Fine-tuning U-Net for ultrasound image segmentation: different layers, different outcomes.

Mina Amiri, Rupert Brooks, and Hassan Rivaz

***Abstract*—One way to resolve the problem of scarce and expensive data in deep learning for medical applications is using transfer learning and fine-tuning a network which has been trained on a large dataset. The common practice in transfer learning is to keep the shallow layers unchanged and to modify deeper layers according to the new dataset. This approach may not work when using a U-Net and when moving from a different domain to ultrasound (US) images due to their drastically different appearance. In this study, we investigated the effect of fine-tuning different sets of layers of a pre-trained U-Net for US image segmentation. Two different schemes were analyzed, based on two different definitions of shallow and deep layers. We studied simulated US images, as well as two human US datasets. We also included a chest X-ray dataset. The results showed that choosing which layers to fine-tune is a critical task. In particular, they demonstrated that fine-tuning the last layers of the network, which is the common practice for classification networks, is often the worst strategy. It may therefore be more appropriate to fine-tune the shallow layers rather than deep layers in US image segmentation when using a U-Net. Shallow layers learn lower level features which are critical in automatic segmentation of medical images. Even when a large US dataset is available, we also observed that fine-tuning shallow layers is a faster approach compared to fine-tuning the whole network.**

***Index Terms*—Ultrasound imaging, Segmentation, Transfer learning, U-Net.**

I. Introduction

TRAINING a deep convolutional neural network (CNN) from scratch is not easy, particularly in medical applications, where generating annotated data requires spending a large amount of time and money. Transfer learning is an alternative to full training, where the knowledge learned by a network on a different and usually large dataset is transferred to another application. This can be done by fine-tuning a few layers or retraining the whole network. It has been shown in several studies that it is feasible to use non-medical images (for instance natural images) as the source dataset for transfer learning to the domain of medical images [1]–[3]. This way, the model benefits from having a large number of available images for training, which at a minimum provides a suitable parameter initialization for further training in the new domain. When the target dataset in the new domain is small, the recommended approach in transfer learning is fine-tuning, e.g. to keep the shallow layers of the network unchanged, and to update the deep layers according to the new dataset [4].

It has been shown that low-level features are learned by shallow layers of a CNN, while more semantic and high-level features are recognized by deeper layers [5]. Therefore, the common approach of fine-tuning the deepest layers of a network stems from the assumption that low-level features of different datasets (associated with shallow layers) are similar, and high-level features of datasets (associated with deeper layers) are specific to those datasets and should be learned independently for each application. This assumption may not hold true in medical applications, for example when applying transfer learning from natural images to ultrasound (US) imaging, the source and target datasets are extremely different. Even basic and low-level features could be substantially different for medical images compared to natural images.

US imaging is a standard modality for many diagnostic and monitoring purposes including heart and vascular imaging, breast cancer screening and fetus monitoring. Breast US segmentation and tumor region extraction is an important step in clinical diagnosis of breast cancer which is the most common form of cancer among women worldwide. Based on the segmentation results, the tumor can be categorized and further clinical actions can be planned. There has been significant research into developing automatic methods for segmentation of breast US images as well as other anatomical structures (such as prostate, kidney, fetus, etc.) [6], [7]. The automatic methods proposed for breast US segmentation can be classified into thresholding-based, clustering-based, watershed-based, graph-based, active contour model, Markov random field and classic machine learning methods [8], [9]. Deep learning techniques such as CNNs have also been widely utilized recently [10]–[12]. U-Net [13], for instance, has been shown to be a fast and precise technique for medical image segmentation, and has been successfully adapted to segment US images [12], [14]–[17]. It was indeed shown to be the best architecture for the segmentation of US images [18]. Several methods have also been proposed for segmentation of breast volumetric images [19], [20] including the 3D version of U-Net [21].

Another important and common application of medical US is monitoring and screening pregnant women. During the US screening examination, several measurements of the fetus are computed to assess its growth. Among them, the head circumference is a critical index of the gestational age and the fetal development process. Several methods have been proposed for automatic head circumference measurement using US data [22]–[25].

M. Amiri is a postdoctoral fellow at the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada. e-mail: amirim@encs.concordia.ca.

R. Brooks is with Nuance Communications and also with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada.

H. Rivaz is with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada.

Many previous works on US have used transfer learning due to the limited data, but unique characteristics of US have not been considered. In particular, US is a coherent imaging modality, where constructive interference of the scattered waves leads to a characteristic speckle noise, which is not present in natural images or images from other medical modalities (e.g. radiographs).

U-Net transfer learning has been studied for magnetic resonance images when a model has been pre-trained on a large number of medical images of a specific disease and has been utilised for a different disease [26]. However, only the last layers and the decoder (expanding) path have been fine-tuned, and no analysis has been done on shallow layers. This is indeed the common approach in transfer learning, which may not be the correct approach in segmentation applications when using U-Net. In this study, we questioned the effectiveness of fine-tuning the last layers of a U-Net in segmentation.

We investigated transfer learning when we had a domain shift from natural images to medical US images. Because of the specific structure of the U-net and its skip connection layers, there is some ambiguity in the definition of deep and shallow layers in this network, something that we had not considered in our recent work [27]. Herein, we analyzed the effect of fine-tuning a pre-trained U-Net network in several different experiments based on two different definitions of deep and shallow layers to find the best strategy for transfer learning of US images. We studied segmentation of simulated US data as well as breast and fetal US images. We also included an X-ray dataset as control. Our main findings are as follows:

- Unlike the common approach in classification, fine-tuning the last layers of a U-Net does not provide good results in US image segmentation.
- Removing the bottleneck (shown as block 5 in Fig. 1) from fine-tuning results in an equivalent performance as fine-tuning the whole network. The number of parameters in the bottleneck is about half the number of parameters in the whole network, an important advantage for freezing the bottleneck.
- It is not just the number of parameters which predicts the performance of a fine-tuning strategy. The depth and connections of a layer are also critical.

II. Methodology

This section provides an overview of the network and different datasets used in this study. Details of pre training, transfer learning and fine-tuning the network are also presented.

A. Network Architecture

We used nearly the same U-Net architecture proposed in the original paper [13], except for having replaced the transposed convolutional layers by bilinear upsampling followed by 2x2 convolution. The network consisted of blocks of two 3x3 convolutional layers with ReLU activation. Each block was connected to the next block by either a maxpooling or an upsampling operation (Fig. 1). Each layer in the first block had 64 filters. After each maxpooling operation, the number of filters was increased by a factor of two and after each upsampling operation, the number of filters was decreased by a factor of two. The last layer was a 1×1 convolutional layer with sigmoid activation to map the feature vector to the interval of 0 and 1. For evaluation purposes, we considered the threshold value of 0.5, so that pixels with values above 0.5 were considered as 1, while pixels with values below 0.5 were considered as 0. We did not use batch normalization, but we used the dropout technique (50%) just after the bottleneck (block 5). In total, this network had around 31 million parameters.

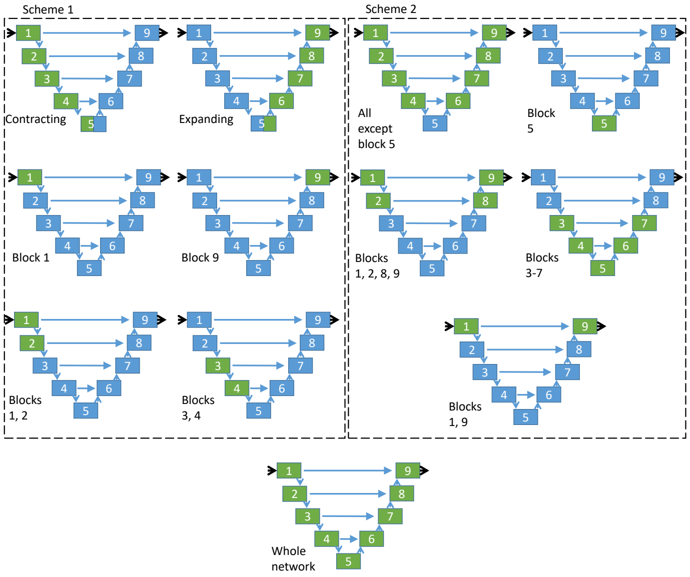

Fig. 1. Schematic of the U-Net [13] and the fine-tuning strategies. Green blocks are the blocks included in fine-tuning, and blue ones are frozen.

B. Experimental Design

Due to the presence of skip connections, the notion of shallow or deep layers in the network is less straightforward in a U-Net configuration than in a typical feedforward classification network. We considered two conceptual subdivisions of the network to explore which parts are more relevant for fine tuning. Within each conceptual subdivision of the network, the experiments empirically compared several possible ways to select components for fine tuning.

One can consider the U-Net as an autoencoder with skip connections. From this point of view, layer 1 (at the start of the encoder) would be the shallowest, and layer 9 (at the output of the decoder) would be the deepest. We called this interpretation scheme 1. We began by dividing the network precisely in half by path length, splitting in the middle of the bottleneck layer. As early experiments showed significantly greater gains when the contracting (encoder) component was fine-tuned, this component was further subdivided, with experiments that quantified the gain from fine-tuning only the first block (1), the first two blocks (1,2) or the second half of the encoder (3,4). We also studied block 9 only as the common approach in transfer learning.

Alternatively, one may consider the top of the “U” to be the shallowest part due to the presence of the skip connections, and the bottleneck to be the deepest part. This is closely related to the interpretation of residual networks as an ensemble of networks within a network [28], where each possible path through the network contributes to the whole, and shorter paths have greater importance. From this point of view, the shorter the minimum paths through the network that a block contributes to, the “shallower” it is. Due to the network structure, this also has the effect that in this interpretation the shallower blocks contribute to more total paths than the deeper ones. Note that layers 1 and 9 are involved in all paths through the network, and layer 5 is only involved on one path through the network. We have described this family of approaches as scheme 2.

In scheme 2, we began by dividing the network as closely in half as possible (in terms of number of parameters). The bottleneck (block 5) alone contains approximately half of the parameters of the network. Further subdivisions consistent with this interpretation were also considered; blocks 1-9 alone, blocks 1,2,8,9 alone and blocks 3-7 alone.

C. Datasets

In order to pre-train the network, we used the XPIE dataset which contains 10000 segmented natural images [29]. The images in this dataset are not gray scale. To have a more similar pre-training dataset to the US dataset, we converted these images into black and white prior to feeding to the network. The pre-trained network was then used for the task of segmentation of US B-mode images.

Our first US dataset consists of 85 simulated US B-mode images (SUS) generated by a MATLAB-based publicly available US simulation software, Field II [30], [31]. The simulation was done for 50 RF lines, with a center frequency of 3.5 MHz and sampling frequency of 100MHz. Each image had a random number of circle or ellipsoid shape hypoechoic lesions. The intensities for the lesions were set k times the background ($0 < k < 1$). The lesions were randomly located inside the simulated phantom. More details of the simulation procedure can be found in [32].

We also studied a breast US imaging dataset (BUS) containing 163 images of the breast with either benign lesions or malignant tumors [11]. We also included another US dataset containing 805 unique 2D images of fetal head (FUS) [33]. The main purpose of this dataset was to automatically measure the head circumference, however, in this study we used it for segmentation of the head. It is evident that a good head segmentation will result in a good head circumference estimation.

In order to investigate whether the results were specific to the US imaging, we repeated the analysis for a chest X-ray dataset with a total of 240 images [34], wherein we used the pre-trained network to segment both lungs. Fig. 2 shows a few examples of images from the different datasets used in this study.

Data Augmentation: As the size of US and X-ray datasets was small, we implemented data augmentation techniques to improve the network performance, invariance and robustness. For these datasets, the network should be robust to shift, flipping, shearing and zooming. We implemented on-the-fly data augmentation techniques by generating smooth deformations of images using random and small degrees of shifting (with the shift range of 0.05 of the total length), zooming (random zoom in the range of ± 0.05), and horizontal flipping. For pre-training the network using natural images, we did not augment the data, as the size of the dataset was large, and the network was only used as a pre-trained network.

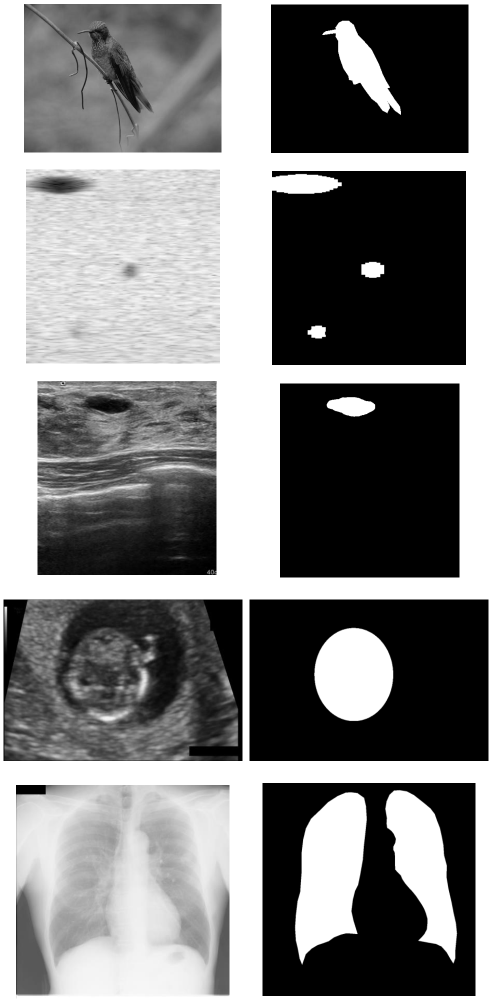

Fig. 2. Some examples from different datasets used in this study and their associated masks. From top to bottom: XPIE dataset, simulated data, breast US dataset, fetal US dataset, and chest X-ray dataset. The XPIE images have very different appearances when compared to X-ray or US images. The speckle noise due to the image formation process is apparent in the US images.

D. Analysis

We first trained a U-Net using the XPIE dataset. The weights of this pre-trained network were then utilized as an initial point for all subsequent fine tunings of the network, across all folds. All images were resized to 256 × 256 pixels and were normalized to [0,1]. We used 5-fold cross validation to evaluate the performance of the network. Each dataset was randomly divided into five folds. Each fold was used as a held out test set while the network is trained using the other 4 folds. In the training procedure, 20% of the training data was used for validation.

To investigate the effect of the dataset size on the results, we repeated the whole five-fold cross validation using 80, 50 and 5 percent of the data as the training set, 20% of which used for validation (respectively 16%, 10% and 1% of the entire dataset). The total number of images used in each scenario for all datasets is presented in Table I. The 50 or 5 percent of the total dataset were selected out of the training data.

We employed the early stopping technique, wherein training stops when a minimum loss was achieved on the validation set and no improvement was observed for 20 epochs, and the best performing model on the validation set was saved for evaluating the test set. Training was performed using ADAM, a binary cross entropy loss and a batch size of 2. Learning rate was set as 10^{-4}. If early stopping did not occur in 200 epochs, the training was considered failed.

To be able to compare different fine-tuning scenarios, all experiments used the same folds, and the performance of the network was then assessed using the same test set (the entire held out fold) for 5, 50 and 80% of the data. To compensate for the fewer number of images when analyzing 5 or 50% of the data and therefore fewer number of optimization steps, we considered the same number of batches in an epoch for all different experiments in each dataset, equal to the number of batches when 80% of the data was used for training.

E. Performance Metric

To evaluate the performance of the network in segmenting the images, we used Dice score. Dice score is an index of similarity between two samples, defined as:

$$Dice\ score = \frac{2TP}{2TP + FN + FP} \tag{1}$$

where TP (true-positive) denotes the number of elements correctly predicted as the mask, FN (false-negative) denotes the number of ground-truth mask elements falsely predicted as the background, and FP (false-positive) denotes the number of elements in the background, falsely predicted as the mask. We employed t-test to compare results of fine-tuning the shallow vs. deep paths in two schemes. We also tested whether transfer learning significantly improved the segmentation and whether including more images in the training phase affected the results. We corrected all results for multiple comparisons using Tukey's method.

III. Results

The results of different fine-tuning schemes on 80, 50 and 5 percent of the datasets are provided in this section. The average Dice score of different strategies for all datasets and all training-set sizes are provided in Table II.

A. The best and the worst strategies

Overall, among all different strategies we included in this study, fine-tuning the whole network, all network except block 5 and the contracting path yielded the best results. It is interesting that removing the bottleneck (block 5), with 45.6% of total number of parameters, from the fine-tuning did not reduce the results considerably. In fact, when the number of images was low (training with 5% of SUS, BUS and X-ray datasets), removing block 5 from fine-tuning resulted in the best performance, even better than fine-tuning the whole network.

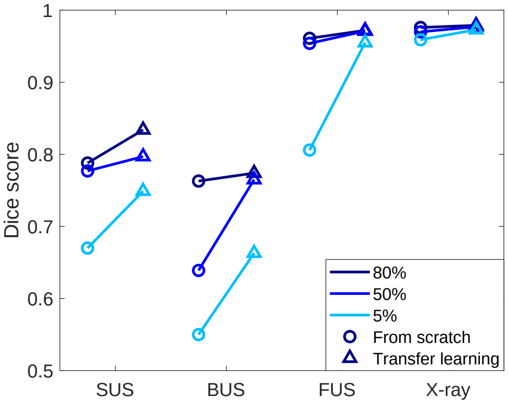

Fig. 3. Average Dice score when the network is trained from scratch compared to when the pre-trained network is totally fine-tuned, for different numbers of images.

Fine-tuning blocks 1, 9 and 5 were the worst strategies. Block 9 is the last block according to scheme 1 and block 5 is the deepest block according to scheme 2. It is therefore evident that fine-tuning last layers in US image segmentation using U-Net is not a good practice.

B. Transfer-learning vs. training from scratch

Fig. 3 represents the results of comparing the segmentation performance when the network was trained from scratch and when the pre-trained network is fine-tuned using different number of images. For all datasets and all different fractions (80%, 50% and 5%) studied in this paper, fine-tuning the whole pre-trained network outperformed training from scratch (Table II). This difference was significant ($p < 0.05$, Tukey corrected) when 5% of the data was used in all datasets, but did not reach statistical significance when a greater number of images was used. However, when training the network from scratch, the training was not as easy and stable as employing transfer learning. The convergence rate was 100% when retraining the pre-trained network even with 5% of the data, but training did not converge in 200 epochs in several folds when training from scratch. When using 80% of the data, training failed in 1 out of 5 folds for SUS and BUS datasets. When using 50% of the data, training failed in 1 out of 5 folds for BUS and X-ray datasets, and when using 5% of the data, training failed in 2 out of 5 folds for SUS, BUS and FUS datasets. The results presented in the figure are derived by averaging over successful training instances. It is important to note that we were able to train the network by changing parameter initialization in the failed folds.

TABLE I
THE NUMBER OF IMAGES IN TRAINING, VALIDATION AND TEST SETS IN EACH EXPERIMENT.

Dataset	Size	80%		50%		5%		Test
		Training	Validation	Training	Validation	Training	Validation	(20%)
SUS	85	54	14	34	9	3	1	17
BUS	163	104	26	66	16	6	2	33
FUS	805	515	129	322	81	32	8	161
X-ray	240	154	38	96	24	10	2	48

TABLE II
THE AVERAGE AND STANDARD DEVIATION (IN PARENTHESES) OF DICE SCORE FOR DIFFERENT DATASETS AND DIFFERENT SIZES OF THE TRAINING SET. THE COLOR CODING REPRESENT THE PERFORMANCE OF EACH SCENARIO IN EACH ROW (FROM GREEN TO RED: FROM THE BEST TO THE WORST PERFORMANCE). THE NUMBER OF PARAMETERS AND THE FRACTION OF TOTAL PARAMETERS IN EACH EXPERIMENT ARE PRESENTED IN FIRST ROWS.

		Whole network	Contra cting	Expan ding	Except block 5	Block 5	Blocks 1,2,8,9	Blocks 3- 7	Block 1	Block 9	Blocks 1, 9	Blocks 1, 2	Blocks 3, 4	from scratch
Number of parameters		3.1e+7	9.4e+6	2.2e+7	1.7e+7	1.4e+7	9.8e+5	3.0e+7	3.8e+4	1.4e+5	1.8e+5	2.6e+5	4.4e+6	3.1e+7
Fraction of total parameters		1	0.303	0.697	0.544	0.456	0.032	0.968	0.001	0.005	0.006	0.008	0.142	1
SUS	80%	0.834 (0.056)	0.823 (0.031)	0.830 (0.042)	0.826 (0.054)	0.661 (0.047)	0.787 (0.051)	0.799 (0.052)	0.620 (0.044)	0.711 (0.033)	0.754 (0.030)	0.744 (0.037)	0.718 (0.036)	0.788 (0.025)
	50%	0.797 (0.029)	0.775 (0.024)	0.772 (0.029)	0.789 (0.035)	0.638 (0.045)	0.784 (0.034)	0.773 (0.026)	0.550 (0.074)	0.666 (0.071)	0.638 (0.034)	0.716 (0.050)	0.707 (0.043)	0.777 (0.036)
	5%	0.749 (0.046)	0.743 (0.041)	0.710 (0.050)	0.754 (0.072)	0.549 (0.058)	0.748 (0.044)	0.700 (0.060)	0.56 (0.060)	0.609 (0.175)	0.731 (0.044)	0.712 (0.033)	0.696 (0.056)	0.670 (0.044)
BUS	80%	0.774 (0.028)	0.767 (0.028)	0.705 (0.028)	0.785 (0.022)	0.699 (0.034)	0.771 (0.022)	0.759 (0.027)	0.670 (0.034)	0.446 (0.045)	0.683 (0.063)	0.699 (0.057)	0.691 (0.073)	0.763 (0.032)
	50%	0.769 (0.035)	0.762 (0.037)	0.720 (0.034)	0.784 (0.037)	0.633 (0.022)	0.729 (0.036)	0.759 (0.049)	0.600 (0.091)	0.374 (0.040)	0.603 (0.098)	0.705 (0.052)	0.723 (0.040)	0.639 (0.023)
	5%	0.663 (0.038)	0.625 (0.050)	0.517 (0.115)	0.666 (0.039)	0.469 (0.113)	0.592 (0.102)	0.628 (0.039)	0.570 (0.074)	0.462 (0.077)	0.606 (0.082)	0.637 (0.069)	0.556 (0.086)	0.550 (0.037)
FUS	80%	0.972 (0.003)	0.970 (.003)	0.951 (0.011)	0.971 (0.002)	0.957 (0.004)	0.956 (.008)	0.971 (0.003)	0.940 (.005)	0.708 (0.029)	0.939 (0.004)	0.945 (0.007)	0.954 (0.003)	0.961 (0.005)
	50%	0.971 (0.003)	0.968 (0.003)	0.959 (0.005)	0.970 (0.003)	0.959 (0.003)	0.960 (0.004)	0.956 (0.005)	0.920 (0.003)	0.675 (0.036)	0.929 (0.007)	0.946 (0.003)	0.953 (0.001)	0.954 (0.007)
	5%	0.955 (0.009)	0.954 (0.009)	0.930 (0.004)	0.952 (0.010)	0.940 (0.009)	0.945 (0.008)	0.955 (0.007)	0.930 (0.011)	0.670 (0.028)	0.931 (0.004)	0.936 (0.007)	0.940 (0.005)	0.806 (0.017)
X- ray	80%	0.979 (0.000)	0.98 (0.001)	0.977 (0.001)	0.979 (0.000)	0.973 (0.001)	0.977 (0.001)	0.978 (0.001)	0.960 (0.001)	0.926 (0.003)	0.960 (0.002)	0.944 (0.002)	0.948 (0.001)	0.976 (0.002)
	50%	0.978 (0.003)	0.977 (0.002)	0.974 (0.001)	0.974 (0.002)	0.955 (0.003)	0.961 (0.001)	0.972 (0.002)	0.910 (0.018)	0.876 (0.013)	0.934 (0.004)	0.954 (0.004)	0.968 (0.001)	0.976 (0.001)
	5%	0.973 (0.001)	0.971 (0.004)	0.968 (0.002)	0.973 (0.003)	0.962 (0.003)	0.960 (0.013)	0.972 (0.002)	0.940 (0.008)	0.897 (0.013)	0.950 (0.012)	0.922 (0.027)	0.946 (0.005)	0.959 (0.010)

C. Fine-tuning only parts of the network

In scheme 1, training the shallow path and freezing the deep path led to better results compared to freezing the shallow path and fine-tuning the deep path for all datasets (Fig. 4). Fig. 5 represents some examples of the segmentation results on the test set of BUS and SUS dataset. It is noteworthy that the number of parameters in the shallow path is less than half of the number of parameters in the deep path (Table II), but still we achieved better results by training fewer number of parameters. Similar results were observed when fewer images were used, and the difference between these two scenarios became even more evident (Fig. 4). The difference was statistically significant for all dataset sizes ($p < 0.05$, Tukey corrected), except for the SUS dataset.

As explained before and seen in Fig.1, we further divided the contracting path into two parts (blocks (1,2) vs. blocks (3,4)). None of these two parts were consistently better than the other (Table II), and fine-tuning the whole contracting path was significantly better than fine-tuning either of these two parts ($p < 0.05$).

The well-known approach to transfer learning in computer vision classification applications, is fine-tuning the last layers. This approach, however, failed in this study. Fine-tuning the last block of the network (block 9) was the worst strategy among all studied strategies, in datasets BUS, FUS and X-ray, and among the three worst strategies in the SUS dataset (Table II). Fine-tuning the first block (block 1) did not prove a good choice, either. However, training block 1 was significantly better than training block 9 in BUS, FUS and X-ray datasets, despite having approximately one fifth the number of parameters.

In scheme 2, similar to scheme 1, fine-tuning the shallow half (all except block 5) outperformed fine-tuning the deep half (block 5). The difference between the two scenarios was statistically significant for all pairs. Fig. 4 depicts the comparison of fine-tuning shallow and deep layers for all fractions of data used in this study. Fig. 5 also shows some examples of BUS and the simulation dataset when shallow or deep layers are fine-tuned.

Continuing the exploration of scheme 2, fine-tuning only blocks (1,2,8,9) did not significantly differ from fine-tuning the equivalent deeper half (blocks 3-7) ($p > 0.05$, Fig. 4). Interestingly, by fine-tuning these 4 blocks of the network (with only 1 million parameters) we were able to achieve a large portion of the performance, and only a small improvement was obtained by training the rest of the network (Fig. 6).

By including both blocks 1 and 9, we observed a slight improvement in the results compared to training block 1 or 9 only. Although there was still a gap when compared to the best results, by only fine-tuning blocks 1 and 9, with only 0.006 of the total number of parameters, a substantial gain was obtained compared to the pre-trained network (Fig. S1).

When enough data is available, it may seem obvious that training the whole network is the best strategy, but in the case of small datasets, this may not be true. We analyzed the performance of the network when different numbers of images were used. With 5% of the data for training, the best segmentation result was obtained by fine-tuning all components except block 5. When using the whole dataset, the results of fine-tuning the whole network was not significantly different from fine-tuning either the contracting path or all except block 5. In addition, the time needed for the network to converge when only shallow layers are fine-tuned is much shorter than that of fine-tuning the whole network. Table III shows the average Dice score and time spent among folds when 80% of the data is used for training for the best three strategies (on a NVIDIA TITAN V GPU). The difference in Dice score is less evident in the FUS and X-ray datasets, but the difference in the required time for training is noticeable.

In order to compare the results in different fine-tuning scenarios, we used random but fixed folds. However, the same pattern was observed when completely random folds were analysed in each experiment (results not shown). Although random folds were studied, the overall results were still showing the importance of fine-tuning shallow layers. We also visualized some example features extracted from a shallow and a deep layer for the pre-trained network and the fine-tuned network in the supplementary material (Fig. S2). Although it is not easy to interpret these features to explain the impact of fine-tuning, association of low-level features to shallow layers and high-level features to deep layers is evident.

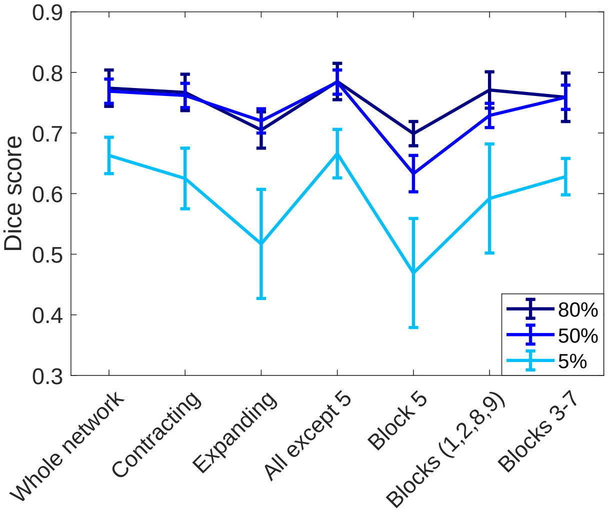

Fig. 4. Average Dice score for different sizes of training set, in different fine-tuning schemes. The figure depicts the example case of BUS dataset.

TABLE III
COMPARISON OF FINE-TUNING THE WHOLE NETWORK VS. THE CONTRACTING PATH (SCHEME 1) AND THE SHALLOW HALF (SCHEME 2) WHEN 80% OF THE DATA IS USED FOR TRAINING, IN TERMS OF AVERAGE DICE SCORE (STD) AND TIME REQUIRED FOR TRAINING EACH FOLD(MIN)

Dataset	Whole network		Contracting		All except block 5	
	Dice	Time	Dice	Time	Dice	Time
SUS	.834	1.36 (0.19)	.823	1.53 (0.20)	.826	**1.30** (0.18)
BUS	.774	1.65 (0.24)	.767	**1.39** (0.21)	.785	1.40 (0.21)
FUS	.972	3.20 (0.52)	.970	2.70 (0.41)	.971	**2.29** (0.40)
X-ray	.979	3.08 (0.50)	.980	2.84 (0.41)	.979	**2.53** (0.45)

D. The effect of size of the training set

Having access to large datasets is essential in deep learning, and can boost the models' performance. Our results followed the same rule: enhanced performance when using more data. The average Dice scores for different experiments are shown for the BUS dataset as an example (Fig. 4). A similar pattern was observed for other datasets as well. Higher values of Dice score were achieved when including all available 80% of the data for training compared to 50 and 5 percent of the data ($p < 0.05$, Tukey corrected). Likewise, better results were achieved when 50 % of the data was used rather than 5% of the data in all datasets and experiments ($p < 0.05$, Tukey corrected). As expected, higher variance in the network performance was observed when using 5 or 50 percent of available images compared to using all images, due to the small size of data used for training.

As expected, the pre-trained network did not work well for our target tasks without fine-tuning. Even by using only 5% of the data the segmentation performance improved substantially, compared to the pre-trained network (Fig. 7).

IV. DISCUSSION

We confirmed what was demonstrated in other studies [1], [2], namely, that fine-tuning pre-trained CNN models, in a

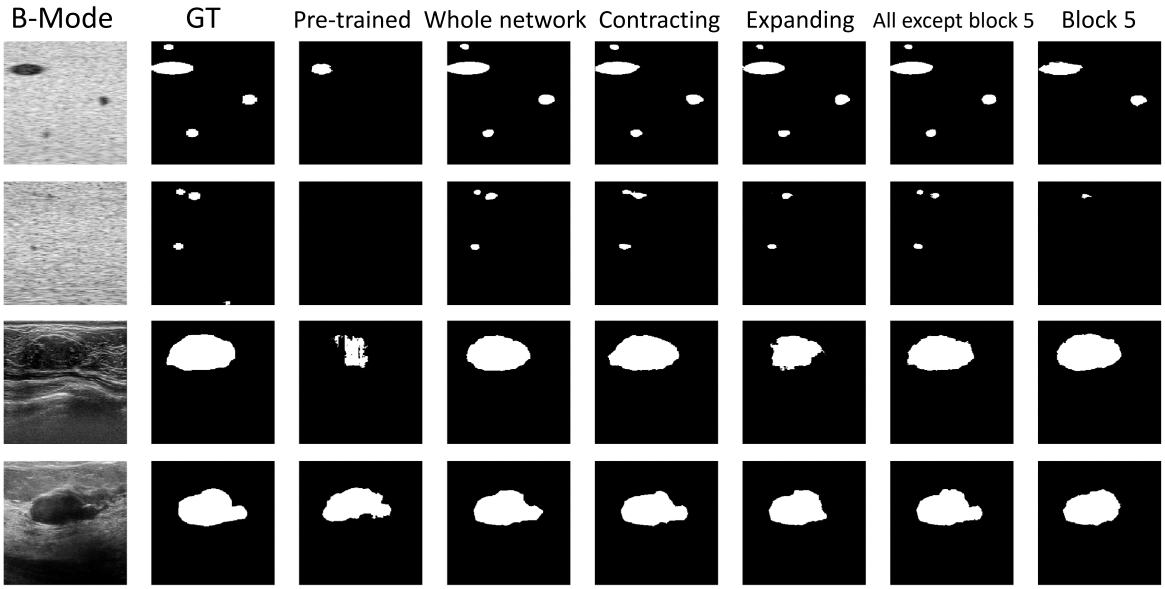

Fig. 5. Comparison of different fine-tuning scenarios on a few examples from the SUS dataset (the top two rows) and BUS dataset (the bottom two rows). The better performance of tuning the contracting path compared to the expanding path (scheme 1) and all blocks except block 5 compared to block 5 is evident. For FUS and X-ray datasets, Dice scores are close to one and the variations between different scenarios are not easily detectable. GT: Ground Truth

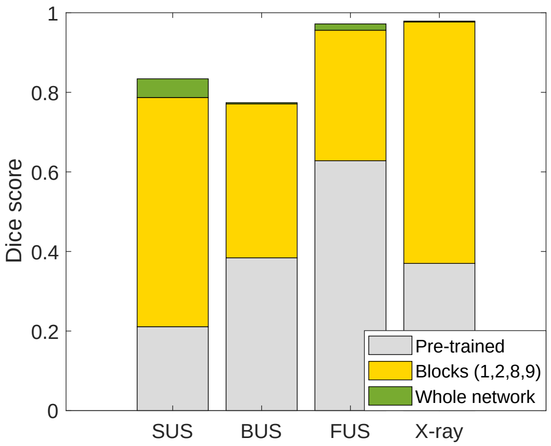

Fig. 6. The impact of fine-tuning the blocks 1,2,8,9, compared to fine-tuning the whole network. The gain added by fine-tuning the whole network (green portion) is not high in BUS, FUS and X-ray datasets. The figure depicts the example case of using 80% of the data for training.

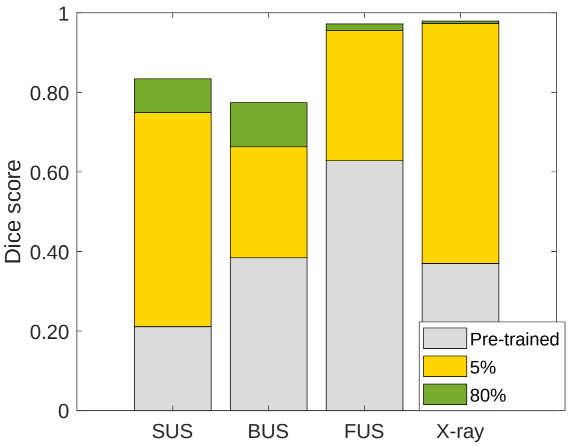

Fig. 7. The impact of using 5% of the available data, compared to the pre-trained network. The gain added by using 80% of the data for training (green portion) is not high in FUS and X-ray datasets. The figure depicts the example case of fine-tuning the whole network.

transfer learning fashion, is useful for medical image analysis and even outperforms training from scratch when limited training data is available. Although there are relatively large differences between natural and medical images, knowledge transfer is still possible from the natural domain to the medical domain. Using natural images as the source dataset is beneficial due to the larger size of the dataset compared to medical images. However, there are some studies favoring the use of medical images over natural images in transfer learning [35], [36], because of the greater similarity between the source and target datasets. Whether there would be any changes in our results by using medical images for pre-training the network warrants further investigation. We compared different fine-tuning scenarios on the same pre-trained network. A pre-trained network on a more similar dataset may expedite the training, but the overall superior performance of shallow layers observed in different fine-tuning schemes is likely independent of the pre-training dataset.

In this study, we used natural images to pre-train the network for segmentation of US images. Simulated US images could also be utilized for pre-training. However, as simulating a large number of US images is time-consuming, the number of available natural images is usually higher than simulated images. Natural images also have a larger variance compared to simulated US images, which is helpful in training the network. The network can therefore benefit from such larger datasets, and provide better segmentation results [37].

Training a network from scratch is not always feasible. For instance, in this study, the training process did not converge at all in some cases (especially when using fewer numbers of images for training). Although with a different initialization or set of hyperparameters, the network may be finally trained, the training procedure was fragile compared to using a pre-trained network. In addition, even when training from scratch succeeded, the results were not as good as those resulting from fine-tuning. Therefore, transfer learning is necessary when the number of training images is small.

We demonstrated that in US image segmentation using U-Net, in schemes 1 and 2, fine-tuning shallow layers of a pre-trained network outperforms fine-tuning deep layers, particularly, when a small number of images are available. We studied 3 different US datasets and one X-ray dataset. In this study we mainly focused on US images. Although we observed similar but less evident results for the X-ray dataset, we cannot generalize the conclusion to X-ray or other imaging modalities without further investigation and inclusion of more datasets. To clarify the effect of speckles and US-specific characteristics as well as properties specific to other imaging modalities in deep learning approaches, more investigation is necessary.

It is important to note that the U-Net is not a simple feedforward architecture. The notion of deep and shallow is ambiguous in a U-Net, because there are short and long paths from the input to the output. In this study, we considered two different definitions of shallow and deep layers. First, the depth of a layer was defined to be the longest possible path to reach it. Second, the depth of a layer was defined to be its distance from the first layer taking the skip connections into account. It is also interesting to note how the skip connections affect the performance, especially with respect to improvement when refining the shallow path in scheme 1. The skip connections provide long range connections, which have influence in the expanding path as well, so it is reasonable that refining the contracting path would outperform refinement of the expanding path.

Fine-tuning the expanding path was not as fast as the contracting path because of higher number of parameters and presence of the skip connections. We did not consider a fixed number of epochs for training in different scenarios, and eliminated the effect of convergence speed of different experiments by employing early-stopping in the training procedure.

The number of parameters involved in the fine-tuning procedure does not necessarily determine the performance. The depth of the fine-tuned layers plays an important role in improving the results. The bottleneck (block 5), for example, has almost half of the total number of parameters in the studied architecture, however, fine-tuning the bottleneck does not provide good results. In fact, fine-tuning all other layers except the bottleneck provides equivalent results to fine-tuning the whole network.

The main contribution of this work was to propose a way to improve the US segmentation when a small amount of data is available. However, our segmentation results are better than a recent article [12] published by the authors of the dataset. There are few reports available on segmentation of fetal head using the same dataset we used in this work [38], [39]. Our results are considerably superior compared to theirs (albeit, our training and validation procedure is not identical to theirs). Regarding the X-ray images, we could get Dice score of 0.979 for segmenting lungs, while a multiclass segmentation has led to 0.974 Dice score for lungs on the same dataset [40].

As depicted in Fig. 6, marginal improvement was obtained when more layers were included in fine-tuning. The progress in the results was much faster for the first few layers, and the slope of improvement declined when adding more layers. This observation is in line with [2] where including more layers resulted in very subtle improvement in segmentation results.

To avoid clouding the issue with confounding variables, we have intentionally used a reasonably well understood architecture and problem to demonstrate the effect of different layers. U-Net is one of the most popular architectures in image segmentation, and this study could be considered as a preliminary investigation; it would be interesting to explore other common architectures in future. It is also noteworthy to study other imaging modalities in more details to delineate their specific properties in the deep learning field.

One of the main limitations of this study is the number of datasets. To generalize the conclusions to other tasks (detection, classification, registration, etc) or other anatomical structures such as heart and vascular system, further investigation is necessary.

V. Conclusion

In US image segmentation with a U-Net, pre-training the network using natural images improves the results, particularly when limited data is available. The common practice of fine-tuning the last layers in transfer learning from one domain to another domain does not work in US image segmentation using U-Net. Moreover, fine-tuning shallow layers of a pre-trained network outperforms fine-tuning deep layers. This could be due to the presence of specific low-level patterns in medical images which are associated with shallow layers of the network. Based on our results, it is recommended to fine-tune shallow layers for small datasets. For large datasets, fine-tuning the whole network or shallow layers are not significantly different, even though fine-tuning the whole network is much slower. The specific U-Net architecture requires distinct transfer learning approaches.

Acknowledgment

The authors would like to thank anonymous reviewers for their constructive feedback, especially in regard to the two different schemes of considering depth in U-Net. This work was supported by in part by Natural Science and Engineering Research Council of Canada (NSERC) Discovery Grant RGPIN-2020-04612. The authors would like to thank NVIDIA for donation of the Titan V GPU.

References

[1] V. Cheplygina, “Cats or cat scans: Transfer learning from natural or medical image source data sets?” *Current Opinion in Biomedical Engineering*, vol. 9, pp. 21 – 27, 2019.

[2] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, “Convolutional neural networks for medical image analysis: Full training or fine tuning?” *IEEE transactions on medical imaging*, vol. 35, pp. 1299–1312, May 2016.

[3] A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle, “Knowledge transfer for melanoma screening with deep learning,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, April 2017, pp. 297–300.

[4] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *NIPS’14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, 2014.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, p. 436, May 2015.

[6] P. Looney, G. N. Stevenson, K. H. Nicolaides, W. Plasencia, M. Molloholli, S. Natsis, and S. L. Collins, “Fully automated, real-time 3d ultrasound segmentation to estimate first trimester placental volume using deep learning,” *JCI Insight*, vol. 3, no. 11, 6 2018.

[7] X. Yang, L. Yu, S. Li, X. Wang, N. Wang, J. Qin, D. Ni, and P.-A. Heng, “Towards automatic semantic segmentation in volumetric ultrasound,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI) 2017*. Springer International Publishing, 2017, pp. 711–719.

[8] Q. Huang, Y. Luo, and Q. Zhang, “Breast ultrasound image segmentation: a survey,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 3, pp. 493–507, Mar 2017.

[9] M. Xian, Y. Zhang, H. Cheng, F. Xu, B. Zhang, and J. Ding, “Automatic breast ultrasound image segmentation: A survey,” *Pattern Recognition*, vol. 79, pp. 340 – 355, 2018.

[10] B. Huynh, K. Drukker, and M. Giger, “Mo-de-207b-06: Computer-aided diagnosis of breast ultrasound images using transfer learning from deep convolutional neural networks,” *Medical Physics*, vol. 43, no. 6Part30, pp. 3705–3705, 2016.

[11] M. H. Yap, G. Pons, R. M. Marti, S. Ganau, M. Sentis, R. Zwiggelaar, A. K. Davison, and R. Marti, “Automated breast ultrasound lesions detection using convolutional neural networks,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, pp. 1218–1226, 2018.

[12] M. H. Yap, M. Goyal, F. M. Osman, R. Martí, E. Denton, A. Juette, and R. Zwiggelaar, “Breast ultrasound lesions recognition: end-to-end deep learning approaches.” *Journal of medical imaging*, vol. 6, p. 011007, 2018.

[13] 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 (MICCAI)*, 2015.

[14] A. Z. Alsinan, V. M. Patel, and I. Hacihaliloglu, “Automatic segmentation of bone surfaces from ultrasound using a filter-layer-guided CNN,” *International Journal of Computer Assisted Radiology and Surgery*, Mar 2019.

[15] M. Amiri, R. Brooks, B. Behboodi, and H. Rivaz, “Two-stage ultrasound image segmentation using u-net and test time augmentation,” *International journal of computer assisted radiology and surgery*, 2020.

[16] J. Yang, M. Faraji, and A. Basu, “Robust segmentation of arterial walls in intravascular ultrasound images using dual path U-Net,” *Ultrasonics*, vol. 96, pp. 24 – 33, 2019.

[17] N. Wang, C. Bian, Y. Wang, M. Xu, C. Qin, X. Yang, T. Wang, A. Li, D. Shen, and D. Ni, “Densely deep supervised networks with threshold loss for cancer detection in automated breast ultrasound,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Cham: Springer International Publishing, 2018, pp. 641–648.

[18] S. Leclerc, E. Smistad, J. Pedrosa, A. stvik, F. Cervenansky, F. Espinosa, T. Espeland, E. A. R. Berg, P. Jodoin, T. Grenier, C. Lartizien, J. Dhooge, L. Lovstakken, and O. Bernard, “Deep learning for segmentation using an open large-scale dataset in 2d echocardiography,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 9, pp. 2198–2210, 2019.

[19] P. F. Gu, W.-M. Lee, M. A. Roubidoux, J. Yuan, X. Wang, and P. L. Carson, “Automated 3d ultrasound image segmentation to aid breast cancer image interpretation.” *Ultrasonics*, vol. 65, pp. 51–8, 2016.

[20] J. Olivier and L. Paulhac, “3d ultrasound image segmentation: Interactive texture-based approaches,” *IntechOpen*, 2011.

[21] F. Milletari, N. Navab, and S. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 Fourth International Conference on 3D Vision*, 2016, pp. 565–571.

[22] S. Rueda, S. Fathima, C. L. Knight, M. Yaqub, A. T. Papageorghiou, B. Rahmatullah, A. Foi, M. Maggioni, A. Pepe, J. Tohka, R. V. Stebbing, J. E. McManigle, A. Ciurte, X. Bresson, M. B. Cuadra, C. Sun, G. V. Ponomarev, M. S. Gelfand, M. D. Kazanov, C. Wang, H. Chen, C. Peng, C. Hung, and J. A. Noble, “Evaluation and comparison of current fetal ultrasound image segmentation methods for biometric measurements: A grand challenge,” *IEEE Transactions on Medical Imaging*, vol. 33, no. 4, pp. 797–813, April 2014.

[23] W. Lu and J. Tan, “Detection of incomplete ellipse in images with strong noise by iterative randomized hough transform (irht),” *Pattern Recognition*, vol. 41, no. 4, pp. 1268 – 1279, 2008.

[24] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu, “Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree,” *IEEE Transactions on Medical Imaging*, vol. 27, no. 9, pp. 1342–1355, Sep. 2008.

[25] I. Zalud, S. Good, G. Carneiro, B. Georgescu, K. Aoki, L. Green, F. Shahrestani, and R. Okumura, “Fetal biometry: a comparison between experienced sonographers and automated measurements,” *The Journal of Maternal-Fetal & Neonatal Medicine*, vol. 22, no. 1, pp. 43–50, 2009.

[26] B. Kaur, P. Lemaître, R. Mehta, N. M. Sepahvand, D. Precup, D. Arnold, and T. Arbel, “Improving pathological structure segmentation via transfer learning across diseases,” in *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. Springer, 2019, pp. 90–98.

[27] M. Amiri, R. Brooks, and H. Rivaz, “Fine tuning u-net for ultrasound image segmentation: which layers?” in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2019.

[28] A. Veit, M. J. Wilber, and S. Belongie, “Residual networks behave like ensembles of relatively shallow networks,” in *Advances in neural information processing systems*, 2016, pp. 550–558.

[29] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, “What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4399–4407.

[30] J. A. Jensen, “Field: A program for simulating ultrasound systems,” in *Medical and Biological Engineering and Computing*, vol. 34, no. 1, 1996, pp. 351–353.

[31] 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, pp. 262–267, 1992.

[32] B. Behboodi and H. Rivaz, “Ultrasound segmentation using u-net: learning from simulated data and testing on real data,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 6628–6631.

[33] T. L. A. van den Heuvel, D. de Bruijn, C. L. de Korte, and B. v. Ginneken, “Automated measurement of fetal head circumference using 2d ultrasound images,” *PLOS ONE*, vol. 13, no. 8, pp. 1–20, 08 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0200412

[34] B. van Ginneken, M. Stegmann, and M. Loog, “Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database,” *Medical Image Analysis*, vol. 10, no. 1, pp. 19–40, 2006.

[35] H. Lei, T. Han, F. Zhou, Z. Yu, J. Qin, A. Elazab, and B. Lei, “A deeply supervised residual network for hep-2 cell classification via cross-modal transfer learning,” *Pattern Recognition*, vol. 79, pp. 290 – 302, 2018.

[36] K. C. Wong, T. Syeda-Mahmood, and M. Moradi, “Building medical image classifiers with very limited data using segmentation networks,” *Medical Image Analysis*, vol. 49, pp. 105 – 116, 2018.

[37] B. Behboodi, M. Amiri, R. Brooks, and H. Rivaz, “Breast lesion segmentation in ultrasound images with limited annotated data,” *ISBI*, 2020.

[38] V. Rajinikanth, N. Dey, R. Kumar, J. Panneerselvam, and N. S. M. Raja, “Fetal head periphery extraction from ultrasound image using jaya algorithm and chan-vese segmentation,” *Procedia Computer Science*, vol. 152, pp. 66 – 73, 2019, international Conference on Pervasive Computing Advances and Applications- PerCAA 2019.

[39] Z. Sobhaninia, S. Rafiei, A. Emami, N. Karimi, K. Najarian, S. Samavi, and S. Soroushmehr, “Fetal ultrasound image segmentation for measuring biometric parameters using multi-task deep learning,” 08 2019.

[40] A. Novikov, D. Major, D. Lenis, J. Hladuvka, M. Wimmer, and K. Bhler, “Fully convolutional architectures for multi-class segmentation in chest radiographs,” *IEEE Transactions on Medical Imaging*, vol. 37, 03 2018.