Deep Learning Methods for Estimation of Elasticity and Backscatter Quantitative Ultrasound

Ali Kafaei Zad Tehrani

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Electrical and Computer Engineering) at
Concordia University
Montréal, Québec, Canada

May 2023

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Ali Kafaei Zad Tehrani**
Entitled: **Deep Learning Methods for Estimation of Elasticity and Backscatter Quantitative Ultrasound**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Chair
Dr. Yuhong Yan

______________________ External Examiner
Dr. Michael Oelze

______________________ External to Program
Dr. Javad Dargahi

______________________ Examiner
Dr. Wei-Ping Zhu

______________________ Examiner
Dr. Wahab Hamou-Lhadj

______________________ Thesis Supervisor
Dr. Hassan Rivaz

Approved ______________________
Dr. Jun Cai, Graduate Program Director

____
Date of Defence ______________________
Dr. Mourad Debbabi, Dean, Engineering and Computer Science

Abstract

Deep Learning Methods for Estimation of Elasticity and Backscatter Quantitative Ultrasound

Ali Kafaei Zad Tehrani, Ph.D.

Concordia University, 2023

Ultrasound (US) imaging is increasingly attracting the attention of both academic and industrial researchers due to being a real-time and nonionizing imaging modality. It is also less expensive and more portable compared to other medical imaging techniques. However, the granular appearance hinders the interpretation of US images, hindering its wider adoption. This granular appearance (also referred to as speckles) arises from the backscattered echo from microstructural components smaller than the ultrasound wavelength, which are called scatterers. While significant effort has been undertaken to reduce the appearance of speckles, they contain scatterer properties that are highly correlated with the microstructure of the tissue that can be employed to diagnose different types of disease. There are many properties that can be extracted from speckles that are clinically valuable, such as the elasticity and organization of scatterers. Analyzing the motion of scatterers in the presence of an internal or external force can be used to obtain the elastic properties of the tissue. The technique is called elastography and has been widely used to characterize the tissue. Estimating the scatterer organization (scatterer number density and coherent to diffuse scattering power) is also crucial as it provides information about tissue microstructure and potentially aids in disease diagnosis and treatment monitoring. This thesis proposes several deep learning-based methods to facilitate and improve the estimation of speckle motion and scatterer properties, potentially simplifying the interpretation of US images. In particular, we propose new methods for displacement estimation in Chapters 2 to 6 and introduce novel techniques in Chapters 7 to 11 to quantify scatterers' number density and organization.

Acknowledgments

I want to offer my sincerest thanks to my knowledgeable, and supportive supervisor, Dr. Hassan Rivaz. His support allowed me to expand my academic knowledge during my Ph.D. at Concordia University. In the difficult time of my research, he guided me to mature and face the reality of academic life. His attention to detail and intuition for supervising academic research resulted in great publications. I especially appreciate Dr. Ivan M. Rosado-Mendez for his invaluable support in editing my research papers and mentoring my academic career. I also extend my gratitude to my colleagues in the IMPACT lab for their guidance and for sharing their experiences with me. Special thanks to funding resources (Natural Science Engineering Council of Canada (NSERC) which facilitated and encouraged me to continue strongly in the rise and falls of my research life. Many thanks to the thesis committee. You provided valuable hints for improving this thesis.

I am deeply indebted to my parents for their emotional and financial support. I wish they would always be proud of me and that I could repay their deeds with my success.

Contents

List of Figures

List of Tables

List of Abbreviations

AUC Area Under Curve

BN Batch Normalization

BCE Binary Cross Entropy

BNN Bayesian Neural Network

CNN Convolutional Neural Network

CNR Contrast to Noise Ratio

DP Dynamic Programming

ESD Effective Scatterer Diameter

EPE End Point Error

EPR Effective Poisson's Ratio

FDS Fully Developed Speckle

HK-distribution Homodyned k-distribution

MLP Multi Layer Perceptron

NCC Normalized Cross Correlation

PSF Point Spread Function

QUS Quantitative Ultrasound

RF Radio Frequency

SR Strain Ratio

SNR Signal to Noise Ratio

SVM Support Vector Machine

UDS Under Developed Speckle

US Ultrasound

USE Ultrasound Elastography

List of Symbols

$\boldsymbol{Ds_{gt}}$	Ground truth scatterer number density
$\widehat{\boldsymbol{Ds}}$	Predicted scatterer number density
$\boldsymbol{f_s}$	Sampling frequency
$\boldsymbol{f_c}$	Center frequency
$\hat{\boldsymbol{I}}$	Warped image
$\boldsymbol{J_0(.)}$	Zero-order Bessel function
$\mathbf{S}$	Stop gradient operation
$\boldsymbol{m}$	Nakagami shape parameter
$\boldsymbol{U}$	Axial displacement
$\boldsymbol{V}$	Lateral displacement
$\boldsymbol{V}$	Lateral displacement
$\boldsymbol{W^f}$	Forward flow

$\boldsymbol{W^b}$	Backward flow
$\boldsymbol{W^f}$	Forward flow
$\boldsymbol{\Vert . \Vert_2}$	L_2 Norm
$\boldsymbol{\epsilon}$	Strain (derivative of displacement)
$\boldsymbol{\phi()}$	Charbonnier loss
$< . >$	Averaging operation
$\boldsymbol{\Omega}$	Nakagami scale parameter
$\boldsymbol{\Gamma}$	Gamma function

Chapter 1

Introduction

Ultrasound (US) imaging is a popular modality in diagnosis and image-guided interventions thanks to its portability, affordability, and non-invasiveness. However, several types of artifacts make the interpretation of US images difficult. Cells, collagen, microcalcifications, and other microstructures are often smaller than the wavelength of the US wave, and as such, scatter the wave and create the granular appearance in B-mode images called speckles. Although speckles make the interpretation and visual inspection of B-mode images difficult, they provide useful information about the physical properties of the tissue which has been used to characterize the tissue [1, 2].

Quantitative ultrasound (QUS) aims to analyze the speckle and obtains parameters which are highly correlated with the physical properties of the tissue. QUS parameters have been utilized to diagnose different types of abnormalities including the breast tumors [3], and fibrosis in the liver [4]. There are different QUS parameters including speed of sound, elasticity, effective scatterer diameter (ESD), scatterer number density, coherent scattering ratio, backscattering coefficient, and attenuation coefficient. These parameters can be obtained by analyzing the backscattered signal in the time domain (scatterer number density and coherent scattering ratio), in frequency domain (backscattering coefficient and attenuation coefficient) or by measuring the motion by an external/internal force (elasticity) [5–8].

In this thesis, we focus on two QUS parameters: elasticity and scatterer number density. We aim to employ deep learning to improve motion estimation for elasticity quantification and boost the scatterer number density estimation performance. The rest of the thesis is organized in the following manner: we first introduce the elasticity and scatterer number density parameters. In the next step, novel methods are proposed for improving the motion estimation in Chapters 2, 3, 4, 5, and 6. New methods are then introduced for scatterer number density estimation in Chapters 7, 8, 9, 10 and 11.

1.1 Strain imaging and elastography

Materials exhibit different behavior in the presence of an external force. Elastic materials tend to return to their original shape after the external force is removed. The stress versus strain curve for these materials is linear and the slope of this line is called elasticity modulus. The linear relation between stress and strain is fixed and independent of time in elastic materials, as opposed to viscoelastic materials in which the stress-strain relationship varies by time. Elasticity represents the material's resistance to deformation in the presence of force. Materials with higher elasticity modulus tend to deform less than materials having lower elasticity modulus [9]. Abnormalities in the tissue might have a different elasticity modulus than normal tissue; therefore, quantifying the elasticity modulus can have clinical values.

Ultrasound elastography (USE) aims to quantify the elasticity by finding the motion from ultrasound data. The source of the force can be internal (motion of the heart in echocardiography), by the operator using the probe (free-hand palpation), or by an acoustic radiation force. Among different USE methods, free-hand palpation has gained much popularity due to its simplicity, low cost, and ease-of-use. The basic idea of the free-hand palpation method is that the operator compresses the tissue by the US probe. The images before and after compression are compared to obtain the displacement of each individual sample. This displacement can be used to obtain a strain map which has relative elasticity information [10, 11].

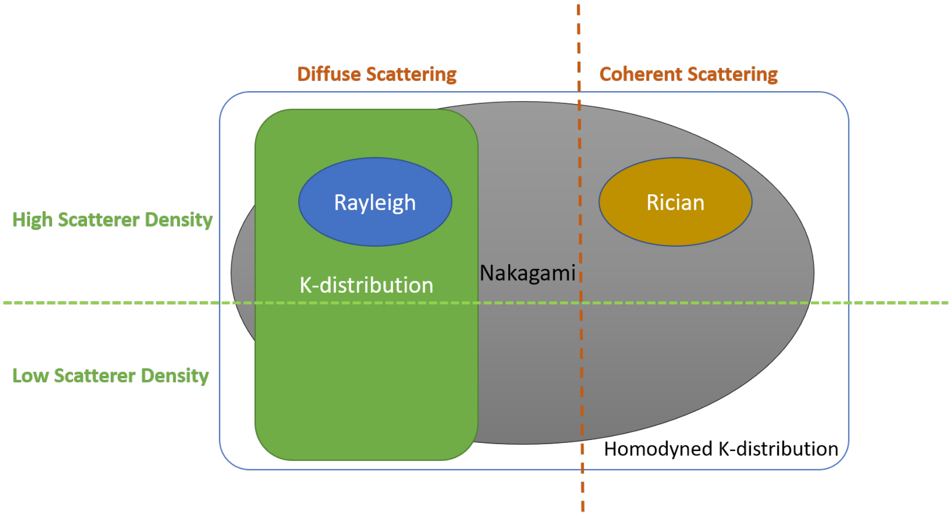


Figure 1.1: Summary of the distribution used to explain envelope data.

1.2 Scatterers number density and scatterer organization

The US radio-frequency (RF) echo is the result of contribution from all scatterers and coherent strong reflectors inside the resolution cell which can be explained by a random walk model. Each scatterer has a random phase and amplitude. In the absence of a coherent strong reflector, the RF echo is the summation of all scatterers inside the resolution cell having independent phases and amplitudes. The random walk model is given in Eq 1.1. The phase (Θ) is uniformly distributed over the interval $[-\pi,\pi)$ and the amplitude (x) has normal distribution having zero mean and standard deviation of σ [12].

$$X = \sum_{i=0}^{N-1} x_i e^{j\Theta_i} \tag{1.1}$$

If N is large enough ($N > 10$), the absolute (envelope) of X can be modeled by Rayleigh distribution, and the scattering type is considered as the diffuse scattering [5]. If RF echo reflected by the scatterers has a strong constant component, the random walk is changed to the Eq 1.2. The parameter S is the constant component which is also called coherent scattering [12]. This type of scattering happens when there are specular reflectors or periodically unresolved scatterers. Under this condition and assuming having large N, the distribution of the echo envelope would be Rice distribution [5, 13].

$$X = S + \sum_{i=0}^{N-1} x_i e^{j\Theta_i} \tag{1.2}$$

When the number of N is not large enough ($N < 10$), the N is modeled by a negative binomial distribution [14]. If there is no coherent component ($S = 0$), K-distribution can be used to model the echo envelope. While Homodyned K-distribution should be used when the coherent component is present [14]. The Nakagami model has also been proposed to explain the envelope data. This model is obtained experimentally by Dr.Nakagami to describe fading in wireless channels [15]. The diagram in Fig. 1.1 summarizes the distribution used to describe envelope data.

1.3 Thesis statement

High-quality and real-time visualization of QUS parametric images, such as elasticity and scatterer number density, can be clinically significant since quantitative measurements are not subject of interpretation. Current speckle tracking methods and QUS parameter extractions are computationally expensive, and their accuracy is limited. Deep learning methods can be employed to tackle the mentioned problems. The deep learning models are efficiently implemented on GPUs to be fast enough to work in real-time. Furthermore, they can find highly non-linear relationships between the inputs and their corresponding outputs. In this thesis, we employ deep learning models to accomplish the following objectives:

1. Improve the quality of the estimation of strain images in elastography.

2. Reduce the computation time to be able to estimate strain images in real-time.

3. Improve estimation of the scatterer number density parametric image using a limited sample size.

4. Eliminating the need of patching for estimating scatterer number density and their organization to avoid unreliable estimates in the boundary regions.

Table 1.1: The presented chapters and their corresponding publication. The journals are in bold.

	Summary	Publication
Chapter 2	An Optical flow CNN is modified to adapt for USE	**IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, 2021
Chapter 3	The proposed network in Chapter 1 is altered to improve the network for large displacements	SPIE (Ultrasonic Imaging and Tomography), 2021
Chapter 4	A bi-directional unsupervised method is introduced for USE	**IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, 2022
Chapter 5	A novel regularization method is proposed to improve lateral displacement estimation	**IEEE Transactions on Medical Imaging**, 2023
Chapter 6	Infusing known operators in deep networks	Under review in medical image computing and computer-assisted intervention (MICCAI), 2023
Chapter 7	A path-based CNN is proposed to classify the scatterer number density	**IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, 2021
Chapter 8	A novel method is proposed to segment the US envelope data without patching	**IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, 2022
Chapter 9	A novel method is introduced to estimate the scatterer number density without patching	SPIE (Ultrasonic Imaging and Tomography), 2023
Chapter 10	HK-distribution parameters are estimated using a Bayesian neural network, which can also quantify the uncertainty	IEEE International Symposium on Biomedical Imaging (ISBI), 2023
Chapter 11	Deep autoencoder feature projection for accurate HK-distribution parameter estimation	**IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, (under preparation)

1.4 Organization of the thesis

The remaining of the thesis is organized as follows. Chapters 2 to 6 are dedicated to motion estimation in USE. Chapters 7 to 11 are allocated to scatterer number density classification, segmentation, and regression. The summary of the chapters provided in this thesis is given in Table 1.1. It should be noted that we did not include some of our publications to improve the flow and eliminate overlap between different chapters.

1.5 List of published works

The list of published/in preparation journal papers is given below.

1. Tehrani AK, Rivaz H. Displacement estimation in ultrasound elastography using pyramidal convolutional neural network. IEEE transactions on ultrasonics, ferroelectrics, and frequency control. 2020 Feb 10;67(12):2629-39.

2. Tehrani AK, Amiri M, Rosado-Mendez IM, Hall TJ, Rivaz H. Ultrasound scatterer density classification using convolutional neural networks and patch statistics. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2021 Apr 26;68(8):2697-706.

3. Tehrani AK, Rosado-Mendez IM, Rivaz H. Robust scatterer number density segmentation of ultrasound images. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2022 Jan 19;69(4):1169-80.

4. Tehrani AK, Sharifzadeh M, Boctor E, Rivaz H. Bi-directional semi-supervised training of convolutional neural networks for ultrasound elastography displacement estimation. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2022 Jan 27;69(4):1181-90.

5. Tehrani AK, Ashikuzzaman M, Rivaz H. Lateral Strain Imaging using Self-supervised and Physically Inspired Constraints in Unsupervised Regularized Elastography. IEEE Transactions on Medical Imaging. 2023 May.

6. Tehrani AK, Rosado-Mendez IM, Cloutier G, Rivaz H. Deep Autoencoder Feature Projection for Accurate Homodyned K-distribution Parameter Estimation in Quantitative Ultrasound. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control (in preparation).

The list of published/submitted conference papers is given below.

1. Tehrani AK, Amiri M, Rivaz H. Real-time and high-quality ultrasound elastography using convolutional neural network by incorporating analytic signal. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC) 2020 Jul 20 (pp. 2075-2078). IEEE.

2. Tehrani AK, Rivaz H. MPWC-Net++: evolution of optical flow pyramidal convolutional neural network for ultrasound elastography. In Medical Imaging 2021: Ultrasonic Imaging and Tomography 2021 Feb 15 (Vol. 11602, pp. 14-23). SPIE.

3. KZ Tehrani A, Mirzaei M, Rivaz H. Semi-supervised training of optical flow convolutional neural networks in ultrasound elastography. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23 2020 (pp. 504-513). Springer International Publishing.

4. Tehrani AK, Amiri M, Rosado-Mendez IM, Hall TJ, Rivaz H. A pilot study on scatterer density classification of ultrasound images using deep neural networks. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC) 2020 Jul 20 (pp. 2059-2062). IEEE.

5. KZ Tehrani A, Rivaz H. Physically inspired constraint for unsupervised regularized ultrasound elastography. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part IV 2022 Sep 16 (pp. 218-227). Cham: Springer Nature Switzerland.

6. Tehrani AK, Rosado-Mendez IM, Rivaz H. Deep Estimation of Speckle Statistics Parametric Images. In 2022 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC) 2022 Jul 11 (pp. 3907-3910). IEEE.

7. Tehrani AK, Rosado-Mendez IM, Rivaz H. Homodyned K-distribution: parameter estimation and uncertainty quantification using Bayesian neural networks. IEEE 20th In-

ternational Symposium on Biomedical Imaging (ISBI), https://arxiv.org/abs/2211.00175, 2022 Oct 31.

8. Tehrani AZ, Rosado-Mendez I, Whitson H, Rivaz H. A deep learning approach for patchless estimation of ultrasound quantitative parametric image with uncertainty measurement. In Proc. of SPIE Vol 2023 (Vol. 12470, pp. 1247010-1).

9. KZ Tehrani A, Rivaz H. Infusing physically inspired known operators for ultrasound elastography displacement estimation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2023 (submitted)

Chapter 2

Displacement Estimation in Ultrasound Elastography using Pyramidal Convolutional Neural Network

Ultrasound imaging is being increasingly used as an inexpensive and easy-to-use imaging modality in numerous diagnosis and image-guided intervention applications. Ultrasound elastography (USE) is an imaging technique that reveals viscoelastic properties of tissue, and has been applied to many applications including breast lesion characterization [16] and ablation monitoring [17–20]. USE compliments B-mode ultrasound by providing biomechanical properties of the tissue [21].

Among different USE methods, free-hand palpation has gained much popularity due its simplicity, low cost and ease-of-use. The basic idea of free-hand palpation method is that the operator compresses the tissue by the ultrasound probe. The images before and after compression are compared to obtain the displacement of each individual sample. This displacement can be used to obtain strain map which has relative elasticity information [10, 11]. The quality of USE mainly depends on the fidelity of the displacement estimation. Window-based [11, 22–25] and optimization-based [26–28] methods are two main approaches for displacement estimation in USE. Window-based methods try to find the displacements of each individual sample by considering a window around the sample in pre- and post-compression images and assuming that the displacement within the window is constant. In the next step, a similarity metric such as normalized cross correlation (NCC) is chosen to find the corresponding windows [11, 22]. Optimization-based methods use a regularized cost function to find the displacements, therefore they are more robust to signal decorrelation and out of plane motion [26, 29, 30]. Global ultrasound elastography (GLUE) is a recent optimization-based method [27] with an implementation available online at code.sonography.ai. GLUE aims to estimate sub-pixel displacement and requires initial estimate of the displacement which is obtained by dynamic programming (DP) [28]. The displacement estimation in USE can also be viewed as a non-rigid registration [31] or optical flow problem [32–34].

Convolutional neural network (CNN) models have been successfully trained to perform many applications such as classification [35] and segmentation [36]. Recently, CNN has been used for optical flow problem [37–40]. FlowNet is among the first attempts to extract optical flow using deep learning architectures [38]. Before FlowNet, patch- and point-based deep learning methods were used. These methods were only able to extract optical flow of a point or a small patch of the images. As such, they were computationally expensive as it was necessary to run them many times to cover the entire image. Two variants of FlowNet were proposed [38]: FlowNetS and FlowNetC. FlowNetS has a U-shape architecture with a contracting and an expanding path, and as such, shares many similarities with U-Net [36]. FlowNetS uses coarse outputs in the refinement section to build the finer outputs and uses multi-scale loss function for optimization. FlowNetC is the other variant of FlowNet that differs from FlowNetS only in the contracting part. Instead of concatenating input images and using a U-shape network, it extracts features of each input separately and exploits a correlation layer to merge information from features of the two images. Although they reported better performance with FlowNetS, Mayer *et al.* [39] show that with better learning schedule and more training data, FlowNetC outperforms FlowNetS.

Following the success of FlowNet, Mayer *et al.* [39] stack several FlowNetS and FlowNetC to improve the accuracy of FlowNet. They show that the optimum architecture is to use FlowNetC as the first block, followed by two FlowNet blocks. This architecture is called

FlowNet2CSS. In order to improve the network accuracy for small displacements, another FlowNet is used and it is trained on a database with small displacements. A fusion network is then proposed to fuse the outputs of FlowNet (trained to provide small displacements) and FlowNet2CSS. This network called FlowNet2 achieves outstanding performance and is the first CNN architecture that outperforms traditional optical flow algorithms. FlowNet2 performs well but with one drawback. It has many parameters (around 160 million), which makes training difficult and renders inference both computationally expensive and memory exhaustive.

Designing a network with fewer parameters is an active field of research. Ranjan and Black [40] proposed spatial pyramide network (SPynet), which has much lower number of parameters but with an accuracy close to that of FlowNet. Recently, pyramid, warping and cost volume network (PWC-Net) [37] was proposed, which not only has fewer number of parameters (around 9 million), but also achieves slightly better accuracy compared to FlowNet2. The main idea of PWC-Net is to use pyramidal structure to estimate the optical flow in each level and warp the features by the estimated flow to reduce the search range of the next level. This network utilizes cost volumes (similar to correlation layer) in each pyramid level to extract correlation between features of the two images, and unlike SPynet, warps the features of the second image instead of the image itself.

There are two important differences between USE displacement estimation and optical flow that limits the use of optical flow CNN models: 1) Accurate subsample displacement estimation is paramount in USE; 2) RF data is characteristically different from images in computer vision because it has a very large frequency content. Therefore, any optical flow method used for USE must preserve and utilize the information of high frequency RF data for an accurate and robust displacement estimation. USE is a new and less explored deep learning application in medical image processing. At the time of writing this paper in 2019, only a few papers tried to apply neural networks for USE [32–34, 41, 42].

A deep learning architecture was proposed by Wu *et al.* [41] to estimate displacement and strain. A patch around the sample of interest is fed to the network and the displacement and the strain of the patch are estimated. Gao *et al.* [33] further improved this network by introducing learning-using-privileged-information (LUPI). LUPI uses displacement as the intermediate loss, and results in better generalization and higher accuracy compared to [41], as well as non-deep learning approaches of DP [28] and optical flow [43]. The main drawback of the networks is that in order to compute the strain and the displacement of an image pair, it is required to run the network many times since this network only takes small patches as the input. In [32, 44], we used FlowNet2 for USE. But since the displacement estimates were not precise even after fine-tuning with Field II simulations [45, 46], they were used as the initial estimator for GLUE, replacing dynamic programming [28] with FlowNet2. In [34], FlowNetCSS is used for USE and it was shown that using simulated images for fine-tuning can be beneficial. The main contribution of our work in this chapter can be summarized as:

- Two networks, namely Modified Pyramid Warping and Cost volume Network (MPWC-Net) and RFMPWC-Net are proposed for USE, both based on PWC-Net. Both of our proposed networks substantially outperform PWC-Net in USE.
- FlowNet2 has been recently exploited for USE [32, 34]. Our proposed networks are based on PWC-Net, and have more than 10 times fewer parameters compared to FlowNet2 while substantially outperforming it in USE. This is paramount as GPU memory is often a critical bottleneck.
- A fine-tuning strategy and a loss function are proposed to improve the displacement estimation and the corresponding strain quality using simulated data.
- The performance of top optical flow CNNs in USE is presented and analyzed.

2.1 Methods

2.1.1 PWC-Net

The core ideas of PWC-Net are to utilize pyramid structures, cost volume and a refinement network. This leads to substantial reduction of number of the parameters and improvement in the accuracy. Using the pyramid structure reduces the displacement required to be estimated in each resolution, resulting in a smaller search range. The coarser resolution finds large displacements and removes these displacements by warping the second image features with

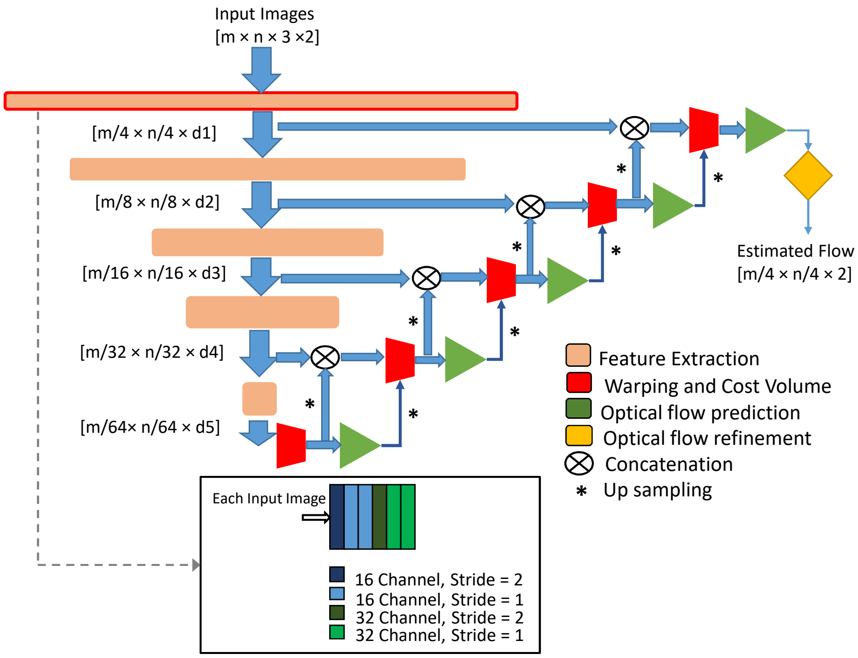


Figure 2.1: PWC-Net structure. The feature extraction layer of the final pyramid is outlined by a red box (all kernels in the box are 3×3). m, n, d_x denote image size in axial direction, lateral direction and number of channels of the corresponding layer, respectively.

the estimated displacements, and the finer resolution estimates the smaller displacements from the warped image. Unlike FlowNet2 that warps the moved images, PWC-Net warps the features of the moved images so that fewer number of parameters are required for optical flow estimation. PWC-Net utilizes cost volume in each pyramid level. Unlike FlowNet2 that uses correlation layer (cost volume) only as the first block and reports over fitting by using more correlation layers, PWC-Net uses cost volume in all pyramid levels, substantially reducing the number of parameters. Finally, PWC-Net employs a refinement network which is a post processing stage to improve the quality of the estimated optical flow in the last pyramid level [37]. As shown in Fig. 2.1, PWC-Net is composed of 4 different blocks: feature extraction, warping and cost volume, optical flow estimation and refinement network.

To compute each pyramid output, first the input images are fed into a CNN in order to extract features from the image pyramid, transforming it to a feature pyramid. Then the warping block warps the second image feature map toward the first one. At the next step, a cost volume is created using the first image feature map and the warped one. This cost volume is then used as an input to the optical flow estimator block in order to estimate the flow. Finally, a refinement network is used to post-process the optical flow. The loss function used in PWC-Net is a multi-scale loss defined in [37]:

$$L(\Theta) = \sum_{l=l_0}^{L} \alpha_l(\|D_\Theta^l(x) - D_{GT}^l(x)\|_q) + \|\Theta\|_2 \tag{2.1}$$

where Θ represents the learnable parameters and D_Θ^l and D_{GT}^l denote the estimated and the ground truth flows at the lth level, respectively. This is a regularized loss function where $q < 2$ is chosen to give less penalty to outliers. Also, $\|\Theta\|_2$ is the weight decay which encourages the learnable weights to have small magnitude in order to improve the generalization of the network. For each output resolution, a weight is considered to contribute (α) in the loss function. Generally, higher weights are given to coarser outputs since coarser outputs contribute to build finer ones. The coarse outputs are employed as intermediate losses, and the corresponding ground truths are obtained by down sampling the displacement.

2.1.2 Proposed Methods for USE

It is common to modify a well-known network for a specific task. As an example, in [47], VGG-16 and ResNet-101 are modified for semantic segmentation by changing the dilations and strides of the convolution layers. In this work, PWC-Net structure is modified for USE wherein accurate subsample displacement estimation using RF data is critical.

PWC-Net contains feature extraction, cost volume and optical flow estimation layer for each pyramid. There are 5 levels and the coarser levels contribute to the estimation of finer resolution levels. As depicted in Fig. 2.1, the output size is 4 times smaller than the input images. The feature extraction part of the final pyramid level (the first feature extraction layer with red outline, shown in the box) downsamples the input by a factor of 4 using two convolution layers with $stride = 2$. The downsampling of the input images is quite reasonable for computer vision images since there is negligible information in high frequencies. This downsampling reduces the computation complexity, improves the network robustness to noise, and more importantly decreases the displacement and the required search range of the cost volume. However, in USE, accurate subsample displacement estimation is essential and there is valuable phase information in high frequencies, rendering this downsampling detrimental. To cope with this issue, we replace the first two convolution layers with $stride = 2$ with convolution layers with $stride = 1$. This modification provides more information related to displacement estimation for each pyramid level, and useful features can be obtained from high frequency RF data.

An important aspect is the input of the network. RF data, B-mode image and envelope of RF data can be used for displacement estimation. Generally, RF data is the most informative signal for estimation of fine displacements, but using RF data might result in unreliable regions in the pyramidal structures. Envelope and B-mode only contain low frequency information of RF data that can be used for approximation of the displacement but they cannot provide accurate displacement. B-mode and envelope can provide useful information in coarse pyramid levels while RF data contains detailed information for high resolution and high-quality displacement estimation. Consequently, two networks are proposed to exploit RF, B-mode and envelope. In both networks, the downsampling operations (strides = 2) of the final pyramid level are removed. In the first network, we concatenate RF data, B-mode and envelope to generate a three-channel input for the network. We name this network modified PWC-Net (MPWC-Net). This network uses information of B-mode and envelope in low resolutions where RF data cannot provide useful information due to information loss and the network exploits RF data in high resolutions to have high quality subsample displacement estimation.

In the second network, RF data, B-mode and envelope are combined in a different fashion. Concatenated RF, B-mode and envelope is used for displacement estimation of all pyramid levels except for the last pyramid level which has the highest resolution and for that level, only RF data is used for displacement estimation. A feature extraction layer is added to extract useful information of only RF data in the final pyramid level. The block diagram of this method is depicted in Fig. 2.2. The last layer has the same structure and weights as the feature extraction layer of the main concatenated inputs so no more training is required. This network produces more accurate displacement compared to MPWC-Net, especially in noisy situations because although B-mode and envelope are helpful in low pyramid levels and remove outlier regions, they reduce the accuracy of the network in the final pyramid

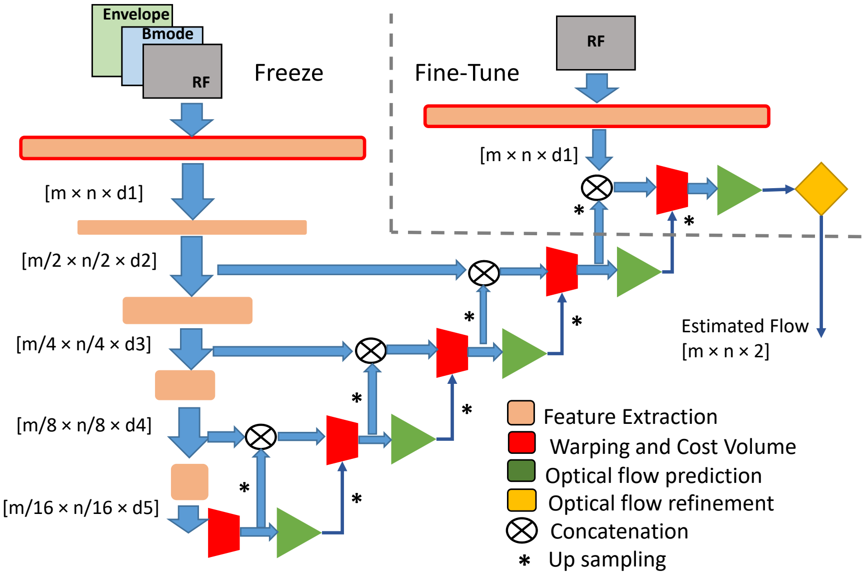


Figure 2.2: Proposed RFMPWC-Net structure. A feature extraction layer is added to use features of only RF data to estimate final resolution. The feature extraction layers with red outline have the same weights. The output size of each feature extraction layer, fine-tuned sub-network and frozen network are specified.

level. We call this network RF Modified PWC-Net (RFMPWC-Net).

2.1.3 Simulation Dataset

As part of the published manuscript of this chapter, we generate a simulation dataset using Field II [45, 46]. The dataset consists of one or two inclusions with random positions. The Young's modulus of the tissue is randomly set between 18 to 23 kPa, and the Young's modulus of the hard inclusion is randomly set to a value in the range of 40 to 60 kPa. The average strain varies between 0.5 to 4.5 % and displacements are estimated by finite element method (FEM) using the ABAQUS software. The cubic interpolation method is used to obtain the displacements of the scatterers from the nodes obtained by FEM. These scatterers are utilized to simulate ultrasound images using the Field II toolbox [45, 46] with a center frequency of 5 MHz.

24 different phantoms with 10 different average strain values and 10 different random scatterer realizations with different positions are simulated (for each phantom 100 images are simulated with a total of 2400 images). 1000 image pairs are randomly sampled from

the mentioned simulated images for training. The test set contains 70 image pairs and it has four different models. The test phantoms have inclusions that differ from training phantom in size, location and shape with average strain values between 1 to 2.5 %. We publicly release this dataset at data.sonography.ai.

2.1.4 Experimental Phantom and *In vivo* Data

Phantom data is collected at Concordia University's PERFORM Centre by an E-Cube R12 research ultrasound machine (Alpinion, Bothell, WA, USA) with a L3-12H linear array at the center frequency of 10 MHz and sampling frequency of 40 MHz. A tissue mimicking breast phantom made by Zerdine (Model 059, CIRS: Tissue Simulation & Phantom Technology, Norfolk, VA) is used which has tissue elasticity of $20 \pm 5kPa$ and contains hard inclusions with elasticity at least twice the elasticity of the tissue.

In vivo data was obtained at Johns Hopkins Hospital from a research Antares Siemens system using a VF 10-5 linear array with a center frequency of 6.67 MHz and a sampling frequency of 40 MHz. Data is collected from three patients in open-surgical RF thermal ablation for liver cancer. More experimental details of the procedure can be found in [28]. The study was approved by the institutional review board with consent of all patients.

2.1.5 Fine-Tuning of the Network

It is common to fine-tune a network that is already trained on a similar task, as opposed to training it from scratch, a process also known as transfer learning [48, 49]. Therefore, we use the FEM and Field II dataset to fine-tune the proposed networks, which are trained on computer vision data. We tested many settings and fine-tuning strategies and found out that only fine-tuning the final resolution pyramid suffices since the network already performs well and only small improvement to the displacement prediction is required. The fine-tuned sub-network is specified in Fig. 2.2. Data augmentation is performed by randomly mirroring in lateral direction and adding white Gaussian noise to the RF data. Subsequently, envelope and B-mode images are obtained and used as different input channels of CNNs.

Regarding the loss function selection, due to the fact that displacement error is small, MSE suppresses this small error and amplifies the outlier regions. In practice, we obtained noisier strain by MSE even though the displacement error was reduced (higher displacement variance with lower displacement error). Therefore, we use norm 0.4 similar to FlowNet2 small displacement network [39] as the main loss function since this norm amplifies small error and attenuates large errors obtained by outliers. Another important point is that total variation (TV) regularization similar to [29, 50] is used to reduce the displacement variations and improve the quality of the strain. The final loss function used for fine-tuning is:

$$loss = \|D_{GT} - D_{\Theta}\|_{0.4} + \frac{\lambda}{N} \|\Delta D_{\Theta} - \varepsilon\|_1 + \gamma \|\Theta\|_2 \tag{2.2}$$

where D_{GT} and D_{Θ} are the ground truth and estimated displacements, respectively and $\|.\|_p$ denotes norm p. ΔD_{Θ} is the axial derivative of the predicted axial displacement, N is the number of samples used for TV computation, and λ, γ are regularization weights. To avoid underestimation bias due to regularization, we regularize by average strain (ε) similar to [28, 29]. We fine-tune the weights of the final pyramid of RFMPWC-Net using this simulation dataset. We also fine-tune MPWC-Net, but do not report the results in this manuscript since fine-tuned RFMPWC-Net performed better than MPWC-Net. We set the weight decay to 0.01 and λ to 0.2. NVIDIA Titan V with 12 GB RAM is used for training and the image size is 2048×256, which enforces us to use batch size of 1 due to memory limits. The network is fine-tuned for 50 epochs and the learning rate is set to 2e-9.

2.2 Results

In this section, the proposed networks are evaluated and compared with existing methods. NCC [22], GLUE [27], FlowNet2 [34, 39], original PWC-Net [37] and our proposed networks (MPWC-Net, RFMPWC-Net and fine-tuned RFMPWC-Net) are evaluated for simulated phantoms, an experimental phantom and *in vivo* data. GLUE is a recent method that has already been extensively used in several challenging simulation, phantom and *in vivo* applications by different research groups [42, 51–53].

To make the comparison fair, the input of deep learning methods (PWC-Net, FlowNet, MPWC-Net and RFMPWC-Net) is the concatenation of B-mode, RF and envelope signals. We use the trained FlowNet2 and PWC-Net weights publicly available on the Pytorch framework [37]. The GLUE code is publicly available, and NCC implementation is similar to [29] where we perform 2D cubic interpolation to calculate subsample displacements. Substantially better results are expected with a multi-level stretching NCC technique. In the simulation experiments, the ground truth displacement is known. Therefore, Normalized Root Mean Squared Error (NRMSE) of axial displacement [54] defined in 2.3 is used as the metric for measuring the displacement prediction accuracy. The results are reported for two different Peak Signal to Noise Ratios (PSNR).

$$NRMSE(\%) = \sqrt{mean((\frac{D_{GT} - D_{\Theta}}{D_{GT}})^2)} \times 100 \tag{2.3}$$

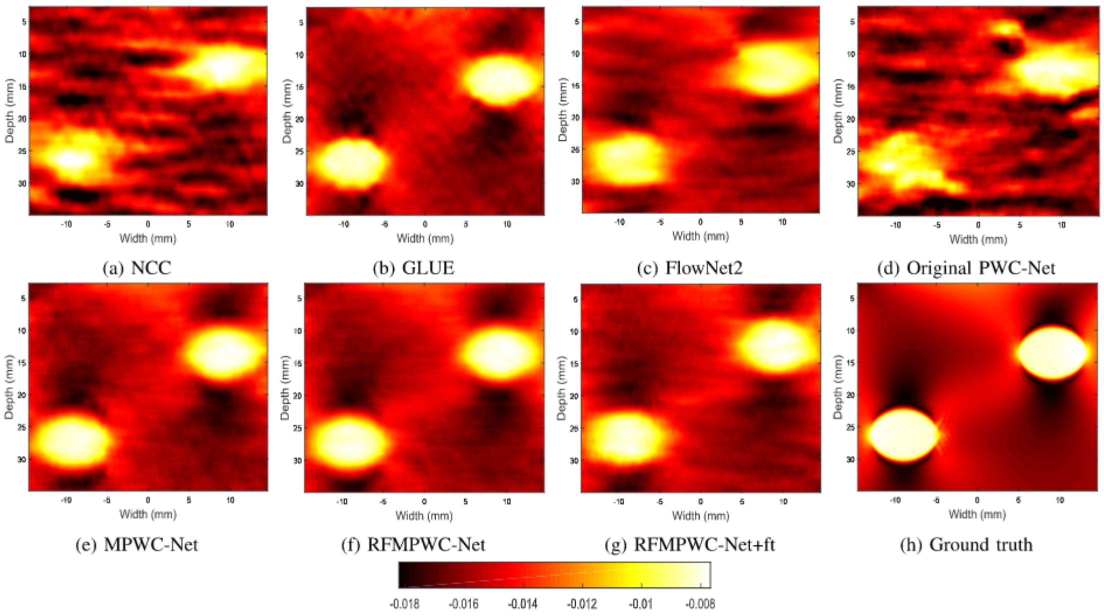
$$PSNR = 20 \times log_{10}(\frac{I_{max}}{\delta}) \tag{2.4}$$

where δ denotes standard deviation of noise and I_{max} is the maximum of image intensity. Noise with normal distribution is added to the RF data in order to obtain noisy simulation images. It should be noted that NRMSE is computed for each test phantom, then mean and standard deviation of NRMSE are reported for ideal and noisy simulated phantoms.

Two popular metrics, contrast to noise ratio (CNR) and strain ratio (SR) are also used to show the strain quality in the experimental and *in vivo* results, which are defined as [10]:

$$SR = \frac{\overline{s}_t}{\overline{s}_b}, \qquad CNR = \sqrt{\frac{2(\overline{s}_b - \overline{s}_t)^2}{{\sigma_b}^2 + {\sigma_t}^2}}, \tag{2.5}$$

where $\overline{s}_t$ and $\overline{s}_b$ are average values of strain in the target and background regions, and σ_t and σ_b are variance values of strain in the target and background regions, respectively. The selected regions in the target and background must be uniform and large enough to be statistically meaningful. It is important to note that CNR is sensitive to mean and variance of the regions. Whereas, SR only measures the differences in the mean value of the selected region. SR is a proper metric to measure the bias error of the strain. Whereas, CNR shows the combination of bias and variance error of the strain. One basic property of elastography methods is that they estimate lower difference between the tissue and the



(a) NCC (b) GLUE (c) FlowNet2 (d) Original PWC-Net

(e) MPWC-Net (f) RFMPWC-Net (g) RFMPWC-Net+ft (h) Ground truth

Figure 2.3: Strain images of a simulated phantom with $PSNR = \infty$.

inclusion due to bias created by different smoothing operations (continuity constraints in GLUE, median filtering or low-pass filtering in NCC and window-based methods, and least squares differentiation). Therefore, in real experiments with unknown ground truth, higher difference usually translates to smaller estimation bias. If a hard inclusion is chosen as the target, the value of SR is less than 1, where lower SR represents higher difference in the strain of the target and background (i.e. lower numbers are generally better). In order to compute reliable CNR and SR, large windows are selected in Fig. 6.2 (h), Fig. 2.6 (h) and Fig. 2.7 (h). The windows are divided into small overlapping patches. CNR and SR are computed for all combination of target and background patches. The mean and standard deviation of the computed CNRs and SRs are reported. To better visualize the results, we show strain images, which are the least squares derivatives of the axial displacement in axial direction.

2.2.1 Simulation Results

In this section the results of the simulated phantoms are presented for different methods. The strain image of a simulated phantom with the displacement calculated by the evaluated methods is depicted in Fig. 2.3 for PSNR= ∞. Our proposed networks perform substantially

Table 2.1: Comparison of different methods for 70 simulated phantoms.

	PSNR=∞	PSNR=30 dB
Method	NRMSE (%)	NRMSE (%)
NCC	1.88±0.51	1.93±0.53
GLUE	**1.10**±0.53	**1.10**±0.53
FlowNet2	1.65±0.46	1.68±0.46
PWC-Net	1.82±0.74	1.82±0.74
MPWC-Net	1.17±0.54	1.28±0.40
RFMPWC-Net	1.18±0.61	1.19±0.62
RFMPWC-Net+ft	1.15±**0.33**	1.18±**0.34**

better than stock deep learning methods in both simulation setups.

It is important to note that although the complexity of FlowNet2 is substantially more than the four other networks (both in training and inference), its results are substantially worse than our proposed networks. By closely inspecting the FlowNet2 results, it is evident that there is a substantial underestimation of strain in hard inclusions, which are not as dark as our proposed methods.

Another important point is that all networks except RFMPWC-Net+ft are trained on computer vision images and RFMPWC-Net+ft is fine-tuned by our dataset. Visually, the results of RFMPWC-Net+ft and RFMPWC-Net are close but RFMPWC-Net+ft is smoother. The quantitative results are given in Table 2.1 for 70 simulated phantoms. According to these results, the results of RFMPWC-Net are close to GLUE even without fine-tuning on ultrasound images, which shows the potential of the networks solely trained on computer vision images. Our fine-tuned variant of RFMPWC-Net performs slightly better than RFMPWC-Net. It is important to note that GLUE results remain very similar for no-noise and noisy conditions which indicates the robustness of GLUE due to optimizing all samples simultaneously.

RFMPWC-Net is more robust to noise compared to MPWC-Net which NRMSE increases 0.09 % in noisy conditions. In order to show the effect of fine-tuning, the strain of one line using RFMPWC-Net and RFMPWC-Net+ft is depicted in Fig. 2.4. As shown, the fine-tuned RFMPWC-Net result (red) has less variations and it is closer to ground truth compared to RFMPWC-Net (blue), which indicates that fine-tuning improves the displacement estimation accuracy. However, the improvements obtained by modifying the structure is more

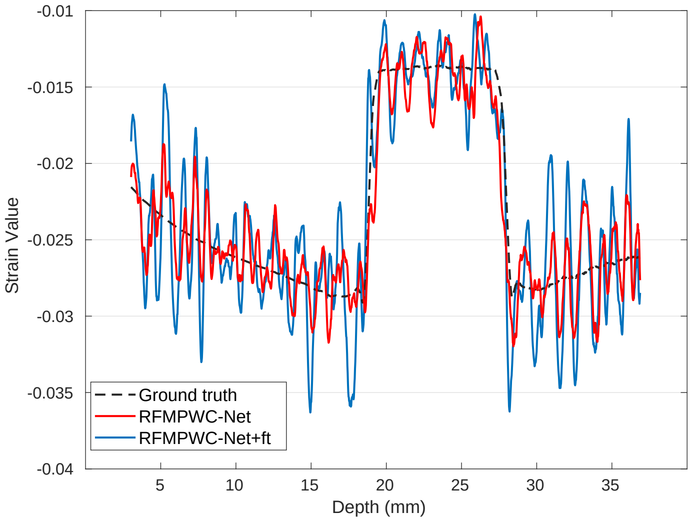


Figure 2.4: One line of strain using a small least square window. RFMPWC-Net (blue), RFMPWC-Net+ft (red) and ground truth (black).

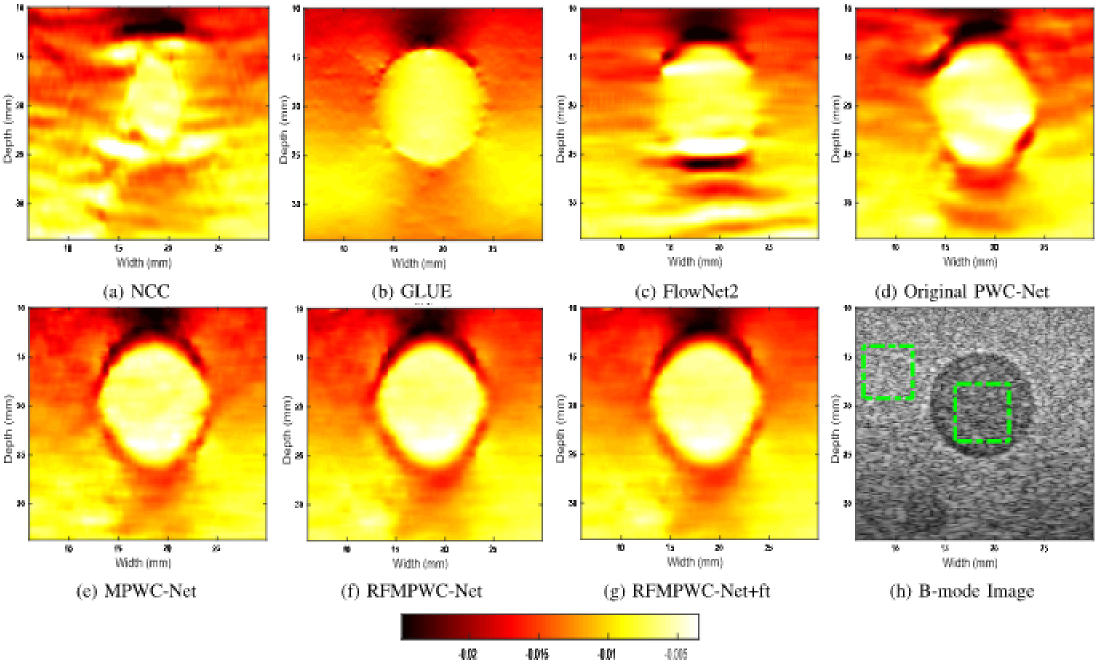
Table 2.2: SR and CNR of the experimental phantom.

Method	CNR	SR
NCC	10.65±2.69	0.399±0.04
GLUE	26.75±7.86	0.459±0.02
FlowNet2	20.19±3.70	0.48±0.02
PWC-Net	20.28±5.82	**0.376**±0.05
MPWC-Net	17.12±4.59	0.425±0.03
RFMPWC-Net	27.06±4.28	0.410±0.03
RFMPWC-Net+ft	**29.15**±5.77	0.382±0.05

tangible (compare Fig. 2.3(d), (e) and (f)) than fine-tuning. The main reason is that the networks trained on computer vision images are already performing well in mapping the inputs to the displacement. The modification of the structure brings substantial improvements to the network accuracy by providing more information to the network.

2.2.2 Experimental Phantom Results

CNR and SR defined in Eq 2.5 are used as quantitative metrics and the visual results are demonstrated in Fig. 6.2. NCC and FlowNet2 fail to obtain acceptable strain and GLUE produces smooth but underestimated strain, which is due to regularization. As such, GLUE result does not have as low SR as the deep learning methods. Nevertheless, GLUE has less variance, which makes the CNR very close to our proposed methods. The quantitative



(a) NCC (b) GLUE (c) FlowNet2 (d) Original PWC-Net

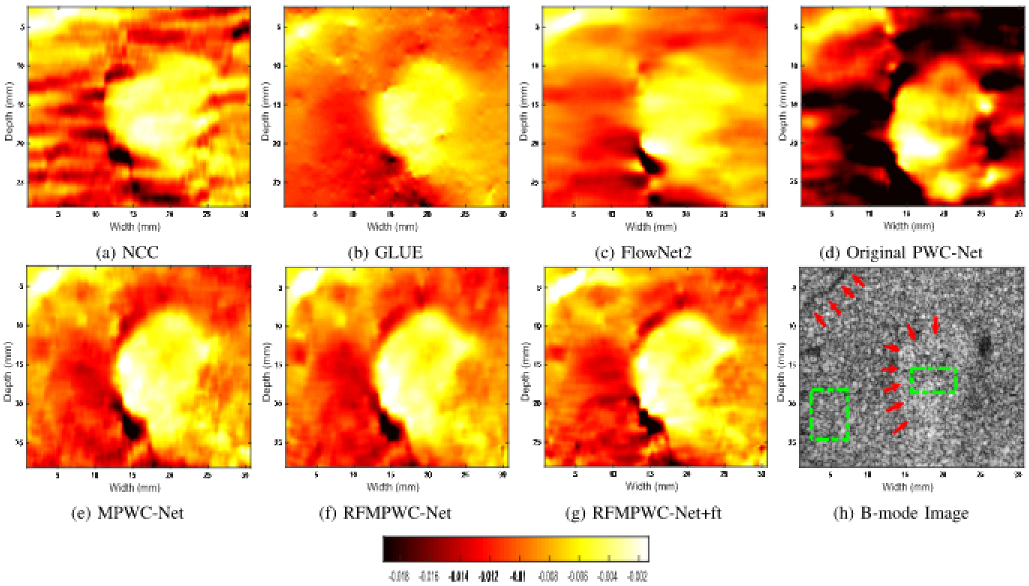
(e) MPWC-Net (f) RFMPWC-Net (g) RFMPWC-Net+ft (h) B-mode Image

Figure 2.5: Strain images of the experimental phantom. The windows used for CNR and SR computation are highlighted in the B-mode image (h). FlowNet2 (c) has high heterogeneity and fails to obtain smooth and high quality strain and the proposed networks have higher contrast compared to GLUE (b).

results in Table 6.1. confirm the visual assessments. GLUE has good CNR (26.75) but poor SR (0.459), whereas PWC-Net has the best SR (0.376) with a moderate CNR (20.28). RFMPWC-Net has higher CNR and better SR than MPWC-Net. RFMPWC-Net has higher CNR than GLUE (27.06 compared to 26.75) and better SR (0.41 compared to 0.459) without using any ultrasound images for training, which indicates the strength of the proposed CNN networks. RFMPWC-net+ft produces the most appealing result among deep learning methods and outperforms all evaluated methods in terms of CNR (29.15) and has good SR (0.382). This shows that fine-tuning of trained networks by ultrasound images has a positive impact on the performance of the network.

2.2.3 *In vivo* Results

Considering Fig. 2.6, GLUE estimates low-variance and high quality but blurry strain. The strain estimated by FlowNet2 is too smooth and many details are lost. PWC-Net also fails to estimate an acceptable strain. MPWC-Net has good strain quality but with a few artifacts,



(a) NCC (b) GLUE (c) FlowNet2 (d) Original PWC-Net

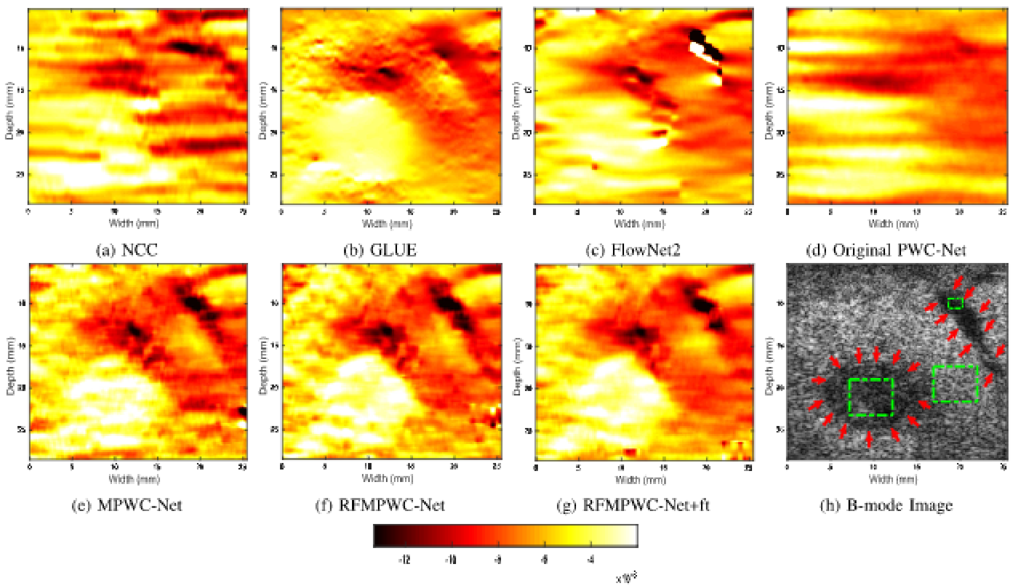
(e) MPWC-Net (f) RFMPWC-Net (g) RFMPWC-Net+ft (h) B-mode Image

Figure 2.6: *In vivo* strain results of the liver of patient 1 before ablation. The tumors are marked with arrows and the windows used for CNR and SR computation are highlighted in the B-mode image (h).

and RFMPWC-Net generates the best. RFMPWC-Net+ft further improves strain quality compared to RFMPWC-Net. Regarding Fig. 2.7, the GLUE result is acceptable but it is over smooth especially in in the top right of image. NCC, FlowNet2 and PWC-Net all fail to estimate strain, and MPWC-Net obtains a high-quality strain compared to PWC-Net. This indicates that our changes in the structure of PWC-Net have substantial impact on the network's performance. RFMPWC-Net has better strain compared to MPWC-Net and most of artifacts are removed in the RFMPWC-Net result. RFMPWC-Net+ft produces a

Table 2.3: Results of *In vivo* data, patient 1 (Fig. 2.6) and patient 2 (Fig. 2.7). GLUE has higher CNR for tumor and RFMPWC-Net results in higher CNR for the the vein. The proposed networks perform comparable to GLUE for *in vivo* data.

	Patient 1		Patient 2 (tumor)		Patient 2 (vein)	
Method	CNR	SR	CNR	SR	CNR	$\frac{1}{SR}$
NCC	9.08±3.22	0.29±0.07	3.60±1.41	0.51±0.09	11.84±6.66	0.590±0.14
GLUE	**19.36**±4.51	0.389±0.06	15.11±5.30	0.441±0.02	11.54±6.13	0.795±0.07
FlowNet2	12.86±0.46	0.463±0.049	9.40±2.04	0.415±0.05	fail	fail
PWC-Net	10.79±4.00	0.451±0.09	5.90± 2.45	0.587±0.09	9.19±4.34	0.835±0.07
MPWC-Net	12.11±3.75	**0.376**±0.07	11.66± 2.2	**0.338**±0.03	11.98±5.69	0.610±0.06
RFMPWC-Net	13.55±4.34	0.396±0.06	12.48±3.23	0.409±0.04	**19.88**±9.41	**0.590**±0.05
RFMPWC-Net+ft	16.63±5.53	0.380±0.05	**15.58**±2.58	0.395±0.04	12.52±3.71	0.601±0.08



(a) NCC (b) GLUE (c) FlowNet2 (d) Original PWC-Net

(e) MPWC-Net (f) RFMPWC-Net (g) RFMPWC-Net+ft (h) B-mode Image

Figure 2.7: *In vivo* strain results of the liver of patient 2 before ablation. The tumor and the vein are marked with arrows and the windows used for CNR and SR computation are highlighted in the B-mode image (h). the GLUE (b) obtains smooth but blurry strain especially close to the vein on the top right of the image. Fine-tuning reduces the artifacts presented in RFMPWC-Net (compare (f) and (g)).

very high-quality strain image and further removes the artifacts.

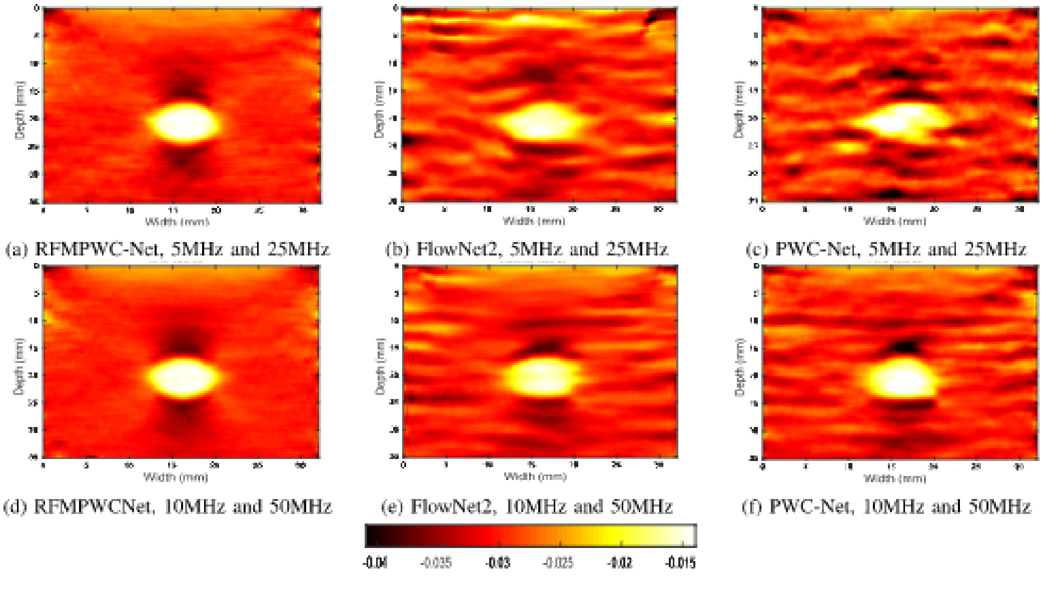
Considering the quantitative results of tumor presented in the first two columns of Table 2.3, GLUE obtains the high CNR in both patients (19.36 and 15.11) but the SR is poor (0.389 and 0.441). NCC and PWC-Net have poor CNR and FlowNet2 has higher CNR compared to them but visually the strain images are not acceptable. MPWC-Net has poor CNR (12.11 and 11.66) but produces the best SR (0.376 and 0.338). This implies that MPWC-Net has high variance in estimation which leads to low CNR but it has low bias in estimation which results in low SR. RFMPWC-Net outperforms MPWC-Net in terms of CNR with slightly worse SR. Fine-tuning improves the CNR with approximately similar SR. RFMPWC-Net+ft produces CNR values very close or even better than GLUE (16.63 and 15.58) with better SR (0.388 and 0.399).

By inspecting the results of the soft target (the vein in up right corner of Fig 2.7 (h)), it is inferred that our 3 networks substantially outperform GLUE in terms of both CNR and SR. RFMPWC-Net has the highest CNR (19.88) by a large margin, which is 8.34 dB and 7.36 dB better than GLUE and RFMPWC-Net+ft, respectively. The main reason that

RFMPWC-Net performs better without fine-tuning is that our database only contains hard inclusions and fine-tuning by this database deteriorates the cases with soft inclusions such as veins. For the vein, $\frac{1}{SR}$ is reported in order to be consistent with other results since the SR value for veins is more than 1. Our networks have the best SR among the compared methods and they have substantially better SR compared to GLUE.

2.2.4 Effect of sampling and center frequencies

The sampling and the center frequencies have critical role in displacement estimation accuracy. In the simulation results, the center and sampling frequency are 5 MHz and 50 MHz, respectively. We simulate a phantom with two different center and sampling frequencies. RFMPWC-Net, FlowNet2 and PWC-Net are tested for the center frequencies 5 and 10 MHz and the sampling frequencies 25 and 50 MHz. As shown in Fig. 2.8, strain obtained by RFMPWCNet (a, d) are high quality and consistent compared to FlowNet2 and PWC-Net.



(a) RFMPWC-Net, 5MHz and 25MHz

(b) FlowNet2, 5MHz and 25MHz

(c) PWC-Net, 5MHz and 25MHz

(d) RFMPWCNet, 10MHz and 50MHz

(e) FlowNet2, 10MHz and 50MHz

(f) PWC-Net, 10MHz and 50MHz

Figure 2.8: Simulation results of RFMPWC-Net, FlowNet2 and PWC-Net for different center and sampling frequencies. (Network, Center frequency and Sampling frequency). RFMPWC-Net quality remains well when sampling frequency decreased (a) or center frequency increased (d) in comparison to the other methods.

2.3 Discussions

In this chapter, two networks based on PWC-Net are proposed for USE. Generally, USE requires high accuracy subsample displacement estimation, which renders efficient use of high frequency information in RF data critical. This is a challenge as stock optical flow networks are not designed to handle RF data.

The PWC-Net is modified for USE displacement estimation by: 1) removing downsampling of the first feature extraction layer (this layer is connected to the input directly) to prevent loss of high frequency information; and 2) concatenating RF data, envelope and B-mode images to feed to the network. by doing this, the low-resolution pyramid levels exploit low-frequency B-mode and envelope information and high-resolution pyramid levels use RF data to obtain accurate displacement.

The main drawback of MPWC-Net is that B-mode and envelope contribute to the final resolution displacement estimation. B-mode and envelope are beneficial in low pyramid levels where RF data cannot be used, but they result in less accurate estimated displacement compared to RF data. Hence, in noisy conditions, MPWC results degrade considerably (as given in Table 2.1). RFMPWC-Net is proposed to resolve this problem by adding a separate sub-network to extract and use only RF data for the final pyramid level.

FlowNet2 network, which is extensively used by the researchers, obtains under-estimated strain and fails for *in vivo* data. Although FlowNet2 has 18 times more learning parameters than PWC-Net and achieves high accuracy in computer vision databases such as MPI-Sintel [39], it performs poorly on ultrasound images. This emphasizes that less complex pyramidal and warping structure is more suitable for ultrasound data.

Fine-tuning is another avenue that is investigated in this chapter, where the networks are tuned by simulated ultrasound images. In the loss function of fine-tuning, TV regularization is used to reduce the variance of displacement estimation. According to our results, fine-tuning improves the strain quality both qualitatively and quantitatively. All ultrasound simulation training data for fine-tuning the networks contain harder inclusions than the background. Nevertheless, the fine-tuned network performed well in a variety of *in vivo* experiments with different kinds of tissue. In the future, we plan to add data with soft inclusions in our training database and expect this to further improve the results. These new simulations will also strengthen the online database that we released.

Another important point about fine-tuning is that we only consider negative strain (post-compression image is the second image) for fine-tuning. The network can be fine-tuned with both positive and negative strain to be used for cases which post-compression image is not determined.

It is also worth mentioning that our proposed networks are very close to GLUE in terms of CNR and have better SR. By comparing the quantitative results presented in Table 6.1 and 2.3, it can be seen that GLUE has higher CNR than our proposed methods in the tumor part of patient 1. However, GLUE has lower CNR than the fine-tuned network for the experimental phantom data and data of patient 2. Another interesting conclusion is that RFMPWC-Net outperforms GLUE and fine-tuned network by a large margin for the vein (19.88 compared to 11.52 and 12.54). The reason for outperforming the fine-tuned network can be explained by the fact that we performed transfer learning using simulation data that only has hard inclusions.

In terms of SR, our proposed methods are the best among compared methods. MPWC-Net has the best SR but moderate CNR. In contrast, RFMPWC-Net and the fine-tuned variant of the network have higher CNR and slightly worse SR compared to MPWC-Net. The proposed methods perform similar to recent elastography methods without any need for parameter tuning. The proposed architectures have very small memory footprints and therefore can be implemented on inexpensive GPUs.

2.4 Summary

This chapter presents a deep learning approach for displacement estimation of the USE. The structure of PWC-Net is modified for our application. Visual and quantitative assessments of simulated phantoms, experimental phantom and *in vivo* data confirm that the proposed methods are suitable for USE and can compete with current state-of-the-art elastography methods.

Chapter 3

MPWC-Net++: Evolution of Optical Flow Pyramidal Convolutional Neural Network for Ultrasound Elastography

3.1 Introduction

The architecture of optical flow CNNs are optimized for computer vision images and there is a large domain gap between computer vision images and USE [55, 56]. To reduce the domain gap, many researchers tried to fine-tune the network by ultrasound images [34, 57]. However, their improvements were not significant. In Chapter 2 [55], we showed that the architecture has a larger impact compared to simple fine-tuning. We proposed that by modifying the structure of the network, substantial improvements can be achieved. We modified PWC-Net by removing the strides to allow the network to use RF data. We named our architecture as MPWC-Net.

MPWC-Net modification is straightforward and does not require re-training the network. However, this simplicity is not without any cost. In this chapter, we introduce these problems and try to mitigate them by some additional modifications. We named the new modified network MPWC-Net++, which is based on MPWC-Net with additional changes to facilitate use of optical flow CNNs in USE.

3.1.1 MPWC-Net

In USE, RF data which has high frequency content is required for accurate displacement estimation. However, the input images are downsampled by a factor of 4 at the first feature extraction blocks in PWC-Net. This leads to a loss of information and degrades the strain image quality. In Chapter 2 [55], we proposed to remove the first two downsampling by replacing stride=2 with stride=1 such that reliable feature can be extracted by the feature extractor blocks. Furthermore, we proposed to use B-mode and envelope alongside RF data to avoid unreliable displacement estimation in coarse levels where RF data does not provide reliable information.

Limitations of MPWC-Net

In PWC-Net, each pyramid level is responsible to estimate the displacement in its range and then remove this displacement by warping the feature map of the second image toward the first one. This facilitates the task of the following pyramid level since it requires only to estimate the residual displacement. This pyramidal structure also allows to limit the search range of the cost volumes to further reduce the computation complexity. Let c, M and S be the cost volume search range, the number of pyramid levels and rate of downsampling for the first pyramid level, respectively. The higher bound of the maximum displacement (D_m) that PWC-Net detects is:

$$D_m = \sum_{i=0}^{M-1} 2^i \times S \times c \tag{3.1}$$

In the original PWC-Net, $c = 4$, $M = 5$ and $S = 4$ which results in the maximum displacement of 496. In MPWC-Net, $S = 1$ hence the higher bound of the maximum displacement would be 124 pixels which limits the application of MPWC-Net to small displacements. This is a limiting factor especially or large images and unsupervised training where the ground truth and maximum displacement is unknown [56].

The optical flow networks are mostly trained on Flyingchair dataset and fine-tuned on other available datasets such as Sintel [37]. Flyingchair is a synthetic dataset produced for training the networks which its displacement range and distribution mimic Sintel dataset. The maximum displacement of Sintel is around 450, and the histogram is shown in Fig. 3.1.

We can see that only a few training data have displacements more than 310 and most of maximum displacements are below 200. The network only sees very limited training data that have displacements close to the higher bounds of PWC-Net (496 pixels). This means that in large displacements, the network cannot perform as well as smaller displacements. This is another factor that further limits the displacement range. The practical displacement range of PWC-Net and MPWC-Net are lower than 496 and 126, respectively.

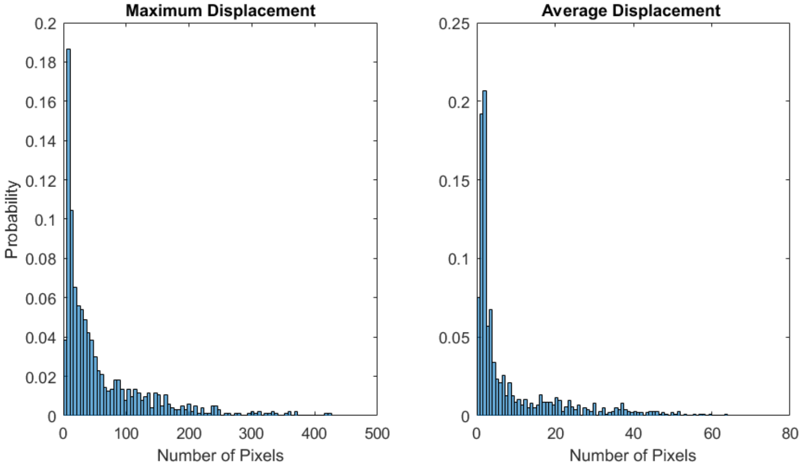


Figure 3.1: Histogram of maximum displacement and average displacement of Sintel training dataset.

Another limiting factor is the input image size. In original PWC-Net, the image is downsampled by 4 in the first two feature extraction convolution blocks. Whereas, in MPWC-Net the is no downsamplping; therefore, there are 16 times more pixels in train and inference. The much higher image and feature size results in significant reduction of speed in both training and test. The inference times are reported in section 3.2.2.

3.1.2 MPWC-Net++

In order to solve the aforementioned problems, we proposed to modify MPWC-Net structure. We used PWC-Net-irr instead of original PWC-Net structure to reduce the number of learning weights.

We also increased the maximum displacement range by changing values of c and S. Different values of these parameters are given in Table 3.1. The new structure must have a search

Table 3.1: Maximum search range (D_m) for different c and S.

Model	c	S	D_m
PWC-Net	4	4	496
MPWC-Net	4	1	126
case 1	4	2	252
case 2	5	1	155
case 3 (MPWC-Net++)	5	2	310
case 4	6	2	372

range close to the Sintel and Flyingchair since it would be trained by these datasets and if it cannot estimate large displacement values, the incorrect loss would be propagated during the training. As shown in Table 3.1, case 3 has a good balance between maximum displacement and the downsampling factor and can track a large portion of large displacements.

According to the selected value of S, we needed to decrease the S from 4 to 2. One of the first two feature extraction blocks needed to have a stride of 2 and the other had the stride of 1 ($S : 2 \times 1 = 2$). We selected the first feature extraction layer and the second one to have stride of 2 and 1, receptively. By this setting, smaller image size is fed to the second feature extraction layer and the computation time is improved. While, this leads to loss of information especially in axial direction which has a high frequency content. To avoid this loss of information, we also changed the kernel size of first feature extraction layer from 3×3 to 5×3 to capture more information in axial direction.

Another modification is that we replaced the input B-mode image with the imaginary part of the RF data which has been shown to improve the performance.

3.2 Results

3.2.1 Training and Results on Validation Set

Due to the change of parameters, we needed to train MPWC-Net++ from scratch. We followed the same training schedule of PWC-Net-irr [58]. The decrease of stride (S) and having higher search range (c) had substantial impact on training time; therefore, we decreased the

Table 3.2: The validation EPE of the networks trained on FlyingChairOcc.; FlowNet2 and PWC-Net are reported from [58].

Network	EPE of FlyingChairOcc
FlowNet2	2.39
PWC-Net	1.89
FlowNet2 (irr)	2.22
PWC-Net (irr)	1.83
MPWC-Net++ (irr)	1.51

number of epochs from 220 to 180. The network was trained using FlyingChairsOcc dataset [58]. The dataset is composed of 22870 image pairs with forward and backward ground truth flows. The displacement distribution is the same as original FlyingChair dataset. The network was trained for 10 days (due to larger search range and reduced stride it took longer time than PWC-Net-irr). We publicly available the trained network weights at data.sonography.ai.

We also investigated the error on the validation set of the mentioned dataset. We used end point error (EPE) which is defined as:

$$EPE = \sqrt{(\widetilde{U_x} - U_x)^2 + (\widetilde{V_x} - V_x)^2} \tag{3.2}$$

where $\widetilde{U_x}$, U_x, $\widetilde{V_x}$, V_x denote estimated and ground truth displacement in axial and lateral, receptively. The EPE of validation set of FlyingChairsOcc for different architecture are given in Table 3.2. Using the blocks iteratively (irr) results in a modest improvement in both FlowNet2 (2.22 *vs* 2.39) and PWC-Net (1.83 *vs* 1.89). MPWC-Net++ results in a substantial improvements compared to PWC-Net with only a few small modifications. After investigating the results, we found the the main reason of the improvement is that the stride (S) is reduced from 4 to 2. This leads to larger input images in each pyramid level and more accurate features can be extracted from the images even for computer vision images.

USE Results

The problem with MPWC-Net is that it has high inference time due to large size of image and cannot handle large displacements. We presents simulation and experimental phantom results for large and small displacements.

Simulation Results

A simulation data with displacement range of 0 to -42 samples is chosen to compare the methods. The results are shown in Fig 3.2. We can see that MPWC-Net has outlier regions where the displacement is high. It should be noted that this network fails in displacement values much lower than the computed higher bound (it fails around 35 which is much lower than 126). PWC-Net does not fail but the strain image quality is low due to a limited use of the high frequency information. MPWC-Net++ results in the strain image as good as MPWC-Net without any outlier region.

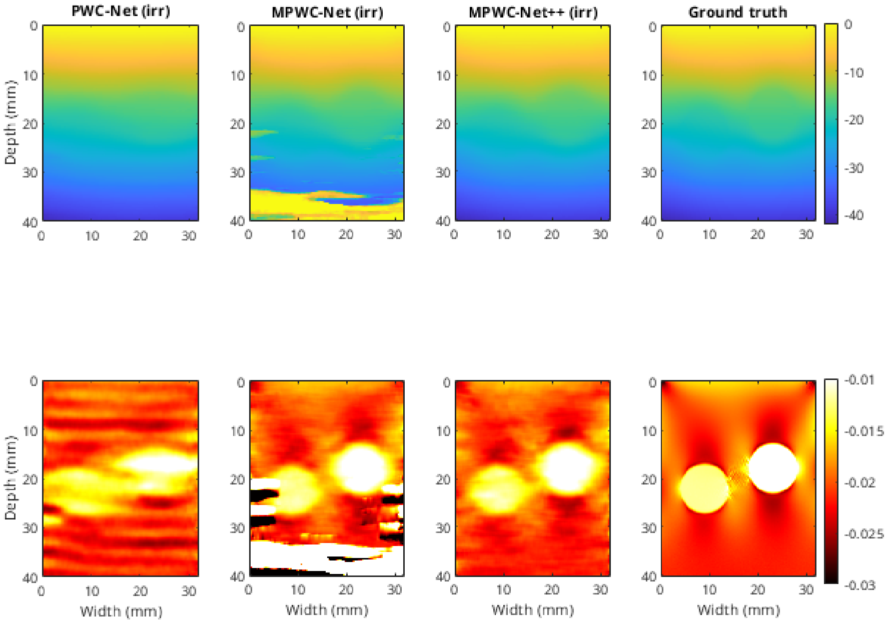


Figure 3.2: Axial displacement (top) and strain images (bottom) of MPWC-Net, PWC-Net, MPWC-Net++ and ground truth.

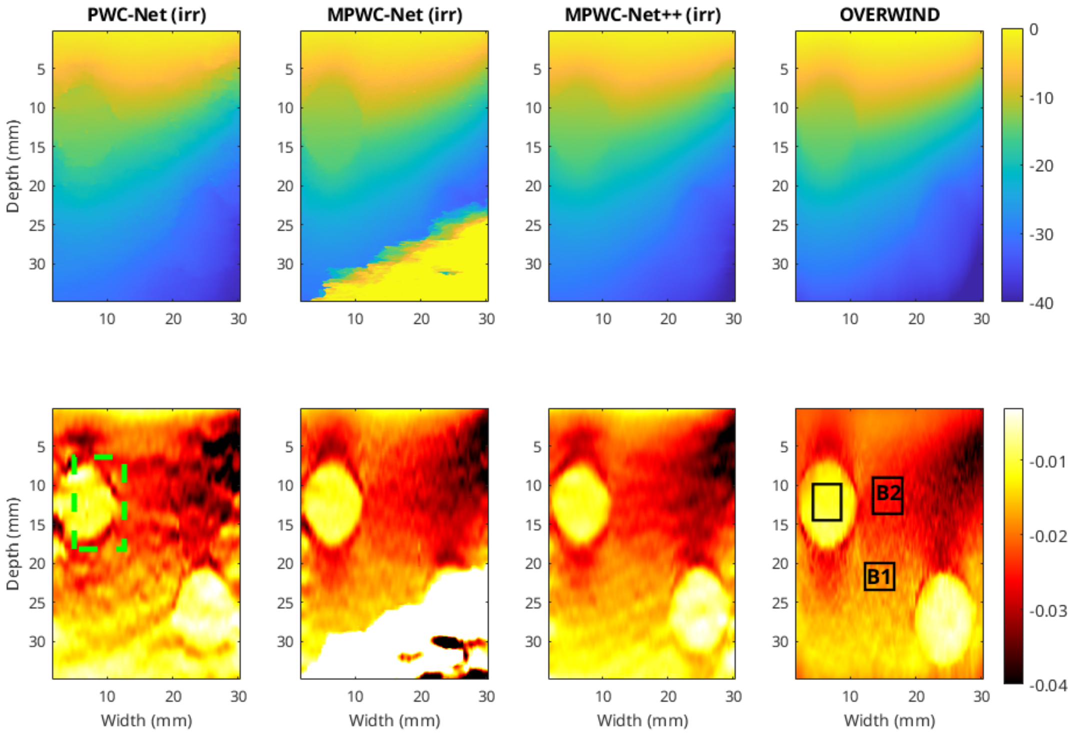


Figure 3.3: Axial displacement (top) and strain images (bottom) of PWC-Net, MPWC-Net, MPWC-Net++ and OVERWIND. Background 1 (B1), background 2 (B2) and the target regions are specified by black windows. The 3D view of the region highlighted by the green window is illustrated in Fig. 3.4.

Experimental Phantom Results

PWC-Net, MPWC-Net and MPWC-Net++ are compared with OVERWIND which is considered as the state-of-the-art method in USE. OVERWIND is an optimization-based method that has been proposed by Mirzaei *et al* [29]. The results are shown in Fig. 3.3. As can be seen from the figure, PWC-Net results in low-quality strain. MPWC-Net produces higher quality strain image compared to PWC-Net but it fails when the displacement is high (botton of the phantom). MPWC-Net++ produces high quality strain without failure in regions where the displacement is high. The strain quality of MPWC-Net++ is also comparable with OVERWIND.

The quantitative results are reported in Table 3.3. The regions for computation of CNR and SR are highlighted in Fig. 3.3 (black windows). We used smaller windows inside the specified regions to compute CNR and SR. The mean and standard deviation are reported.

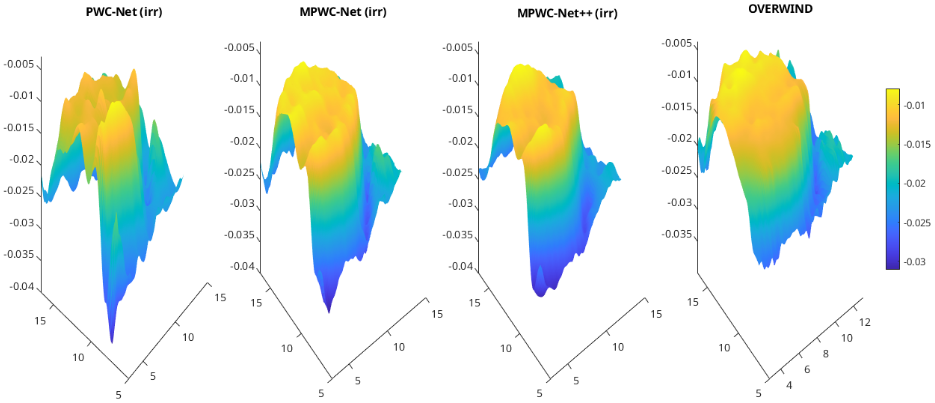


Figure 3.4: 3D side view of the strain of the region specified in the Fig. 3.3.

Table 3.3: Quantitative results of the experimental phantom comparing target with background 1 and background 2: mean±standard deviation.

Method	Background 1		Background 2	
	CNR	SR	CNR	SR
OVERWIND	**25.21**±4.82	0.413±0.019	12.86±1.85	0.639±0.022
PWC-Net	12.67±3.14	0.418±0.021	7.26±2.60	0.624±0.074
MPWC-Net	20.59±3.46	**0.392**±0.023	8.63±4.17	**0.526**±0.209
MPWC-Net++	24.99±2.67	0.425±0.014	**14.52**±2.72	0.632± 0.029

The CNR and SR are reported for one target region against two background regions. OVERWIND outperforms other methods in terms of CNR for background1 (comparing the target region with the background 1). MPWC-Net++ results in a CNR close to OVERWIND for background 1. Whereas, MPWC-Net++ has the highest CNR for background 2 (comparing the target region with the background 2). It is higher than OVERWIND by 1.66 db and it is higher than other networks by a large margin. It is evident from the results that MPWC-Net++ not only fixes the low displacement range of MPWC-Net, but also improves the CNR and the overall quality of the strain image considerably.

Regarding SR, the lower value indicates lower bias. MPWC-Net has the lowest SR value among the compared methods which represents that it has the lowest bias error. MPWC-

Net++ has SR values slightly higher than MPWC-Net but the SR values are very close to MPWC-Net.

The 3D view of the region specified in Fig. 3.3 by the green window is shown in Fig. 3.4. It can be seen from the figure that OVERWIND and MPWC-Net++ have low heterogeneity in the specified region.

Table 3.4: Inference time of different architectures in ms.

Method	Time in ms
PWC-Net	42.28
MPWC-Net	1409.40
MPWC-Net++	495.20

3.2.2 Inference Time

Inference time is another important factor that needs to be evaluated. The inference time is crucial for real-time applications. We excluded OVERWIND due to the fact that it runs on CPU, whereas the other methods run on GPU. The inference times of single image pair of size 2048 × 256 are reported in Table 3.4. MPWC-Net++ is three times faster than MPWC-Net due to the fact that $S = 2$ instead of $S = 1$.

3.3 Summary

In this chapter, we presented a modification to MPWC-Net based on physics of ultrasound data and distribution of the computer vision training data. These modifications mitigated the limitations of MPWC-Net in estimating large displacements in USE and improved the inference time which is crucial for real-time applications such as image-guided surgery.

Chapter 4

Bi-Directional Semi-Supervised Training of Convolutional Neural Networks for Ultrasound Elastography Displacement Estimation

4.1 Introduction

Deep learning-based methods, which have been recently proposed for USE, employ CNNs to obtain the displacement map. The first few works used the optical flow CNNs as black boxes for USE [34] or as the initial estimator for optimization-based methods instead of DP [32, 44]. However, the computer vision images and US data are vastly different and the CNN architectures used for the former are not optimized for high-frequency RF data. Motivated to address this issue, we modified the well-known PWC-Net architecture [37] to be adapted to USE, considering the physics of RF data [55] (Chapter 2). We called the network modified PWC-Net (MPWC-Net) and obtained substantially more accurate displacement compared to PWC-Net. In another work, we proposed MPWC-Net++ which was an improved version of

MPWC-Net with a higher search range and more accurate output displacement [59] (Chapter 3). These methods require a GPU to run efficiently, and can perform high frame-rate USE given the rapidly increasing computational power of GPUs. However, their main drawback is that they have a larger variance compared to conventional methods due to the fact that they are not regularized in contrast to conventional methods. Consequently, their strain images have lower overall quality compared to conventional methods [59].

Unsupervised training was another avenue that has been followed by the researchers. Delaunay *et al.* trained a U-Net using real US data [60] and developed a recurrent network to deal with a sequence of frames [61]. In [62], we used a light network, referred to as LiteFlowNet [63] and trained it in a semi-supervised fashion. We first used computer vision datasets with known ground truths to train the network using supervised techniques. In the next step, real US data was used to fine-tune the network using an unsupervised method. We substantially improved the strain image quality by using this technique without requiring a large amount of training data.

In this chapter, we follow the semi-supervised training approach. The overview of the method is depicted in Fig. 11.2. We first employ computer vision datasets to train the network in a supervised fashion. We use MPWC-Net++, which has shown high performance in USE. We then fine-tune the network by real US data, and extend our idea of semi-supervised method by proposing bi-directional unsupervised fine-tuning. We change the structure of MPWC-Net++ to estimate both forward and backward displacements, which is more efficient than running the network two times to estimate them. Consistency loss is also proposed which is obtained by comparing forward and backward strains. Furthermore, we shed light on the choice of weights for regularization by relating some of them to others. We demonstrate the high-performance of the proposed methods using experimental phantom and *in vivo* data.

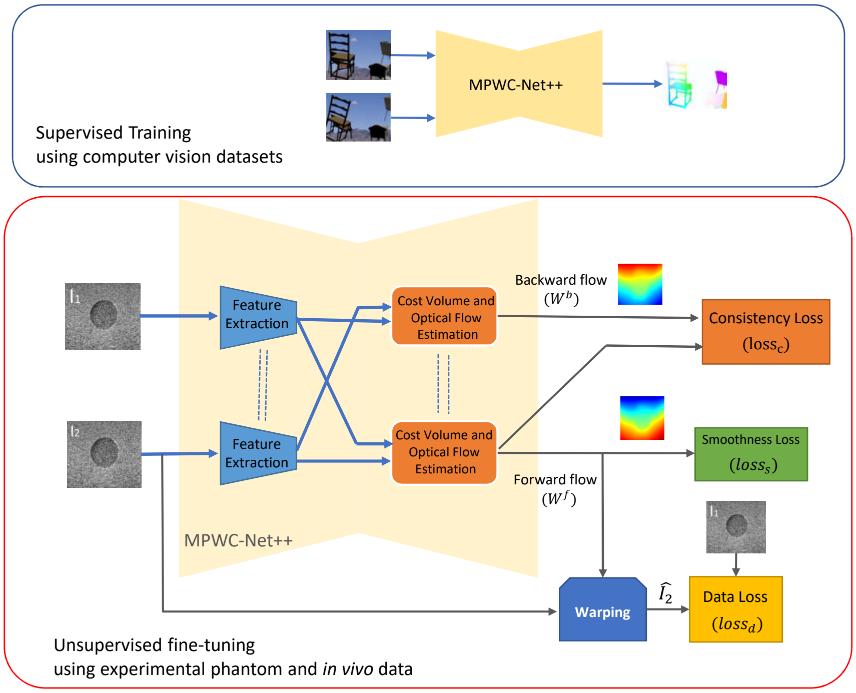


Figure 4.1: Overview of the proposed semi-supervised training method. The network is first trained using computer vision datasets by supervised methods (the block on top). The network is fine-tuned by real US data using the proposed unsupervised training method (bottom block). The network structure is altered to be able to estimate both forward and backward flows. The layers connected by the dashed lines share weights. The cost volume and optical flow estimation blocks with shared weights are used to estimate both forward and backward flows.

4.2 Material and Method

4.2.1 Deep Supervised CNNs

In this section, several CNNs used in USE displacement estimation are explained.

PWC-Net

Sun *et al.* proposed using cost volume and warping of the features for optical flow estimation [37]. They achieved state-of-the-art performance in different computer vision datasets. PWC-Net used a pyramidal structure, in which the optical flow was estimated in different resolution levels. At each pyramid level, the features of the second image were warped by the estimated flow of the previous pyramid to reduce the flow in the next pyramid level. In the next step, cost volume was employed to compare features of the fixed and warped moved images. In the last step, the optical flow was estimated and used in the next pyramid. Using pyramid structure resulted in the reduction of the number of learnable weights and improved the performance of optical flow estimation [37]. Recently, a variant of PWC-Net called Iterative Residual Refinement PWC-Net (IRR-PWC-Net) was proposed [58]. This network reduced the number of learnable weights even further by using the optical flow estimation block iteratively.

MPWC-Net

PWC-Net was proposed for computer vision images originally, and was not well suited to the high-frequency RF data. We modified the structure of PWC-Net and proposed MPWC-Net for. We removed the strides of the first feature extraction layer to preserve high-frequency information in RF data. In order to avoid failure of the network in low pyramid levels, where RF data does not have enough information due to downsampling, we used envelope and B-mode images as additional input channels [55]. We obtained competitive performance with conventional optimization-based methods. The code and the simulation dataset for fine-tuning are available online at code.sonography.ai.

MPWC-Net++

MPWC-Net had a low displacement range since strides of the first feature extraction layer were removed. In addition, we showed that the real displacement range is much lower than the theoretical one since only a small quantity of training data have high displacement ranges. Therefore, the network is not trained enough to deal well with large displacements, and the predicted flows are noisy in this condition. To address these problems, instead of removing both strides, we only removed one of them and kept the other one. Also, the search range of the cost volume was increased from 4 to 5 in each pyramid level. By doing these modifications, the network had higher search range and could work better for large displacements. Furthermore, we used IRR-PWC-Net [58] since it has a more efficient structure. Although we applied these modifications to improve the performance in the USE application, they also led to an improvement even for the computer vision dataset. [59].

However, the improvements came with costs. Due to modifying the network structure, we had to train the network from scratch (MPWC-Net did not require training from scratch). The training itself was slower compared to the original PWC-Net for two reasons: first, having larger feature maps after removing strides; second, increasing the cost volume search range. The MPWC-Net++ was trained from scratch using computer vision images while LiteFlowNet used in [56] was pre-trained and we only fine-tuned it using real US data. Training from scratch takes 840 hours using an NVIDIA Tesla P6 GPU, substantially more time compared to the fine-tuning since the network is randomly initialized and a large dataset (for MPWC-Net++ 22000 pairs) is employed to train the network. The network weights are publicly available online at code.sonography.ai.

4.2.2 Semi-supervised Method

Simulation data does not model non-linear or multiple scattering effects present in real US data [56, 60]. Therefore, we proposed to use real US data for fine-tuning. This method was semi-supervised since we first used a pre-trained network trained on computer vision images by supervised methods. In the next step, real US data were used to fine-tune the network.

The moved image was warped with the forward flow and compared with the first image which is called photometric loss. This loss alone resulted in noisy displacements; therefore, inspired by the physics of RF data, we proposed using the first second-order derivatives of displacement in the axial and lateral direction as the regularization.

In order to preserve the information of high-frequency RF data, we were not allowed to downsample images and had to use large image sizes during unsupervised fine-tuning which is challenging due to GPU memory limitations. We used a light network (LiteFlowNet [63]) and gradient checkpointing [64] to be able to train the network on our GPU (Nvidia TITAN V with 12 GB of RAM). Furthermore, we limited the training to only forward flow and the backward flow was used to detect occluded regions.

4.2.3 Proposed Method

Let $I_1, I_2 \in \mathbb{R}^{3\times W\times H}$ denote the fixed and moved images having 3 channels with width W and height H, $W^f \in \mathbb{R}^{2\times W\times H}$ and $W^b \in \mathbb{R}^{2\times W\times H}$ denote forward flow ($I_1 \rightarrow I_2$) and backward flow ($I_2 \rightarrow I_1$), respectively. The data loss function for unsupervised fine-tuning can be defined as [59]:

$$loss_d = \Phi(I_1 - \hat{I_2})_{w\times w} \tag{4.1}$$

where $\hat{I_2}$ is the second image warped by the W^f and unlike [56], a window of size $(w \times w)$ around the sample of interest is selected to compute the loss to reduce the noise caused by interpolation step of warping operation (here we use a 3×3 window). Φ is the Charbonnier loss that has been widely used in unsupervised optical flow training [65] and defined in Eq 4.2 [66].

$$\Phi(x) = |(x^2 + \varepsilon)^{\alpha}|_1 \tag{4.2}$$

where $|.|_1$ denotes the L1 norm and α can be altered to give different importance to x. We used $\alpha = 0.5$ for the data loss (would be L1 norm) and $\alpha = 0.2$ for smoothness and consistency loss to emphasize small values of x. It should be mentioned that inspired by [62], RF data, the envelope and imaginary part of Hilbert transform of RF data are utilized as 3 separate channels of input images.

In order to have a smooth displacement field, the first-order derivatives of the displacements in axial and lateral directions are used for regularization:

$$\begin{gathered} loss_s^1 = \lambda_{11}\Phi\ (\frac{\partial W_a^f}{\partial a} - <\frac{\partial}{\partial a}W_a^f>) + \lambda_{12}\Phi\ (\frac{\partial W_a^f}{\partial l}) + \\ \lambda_{21}\Phi\ (\frac{\partial W_l^f}{\partial a}) + \lambda_{22}\Phi\ (\frac{\partial W_l^f}{\partial l}) \end{gathered} \tag{4.3}$$

where W_a^f, W_l^f, $\frac{1}{\partial a}$, $\frac{1}{\partial l}$ and λ denote axial, lateral displacements, the derivative in axial and lateral directions and their corresponding weights, respectively. The axial derivative of the axial displacement is subtracted by its mean ($< . >$ denotes the mean value) to reduce the bias of the regularization similar to [29].

The second-order derivatives of the displacements have been found useful for USE [56, 60, 67]. Hence, they can be used to regularize the displacements:

$$\begin{aligned} loss_s^2 = \lambda_{31}\Phi \ (\frac{\partial^2 W_a^f}{\partial^2 a}) + \lambda_{32}\Phi \ (\frac{\partial^2 W_a^f}{\partial a \partial l}) + \\ \lambda_{41}\Phi \ (\frac{\partial^2 W_l^f}{\partial l \partial a}) + \lambda_{42}\Phi \ (\frac{\partial^2 W_l^f}{\partial^2 l}) \end{aligned} \tag{4.4}$$

Unlike [61] that used the first and second-order derivatives of only axial displacement, we used the first and second-order derivatives of both axial and lateral displacement in both directions. The second-order derivatives do not introduce bias but they require higher weights to be as effective as the first-order derivatives.

Hyper-Parameter Tuning

It can be seen that there are eight hyper parameters that we need to set before training the network. In our recent work [56], we set them empirically, while in this chapter, we tried to reduce the number of hyper-parameters by relating some of the them to others using US principles.

The distance between two adjacent samples in the axial and lateral direction is also vastly different since the sampling frequencies and the number of samples is widely dissimilar. The distance between two adjacent samples in the axial direction can be obtained by $c/(2f_s)$, where c denotes sound speed and f_s is the sampling frequency. A rough approximation of the lateral distance between two samples would be the distance between two adjacent A-lines which is much larger than the axial distance in a typical US image. Therefore, λ_{12} and λ_{22} must be several times smaller than λ_{11} and λ_{21}.

As noted in [67], the second-order derivatives are much smaller than the first-order ones. Therefore, to be as effective as the first-order derivatives, their weight should be several times larger than the first-order derivatives. We set this weight to be $\lambda_{31} = 5\lambda_{11}$. We also set the lateral derivative weights similar to the first-order derivatives ($\lambda_{41} = \beta\lambda_{31}$). Finally, the smoothness regularizer can be written as:

$$
\begin{aligned}
loss_s = \Phi\ (\frac{\partial W_a^f}{\partial a} - < \frac{\partial}{\partial a} W_a^f >) + \beta\Phi\ (\frac{\partial W_a^f}{\partial l}) + \\
0.5\Phi\ (\frac{\partial W_l^f}{\partial a}) + 0.5\beta\Phi\ (\frac{\partial W_l^f}{\partial l}) + \\
5\Bigg\{\Phi\ (\frac{\partial^2 W_a^f}{\partial^2 a}) + \beta\Phi\ (\frac{\partial^2 W_a^f}{\partial a \partial l}) + \\
0.5\Phi\ (\frac{\partial^2 W_l^f}{\partial l \partial a}) + 0.5\beta\Phi\ (\frac{\partial^2 W_l^f}{\partial^2 l})\Bigg\}
\end{aligned}
\tag{4.5}
$$

where β depends on the ratio of the sampling frequency in axial and lateral directions. Setting the weights does not require an exact calculation of the sampling frequencies and should be less than 1; we set this hyper parameter to 0.1. Finally, for incompressible materials, the Poisson's ratio is approximately 0.5 [68], which means that the strain in the lateral direction is half of the axial one. Therefore, λ_{21} can be substituted by $0.5\lambda_{11}$. It should be noted that the explained method to tune the weights of the regularizers is only a rough estimate of the optimal values; therefore, the training is not too sensitive to the variations of these weights and even changing the weights by as much as 100% yields similar results.

Bi-directional Strain Consistency

In this chapter, inspired by recent unsupervised methods in optical flow estimation [69], we proposed to utilize forward and backward consistency in addition to the data and smoothness losses. In unsupervised optical flow methods, the difference between forward and backward displacement was used for consistency loss [69].

Strain images are often showed in USE as a surrogate of the elastic modulus. Therefore, it would be useful to utilize the derivatives of the displacements for the consistency loss. Assuming a uniform tissue, the estimated forward and backward axial strain can be written as:

$$
\begin{aligned}
\varepsilon^f &= \varepsilon_{gt} + \mathcal{N}(\mu,\, \sigma^2) \\
\varepsilon^b &= -\varepsilon_{gt} + \mathcal{N}(-\mu,\, \sigma^2)
\end{aligned}
\tag{4.6}
$$

where we assumed that the strain true value is ε_{gt} and the error is modeled by a normal distribution with the bias and variance of μ and σ^2, respectively. The forward and backward consistency loss can be defined as:

$$loss_c = \Phi(\varepsilon^f + \varepsilon^b) \tag{4.7}$$

Substituting Eq 4.6 into Eq 4.7, yields:

$$loss_c = \Phi(\mathcal{N}(0,\, \sigma^2/2)) \tag{4.8}$$

This equation indicates that minimizing the $loss_c$, results in reducing the variance in estimation of forward and backward strains. Similar to the smoothness loss ($loss_s$), the strain in both axial and lateral directions were employed which can be written as:

$$loss_c = \Phi\ (\frac{\partial W_a^f}{\partial a} + \frac{\partial W_a^b}{\partial a}) + 0.5\beta\Phi\ (\frac{\partial W_l^f}{\partial l} + \frac{\partial W_l^b}{\partial l}) \tag{4.9}$$

where we used the same weights of the smoothness loss.

By using the loss functions defined in Eq 5.1,5.3, and 4.9, the total loss function can be written as:

$$loss = loss_d + \lambda loss_s + \gamma loss_c \tag{4.10}$$

Thanks to reducing the number of hyper-parameters, only λ and γ should be tuned for the training which can be done based on the training data. Too large values of λ and γ lead to a blurry strain image, while too small values result in noisy strain images. We set these hyper-parameters ($\lambda = 0.03$ and $\gamma = 0.05$) by visually inspecting the strain images of the validation set after training with different values of λ and γ. It is worth mentioning that similar data and smoothness loss can be used for the backward flow. However, the consistency loss that we added has a similar behavior since it tries to make the backward strain close to the inverse of the forward one.

Adding the consistency loss demands high memory since both forward and backward flows are used for backpropagation. We utilized an NVIDIA A100 GPU with 40 GB of RAM to be able to train using the proposed loss function and simultaneously avoid downsampling to preserve RF data high frequency information. Since training this network might be infeasible for some researchers, we will make the network's weights publicly available online at code.sonography.ai after acceptance of this manuscript.

4.2.4 Data Collection

Experimental Phantom

We used a tissue-mimicking breast phantom made from Zerdine (Model 059, CIRS: Tissue Simulation & Phantom Technology, Norfolk, VA) for data collection. The phantom contained a number of hard inclusions. The background has elastic modulus of 20 kPa and the inclusions have at least twice elastic modulus of the background. This phantom was utilized to obtain training and test data. We made sure that different parts of the phantom were imaged for training and testing to avoid data leakage.

We employed Alpinion E-Cube R12 research US machine (Bothell, WA, USA) for training and test. The L3-12H linear array probe with the center frequency of 10 MHz and the sampling frequency of 40 MHz was utilized for image acquisition.

In vivo **data**

In vivo data was collected at Johns Hopkins Hospital using a research Antares Siemens system by a VF 10-5 linear array. The sampling frequency was 40 MHz and the center frequency was 6.67 MHz. Data was collected from patients with liver cancer during open-surgical RF thermal ablation. For more information, please see [28]. The institutional review board approved the study with the consent of the patients.

4.2.5 Training schedule

We first collected data using the Alpinion machine from the breast phantom. We then selected image pairs having maximum axial displacement larger than 3 pixels. In total, 2200 image pairs were used for training. The networks were trained for 40 epochs and the learning rate was set to 30e-6 which reduced by 1/2 every 10 epochs. For *in vivo* data results, we also fine-tuned the network trained by the bi-directional method using *in vivo* data. This network was fine-tuned using 500 *in vivo* image pairs for 20 epochs and the learning rate was 20e-6 and reduced by a factor of 2 every 5 epochs. In our experiments, we named this network as Bi-directional Unsupervised+ft.

4.2.6 Compared methods

The compared methods are listed below.

1) OVERWIND is an optimization-based method that estimates sub-pixel displacement. This method requires the initial displacement which is obtained by DP[28]. OVERWIND considers a window around each sample and uses total variation for the regularization. This method obtains high quality strain images [29].

2) The recently proposed network, MPWC-Net++ which is the modified version of MPWC-Net. This network is only trained on computer vision images and no training on US data is done [59].

3) We fine-tune MPWC-Net++ using the unsupervised technique without the consistency loss. Also similar to [56], it has the second-order derivatives only in the direction of the displacement ($\lambda_{32}, \lambda_{41} = 0$). In this case, the unsupervised fine-tuning would be similar to the semi-supervised method [56] with some minor improvements. The difference is that a better network (MPWC-Net++) with more suitable regularization weights is employed.

4) Our proposed bi-directional unsupervised fine-tuning method. In this method, the consistency loss is added to the unsupervised loss function and all the second-order derivatives are employed in the smoothness loss.

5) For *in vivo* data section, we also fine-tune the bi-directional network using *in vivo* data.

We compare our bi-directional semi-supervised method with recent methods in USE: OVERWIND is a high-performance and non-deep learning method. MPWC-Net++ is one of the best networks used for USE without training on US data. The unsupervised variant of MPWC-Net++ combines this high-performance network with the unsupervised fine-tuning [56].

Table 4.1: CNR results (higher is better). The bold font highlights the best, and the underline indicates the best deep learning results. Numbers marked with asterisks indicate results that are not statistically significant (p-value> 0.01), e.g. OVERWIND and the proposed method in Fig. 4.2.

	Fig. 4.2	Fig. 4.3	Fig. 4.4	Fig. 4.5 (1)	Fig. 4.5 (2)
OVERWIND	27.26±4.27*	**25.28±6.62**	12.40±2.38	**12.83±5.35***	**27.84±8.82**
MPWC-Net++	12.02±1.59	8.74±1.85	5.12±1.11	4.83±2.09	7.82±3.66
Unsupervised	**31.78±7.47**	15.12±4.12	9.94±1.64	8.67±4.09	14.37±4.27
Bi-directional Unsupervised	27.71±5.20*	17.19±4.45	**13.84±4.49**	12.82±4.83*	21.40±3.69

Table 4.2: SR(%) results (lower is better). The bold font highlights the best, and the underline indicates the best deep learning results. Numbers marked with asterisks and star indicate results that are not statistically significant (p-value> 0.01).

	Fig. 4.2	Fig. 4.3	Fig. 4.4	Fig. 4.5 (1)	Fig. 4.5 (2)
OVERWIND	62.26±0.71	40.21±3.2	50.04±3.08*	61.78±5.74*	36.56±2.00
MPWC-Net++	61.56±1.70*	35.08±6.26	49.29±4.41*	69.43±8.46	40.20±10.3
Unsupervised	61.27±0.67*	**25.89±3.97**	48.15±6.16**	60.96±9.67*	32.06±7.47
Bi-directional Unsupervised	**59.24±0.50**	28.12±4.12	**45.82±2.53**	**58.79±5.72**	**30.32±2.90**

4.3 Results

4.3.1 Experimental Phantom Results

The results of different parts and compression levels of the experimental phantom are shown in Figs. 4.2,4.3,4.4 and 4.5. In Fig. 4.2, the inclusions are not visible in the B-mode images, while they can be detected by USE methods. Comparing deep learning methods, unsupervised training substantially improves the strain quality of MPWC-Net++. Our proposed bi-directional method obtains similar or higher quality strain images compared to the unsupervised method, and substantially better results in all experiments compared to MPWC-Net++, especially when the compression is low (for example, Fig. 4.4 and 4.5). It can be seen that for those images MPWC-Net++ does not provide a clear image of the inclusion, while the proposed method obtains the highest quality strain images among the compared deep learning methods. OVERWIND obtains high-quality strain images and the proposed method performs comparably to OVERWIND in terms of the strain quality.

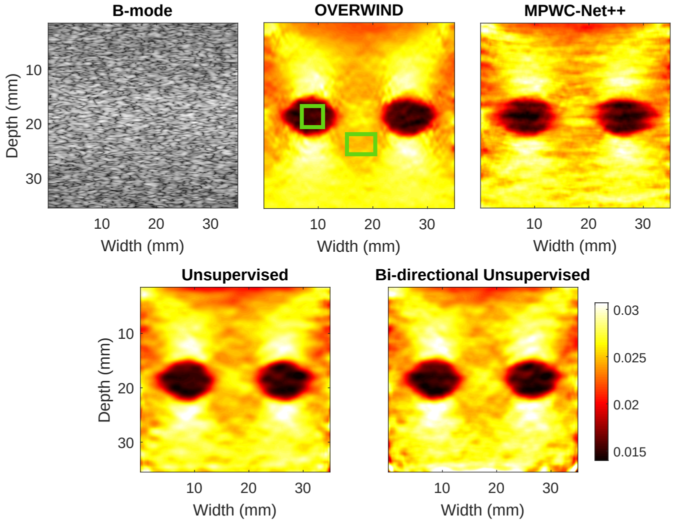


Figure 4.2: Phantom Result 1 with the maximum strain value of 3%. Green boxes indicate windows for computing CNR and SR.

The quantitative results are listed in Table 4.1 (for CNR) and 4.2 (for SR). In terms of CNR, our proposed method substantially increases the CNR of MPWC-Net++ and outperforms the unsupervised method in most cases. To be more specific, the bi-directional unsupervised method increases the CNR of MPWC-Net++ from 12.02, 8.74, 5.12, 3.73 and 10.25 to 27.71, 17.19, 13.84, 12.82, and 21.40, respectively. It also obtains CNR values close to OVERWIND or even better (in Fig. 4.4).

The SR results agree with [55] where we showed that MPWC-Net has better SR compared to another optimization-based method. In most cases (except Fig. 4.3)), bi-directional unsupervised method has the better SR and close to the lowest ones in that case.

The visual and quantitative results of the experimental phantoms confirm that our proposed method improves the overall quality of the strain images.

Smoothing window effect on strain image

After displacement estimation, a smoothing window along with the derivative kernel are used to reduce the error and compute the derivative of the displacement. Larger windows smooth the displacement more but sacrifice the resolution of the strain image. Therefore, displacement estimation methods that require smaller windows are preferred. When a USE displacement estimation method does not require a large smoothing window, it shows that

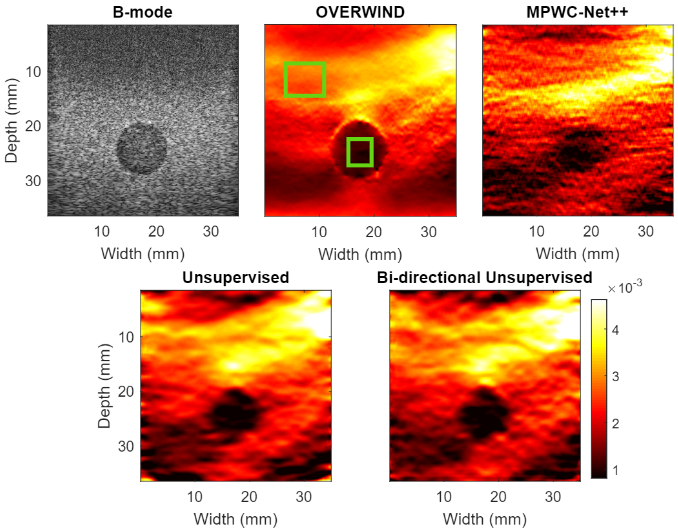


Figure 4.3: Phantom Result 2 with the maximum strain value of 0.4%. Green boxes indicate windows for computing CNR and SR.

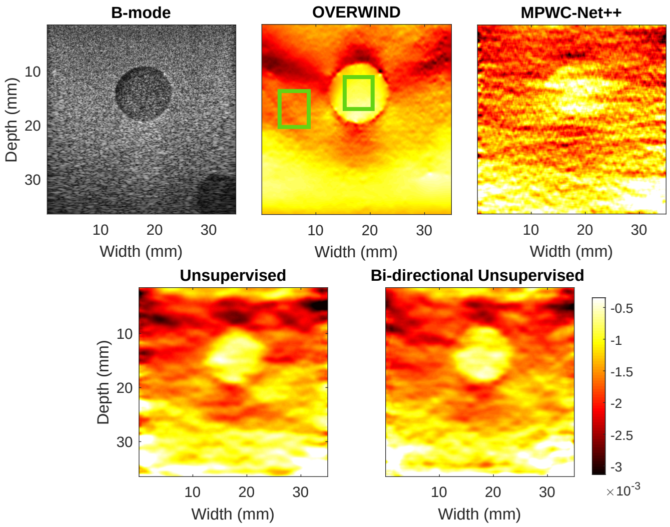


Figure 4.4: Phantom Result 3 with the maximum strain value of 0.3%. Green boxes indicate windows for computing CNR and SR.

the method produces a displacement map with a low variance error. To compare the methods, we compute the strain image of two image pairs with smoothing windows of sizes 5, 15, 30, and 40. The CNR values of different smoothing window lengths are shown in Fig. 4.6. It can be seen that OVERWIND has high CNR values even when the smallest smoothing window is employed. Unsupervised and bi-directional unsupervised methods have good CNR values close to that of OVERWIND. It should be mentioned that the difference between the

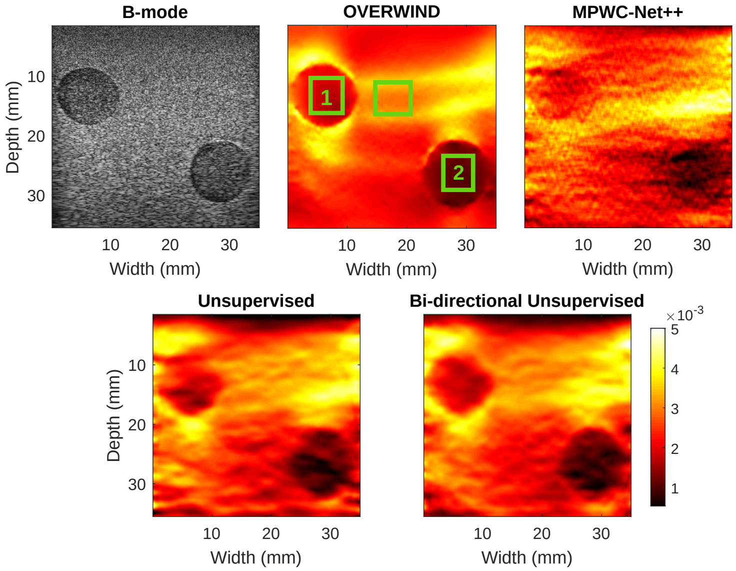


Figure 4.5: Phantom Result 4 with two inclusions having different elasticity with the maximum strain value of 0.5%. Green boxes indicate windows for computing CNR and SR.

unsupervised and bi-directional methods in Fig. 4.6 was not statistically significant (p-value = 0.112). MPWC-Net++ has very low CNR when the smoothing window is small. It indicates that this method is highly sensitive to the length of smoothing window and requires larger ones to produce acceptable strain images, whereas OVERWIND and the two unsupervised methods do not need a large smoothing window to produce reliable strain images and have low variance errors. The strain images are shown in Fig. 4.7 for smoothing windows of 5 (top), 15 (middle) and 30 (bottom), and the target and background windows for computation of CNR are highlighted. We can see that MPWC-Net++ generates noisy strain images for small smoothing windows, where the inclusion is not visually detectable. However, both unsupervised fine-tuning methods provide a better performance, close to OVERWIND, and generate less variations compared to MPWC-Net++.

4.3.2 Lateral Strain

The lateral strain ($\frac{\partial W_l}{\partial l}$) has much lower quality than the axial strain since the main movement is in the axial direction, and the lateral sampling frequency and resolution are low. The lateral strain can be utilized in inverse problem methods to find the elastic modulus [70]. Fig. 4.8 shows the lateral strain obtained by the compared methods. It can be seen that MPWC-

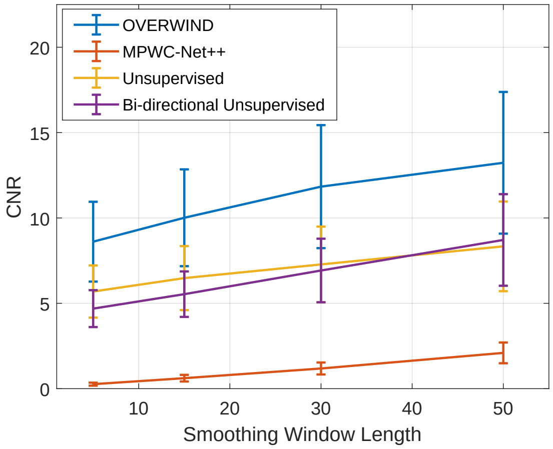


Figure 4.6: CNR values of the compared method using different smoothing window lengths. The strain images are shown in Fig. 4.7.

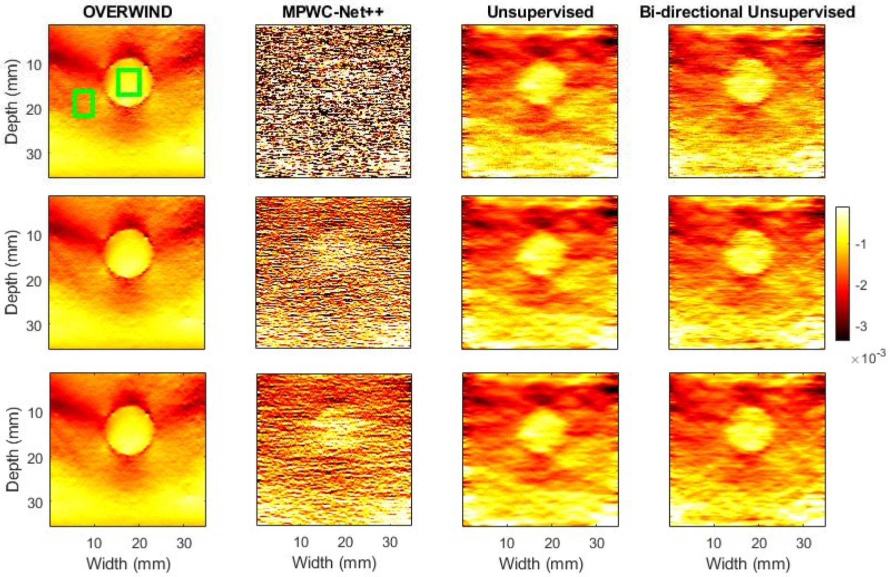


Figure 4.7: The strain images of the compared methods. Smoothing window size is 5 (top), 15 (middle) and 30 (bottom).

Net++ and the unsupervised method obtain very noisy strain images and the inclusion is hardly visible. However the proposed bi-directional method and OVERWIND obtain acceptable strain images and the inclusion can be detected.

4.3.3 *In vivo* Results

Compared methods are evaluated with two *in vivo* data belonging to two patients. We also fine-tuned the bi-directional network using *in vivo* data to find out if further improvements

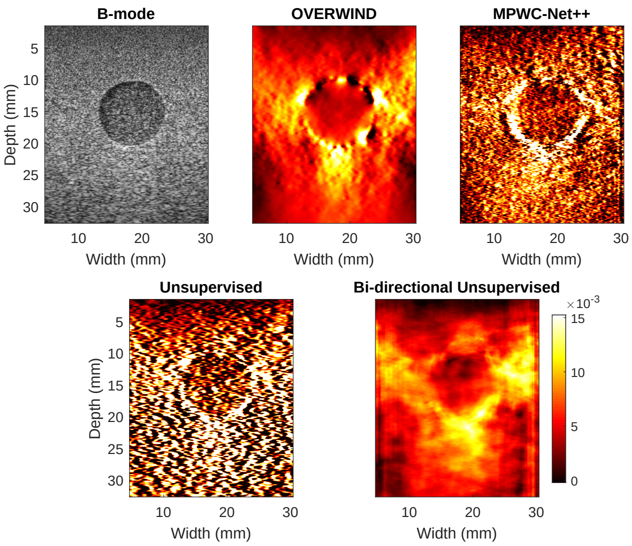


Figure 4.8: The lateral strain results. The proposed bi-directional method provides a smooth strain image close to OVERWIND, while MPWC-Net++ and the unsupervised method generate noisy strain images and the inclusion is barely visible.

Table 4.3: Quantitative results of *in vivo* data. The pairs marked by asterisk or star are not statistically significant (p-value$>$ 0.01).

	***In vivo* data 1**		***In vivo* data 2**	
	CNR	SR(%)	CNR	SR(%)
OVERWIND	**17.28±5.31**	21.80±4.16	7.18±1.58*	51.40±6.06*
MPWC-Net++	11.51±3.08	25.50±6.63	7.31±2.47*	48.89±11.61*
Unsupervised	11.91±2.62	19.20±5.30	6.73±3.09	47.83±20.83$^{\star *}$
Bi-directional Unsupervised	$\underline{16.27 \pm 5.26}$	19.35±5.46	7.91±3.18	$\underline{\mathbf{45.60 \pm 11.83}}$
Bi-directional Unsupervised + ft	14.37±4.40	$\underline{\mathbf{19.19 \pm 6.10}}$	$\underline{\mathbf{8.86 \pm 2.64}}$	46.14±12.91

can be achieved. The strain images of the compared methods are given in Fig. 4.9 and 4.10. OVERWIND produces high-quality strain images with low noise, while the strain images obtained by MPWC-Net++ have some over-smoothing especially in the lateral direction. Both unsupervised methods substantially improve the strain image qualities of MPWC-Net++. The bi-directional+ft also obtains high quality strain images, but the difference with the bi-directional method is not discernible.

The quantitative results are given in Table 4.3. OVERWIND achieves the highest CNR for *in vivo* data 1, while bi-directional unsupervised+ft has the best CNR for *in vivo* data 2. It can also be seen that fine-tuning on *in vivo* data does not results in considerable CNR improvement (it has slightly better CNR than bi-directional for *in vivo* data 2 and worse

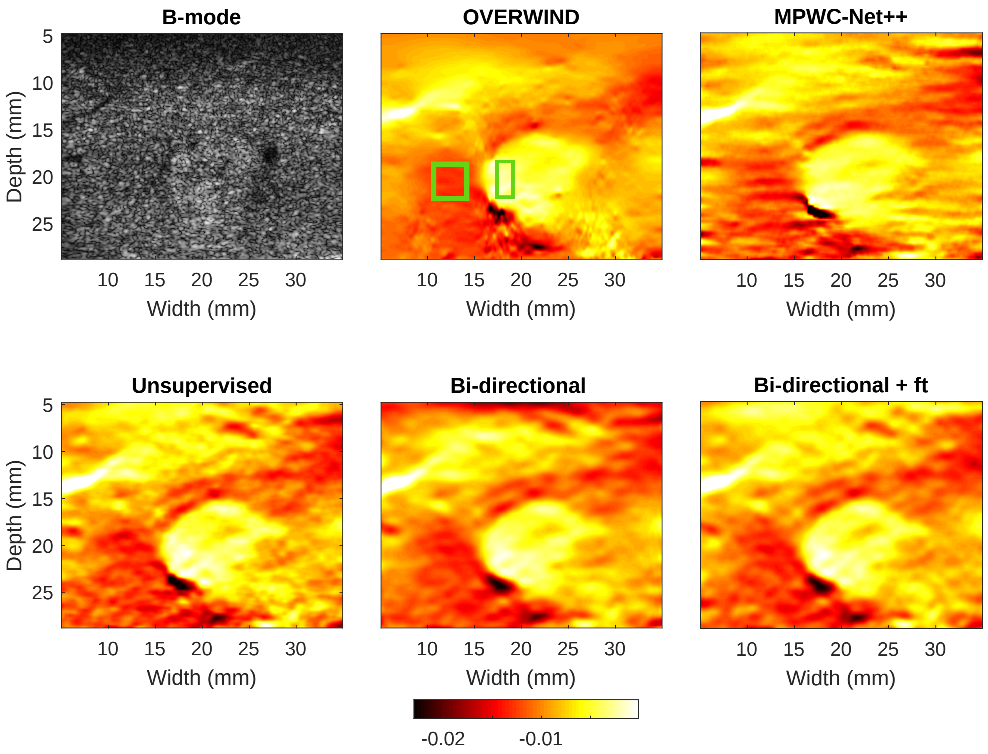


Figure 4.9: Strain images of *in vivo* data 1. The tumor has a lower absolute strain value but looks brighter since the strain is negative.

CNR for *in vivo* data 1). In terms of SR, bi-directional and bi-directional+ft have the best SR values. MPWC-Net++ and OVERWIND have the highest SR among the compared methods.

4.4 Discussion

In this chapter, we employed semi-supervised training to improve the performance of an optical flow network for USE. Although we used MPWC-Net++, which outperformed other networks for USE, the training method can be applied to other networks as well. It should be mentioned that the optical flow networks usually have pyramidal structure meaning that the displacements are estimated in different resolutions. Similar to previous works [55, 56] and unsupervised optical flow works [65], we only used the last output resolution for fine-tuning.

The optical flow CNNs trained on computer vision images do not employ regularizations due to abrupt changes in scenes such as a moving car in front of a fixed background. How-

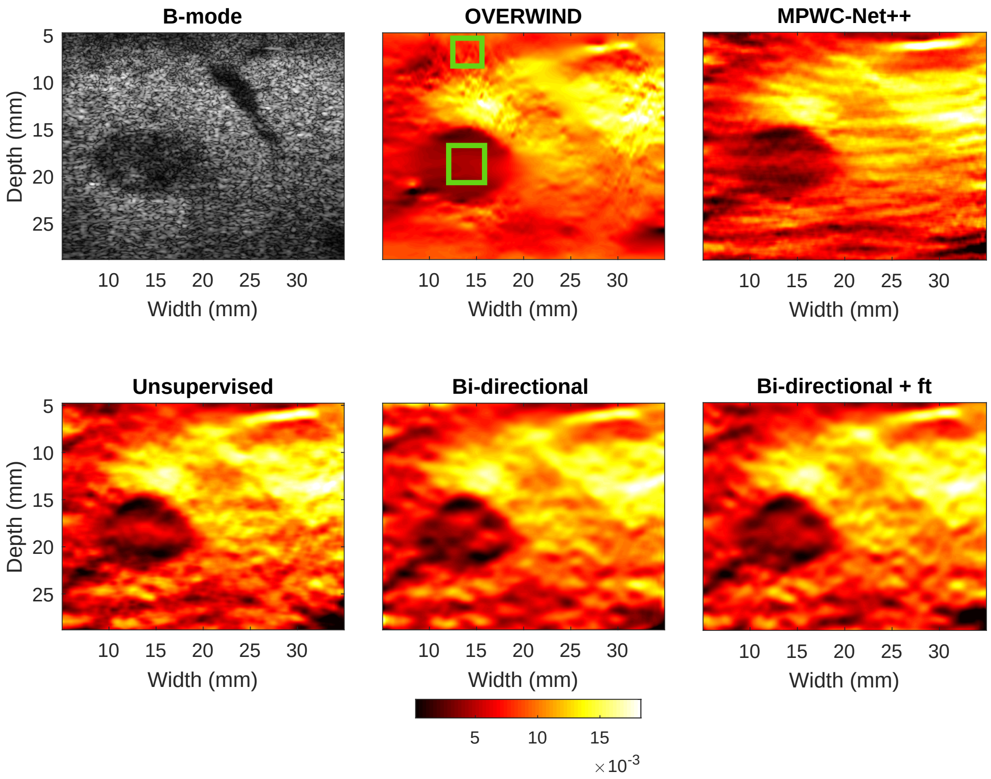


Figure 4.10: Strain images of *in vivo* data 2. The tumor has a lower absolute strain value and looks darker since the strain is positive.

ever, the displacement in USE is usually smooth without any sudden changes. The effect of the absence of the regularization can be seen in Fig. 4.7 (top), where the smoothing window is very small. The strain estimated by MPWC-Net++ has a high variance, whereas the bi-directional unsupervised method provides smooth and high-quality strain images by incorporating smoothness and consistency constrain in the estimated displacement.

The values reported in Tables 4.1, 4.2 and 4.3 are the mean and standard deviation of CNR and SR values. We have conducted statistical analysis to find out that if the difference between obtained values is statistically significant. We employ Friedman test [71] for statistical analysis.

Regarding the choice of weights, the explained method of tuning the weights only gives a rough estimate of the optimum weights. However, the output strain image is not considerably sensitive to these weights and similar performance can be obtained by different weights.

Running time is another important aspect that needs to be investigated. The deep learning methods shine in this aspect, our proposed method can provide high quality strain images close to OVERWIND for real-time applications. To give a general view about the computation time, OVERWIND takes 26 seconds for an image pair of size 1920 × 384 on CPU (8^{th} generation, core i7). It should be noted that this is a Matlab implementation, and an optimized implementation in C will be much faster. MPWC-Net++ and bi-directional MPWC-Net++ take 0.166 and 0.174 seconds on NVIDIA A100 GPU, respectively. The bi-directional variant of MPWC-Net++ takes slightly more time than MPWC-Net++ since it estimates both forward and backward displacements; however, it still takes much less time than running the network two times and computing forward and backward displacements separately.

4.5 Summary

In this chapter, we proposed a bi-directional semi-supervised deep learning method. We used strain consistency along with data and smoothness loss. We also employed the second-order derivatives regularization of axial and lateral displacements in both directions. Furthermore, we reduced the number of hyper-parameters by relating some of them to others by taking into account the underlying principles of the US. We showed that our proposed method substantially improved current optical flow networks used for USE. We validated our method using different experimental phantom and *in vivo* data. Our proposed method obtained strain images close to OVERWIND.

Chapter 5

Lateral Strain Imaging using Self-supervised and Physically Inspired Constraints in Unsupervised Regularized Elastography

5.1 Introduction

In the previous chapter, we investigated unsupervised training on real ultrasound data which prepares the trained model to extract more suitable features from RF data. Prior knowledge of displacement map continuity is also utilized in the forms of different regularization strategies. In [56], a combination of first and second-order derivatives of the displacements is employed as the regularization, which has been found beneficial in the recent optimization-based methods [67]. Wei *et al.* adapted MaskFlownet [72] to USE and trained the network using an unsupervised method. They also performed a detailed comparison of their network with MPWC-Net++ [73].

In classical methods, Babaniyi *et al.* [74] considered plane stress and incompressibility assumptions to refine the estimated displacement. Guo *et al.* first introduced a refinement method that incorporated the incompressibility and plane strain assumptions in an iterative approach [75] that substantially improved the lateral strain. Other lateral strain imaging works mainly focus on modifying the imaging technique to have a higher resolution in lateral direction [76–78], and, as such, cannot be applied to the already available US data. The smoothness of the derivatives of the displacements is the only prior knowledge of USE physics used in previous unsupervised training. No deep learning work considers the physics of the compression of the tissue into account. Also, no deep learning method has focused on improving the quality of the lateral displacement estimation, which is challenging but it is required for elasticity [70] and Poisson's ratio imaging [54].

In our preliminary work, we investigated the feasibility of improving the lateral displacement by employing the prior knowledge of compression physics [79], where we introduced physically inspired constraint for unsupervised regularized elastography (PICTURE). In this chapter, the method is explained in more detail, and new extensive experiments are performed to better evaluate the effectiveness of the technique. We also introduce self-supervision in USE and propose sPICTURE, which further boosts the performance.

5.2 Method

5.2.1 Unsupervised Training

Let $I_1, I_2 \in \mathbb{R}^{3\times W\times H}$ denote the pre- and post-compression US data, respectively. The subscript 3 refers to three channels of RF data, the envelope of RF data, and the imaginary part of the analytic signal similar to [62]. The unsupervised cost function is composed of data loss and smoothness regularization loss. The data loss (L_D) in unsupervised training can be obtained by comparing I_1 with the warped I_2 ($\hat{I_2}$) by the displacement map W. The data loss can be written as [80, 81]:

$$L_D = ||(I_1 - \hat{I_2})||_{1(N\times N)} \tag{5.1}$$

where $||.||_1$ denotes norm 1 (as suggested by [37, 58], L2 norm is not suitable; therefore, a norm lower is employed), and a window of size $N \times N$ (here 3×3) is considered around each sample to reduce the effect of noise. For the regularization, we adopt the method of [56, 81] where the strains and their first-order derivatives are employed. The strains can be obtained by taking the derivative of displacement in direction (x) with respect to the direction (y):

$$
\begin{aligned}
&\varepsilon_{xy} = \frac{\partial W_x}{\partial y} \\
&x, y \in 1, 2, 3
\end{aligned}
\tag{5.2}
$$

we assumed that the subscripts 1, 2, and 3 denote axial, lateral, and out-of-plane directions, respectively. By this assumption, ε_{11}, ε_{22} and, $(\varepsilon_{21} + \varepsilon_{12})/2$ are the axial, lateral and, shear strains, respectively. The smoothness loss can be defined as:

$$
\begin{aligned}
L_S &= L_{s1} + \gamma L_{s2} \\
L_{s1} &= ||\varepsilon_{11} - < \varepsilon_{11} > ||_1 + \beta ||\varepsilon_{12}||_1 + \frac{1}{2}||\varepsilon_{21}||_1 + \frac{1}{2}\beta||\varepsilon_{22}||_1 \\
L_{s2} &= \Big\{ ||\ (\frac{\partial \varepsilon_{11}}{\partial a})||_1 + \beta||\ (\frac{\partial \varepsilon_{11}}{\partial l})||_1 + \\
&0.5||\ (\frac{\partial \varepsilon_{22}}{\partial a})||_1 + 0.5\beta||\ (\frac{\partial \varepsilon_{22}}{\partial l})||_1 \Big\}
\end{aligned}
\tag{5.3}
$$

where L_S is the total smoothness loss, $< . >$ denotes averaging operation, and β depends on the ratio of spatial distance between two samples in lateral to the axial direction and it is set to 0.1 similar to [81]. L_{s1} and L_{s2} are the regularization of first- and second-order derivatives of the displacements. γ is a hyperparameter that controls the weight of the second-order derivatives smoothness loss.

5.2.2 Hooke's Law and compression physics

Assuming that the tissue is linear elastic and isotropic, the following two sections show the relation between the lateral and axial displacements under uniform compressions.

Homogeneous Material

Hooke's law can be formulated as [9]:

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -v & -v & 0 & 0 & 0 \\ -v & 1 & -v & 0 & 0 & 0 \\ -v & -v & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2+2v & 0 & 0 \\ 0 & 0 & 0 & 0 & 2+2v & 0 \\ 0 & 0 & 0 & 0 & 0 & 2+2v \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} \tag{5.4}$$

where E, σ, and v represent Young's modulus, stress tensors, and Poisson's ratio, respectively. When there is a compression of the material in one direction, there is an expansion in the other direction, which depends on the Poisson's ratio of the material. In free-hand palpation, it can be assumed that the external force is only in the axial direction (uniaxial stress); therefore, other stress components except σ_{11} can be ignored. This assumption simplifies Eq 5.4 and leads to [9]:

$$\varepsilon_{11} = \frac{\sigma_{11}}{E}, \varepsilon_{22} = -v\frac{\sigma_{11}}{E}, \varepsilon_{33} = -v\frac{\sigma_{11}}{E} \tag{5.5}$$

which indicates that the lateral strain (ε_{22}) can be directly obtained by the axial one (ε_{11}) using $-v \times \varepsilon_{11}$.

Inhomogeneous Materials

Tissues cannot be assumed to be homogeneous due to the presence of irregularities and boundary regions; therefore, the lateral strain cannot be directly obtained by the axial one and Poisson's ratio. In this condition, the total strain (ε_{xy}) is obtained by adding the elastic strain (e_{xy}) and eigenstrain (ε^*_{xy}) [82]:

$$\varepsilon_{xy} = e_{xy} + \varepsilon^*_{xy} \tag{5.6}$$

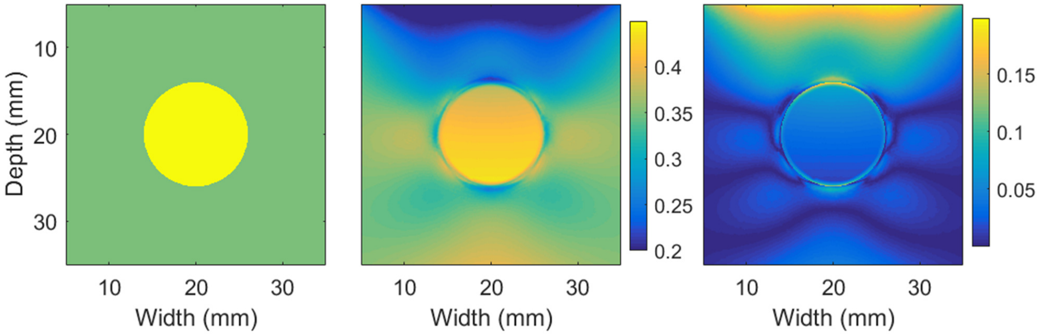


Figure 5.1: From left to right: the Poisson's ratio, the EPR, and their absolute difference for a simulated phantom. The Poisson's ratio and the EPR have the same colorbar.

Eigenstrain is added to consider the variation of total strain from the elastic one in the presence of inhomogeneity. It is maximum on the inhomogeneity boundaries and decays to zero further from the boundaries [82]. Although the lateral strain does not linearly depend on the axial one anymore, they are still highly correlated. Also, $-\varepsilon_{22}/\varepsilon_{11}$ does not obtain the Poisson's ratio anymore and it is called effective Poisson's ratio (EPR) [54]. In uniform regions far from inhomogeneities, EPR converges to Poisson's ratio. For illustration purpose, EPR and Poisson's ratio of a finite element simulation using ABAQUS software (Providence, RI) is depicted in Fig. 5.1. It can be observed that EPR is more dissimilar to Poisson's ratio at the top and bottom of the phantom and around the inclusion. Poisson's ratio and EPR under arbitrary deformation have the range between 0.2 and 0.5 [83, 84]. Although the exact value of EPR is not known, it has been used as an approximation of Poisson's ratio to characterize tissues [54, 84]. We propose to use this range as a prior information to guide the network to refine the lateral displacement. Guo *et al.* assumed tissue incompressibility (Poisson's ratio = 0.5) and plane strain (strain in out-of-plane direction = 0) to refine the displacements. However, Poisson's ratio depends on the type of the tissue (refer to [85, 86] for Poisson's ratio of breast and liver). In this work, we do not make those assumptions, and only a feasible range of Poisson's ratio is enforced.

5.2.3 PICTURE

We propose to utilize the accepted range of EPR as a prior information in the form of regularization. To do that, we first need to calculate EPR (V_e) from the estimated axial and lateral strains:

$$V_e = \frac{-\tilde{\varepsilon_{22}}}{\mathbf{S}(\tilde{\varepsilon_{11}})} \tag{5.7}$$

where $\tilde{\varepsilon_{22}}$ and $\tilde{\varepsilon_{11}}$ are the lateral and axial strains obtained from estimated displacements. The parameter $\mathbf{S}$ denotes stop gradient operation, which is used to stop backpropagation of the loss to the axial displacement estimation. It is used to avoid the estimated axial displacement being altered by the noisy lateral one. To find out the EPR values outside the accepted range, a mask (M) is defined using the minimum (V_{emin}) and maximum (V_{emax}) allowed EPR values.

$$M(i,j) = \left\{ \begin{matrix} 0 & V_{emin} < V_e(i,j) < V_{emax} \\ 1 & otherwise \end{matrix} \right\} \tag{5.8}$$

We assume the V_{emin} and V_{emax} values to be 0.1 and 0.6 (no noticeable change was observed by small changes of these values) to have a small margin of error. In order to penalize the EPR values outside of the accepted range, PICTURE loss is defined as:

$$L_{vd} = \left|\left| M \otimes (\tilde{\varepsilon_{22}} + \bar{V_e} \times \mathbf{S}(\tilde{\varepsilon_{11}})) \right|\right|_2 \tag{5.9}$$

where $\otimes$ is the Kronecker product to select EPR values outside the accepted range, and $\bar{V_e}$ is an estimate of true average EPR. It is obtained by averaging EPR values that are inside the accepted range, which can be formulated as:

$$\bar{V_e} = \frac{\sum_{i,j}(1 - M_{(i,j)})V_e(i,j)}{\sum_{i,j}(1 - M_{(i,j)})} \tag{5.10}$$

Eq 5.9 tries to constrain EPR to be inside the accepted range.

The first-order derivatives of V_e are also employed to enforce the smoothness of EPR.

$$L_{vs} = ||\frac{\partial V_e}{\partial a}||_1 + \beta \times ||\frac{\partial V_e}{\partial l}||_1 \tag{5.11}$$

The final PICTURE loss can be written as:

$$L_V = L_{vd} + \lambda_{vs} \times L_{vs} \tag{5.12}$$

where λ_{vs} is defined to weight the smoothness part.

5.2.4 Self-Supervised Learning

Self-supervised learning (SSL) is a technique that has recently been applied to unsupervised optical flow networks [87, 88]. The basic procedure is that the input images are fed to the network during the unsupervised training, and the displacements are obtained during the *first pass*. In the next step, the input images are transformed to make them more challenging than before and the new displacement is obtained during the *second pass*. In the last step, the differences between the displacements of the first and second passes are penalized:

$$L_{SSL} = ||\mathbf{S}(W) - \tilde{W}||_1 \tag{5.13}$$

where W is the estimated displacement in the first pass (no transformation), $\tilde{W}$ is the obtained displacement from the second pass (with transformed inputs), and stop gradient ($\mathbf{S}$) is used to avoid backpropagation into the first pass. Substantial improvements were reported for unsupervised training employing SSL using different transformations. In [88], superpixels [89] of input images were identified and the content of randomly selected superpixels were replaced by pure noise. The method outperformed other unsupervised methods in different optical flow benchmarks. In [87], cropping, affine, and other kinds of transformations were utilized. SSL was also compared by data augmentation (instead of SSL, the transformed images were considered as new inputs). SSL outperformed data augmentation, which indicated that SSL was not a simple synthetic data generation like data augmentation.

In this chapter, we employ two data transformations: cropping and adding noise to specific regions. Cropping may cause loss of information, especially in areas where displacement is high. On those areas, estimating displacement is complex for the network since the corresponding part of the first image might be outside the cropped second image. We also add

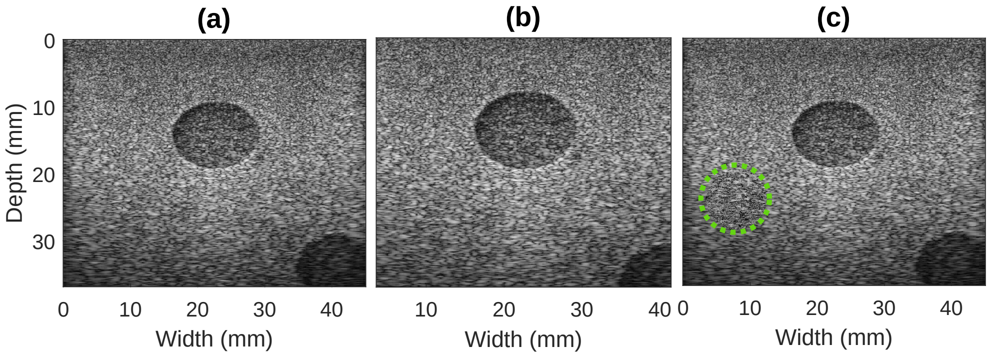


Figure 5.2: B-mode input image (a), cropped image (resized for the purpose of visualization) (b) and, image with added noise (the area of added noise is highlighted) (c).

large Gaussian noise to randomly selected circular regions. An example of a transformed image is shown in Fig. 5.2. SSL can guide the network to have a more reliable estimation when there is a loss of information due to cropping or noisy data.

5.2.5 Loss Function And Network Architecture

The loss function is composed of data loss (Eq 5.1), smoothness loss (Eq 5.3), PICTURE loss (Eq 5.9), and SSL regularization (Eq 5.13):

$$loss = \underbrace{L_D + \lambda_s L_S + \lambda_v L_V}_{first\ pass} + \underbrace{\lambda_{sl} L_{SSL}}_{second\ pass} \tag{5.14}$$

where the hyper-parameters, λ_s, λ_v, and λ_{sl} are the weights of smoothness regularization, PICTURE loss, and SSL regularization, respectively. The SSL loss only affects the second pass in which the input US data are transformed, while the other losses affect the first pass.

We employed MPWC-Net++ and it is trained for 25 epochs, the learning rate is set to 5×10^{-6}, which is halved every 5 epochs.

5.2.6 Datasets

Simulation data

A phantom is simulated using Field II [45, 46], and the motion is obtained by the ABAQUS finite element analysis software (Providence, RI). The phantom contains an inclusion with

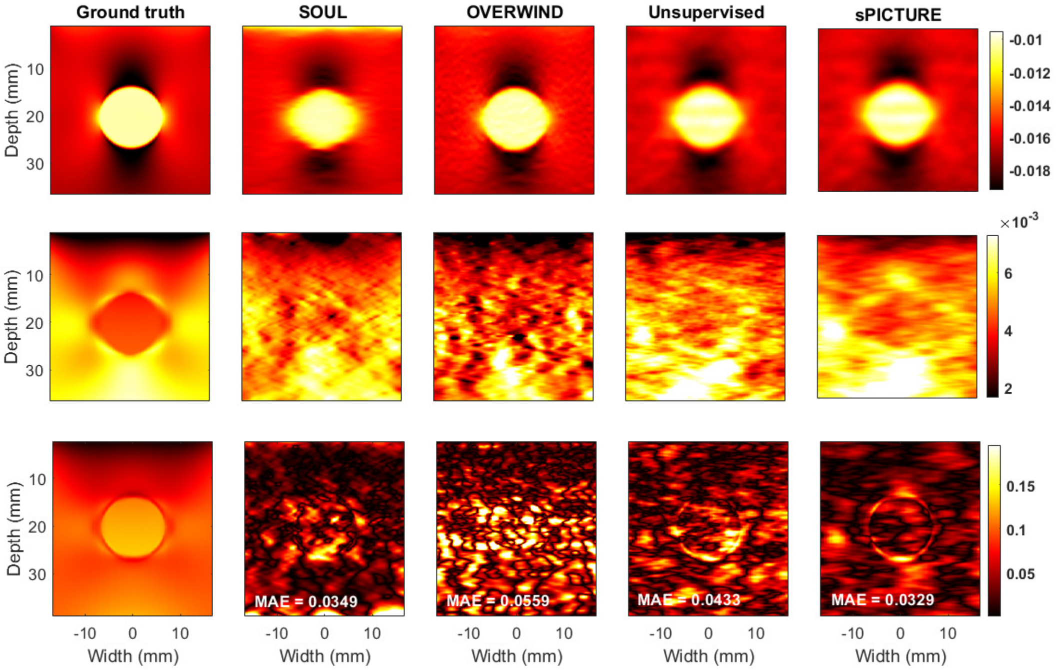


Figure 5.3: Axial (top row) and lateral (middle row) strains in a simulated phantom. Ground truth EPR (bottom row), the absolute error, and the mean absolute error (MAE) shown for each method.

Poisson's ratio of 0.45 and the Young's modulus of 40 kPa. The background has Poisson's ratio of 0.35 and the Young's modulus of 21 kPa. The Poisson's ratio and EPR of this phantom are shown in Fig. 5.1. Different Poisson's ratios for background and the inclusion are chosen to investigate if it is detectable by the networks. Compared to our recent simulation dataset [55], the number of lines in FIELD II is increased to 190, the number of active elements is increased to 96, and the obtained US images are also upsampled in the lateral direction by 2 to provide high lateral resolution.

In addition, 1200 pairs of publicly available simulated phantoms from [55] are employed for training and 70 pairs for quantitative evaluation of the compared methods. These phantoms have a Poisson's ratio of 0.49 and have one or two hard inclusions in different locations. The Young's modulus for the background and inclusions are around 20 kPa and 45-60 kPa, respectively.

Table 5.1: Quantitative results for 70 simulated phantoms. Mean and standard deviation (±) of the MAE of displacements and SSIM of strains are reported. The pairs marked by asterisk are not statistically significant (p-value>0.05, using Friedman test).

	Axial		Lateral	
	MAE (μm)	SSIM (%)	MAE (μm)	SSIM (%)
SOUL	**2.2**±1.5	**99.80**±0.06	8.00±4.1*	97.70 ± 1.56*
OVERWIND	**2.2**±1.5	**99.80**±0.07	9.40±4.6	93.48±4.10
Unsupervised	2.7±1.6*	99.43 ± 2.10	8.70±4.1	96.42±1.79
sPICTURE	2.7±1.7*	99.55 ± 1.80	**8.00**±3.8*	**97.73** ±1.29*

Experimental phantom data

2200 frame pairs of experimental phantom data explained in Chapter 2 are employed for training of the network. In order to avoid data leakage, different parts of the phantom were imaged for evaluation and test.

in vivo data

Data were acquired from patients with liver cancer during open-surgical RF thermal ablation at Johns Hopkins Hospital. A research Antares Siemens system by a VF 10-5 linear array was used for data collection. The sampling and center frequencies were 40 MHz and 6.67 MHz, respectively. The study was approved by the institutional review board with the consent of all patients. 500 RF frame pairs from after ablation were selected for the training of the networks, and RF data from 2 patients before ablations were employed for test to prevent using similar data during the train and test phases.

5.3 Results

The evaluated methods are listed below:

- Second-order ultrasound elastography (SOUL) is an optimization-based method which employs L2-norm and second-order regularization to have smooth strain images with high target-background contrast [67].

- Total variation regularization and window-based time delay estimation (OVERWIND)

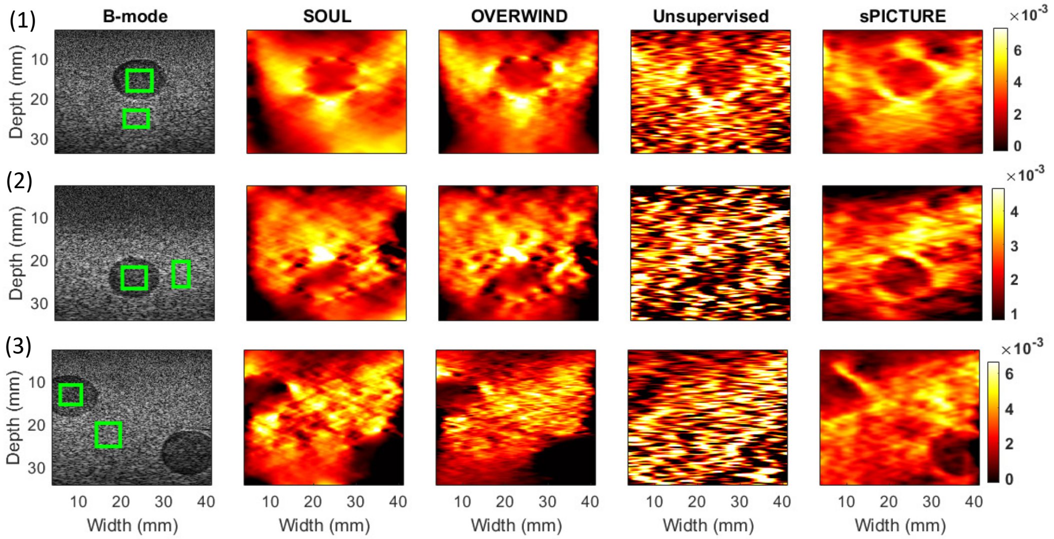


Figure 5.4: The experimental phantom lateral strains obtained by the evaluated methods. The target and background windows for calculation of CNR and SR are marked in the B-mode images. The samples 1, 2 and 3 are taken from different locations of the tissue-mimicking breast phantom. Hard inclusions have lower absolute values than the background.

is a method that incorporates windowing into the optimization cost function [29].

- Unsupervised method ($\lambda_v = 0$ and $\lambda_{sl} = 0$ in Eq 5.14, similar to the unsupervised method in [81]).

- The PICTURE without SSL ($\lambda_{sl} = 0$ in Eq 5.14, only used in ablation experiment).

- The proposed method named sPICTURE entails both PICTURE and SSL losses (Eq 5.14).

It should be mentioned that for simulation results the network for unsupervised method and sPICTURE is trained on simulation data. For the experimental phantom results, it is trained on the experimental phantom dataset, and for *in vivo* results, it is trained on the available *in vivo* dataset. We also tuned the hyperparameters of the optimization-based methods (SOUL and OVERWIND) for each dataset separately to ensure that the best results are obtained from those methods.

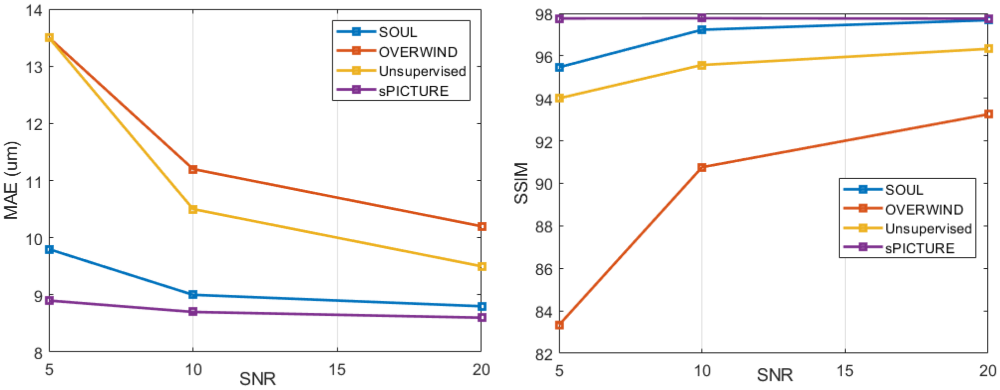


Figure 5.5: MAE of lateral displacements (left), and SSIM of lateral strains (right) for different SNR values of simulation test data.

5.3.1 Simulation Results

The axial, lateral strains, and the EPR of the simulated phantom obtained by the compared methods are illustrated in Fig. 5.3. All methods obtain high-quality axial strains. The axial strain of the unsupervised method and sPICTURE are quite similar since PICTURE is only applied to the lateral displacement and keep the axial one untouched. Comparing the lateral strains (second row), sPICTURE reduces the noise presented in the unsupervised method.

The mean and standard deviations of quantitative metrics are reported for 70 simulated phantoms. Since the ground truth is available, MAE of displacement and SSIM of strain are reported for the axial and lateral displacements and strains. SOUL and OVERWIND have the lowest MAE error and highest SSIM in the axial direction. sPICTURE performs similar to the unsupervised method since PICTURE does not affect the axial direction. In lateral displacement estimation, sPICTURE reduces the lateral MAE of unsupervised method from 25.0 to 7.9, a decrease of more than three folds. It also outperforms the optimization-based methods in terms of MAE, with SSIM values close to those of SOUL.

simulation results for different signal to noise ratios (SNR)

Random Gaussian noise with different SNRs is added to the test RF data to evaluate the robustness of the compared method to noise. MAE of lateral displacement and SSIM of lateral strain are plotted in Fig. 5.5. It can be observed that sPICTURE has a low MAE even for an SNR as low as 5 dB which demonstrates the high robustness of this method.

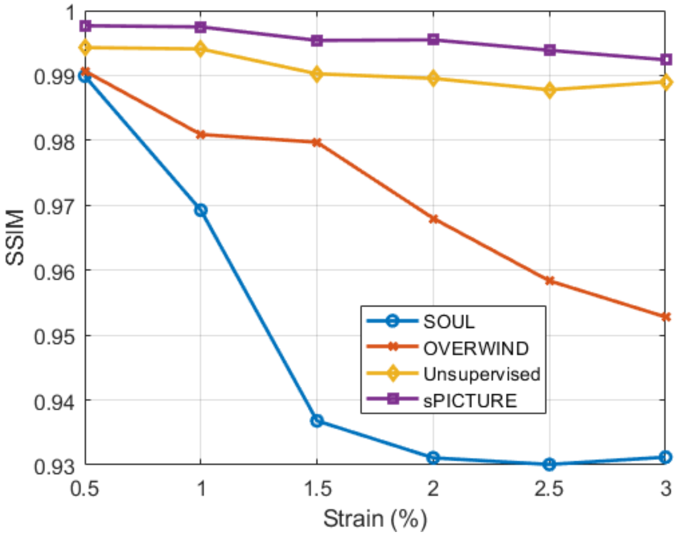


Figure 5.6: SSIM of lateral strains versus different maximum strains.

Table 5.2: Quantitative results of lateral strains for experimental phantoms and *in vivo* data. Mean and standard deviation (±) of CNR (higher is better) and SR (lower is better) of lateral strains are reported. The pair marked by asterisk is not statistically significant (p-value>0.05, using Friedman test).

	Phantom 1		Phantom 2		Phantom 3		*In vivo* 1		*In vivo* 2	
	CNR	SR	CNR	SR	CNR	SR	CNR	SR	CNR	SR
SOUL	11.01±4.52	0.483±0.038	3.63±1.20	0.676±0.081*	2.33±0.81	0.327±0.140	3.25±1.07	0.428±0.137	1.26± 0.843	1.110±0.124
OVERWIND	7.21±1.91	0.456±0.057	2.35±0.68	0.676±0.092*	3.38±1.45	0.285±0.155	1.92±0.96	0.584±0.150	0.84±0.61	1.048±0.202
Unsupervised	2.31±0.30	**0.454**±0.061	1.02±0.30	0.530±0.105	0.26±0.18	0.677±0.743	0.24±0.18	0.905±0.159	0.50±0.36	1.125 ± 0.437
sPICTURE	**11.20**±2.18	0.511±0.059	**9.14**±2.80	**0.527**±0.044	**7.07**±1.76	**0.278**±0.065	**7.80**±2.01	**0.242**±0.066	**4.34**±1.39	**0.640**±0.075

simulation results for different compression

A phantom from simulation test data having different applied compressions, resulting in different maximum strains, is selected, and SSIM of lateral strain are illustrated in Fig. 5.6. By increasing the maximum strain, the SSIM of all compared method decreased which is expected. It should be noted that the graph shows that sPICTURE has the highest SSIM among the compared methods which is also robust to the variations of applied compression.

5.3.2 Experimental Phantom Results

The lateral strains of experimental phantom results are shown in Fig 5.4 and the quantitative results are reported in Table 5.2.

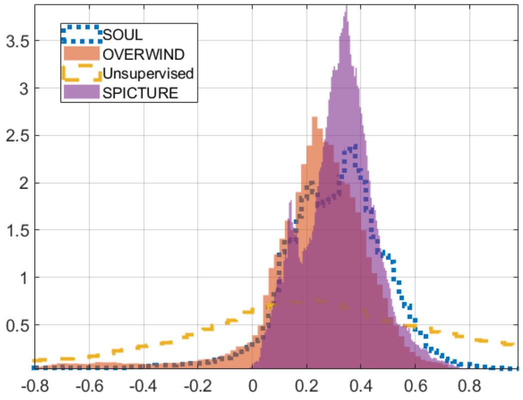


Figure 5.7: The histograms of the EPR of different methods for phantom 3. sPICTURE has limited the EPR to the feasible range for USE.

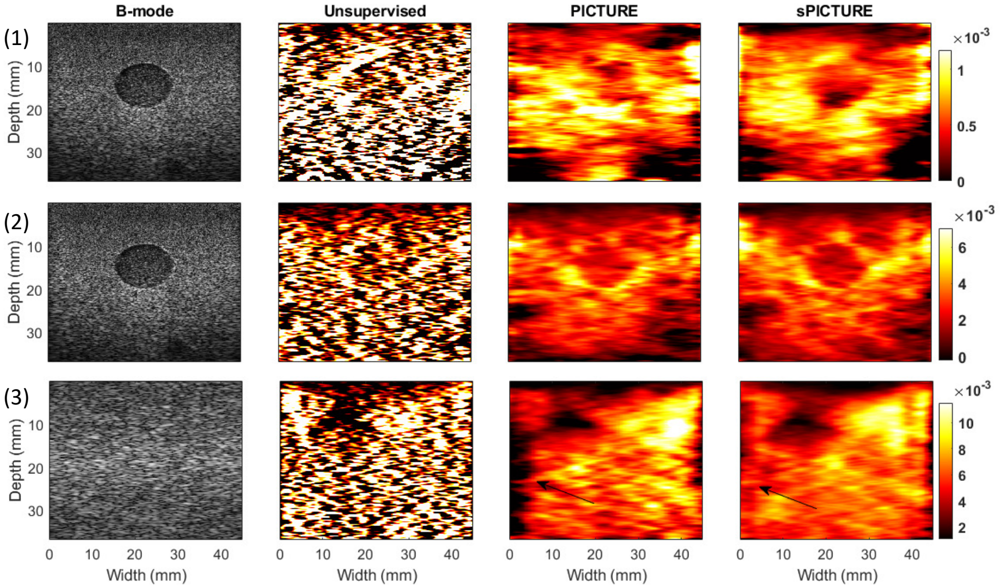


Figure 5.8: Ablation experiment results. In (3), the inclusion is not visible in B-mode image and arrows show that SSL improves the estimation in boundary regions. The samples 1, 2 and 3 are taken from different locations of the tissue-mimicking breast phantom.

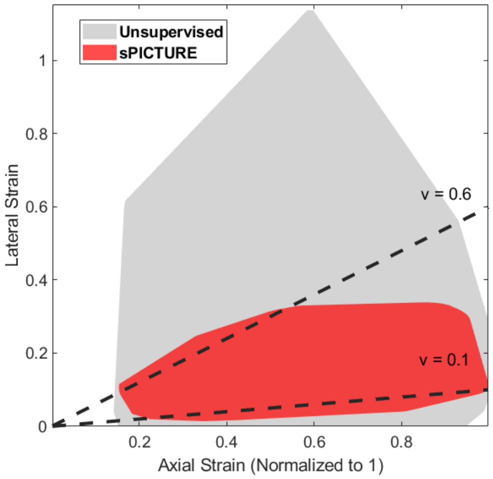


Figure 5.9: Normalized axial strain versus the corresponding lateral strain. The region where the samples of the methods lie for experimental phantom 2 are specified. The regions are obtained from convex hull of strain samples. EPR equals to 0.1 and 0.6 are highlighted by the dashed lines.

Unsupervised method has unacceptable results in which the inclusions are not visually detectable while it provides high-quality axial strain images comparable to optimization-based methods. This is an important observation since this shows that the unsupervised loss (composed of data and smoothness losses), which has been used widely in computer vision optical flow estimation, is not a suitable loss in USE.

sPICTURE provides high-quality lateral strain images and performs the best in terms of quantitative results among the compared methods. By comparing the unsupervised and sPICTURE results, it can be seen how the added PICTURE regularization and the SSL lead to the improvement of the obtained strain images. The added regularizations convert the unreliable and noisy lateral strains obtained by unsupervised method to the high-quality strain images. It should be mentioned that sPICTURE and unsupervised methods are both trained using the same data and weights for smoothness regularization. Furthermore, sPICTURE obtains substantially higher quality lateral strain images than the compared optimization-based methods (both visually and quantitatively).

To further analyze the results, the histograms of the EPR of phantom data 3 are depicted in Fig. 5.7. As mentioned earlier, EPR range is similar to the Poisson's ratio range (0.2 to 0.5, excluding the boundary regions). In PICTURE loss, we penalize EPR values outside

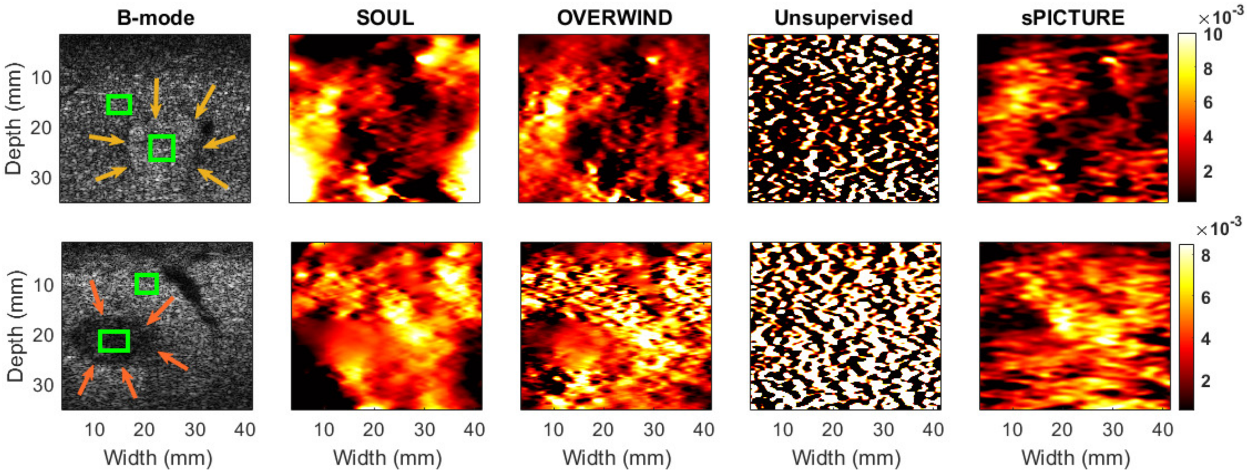


Figure 5.10: The *in vivo* lateral strains obtained by the evaluated methods. The target and background windows for calculation of CNR and SR are marked in the B-mode images.

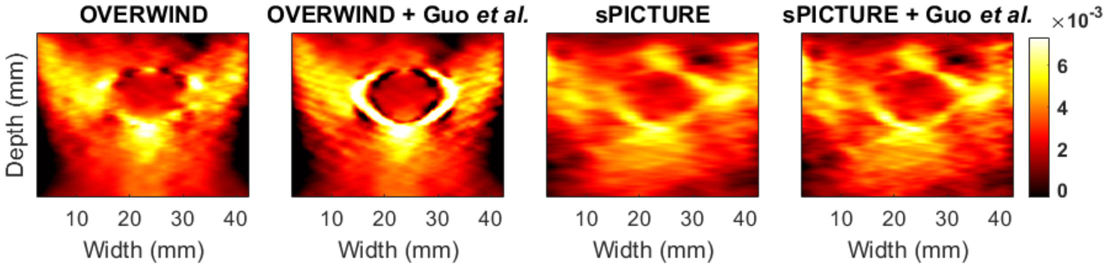


Figure 5.11: Lateral strains of OVERWIND and sPICTURE after applying the method of Guo *et al.* [75] to real phantom data (1).

the 0.1-0.6 range. The histogram of EPR of unsupervised method covers a wide range of positive and negative values which indicates that many lateral strain values obtained by this method are incorrect. The histogram of the EPR values of OVERWIND and SOUL are more limited than unsupervised method, but they contain values that are negative or higher than 0.8 which is not possible in the phantom. sPICTURE has a more limited range of EPR values but still has some values outside the specified range. The reason is that the proposed PICTURE regularization is only applied during the training phase. Although the proposed PICTURE regularization reduces the range of EPR values, it does not guarantee that all the values fall into the specified range in test time.

Ablation Experiment

In order to investigate the impact of PICTURE loss and SSL separately, an ablation experiment is conducted. Fig. 5.8 shows the visual comparison of unsupervised method (without PICTURE and SSL), PICTURE (without SSL), and sPICTURE. It is visually clear that both PICTURE and SSL contribute to the improvements obtained by sPICTURE. Without PICTURE, unsupervised method provides noisy and impractical lateral strain images. PICTURE substantially improves the lateral strain image quality and SSL further boosts the quality of the lateral strain image. For instance, in sample (1), the inclusion location can be detected more accurately in sPICTURE compared to PICTURE. Also, the estimation in boundary regions is improved in sPICTURE since it is trained to deal with cropping with SSL. It should be mentioned that SSL without PICTURE was also tested, but it performed inferior to PICTURE.

Experimental Results after applying lateral displacement refinement

Lateral displacement refinement of Guo *et al.* [75] is applied using the initial displacement obtained by OVERWIND and sPICTURE. It can be observed that this method further improves the lateral displacement estimation, and the initial displacement obtained by sPICTURE provides a high-quality initial value for this method.

5.3.3 *In vivo* Results

The *in vivo* lateral strains of two patients with liver cancer are depicted in Fig. 5.10, quantitative results are reported in Table 5.2. The tumors are more visually detectable in sPICTURE compared to the other methods. Also, quantitative results denote that sPICTURE has the highest CNR and lowest SR values among the compared methods, which confirms the visual analysis.

5.4 Discussion

In this chapter, a physically inspired regularization to improve the lateral displacement estimation has been proposed. It confines the range of EPR by employing the high-quality axial strain and the known range of EPR values. One limitation of the proposed method is that PICTURE similar to any other form of regularizations is only applied during the training. Even though it limits the range of EPR values in the test time, it does not guarantee that all EPR values be within that range. We observed only a few samples lie outside of the defined range and fixing them during the test time inspired by known operators [90] can be an area of future works.

It should be mentioned that PICTURE can also have statistical interpretation. The lateral displacement prediction can be viewed as the estimation of a parameter from under sampled and heavily smoothed observations. The conventional methods estimate this parameter in a maximum likelihood (ML) manner without any prior information (only smoothness is considered). However, PICTURE can be viewed as maximum a posteriori (MAP) estimate in which the prior information from compression physics is utilized to find the parameters. Therefore, more reliable lateral displacement can be estimated compared to the conventional methods. To clarify this, the graph of lateral versus axial strains is depicted in Fig. 5.9. PICTURE enforced the strain samples to lie within $v = 0.6$ and $v = 0.1$. The areas where the samples of unsupervised method and sPICTURE lie for experimental phantom data 2 are illustrated in the figure. It can be observed that most of the strain samples of sPICTURE lie within the correct range. sPICTURE moved the lateral sample values to the area of the prior knowledge.

Self-supervision was another regularization that has been used in this work. SSL can prepare the model to deal with corrupted data. In this chapter, we applied cropping and added noise. Cropping helps the model deal with boundary regions where finding the correspondence between pre and post-compression images is difficult. Adding noise can also be useful in some scenarios for instance when there is a loss of signal due to high attenuation or there is a cyst where clutter is stronger than the true signal. Applying other forms of transformation such as acoustic noise (reverberation and multiple scattering), inducing decorrelation, and downsampling can be an area of future works.

In this chapter, the performance of lateral displacement refinement method of [75] using initial value from sPICTURE and OVERWIND is also investigated. This method is considered as a post-processing method that relies on the initial displacements. We showed that high-quality lateral displacement of sPICTURE can be considered as a good initial value for this method and improves the results of this refinement method.

The complexity of the training is another issue that should be discussed. We utilized two parallel NVIDIA A100 GPUs with 40 GB of memory each. Even with this size of memory, the maximum batch size that we could train the network was 8. The main reason is that the image sizes are usually large to preserve high-frequency RF data contents and the memory usage is also doubled by the second pass required in SSL. Only the training phase is memory intensive, and inference can be performed with only 5 GB of memory in 140 *ms* (for an US data of size 2048 × 256) similar to MPWC-Net++.

5.5 Summary

In this chapter, we proposed PICTURE to improve the lateral strain images in USE using physically inspired priors. We further improved the method in sPICTURE by adding the self-supervision to the method. The effectiveness of the proposed method is validated using simulation, experimental phantom, and *in vivo* data.

Chapter 6

Infusing physically inspired known operators for ultrasound elastography displacement estimation

In this chapter, we aim to embed two lateral displacement refinement algorithms in the CNNs to improve the lateral strains. The first algorithm limits the range EPR inside the feasible range during the test time. It is important to note that in contrast to [79], the EPR range is enforced using the regularization during the training phase and the known operators framework during the test phase; therefore, it is enforced during both training and test phases. The second algorithm employs the refinement method proposed be Gou *et al.* [75] which exploits incompressibility constraint to refine the lateral displacement.

6.1 Materials and Methods

In this section, we first provide a brief overview of PICTURE and underlie some differences to this work. We then introduce our method for incorporating known operators into our deep model and outline our unsupervised training technique. We then present the training and test datasets and finish the section by demonstrating the network architecture.

6.1.1 PICTURE

Let ε_x denote axial ($x = 1$), lateral ($x = 2$), and out-of-plane ($x = 3$) strains. Assuming linear elastic, isotropic, and homogeneous material, the lateral strain can be obtained from the axial strain and the Poisson's ratio by $\varepsilon_2 = -v \times \varepsilon$. Real tissues are inhomogeneous, and boundary conditions exist; therefore, the lateral strain cannot be directly obtained by the axial strain and the Poisson's ratio alone. In such conditions, EPR, which is defined as $v_e = \frac{-\varepsilon_{22}}{\varepsilon_{11}}$ can be employed [82]. EPR is spatially variant, and it is not equal to Poisson's ratio, but it has a similar range of Poisson's ratio, i.e., between 0.2 and 0.5 [83]. In PICTURE, a regularization was defined to exploit this range and the out-of-range EPRs were penalized [79]. PICTURE loss can be obtained from the following procedure:

1- Detect out-of-range EPRs by:

$$M(i,j) = \begin{Bmatrix} 0 & v_{emin} < \widetilde{v_e}(i,j) < v_{emax} \\ 1 & otherwise \end{Bmatrix} \tag{6.1}$$

where $\widetilde{v_e}$ is the EPR obtained from the estimated displacements. v_{emin} and v_{emax} are two hyperparameters that specify the minimum and maximum accepted EPR values, which are assumed to be 0.1 and 0.6, respectively.

2- Penalize the out-of-range lateral strains using:

$$\begin{gathered} L_{vd} = |(\varepsilon_{22} + < \widetilde{v_e} > \times \mathbf{S}(\varepsilon_{11}))|_2 \\ \bar{V_e} = \frac{\sum_{i,j}(1 - M_{(i,j)})V_e(i,j)}{\sum_{i,j}(1 - M_{(i,j)})} \end{gathered} \tag{6.2}$$

where $< \widetilde{v_e} >$ is the average of EPR values within the feasible range. The operator $\mathbf{S}$ denotes stop gradient operation, which is employed to avoid the axial strain being affected by this regularization. It should be noted in contrast to [79] in which only out-of-range samples were contributing to the loss, in this work, all samples contribute to L_{vd} to reduce the estimation bias.

3- Smoothness of EPR is considered by:

$$L_{vs} = |\frac{\partial v_e}{\partial a}|_1 + \beta \times |\frac{\partial v_e}{\partial l}|_1 \qquad (6.3)$$

4- PICTURE loss is defined as $L_V = L_{vd} + \lambda_{vs} \times L_{vs}$, where λ_{vs} is the weight of the smoothness loss. PICTURE loss is added to the data and smoothness losses of unsupervised training.

6.1.2 Known Operators

The known operators are added to the network in the inference mode only due to the high computational complexity of unsupervised training (outlined in the next section). We employ two known operators to impose physically known constraints on the lateral displacement.

The first known operator (we refer to it as Poisson's ratio clipper) limits the EPR to the feasible range of $v_{emin} - v_{emax}$. Although PICTURE tries to move all EPR values to the feasible range, in [79], it was shown that some samples in test time were still outside of the feasible range. Poisson's ratio clipper is an iterative algorithm since the lateral strains are altered by clipping the EPR values and affecting the neighbor samples' strain values.

The second algorithm employs the incompressibility of the tissue which can be formulated by:

$$\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 0 \qquad (6.4)$$

In free-hand palpation, the force is approximately uniaxial ($\varepsilon_3 \simeq \varepsilon_2$); therefore Eq 6.4 can be written as:

$$\varepsilon_1 + 2 \times \varepsilon_2 = 0 \qquad (6.5)$$

Guo *et al.* enforced incompressibility in an iterative algorithm [75]. We made a few changes to increase the method's robustness by adding Gaussian filtering in each iteration. It should be noted that the algorithm can be employed for compressible tissues as well, and the incompressibility constraint is employed for the refinement of the obtained displacement. The proposed algorithms are outlined in Algorithm 1 and 2. The network architecture with the known operators is illustrated in Fig. 6.1. It is worth highlighting that known operators offer a compelling alternative to regularization. While the latter involves adjusting trained

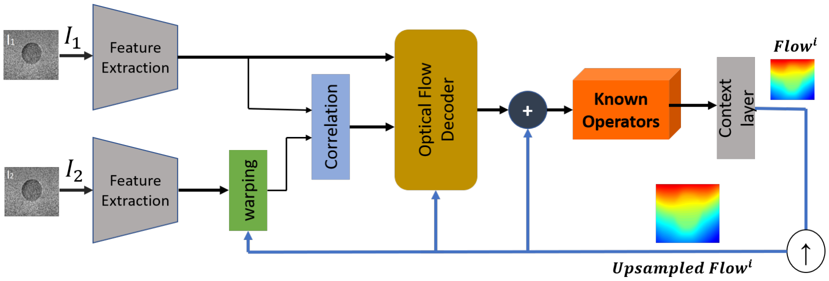


Figure 6.1: MPWC-Net++ architecture with known operators. The network is iterative with 5 pyramid levels. The known operators are added after optical flow estimation, and refine the estimated lateral displacement in each pyramid level (added from level 3) to provide improved lateral displacement to the next pyramid level.

weights based on the training data and keeping them fixed during testing, the former relies on iterative refinement that is adaptable to the test data and does not require any learnable weights.

Algorithm 1: Poisson's ratio clipper

```
input  : Lateral displacement $w_l$, axial displacement $w_a$, $v_{emin}$,$v_{emax}$, $iteration$
output: Refined lateral displacement $w_{ref}$
$w_{ref} \leftarrow w_l$
for $q \leftarrow 1$ to $iteration$ do
    $e_{22} \leftarrow \frac{\partial w_l}{\partial l}$                    // gradient in lateral direction.
    $e_{11} \leftarrow \frac{\partial w_a}{\partial a}$                    // gradient in axial direction.
    $epr \leftarrow \frac{-e_{22}}{e_{11}}$
    $epr(epr < v_{emin}) \leftarrow v_{emin}$                  // Clip epr less than $v_{emin}$
    $epr(epr > v_{emax}) \leftarrow v_{emax}$                  // Clip epr less than $v_{emax}$
    $w_{ref}(:, 2\ to\ end) \leftarrow w_{ref}(:, 1\ to\ end - 1) + epr \times e_{11}$
        // use the displacement of previous line and the clipped epr to
    find the displacement of the next line
```

6.1.3 Unsupervised Training

We followed a similar unsupervised training approach presented in [79] for both PICTURE and kPICTURE methods. The loss function can be written as:

$$Loss = L_D + \lambda_S L_S + \lambda_V L_V \tag{6.6}$$

Algorithm 2: Guo *et al.* refinement [75] employed as known operator

```
input : Lateral displacement $w_l$, Axial displacement $w_a$ of size $w \times h$, $iteration$, $\lambda_1$,
        $\lambda_2$
output: Refined lateral displacement $w_{ref}$
$w_{ref} \leftarrow w_l$
for $q \leftarrow 1$ to $iteration$ do
    for $i, j$ in $w, h$ do
        $\delta = W_l(i,j-1) - 2W_l(i,j) + W_l(i,j+1) + W_a(i+1,j+1) - W_a(i-1,j) -$
          $W_a(i,j-1) + W_a(i-1,j-1) + \lambda_1(W_{ref}^{q-1} - W_{ref}^{q-2})$
        $w_{ref}^q = Gauss(w_{ref}^{q-1} + \lambda_2 \times \delta)$  // Gaussian filtering to reduce noise,
          $\lambda_2$ controls the weight of updating $w_{ref}^q$
```

where L_D denotes photometric loss which is obtained by comparing the pre-compressed and warped compressed RF data, L_S is smoothness loss in both axial and lateral directions. λ_S and λ_V specify the weights of the smoothness loss and PICTURE loss, respectively.

6.1.4 Network architecture and training

We employed MPWC-Net++ [59] which has been adapted from PWC-Net-irr [58] for USE. The network architecture with the added known operators is shown in Fig. 6.1. The training schedule is similar to [79], known operators are not present in the training and only employed during the test phase. The known operators are added in different pyramid levels. This has the advantage of correcting lateral displacements in different pyramid levels. The known operators are added to the last 3 pyramid levels (there are 5 pyramid levels in this network) since the estimate in the first 2 pyramid levels are not accurate enough and adding the known operators would propagate the error.

6.2 Results and Discussions

6.2.1 Compared methods

kPICTURE is compared to the following methods:

- OVERWIND, an optimization-based USE method [29].

Table 6.1: Quantitative results of lateral strains for experimental phantoms. Mean and standard deviation (±) of CNR (higher is better) and SR (lower is better) of lateral strains are reported. The pair marked by † is not statistically significant (p-value>0.05, using Friedman test). The differences between all other numbers are statistically significant (p-value<0.05).

	sample (1)		sample (2)		sample (3)		*in vivo data*	
	CNR	SR	CNR	SR	CNR	SR	CNR	SR
OVERWIND	11.34± 1.32	0.318±0.030	3.71±1.07	0.505±0.089	3.61 ±0.58	0.415±0.050	2.07±0.94	0.196±0.255
OVERWIND + Gou *et al.*	13.26 ± 1.89	0.313 ± 0.029	4.28±1.31	0.503±0.083	4.08±0.62	**0.411**±0.049	2.39±0.89	**0.170**±0.233
PICTURE	9.037±0.88	0.407±0.022	5.37±1.33	0.449±0.060†	1.63±0.95	0.840±0.077	4.36±1.81	0.334±0.149
kPICTURE	**24.40**±7.02	**0.290**±0.038	**7.81**±1.68	**0.446**±0.056†	**5.49**±2.20	0.598±0.123	**5.54**±2.54	0.504±0.141

- The post-processing method of Guo *et al.* [75], which employs the output of OVERWIND as the initial displacement (OVERWIND+ Guo *et al.*).

- PICTURE, which penalize EPR values outside of feasible range [79].

We decided to compare with PICTURE instead of sPICTURE [91] (PICTURE with self-supervision) since self-supervision is not related to the physics of motion. To focus on the effectiveness of the known operators, we, therefore, provide a comparison to its corresponding method PICTURE. We also employed a similar hyper-parameters and training schedule for experimental phantom and *in vivo* data.

6.2.2 Results and Discussions

The lateral strains of ultrasound RF data collected from three different locations of the tissue-mimicking breast phantom are depicted in Fig. 6.2, and the quantitative results are given in Table 6.1. Visual inspection of Fig. 6.2 denotes that the method proposed by Gou *et al.* [75] improves the displacement obtained by OVERWIND. For example, the inclusion borders in sample 2 are much more clearly visible. The strain images obtained by kPICTURE have a much higher quality than those of PICTURE. Furthermore, kPICTURE has the highest quality strain images among the compared methods. For example, the inclusion on the bottom in sample 1 (highlighted by the arrows) is clearly visible in kPICTURE, a substantial improvement over all other methods that do not even show the inclusion.

The histograms of EPR values of OVERWIND+Gou *et al.*, PICTURE and kPICTURE are illustrated for the experimental phantom sample (1). To improve visualization, OVER-

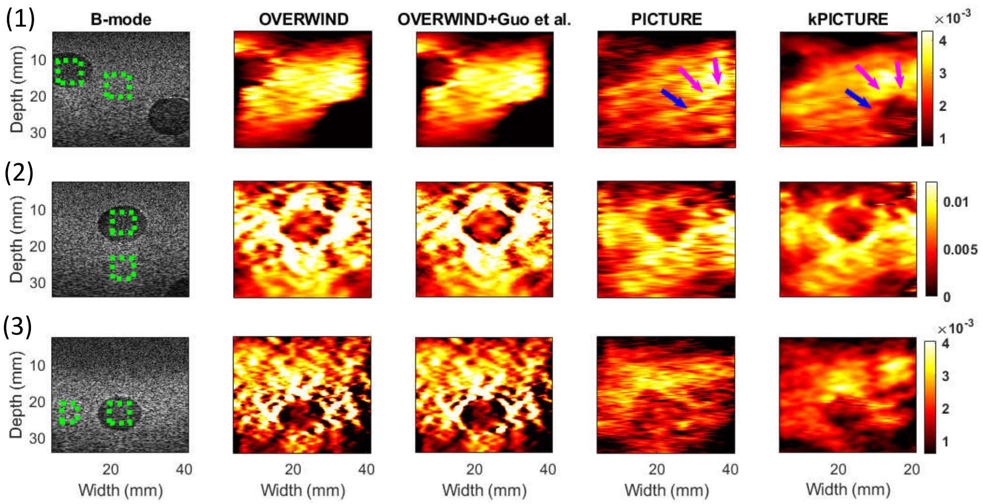


Figure 6.2: Lateral strains in the experimental phantom obtained by different methods. The target and background windows for the calculation of CNR and SR are marked in the B-mode images. The inclusion on the bottom of sample (1) is highlighted in PICTURE and kPICTURE strain images by purple and blue arrows. The samples 1, 2, and 3 are taken from different locations of the tissue-mimicking breast phantom.

WIND results are not included because the histogram was similar to that of OVERWIND+Gou *et al.*. Although PICTURE limits the range of EPR using a regularization (Eq 6.2), some EPR values are outside the feasible range. kPICTURE further limits the EPR values; only a small number of samples are outside of the physically plausible range.

The lateral strain results of *in vivo* data are depicted in Fig. 6.3 (b). While PICTURE may produce an adequate strain image, it still contains noisy regions. On the other hand, kPICTURE delivers exceptionally refined strain images and surpasses the other compared methods. The quantitative results given in table 6.1 also confirm the visual inspection.

It should be noted that after incorporating the known operators, the inference time of the network increased from an average of 195 ms to 240 ms (having 10 iterations for algorithm 1 and 100 iterations for algorithm 2).

6.3 Summary

In this chapter, we proposed to incorporate two known operators inside a USE network. The network is trained by physically inspired constraints specifically designed to tackle the long-standing illusive problem of lateral strain imaging. The proposed operators provide a

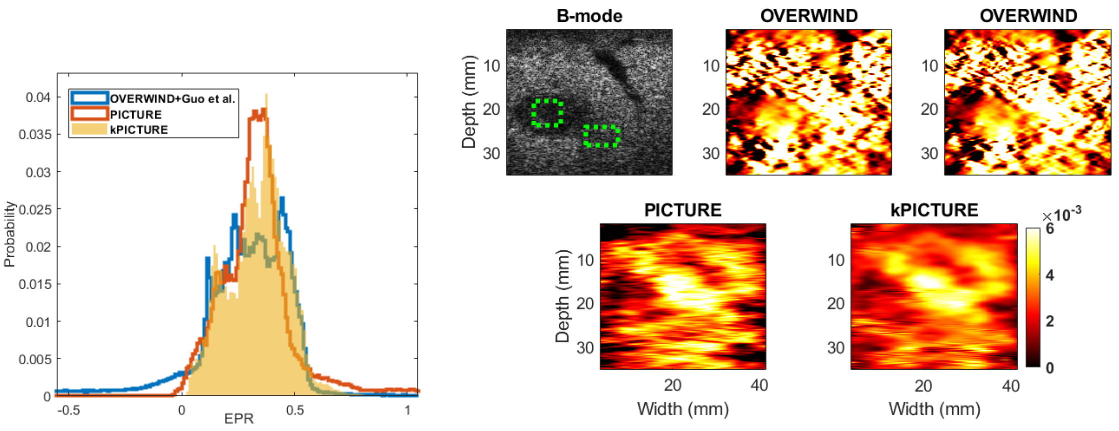


Figure 6.3: The histogram of EPR values for experimental phantom sample 1 (a). The *in vivo* results of the compared methods (b).

refinement in each pyramid level of the architecture and substantially improve the lateral strain image quality. Tissue mimicking phantom and *in vivo* results show that the method substantially outperforms previous displacement estimation method in the lateral direction.

Chapter 7

Ultrasound Scatterer Density Classification Using Convolutional Neural Networks and Patch Statistics

7.1 Introduction

Ultrasound imaging is increasingly attracting the attention of researchers and clinicians due to being a real-time and non-ionizing imaging modality, and being less expensive and more portable compared to other medical imaging techniques. However, several types of artifacts make interpretation of ultrasound images difficult. Cells, collagen, microcalcifications, and other microstuctural components are often smaller than the wavelength of the ultrasound wave, and scatter the wave and create the granular appearance in B-mode images called speckles. The scattered signal from scatterers provides useful information about characteristics of the scatterers, which are highly related to the tissue properties. Quantitative ultrasound (QUS) measures the tissue characteristics by analysing the ultrasound signal [92–99]. It aims to provide quantitative estimations of tissue characteristics which cannot be otherwise obtained from the B-mode image. It has been employed in many different applications such as liver fibrosis and steatosis assessment [2, 100], bone quality measurement [1], breast tumor classification [15, 101] and cardiac tissue characterization [102]. Improving QUS techniques can eventually broaden the applications of this safe and cost-effective method in diagnosis and treatment of a large number of disorders.

QUS methods can be classified into two broad categories: spectral-based and envelope-based methods [5]. Parameters such as the backscatter coefficient and attenuation coefficient can be estimated by spectral-based methods, usually with a requirement of a reference phantom to remove system-dependent effects[7, 96, 98, 103]. In envelope-based methods, different characteristics of the tissue are usually estimated by analysing and modelling the envelope of the RF data by fitting a probability density function. The sample size, wave frequency, and the attenuation can affect the accuracy of the distribution modelling, and therefore its parameter estimations [4, 104].

Among different QUS parameters, the scatterer density has attracted a great attention. If there are many scatterers (more than 10 in a resolution cell (an ellipsoidal volume defined by - 6 dB point of the beam profile[13])), the envelope data is considered as a fully developed speckle (FDS), and when the number of scatterers is low, it is considered as under-developed speckle (UDS). Classifying scatterer density into FDS and UDS is very critical since for estimation of the QUS parameters, different assumptions must be taken for UDS regions. Disregarding the density of scatterers results in unreliable estimates of other QUS parameters [105]. Reliable classification of UDS and FDS can pave the way for differentiating tissues with many small scatterers from those with few strong scatterers, and potentially use them as disease biomarkers. In addition, many downstream ultrasound applications usually work better under FDS conditions, such as sensorless 3D ultrasound [106] and elastography [10]. Furthermore, the presence of UDS can also affect the accuracy and precision of other biomarkers that are currently being explored in different clinical settings, such as the evaluation of fat infiltration in the liver. Detecting this condition during data acquisition can help define ways to improve the accuracy of the biomarker by implementing spatial or angular compounding strategies to compensate for the limited samples of the scattering process under UDS [107].

The statistics of echo-envelope data, extracted by either model-based or model-free parameters, provide information about tissue properties. Model-based parameters try to fit a distribution to the envelope data. If envelope data is FDS, the RF data can be modelled by the Gaussian distribution; therefore, envelope values follow the Rayleigh distribution [92, 108, 109]. However, the Rayleigh distribution fails to model the envelope statistics of UDS regions and other distributions such as K-distribution[102], Homodyned K-distribution[14, 110] and Nakagami distribution[15] can be utilized. Among these, the Homodyned K-distribution is the most comprehensive but the most complex one that does not have a closed-form solution. The Nakagami distribution provides a good estimate of the envelope signal with low-complexity and is widely used in QUS studies.

The Nakagami distribution, applied to ultrasound data for the first time by Shankar *et al.* [15], can be used to describe different probability density functions, and to characterize various scatterer patterns in tissues. It has been shown to be useful in discriminating different scatterer and tissue types. The Nakagami image can depict tissue properties that are not visible in ultrasound B-mode images, and has been employed in several studies for tissue characterization [2, 3, 111, 112].

Model-free parameters such as the envelope signal to noise ratio (SNR), skewness (S) and entropy [108, 113] are statistical parameters that change with different scatterer distributions. Entropy-based parametric imaging is a QUS imaging technique, which uses a small sliding window throughout the image to measure the entropy (the overall level of variations) of the backscattered RF signal. It has been shown to be effective in differentiating tissues with different scattering properties, and can provide higher accuracy in a smaller patch size compared to Nakagami imaging [113].

DL techniques have been utilised in many fields of medical image processing. They have also proved useful in different ultrasound applications such as segmentation[114, 115], super resolution imaging [116–118] and elastography [55, 119, 120]. A few studies have also attempted to tackle the challenge of extracting quantitative measures from ultrasound images using DL techniques. Byra *et al.* [121] used Nakagami images to train a convolutional neural network for the task of breast lesion classification. Wang *et al.* [122] have proposed a 3D convolutional network for breast cancer detection. However, the appearance and even statistics of ultrasound images can change with changes in imaging parameters such as time gain compensation and focal points. Such changes are well studied in DL and are referred to as domain shift [123]. If not accounted for, domain shift renders DL estimates grossly inaccurate. In fact, this is one of the reasons that DL is less explored in QUS compared to other ultrasound applications.

In a recent work, we designed a CNN to classify FDS and UDS [124]. The CNN was fed with envelope data and the RF data spectrum from small patches of RF data, and was compared with a multi-layer perceptron (MLP) classifier, which used SNR and skewness as inputs. We used patches to analyse a small area of the image and therefore, to provide a high resolution. The effect of patch size was also investigated (with patches sized 5 to 10 × wavelength). The results showed that in small patch sizes, the CNN outperformed the MLP classifier, whereas for larger patch sizes, where the statistics of the patch could be reliably estimated, the MLP classifier outperformed the CNN.

In another recent work, we segmented simulated images with three different scatterer densities using a U-Net [114]. We found that the network was able to segment precisely when the intensity difference between the inclusion and the background was high and thus the network could associate the intensity to the scatterer density.

In [125], the mean scatterer intensity, which is another QUS parameter, was estimated for the whole image. They assumed that all regions have FDS, which is a limiting factor in real ultrasound images. In this study, we aim to classify FDS from UDS regions using CNNs in small patches (Note that the patch size is different for simulation and experimental phantom data) where classical statistical parameters commonly used in QUS studies cannot be estimated accurately. Our ultimate goal is to reveal tissue scatterer information similar to that of Table 7.1 using ultrasound envelope patches. The intermediate goal of this chapter is to quantitatively evaluate the scatterer density under arbitrary conditions (i.e. different imaging settings). We use the ultrasound envelope data as the input to the network, since statistics such as SNR and Nakagami parameters are histogram-based, meaning that they ignore image texture. We hypothesize that the texture of the ultrasound envelope image contains crucial information which can be useful to determine the density of scatterers.

We use a large amount of simulated data to train the networks, and test the networks on simulated and experimental phantom data. We show that the CNN networks are more robust to the domain shift [123] compared to statistics used in conventional QUS methods. We modify well-known classification networks such as MobileNet V2 [126], Inception [127], DenseNet [128] and ResNext [129] for the task at hand, and train them using simulation data. The aforementioned networks are tested on unseen phantom data which are being imaged by a different imaging setting. In the next step, we combine statistics with CNNs by two different methods: fusion strategy and deep supervision. Our contributions can be summarized as follows:

- Different CNN architectures are utilized to classify scatterer density using envelope data.
- A novel training strategy and input channel are proposed to avoid over-fitting on domain information which enabled us to classify ultrasound patches without any reference phantom.
- The networks are further improved by exploiting patch statistics.
- The three different classifiers of support vector machine (SVM), random forest and MLP are used to classify based on patch statistics.
- Experimental phantom data is employed to validate our work in different imaging settings.

7.2 Methods

In this section, we first describe different datasets we analysed. We then present the scatterer density classification methods developed in this work, which include both classical (SVM, random forest and MLP) and DL methods (CNN and CNN with patch statistics as additional inputs), and provide intuitions for using different inputs.

7.2.1 Data

We employed two different datasets to investigate the performance of our proposed methods as outlined below.

Simulation data

We simulated 200 phantoms of size 30 mm× 30 mm× 1 mm using the Field II pro toolbox [45], with the center frequency of 6.67 MHz. The sampling frequency was 100 MHz and the RF signals were then downsampled to 50 MHz.

We randomly distributed point scatteres in the phantoms. In 100 FDS phantoms, we included 16 scatterers per resolution cell. In the remaining 100 UDS phantoms, we included 2 scatterers per resolution cell. The resolution cell size was determined by calculating the correlation between the data and a moving window in different regions [130]. The size was 0.15 mm^2 at the focal point (The out of plane resolution cell size is not computed). We randomly cropped 5000 patches of size 256×32 (4.04 mm × 5 mm which is 17 and 21 × wavelength in axial and lateral directions, respectively) from different depths as the training set and 1000 patches as the validation set. For the test set, we simulated 20 more phantoms with a random scatterer density value of 2 or 16 ± 10% in order to make the test data more challenging. We randomly selected 500 patches from these phantoms as the test set to evaluate the methods. This dataset will be publicly available online at data.sonography.ai.

Experimental phantom

Three different phantoms were used to validate our method. The phantoms were of size 15cm× 5cm× 15cm, built from homogeneous mixture of agarose gel media and glass beads as scattering agents. The glass bead diameter range and bead concentration in the phantoms are reported in Table 7.1. For more information on construction details, the speed of sound and attenuation coefficient of these phantoms refer to [131]. The phantoms were imaged using an 18L6 transducer operating at 10 MHz frequency using an Acuson S2000 scanner (Siemens Medical Solutions, Malvern, PA) and we used envelope of RF data which was acquired using Axius Direct Ultrasound Research Interface [132]. There are 456 A lines, separated by 0.1242 mm and the depth is 40mm. The sampling frequency is 40MHz and the exact operation frequency was 8.89MHz. However, because of attenuation, the center frequency of the spectrum was lower. We computed the resolution cell size using correlation method at different depths and it varied between 0.284 mm^3 (at the top where resolution

Table 7.1: Characteristics of the experimental phantoms and their scatterer concentration per resolution cell using 18L6 transducer (the range shows the minimum and maximum values derived from different depths).

Phantom	Diameter of Random Scatterers (m)	Scatterers Concentration per mm3	Scatterers Concentration per resolution cell
A (High)	5-40	236	8.50-67
B (Medium)	75-90	9	0.32-2.55
C (Low)	126-151	3	0.11-0.85

was poor) and 0.036 mm^3 (at the focal point where resolution was the highest). This high variation of the resolution cell size can have an adverse effect on the classification, especially when this variation is not observed by the network during training. The numbers of scatterers per resolution cell for different depth are given in Table 7.1.

We used the experimental phantoms as the test data to evaluate the performance of different models optimized or trained on the simulation data. Phantom A (high concentration) belongs to the FDS class and Phantoms B (medium concentration) and C (low concentration) belong to the UDS class.

7.2.2 Classical Statistical Parameters

Several parameters have been proposed in literature for estimating the scatterer density in ultrasound images. SNR and skewness are among the most important parameters proposed to classify different scatterer densities:

$$\begin{gathered} R = SNR = \frac{\overline{A^v}}{\sqrt{\overline{A^{2v}} - (\overline{A^v})^2}}, \\ S = skewness = \frac{\overline{(A^v - \overline{A^v})^3}}{(\overline{A^{2v}} - (\overline{A^v})^2)^{1.5}} \end{gathered} \tag{7.1}$$

where A is the envelope of RF data, v is the signal power and $\overline{(...)}$ denotes mean operation. While in [92], v smaller than 1 was suggested due to having higher dynamic range and lower estimation error, Prager *et al.* proposed 1.8 as the optimal value [133] in terms of the estimation error. We analysed both recommended values of 0.5 and 1.8, and obtained significantly better results on the validation set using the 0.5 value (area under curve (AUC) of 0.894 vs. 0.876 when employing the MLP, and 0.802 vs. 0.794 when employing the SVM classifier). We, therefore, set v to 0.5 in this study.

When the patch size is big enough, the estimation error of R and S, and therefore the classification error based on these parameters is low. But for small-size patches, the classification becomes difficult [92, 124]. This is especially important in clinical applications where tissues are rarely homogeneous and a large patch may include different scattering properties [108, 110].

Entropy has been employed for scatterer density classification [134]:

$$E = \sum_{n=1}^{N} p(i) log[p(i)] \tag{7.2}$$

where E denotes entropy, and statistical histogram of the envelope data square is represented by p, and N is the number of bins for calculating the histogram, which is arbitrarily set to 100 in this study. Entropy increases as the density of scatterers increases (moving from UDS to FDS). The entropy measure is shown to be effective when using a small window for QUS analysis [113].

Another parameter that has been shown useful in estimation of scatterer density is the Nakagami model parameters m (a maximum likelihood estimator of the shape parameter) and T (a generalized likelihood ratio test statistic) [109]:

$$\begin{gathered} m = \frac{(\overline{A^2})^2}{var[A^2]}, \\ T = 2K(log\frac{m^m}{\Gamma(m)} + (m-1)[\overline{log(I)} - log(\overline{I}) - 1]) \end{gathered} \tag{7.3}$$

where A is the envelope data and Γ represents the Gamma function. I is a vector representing K independent and identically distributed samples of the intensity from a specific patch. Different values of m explain different properties; when the m parameter approaches 1, the distribution approaches the Rayleigh distribution. The m parameter above and below 1 represent post- and pre-Rayleigh distributions, respectively, which are forms of a more general family of distributions, called Rician [13, 15, 92, 135].

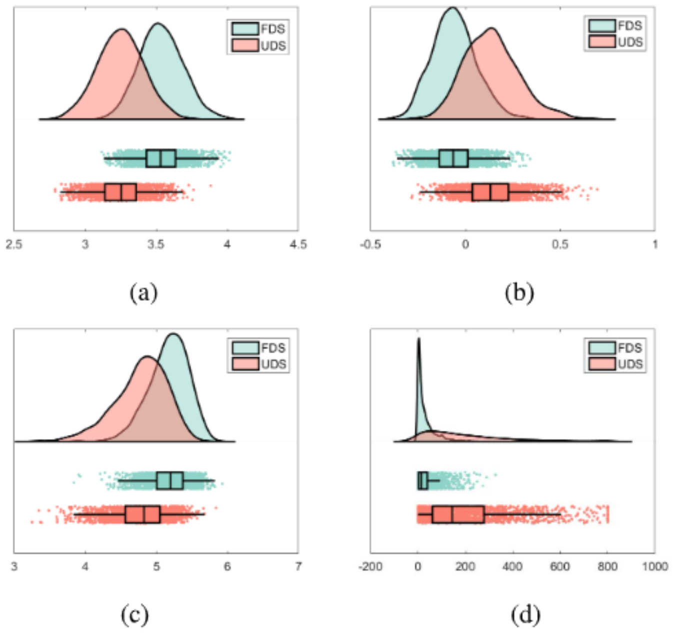


Figure 7.1: The distribution of the patch statistics for FDS and UDS in simulated training data. The patch size is small enough such that FDS and UDS classes overlap.

There is a strong correlation between features m and T. The features m and R are also highly correlated. We therefore, remove the feature m from the feature list to eliminate the redundancy between different features. We consider aforementioned parameters as a set of features to classify FDS and UDS patches.

Fig. 7.1 shows the distribution of different features, extracted from the simulation training data for UDS and FDS classes. The patch size is small so that for all features, a considerable overlap exists between the distributions of the two classes, which makes the classification highly erroneous using only a single feature. As opposed to our previous work [124] where only parameters R and S were used for classification, we use R, S, entropy and T together to obtain higher performance in classification. As shown in Fig. 7.1, the dynamic ranges of the features are not similar, hence we employ normalization (they are normalized to be in range 0-1) across each feature in the training data. The test and validation data were also normalized using the same coefficient obtained from the training data.

7.2.3 Machine Learning Methods

In order to classify FDS and UDS classes, we developed classical machine learning techniques in addition to DL methods. In this section, we describe the details of these classic techniques.

Support vector machine (SVM)

We used SVM as a classical machine learning algorithm to classify FDS versus UDS. We analysed different SVMs with linear and non-linear (Radial Basic Function (RBF) and polynomial) kernels. An SVM with an RBF kernel led to the best results on the validation set, and was selected throughout this chapter. We did a search to find optimum value of the C parameter of SVM, and subsequently, $C = 2.65$ was chosen.

Random forest classifier

Random forest is a learning method based on the decision tree algorithm and ensemble of different trees' outputs, and is among the top classification algorithms. By changing different parameters of a random forest model, we found the best performing model on the validation set, and used this model to classify different patches of simulation and experimental phantom data.

7.2.4 Deep Learning Methods

Multi-layer perceptron (MLP)

We proposed an MLP structure to classify FDS and UDS groups. To find the best network architecture for classifying scatterer density using the aforementioned features, we investigated the performance of different MLP architectures on the validation data. We obtained the best results with a 3-layer network. Further increase in the number of layers did not improve the results and lead to overfitting, a common problem with MLPs. We also analysed different numbers of neurons in each hidden layer. We incrementally increased the number of neurons in two hidden layers. Including 128 neurons in the first hidden layer, and 32 neurons in the second hidden layer led to the best result. However, it is important to note that the results reached a plateau and did not change substantially by changing the number of neurons. We employed Dropout [136] in the second layer. The activation functions were Leaky Relu for the first two layers and Sigmoid for the last layer. The loss function was binary cross entropy and the network was optimized using the Adam optimizer.

Convolutional neural network (CNN)

We used different state-of-the-art pretrained CNN networks to classify scatterer density. ResNext [129], MobileNet V2 [126], Inception [127] and DenseNet [128] were employed. In order to use these networks, we replaced the last fully connected layer with a fully connected layer with one output having Sigmoid activation function. We used the pre-trained weights of the networks for initialization to facilitate the training and avoid over-fitting.

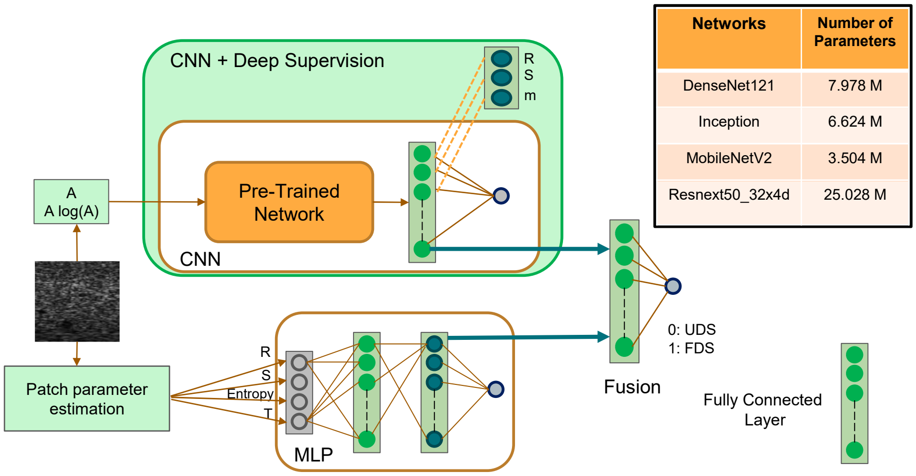
CNN with patch statistics as additional inputs

To further enhance the network, we proposed to utilize the patch statistics (R, S, entropy and T) as additional inputs. We tested different settings to determine the optimal way to fuse the information of patch statistics to the CNN. Fig. 7.2 shows the outcome (A denotes envelope). The CNN part is the same as the CNN network described in Section 7.2.4 and the patch statistic classifier part is similar to the MLP explained in Section 7.2.4. These parameters are fed to an MLP to generate a feature map, which is concatenated to the feature map obtained from the CNN. The resulted feature map is then used for a final classification.

Our first intuition was to train the whole network end to end. However, the CNN and the MLP have vastly different numbers of parameters and this resulted in a low generalization and a large sensitivity to the initial seeds. To mitigate the imbalanced number of parameters, we proposed training each part separately. We then trained the fusing part while the CNN and MLP weights were kept fixed.

Deeply supervised CNN (Multi-task learning)

Another way of using patch statistics is that using them as additional outputs. Multi-task learning [137] and deep supervision [138] have shown to improve the generality and performance of CNNs. We added R, S and m as additional outputs to force the networks learn in a way to have more generalization abilities. Values of R, S and m are highly



Networks	Number of Parameters
DenseNet121	7.978 M
Inception	6.624 M
MobileNetV2	3.504 M
Resnext50_32x4d	25.028 M

Figure 7.2: Proposed architectures for different networks. MLP, CNNs, CNNs with deep supervision and CNNs with fusion with MLP. The pre-trained networks with their number of parameters in Million are specified.

correlated with the scatterer density, which led us to train the network to learn these features as additional outputs. The loss function for the networks with additional outputs was defined as a weighted summation of scatterer classification loss and parameter estimation part which can be written as:

$$\begin{aligned} loss &= \frac{1}{N}\sum_{i=1}^{N}(y_i log(\tilde{y_i}) + (1-y_i)log(1-\tilde{y_i}) \\ &+\beta \times ((\tilde{R}-R)^2 + (\tilde{S}-S)^2 + (\tilde{m}-m)^2)) \end{aligned} \tag{7.4}$$

where y is the classification labels, $\tilde{y}$ is the predicted scatterer density, R, $\tilde{R}$, S, $\tilde{S}$, m and $\tilde{m}$ are ground truth and estimated values of SNR, Skewness and Nakagami parameter, respectively. N is the number of data in the corresponding mini-batch and β is the weight associated to the axillary loss. The classification performance is not very sensitive to β, which is set empirically to 0.2.

7.2.5 Training Schedule

To augment the data, random Gaussian noise, elastic deformation and random flipping in lateral direction were employed. The networks were trained with the Adam optimizer and the binary cross entropy was used as the loss function. Due to the fact that there were different networks with different inputs and to have a good generalization, we adopted a variant of early stopping which could be considered as a form of implicit regularization [139]. For early stopping, the validation AUC was selected as the stopping criteria; when the best validation AUC was reached during the training and remained the best after 20 epochs, we stopped the training. The cyclic learning rate was also used in order to avoid bad local minima [140].

Reduction of domain specific information

Batch normalization has been used widely in the classification networks. It has been found to facilitate the training and remove covariance shift [141]. During training, batch normalization layers compute the mean and standard deviation of the layer. The output of the layer is normalized by computed mean and standard deviation. During the test time, the computed mean and variance of the training data are used for normalization. The networks we employed contain several batch normalization layers. Domain information are mostly kept in the estimated mean and standard deviation of batch normalization layers [142]. In order to avoid learning domain specific information, we proposed not to update the mean and standard deviation of batch normalization layers during the training. We used pre-trained values of the mean and standard deviation for each batch normalization layers. The effect of batch normalization is studied in Section 7.3.3.

7.2.6 Input Channels

In [14], log compression of envelope along with the envelope have been used ($log(A)$ and $A^2 \times log(A^2)$) for estimating statistics using the Homodyned K-distribution. Inspired by their work, we used $A \times log(A)$ as a novel input to the proposed CNNs. We therefore used the amplitude A and $A \times log(A)$. The effect of including $A \times log(A)$) is studied in Section 7.3.3.

7.2.7 Evaluation Metrics

To evaluate the classification performance, we used AUC of the Receiver operating characteristic curve (ROC), accuracy, sensitivity (recall), precision and also Youden's Index [143].

Table 7.2: Simulation Results

Model	Fusion	DS	AUC	Sensitivity	Precision	Accuracy	Youden's Index
SVM	✗	✗	0.892 (0.867-0.920)	0.816	0.800	0.802	0.620 (0.51)
Random Forest	✗	✗	0.894 (0.868-0.919)	0.816	0.806	0.806	0.620 (0.04)
MLP	✗	✗	0.890 (0.860-0.917)	0.850	0.786	0.806	0.610 (0.52)
MobileNet V2	✗	✗	0.949 (0.929-0.964)	0.867	0.870	0.866	0.7529 (0.31)
MobileNet V2	✓	✗	0.905 (0.887-0.927)	0.831	0.830	0.828	0.656 (0.50)
MobileNet V2	✗	✓	0.950 (0.930-0.965)	0.847	0.885	0.866	0.755 (0.38)
Inception	✗	✗	0.969 (0.952-0.981)	**0.988**	0.823	0.886	0.825 (0.90)
Inception	✓	✗	0.970 (0.953-0.981)	0.906	0.913	0.908	0.837 (0.61)
Inception	✗	✓	0.945 (0.923-0.960)	0.914	0.857	0.878	0.766 (0.45)
ResNext50_32x4d1	✗	✗	**0.975** (0.957-0.984)	0.926	0.920	**0.920**	0.848 (0.47)
ResNext50_32x4d1	✓	✗	0.918 (0.893-0.939)	0.851	0.819	0.828	0.677 (0.52)
ResNext50_32x4d1	✗	✓	0.973 (0.956-0.984)	**0.988**	0.850	0.906	**0.855** (0.73)
DenseNet121	✗	✗	0.964 (0.947-0.976)	0.863	0.917	0.889	0.798 (0.16)
DenseNet121	✓	✗	0.947 (0.923-0.961)	0.925	0.840	0.872	0.759 (0.59)
DenseNet121	✗	✓	0.967 (0.952-0.978)	0.851	**0.923**	0.888	0.801 (0.45)

We estimated the 95% confidence interval of the metrics by employing boot strapping (i.e. sampling the data with replacement, for 1000 times). Youden's Index is a measure of both sensitivity and specificity:

$$\begin{aligned} J &= \frac{TP}{TP+FN} + \frac{TN}{TN+FP} - 1 \\ &= Sensitivity + Specificity - 1 \end{aligned} \tag{7.5}$$

where TP, FN, TN and FP denote true positive, false negative, true negative and false positive, respectively. It should be noted that accuracy, sensitivity and precision are reported for the threshold of 0.5. We calculated the Youden's Index for different threshold values and the highest values are compared between different models. The value of threshold which results in highest Youden's index can be considered as the optimal threshold value to have highest average of sensitivity and specificity.

7.3 Results

In this section, we provide the results of the proposed models for classification of FDS and UDS classes when analyzing different datasets. We tested three classifiers without including any CNN (i.e. a SVM, a random forest and an MLP model). Different CNNs were also analyzed, by exploiting the patch statistics using fusion (Fusion) and also deep supervision (DS). All DL models were trained according to the training schedule explained in Section 7.2.5. The weights of the top-performing networks will be publicly available online at code.sonography.ai.

7.3.1 Simulation Results

All proposed models were evaluated on the simulation data. Envelope and envelope multiplied by log compressed envelope are the input channels of all CNNs. Networks alone, fused with patch statistics (section 7.2.4) and with axillary outputs (section 7.2.4) are evaluated. The results are shown in Table 7.2. The values inside the parenthesis in the AUC column represent the confidence interval and the value inside parenthesis in Youden's index represents the threshold which results in the highest Youden's index. As seen in Table 7.2, CNN-based models provide better results compared to the MLP and SVM and random forest models which use only patch statistics. ResNext50_32x4d1 has the highest AUC (0.975) and accuracy (0.920). Whereas DenseNet121 with deep supervision has the highest precision (0.923). Both sensitivity and precision of CNN models are high meaning that the networks perform very well on identifying both positive and negative classes. The most of the optimum threshold of the Youden's Index was close to 0.5 which is the threshold value used for classification.

7.3.2 Experimental Phantom Results

The results of classifying small patches from phantom A vs. phantoms B and C are provided in Table 7.3. The patch size in terms of number of pixels is the same as the simulation data but it differs in terms of size in mm (4.92 mm× 4.28 mm). Due to the presence of domain shift, machine learning methods which use only patch statistics (SVM, random forest and MLP) have good precision but poor sensitivity. While, CNN models have high sensitivity and moderate precision (as opposed to simulation results where both sensitivity and precision were high). we expected that by adding patch statistics information, the sensitivity is decreased and precision improves. The effect of domain shift can be seen in the optimal threshold value of Youden's index. While in simulation the optimal values are close to 0.5, in experimental results most of them are far from 0.5.

Table 7.3: Experimental phantom results.

Model	Fusion	DS	AUC	Sensitivity	Precision	Accuracy	Youden's Index
SVM	✗	✗	0.646 (0.623-0.669)	0.235	**0.873**	0.733	0.350 (0.03)
Random Forest	✗	✗	0.895 (0.880-0.913)	0.389	0.821	0.768	0.710 (0.46)
MLP	✗	✗	0.887 (0.879-0.895)	0.303	0.558	0.744	0.716 (0.20)
MobileNet V2	✗	✗	0.941 (0.935-0.948)	**0.978**	0.562	0.738	**0.830** (0.98)
MobileNet V2	✓	✗	0.886 (0.878-0.894)	0.446	0.754	0.767	0.707 (0.11)
MobileNet V2	✗	✓	**0.947** (0.940-0.953)	**0.978**	0.530	0.703	**0.830** (0.99)
Inception	✗	✗	0.895 (0.886-0.901)	0.945	0.685	0.835	0.726 (0.51)
Inception	✓	✗	0.890 (0.883-0.897)	0.950	0.636	0.802	0.684 (0.54)
Inception	✗	✓	0.946 (0.939-0.952)	0.975	0.617	0.790	0.825 (0.96)
ResNext50_32x4d1	✗	✗	0.872 (0.864-0.880)	0.967	0.576	0.752	0.720 (0.94)
ResNext50_32x4d1	✓	✗	0.905 (0.898-0.913)	0.467	0.809	0.786	0.746 (0.17)
ResNext50_32x4d1	✗	✓	0.932 (0.924-0.938)	0.977	0.544	0.720	0.783 (0.98)
DenseNet121	✗	✗	0.875 (0.866-0.883)	0.875	0.688	0.770	0.701 (0.15)
DenseNet121	✓	✗	0.900 (0.893-0.908)	0.872	0.770	0.869	0.789 (0.27)
DenseNet121	✗	✓	0.918 (0.911-0.925)	0.871	0.768	**0.870**	0.776 (0.21)

Comparing the networks, MobileNet V2 has the highest sensitivity but the precision is low. Inception has the highest average of sensitivity and precision among the CNNs that do not exploit patch statistics. DenseNet121 achieves the highest precision (0.688) compared to other CNNs (without supervision or fusion). Exploiting patch statistics by fusion method increases the precision of the most of the networks (except Inception) and decreases the sensitivity in some models. By adding the patch statistics using deep supervision, the precision which is low in CNN models (0.562-0.688) was improved (0.628-0.809) with a slight decrease of sensitivity in some cases. Networks exploiting the patch statistics by deep supervision have higher AUCs compared to CNNs and CNNs with fusion. DenseNet + deep supervision achieves the highest accuracy among all the evaluated models.

7.3.3 Ablation Experiment

We conducted ablation experiments [144] (note that we are not referring to the thermal ablation treatment) on one of the networks (DenseNet121) to validate the input choice (Section 7.2.6) and training strategy (Section 7.2.5). The results are listed in Table 7.4. DenseNet121+BN denotes training DenseNet121 with updating batch normalization layers (training all layers). DenseNet121-Alog(A) represents that the network input is only envelope and the proposed input (Alog(A)) has been removed. Although precision of DenseNet121+BN

Table 7.4: Ablation experiment on DenseNet121. DenseNet121+BN : Batch normalization layers statistics are updated during training. DenseNet121-Alog(A) : Only envelope (A) is used as input and Alog(A) is not employed. DenseNet121: Proposed training strategy which batch normalization layers statistics are kept fixed during the training and Alog(A) is used alongside the envelope as an input channel.

	Simulation					Experimental Phantom				
Model	**AUC**	**Sensitivity**	**Precision**	**Accuracy**	**Youden's Index**	**AUC**	**Sensitivity**	**Precision**	**Accuracy**	**Youden's Index**
DenseNet121+BN	0.955 (0.935-0.969)	0.894	0.884	0.886	0.782 (0.23)	0.751 (0.738-0.762)	0.223	**0.755**	0.717	0.421 (0.02)
DenseNet121-Alog(A)	**0.966** (0.950-0.977)	**0.922**	0.900	**0.908**	**0.816** (0.57)	0.781 (0.771-0.790)	0.445	0.568	0.702	0.529 (0.02)
DenseNet121	0.964 (0.947-0.976)	0.863	**0.917**	0.890	0.798 (0.16)	0.875 (0.866-0.883)	**0.875**	0.688	**0.826**	**0.701** (0.15)

is slightly better than the proposed method (0.755 Vs 0.688), the sensitivity is very poor (0.223 Vs 0.875). Accuracy is also lower (0.717 Vs 0.826); therefore, it can be concluded that freezing batch normalization statistics is beneficial. Regarding the selection of the input channels, it can be observed that adding envelope multiplied by log compressed envelope deteriorates the simulation results. However, it substantially improves the experimental phantom results which indicates that adding Alog(A) can reduce over-fitting on the simulation data.

7.3.4 Visualizing Experimental Phantoms and Ablation Experiment Results

Fig. 7.3 depicts some examples of the studied images using different models. We split each image into overlapping patches (50% overlap), and feed all patches to the networks. As seen in Fig. 7.3, CNNs perform very well in classification of patches from phantoms A and C but they perform modestly for patches from phantom B which belongs to the class 0 (UDS). Whereas, MLP produces low probability for all three phantoms. The main reason is that the statistics are also system dependent. It can be observed that by using proposed training strategy, the CNNs have less system dependency and produce better results compared to MLP, which uses only patch statistics. Among the CNNs, the model with deep supervision (DenseNet121+DS) has the best results on the challenging phantom B with introducing slight error on the phantom A. DenseNet121+BN which mean and standard deviation of batch normalization layers are trained as well as other layers, produces poor results on phantom A that shows the benefits of using the proposed training strategy.

We should also mention that the networks estimate higher probability of FDS in the top and bottom regions of the phantoms. The main reason is that in these regions, the resolution cell size is larger than that at the focal region, leading to a larger number of scatterers per resolution cell.

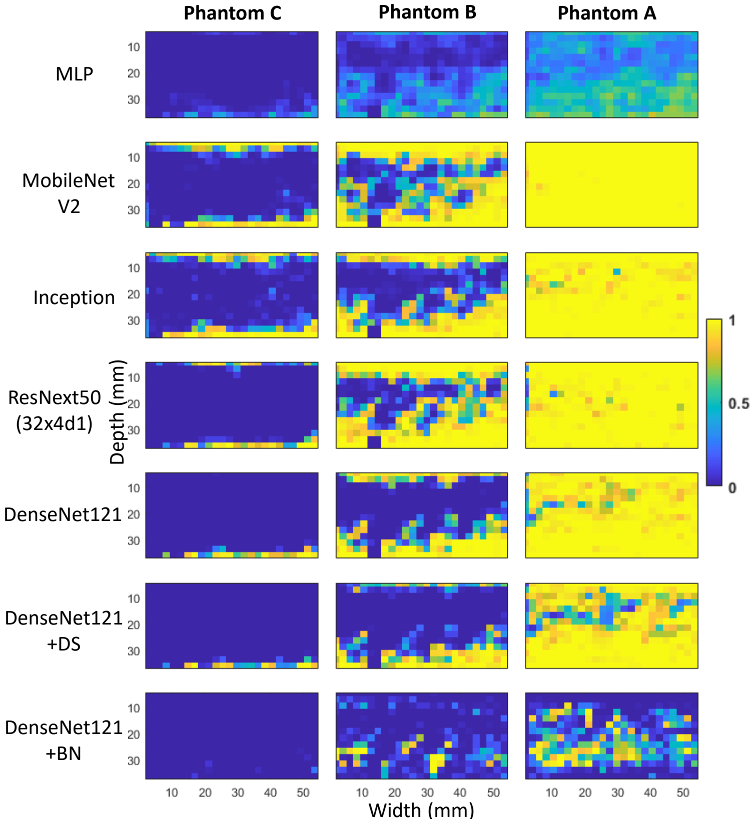


Figure 7.3: The results of MLP, MobileNet V2, Inception, ResNext50, DenseNet121, DenseNet121+DS (with deep supervision) and DenseNet121+BN (with updating batch norm coefficients) models on the experimental phantoms. The color code represents the predicted output of the networks, from 0 (UDS) to 1 (FDS). Correct classes are 0 (UDS) for phantoms C and B, and 1 for phantom A.

7.4 Discussion

The density of scatterers in different parts of a tissue is an important property of that tissue which may discriminate normal and abnormal regions. Ultrasound images can be utilized to estimate this property non-invasively. This will eventually guide invasive procedures such as biopsy, leading to less expensive and safer diagnosis methods for different types of diseases.

In this work, we employed DL techniques to classify the scatterer density in ultrasound images. Based on our results, different numbers of scatterers result in different texture patterns in the ultrasound image. We used CNNs to detect texture patterns and employ histogram-based features (SNR, skewness, entropy and T) to improve the performance. Regarding the choice of features, we only used histogram-based features since CNNs can detect texture-based features but they need a large receptive field (as large as the entire patch) to be able to detect histogram-based features. Further investigations may reveal effectiveness of other features for CNNs.

The network trained on simulation data was able to classify the experimental phantom data, despite the fact that the number of scatterers and the imaging properties are completely different in these two datasets.

Comparing the two methods of exploiting patch statistics, deep supervised networks have higher AUCs than the fusion method. In addition to this, deep supervision does not need calculation of the patch statistics separately which reduce computation overhead. The networks we employed have a large receptive field which enables them to estimate statistic information related to the entire patch. Networks with a small receptive field are not able to estimate these statistics as well as deeper ones.

In a fixed imaging setting, a larger number of scatterers results in a brighter ultrasound image. However, by changing the imaging machine settings, the image intensity can vary. Even though the average density of ultrasound images contains information about the scatterers concentration, it is not a reliable feature for classifying the number of scatterers, as it can be easily altered by changing the imaging setting. We eliminated the effect of the average intensity by normalizing each individual patch such that the intensity of all studied patches was in the range [0,1]. It should be noted that the normalization method is not robust for regions having saturation. Those regions should be excluded to be able to use the proposed method correctly. We also reduced the domain related information by avoiding updating statistic coefficients of batch normalization layers during the training. Using reference phantoms to reduce the system specific effects can be an area of future works.

The effective number of scatterers per resolution cell varies by depth. Generally, at the focal point, the resolution cell is the smallest. Therefore, there are fewer scatterers per resolution cell at the focal point compared to other regions.

We included the data recorded from three phantoms in this study. The density of the scatterers is not the only parameter which differs between these phantoms. The size of the included scatterers is also different (Table 7.1). However, considering the operating frequency, the size of the scatterers is still smaller than the wavelength, and does not substantially affect the results. Phantoms with different scatterer sizes and densities warrants further investigations.

7.5 Summary

In this chapter, we proposed different CNN models to classify small patches of ultrasound images as FDS or UDS. We proposed to use both envelope and envelope multiplied by log compressed envelope as two separate input channels to the proposed CNNs. We also proposed to freeze batch normalization layers during the training to avoid learning domain specific information. We further benefited from patch statistics using fusion and axillary outputs (deep supervision). We did not use any data from experimental phantom during the training phase, which can degrade the results due to the domain shift. Nevertheless, we found that CNN models result in high sensitivity but moderate precision for experimental phantoms, due to this domain shift. The precision is improved by using statistic information as additional inputs/outputs with a slight decrease in sensitivity. On a broad picture, these results show the ability of our model to learn scatterer density from simulation data alone without the need to perform transfer learning on experimental data.

Chapter 8

Robust Scatterer Number Density Segmentation of Ultrasound Images

8.1 Introduction

In conventional envelope-based methods, the envelope data is divided into small overlapping windows (a strategy here we refer to as patching), and an inverse problem approach is used to estimate the scatterer number density and other QUS parameters [14, 109]. The size of the patch should be large enough to contain adequate independent samples. Large patches reduce the spatial resolution and cannot identify small regions having different QUS parameters. The size of the patch is an important hyper-parameter that the user should optimize.

CNNs have been used rarely for QUS due to requirement of large training data and the system specific nature of quantitative applications. Recently, a patch-based CNN was developed by our group to classify regions in the ultrasound image according to their scatterer number density [145]. This prior work was one of the first reports on the use of CNN on QUS, particularly for scatterer number density classification and resulted in promising results (AUC as high as 0.975 in tissue mimicking phantoms). However, the patch-based nature of this work causes limitations especially when applying the method to real world scenarios where there are regions with irregular shapes with heterogeneous composition. The present work significantly advances QUS methods based on speckle statistics classification by avoiding the use of patches.

In this chapter, we segment the scatterer number density of the whole ultrasound image using a fully convolutional neural network. A simple method is introduced to generate a large dataset having different scatterer number density and mean scatterer amplitude shapes. The generated dataset enables us to classify samples of ultrasound envelope data without patching requirement. Furthermore, we uniformly sample the imaging parameters from a wide range to make the trained network robust to change of imaging parameters. We then use the Nakagami parameter in a multi-task manner to reduce over-fitting of the network. The idea of the reference-based methods [96, 103, 109, 146] is also adapted for CNNs to further boost the performance. We validate our method using simulation data, experimental phantoms and *in vivo data*. Our contributions are summarized as follows:

- The scatterer number density is estimated for the whole image instead of estimating for each individual patch. To the best of our knowledge, this is the first approach without patching for segmentation based on scatterer number density.

- The proposed method is evaluated for a variety of different computational and physical phantoms and *in vivo* breast ultrasound, imaged with different ultrasound scanners and transducers.

- Nakagami parametric imaging is employed as multi-task learning to improve the performance of network.

- The reference phantom idea of conventional methods is adopted to CNN for domain transformation to further enhance the performance using only a small amount of phantom data.

- The proposed method is compared with a previously developed reference-based method and a patch-based CNN recently proposed by our group.

- A fast and simple ultrasound image generation is employed which enables us to generate thousands of images with diverse imaging and scatterer properties for training our segmentation network. The generated dataset contains different scatterer number density and mean scatterer amplitude.

8.2 Methods

8.2.1 Background

Conventional methods to evaluate the scatterer number density

Rosado-Mendez *et al.* used a window around the region of the interest (here, we call it a patch) and employed echo amplitude signal-to-noise ratio, SNR, defined in Eq 8.1, the generalize spectrum and the first order statistics of the phase information to classify ultrasound patches into different groups of low-scatterer number density, diffuse scattering and coherent scattering [109].

$$\text{SNR} = \frac{(mean[A])}{\sqrt{Var[A]}} \tag{8.1}$$

In Eq 8.1, A denotes envelope. The SNR increases with scatterer number density. If the patch is FDS, this value would be close to 1.91 [13]. A reference phantom with high scatterer number density was used to account for spatial variations of the resolution cell size due to diffraction effects. A patch of ultrasound envelope data was classified as FDS if the SNR value was close to the SNR value of the reference phantom patch from the same depth. If the SNR value was lower than the reference one, the patch was considered as UDS and if it was higher, it was considered as non-resolved periodicity. Here, we assumed that if the absolute difference was lower than 3% (it was set empirically) of the reference SNR, the region is considered as FDS. Our method is compared with this algorithm (here it is called "Reference Method") for experimental phantom evaluations.

Deep learning methods

In [124] , we proposed a CNN to classify scatterer number density of a small patch of envelope data. The network was fed with envelope data and the spectrum of RF data, and was compared with a MLP classifier, which used two statistical parameters (signal to noise ratio and skewness) as inputs. The method achieved a segmentation accuracy of 92.2% for simulation test data with a patch size of $1.5mm \times 1.5mm$. However, it was limited to a single imaging setting and was also a patch-based method that required to apply the network many times over multiple patches to generate the scatterer number density map of the whole image.

Furthermore, the variation of mean scatterer amplitude was not considered.

In [114], we used a U-Net to segment scatterer number density of ultrasound images. Simulation data having inclusions with different densities were generated. The results showed that the network was able to segment the simulation test data with 1 and 10 scatterers per resolution cell with a precision of 99.2% and 67.5% and sensitivity of 98.8% and 79.7%, respectively, provided that there was a noticeable difference in intensity of the corresponding regions. However, the change of amplitude due to mean scatterer amplitude was not considered.

Zhang *et al.* [125] recently used a U-Net to estimate the pixel-wise mean scattering intensity. They generated a dataset with random shapes and considered different values of mean scattering intensity for each region. They assumed that all areas were FDS; hence, the network was able to associate different values of intensity to the mean scattering intensity of the scatterer distribution. However, the FDS assumption does not hold for many organs and limits the generalization to FDS tissues.

In our recent work [145] (Chapter 7), we used state-of-art CNN architectures as well as patch statistics to classify scatterer number density of patches. We simulated a training dataset with a fixed imaging parameters using Field II [45] which is available online at `code.sonography.ai`. CNNs which employed envelope echo signals outperformed machine learning methods such as SVM, Random Forest and MLP which only used patch statistics. Fusion and multi-task learning (MTL) were also utilized to combine the information of the statistics and textures. MTL was shown to be an appropirate choice for unseen experimental phantom data. This method is used for comparison and is labeled as “Patch-based CNN”.

8.2.2 Datasets and Data Generation

Data Generation

A large and diverse dataset is required to train a fully convolutional neural network. In fact, one of the reasons that patch based methods for scatterer classification were developed was to reduce data requirements of CNNs.

In this section, a simple but effective data generation scheme is introduced. Ultrasound simulation tools such as Field II [45] can be used to generate ultrasound simulation data. These methods are computationally expensive but have been used to generate medium size datasets such as the dataset proposed in [145]. Field II takes several minutes on a typical machine to generate an image. A common trend to generate medium size datasets is to use clusters but it is still infeasible to generate very large datasets (more than 10,000 images).

We employ a simple method to generate a large number of images in a short amount of time. This data generation method shares some similarities with the method used in [125], in that it considers variations in mean scatterer intensity. The main difference of the method proposed here is that it also considers variations in scatterer number density. This makes the data set more realistic and suitable for scatterer number density segmentation. It should be mentioned that unlike Field II simulations, which are scatterer-based and defined in arbitrary coordinates, the approach here (and in [125]) is grid-based, where each discrete grid position is assigned scattering properties.

Assuming weak scattering (using the first order Born approximation), the interaction of scatterers with ultrasound waves can be modeled by a 2D linear time varying convolution [46, 147, 148].

$$r[a,l] = g[a,l] * h[a,l] + \eta[a,l] \tag{8.2}$$

where $g[a,l]$ is the scatterer echogenicity map, $h[a,l]$ is the spatially varying Point Spread Function (PSF) at axial and lateral positions a, l, and η indicates additive white Gaussian noise. The scatterer echogenicity in a specified location is a Gaussian random variable sampled from a Bernoulli distribution which can be written as [125]:

$$g(a,l) = K(a,l) \times A(a,l) \tag{8.3}$$

where $K(a,l)$ denotes a sample at the axial and lateral location a and l, respectively, from a Bernoulli distribution with value of 1 with the probability of p (probability of presence of a scatterer) and 0 with the probability of $1-p$. A denotes the amplitude of the scatterers and it is sampled from $N(\mu_s, \sigma^2)$ where N denotes normal distribution with variance of σ^2 and mean of μ_s which corresponds to the mean scatterer amplitude.

Equation 8.3 can be used for a phantom having fixed values of mean scatterer amplitude

(μ_s) and scatterer number density. In order to incorporate different shapes of scatterer number densities and mean scattering intensity, we define a scatterer number density binary mask (SC) and a mean scatterer amplitude binary mask (MS) to control $K(a,l)$ and μ_s as the following:

$$\begin{aligned} K(a,l) &= \begin{Bmatrix} K_1(a,l) & SC=0 \\ K_2(a,l) & SC=1 \end{Bmatrix} \\ \mu_s(a,l) &= \begin{Bmatrix} \mu_{s1}(a,l) & MS=0 \\ \mu_{s2}(a,l) & MS=1 \end{Bmatrix} \end{aligned} \tag{8.4}$$

where $K_1(a,l)$ and $K_2(a,l)$ are the Bernoulli distributions associated to the different values of scatterer number density binary mask. $\mu_{s1}(a,l)$ and $\mu_{s2}(a,l)$ are different values of mean scatterer amplitude assigned to the different values of mean scatterer amplitude mask. Inserting Eq 8.4 into Eq 8.3 leads to:

$$g = \begin{Bmatrix} \sum_a \sum_l K_1(a,l) \times A(a,l) & SC=0, MS=0 \\ \sum_a \sum_l K_1(a,l) \times A(a,l) & SC=0, MS=1 \\ \sum_a \sum_l K_2(a,l) \times A(a,l) & SC=1, MS=0 \\ \sum_a \sum_l K_2(a,l) \times A(a,l) & SC=1, MS=1 \end{Bmatrix} \tag{8.5}$$

The Eq 8.5 can be extended to more than 4 states by using non-binary SC and MS masks.

We consider the simplifying assumption that PSF ($h[a,l]$) is constant throughout the image. Therefore, the time varying convolution is converted to a time invariant one. The PSF can be modelled by a 2D Gaussian function modulated by a cosine function in axial direction [147, 148].

$$h[a,l] = e^{-\frac{1}{2}(\frac{a^2}{{\sigma_a}^2}+\frac{l^2}{{\sigma_l}^2})} \times cos(2\pi f_c a) \tag{8.6}$$

where f_c, ${\sigma_a}^2$ and ${\sigma_l}^2$ denote the center frequency, axial and lateral width of the Gaussian profile of the PSF, respectively.

We generated 7000 random binary shapes and assigned different scatterer number densities and mean scatterer amplitude to each region. The imaging parameters σ_s, σ_l, f_c and speed of sound (v) are also sampled from a uniform distribution. The imaging parameters and their ranges are specified in Table 8.1. We generated 15000 images for training and validation using 6500 random binary masks, and an additional 500 test images using 500

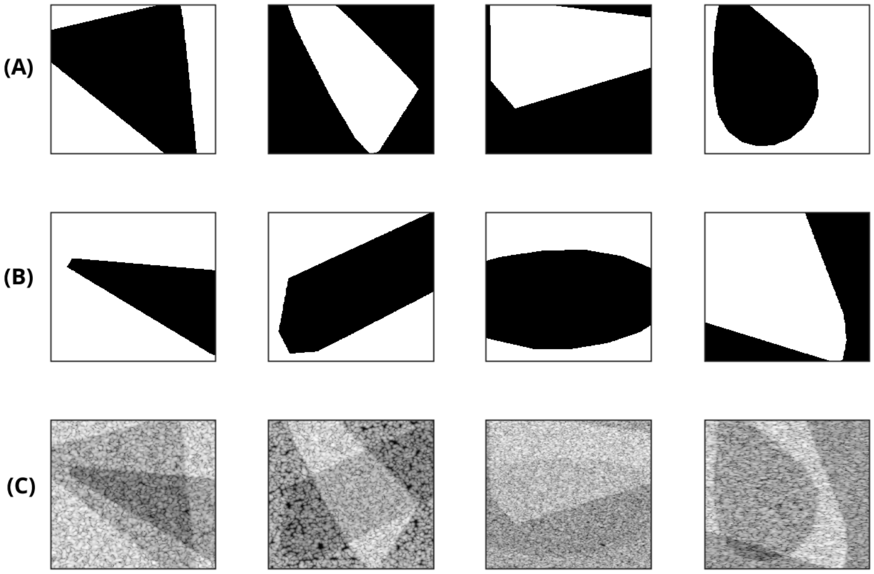


Figure 8.1: Four examples of the generated dataset. Scatterer number density mask (A), mean scatterer amplitude mask (μ_s) (B) and obtained B-mode image (C).

binary masks. The test binary masks are not used in training data generation to avoid data leakage. Some examples of the generated dataset are illustrated in Fig. 8.1. It can be seen from the figure that the intensity of the output image largely depends on both scatterer number density and mean scatterer amplitude; therefore, both of them must be taken into account when dealing with ultrasound images.

In [125], the ultrasound images were assumed to be FDS; therefore, they associated the intensity to the mean scatterer amplitude. In contrast to our recent work [114], it was assumed that all ultrasound images had the same mean scatterer amplitude; therefore, the images could be segmented by considering the intensity. In contrast to the previous works, the present method considers that both mean scatterer amplitude and scatterer number density can vary independently. Figure 8.2 part 1 depicts the training data generation step. The test set of this dataset will be available online after acceptance of this manuscript at `code.sonography.ai` similar to our previous works [55, 145].

Experimental Phantoms

Three homogeneous phantoms, A, B and C are employed for the evaluation (they were also employed in Chapter 7). They had a size of 15cm× 5cm× 15cm, made from a mixture of

Table 8.1: Parameters of image generation and their ranges.

Parameter	Range
Scatterer Number Density (UDS)	1-2
Scatterer Number Density (FDS)	11-16
Mean Scatterer Amplitude (μ_s)	0.3-1.3
Standard Deviation of Scattering Amplitude (σ_s)	0.03 (fixed)
Center Frequency (f_c)	4 - 7 MHz
Sampling Frequency (f_s)	60 - 100 MHz
Speed of Sound (v)	1510 - 1560 m/s
F Number	1.5 - 2.5
Number of Excitation Pulses	[3-5]
Standard Deviation of PSF in axial (σ_a)	0.1 - 0.3 mm
Standard Deviation of PSF in lateral (σ_l)	0.13 - 0.4 mm

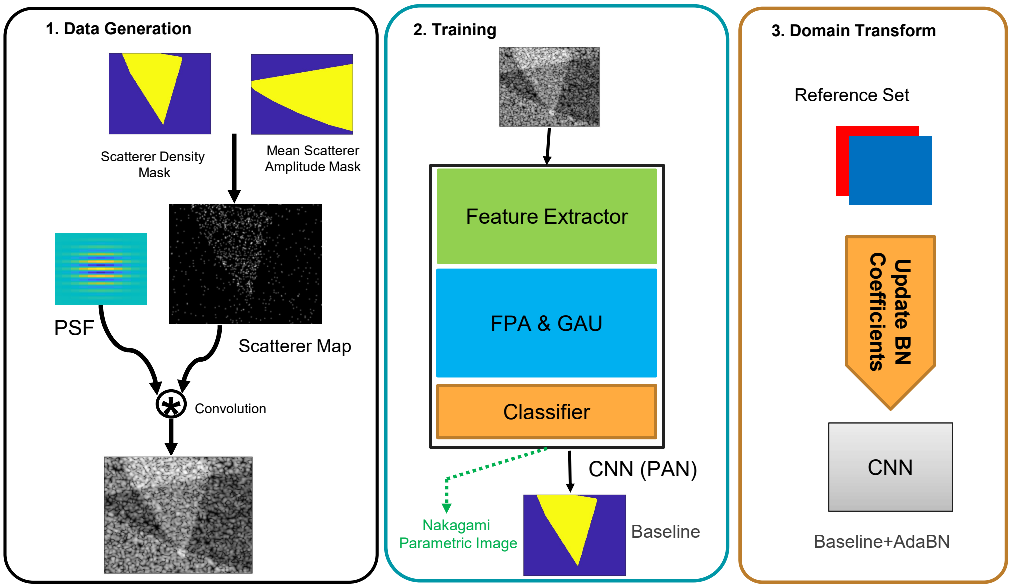


Figure 8.2: Overview of the framework used in this chapter.

agarose gel media and glass beads as scattering agents. For more information about these phantoms refer to [131]. An Acuson S2000 scanner (Siemens Medical Solutions, Malvern, PA) with an 18L6 transducer having the center frequency of 8.89 MHz was used to image the phantoms and RF data was acquired using Axius Direct Ultrasound Research Interface [132]. We used the correlation method to compute the resolution cell size [130]. The axial and lateral correlation cell size were 0.26 mm and 0.49 mm at the focal point, respectively.

Phantom with Inclusions (Phantom D)

This phantom was imaged by a Verasonics Vantage 128 system (Verasonics, Kirkland, WA) using a L11-5v transducer operated at 8 MHz. This phantom has three inclusions with different echogenicities and scatterer densities [146]. We computed the resolution cell size using correlation method [130], which was 0.149 mm and 0.237 mm at the focal point in the axial and lateral direction, respectively.

CIRS phantom (Phantom E)

A multi-purpose, multi-tissue CIRS phantom (model 040GSE, Norfolk, VA) was also used in evaluating the performance of the proposed method. It contains inclusions with no scatterers and different scatterer number densities. Data was collected with an E-CUBE 12 Alpinion machine using L3-12H transducer. The center and sample frequencies were 8.5 MHz and 40 MHz, respectively.

The background has high scatterer number density (FDS) and there are inclusions with different scatterer number densities. We selected this phantom to evaluate the proposed method for the cases where the scatterer number density was between the lowest (no scatterer) and the highest (FDS).

The phantom was also imaged using Verasonics Vantage 256 system (Verasonics, Kirkland, WA) using a L11-5v transducer. No reference phantom is available for this phantom; therefore, the proposed method is only compared with the patch-based CNN.

In vivo data

We used breast ultrasound images recorded by a Siemens Sonoline Elegra System (Issaquah, WA) with the sampling frequency of 36 MHz, center frequency of 7.5 MHz and a lateral beam spacing of 200 µm. I-Q echo data were recorded in a file on the imaging system when data acquisition was stopped (frozen on the imaging system). The IQ data were converted to RF echo data offline using the known demodulation frequency of the imaging system. More information about this dataset and the recording procedure is provided in [149].

8.2.3 Nakagami Parametric image

The Nakagami distribution is a flexible tool to model different scatterer patterns based on envelope amplitudes. It can be defined as [15]:

$$f(A, m, \Omega) = \frac{2m^m}{\Gamma(m)\Omega^m} A^{(2m-1)} \times e^{-\frac{m}{\Omega}A^2} \tag{8.7}$$

where A denotes the envelope amplitude, m represents the shape parameter, Ω is the scale parameter and Γ denotes the Gamma function. m is found to be correlated with scatterer number density, values close to 1 reflect high scatterer number density, and low values represent low scatterer number density. m can be estimated by the maximum likelihood method [15, 131]. The Nakagami parameter is also machine dependent meaning that machine settings can change the value of m for the same tissue [4].

In order to obtain the parametric image of m, patches of envelope data with overlaps are extracted and the m parameter is estimated using maximum likelihood estimator [109, 111]. The patches must be large enough to provide statistically reliable estimates of m. But very large patches reduce the spatial resolution of the Nakagami parametric image and might result in loss of information especially for small targets. We ensured that the window for estimation of the Nakagami parameter is at least 8 times larger than the resolution cell size. Some examples of the obtained parametric images are shown in Fig. 8.3, which shows that the Nakagami parameter is mostly sensitive to the c6hanges of scatterer number density, whereas brightness changes in the envelop images have little effects on Nakagami parameter. We used Nakagami parametric images in two different fashions; as the input of the network and as an auxiliary output for multi-task learning [137].

8.2.4 Network Architecture and Training

Semantic segmentation networks are able to segment more than 200 different classes within an image. Among different networks, the Pyramid attention network (PAN) [150] has shown its superior performance over other well-known networks such as PSPNet [151] and DeepLabv2 [47]. We used this network for our segmentation task. The network is composed of three main blocks.

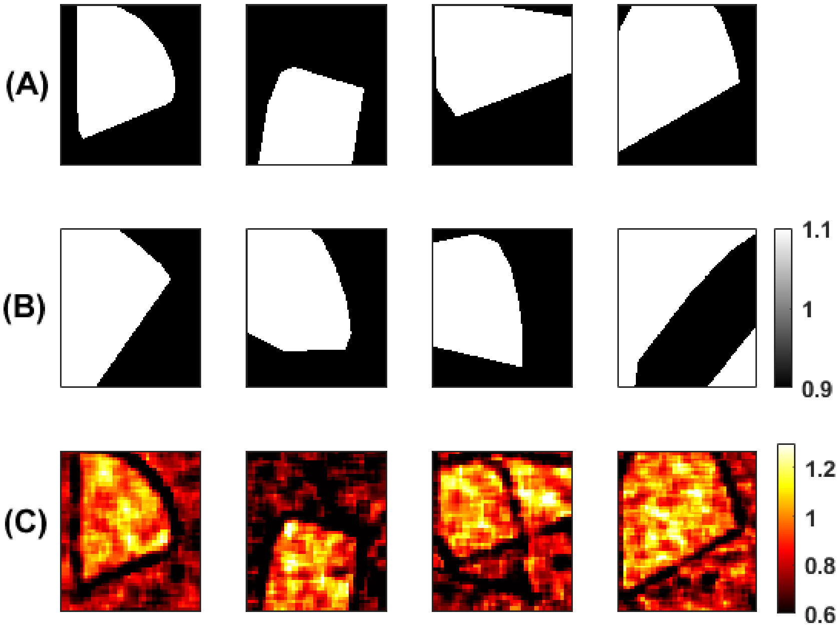


Figure 8.3: Four examples of the generated dataset. Scatterer number density mask (A), mean scatterer amplitude mask (μ_s) (B) and obtained Nakagami parametric image m (C). The mean amplitude values in these 4 samples are either 0.9 or 1.1.

1) Feature extractor: this module is employed to extract features from raw input images. ResNet50 [152] is utilized for the feature extraction. To avoid loss of information, features from different levels of ResNet50 are used to keep both spatial and semantic information.

2) Feature pyramid attention (FPA): This module is an attention module and it is used to have a precise pixel-level attention for high level features [151].

3) Global attention upsample (GAU): This module provides channel-wise attention mechanism to emphasize more important channels of low-level features using high-level features.

This network was employed to obtain scatterer number density segmentation with 1/4 of the original input resolution. It should be noted that the aim of this chapter is to investigate the performance of a state of the art CNN in scatterer number density segmentation, and not to compare different networks for this task.

The loss function for the segmentation task is the summation of binary cross entropy and the Dice loss. For the MTL network (with auxilary output), the loss is added by smooth L1 loss of the Nakagami parametric image. The loss for this network can be written as:

$$\begin{gathered} loss = BCE(\widehat{Ds}, Ds_{gt}) + Dice(\widehat{Ds}, Ds_{gt}) \\ +\beta|\widehat{m} - m_{gt}|_1, \end{gathered} \tag{8.8}$$

where BCE, Dice and $|.|_1$ denote binary cross entropy, Dice loss and smooth L1 norm, respectively. $\widehat{Ds}$, Ds_{gt}, $\widehat{m}$ and m_{gt} represent predicted, ground truth scatterer number density and predicted and ground truth Nakagami parametric image, respectively. β is the weight associated to the auxiliary loss (MTL) of the estimated Nakagami parametric image which is set to 0.1 to have lower weight than the main task. The auxiliary loss can be viewed as a regularizer that avoids over-fitting to the training data. The auxiliary task should be related to the main task; therefore, we selected the Nakagami parameter since it is highly correlated with the scatterer number density.

The networks were trained using the Adam optimizer for 20 epochs and the weights with the best validation results were used for evaluation. The learning rate was set to 1e-5 for the first 10 epochs and then reduced to 1e-6 for the last 10 epochs. It should be noted that we used feature extraction block pre-trained on ImageNet to speed up the training. The top performing network weights will be available online after acceptance of the manuscript at `code.sonography.ai`.

8.2.5 Batch Normalization and Adaptive Batch Normalization

Batch normalization (BN) has been found to be very useful in deep learning networks. It can speed up the training and remove covariance shift [153]. Having the input of the BN layer $X \in \mathbb{R}^{n\times c\times h\times w}$, where n is the number of data in a mini-batch, c, h and w denote the number of channels, feature height and width, respectively. The BN layer performs the following operation on X:

$$\begin{gathered} \widehat{x}_j = \frac{x_j - E[X_j]}{\sqrt{Var[X_j]}}, \\ y_j = \gamma_j \widehat{x}_j + \beta_j, \end{gathered} \tag{8.9}$$

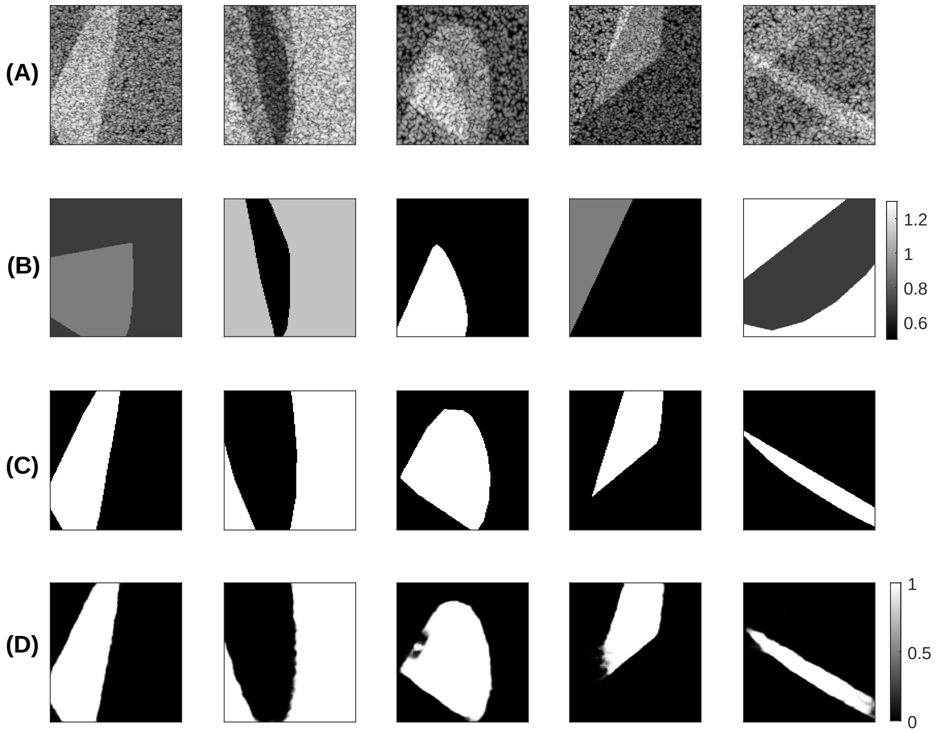


Figure 8.4: Four examples of the generated dataset with different imaging parameters. B-mode images (A), mean scatterer amplitude mask (μ_s) (B), scatterer number density mask (C) and the output of Baseline (D).

where j is index of channel, x_j and y_j denote input/output of the BN layer for channel j of one data sample. γ_j and β_j are learnable parameters that are optimized during the training. The mean ($E[X_j]$) and variance ($Var[X_j]$) of the mini-batch are computed in the training. During the test time, the computed coefficients in the training are used for normalization. By applying Eq 8.9, the distribution of all mini-batches remains the same. Considering a binary classification problem, if the size of mini-batch is too small that data belonging to only one class appears inside the mini batch, the obtained statistics would be biased toward that class which degrades performance. To mitigate this problem, other normalization techniques such as Group Normalization has been proposed for small mini-batch sizes [154].

Another aspect of BN is its impact on domain shift. The weights of the networks mostly preserve information about the classes, whereas, the BN coefficients contain information about the domain of the training data [142]. Motivated by this, the adaptive batch normalization (AdaBN) has been proposed for domain adaptation. The basic idea of AdaBN is that the coefficients of BN obtained during training are not suitable for the test data if

Table 8.2: The mean and standard deviation of the simulation results of scatterer number density segmentation. Baseline-nm, Baseline-ni and Baseline are from PAN network.

	IOU			**Accuracy**			**Sensitivity**			**Precision**		
	top 100%	top 10%	top 5%	top 100%	top 10%	top 5%	top 100%	top 10%	top 5%	top 100%	top 10%	top 5%
U-Net	0.950±0.038	0.786±0.068	0.711±0.076	0.973±0.023	0.858±0.050	0.796±0.060	0.973±0.020	0.823±0.042	0.742±0.050	0.976±0.031	0.905±0.058	0.878±0.066
Baseline-nm	0.976±0.023	0.922±0.037	0.898±0.040	0.989±0.015	0.958±0.032	0.938±0.035	0.987±0.019	0.945±0.037	0.922±0.041	0.988±0.012	0.959±0.017	0.948±0.017
Baseline-ni	0.967±0.041	0.875±0.080	0.829±0.092	0.984±0.022	0.931±0.044	0.898±0.048	0.986±0.028	0.932±0.070	0.895±0.090	0.980±0.031	0.918±0.060	0.890±0.071
Baseline	**0.981±0.017**	**0.941±0.023**	**0.925±0.023**	**0.991±0.008**	**0.972±0.015**	**0.962±0.015**	**0.990±0.012**	**0.963±0.022**	**0.948±0.024**	**0.990±0.011**	**0.966±0.016**	**0.955±0.017**

there is a domain shift.

In AdaBN, the coefficients of the BN layers are updated using data belonging to the test domain [142]. The main difference between AdaBN and transfer learning (fine-tuning) is that the weights of the network are not altered; therefore, only a small amount of data belonging to the test domain is required to update the BN coefficient. We call this data "reference set" since it acts very similar to the reference phantoms used in QUS. The reference set must have a balanced amount of data from each class to avoid a biased mean and variance. The reference set can be as small as two frames of data for binary classification task. In fact, for domain transformation only parameters of Eq 8.9 are required to be learned. In our experiment, two frames from phantom A (class 1) and phantom B (class 0) were employed. This method of domain transformation requires very small amount of data and does not leads to over-fitting since only a few parameters are learned by back propagation. AdaBN is well-suited for our task since the effect of imaging parameters can be reduced using domain transformation techniques. It can be used to transfer the baseline to any machine setting by only small amount of data.

8.3 Results

8.3.1 Simulation Results

Ultrasound grid-based simulation

The test set of simulation data contains 500 test images obtained by the method explained in section 8.2.2. Intersection over union (IOU), accuracy, sensitivity and precision are employed to evaluate scatterer number density segmentation performance. Dice similarity score is

Table 8.3: The summary of the networks employed in this chapter. The parameters A, $Alog(A)$ and m denote envelope, envelope multiplied by log compressed envelope and Nakagami parametric image, respectively.

Name	Input	Multi-task	Training data
Baseline-nm	A and $Alog(A)$	No	Grid-based Simulation
Baseline-ni	A, $Alog(A)$ and m	No	Grid-based Simulation
Baseline	A and $Alog(A)$	yes (m)	Grid-based Simulation
Baseline-AdaBN	A and $Alog(A)$	yes (m)	Grid-based simulation + updating BN layers using experimental data

excluded since it is highly correlated with IOU. All metrics are reported for the whole test set, 10% and 5% of the test set having the worst results. The results are given in Table 8.2. The U-Net architecture is the same as [155]. "Baseline-nm" denotes the network (PAN) having A (envelope) and $Alog(A)$ (envelope × log compressed envelope) as input channels. "Baseline-ni" denotes the same network having Nakagami parametric image as well as A and $Alog(A)$ as input channels. "Baseline" is the proposed network with A and $Alog(A)$ as inputs but having Nakagami parametric image as an auxiliary output. A summary of different methods is presented in Table 8.3.

According to Table 8.2, PAN performs better than U-Net which is expected due to use of different attention mechanism. Adding Nakagami parametric image as an input channel (Baseline-ni) deteriorates the performance (compared with Baseline-nm); however, adding Nakagami parametric images as an auxiliary output (Baseline) improves the performance in all metrics. The lower performance of the network having Nakagami parametric image as an additional input is ascociated to the fact that the texture of Nakagami parametric image does not have valuable information about the scatterer number density, while, its value matters for prediction of scatterer number density. This result also agrees with our recent work [145] where we found that MTL performs better than adding the statistics as additional inputs. Some examples of predicted segmentation results by Baseline are depicted in Fig. 8.4. It can be seen that the network can segment the scatterer number density well even in presence of different values of mean scatterer amplitude which results in different intensities. It should be noted that in simulation results, there is no domain shift which leads to high performance in all compared methods. Figure 8.2 part 2 depicts the simulation training and test step.

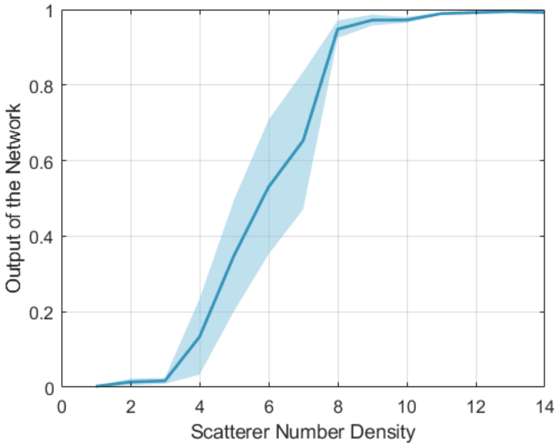


Figure 8.5: The mean and the 95% confidence interval range of the output of the network for the scatterer number density in the range 1 to 14.

The Network prediction for different values of scatterer number density

The training data only contained scatterer number densities of 1 to 2 scatterers per resolution cell for UDS and 11 to 16 for FDS. To investigate the performance of the algorithm when used on data with scatterer number densities from 1 to 14 scatterers per resolution cell, 10 phantoms were simulated with $\{1, 2, .., 14\}$, scatterers per resolution cell. Figure 6 shows the mean value and the 95% confidence interval (as the shading) of the probability of FDS produced by the network. It is clear that when the scatterer number density is close to 1 or 10, the confidence interval (the shaded area) is small while, for scatterer number densities in the range of 4-7, the confidence intervals are wider. The training data could have higher values of scatterer number density for UDS class. For instance, the UDS class could contain values of 4-6 scatterers per resolution cell. This would result in the reduction of the probability of FDS of the network for those values. However, we simulated the training data which contains the very low scatterer number density for UDS. In this way, the value provides some insights about how far the scatterer number density is from FDS.

Field II simulation

A numerical phantom is simulated using Field II simulation toolbox [45]. The transmit focus is at 15 mm, the F number is 2.12 and other imaging parameters are the same as [145]. The proposed method is compared with the patch-based CNN trained on the Field

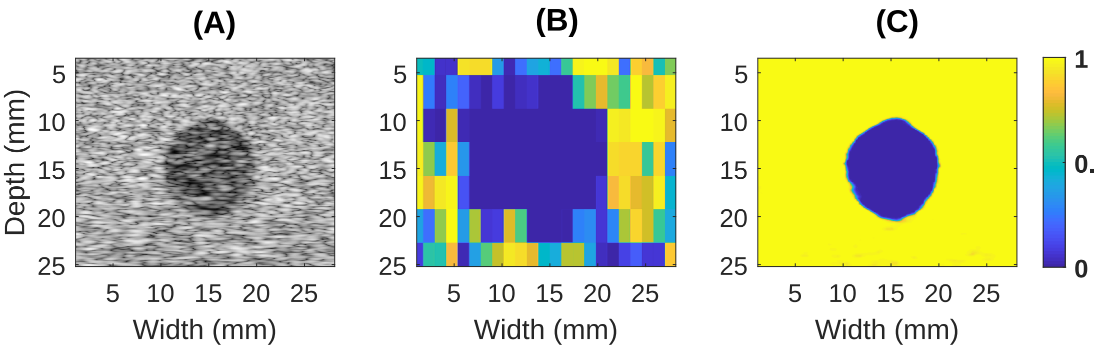


Figure 8.6: The Field II simulation results. B-mode image (A), the segmentation results of the patch-based CNN (B) and our segmentation method (C).

II dataset with the same imaging parameters. In contrast, the proposed method is only trained on the proposed fast simulation method. The aim of this evaluation is to investigate the performance of the segmentation network trained on the simplified fast US simulation on data obtained by the Field II toolbox. The phantom has an inclusion with 2 scatterers per resolution cell and the background having 11 scatterers per resolution cell. The results are illustrated in Fig. 8.6. The patch-based method cannot clearly identify the boundary of the inclusion, while our proposed method clearly detects the inclusion. Another point is that the patch-based CNN miss-classifies some of the patches outside of the boundary of the inclusion while, the proposed method obtains consistent segmentation of the phantom.

8.3.2 Phantom Results

The networks with MTL output (Baseline) and without it (Baseline-nm) are employed for the evaluation. The Baseline is also adapted to the new domain using AdaBN technique (Section 8.2.5) to evaluate the performance when BN layers are updated by the new domain (Baseline+AdaBN). Figure 8.2 part 3 illustrates the general framework used for experimental phantoms. We also used the recent reference phantom method [109] and the patch-based CNN [145] for comparison (Section 8.2.1).

Homogeneous Phantoms Results

Baseline, Baseline-nm, Baseline+AdaBN, patch-based CNN (DenseNet + deep supervision) and the reference methods are compared for Phantoms A, B and C. AUC, Accuracy and F1

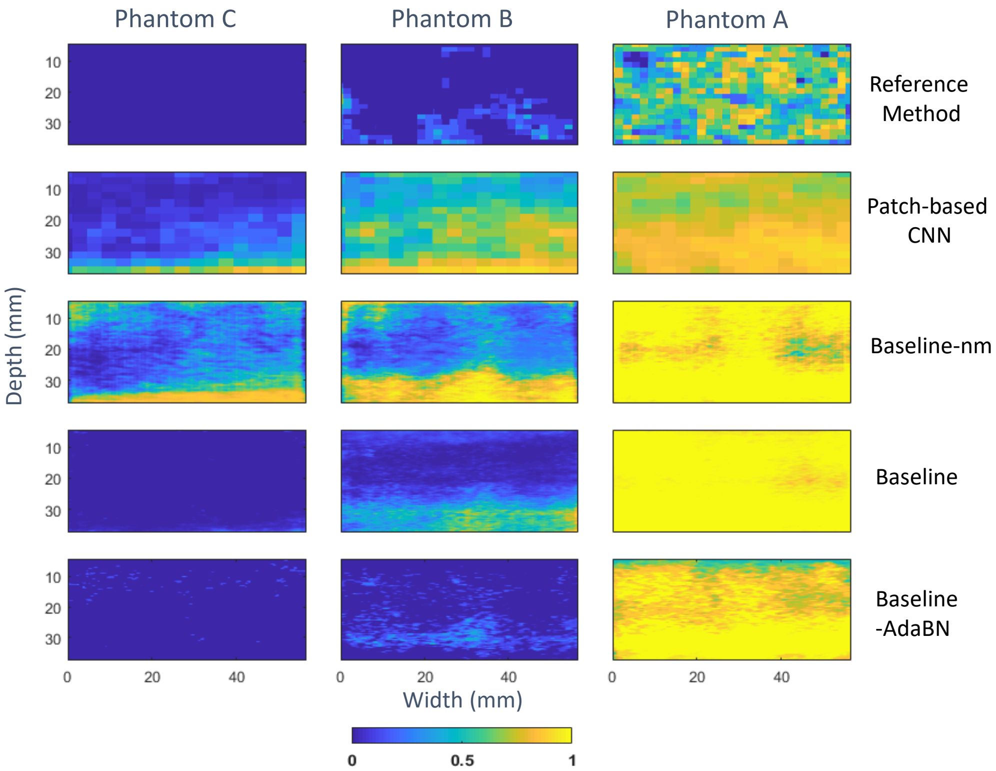


Figure 8.7: Homogeneous phantom results. The color code represents the predicted output of the networks, from 0 (UDS) to 1 (FDS). Correct classes are 0 (UDS) for phantoms C and B, and 1 for phantom A. Using Nakagami parametric image as an axillary output (Baseline) substantially improves the accuracy compared to the network without MTL (Baseline-nm).

$(2\frac{precision \times sensitivity}{precision + sensitivity})$ [156] are used as the quantitative metrics and given in Table 8.4. The average of the results over 8 frames are shown in Fig. 8.7. Adding the Nakagami parametric image (MTL) improves the results (compare Baseline and Baseline-nm) which demonstrates that MTL helps networks to be more robust to domain shift. It can be seen that the network trained only on the generated dataset performs well on this dataset without having any domain-specific information which demonstrates the strength of the proposed method. However, the network detected that the bottom of the phantom B to have relatively high probability of FDS.

The reference phantom used for this evaluation has the same scatterer number density as the phantom A (high). The imaging setting is also the same. The reference phantom method can discriminate well phantoms C and B. Also, it performs acceptable on phantom A. It should be mentioned that this method requires patching and each patch needs to be

Table 8.4: Homogeneous phantom results

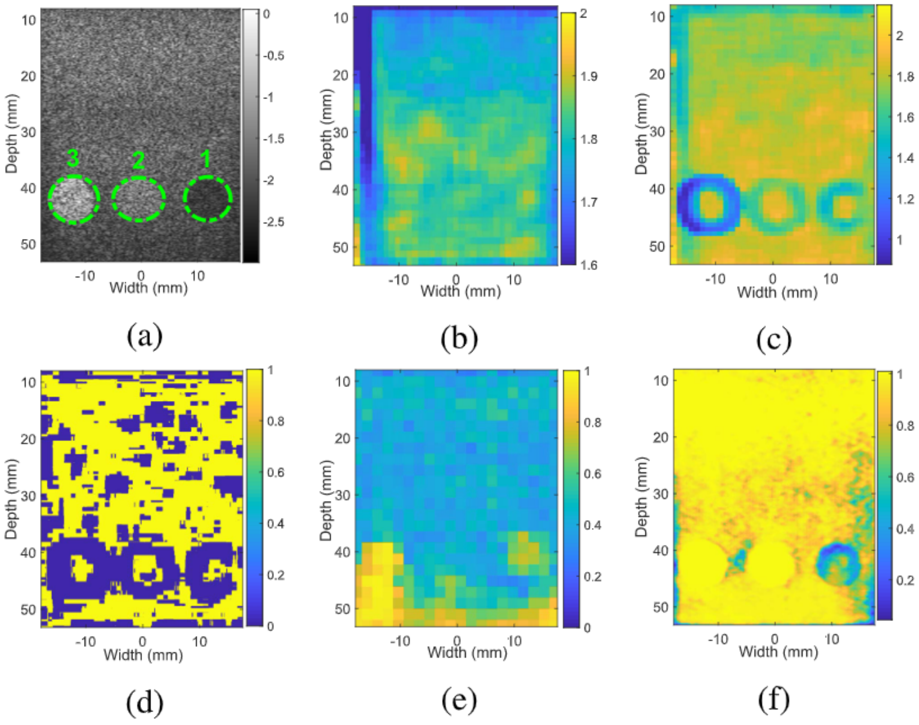
	AUC	**Accuracy**	**F1**
Reference Method	0.987	0.912	0.848
Patch-based CNN	0.780	0.740	0.678
Baseline-nm	0.974	0.797	0.766
Baseline	1.00	0.957	0.934
Baseline+AdaBN	1.00	0.999	0.998

compared with reference phantom patch from the same depth to consider its variations. Our network with adaBN (Baseline+AdaBN) performs better than Baseline especially for phantom B. Furthermore, it outperforms the reference method on phantom A. It should be noted that two frames of phantom A and B are utilized for updating the BN layer statistics and weights of network are kept fixed. The BN impact can be observed by comparing the Baseline and Basline+AdaBN. The only difference between these two networks is that in the latter BN statistics are updated by the test domain.

The reference method requires a reference phantom imaged by the same ultrasound machine which may be not available. However, the patch-based CNN and our proposed network trained on generated dataset (Baseline) performs well without any information about the test domain. Comparing the proposed method and the patch-based CNN, Baseline performs substantially better than the patch-based CNN especially on the phantom B which is more challenging than phantom C and A. This demonstrates that the proposed method is more robust to the change of domains compared to the patch-based CNN due to the fact that the network is trained using the training dataset with diverse imaging parameters.

Phantom D Results

The B-mode image of the phantom with inclusions is shown in Fig. 8.8 (a). This phantom contains three inclusions which are specified in the figure. The scattering agent of inclusions and the background are the same (mean scatterer amplitude is fixed); therefore, the intensity can be associated to the density of scatterers. We know *a priori* that the background is FDS. The inclusions 2 and 3 have higher intensity which means that the have higher scatterer number density than background; therefore, they are also FDS. The inclusion 1 has lower



(a) (b) (c)

(d) (e) (f)

Figure 8.8: The phantom D results. B-mode image (a), SNR (mean) of reference phantoms (b), SNR parametric image (c), reference phantom method (d), patch-based CNN (e) and the proposed method (Baseline) (f). Inclusions are specified in the B-mode image. Inclusion 1 belongs to UDS class and other parts belong to FDS class. The inclusions are numbered from the lowest to the highest scatterer number density.

density and it is UDS. For the reference method, we use other part of the phantom without inclusion having high scatterer number density imaged by the same machine as the reference phantom. The SNR parametric image of reference phantom is obtained by averaging the SNR parametric image of 10 frames. The SNR of the reference phantom is illustrated in Fig. 8.8 (b). We also obtained the average SNR of 12 frames of the phantom D which are depicted in Fig. 8.8 (c). The SNR value of the reference phantom is 1.725±0.225 which is expected for phantoms with high scatterer number density. However, SNR of phantom D is as low as 1 in some regions. The main reason is that SNR on the borders of the regions with different scatterer number densities, is not valid and reliable since on the borders of the inclusions the patch contains two different distribution and the calculated value is not reliable anymore. Therefore, patch-based methods usually fail in these regions. The output of the reference phantom method is shown in Fig. 8.8 (d). As anticipated, most of the background are correctly predicted as FDS. While, the inclusions 2 and 3 borders are incorrectly classified as UDS. It should be noted that the drop of the predicted scatterer number density between the inclusions 2 and 3 is an artifact. It can be caused by side lobes, off-axis scattering and having limited number of samples between the two inclusion. It is worth mentioning that in the grid-based simulation, the echogenicity is directly determined by the tissue scattering (mean scattering and the scatterer number density). Therefore, artifacts such as refraction effects caused by rounded boundaries and shadowing and enhancement caused by attenuation could provide unreliable classifications.

The Baseline method which is trained on simulation data, is employed for obtaining the predicted scatterer number density mask. The output segmentation mask of the proposed method average and standard deviation across 12 frames are shown in Fig. 8.8 (e) and (f). The patch-based methods correctly shows that the inclusion 3 has higher scatterer number density than the other regions. However, it fails to detect the lower scatterer number density of inclusion 1. Our proposed network correctly classifies the background and inclusions 2 and 3. Inclusion 1 can be also well discriminated from the other two inclusions. It should be mentioned that no reference phantoms have been employed for the proposed method. Without any reference phantom requirement, the proposed method outperforms the conventional reference phantom method which shows the potential of CNNs in QUS analysis.

8.3.3 Phantom E Results

The average of the results over 25 frames are shown in Fig. 8.9. As shown in the B-mode image, there are 4 inclusions. The first inclusion is anechoic, while the other inclusions have fewer scatterer number density compared to the background. The inclusions are numbered from the lowest scatterer number density to the highest. The patch-based CNN can detect the approximate location of inclusions 1, 2 and 3. However, the boundaries of them are not accurately identified due to the finite size of the patch. The proposed method provides substantially higher quality scatterer number density segmentation compared to the patch-based CNN.

It should be mentioned that the network is also able to discriminate the inclusions 1 and

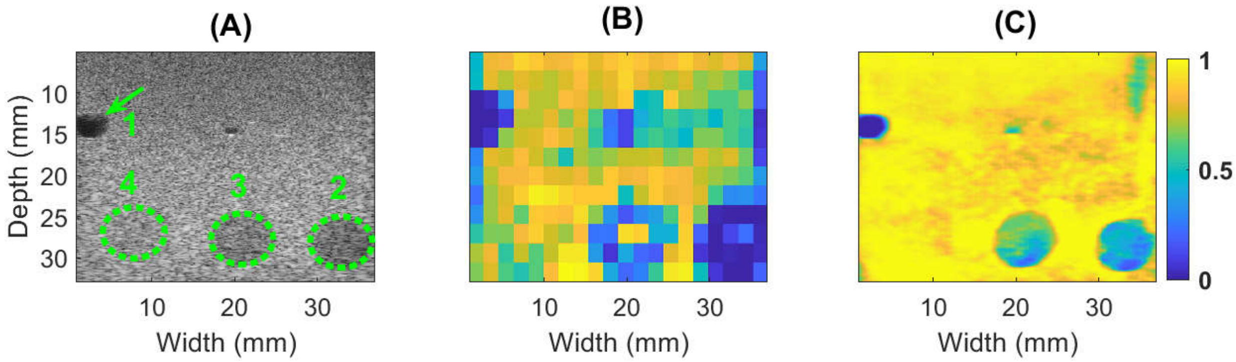


Figure 8.9: Scatterer-density segmentation of the CIRS phantoms scanned with the Alpinion system. (A) B-mode image, (B) segmentation result using the patch-based CNN [145], and (C) segmentation using the patch-less CNN (Baseline). Inclusions are numbered from the lowest to the highest scatterer number density. The color bar indicates probability of FDS.

2 in Fig. 8.9. The average of the network's output for the inclusion 1 and 2 are 0.441 and 0.577, respectively, which indicates that the network gives higher probability of FDS for the inclusion that has the higher scatterer number density. This can be employed to characterize the US images having scatterer number densities between 0-10 within the resolution cell.

8.3.4 *In vivo* Results

The patch-less CNN segmentation was tested on echo signals acquired from two breast fibroadenomas. Reference phantoms with known scatterer number density imaged using the same machine setting are not available for this dataset. According to [3], the Nakagami m parameter, which is highly correlated with scatterer number density, of fibroadenomas tends to be lower than values in normal breast tissues, which have high scatterer number density close to FDS limit (here, we only considered fat regions and other regions such as ductal cells are excluded from our study). Therefore, the network should be able to discriminate fibroadenomas and normal breast tissue. The results are presented in Figs. 8.10 and 8.11. The network classifies normal breast tissue as FDS; while, the regions of the fibroadenomas are segmented as UDS. The patch-based CNN method was also used in this data but it failed at detecting the lesion from the background. It should be noted that knowing the exact ground truth scatterer number density of the *in vivo* data requires acoustic microscopy image analysis which is out of scope of this work.

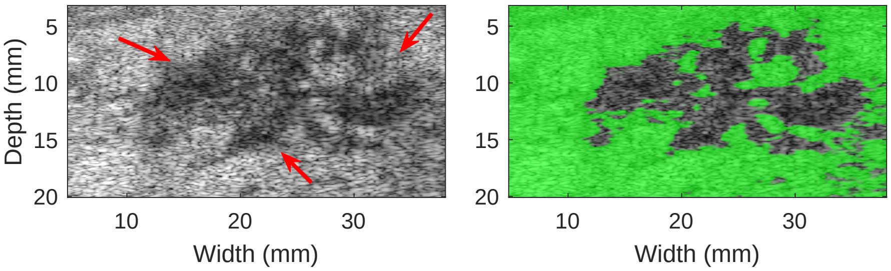


Figure 8.10: Scatterer-density segmentation of *in vivo* breast fibroadenoma (1). B-mode image (left) and the segmentation results of the proposed method overlaid on the B-mode image (right). The green mask denotes predicted FDS regions. The fibroadenoma tumor is specified on the B-mode image.

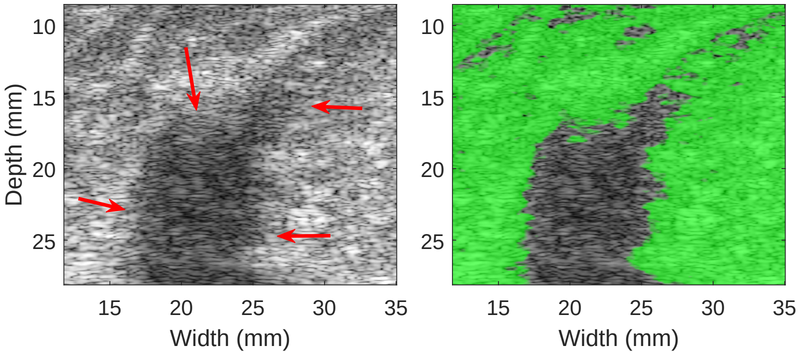


Figure 8.11: Scatterer-density segmentation of *in vivo* breast fibroadenoma (2). B-mode image (left) and the segmentation results of the proposed method overlaid on the B-mode image (right). The green mask denotes predicted FDS regions. The fibroadenoma tumor is specified on the B-mode image.

8.4 Discussion

Segmentation of the density of scatterers is very important for estimation of other QUS parameters since different considerations must be taken for areas with low scatterer number density [145]. In this chapter, the scatterer number density of US images are segmented. The mean scatterer amplitude and scatterer number density both contribute to the intensity of envelope data. Therefore, intensity cannot be used as a reliable feature and both must be taken into account during the training.

In this chapter, we generated a very large dataset having varied mean scatterer amplitude and scatterer number density. To the best of our knowledge, this is the first time such a large dataset is generated which considers that both mean scatterer amplitude and scatterer number density are varied. The generated dataset can be used to train different networks. The imaging settings are also varied and randomly sampled from uniform distributions to increase the generalization of the dataset and improve the robustness of the network when images are obtained from different machine settings. Generating such a large dataset (15000 images) is computationally prohibitive with ultrasound simulation toolboxes such as Field II. Therefore, a grid-based method was employed to generate such a large dataset within manageable times. Field II takes 50 minutes to simulate a phantom (with imaging settings of [145]), whereas our simplified grid-based method only takes 1 second to simulate the same phantom.

The performance of the network without having any information about the test domain was investigated. The Nakagami parametric image was employed as an additional output to reduce the system dependency. The network trained by the generated dataset was able to correctly classify the homogeneous phantoms and segment the phantom with inclusions without any information about the test domain.

The statistical parameters used in quantifying scatterer number density are system dependent. Reference phantoms have been utilized to reduce the effects of machine settings. We investigated how to use a few frames of test domain to update the network. We showed that updating BN layers statistics is enough to adapt the network to the new domain. The statistics of BN layers was updated using a reference set which is a mini-batch composed of data belonging to both classes. Unlike transfer learning, this method only needs a few frames of both classes which facilitates the utilization of reference phantoms for CNNs. The network adopted by the reference set (Baseline+AdaBN) performs better than the network trained on generated dataset especially for the phantom B.

Considering phantom D, We evaluated the reference phantom method using SNR of a reference phantom averaged across 10 frames as well as a recent patch-based CNN. The previously proposed reference phantom method was able to correctly classify most of background regions. However, the inclusions especially on the borders were not classified correctly. Also, the patch-based CNN was able to detect inclusion 3 but fails on the other inclusions. Also, it did not predict the background as FDS with a high confidence (the average of the output of the network for background region was around 0.5). The reason was that the network was trained on fixed imaging settings. The proposed method correctly segmented this phantom without using any patching. Our results confirmed that the proposed segmentation method was more robust than the patch-based CNN since it was trained on a large dataset comprised of diverse imaging settings, scatterer number densities and mean scattering amplitudes.

The recent patch-based CNN [145] is also employed for evaluation. The AUC of this method for homogeneous phantom results (0.78) was lower than the value obtained in [145]. The reason is that in [145], the results are reported for single frame while, in this chapter the average of the results over multiple frames are reported. The drop of performance of patch-based method can also be observed in Fig. 8.6 due to the beam divergence in the bottom of the simulated phantom.

This work demonstrates the potential of using CNNs for QUS. The goal of the chapter is not evaluating different available segmentation networks for scatterer number density segmentation but to provide a general framework to segment scatterer number density using CNNs. Other segmentation networks can also be employed for this task. To the best of our knowledge, this is the first work that segments the scatterer number density of the whole image without patching.

The proposed method has various limitations. First, the simulation approach used to generate the training dataset assumed a homogeneous point spread function within each image. However, different PSF sizes were used in generating different images in the training set. In practice, the PSF changes with depth, resulting in a variation of the effective number of scatterers inside the resolution cell. Despite this limitation, the simplified model shows promising results in Field II simulations and experimental phantoms where the PSF changes because the task here is the segmentation based on FDS and UDS, and not estimating the scatterer number density per resolution cell.

8.5 Summary

In this chapter, the scatterer number density of ultrasound images was segmented using CNN without any patching. To be able to train the network, a large and diverse dataset was required hence different shapes of mean scatterer amplitude and scatterer number density was considered to generate this dataset. We also investigated how to use statistical parametric images such as the Nakagami parametric image to improve the performance in presence of domain shift. In addition to this, inspired by reference methods used in QUS algorithms, we proposed to use a reference set composed of a few frames from both classes. This reference set was utilized to update the statistics of BN layers. We showed that updating the BN layers is adequate and there is no need to update the network weights (fine-tuning) which substantially reduces the amount of data required for domain adaptation. The method was tested on five experimental phantoms and two *in vivo* data imaged by different US scanners. The network was only trained on simulations using the grid approach. Despite this limitation, our method was able to segment the scatterer number density of different simulated data, experimental phantoms and *in vivo* data.

Chapter 9

A deep learning approach for patchless estimation of ultrasound quantitative parametric image with uncertainty measurement

9.1 Introduction

The HK-distribution parametric images are formed by dividing the envelope data into small overlapping patches and estimating parameters within the patches independently. Estimating the parametric images is challenging and the parametric images are often noisy but still have been found useful in clinical applications to detect different abnormalities [111]. The main challenge in estimating the HK-distribution parametric images is the small size of the patches. Increasing the size of the patches might not be helpful since the spatial resolution will be lost and the heterogeneity inside the patch is increased (due to heterogeneity of tissue types or spatially variant nature of point spread function). Correlated samples also should be skipped since they introduce bias to the estimation of HK-distribution parameters. Therefore, a large number of samples inside each patch should be ignored to reduce the correlation between the samples which further reduces the number of samples.

In this chapter, an ultrasound simulation method is employed to generate the training data. Unlike sampling from HK-distribution, this method of simulation data generation contains correlated samples which is a more realistic method and closer to the real ultrasound data [157]. In addition to this, multiple frames from experimental phantom data are employed to obtain a more accurate parametric images and quantify the uncertainty map of the network. The uncertainty map would enable the clinicians to find out how reliable the estimations are in each region.

9.2 Materials and Methods

9.2.1 HK-distribution

The HK-distribution can be formulated as [14]:

$$P_{HK}(A|\varepsilon, \sigma^2, \alpha) = A \int_0^\infty u J_0(u\epsilon) J_0(uA)(1 + \frac{u^2\sigma^2}{2})^{-\alpha} du \tag{9.1}$$

where A denotes the envelope of the backscattered echo ultrasound data, u is the variable of integral that needs to be integrated from zero to infinity, $J_0(.)$ is the zero-order Bessel function, and α is the scatterer clustering parameter which reflects to the scatterer number density. The parameter α and the ratio of coherent signal power (ϵ^2) to the diffuse one ($2\sigma^2\alpha$) denoted as k have been employed as the parameters of HK-distribution to characterize the tissues.

9.2.2 Data Generation

A diverse dataset with known scatterer number density is required to train the network. We followed the fast grid-based method of Chapter 8 [157] with the difference that here the scatterer number density can be any value in the range of 1-20. Assuming weak scattering, the ultrasound RF data can be obtained by the 2D convolution of point spread function (PSF) and the tissue reflectivity function (TRF) [139, 157].

$$s_{(a,l)} = TRF_{(a,l)} * h_{(a,l)} \tag{9.2}$$

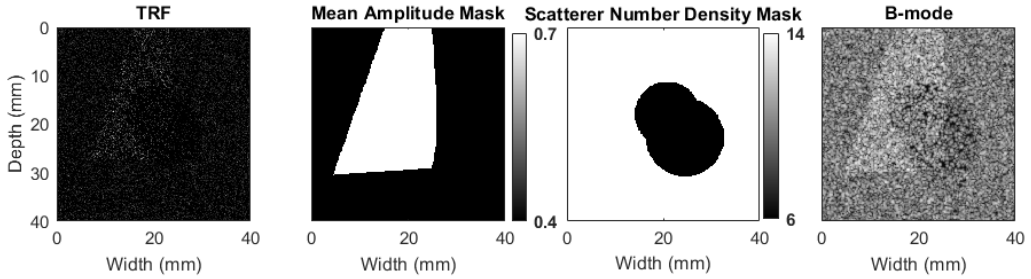


Figure 9.1: One example of generated simulation data.

Where h denotes the PSF. In order to have a known scatterer number density, the PSF is assumed to be spatially invariant which can be described as:

$$h[a,l] = e^{-\frac{1}{2}(\frac{a^2}{{\sigma_a}^2}+\frac{l^2}{{\sigma_l}^2})} \times cos(2\pi f_c a) \tag{9.3}$$

where the PSF is modeled as a 2D Gaussian modulated with a cosine with the center frequency of f_c in the axial direction. The parameters ${\sigma_a}^2$, and ${\sigma_l}^2$ denote the axial and lateral width of the Gaussian function and they are related to the resolution cell. The TRF is the 2D map that contains all point scatterers. In each grid point, only one scatterer is allowed to be present and its amplitude is sampled from normal distribution. The mean of the normal distribution is randomly selected from values 1 to 5, the variance is fixed to 0.02 of the mean value. The TRF is constructed using shapes having different values of scatterer number density and mean scattering amplitudes. One sample generated by this method is shown in Fig. 9.1. We generated 11000 samples to train the network. It should be noted that the ground truth value of scatterer number density is known and depends on how the resolution cell size is defined. Here, we define the resolution cell in each direction as $3 \times \sigma_x$, where x is the axial or lateral directions. Different definition of the resolution cell results in different value of the scatterer number density. Please refer to [157] for more information about the grid-based simulation method. In order to reduce the correlation, samples are skipped from both axial and lateral directions to have an image of size 256×128, and an average correlation between samples of 0.28 is obtained.

9.2.3 Conventional Patch-based Methods

Hruska *et al.* employed SNR, skewness, and kurtosis to estimate HK-distribution parameters [158]. Destrempes *et al.* proposed XU estimator in which two log moment statistics (named as X and U) were employed, and reported improved parameter estimation [14]. The X, and U can be defined as:

$$\begin{aligned} X &=< Ilog(I) > / < I > - < log(I) >, \\ U &=< log(I) > -log(< I >), \end{aligned} \tag{9.4}$$

where $< . >$ denotes sample mean, and $I = A^2$ is the intensity of backscattered signal. The XU estimator iteratively solves a constraint optimization using bisection method to find the parameters. We employed this method for comparison.

9.2.4 Experimental Phantom Data

Several experimental phantoms have been employed to evaluate the method.

1) Layered Phantom: A phantom with a middle layer having different properties than the top and bottom layers reported in previous publication [159] has been employed. The phantom was constructed by an emulsion of ultrafiltered milk and water-based gelatin, and 5–43 μm diameter glass beads were utilized as the source of scattering. A Siemens Acuson S2000 scanner (Siemens Medical Solutions USA, Inc.) with 18L6 probe linear transducer having the center frequency of 8.9 MHz was used for data collection. The layer with higher intensity has a higher backscattering coefficient and scatterer concentration. The backscattering coefficient of this layer is 6.37×10^{-3} $cm^{-1}sr^{-1}$ and it is 3.52×10^{-3} cm^{-1} sr^{-1} for other parts at the center frequency.

2) CIRS Phantom: Data from a multipurpose CIRS phantom (model 040GSE, Norfolk, VA, USA) was collected at Concordia university using E-CUBE 12 Alpinion machine by an L3-12H transducer with a sampling rate of 40 MHz and center frequency of 8.5 MHz.

3) Gammex Phantom: Data was collected at university of Wisconsin using a Verasonics Vantage 128 System (Verasonics,Kirkland, WA, USA) machine with an L11-5v transducer operating at 8 MHz.

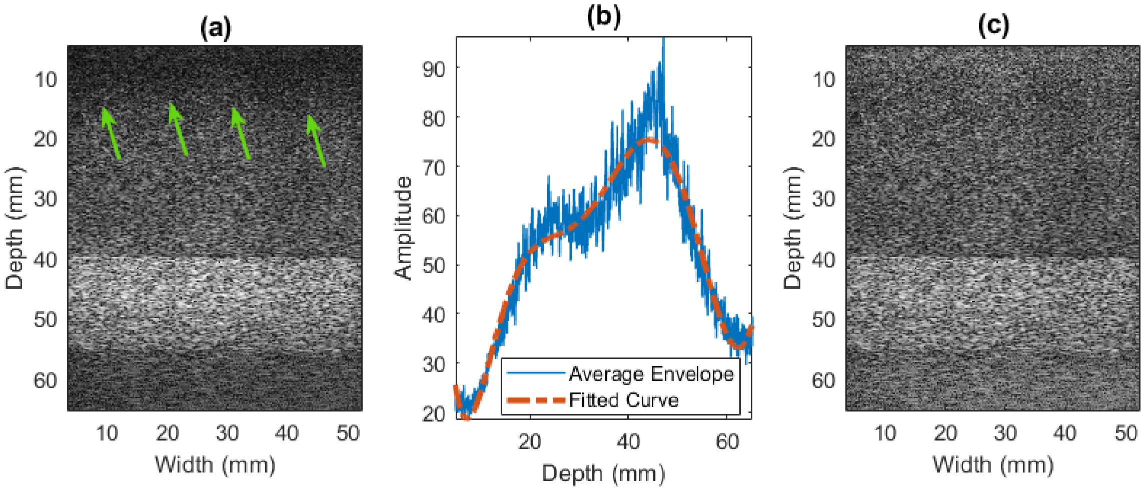


Figure 9.2: Layered phantom before the system effect compensation (a). The envelope amplitude average of reference phantoms and the fitted curve (b). The compensated layered phantom (c). The intensity variation of the phantom before compensation is clear and the area with lower intensity is marked. The phantom data has been reported in previous publication [159].

9.2.5 Pre-Processing of experimental phantom data

Experimental phantom data may contain variations of intensity due to beamforming, focusing and time gain control (TGC) which are not present in the simulation data and leads to performance decay of parameter estimation. Inspired by the reference phantom method which cancels out the system effect [109], we employ an intensity normalization technique to reduce the system effect on the envelope data intensity. First, the average of envelope amplitudes in each depth is computed across several frames of uniform reference phantoms imaged by the same imaging settings. In the next step, a curve is fitted to the obtained average envelope amplitude. The inverse of the fitted curve can be utilized to normalize the experimental phantom data imaged by the same settings. The procedure is depicted in the Fig. 9.2. It can be observed that the normalized data has a more uniform intensity across different depth compared to the pre-normalized one.

9.2.6 Multi-frame acquisition and data processing for experimental phantoms

For the layered phantom, the probe is moved in the out-of-plane direction to collect 12 frames from this phantom. We skipped 7 samples in the axial and 1 sample in the lateral direction from the acquired envelope data to reduce the correlation from 0.75 (no skipping) to 0.141. The patch size for XU algorithm is 4.5×4.5 *mm* with 75% overlap.

For the CIRS phantom, the probe is moved in the out-of-plane direction to collect 10 frames from this phantom. The patch size for XU algorithm is 4.5×4.5 *mm* with 75% overlap. We skipped 4 samples in the axial and 0 sample in the lateral direction from the acquired envelope data to reduce the correlation from 0.95 (no skipping) to 0.49. In another experiment, we skipped 5 samples in the axial and 1 sample in the lateral direction to have the correlation of 0.36 to see the effect of different correlations. It should be noted that skipping more samples results in reduction of the bias but lower number of samples would be available to estimate the scatterer number density which makes estimation of small parts challenging.

For the Gammex phantom, the probe is moved in the out-of-plane direction to collect 12 frames from this phantom. The patch size for XU algorithm is 4.0×4.0 *mm* with 75% overlap. We skipped 6 samples in the axial and 1 sample in the lateral direction from the acquired envelope data to reduce the correlation from 0.95 (no skipping) to 0.55. In another experiment, we skipped 13 samples in the axial and 1 sample in the lateral direction to have the correlation of 0.19 to see the effect of different correlations.

9.2.7 Prediction and uncertainty quantification

For experimental phantom data, several frames are collected by sweeping the probe in the out-of-plane direction. The final estimate and its corresponding uncertainty can be quantified as:

$$\begin{aligned} \widetilde{S} &= \frac{1}{N_f}\sum_{i=1}^{N_f} \widetilde{S}_i, \\ Uncertainty &= \frac{\sqrt{\frac{1}{N_f}\sum_{i=1}^{N_f}(\widetilde{S}_i - \widetilde{S})^2}}{\widetilde{S}}, \end{aligned} \tag{9.5}$$

where the final estimate ($\widetilde{S}$) is calculated by averaging the estimates ($\widetilde{S}_i$) obtained from N_f frames. The proposed uncertainty measures how different the estimates are across different frames. If the estimates from all frames are similar, the uncertainty will be low. On the contrary, high uncertainty is expected when the estimates are different. It should be noted that this is a frame-wise uncertainty measurement meaning that if there is no difference in the estimated value across different frames, the uncertainty is low even if the estimated values are incorrect.

9.2.8 Network architecture and training

We employed DeeplabV3 which has shown excellent performance in semantic segmentation [160]. The last layer of the model is altered to have one output as the scatterer number density. The following loss function is employed:

$$loss = \frac{1}{N} \sum (log_{10}(\widetilde{S}) - log_{10}(S))^2 \tag{9.6}$$

where $\widetilde{S}$ and S are the predicted and ground truth scatterer number density, respectively. The log compression is employed to avoid the bias toward high scatterer number density values. Rectified adam (RAdam) is employed as the optimizer which has shown outperforming Adam and be robust to the selection of learning rate [161]. The network is trained on single NVIDIA RTX 3090 with 24 GB of memory.

9.3 Results

9.3.1 Simulation Results

The method is evaluated on 1000 test data generated in a similar way of training data. Due to the fact that the value of the parameter α is not known for the simulation data, we cannot compare it with the XU estimator. The results are given in Table 9.1. The metrics are evaluated on each sample of test dataset and, the average and standard deviation are reported. Four samples of the simulation results are depicted in Fig. 9.3.

Table 9.1: The simulation results of the proposed method.

	RMSE	RRMSE	MAE
Patchless CNN	1.12±0.59	0.42±0.19	0.85±0.55

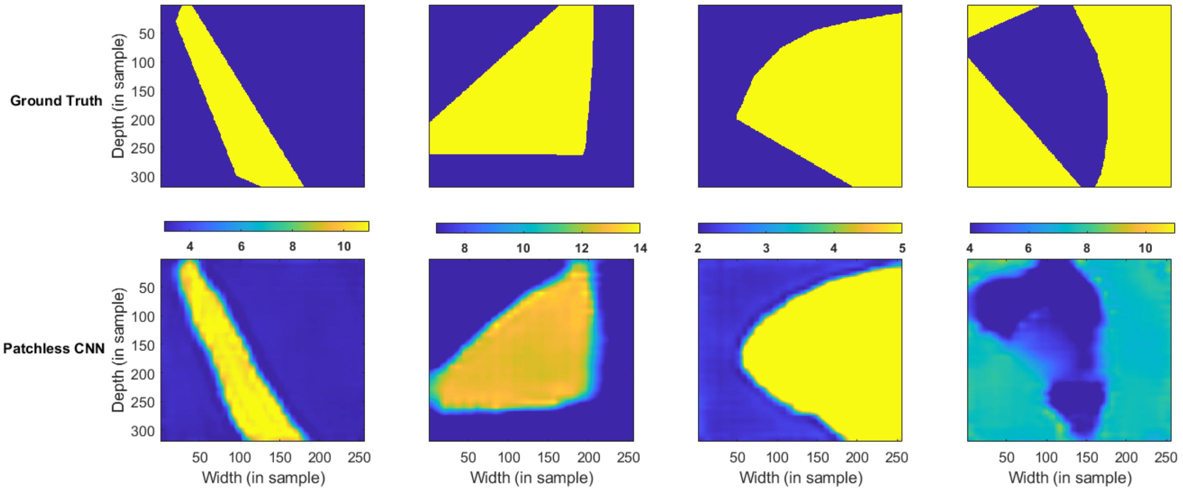


Figure 9.3: Four examples of the patchless CNN estimation of scatterer number density. The parametric image values are underestimated in the forth sample.

9.3.2 Experimental phantom results

Layered phantom results

The predicted results of the CNN and XU are the average across the collected frames. The uncertainty is measured by dividing the standard deviation of the results by their mean value. The results of the patchless CNN and XU algorithm are given in Fig. 9.4. The uncertainty map has a higher value in the region where the CNN incorrectly estimates a lower value in the phantom (the region is marked with red arrows).

In order to quantitatively investigate the presented results, the ratio of the mean of obtained values in regions R2 and R1 (highlighted in the B-mode image) are computed and compared with the ratio of their known backscattering coefficient. The ratio is $\frac{6.37}{3.52} = 1.81$ for the backscattering coefficient, it is $\frac{16.34 \pm 3.72}{10.99 \pm 2.73} = 1.61 \pm 0.63$ for XU, and $\frac{5.19 \pm 0.31}{1.81 \pm 0.03} = 1.81 \pm 0.11$ for the patchless CNN. It can be seen that the ratio for the patchless CNN is closer to the true backscattering coefficient ratio compared to the XU method which demonstrates that this method has a lower bias compared to XU.

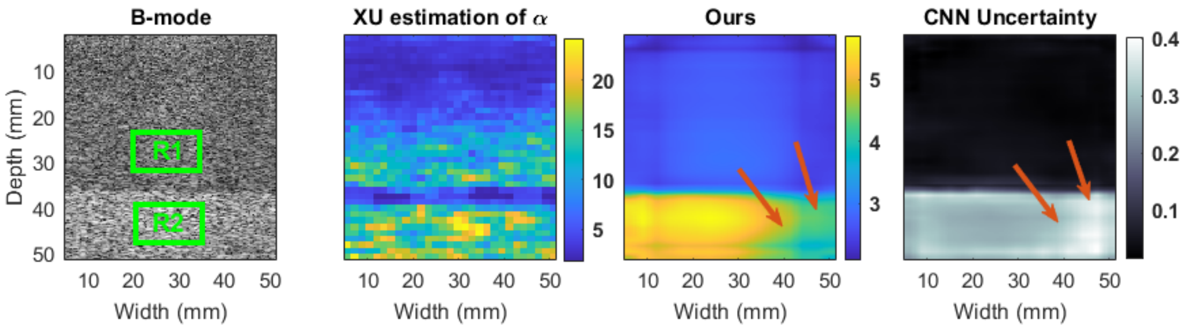


Figure 9.4: From left to right: B-mode image, estimated α by XU algorithm, patchless CNN, and uncertainty. The area that patchless CNN incorrectly underestimated the scatterer number density is marked and uncertainty is higher in that area.

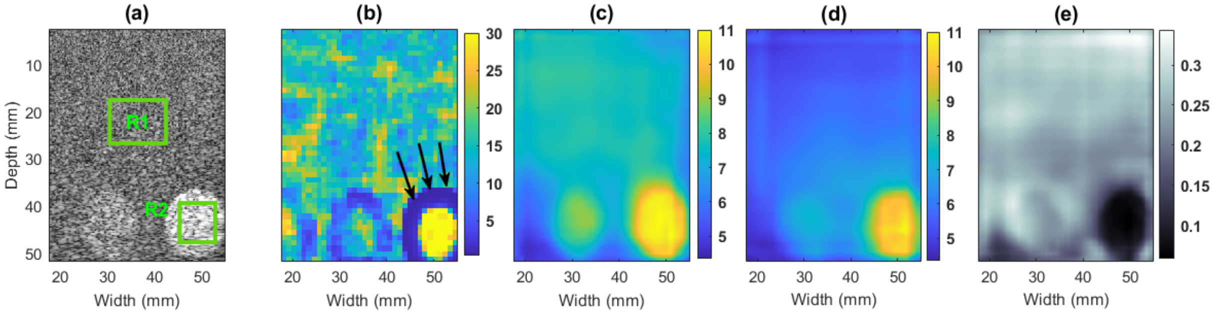


Figure 9.5: Results of CIRS phantom. B-mode image (1), estimated α by XU algorithm (b), patchless CNN using samples with correlation 0.49 (skip 4 samples in axial direction) (c), patchless CNN using samples with correlation 0.36 (skip 5 samples in axial and 1 in the lateral direction) (d), and uncertainty obtained from c (e).

CIRS phantom results

The results are shown in Fig. 9.5. The XU algorithm fails to obtain reliable values on the borders (marked in b) of the inclusions due to presence of two different distributions inside the patches of those regions, while the CNN results do not suffer from this issue. The CNN provides higher contrast and less bias with the samples with lower correlation, but due to skipping more samples, the inclusion borders are not as clear as with the higher correlation one. This demonstrates a trade off between skipping the samples to reduce the correlation (lower bias) and spatial resolution. This issue was less identifiable in the layered phantom since there were adequate samples in each layer; therefore, we were able to skip enough samples to reduce the bias.

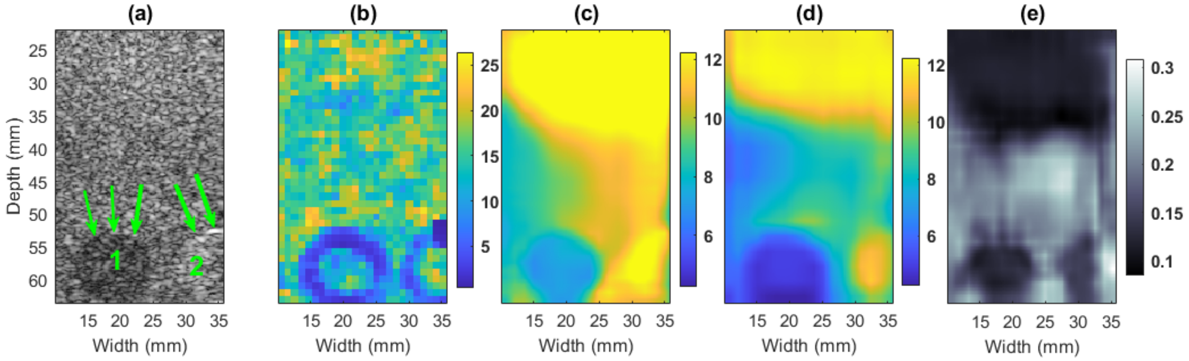


Figure 9.6: Results of Gammex phantom. B-mode image (1), estimated α by XU algorithm (b), patchless CNN using samples with correlation 0.55 (skip 6 samples in axial and 1 in lateral direction) (c), patchless CNN using samples with correlation 0.19 (skip 13 samples in axial and 1 in the lateral direction) (d), and uncertainty obtained from c (e).

Gammex phantom results

The results are shown in Fig. 9.6. The parametric image obtained by XU algorithm has artifacts around the inclusions similar to Fig. 9.5. The parametric image with higher correlation (c) has higher spatial resolution but it has more biased values (compare the inclusion 1 in (c) and (d)). The inclusion 1 is clearly detected by the CNN for both inputs (c and d), but in (c) the inclusion has a more biased values than (d) due to presence of higher correlation. The inclusion 2 is also hardly detected by the CNN since only part of it is available.

9.4 Discussion

In this chapter, a deep learning frame work is proposed to quantify the scatterer number density. One important thing to note is that the method of calculating the scatterer number density in simulation data generation dictates the output of the network. In other word, different definition of resolution cell in simulation data generation can change the ground truth value which also alters the output of the network.

Another crucial point is that the correlation between samples can affect the results. Most of recent work focus on estimating the scatterer number density using training data obtained from i.i.d samples. Although these samples contain more information than the correlated ones, achieving this condition in real data is difficult and requires skipping many samples which is not efficient in parametric image reconstruction. In this work, the approach is changed by generating simulation data using the grid-based method to have correlation between the samples. The disadvantage of this approach is that the exact value of α is not known, and generating coherent component is complex.

The impact of different correlations for the experimental phantoms is also investigated. Generally, lower correlation can help having less bias in estimation but achieving this might cause losing spatial resolution due to requiring skipping many samples.

9.4.1 Summary

In this chapter, a patchless deep learning solution was proposed to obtain scatterer number density parametric images. The method was validated using simulation data and three experimental phantoms. Frame-wise uncertainty map was also obtained from the parametric images of the multiple frames. In order to enable the network to predict the experimental phantom data having correlated samples, a grid-based simulation in which correlated samples were present was employed for training data generation. The proposed method was trained on simulation data and was able to reconstruct scatterer number density parametric image of experimental phantoms imaged by different scanners.

Chapter 10

Homodyned K-distribution: parameter estimation and uncertainty quantification using Bayesian neural networks

In our previous works, we employed CNN to classify and segment the US data into fully developed (high scatterer number density) and underdeveloped (low scatterer number density) speckle [145, 157]. Recently, an artificial neural network (ANN) was introduced by Zhou *et al.* [162]. The proposed method was a MLP that employed speckle statistics to estimate the parameters of HK-distribution.

The ANN estimator employs MLP layers which are prone to overfitting. In addition to this, there is no metric to investigate the reliability of the estimated value. In this chapter, we address these two issues and aim to improve the estimation of the HK-distribution parameters and quantify the uncertainty using bayesian neural networks (BNN). The proposed method can also be used to extract QUS parametric images, and detect the regions with high uncertainty.

10.1 Material and Method

10.1.1 Homodyned K-distribution parameters

The Homodyned K-distribution (HK-distribution) is defined as [14]:

$$P_{HK}(A|\varepsilon, \sigma^2, \alpha) = A\int_0^\infty uJ_0(u\epsilon)J_0(uA)(1+\frac{u^2\sigma^2}{2})^{-\alpha}du \tag{10.1}$$

where α is the scatterer clustering parameter that depends on the scatterer number density, A is the envelope of the backscattered echo signal, and $J_0(.)$ denotes the zero-order Bessel function. The coherent signal power is ϵ^2, and the diffuse signal power can be obtained by $2\sigma^2\alpha$ [14]. The parameter k is defined as the ratio of coherent to diffuse signal power $(\frac{\epsilon^2}{2\sigma^2\alpha})$ and along with α has been employed widely for tissue characterization and we refer to them as HK-distribution parameters. The main purpose of this chapter is to estimate k and $log_{10}(\alpha)$ (similar to [162]) and quantify the uncertainty of their estimation.

In order to generate training data for both ANN and BNN, sampling from HK-distribution is required. Similar to [158, 162], we employed the following equation produce synthetic samples from HK-distribution.

$$a_i = \sqrt{\left(\varepsilon + X\sigma\sqrt{Z/\alpha}\right)^2 + \left(Y\sigma\sqrt{Z/\alpha}\right)^2} \tag{10.2}$$

where X and Y are independent and identically distributed (i.i.d) samples from unit Normal distribution, a_i is the generated sample from HK-distribution, and Z is sampled from the Gamma distribution with shape parameter α and scale parameter of 1. To generate training data, $log_{10}(\alpha)$ is randomly selected from values ranging -0.3 to 1.4 which corresponds to α of 0.5 to 25. k is also randomly selected from values ranging 0 to 1.

Different sizes of data, results in different values for the calculated feature. We generated different sizes of data (we refer to it as N_s) to train the networks (similar to [162]). The network is trained for each size separately using 10000 generated training data. The test data is generated with the same range of parameters, on total 31 and 11 distinct $log_{10}(\alpha)$ and k values, respectively. For each value of $log_{10}(\alpha)$ and k, 100 test sets are generated;

therefore there are $31 \times 11 \times 100$ samples of test data for each N_s.

10.1.2 ANN estimator

In [162], Zhou *et al.* proposed an ANN approach to estimated HK-distribution parameters and out-performed the XU optimization method [14]. The procedure was as follows. First, SNR, skewness, Kurtosis, X and U statistics were computed. The equations to compute the parameters are given as:

$$\begin{aligned} R_v &= \frac{\overline{A^v}}{\sqrt{\overline{A^{2v}} - (\overline{A^v})^2}}, S_v = \frac{\overline{(A^v - \overline{A^v})^3}}{(\overline{A^{2v}} - (\overline{A^v})^2)^{1.5}}, \\ K_v &= \frac{\overline{A^{4v}} - 4\overline{A^v} \times \overline{A^{3v}} + 6\overline{A^{2v}} \times \overline{A^v}^2 - 3\overline{A^v}^4}{(\overline{A^{2v}} - \overline{A^v}^2)^2}, \\ U &= \overline{log(I)} - log(\overline{I}), X = \overline{I \times log(I)}/\overline{I} - \overline{log(I)}, \end{aligned} \tag{10.3}$$

where A is the envelope data, I is the intensity ($I = A^2$), and v is $\{0.72, 0.88\}$ as suggested by Hruska *et al.* [158] and Gao *et al.* [163].

In the next step, $R_{0.88}$, $R_{0.72}$, $S_{0.88}$, $S_{0.72}$, $K_{0.88}$, $K_{0.72}$, X, and U were employed as inputs of a MLP (ANN) as suggested by [163] to train the network which estimated the $log_{10}(\alpha)$ and k. We implemented this method for comparison and used the same network architecture (2 hidden layers with 10 and 4 nodes). We refer to this method as ANN.

10.1.3 Bayesian neural network (BNN)

Let Y, W, and X be the target, weights and input vectors, respectively. Assuming the training data be $D = \{X_i, Y_i\}$, training a NN can be defined as:

$$W^* = \arg\max_{W} \{P(D|W)\} \tag{10.4}$$

where the optimum weights (W^*) are learned during the training and used in the test to predict Y. In BNN, the weights of the neural network are not fixed, and each weight is sampled from a distribution. During the training instead of learning the weights, the parameters of the distribution, from which the weights are sampled, are learned. Predicting Y can be formulated as [164]:

$$P(Y|D) = \int_W P(Y|W)P(W|D) \tag{10.5}$$

where $p(W|D)$ is the posterior distribution of the weights which is learned during the training. The integration over all possible values of W is intractable and computationally expensive. To resolve this issue, the posterior distribution $P(W|D)$ is sampled and the prediction $\tilde{Y}$ can be obtained by:

$$\begin{gathered} \tilde{Y} \simeq \frac{1}{N}\sum_{i=1}^{N}\tilde{Y}_i, \\ W_i \sim P(W|D), \tilde{Y}_i \sim P(Y|W_i), \end{gathered} \tag{10.6}$$

where the operator $\sim$ denotes sampling from the distribution. Eq 10.6 can be simply explained as running the trained network multiple times (each forward pass of the network gives $\tilde{Y}_i$) and computing the mean value of predictions as the final estimated value. Uncertainty can also be quantified as the standard deviation of the predictions $\tilde{Y}_i$ which can be written as:

$$uncertainty = \sqrt{Var(Y_i)}, \tag{10.7}$$

The Mean absolute error (MAE) loss is utilized for training which is also sampled multiple times by forwarding the inputs and sampling from the weights multiple times (here 6) to have a better approximation of the loss value. Two Bayesian hidden layers having 64 and 200 nodes with leaky Relu activation functions were employed and Adam optimizer is utilized for optimization.

10.2 Results

10.2.1 Simulation Results

The accuracy of the estimators of HK-distribution parameters heavily depends on the number of available i.i.d samples. We evaluated the methods using different number of samples (N_s). The relative root mean square error (RRMSE) and MAE are employed as the metrics which

Table 10.1: RRMSE and MAE of $log_{10}(\alpha)$ using different numbers of HK-distribution samples (N_s).

	ANN		BNN	
	RRMSE	**MAE**	**RRMSE**	**MAE**
$N_s = 65536$	0.054	0.048	**0.012**	**0.035**
$N_s = 16384$	0.052	0.061	**0.029**	**0.054**
$N_s = 4096$	0.125	0.091	**0.090**	**0.083**
$N_s = 1024$	0.393	0.129	**0.388**	**0.123**

Table 10.2: RRMSE and MAE of k using different numbers of HK-distribution samples (N_s).

	ANN		BNN	
	RRMSE	**MAE**	**RRMSE**	**MAE**
$N_s = 65536$	0.143	0.074	**0.122**	**0.053**
$N_s = 16384$	**0.218**	0.084	0.235	**0.073**
$N_s = 4096$	0.359	0.118	**0.291**	**0.103**
$N_s = 1024$	0.538	0.153	**0.460**	**0.139**

can be defined as [158, 162]:

$$\begin{aligned} RRMSE &= \sqrt{\frac{<(y-\widetilde{y})^2>}{|y|+\varepsilon}}, \\ MAE &=< |y-\widetilde{y}| >, \end{aligned} \tag{10.8}$$

where $< . >$ denote averaging operation and ε is a small number (here 0.001) to avoid division by zero. The simulation results for $log_{10}(\alpha)$ and k are given in Tables 10.1 and 10.2, respectively. According to the tables, the proposed BNN has lower error compared to ANN for estimation of both $log_{10}(\alpha)$ and k in the most of sample sizes.

The RRMSE and MAE error maps are shown for $N_s = 16384$ and different ground truth values of $log_{10}(\alpha)$ and k . RRMSEs high values around the ground truth zero are due to the division by the small number. For better visualization, RRMSEs are plotted in log scale. Fig. 10.1 shows that the proposed BNN method has lower error than ANN (notice the blue regions in RRMSEs).

The proposed method can also provide uncertainty of the prediction (Eq 10.7). Fig. 10.2 shows the uncertainty of the estimation of the parameters. It can be seen that areas in Fig.

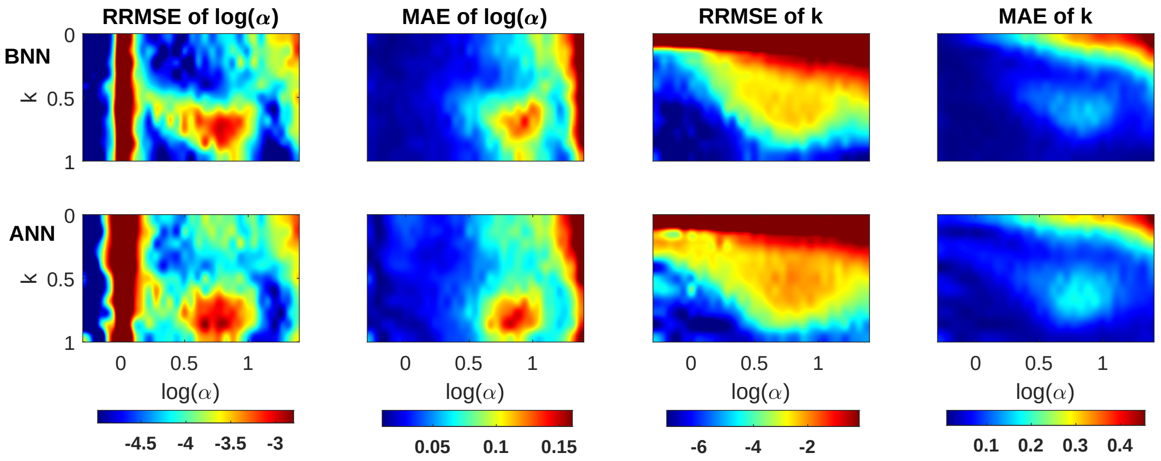


Figure 10.1: The RRMSE and MAE error maps of BNN (top) and ANN (bottom) for $N_s = 16384$. The RRMSEs are shown in log scale for better visualization.

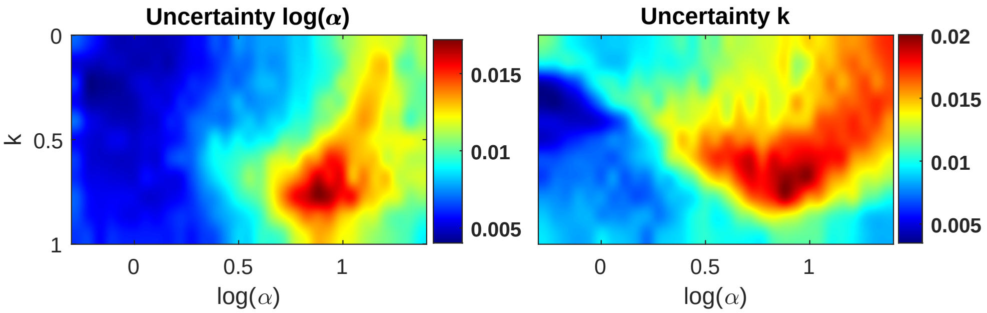


Figure 10.2: The estimated uncertainty (standard deviation of predictions) of $log_{10}(\alpha)$ and k for $N_s = 16384$ using BNN. The areas with high uncertainty correspond to areas with high error in Fig. 10.1.

10.1 that high error is presents, the uncertainty is high which can provide an insight about the reliability of the estimation.

10.2.2 Experimental Phantom Results

The two layered phantom used in the previous chapter was employed for the evaluation. The middle layer was made to have a higher backscattering coefficient than the other two layers by increasing the concentration of scatterers (higher α). The backscattering coefficient of top and bottom layers is 3.52×10^{-3} cm^{-1} sr^{-1} and it is 6.37×10^{-3} $cm^{-1}sr^{-1}$ for the middle layer at the center frequency. Data from this phantom has been reported in the previous publication [159].

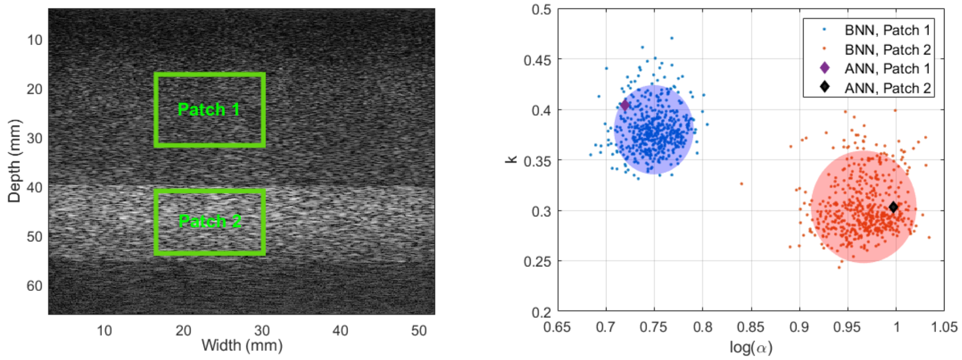


Figure 10.3: B-mode image of the layered phantom (left) and predictions for the patches specified in the b-mode image (right) for BNN and ANN trained on $N_s = 16384$. The shaded areas show two times of the standard deviation of the predictions.

The B-mode image of the phantom is shown in Fig. 10.3 (top). Two large patches of size 14.40×13.6 mm (patch 1) and 12.68×13.6 mm are extracted from low and high scatterer concentration layers, respectively. In order to avoid introducing bias, neighbor samples (14 samples in axial and 3 in lateral) are skipped to reduce the correlation between samples before computing the features. The obtained features are averaged over 12 frames and then given to the networks. The features were passed to the BNN multiple times to acquire different samples of the predicted distribution. The results are shown in Fig. 10.3.

The patch 2 has higher α than the patch 1 which is expected since patch 2 has A higher scatterer concentration. Although the phantom has very low coherent components, the predicted k parameter is discernible. One possible explanation could be the false coherency due to low number of samples [109]. Comparing the two methods, ANN only provides a single estimate of the parameters while, BNN offers the distribution of the parameters which can be sampled multiple times. By looking closely at the BNN results, it can be observed that the network has a higher uncertainty for patch 2. This has physical interpretation that by increasing the scatterer number density, the estimation would be more difficult and a higher uncertainty is obtained.

The exact value of α is not known for the phantom but the ratio of high to low scatterer density is close to the ratio of their corresponding backscattering coefficients which is known. The mean value ± standard deviation of the BNN prediction of $log_{10}(\alpha)$ for the patch 1 is 0.749 ± 0.0206, and it is 0.967 ± 0.0273 for the patch 2. The ratio of backscattering coefficients of patch 2 to patch 1 is $\frac{6.37}{3.52} = 1.81$. The ratio of the predicted α values is $\frac{10^{0.967 \pm 0.0273}}{10^{0.749 \pm 0.0206}} = 1.65 \pm 0.128$. It can be observed that the ratio of the estimated α values is very close to the ground truth ratio of backscattering coefficients.

10.3 Summary

In this chapter, a Bayesian Neural Network (BNN) is proposed to estimate HK-distribution parameters. The method provides the distribution of estimated parameters which can be sampled multiple times to acquire the mean prediction and uncertainty. It is compared with a recent neural network approach using simulation and experimental phantom data.

Chapter 11

Deep Autoencoder Feature Projection for Accurate Homodyned K-distribution Parameter Estimation in Quantitative Ultrasound

11.1 Introduction

HK-distribution parameters are physically meaningful, the scatterer clustering parameter (α) and the ratio of the coherent to diffuse scattering power (k) are two parameters of this distribution that are related to scatterer number density, and microstructrual organization of scatterers, respectively. They have been used extensively for tissue characterization including non-alcholic fatty liver disease (NAFLD) staging [165]. Envelope statistics such as point-wise signal-to-noise ratio (SNR), skewness, kurtosis, and the log-based moments are usually employed to estimate HK parameters. Hruska *et al.* employed SNR, skewness, and kurtosis to estimate the HK parameters by minimizing the difference between their sample estimates and the theoretical values [158]. Destrempes *et al.* proposed to employ two log-based moments X and U and used bi-section interpolation to estimate the parameters [14]. They reported improved estimation by using these two moments (we refer to this method as XU method). Liu *et al.* proposed to utilize several statistics (on total 16) and compared them with the theoretical values using table search [166]. They also performed an extensive analysis about the feature selection for each HK parameter estimation.

Recently, deep learning methods are being used more frequently for HK parameter estimation. Zhou *et al.* used a MLP that takes envelope statistics as input, and outputs the HK parameters [162]. We proposed a Bayesian neural network (BNN) to do the task which also was able to quantify the uncertainty of the estimation [167], where we reported lower error compared to the method of Zhou *et al.* [162].

In this chapter, we consider the feature space as a high-dimensional space where each patch is represented by a point. We show that the envelope statistics features lie in a hyperplane that covers only small volume of feature space, and hypothesize that points deviate from this hyperplane because of statistical errors. We then proposed to project the points into this hyperplane using an autoencoder. This novel method is validated using simulations, experimental phantoms, and *in vivo* data.

11.2 Material and Method

11.2.1 Homodyned K-distribution Parameters Estimation Problem Formulation

Homodyned K-distribution

The Homodyned K-distribution can be defined as [14]:

$$P_{HK}(A|\varepsilon, \sigma^2, \alpha) = A \int_0^\infty uJ_0(u\epsilon)J_0(uA)(1 + \frac{u^2\sigma^2}{2})^{-\alpha} du \tag{11.1}$$

where A is the envelope of the backscattered echo signal, α is the scatterer clustering parameter that is related to the scatterer number density, and $J_0(.)$ denotes the zero-order Bessel function. The coherent signal power can is represented by ϵ^2, and the diffuse signal power is $2\sigma^2\alpha$ [14]. The scatterer clustering parameter (α), and coherent to diffuse scattering ratio

$k = \frac{\epsilon}{\sigma\sqrt{\alpha}}$ are the HK-distribution parameters that are required to be estimated. The envelope statistics (SNR, skewness, kurtosis of the fractional amplitude, and log-based moments) can be obtained by:

$$R_v = \frac{E[A^v]}{\sqrt{E[A^{2v}] - (E[A^v])^2}},$$

$$S_v = \frac{E[(A^v - E[A^v])^3]}{(E[A^{2v}] - (E[A^v])^2)^{1.5}},$$

$$K_v = \frac{E[A^{4v}] - 4E[A^v] \times E[A^{3v}] + 6E[A^{2v}] \times E[A^v]^2 - 3E[A^v]^4}{(E[A^{2v}] - E[A^v]^2)^2}, \tag{11.2}$$

$$U = E[log(I)] - log(E[I]),$$

$$X = E[I \times log(I)]/E[I] - E[log(I)],$$

where $I = A^2$ and v is the power where we selected the values $0.72, 0.88$ suggested by [158].

HK parameters inverse problem

Let the envelope statistic features of single patch be:

$$F \in \mathbb{R}^{M\times 1} = [R_{0.72}, R_{0.88}, S_{0.72}, S_{0.88}, K_{0.72}, K_{0.88}, X, U]^T \tag{11.3}$$

and the HK parameters be denoted as $\theta \in \mathbb{R}^{2\times 1} = [\log_{10}(\alpha), k]^T$. The forward and inverse problems can be illustrated as:

$$\begin{aligned} \theta \in \mathbb{R}^{2\times 1} &\xrightarrow{forward} F \in \mathbb{R}^{M\times 1}, \\ \widetilde{\theta} \in \mathbb{R}^{2\times 1} &\xleftarrow{inverse} \widetilde{F} \in \mathbb{R}^{M\times 1}, \end{aligned} \tag{11.4}$$

where $\widetilde{F}$ denotes the sample estimate of the envelope statistics, and $\widetilde{\theta}$ represents the estimated θ. M is the number of envelope statistic features which we employed eight features similar to [162]. The forward problem can be viewed as obtaining the envelope statistics from the known parameters of the distribution. Hruska *et al.* showed that moments of HK-distribution can be analytically obtained from a known HK-distribution by [158]:

$$\begin{aligned} E[A^v] = \int_0^\infty (\frac{2\sigma^2}{\alpha})^{v/2} \frac{\Gamma(1+v/2)x^{v/2+\alpha-1}}{\Gamma(\alpha)e^x} \\ 1\mathrm{F}_1(-v/2; 1; \frac{-\alpha\varepsilon^2}{2\sigma^2 x})dx, \end{aligned} \tag{11.5}$$

where $1\mathrm{F}_1(a, b, x)$ is confluent hypergeometric function of the first kind. The feature vector then can be obtained by inserting $E[A^v]$ obtained from Eq 11.5 into Eq 11.2. The sample mean is used to approximate $E[A^v]$, and the calculated sample envelope statistic features $(\widetilde{F})$ is employed to estimate HK parameters $(\widetilde{\theta})$.

Estimation of HK parameters from envelope statistics can be viewed as an inverse problem which maps F from feature space with M dimension to the 2-dimensional HK parameter space. The low dimensional space of HK parameters enforces the feasible feature values to lie in a low dimensional manifold. Inspired by this, we designed a model projection neural network based on denoising autoencoder to project the noisy features (F) into the feasible hyperplane.

11.2.2 Model Projection Autoencoder (MPAE)

Autoencoders are neural networks that receive the input and map them to a lower-dimensional representation by transforming into more informative lower dimensional features. In the encoder part, the input dimension is reduced using several hidden layers, and in the decoder part, the input is reconstructed using the lower-dimensional representation. These networks have been found useful in many applications, such as denoising and dimensionality reduction, where it was shown that they act like a non-linear principal component analysis (PCA) [168]. Denoising autoencoders employ the corrupted data and try to reconstruct the clean data. Vincent *et al.* showed that if data lie in a low-dimensional manifold, the corrupted data will be further away from this manifold, and denoising autoencoders can project the corrupted data into the low-dimensional manifold [169]. This idea is illustrated in Fig. 11.1.

Our proposed method based on autoencoders is illustrated in Fig. 11.2. We consider the sample estimates of the features as the corrupted input data and the theoretical value from the forward problem as the clean output features. A model projection autoencoder (MPAE) is utilized to obtain clean features (Projection step). An estimator is employed to estimate the HK parameters from the reconstructed features.

Network architecture and training schedule

The network comprises seven layers; the encoder part has four layers with $64, 32, 32, b$ nodes, and the decoder has three layers with $32, 32, 8$ nodes. The parameter b is the number of nodes in the bottleneck, which determines the size of a low-dimensional manifold. We investigate different sizes of bottlenecks in the result section. The activation functions of all layers were leaky ReLu except the last layer, which does not have any activation function. We test placing dropouts in different layers and obtain the best performance on placing a dropout on the second encoder layer with the probability of 0.2. The features are normalized to have zero mean and standard deviation of 1. We employed the combination of smooth L1 and MSE as the loss function. For each sample size of $N_s = 65536$, $N_s = 4096$, and $N_s = 1024$, a separate MPAE is trained since the noise distribution is different for each sample size.

11.2.3 HK Parameters Estimators

We employ BNN as the estimator after MPAE feature reconstruction. The method is also compared with XU estimator.

BNN estimator

In [167], we developed a BNN to estimate the HK parameters. The BNN estimator outperformed the ANN counterpart [162] and was able to quantify the uncertainty. In BNN, the network weights are sampled from a distribution learned in the training phase. Each time the network runs, the weights would be different; therefore, the network is run for each feature vector multiple times during the inference, and the average value and standard deviations are considered as the prediction and uncertainty, respectively.

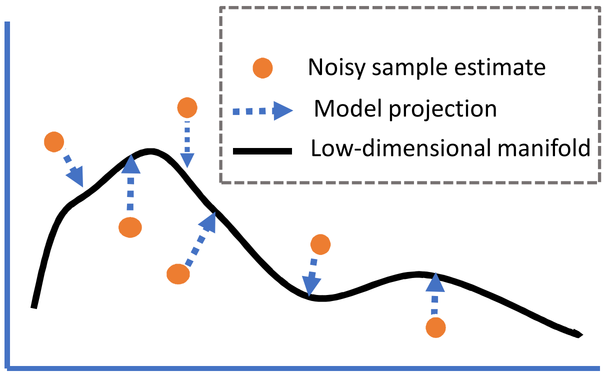


Figure 11.1: The theoretical values of envelope statistics lie in a low-dimensional manifold. Sample estimates are corrupted by noise and lie further away from the manifold. The MPAE projects the noisy sample estimates into the low-dimensional manifold.

The theoretical values of the statistical features were employed for training the BNN, which allowed us to use one network for all sample sizes, similar to the generalized neural network in [170].

XU estimator

Destrempes *et al.* proposed to employ two log-based moments, X and U as the statistical features [14]. They used bi-section interpolation to find the intersection between the theoretical values and the sample estimates of X and U. They reported improved estimation using these two moments compared to R, S, and K. We compared our proposed method with this algorithm.

11.2.4 Datasets and data generation

Simulation data

Training data is generated by sampling from HK-distribution. The equation suggested by [158, 162] is usually employed to generate the simulation data:

$$a_i = \sqrt{\left(\sqrt{2k} + X\sigma\sqrt{Z/\alpha}\right)^2 + \left(Y\sigma\sqrt{Z/\alpha}\right)^2} \tag{11.6}$$

where α and k are the scatterer clustering parameter and coherent to diffuse scattering ratio. a_i is the generated sample, X and Y are independent and identically distributed

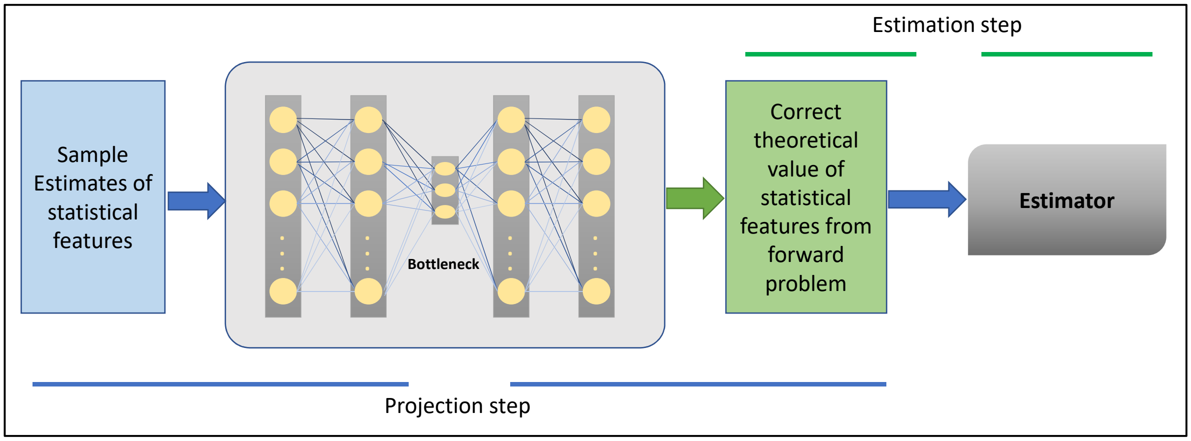

Estimation step
Sample Estimates of statistical features
Bottleneck
Correct theoretical value of statistical features from forward problem
Estimator
Projection step

Figure 11.2: The proposed framework for estimation of HK parameters. The sample estimate of envelope statistical features (F) is projected into a low-dimensional space (bottleneck) and the clean feature is reconstructed (projection step). The reconstructed features are given to the estimators of HK parameters (estimation step).

(i.i.d) samples from Normal distribution. Z is sampled from the Gamma distribution with scale parameter of 1 and shape parameter of α. This equation generates i.i.d samples from HK-distribution, which differs from real envelope samples from experimental echo signals due to the correlation among samples. Generating correlated samples from HK-distribution directly is not straightforward; therefore, in order to generate more realistic samples, we propose to generate correlated samples by employing correlated normal distributions of X and Y:

$$\begin{aligned} a_i &= \sqrt{\left(\sqrt{2k} + X_i\sigma\sqrt{Z/\alpha}\right)^2 + \left(Y_i\sigma\sqrt{Z/\alpha}\right)^2}, \\ X_i &= \rho X_{i-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\,1), \\ Y_i &= \rho Y_{i-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\,1), \end{aligned} \tag{11.7}$$

where X_i and Y_i are correlated with previous samples and ρ controls the correlation. The correlation coefficient of HK-distribution versus ρ is illustrated in Fig. 11.3 for $\alpha = 3, k = 0.1$. We select a small value of $\rho = 0.2$ for training and evaluate the performance for different values of ρ for the evaluation.

To generate training data, $log_{10}(\alpha)$ is randomly selected from values ranging -0.3 to 1.3, corresponding to α of 0.5 to 20. k is also randomly selected from values ranging from 0 to 1.25. Results are reported for the sample sizes of $N_s = 65536$, $N_s = 4096$, and $N_s = 1024$.

The simulation test data is generated for 31 different values of $log_{10}(\alpha) \in \{-0.3, ..., 1.3\}$, and 11 values of $k \in \{0, ..., 1.25\}$. For each value of $log_{10}(\alpha)$ and k, 10 realizations are generated, giving 3410 sample sets for each sample size and ρ. Data is generated for three sample sizes, $N_s \in \{65536, 4096, 1024\}$, and two correlation values, $\rho \in \{0.2, 0.6\}$.

Evaluation metric for simulation test data: The Relative Root Mean Square Error (RRMSE) and the Mean Absolute Error (MAE) are employed to evaluate the methods for simulation test data in which the ground truth is known, and they can be defined as:

$$\begin{aligned} RRMSE &= \sqrt{\frac{< (y - \widetilde{y})^2 >}{|y| + \gamma}}, \\ MAE &= < |y - \widetilde{y}| >, \end{aligned} \tag{11.8}$$

where y and $\widetilde{y}$ are the ground truth and estimated parameter, respectively, $< . >$ is the averaging operation and γ is a small non-negative value (here we use 0.05) to avoid division by zero.

Experimental Phantom Data

A three-layered phantom having different scatterer number density was constructed from an emulsion of ultrafiltered milk and water-based gelatin. 5-43 μm diameter glass beads (3000E, Potters Industries, Valley Forge, PA, USA) were used as the source of scattering. Data was acquired by a 18L6 probe, linear array transducer operating at the center frequency of 8.9 MHz, using a Siemens Acuson S2000 scanner (Siemens Medical Solutions USA, Inc.). Data collected from this phantom was primarily reported in [159]. The top and bottom layers have the same properties, whereas the middle layer has a higher backscattering coefficient and scatterer concentration. The backscattering coefficient of the middle layer is 6.37×10^{-3} $cm^{-1}sr^{-1}$ and it is 3.52×10^{-3} $cm^{-1}sr^{-1}$ for the top and bottom layers at the center frequency. In this phantom, the only source of intensity change to have different backscattering coefficients is the density of scatterers. Therefore, the ratio of backscattering coefficients, $\frac{6.37\times10^{-3}cm^{-1}sr^{-1}}{3.52\times10^{-3}cm^{-1}sr^{-1}} = 1.81$, can be an indicator of the true ratio of α which we employ as a metric to verify the estimated α.

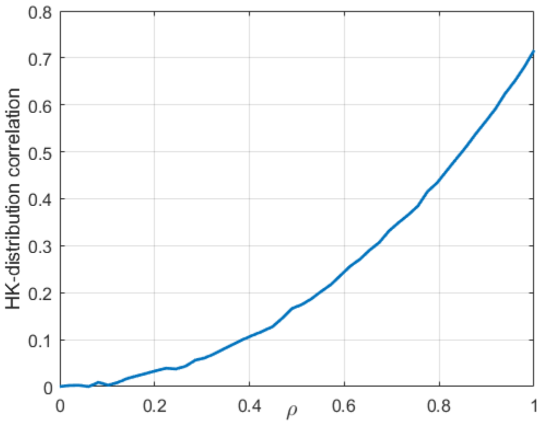


Figure 11.3: The correlation coefficient of HK-distribution samples generated by Eq 11.7 versus ρ for $\alpha = 3, k = 0.1$.

Table 11.1: p-values Wilcoxon sign tests for $log_{10}(\alpha)$ and $\rho = 0.2$. If a p-value is less than 0.0001, it is shown by * to represent high statistical significance.

Methods	Ns = 65536	Ns = 4096	Ns = 1024
XU vs BNN	*	*	*
XU vs MPAE-b3	*	*	*
XU vs MPAE-b2	*	*	*
MPAE-b3 vs BNN	*	*	*
MPAE-b3 vs MPAE-b2	*	*	*
MPAE-b2 vs BNN	*	*	*

In vivo Data

Duck liver data were used to compare the performance of the evaluated methods. The protocol was approved by the animal ethical care committee of the University of Montreal hospital research centre, Montreal, QC, Canada. The ducks' liver data were collected before and 14 days after force feeding to study the formation of fatty liver. Data acquisition was performed as part of a study conducted by Bhattet *et al.* [171]. A Verasonics Vantage programmable system (Verasonics Inc., Kirkland, WA) with an ATL L7-4 linear probe (Philips, Bothell, WA) operating at the center frequency of 5 MHz was employed for data acquisition. Two ducks were randomly selected to investigate the performance of the compared methods.

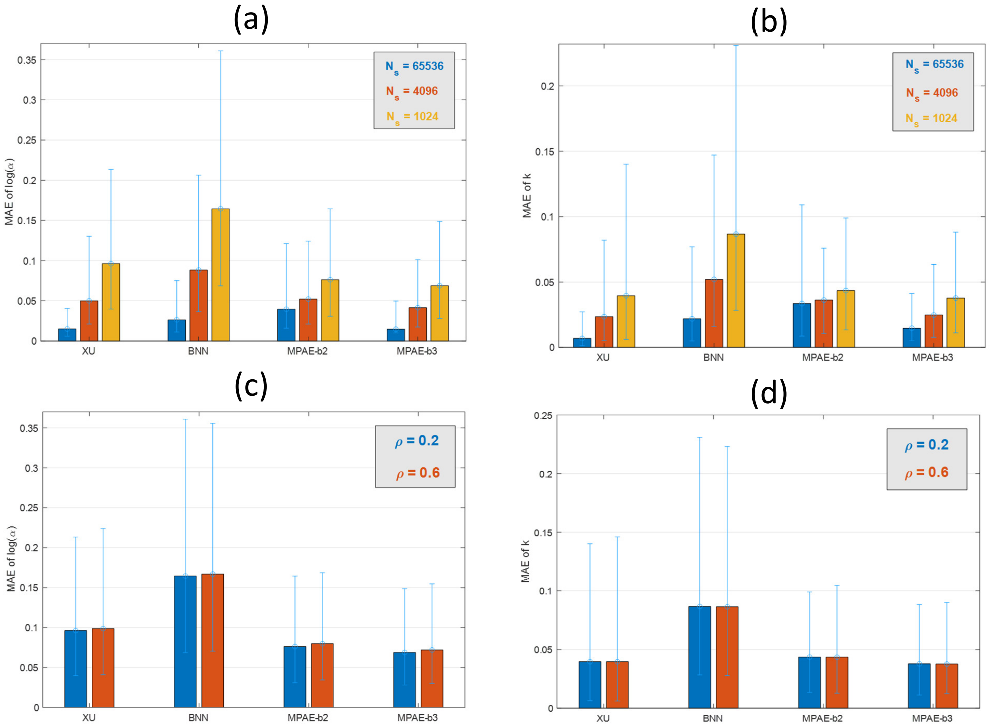


Figure 11.4: The plots of the median and 25%-75% percentile of MAE errors of different evaluated methods. The median and 25%-75% percentile of MAE of $log_{10}(\alpha)$ of different sample sizes (a). The median and 25%-75% percentile of MAE of k for different sample sizes (b). The median and 25%-75% percentile of MAE of $log_{10}(\alpha)$ for $\rho = 0.2, 0.6$ using the sample size of $N_s = 1024$ (c). The median and 25%-75% percentile of MAE of k for $\rho = 0.2, 0.6$ using the sample size of $N_s = 1024$ (d). $MPAE - b2$ and $MPAE - b3$ denote MPAE with bottleneck size of 2 and 3 is used for feature reconstruction and BNN is employed as the estimator.

11.3 Results

In this section, we evaluate the performance of the proposed method and compare it with the XU method.

11.3.1 Simulation Results

Three methods, XU, BNN, and MPAE are evaluated. It should be noted that for MPAE method, we employ BNN as the estimator after the feature reconstruction of MPAE.

The median MAE of simulation test data are illustrated in Fig. 11.4. By inspecting the results we can observe:

MPAE performance: from Fig. 11.4 (a) and (b), it can be seen that MPAE substantially

Table 11.2: p-values of Wilcoxon sign tests for k and $\rho = 0.2$. If a p-value is less than 0.0001, it is shown by * to represent high statistical significance.

Methods	Ns = 65536	Ns = 4096	Ns = 1024
XU vs BNN	*	*	*
XU vs MPAE-b3	*	*	*
XU vs MPAE-b2	*	0.140	*
MPAE-b3 vs BNN	*	*	*
MPAE-b3 vs MPAE-b2	*	*	*
MPAE-b2 vs BNN	0.012	*	*

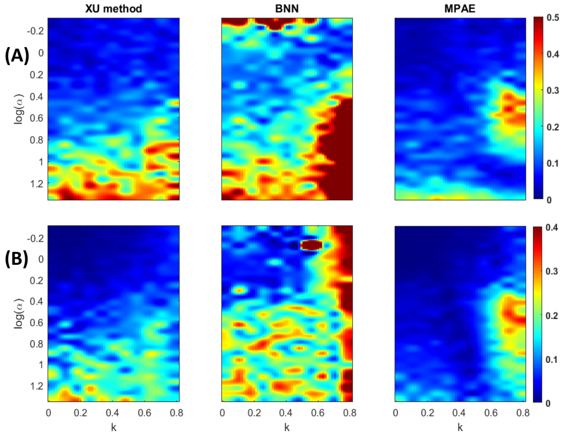


Figure 11.5: The MAE error map of $log_{10}(\alpha)$ (A), and k (B) for $N_s = 1024, \rho = 0.2$. The error is averaged over 10 realizations for each grid point. MPAE has the bottleneck size of 3.

improves the performance of BNN estimator. For a large sample size of $N_s = 65536$, XU estimator has the lowest error among the compared methods but the sample size is decreased, MPAE outperforms the XU estimator.

Bottleneck size: Two bottleneck sizes of 2 ($MPAE{-}b2$) and 3 ($MPAE{-}b3$) are evaluated. The error of the bottleneck size of 3 is lower than 2 in all cases, indicating that at least three variables are needed to encode QUS properties.

Correlation (ρ): To see the effect of correlation, we trained MPAE on low correlation value ($\rho = 0.2$), and the method is evaluated on $\rho = 0.2, 0.6$. It can be seen from Fig. 11.4 (c) and (d) that increasing the correlation results in a slight increase in the $log_{10}(\alpha)$ error, while there is no significant change for k.

Error map: The reported results are the overall error that do not show the error variations for different HK parameter values. The error map for different values of $log_{10}(\alpha)$ and k are

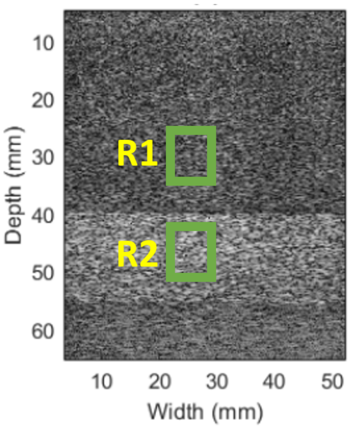


Figure 11.6: The B-mode image of the layered phantom. Two patches for extraction of the statistical features are specified. The patches are moved laterally across several frames to extract multiple features.

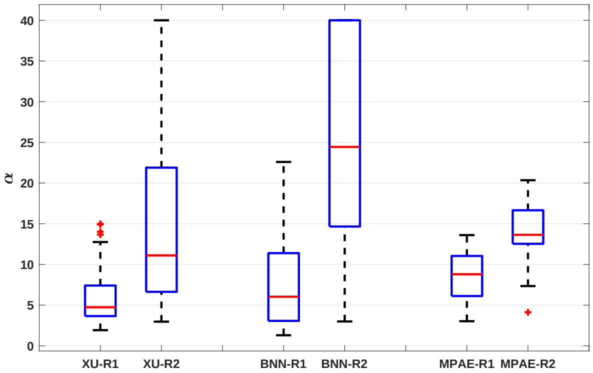


Figure 11.7: The boxplot of α of the layered phantom using the evaluated methods. MPAE denotes using MPAE with bottleneck of 3 and trained with $N_s = 4096$. The correct ratio of α of R1 and R2 is 1.81. The average ratios of the estimated α are 2.73, 3.51, and 1.67 for XU, BNN, and MPAE, respectively (using 90% percentile). MPAE provides the lowest variance in estimation and has the closest ratio to the correct one.

illustrated for $N_s = 1024, \rho = 0.2$ in Fig. 11.5. MPAE substantially decreases BNN's error, yielding a lower error than the XU method.

Statistical test: Wilcoxon sign test is performed to see if the difference between the median of the errors of the methods is statistically significant. Wilcoxon sign test is a paired test that is chosen since the MAE errors are not normally distributed [172]. The p-values are reported in Table 11.1 for $log_{10}(\alpha)$ and in Table 11.2 for k. The majority of the pairs are statistically different (except XU vs MPAE-b2 and MPAE-b2 vs BNN for k)

Table 11.3: The correlation between the parametric images of the evaluated methods.

Methods	Duck A before force feeding	Duck B before force feeding	Duck A after force feeding	Duck B after force feeding
XU, BNN	0.495	0.399	0.619	0.674
MPAE, BNN	0.499	0.692	0.831	0.897
MPAE, XU	0.752	0.774	0.563	0.742

11.3.2 Experimental Phantom Results

The B-mode image of the phantom is illustrated in Fig. 11.6. A 12.56 × 7.81mm patch is employed for statistical feature extraction. To reduce the correlation, 5 samples in the axial direction corresponding to 1 pulse length and 1 sample in the lateral direction are skipped, which gives 3400 samples (N_s). The patch is moved laterally across 12 frames to obtain 60 samples of feature statistics for each of the two regions (R1 and R2). Although the exact value of α is unknown, the backscattering coefficient ratio can be used to verify the results.

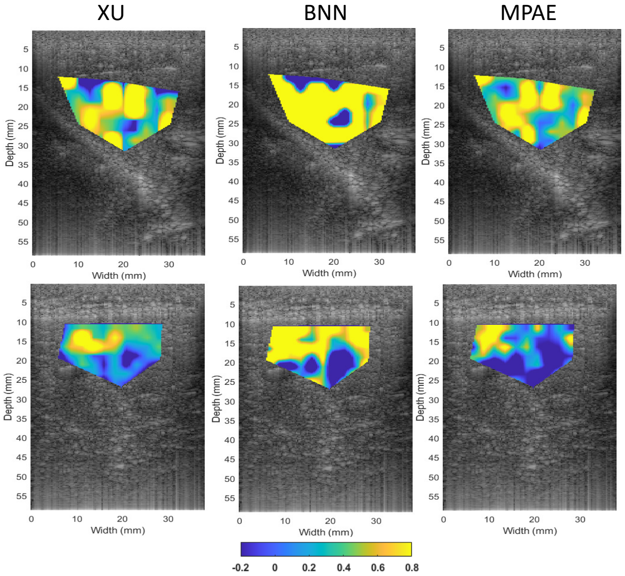


Figure 11.8: The parametric image of $log_{10}(\alpha)$ of duck A before force feeding (top row), and after (bottom row).

The box plot of the evaluated methods is illustrated in Fig. 11.7. $MPAE$ denotes using MPAE with the bottleneck of 3, trained with $N_s = 4096$. The correct ratio of α of R1 and R2 is 1.81 (obtained from the backscattering coefficient). The average ratios of estimated α are 2.73, 3.51, and 1.67 for XU, BNN, and MPAE, respectively (using 90% percentile).

The variances of R1 region are 3.32, 5.76, and 2.78 for XU, BNN, and MPAE, respectively,

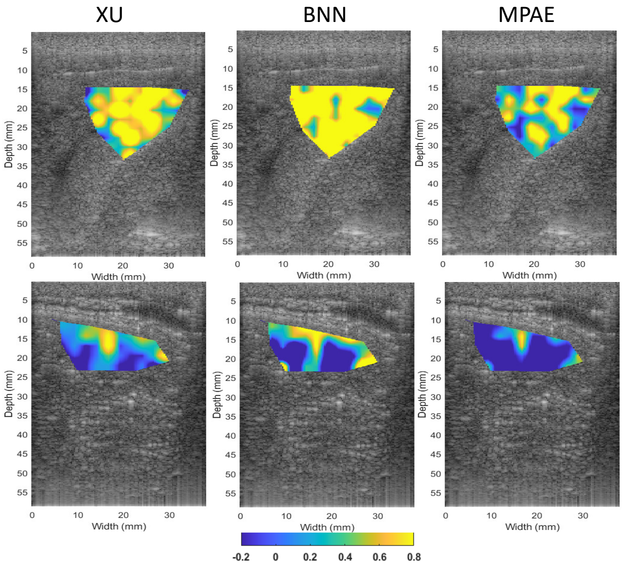


Figure 11.9: The parametric image of $log_{10}(\alpha)$ of duck B before force feeding (top row), and after (bottom row).

and they are 12.15, 12.37, and 3.15 for region R2. MPAE provides a close average ratio of α values and has substantially lower variance compared to the other methods, which confirms that using this network prior to an estimator (here, BNN is employed) is beneficial. The over-estimation of α and the high variance can be observed in XU and BNN results. Zhou *et al.* showed that when the sample size is not large enough, over-estimation of α occurs [165], which is also confirmed by our results. MPAE has a substantially lower bias, and the average α values (1.67) ratio is close to the actual ratio (1.81).

11.3.3 *In vivo* Results

To compute statistical features, we selected patches of size $5.5mm \times 5.5mm$. These patches were moved with a 63% overlap to ensure complete coverage of the liver area. To minimize the impact of sample heterogeneity, we employed a patchless deep neural network, previously developed for scatterer number density regression (Chapter 9), and applied a k-medoid clustering algorithm to select only samples that belong to the same class as the center sample. This allowed us to compute feature statistics accurately and avoid errors arising from sample heterogeneity.

The parametric images of $log_{10}(\alpha)$ are illustrated in Fig. 11.8 for duck A and duck B

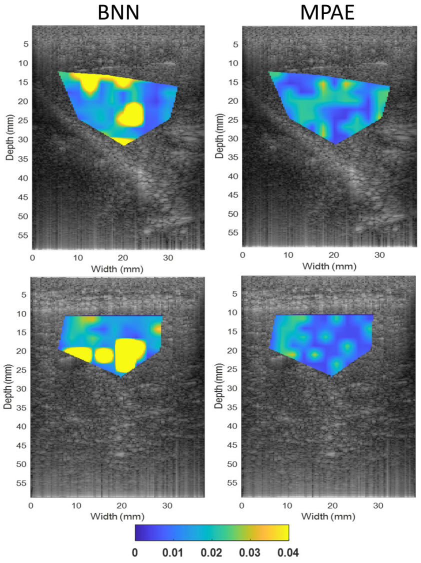


Figure 11.10: The parametric image of the uncertainty of BNN of duck A before force feeding (top row), and after (bottom row).

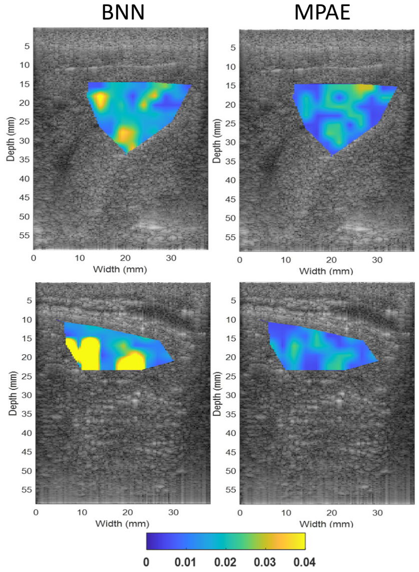


Figure 11.11: The parametric image of the uncertainty of BNN of duck B before force feeding (top row), and after (bottom row).

in Fig. 11.9 before force feeding (top row) and after (bottom row). The exact parametric image is unknown; Gesnik *et al.* reported that the average of $1/\alpha$ of the whole liver region considering all studied ducks increases from $1/\alpha = 0.69 \pm 0.10$ to $1/\alpha = 0.94 \pm 0.07$ after 14 days of force feeding [173]. Therefore, generally, lower α is expected after force feeding than before that. According to the given parametric images, XU and MPAE results are highly correlated. XU and MPAE both produce high values in the same locations; while BNN produces very high values of $log_{10}(\alpha)$ in most parts.

The correlation between different pairs are reported in Table 11.3 which indicates that the correlation between MPAE and XU is higher than the correlation between XU and BNN for most cases (except duck A after force feeding). Both ducks' liver $log_{10}(\alpha)$ parametric images obtained by XU and MPAE have reduction of portion of high values after force feeding (Figs. 11.8 and 11.9) which is expected as the reduction was reported in [173].

Uncertainty of the estimation is another aspect that can be investigated. As discussed earlier, BNN estimator can also provide the uncertainty by taking the variance of the estimates using different realizations of the network's weights. The parametric image of the uncertainty is illustrated in Fig. 11.10 for duck A, and in Fig. 11.11 for duck B. It is clear that BNN has substantially lower uncertainty when the features are reconstructed using MPAE.

11.4 Discussion

In this chapter, we proposed a model projection autoencoder to reconstruct clean statistical features from the noisy sample estimate. Any estimator can be used after the proposed method to estimate HK parameters. In the manuscript, we investigated BNN for the estimation step. However, any method could be used in the estimation step.

Source of noise is another aspect that should be investigated further. In this chapter, the noise emanates from the low sample size. Other sources of noise, including the presence of samples from other HK-distributions can be used as the source of noise and requires further investigation.

11.5 Summary

In this chapter, we propose a model projection neural network based on denoising autoencoders to reconstruct refined statistical features to improve HK-distribution parameter estimation. The method is validated using simulation data, experimental phantom, and clinical data.

Chapter 12

Conclusion and Future Work

In this thesis, novel methods were developed for ultrasound-based displacement estimation and QUS parameter estimation from backscattered envelope data. The presented methods were either under review or published in peer-reviewed journals and conferences. In this chapter, we first provide concluding remarks, and then suggest avenues for future work.

12.1 Conclusion

In this thesis, several innovative methods were proposed to improve displacement estimation in USE and enveloped-based QUS parameters. In USE displacement estimation, networks' architecture was modified to facilitate RF data feature extraction [55, 59]. Two modified networks, MPWC-Net and RFMPWC-Net were proposed. We also improved the displacement range of MPWC-Net by increasing the search range of the correlation layer. Semi-supervised training was another avenue we followed to enhance the performance further [56, 81], and the improvement was validated using experimental phantom and *in vivo* data. Lateral strain images obtained by semi-supervised training were noisy and unreliable; therefore, we focused on improving the lateral displacement by incorporating the physical constraints [79, 91]. We demonstrated that by adding the constraint as a regularization, the lateral strain, which is required for effective Poisson's ratio imaging and for solving the inverse problem of estimating the elasticity modulus, can be improved substantially. The method was evaluated using simulation data, experimental phantom, and *in vivo* data.

On the QUS front, we started with patch-based classification [145] and extended that to full image segmentation of scatterer number density [157]. A fast grid-based simulation method was proposed that enabled us to generate thousands of US envelope data in a short amount of time with diverse imaging settings. In the next step, scatterer number density regression was accomplished [174] by using the fast grid-based simulation as the training data generation method. Improving the patch-based estimation and quantifying the uncertainty was a problem we addressed in the next step [167]. Uncertainty quantification is essential and clinically relevant since it allows the clinician to effectively utilize the parametric images only when the results are reliable enough. The proposed architecture obtained the uncertainty by sampling the network's weights from distributions whose parameters were learned during the training. Furthermore, a model projection autoencoder was proposed to reduce the noise due to the low sample size on the estimated statistical features. The autoencoder tried to map the noisy sample estimate to the theoretical values. We validated the method using simulation, experimental phantom, and duck liver data.

12.2 Future Work

Regarding USE, improved network architecture can be an area of future work. More advanced architectures such as Transformers [175], which have first been used in ultrasound images in for the classification of breast cancer with promising results [176], can be employed to improve the feature extraction or optical flow estimation part of the network.

Another modification in the network architecture is to consider the ambiguity in the warping operation. Asymmetric feature matching, proposed in [72], can be a good choice. In this work, the correlation layers considered the ambiguity in the optical flow caused by occlusion in the warping operation. The proposed module can be integrated into any optical flow network. Another area of modifying the architecture is to employ a more sophisticated correlation module. Globally optimized correspondence is a recent correlation module [177] that has been recently utilized in optical flow networks for USE [178]. We hypothesize that by using this module, small displacement tracking can be improved since a higher quality correlation between features can be achieved.

Large displacements are also complex to estimate; in MPWC-Net++, we alleviated this issue at the expense of increasing the computation cost by enlarging the correlation search range. Registration methods can be employed to compensate for the large motion and then a more compact network architecture can be used to find the residual motion [179].

Beamforming also affects displacement estimation, especially in the lateral direction. The beamforming method we used was delay and sum (DAS). More advanced beamforming methods can produce narrower PSF which improves the displacement estimation.

Regarding the QUS parameter estimation, we assumed that there is a single distribution within a patch. However, this assumption is not valid for *in vivo* data which is highly heterogeneous. We partially alleviated this issue by using the patchless estimator as a preprocessing step to find where there were changes of scatterer number density. Clustering was used in the next step to select only samples that belong to the same class. Further investigation is required since the patchless estimator only finds large variations across the region of interest. There might be samples that are sparsely distributed and cause errors, especially when the size of the patch is small. A possible approach is to have a network that clusters and segments the samples based on their distribution. In our previous work, we segmented the entire data by high- and low-number of scatterers, but here the problem is a more complex one since the network should be able to segment each individual sample separately.

Aberration and other ultrasound artifacts can adversely affect the QUS estimation methods. Aberration correction methods can be utilized before the QUS parameter estimation network to compensate for the aberration effect.

In Chapter 10, we investigated an autoencoder to reconstruct the statistical features from the sample estimates. The network was trained in a supervised manner. Self-supervision can be used to employ experimental phantom and *in vivo* data during training to mitigate the domain shift problem.

The presence of the domain shift is also another problem that we partially alleviated in Chapters 7 and 8. We mainly focused on BN layers and proposed to employ AdaBN layers, but more complicated methods, such as zone training and calibration, can be employed. Zone training presented in [180] can aid in reducing the diffraction effect and reduce the impact of changing the imaging settings. Reference phantom normalization has also been recently applied to deep learning methods in QUS [181], and their influence can be investigated in our methods.

Estimating other QUS parameters, such as attenuation and backscattering coefficients, can be an area of future work. Currently, we investigated the estimation of coherent to diffuse scattering ratio and scatterer number density. Different QUS parameters should be employed to diagnose the abnormalities in the tissue and using only one QUS biomarker would not be reliable enough.

Bibliography

[1] K.-Y. Chin and S. Ima-Nirwana, "Calcaneal quantitative ultrasound as a determinant of bone health status: what properties of bone does it reflect?," *International journal of medical sciences*, vol. 10, no. 12, p. 1778, 2013.

[2] M.-C. Ho *et al.*, "Using ultrasound nakagami imaging to assess liver fibrosis in rats," *Ultrasonics*, vol. 52, no. 2, pp. 215–222, 2012.

[3] P.-H. Tsui, C.-K. Yeh, C.-C. Chang, and Y.-Y. Liao, "Classification of breast masses by ultrasonic nakagami imaging: a feasibility study," *Physics in Medicine & Biology*, vol. 53, no. 21, p. 6027, 2008.

[4] P.-H. Tsui, Z. Zhou, Y.-H. Lin, C.-M. Hung, S.-J. Chung, and Y.-L. Wan, "Effect of ultrasound frequency on the nakagami statistics of human liver tissues," *PloS one*, vol. 12, no. 8, 2017.

[5] M. L. Oelze and J. Mamou, "Review of quantitative ultrasound: Envelope statistics and backscatter coefficient imaging and contributions to diagnostic ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 2, pp. 336–351, 2016.

[6] J. Bercoff, M. Tanter, and M. Fink, "Supersonic shear imaging: a new technique for soft tissue elasticity mapping," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 51, no. 4, pp. 396–409, 2004.

[7] F. Deeba *et al.*, "A spatially weighted regularization method for attenuation coefficient

estimation," in *2019 IEEE International Ultrasonics Symposium (IUS)*, pp. 2023–2026, IEEE, 2019.

[8] T. J. Hall, Y. Zhu, and C. S. Spalding, "In vivo real-time freehand palpation imaging," *Ultrasound in medicine & biology*, vol. 29, no. 3, pp. 427–435, 2003.

[9] A. C. Ugural and S. K. Fenster, *Advanced strength and applied elasticity.* Pearson education, 2003.

[10] J. Ophir, S. K. Alam, B. Garra, F. Kallel, E. Konofagou, T. Krouskop, and T. Varghese, "Elastography: ultrasonic estimation and imaging of the elastic properties of tissues," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 213, no. 3, p. 203–233, 1999.

[11] T. Varghese, E. Konofagou, J. Ophir, S. Alam, and M. Bilgen, "Direct strain estimation in elastography using spectral cross-correlation," *Ultrasound in medicine & biology*, vol. 26, no. 9, p. 1525–1537, 2000.

[12] R. F. Wagner, M. F. Insana, and D. G. Brown, "Statistical properties of radio-frequency and envelope-detected signals with applications to medical ultrasound," *JOSA A*, vol. 4, no. 5, pp. 910–922, 1987.

[13] R. F. Wagner, "Statistics of speckle in ultrasound b-scans," *IEEE Trans. Sonics & Ultrason.*, vol. 30, no. 3, pp. 156–163, 1983.

[14] F. Destrempes, J. Porée, and G. Cloutier, "Estimation method of the homodyned k-distribution based on the mean intensity and two log-moments," *SIAM journal on imaging sciences*, vol. 6, no. 3, pp. 1499–1530, 2013.

[15] P. M. Shankar *et al.*, "Classification of ultrasonic b-mode images of breast masses using nakagami distribution," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 2, pp. 569–580, 2001.

[16] T. J. Hall, Y. Zhu, C. S. Spalding, and L. T. Cook, "In vivo results of real-time freehand elasticity imaging," in *2001 IEEE Ultrasonics Symposium. Proceedings. An International Symposium (Cat. No. 01CH37263)*, vol. 2, p. 1653–1657, IEEE, 2001.

[17] A. Kling and J. Jiang, "Potential of determining thermal dose for ablation therapies using ultrasound elastography: An ex vivo feasibility study," in *2018 IEEE International Ultrasonics Symposium (IUS)*, p. 1–4, IEEE, 2018.

[18] J. Jiang, T. Varghese, C. L. Brace, E. L. Madsen, T. J. Hall, S. Bharat, M. A. Hobson, J. A. Zagzebski, and F. T. Lee, "Young's modulus reconstruction for radio-frequency ablation electrode-induced displacement fields: a feasibility study," *IEEE transactions on medical imaging*, vol. 28, no. 8, p. 1325–1334, 2009.

[19] F.-F. Lee, Q. He, J. Gao, A. Pan, S. Sun, X. Liang, and J. Luo, "Evaluating hifu-mediated local drug release using thermal strain imaging: Phantom and preliminary in-vivo studies," *Medical physics*, 2019.

[20] R. M. Sigrist, J. Liau, A. El Kaffas, M. C. Chammas, and J. K. Willmann, "Ultrasound elastography: review of techniques and clinical applications," *Theranostics*, vol. 7, no. 5, p. 1303, 2017.

[21] H. Zhi, B. Ou, B.-M. Luo, X. Feng, Y.-L. Wen, and H.-Y. Yang, "Comparison of ultrasound elastography, mammography, and sonography in the diagnosis of solid breast lesions," *Journal of ultrasound in medicine*, vol. 26, no. 6, p. 807–815, 2007.

[22] A. Nahiyan and M. K. Hasan, "Hybrid algorithm for elastography to visualize both solid and fluid-filled lesions," *Ultrasound in medicine & biology*, vol. 41, no. 4, p. 1058–1078, 2015.

[23] J. Luo and E. E. Konofagou, "A fast normalized cross-correlation calculation method for motion estimation," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 57, no. 6, p. 1347–1357, 2010.

[24] J. Jiang and T. J. Hall, "A parallelizable real-time motion tracking algorithm with applications to ultrasonic strain imaging," *Physics in Medicine & Biology*, vol. 52, no. 13, p. 3773, 2007.

[25] M. Mirzaei, A. Asif, M. Fortin, and H. Rivaz, "3d normalized cross-correlation for

estimation of the displacement field in ultrasound elastography," *Ultrasonics*, vol. 102, p. 106053, 2020.

[26] T. J Hall, P. E Barboneg, A. A Oberai, J. Jiang, J.-F. Dord, S. Goenezen, and T. G Fisher, "Recent results in nonlinear strain and modulus imaging," *Current medical imaging reviews*, vol. 7, no. 4, p. 313–327, 2011.

[27] H. S. Hashemi and H. Rivaz, "Global time-delay estimation in ultrasound elastography," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 64, no. 10, p. 1625–1636, 2017.

[28] H. Rivaz, E. M. Boctor, M. A. Choti, and G. D. Hager, "Real-time regularized ultrasound elastography," *IEEE transactions on medical imaging*, vol. 30, no. 4, p. 928–945, 2010.

[29] M. Mirzaei, A. Asif, and H. Rivaz, "Combining total variation regularization with window-based time delay estimation in ultrasound elastography.," *IEEE transactions on medical imaging*, vol. 38, no. 12, pp. 2744–2754, 2019.

[30] M. Ashikuzzaman, C. J. Gauthier, and H. Rivaz, "Global ultrasound elastography in spatial and temporal domains," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 66, no. 5, p. 876–887, 2019.

[31] A. K. Z. Tehrani, M. Mozaffarzadeh, Z. Mardi, S. H. Hozhabr, M. Mehrmohammadi, and B. Makkiabadi, "Application of demons algorithm in ultrasound elastography using b-mode ultrasound images," in *Photons Plus Ultrasound: Imaging and Sensing 2019*, vol. 10878, p. 108786P, International Society for Optics and Photonics, 2019.

[32] M. G. Kibria and H. Rivaz, "Gluenet: Ultrasound elastography using convolutional neural network," in *Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation*, p. 21–28, Springer, 2018.

[33] Z. Gao, S. Wu, Z. Liu, J. Luo, H. Zhang, M. Gong, and S. Li, "Learning the implicit strain reconstruction in ultrasound elastography using privileged information," *Medical image analysis*, vol. 58, p. 11–18, 2019.

[34] B. Peng, Y. Xian, and J. Jiang, "A convolution neural network-based speckle tracking method for ultrasound elastography," in *2018 IEEE International Ultrasonics Symposium (IUS)*, p. 206–212, IEEE, 2018.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, p. 1097–1105, 2012.

[36] 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*, p. 234–241, Springer, 2015.

[37] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 8934–8943, 2018.

[38] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

[39] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2462–2470, 2017.

[40] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 4161–4170, 2017.

[41] S. Wu, Z. Gao, Z. Liu, J. Luo, H. Zhang, and S. Li, "Direct reconstruction of ultrasound elastography using an end-to-end deep neural network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, p. 374–382, Springer, 2018.

[42] C. Hoerig, J. Ghaboussi, and M. F. Insana, "Data-driven elasticity imaging using cartesian neural network constitutive models and the autoprogressive method," *IEEE transactions on medical imaging*, vol. 38, no. 5, p. 1150–1160, 2018.

[43] X. Pan, K. Liu, J. Shao, J. Gao, L. Huang, J. Bai, and J. Luo, "Performance comparison of rigid and affine models for motion estimation using ultrasound radio-frequency signals," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 62, no. 11, p. 1928–1943, 2015.

[44] M. Kibria and H. Rivaz, "Global ultrasound elastography using convolutional neural network," *arXiv preprint arXiv:1805.07493*, 2018.

[45] J. A. Jensen, "Field: A program for simulating ultrasound systems," in *10TH Norticbaltic Conference On Biomedical Imaging, Vol. 4, Suppliment 1, Part 1: 351–353*, Citeseer, 1996.

[46] J. A. Jensen and N. B. Svendsen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 39, no. 2, p. 262–267, 1992.

[47] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, p. 834–848, 2017.

[48] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, p. 3320–3328, 2014.

[49] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," *arXiv preprint arXiv:1801.05746*, 2018.

[50] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, p. 694–711, Springer, 2016.

[51] S. Rezajoo and A. R. Sharafat, "Robust estimation of displacement in real-time freehand ultrasound strain imaging," *IEEE transactions on medical imaging*, vol. 37, no. 7, p. 1664–1677, 2018.

[52] C. Rabin and N. Benech, "Quantitative breast elastography from b-mode images," *Medical physics*, 2019.

[53] T. Ersepke, T. C. Kranemann, and G. Schmitz, "On the performance of time domain displacement estimators for magnetomotive ultrasound imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 66, no. 5, p. 911–921, 2019.

[54] M. T. Islam, A. Chaudhry, S. Tang, E. Tasciotti, and R. Righetti, "A new method for estimating the effective poisson's ratio in ultrasound poroelastography," *IEEE transactions on medical imaging*, vol. 37, no. 5, p. 1178–1191, 2018.

[55] A. K. Tehrani and H. Rivaz, "Displacement estimation in ultrasound elastography using pyramidal convolutional neural network," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 67, no. 12, pp. 2629–2639, 2020.

[56] A. K. Tehrani, M. Mirzaei, and H. Rivaz, "Semi-supervised training of optical flow convolutional neural networks in ultrasound elastography," *International conference on Medical Image Computing & Computer Assisted Intervention (MICCAI)*, 2020.

[57] B. Peng, Y. Xian, Q. Zhang, and J. Jiang, "Neural network-based motion tracking for breast ultrasound strain elastography: An initial assessment of performance and feasibility," *Ultrasonic Imaging*, p. 0161734620902527, 2020.

[58] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5754–5763, 2019.

[59] A. K.Z. Tehrani and H. Rivaz, "Mpwc-net++: evolution of optical flow pyramidal convolutional neural network for ultrasound elastography," in *Medical Imaging 2021: Ultrasonic Imaging and Tomography*, vol. 11602, p. 1160206, International Society for Optics and Photonics, 2021.

[60] R. Delaunay, Y. Hu, and T. Vercauteren, "An unsupervised approach to ultrasound elastography with end-to-end strain regularisation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 573–582, Springer, 2020.

[61] R. Delaunay, Y. Hu, and T. Vercauteren, "An unsupervised learning approach to ultrasound strain elastography with spatio-temporal consistency," *Physics in Medicine & Biology*, 2021.

[62] A. K. Tehrani, M. Amiri, and H. Rivaz, "Real-time and high quality ultrasound elastography using convolutional neural network by incorporating analytic signal," in *42nd Annual International Conferences of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2020.

[63] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8981–8989, 2018.

[64] C.-L. Hung, C.-f. Hsin, H.-H. Wang, and C. Y. Tang, "Optimization of gpu memory usage for training deep neural networks," in *International Symposium on Pervasive Systems, Algorithms and Networks*, pp. 289–293, Springer, 2019.

[65] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, "What matters in unsupervised optical flow," *arXiv preprint arXiv:2006.04902*, vol. 1, no. 2, p. 3, 2020.

[66] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

[67] M. Ashikuzzaman, A. Sadeghi-Naini, A. Samani, and H. Rivaz, "Combining first and second order continuity constraints in ultrasound elastography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2021.

[68] E. J. Chen, J. Novakofski, W. K. Jenkins, and W. D. O'Brien, "Young's modulus measurements of soft tissues with application to elasticity imaging," *IEEE Transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 43, no. 1, pp. 191–194, 1996.

[69] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[70] N. Mohammadi, M. M. Doyley, and M. Cetin, "Ultrasound elasticity imaging using physics-based models and learning-based plug-and-play priors," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1165–1169, IEEE, 2021.

[71] E. Theodorsson-Norheim, "Friedman and quade tests: Basic computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples," *Computers in biology and medicine*, vol. 17, no. 2, pp. 85–99, 1987.

[72] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu, *et al.*, "Maskflownet: Asymmetric feature matching with learnable occlusion mask," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6278–6287, 2020.

[73] X. Wei, Y. Wang, L. Ge, B. Peng, Q. He, R. Wang, L. Huang, Y. Xu, and J. Luo, "Unsupervised convolutional neural network for motion estimation in ultrasound elastography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2022.

[74] O. A. Babaniyi, A. A. Oberai, and P. E. Barbone, "Recovering vector displacement estimates in quasistatic elastography using sparse relaxation of the momentum equation," *Inverse problems in science and engineering*, vol. 25, no. 3, pp. 326–362, 2017.

[75] L. Guo, Y. Xu, Z. Xu, and J. Jiang, "A pde-based regularization algorithm toward reducing speckle tracking noise: A feasibility study for ultrasound breast elastography," *Ultrasonic imaging*, vol. 37, no. 4, pp. 277–293, 2015.

[76] Y. Wang, X. Xie, Q. He, H. Liao, H. Zhang, and J. Luo, "Hadamard-encoded synthetic transmit aperture imaging for improved lateral motion estimation in ultrasound elastography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 69, no. 4, pp. 1204–1218, 2022.

[77] M. Mirzaei, A. Asif, and H. Rivaz, "Virtual source synthetic aperture for accurate lateral displacement estimation in ultrasound elastography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 5, pp. 1687–1695, 2020.

[78] S. Selladurai and A. K. Thittai, "Strategies to obtain subpitch precision in lateral motion estimation in ultrasound elastography," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 65, no. 3, pp. 448–456, 2018.

[79] A. KZ Tehrani and H. Rivaz, "Physically inspired constraint for unsupervised regularized ultrasound elastography," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 218–227, Springer, 2022.

[80] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[81] A. K. Tehrani, M. Sharifzadeh, E. Boctor, and H. Rivaz, "Bi-directional semi-supervised training of convolutional neural networks for ultrasound elastography displacement estimation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2022.

[82] L. Ma and A. M. Korsunsky, "The principle of equivalent eigenstrain for inhomogeneous inclusion problems," *International Journal of Solids and Structures*, vol. 51, no. 25-26, pp. 4477–4484, 2014.

[83] P. Mott and C. Roland, "Limits to poisson's ratio in isotropic materials—general result for arbitrary deformation," *Physica Scripta*, vol. 87, no. 5, p. 055404, 2013.

[84] R. Righetti, J. Ophir, S. Srinivasan, and T. A. Krouskop, "The feasibility of using

elastography for imaging the poisson's ratio in porous media," *Ultrasound in medicine & biology*, vol. 30, no. 2, pp. 215–228, 2004.

[85] D. George, M. Baniassadi, Y. Hoarau, M. Kugler, and Y. Rémond, "Influence of the liver vascular distribution on its overall mechanical behavior: a first approach to multiscale fluid-structure homogenization," *Journal of Cellular Immunotherapy*, vol. 4, no. 1, pp. 35–37, 2018.

[86] R. H. Griesenauer, J. A. Weis, L. R. Arlinghaus, I. M. Meszoely, and M. I. Miga, "Breast tissue stiffness estimation for surgical guidance using gravity-induced excitation," *Physics in Medicine & Biology*, vol. 62, no. 12, p. 4756, 2017.

[87] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6489–6498, 2020.

[88] P. Liu, M. Lyu, I. King, and J. Xu, "Selflow: Self-supervised learning of optical flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4571–4580, 2019.

[89] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Computer Vision, IEEE International Conference on*, vol. 2, pp. 10–10, IEEE Computer Society, 2003.

[90] A. K. Maier, C. Syben, B. Stimpel, T. Würfl, M. Hoffmann, F. Schebesch, W. Fu, L. Mill, L. Kling, and S. Christiansen, "Learning with known operators reduces maximum error bounds," *Nature machine intelligence*, vol. 1, no. 8, pp. 373–380, 2019.

[91] A. K. Z. Tehrani, M. Ashikuzzaman, and H. Rivaz, "Lateral strain imaging using self-supervised and physically inspired constraints in unsupervised regularized elastography," *IEEE Transactions on Medical Imaging*, vol. 42, no. 5, pp. 1462–1471, 2023.

[92] V. Dutt and J. F. Greenleaf, "Speckle analysis using signal to noise ratios based on fractional order moments," *Ultrasonic Imaging*, vol. 17, pp. 251–268, 1995.

[93] K. Nam, J. A. Zagzebski, and T. J. Hall, “Simultaneous backscatter and attenuation estimation using a least squares method with constraints,” *Ultrasound in Medicine and Biology*, vol. 37, no. 12, pp. 2096–2104, 2011.

[94] M. F. Insana, R. F. Wagner, D. G. Brown, and T. J. Hall, “Describing small-scale structure in random media using pulse-echo ultrasound,” *The Journal of the Acoustical Society of America*, vol. 87, no. 1, pp. 179–192, 1990.

[95] J. Mamou and M. L. Oelze, *Quantitative ultrasound in soft tissues.* Springer, 2013.

[96] Z. Vajihi, I. M. Rosado-Mendez, T. J. Hall, and H. Rivaz, “Low variance estimation of backscatter quantitative ultrasound parameters using dynamic programming,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 11, pp. 2042–2053, 2018.

[97] J. Rouyer, T. Cueva, T. Yamamoto, A. Portal, and R. J. Lavarello, “In vivo estimation of attenuation and backscatter coefficients from human thyroids,” *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 63, no. 9, pp. 1253–1261, 2016.

[98] F. L. Lizzi, S. K. Alam, S. Mikaelian, P. Lee, and E. J. Feleppa, “On the statistics of ultrasonic spectral parameters,” *Ultrasound in medicine & biology*, vol. 32, no. 11, pp. 1671–1685, 2006.

[99] N. J. Bureau, F. Destrempes, S. Acid, E. Lungu, T. Moser, J. Michaud, and G. Cloutier, “Diagnostic accuracy of echo envelope statistical modeling compared to b-mode and power doppler ultrasound imaging in patients with clinically diagnosed lateral epicondylosis of the elbow,” *Journal of Ultrasound in Medicine*, vol. 38, no. 10, pp. 2631–2641, 2019.

[100] A. M. Pirmoazen, A. Khurana, A. El Kaffas, and A. Kamaya, “Quantitative ultrasound approaches for diagnosis and monitoring hepatic steatosis in nonalcoholic fatty liver disease,” *Theranostics*, vol. 10, no. 9, p. 4277, 2020.

[101] A. Larrue and J. A. Noble, “Modeling of errors in nakagami imaging: Illustration

on breast mass characterization," *Ultrasound in Medicine and Biology*, vol. 40, no. 5, pp. 917–930, 2014.

[102] L. Clifford, P. Fitzgerald, and D. James, "Non-rayleigh first-order statistics of ultrasonic backscatter from normal myocardium," *Ultrasound in Medicine and Biology*, vol. 19, no. 6, pp. 487–495, 1993.

[103] L. X. Yao, J. A. Zagzebski, and E. L. Madsen, "Backscatter coefficient measurements using a reference phantom to extract depth-dependent instrumentation factors," *Ultrasonic imaging*, vol. 12, no. 1, pp. 58–70, 1990.

[104] G. Cloutier, M. Daronat, D. Savéry, D. Garcia, L.-G. Durand, and F. S. Foster, "Non-gaussian statistics and temporal variations of the ultrasound signal backscattered by blood at frequencies between 10 and 58 mhz," *The Journal of the Acoustical Society of America*, vol. 116, no. 1, pp. 566–577, 2004.

[105] E. Jakeman, "Speckle statistics with a small number of scatterers," *Optical Engineering*, vol. 23, no. 4, p. 234453, 1984.

[106] A. H. Gee, R. J. Housden, P. Hassenpflug, G. M. Treece, and R. W. Prager, "Sensorless freehand 3d ultrasound in real tissue: speckle decorrelation without fully developed speckle," *Medical image analysis*, vol. 10, no. 2, pp. 137–149, 2006.

[107] N. Rubert and T. Varghese, "Scatterer number density considerations in reference phantom-based attenuation estimation," *Ultrasound in medicine & biology*, vol. 40, no. 7, pp. 1680–1696, 2014.

[108] H. Rivaz, E. M. Boctor, and G. Fichtinger, "Ultrasound speckle detection using low order moments," in *2006 IEEE Ultrasonics Symposium*, pp. 2092–2095, Oct 2006.

[109] I. M. Rosado-Mendez, L. C. Drehfal, J. A. Zagzebski, and T. J. Hall, "Analysis of coherent and diffuse scattering using a reference phantom," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 9, pp. 1306–1320, 2016.

[110] R. W. Prager, A. H. Gee, G. M. Treece, and L. H. Berman, "Analysis of speckle in ultrasound images using fractional order statistics and the homodyned k-distribution," *Ultrasonics*, vol. 40, no. 1-8, pp. 133–137, 2002.

[111] P.-H. Tsui, C.-K. Yeh, C.-C. Chang, and W.-S. Chen, "Performance evaluation of ultrasonic nakagami image in tissue characterization," *Ultrasonic imaging*, vol. 30, no. 2, pp. 78–94, 2008.

[112] P.-H. Tsui, C.-C. Huang, L. Sun, S. H. Dailey, and K. K. Shung, "Characterization of lamina propria and vocal muscle in human vocal fold tissue by ultrasound nakagami imaging," *Medical physics*, vol. 38, no. 4, pp. 2019–2026, 2011.

[113] P.-H. Tsui *et al.*, "Small-window parametric imaging based on information entropy for ultrasound tissue characterization," *Scientific Reports*, vol. 7, no. 41004, 2017.

[114] M. Amiri, A. K. Tehrani, and H. Rivaz, "Segmentation of ultrasound images based on scatterer density using u-net," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2063–2066, IEEE, 2020.

[115] K. M. Meiburger, U. R. Acharya, and F. Molinari, "Automated localization and segmentation techniques for b-mode ultrasound images: A review," *Computers in biology and medicine*, vol. 92, pp. 210–235, 2018.

[116] R. J. Van Sloun, O. Solomon, M. Bruce, Z. Z. Khaing, Y. C. Eldar, and M. Mischi, "Deep learning for super-resolution vascular ultrasound imaging," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1055–1059, IEEE, 2019.

[117] K. G. Brown, D. Ghosh, and K. Hoyt, "Deep learning of spatiotemporal filtering for fast super-resolution ultrasound imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020.

[118] S. Goudarzi, A. Asif, and H. Rivaz, "Multi-focus ultrasound imaging using generative

adversarial networks," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 1118–1121, 2019.

[119] Z. Gao *et al.*, "Learning the implicit strain reconstruction in ultrasound elastography using privileged information," *Medical image analysis*, vol. 58, p. 101534, 2019.

[120] E. Evain, K. Faraz, T. Grenier, D. Garcia, M. De Craene, and O. Bernard, "A pilot study on convolutional neural networks for motion estimation from ultrasound images," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020.

[121] M. Byra, H. Piotrzkowska-Wróblewska, K. Dobruch-Sobczak, and A. Nowicki, "Combining nakagami imaging and convolutional neural network for breast lesion classification," in *2017 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, 2017.

[122] Y. Wang *et al.*, "Deeply-supervised networks with threshold loss for cancer detection in automated breast ultrasound," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 866–876, 2020.

[123] K. Stacke, G. Eilertsen, J. Unger, and C. Lundström, "A closer look at domain shift for deep learning in histopathology," *arXiv preprint arXiv:1909.11575*, 2019.

[124] A. K. Tehrani, M. Amiri, I. M. Rosado-Mendez, T. J. Hall, and H. Rivaz, "A pilot study on scatterer density classification of ultrasound images using deep neural networks," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2059–2062, IEEE, 2020.

[125] L. Zhang, V. Vishnevskiy, and O. Goksel, "Deep network for scatterer distribution estimation for ultrasound image simulation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020.

[126] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[127] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[128] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[129] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[130] H. Rivaz, R. Zellars, G. Hager, G. Fichtinger, and E. Boctor, “9c-1 beam steering approach for speckle characterization and out-of-plane motion estimation in real tissue,” in *2007 IEEE Ultrasonics Symposium Proceedings*, pp. 781–784, IEEE, 2007.

[131] I. M. Rosado-Mendez, *Advanced spectral analysis methods for quantification of coherent ultrasound scattering: Applications in the breast.* PhD thesis, The University of Wisconsin-Madison, 2014.

[132] S. S. Brunke, M. F. Insana, J. J. Dahl, C. Hansen, M. Ashfaq, and H. Ermert, “An ultrasound research interface for a clinical system.,” *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 54, no. 1, pp. 198–210, 2007.

[133] R. Prager, A. Gee, G. Treece, and L. Berman, “Speckle detection in ultrasound images using first order statistics,” *University of Cambridge, Department of Engineering*, 2001.

[134] P.-H. Tsui and C.-C. Chang, “Imaging local scatterer concentrations by the nakagami statistical model,” *Ultrasound in medicine & biology*, vol. 33, no. 4, pp. 608–619, 2007.

[135] T. Tuthill, R. Sperry, and K. Parker, “Deviations from rayleigh statistics in ultrasonic speckle,” *Ultrasonic imaging*, vol. 10, no. 2, pp. 81–89, 1988.

[136] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[137] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, pp. 30–43, 2018.

[138] L. Wang, C.-Y. Lee, Z. Tu, and S. Lazebnik, "Training deeper convolutional networks with deep supervision," *arXiv preprint arXiv:1505.02496*, 2015.

[139] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[140] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, IEEE, 2017.

[141] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[142] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[143] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

[144] L. Du, "How much deep learning does neural style transfer really need? an ablation study," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3150–3159, 2020.

[145] A. K. Tehrani, M. Amiri, I. M. Rosado-Mendez, T. J. Hall, and H. Rivaz, "Ultrasound scatterer density classification using convolutional neural networks and patch statistics," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2021.

[146] N. Jafarpisheh, T. J. Hall, H. Rivaz, and I. M. Rosado-Mendez, "Analytic global regularized backscatter quantitative ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020.

[147] B. Burger, S. Bettinghausen, M. Radle, and J. Hesser, "Real-time gpu-based ultrasound simulation using deformable mesh models," *IEEE transactions on medical imaging*, vol. 32, no. 3, pp. 609–618, 2012.

[148] H. Gao, H. F. Choi, P. Claus, S. Boonen, S. Jaecques, G. H. Van Lenthe, G. Van der Perre, W. Lauriks, and J. D'hooge, "A fast convolution-based methodology to simulate 2-dd/3-d cardiac ultrasound images," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 56, no. 2, pp. 404–409, 2009.

[149] Y. Zhu and T. J. Hall, "A modified block matching method for real-time freehand strain imaging," *Ultrasonic imaging*, vol. 24, no. 3, pp. 161–176, 2002.

[150] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[151] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

[152] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[153] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[154] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[155] M. Buda, A. Saha, and M. A. Mazurowski, "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm," *Computers in biology and medicine*, vol. 109, pp. 218–225, 2019.

[156] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[157] A. K. Tehrani, I. M. Rosado-Mendez, and H. Rivaz, “Robust scatterer number density segmentation of ultrasound images,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 69, no. 4, pp. 1169–1180, 2022.

[158] D. P. Hruska and M. L. Oelze, “Improved parameter estimates based on the homodyned k distribution,” *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 56, no. 11, pp. 2471–2481, 2009.

[159] K. Nam, I. M. Rosado-Mendez, L. A. Wirtzfeld, G. Ghoshal, A. D. Pawlicki, E. L. Madsen, R. J. Lavarello, M. L. Oelze, J. A. Zagzebski, W. D. O’Brien Jr, *et al.*, “Comparison of ultrasound attenuation and backscatter estimates in layered tissue-mimicking phantoms among three clinical scanners,” *Ultrasonic imaging*, vol. 34, no. 4, pp. 209–221, 2012.

[160] S. C. Yurtkulu, Y. H. Şahin, and G. Unal, “Semantic segmentation with extended deeplabv3 architecture,” in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2019.

[161] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” in *International Conference on Learning Representations*, 2019.

[162] Z. Zhou, A. Gao, W. Wu, D.-I. Tai, J.-H. Tseng, S. Wu, and P.-H. Tsui, “Parameter estimation of the homodyned k distribution based on an artificial neural network for ultrasound tissue characterization,” *Ultrasonics*, vol. 111, p. 106308, 2021.

[163] A. Gao, S. Wu, D.-I. Tai, Z. Zhou, and P.-H. Tsui, “Ultrasonic evaluation of liver fibrosis using the homodyned k distribution with an artificial neural network estimator,” in *2021 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, IEEE, 2021.

[164] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, “Hands-on

bayesian neural networks—a tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29–48, 2022.

[165] Z. Zhou, J. Fang, A. Cristea, Y.-H. Lin, Y.-W. Tsai, Y.-L. Wan, K.-M. Yeow, M.-C. Ho, and P.-H. Tsui, "Value of homodyned k distribution in ultrasound parametric imaging of hepatic steatosis: An animal study," *Ultrasonics*, vol. 101, p. 106001, 2020.

[166] Y. Liu, B. He, Y. Zhang, X. Lang, R. Yao, and L. Pan, "A study on a parameter estimator for the homodyned k distribution based on table search for ultrasound tissue characterization," *Ultrasound in Medicine & Biology*, 2023.

[167] A. K. Tehrani, I. M. Rosado-Mendez, and H. Rivaz, "Homodyned k-distribution: parameter estimation and uncertainty quantification using bayesian neural networks," *arXiv preprint arXiv:2211.00175*, 2022.

[168] S. Ladjal, A. Newson, and C.-H. Pham, "A pca-like autoencoder," *arXiv preprint arXiv:1904.01277*, 2019.

[169] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[170] X. Wu, K. Lv, S. Wu, D.-I. Tai, P.-H. Tsui, and Z. Zhou, "Parallelized ultrasound homodyned-k imaging based on a generalized artificial neural network estimator," *Ultrasonics*, p. 106987, 2023.

[171] M. Bhatt, L. Yazdani, F. Destrempes, L. Allard, B. N. Nguyen, A. Tang, and G. Cloutier, "Multiparametric in vivo ultrasound shear wave viscoelastography on farm-raised fatty duck livers: human radiology imaging applied to food sciences," *Poultry science*, vol. 100, no. 4, p. 100968, 2021.

[172] J. H. McDonald, *Handbook of biological statistics*, vol. 2. sparky house publishing Baltimore, MD, 2009.

[173] M. Gesnik, M. Bhatt, M.-H. R. Cardinal, F. Destrempes, L. Allard, B. N. Nguyen, T. Alquier, J.-F. Giroux, A. Tang, and G. Cloutier, “In vivo ultrafast quantitative ultrasound and shear wave elastography imaging on farm-raised duck livers during force feeding,” *Ultrasound in medicine & biology*, vol. 46, no. 7, pp. 1715–1726, 2020.

[174] A. K. Z. Tehrani, I. M. Rosado-Mendez, H. Whitson, and H. Rivaz, “A deep learning approach for patchless estimation of ultrasound quantitative parametric image with uncertainty measurement,” in *Medical Imaging 2023: Ultrasonic Imaging and Tomography* (C. Boehm and N. Bottenus, eds.), vol. 12470, p. 1247010, International Society for Optics and Photonics, SPIE, 2023.

[175] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[176] B. Gheflati and H. Rivaz, “Vision transformers for classification of breast ultrasound images,” in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 480–483, IEEE, 2022.

[177] P. Truong, M. Danelljan, L. V. Gool, and R. Timofte, “Gocor: Bringing globally optimized correspondence volumes into your neural network,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 14278–14290, 2020.

[178] S. Wen, B. Peng, X. Wei, J. Luo, and J. Jiang, “Convolutional neural network-based speckle tracking for ultrasound strain elastography: An unsupervised learning approach,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2023.

[179] H. Zhou and H. Rivaz, “Registration of pre-and postresection ultrasound volumes with noncorresponding regions in neurosurgery,” *IEEE journal of biomedical and health informatics*, vol. 20, no. 5, pp. 1240–1249, 2016.

[180] U. Soylu and M. L. Oelze, “A data-efficient deep learning strategy for tissue characterization via quantitative ultrasound: Zone training,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2023.

[181] U. Soylu and M. L. Oelze, “Calibrating data mismatches in deep learning-based quantitative ultrasound using setting transfer functions,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2023.