming with deep neural networks”. In: Physics in medicine and biology (2019).

[15] 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. Springer. 2015, pp. 234–241.

[16] R. J. G. van Sloun, R. Cohen, and Y. C. Eldar. “Deep Learning in Ultrasound Imaging”. In: Proceedings of the IEEE (2019), pp. 1–19. DOI: 10.1109/JPROC.2019.2932116.

[17] D. Hyun et al. “Beamforming and Speckle Reduction Using Neural Networks”. In: IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 66.5 (May 2019), pp. 898–910. DOI: 10.1109/TUFFC.2019.2903795.

[18] Ted Christopher. “Finite amplitude distortion-based inhomogeneous pulse echo ultrasonic imaging”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 44.1 (1997), pp. 125–139.

[19] B. Ward, A. C. Baker, and V. F. Humphrey. “Nonlinear propagation applied to the improvement of resolution in diagnostic medical ultrasound”. In: The Journal of the Acoustical Society of America 101.1 (1997), pp. 143–154. DOI: 10.1121/1.417977. eprint: https://doi.org/10.1121/1.417977. URL: https://doi.org/10.1121/1.417977.

[20] M. A. Averkiou, D. N. Roundhill, and J. E. Powers. “A new imaging technique based on the nonlinear properties of tissues”. In: 1997 IEEE Ultrasonics Symposium Proceedings. An International Symposium (Cat. No.97CH36118). Vol. 2. Oct. 1997, 1561–1566 vol.2. DOI: 10.1109/ULTSYM.1997.663294.

[21] TA Whittingham. “Tissue harmonic imaging”. In: European radiology 9.3 (1999), S323–S326.

[22] Kazuyuki Dei. “Model-Based Ultrasound Imaging for Challenging Acoustic Clutter Suppression”. PhD thesis. Vanderbilt University, 2019.

[23] Jeremy J Dahl et al. “Coherence beamforming and its applications to the difficult-to-image patient”. In: 2017 IEEE International Ultrasonics Symposium (IUS). IEEE. 2017, pp. 1–10.

[24] M. A. Lediju Bell, J. J. Dahl, and G. E. Trahey. “Resolution and brightness characteristics of short-lag spatial coherence (SLSC) images”. In: IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 62.7 (July 2015), pp. 1265–1276. DOI: 10.1109/TUFFC.2014.006909.

[25] T. Szasz, A. Basarab, and D. Kouamé. “Beamforming Through Regularized Inverse Problems in Ultrasound Medical Imaging”. In: IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 63.12 (Dec. 2016), pp. 2031–2044. DOI: 10.1109/TUFFC.2016.2608939.

[26] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[27] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

[28] 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.

[29] James Bergstra and Yoshua Bengio. “Random search for hyper-parameter optimization”. In: Journal of Machine Learning Research 13.Feb (2012), pp. 281–305.

[30] Barret Zoph and Quoc V Le. “Neural architecture search with reinforcement learning”. In: arXiv preprint arXiv:1611.01578 (2016).

[31] Hieu Pham et al. “Efficient neural architecture search via parameter sharing”. In: arXiv preprint arXiv:1802.03268 (2018).

[32] Esteban Real et al. “Large-scale evolution of image classifiers”. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org. 2017, pp. 2902–2911.

[33] Haifeng Jin, Qingquan Song, and Xia Hu. “Auto-keras: An efficient neural architecture search system”. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM. 2019, pp. 1946–1956.

[34] 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.

[35] 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. Springer. 2015, pp. 234–241.

[36] Dana H Ballard. “Modular Learning in Neural Networks.” In: AAAI. 1987, pp. 279–284.

[37] Christian Szegedy et al. “Going deeper with convolutions”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 1–9.

[38] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. “Image orientation estimation with convolutional networks”. In: German Conference on Pattern Recognition. Springer. 2015, pp. 368–378.

[39] Yeo Hun Yoon et al. “Efficient b-mode ultrasound image reconstruction from sub-sampled rf data using deep learning”. In: IEEE transactions on medical imaging 38.2 (2018), pp. 325–336.

[40] Dongwoon Hyun et al. “Beamforming and Speckle Reduction Using Neural Networks”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 66.5 (2019), pp. 898–910.

[41] Jørgen Arendt Jensen and Niels Bruun Svendsen. “Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers”. In: IEEE transactions on ultrasonics, ferroelectrics, and frequency control 39.2 (1992), pp. 262–267.

[42] Jørgen Arendt Jensen. “Field: A program for simulating ultrasound systems”. In: 10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1: 351–353. Citeseer. 1996.

[43] Wenjie Luo et al. “Understanding the effective receptive field in deep convolutional neural networks”. In: Advances in neural information processing systems. 2016, pp. 4898–4906.