TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

Enhancement of clinical optoacoustic and ultrasound images

Authors	Sunita Gopal, Eva M. Höck, Fabian Pieroth, Tom H. Wollschläger
Mentors	Dr. Dominik Jüstel, Jan Kukacka MSc, (Institute for Biological and Medical Imaging, Helmholtz Zentrum München)
Co-Mentor	Michael Rauchensteiner MSc
Project lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics)

Abstract

Medical optoacoustic and ultrasound imaging are methods to generate images of tissue types and structures a few centimeters deep inside the tissue. In ultrasound imaging, acoustic ultrasonic waves are emitted and their reflection at different structures in the tissue is recorded. For the optoacoustic images, short laser pulses are shot at the tissue, which is warmed up by the light absorption. This in turn generates an acoustic sound wave which propagates back through the tissue and is recorded. From these recorded signals, the two-dimensional images can be reconstructed under different model assumptions, one of which is the model assumed for the speed of the acoustic wave in the various media it passes through (speed of sound model). Low quality images with a very simple, constant speed of sound model can be obtained in real time, while higher quality images with a more complex speed of sound model are more computationally intensive. In this project, we present a supervised learning approach to this problem, which is based on Convolutional Neural Networks. In the first part (Subproject 1), the task is to map low to high quality images, both reconstructed with the same speed of sound model. Here, the difference between input and target images is mostly noise. We trained a deep Convolutional-Transpose-Convolutional model with skip connections to be able to delete some of the noise while keeping most of the structures in the images. Furthermore, we propose three approaches to reduce the dimensionality of the optoacoustic data. In the second part (Subproject 2), the target images were reconstructed with a more complex, dual speed of sound model. So, additionally to the denoising, the task was also to learn the translation and deformation due to different speeds of sound. For this, attention masks are employed, which allow to break up the parameter sharing in convolutional layers. In the original task of Subproject 2, the speeds of sound are arbitrary, which poses a hard problem for a convolutional architecture. Here, we were able to learn the deformation but were not able to keep the details. In a relaxed version of the task, the model is presented with two input images, each one being reconstructed with one of the two speeds of sound assumed for the reconstruction of the target image. In this setting we showed that our model is able to learn the deformation while increasing the image quality. The results show that the deformations are highly dependent on the depth of the image and that the parameter sharing is preventing the network to learn locally different deformations. We further show that providing the network with two input images with the corresponding speeds of sound of the target significantly improves the quality of the prediction.

Contents

List of Figures

1 Introduction

Ultrasound and optoacoustic imaging is being developed and used in the medical field to generate images of tissue and structures a few centimeters below the skin. The clinical handheld Multispectral Optoacoustic Tomography (MSOT) Acuity device of iThera Medical produces both of these image types simultaneously. For the ultrasound images, acoustic ultrasonic waves are emitted and their reflection at different structures in the tissue is recorded. For the optoacoustic images, short laser pulses in 28 different wavelengths are shot at the tissue, which is warmed up by the light absorption. This in turn generates an acoustic sound wave which propagates back through the tissue and is recorded. So the combination of these two methods provides information about the acoustic reflection properties (acoustic contrast) and the light absorption (optical contrast) of the tissue.
From these recorded signals, the actual two-dimensional images can be reconstructed under varying model assumptions. One of these model assumptions concerns the speed of sound model: Calculating the location in the tissue depends on the speed of the acoustic wave, which varies with the medium it propagates through. Examples for simple speed of sound models are the homogeneous model, in which one constant speed in all media is assumed, and the dual speed of sound model, which allows two different speeds, one inside the tissue and one inside the coupling medium in the ultrasound probe. On the iThera Medical MSOT Acuity, the reconstruction is done assuming a constant speed of sound model and other suboptimal model assumptions, leading to low quality images. There exist methods to obtain higher quality images assuming a more complex speed of sound model and more suitable model assumptions, however, these are to computationally complex for real-time applications.
In this project, we explored an supervised learning approach to this problem: Provided with a dataset of low quality input and high quality target images, the goal was to develop and train models to generate high quality output images from low quality input data. The overall learning problem is presented in Section 2. In Section 3, a more detailed discussion of the image formation process and the data used for this process is given, alongside with an outline of the background in deep learning in computer vision, optimization, data processing and augmentation.
The project is split into two parts:
In Subproject 1, the input and target images have been reconstructed using the same, homogeneous speed of sound model. Here, the difference in quality stems from a large amount of noise being present in the low quality images. In Section 4, we propose a Convolutional Neural Network model to solve this task for the ultrasound images. Section 5 regards the same task with optoacoustic images, which consist of 28 wavelength channels (like color channels in an RGB image). This high dimensionality results in computational complexity which we alleviate by compressing the data through various means before training the same base architecture we used for the ultrasound images.
In Subproject 2 (Section 6), additionally to the difference in noise level and image quality, the ultrasound input and target data has been reconstructed with different speed of sound models: the low quality input data with the simple, homogeneous model, and the high

quality target images with the more complex dual model. The difference in speed of sound model has the effect of the image pairs being translated and deformed version of each other. So the model not only has to denoise the input images but also deform them to produce output images that resemble the target data. In Section 6.1, only the speed of sound in the coupling medium is fixed for the target images, while the speed value in the homogeneous model and the speed value in the tissue of the dual model are considered to be arbitrary. In Section 6.2, the task is relaxed by providing the model with 2 input images having been reconstructed with the constant speed of sound that is assumed for the target images in the coupling medium and the tissue, respectively.
In Section 7, the results of the different subprojects are discussed before the conclusion and outlook is given in Section 8.

2 Problem Statement

For each of the different tasks in our project the data is generated in the following way. There is some signal $s \in \mathcal{S}$, where $\mathcal{S}$ is the signal space, which is retrieved from the detectors. Additionally there are two reconstruction algorithms $R_1, R_2 : \mathcal{S} \to \mathbb{R}^d$ and some abstract quality measure $d : \mathbb{R}^d \to [0, 1]$. The algorithms satisfy, that

$$d\left(R_1(s)\right) \geq d(R_2(s)).$$

So the quality of the first reconstruction algorithm is in general higher than of the second one. The idea is to find a mapping $N : \mathbb{R}^d \to \mathbb{R}^d$ and a suitable norm $||.||$ (e.g. the L2-norm) with $||N(R_2(s)) - R_1(s)||$ being small, while we achieve $d(N(R_2(s))) \approx d(R_1(s))$. Which means that we have similar images with almost the same quality as with the use of R_2. In the case of neural networks the norm is our loss function.

As the computational complexity of the reconstruction algorithm R_1 is too high, we need a good approximation of it. A simpler one, like R_2, looses unfortunately too much quality. So the use of classical means, like deducing an inverse mapping directly not only needs vast knowledge of the topic, but might pose an unsolvable problem. Nonetheless, the described problem is a classical task in computer vision, where convolutional neural networks have shown exceptional results in the past years [13]. They have shown the capability to extract the needed information directly from the visible structure of the images and reconstruct the images in a high detail [20]. Therefore it is a reasonable attempt to try to approximate the mapping N with a neural network.

Overall Approach

The dataset for our tasks is retrieved from a set of signals $S = \{s_1, .., s_n\} \subset \mathcal{S}$ through R_1 and R_2 to generate our input data $X = \{R_2(s_1), .., R_2(s_n)\} = \{x_1, .., x_n\}$ with corresponding labels $Y = \{R_1(s_1), .., R_1(s_n)\} = \{y_1, .., y_n\}$. The idea is to find a suitable mapping for all possible pairs of images that can occur with the reconstruction algorithms R_1 and R_2. As we can only draw a small sample from the space of signals $\mathcal{S}$ and do not know further details of the reconstruction algorithms as well as the true distribution of the signals, we have to take up to three kinds of errors into account. Which are optimization, estimation and approximation error [14].
The optimization error quantifies how good the optimization method that led to our hypothesis is relative to the optimal empirical risk minimizer. This is in case of neural networks given by the global minimum of the loss as a function of the weights. The classical optimization method of neural networks is stochastic gradient descent or some adjusted form of it [9], which is explained in more detail in 3.4. The success of this method depends heavily on the quantity of the data, as well as the infrastructure for the needed computations. The estimation error measures how well the hypothesis deduced from empirical data can generalize to the best neural network we could produce for this problem. The quantity of this error depends mostly on the quality and the overall distribution of the training data. The empirical data will consist of only a very small

sample of different patients as well as different body parts. To oppose these limitations we need to acquire more varied data over the time of the project, which is addressed in 3.5. Lastly the approximation error describes how good we could possibly be with using an explicit model. Even though neural networks are capable of learning complex non-linear relations there might be a non-trivial information loss from a given signal s when comparing the images generated by $R_1(s)$ and $R_2(s)$.

3 Literature Review and Background

In this section, we present the theoretical background of this project and introduce terminology and concepts that are used throughout this document. Firstly, the basics of image formation in ultrasound and optoacoustic imaging are outlined, with some details on the reconstructions used to obtain the data for this project. Also, the background in deep learning in computer vision is presented, before we explain the optimization techniques we used. After that, some data preprocessing and augmentation methods are introduced, to which we will refer when discussing the different subprojects in Sections 4 to 6. Lastly, the computational infrastructure that was used to train the models is mentioned.

3.1 Image formation in ultrasound and optoacoustic imaging

One task in medical imaging is to obtain information or images of objects underneath the surface of the skin, inside the tissue. When using optical means, the light is scattered too much inside the tissue and therefore can not be focused. Optoacoustic and ultrasound imaging provide a solution for this problem by relying on acoustic waves instead of light.
In ultrasound imaging, ultrasonic pulses are emitted from a transducer and travel through the tissue where they are reflected. These reflections travel back to the ultrasound probe and are recorded by detectors.
In optoacoustic imaging, short laser pulses of wavelengths in the visible or near infrared spectrum are shot at the tissue, where the photons are scattered at or absorbed by tissue molecules. How much of the light of a specific wavelength is absorbed is determined by the chromophore type of the molecule (an analogue concept to color). The absorption of photons gives the molecule more energy, which leads to thermalization, i.e. rising temperature, and pressure exerted onto the neighbouring tissue. This pressure generates an ultrasonic wave that propagates back through the tissue and is recorded by detectors. This creation of an acoustic wave from light is called the *photoacoustic effect* (see [2]).
In both imaging techniques, the acoustic signal detected has to be converted into an image of the tissue. This means going backwards in the processes described above, from the detected signal until the reflection (ultrasound) or thermalization (optoacoustic) in the tissue. This mathematical problem is called the *acoustic inverse problem*.
The speed and path of the ultrasonic wave propagating through the tissue is dependent on the speed of sound in the different media it travels through. Since it is impossible to model the speed of sound exactly, at every point on its way, one has to make the assumption of a simplified speed of sound model. The easiest is the *homogeneous speed of sound model*, in which we assume that the speed of sound is constant. A more suitable model is the *dual speed of sound model*, in which two different speeds are assumed, one in the coupling medium inside the probe (in our case heavy water), and a different, constant speed inside the tissue. See Figure 1 for a schematic depiction of the consequently differently reconstructed locations when assuming the homogeneous speed of sound model (straight path) and the dual speed of sound model (refracted path).

In Subproject 1 of this project, both input and target images were reconstructed with

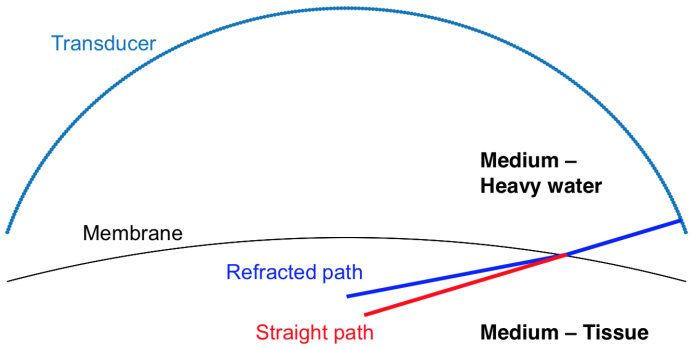

Figure 1: Refraction of the sound path due to the change of medium, figure taken from [8]

the homogeneous speed of sound model (see Sections 4 and 5), whereas in Subproject 2 (Section 6), the target images were reconstructed under the assumption of the dual speed of sound model. See Section 6.1.1 for a detailed problem statement of Subproject 2.

3.2 Specifics of the data used for this project

The data for this project was obtained using the clinical handheld Multispectral Optoacoustic Tomography (MSOT) Acuity device of iThera Medical, which generates ultrasound and optoacoustic data. For the optoacoustic scans, laser pulses of 28 different wavelengths are used, leading to images with 28 channels. For every pixel, these 28 channels represent the absorption spectrum at that point, that is dependent on the type of tissue. This spectrum can be decomposed into several base components, representing chromophore types like water, blood and fat (see Section 5). We received two differently reconstructed versions of the ultrasound and optoacoustic signals, which we will call *low quality* and *high quality* data.
For the ultrasound imaging, each of the 256 ultrasound transducers is transmitting in turn, while the others are recording the reflected signal. The low quality ultrasound images are reconstructed using the synthetic aperture method, which is done directly on the iThera MSOT. For the high quality ultrasound images on the other hand, each of the 256 images is reconstructed separately using backprojection, and they are combined to one via maximum intensity projection. See Figure 3.2 for an example of a pair of low and high quality ultrasound images.

The low quality optoacoustic images are reconstructed with filtered backprojection, which is a closed form inversion formula and has the drawbacks of some model assump-

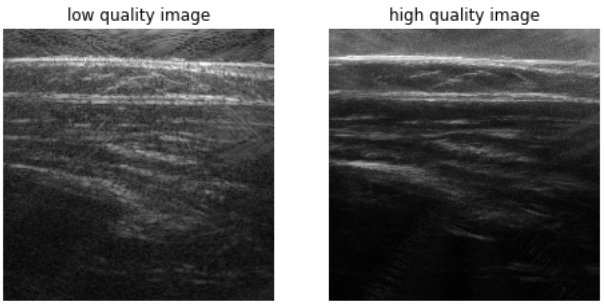

Figure 2: Example of a pair of low and high quality ultrasound images

tions that are not fulfilled, such as a complete continuous ring detector. For the high quality optoacoustic images, a more suitable forward model is used that for instance takes into account the discrete nature of the detector. According to this model, the images are obtained via least squares inversion with Tikhonov regularization.
For an overview of these and other reconstruction algorithms see [16]. We do not cover these reconstruction methods further because we do not model the differences in reconstruction with our models, rather we consider the task of generating high quality images from low quality input to be image-to-image-translation with a strong denoising component.

3.3 Deep learning in Computer Vision

Machine learning models suited for the scale of this task have in recent years had the form of Deep Learning, neural network models. Even more recently, since the breakthrough achieved with the AlexNet for image classification (see [10]), deep Convolutional Neural Networks (CNNs) have been the architecture of choice for these learning tasks in computer vision.
In convolutional neural networks, the convolutional layers are composed of convolutional kernels of predefined height and width with learnable parameters that is applied all to the input taking steps of predefined size stride in horizontal and vertical direction. For example, with stride 2, the height and width of the output feature map are half of those of the input. Such a convolutional layer produce a set of feature maps to which a non-linear activation function is applied (e.g. ReLu which sets all negative values to 0).
As described in Section 2, all of our tasks involve image to image translation and denoising of the image in some form. One popular neural network architecture for this task is the Convolution-Transpose-Convolution model: A series of convolutional layers (contracting path) with increasing number of feature maps of decreasing height and width is followed by a series of transpose convolutional layers (expanding path) that symmetrically increase heigth and width and decrease the number of feature maps.
This architecture has been used in [15] for image segmentation, with the difference of using upsampling instead of transpose convolution, and in [13] for image restoration. The

models in these papers also incorporate skip connections: Taking the output of some of the layers in the contracting path and combining them with their symmetric counterparts in the expanding path. In [13], this is done by simply adding the skip connections, in [15] by concatenating the skip connections to their counterpart and convolving.

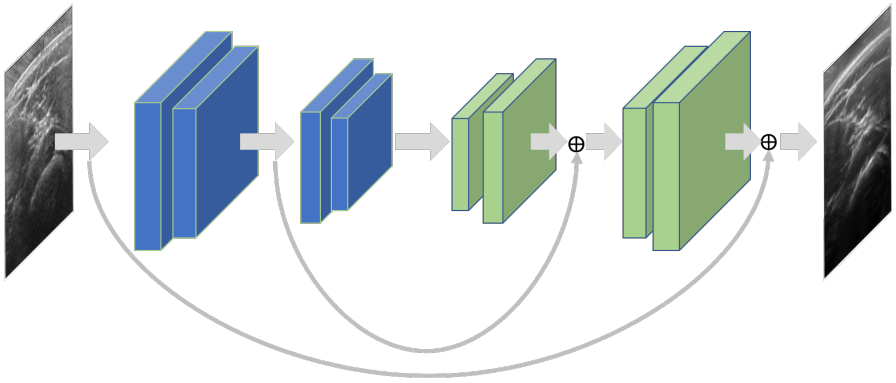

Figure 3: Schematic depiction of the base architecture

The models we used for the different tasks are all based on the Convolution-Transpose-Convolution model with skip connections, with adaptations according to the specific task. See Figure 3 for a schematic depiction.

3.4 Optimization

Loss function The loss function we used is the *Mean Squared Error* (MSE), calculated as the averaged squared Euclidean norm between the network prediction and the target. We also tried using the l_1 loss and a combination of the two without obtaining better results.

Optimization algorithm The optimization in neural network training is done via gradient based iterative minimization methods. For this, the gradient of the loss function with respect to the weights is obtained via the backpropagation algorithm which employs the chain rule of differentiation in order to backtrack the gradient through multiple layers of the network.
For the standard Gradient Descent algorithm, the gradient of the loss function has to be computed using all of the data sampled which is not feasible for the scale of model and data set we were confronted with. Instead, for variants of Stochastic Gradient Descent, the gradient is only computed for a randomly selected mini-batch of training samples. In particular, we used the Adam optimization algorithm (proposed in [9]), which uses exponential moving averages of the gradient and the squared gradient in order to take larger steps when recent gradients agree on the direction while also scaling the update parameterwise according to the estimated variance of each coordinate.

Learning rate schedule We used a one-cycle policy [19] during the training process. This means varying the learning rate during training, starting at a low one, going up to a high one for a specific number of epochs and then go back to the starting learning rate in the same number of epochs. Afterwards, we decrease the learning rate going to zero for a few additional epochs. Starting with a low learning rate is called warm-up phase and helps that the gradient descent algorithm goes into the right direction, minimizing the loss in the first iterations [4]. Then, due to the higher learning rates the algorithms might overstep really narrow and steep minima and can land in a more wide and flat minimum if we look at the space spanned by the weights of the network and the loss. The decrease of the learning rate helps the optimizer to step deep into that newly found wide minimum. For the same purpose, in the last epochs the optimizer is used with almost vanishing rates. Furthermore, [19] has shown that with this learning rate schedule faster convergence can be achieved.

Splitting the data set We split the data into train, validation and test data. The models are only trained with the train data. From the initial batch of data we were provided, we took 2 test samples to have test results to compare different models. Additionally, we were able to test the current models data from new batches we received. The validation data is separated from the train data on runtime and used to monitor the training and to produce the train and validation loss curve.

Regularization When training a Machine Learning model one has to keep another objective in mind: generalization, which means the performance on unseen data. During the training, the loss of the training samples is minmized which can lead to the phenomenon of overfitting, i.e. the model showing very good performance on the training data but not being able to produce good predictions on validation or test data. One approach to alleviate this problem is to artificially increase the number and diversity of training samples via data augmentation, see Section 3.5.
Another approach is altering the objective function used for optimization via adding an regularizing term like the squared l_2 norm of the weights W to the loss function L:

$$\tilde{L}(y, f(x, W)) = L(y, f(x, W)) + \frac{\lambda}{2}\|W\|_2^2$$

with x and y being the input and target data, f being the model and λ being the regularization parameter. This is a simple form of Tikhonov regularization. Through adding this regularization term to the objective function, increasing complexity of the model punished, which helps to avoid overfitting. Applying standard stochastic gradient descent to this altered objective function leads to the weight update having the form

$$W_{new} = W_{old} - \alpha \nabla_W L(y, f(x, W_{old})) - \lambda W_{old}$$

Altering the update of the optimizer in this form is the idea of *weight decay*. For other SGD variants like Adam weight decay and l_2 regularization are not equivalent. In [12], the authors propose AdamW, an algorithm to incorporate weight decay into Adam, that

has shown better generalization performance than l_2 regularization.

Batch normalization During Neural Network training, the distribution of inputs into the different layers will change because the weights are adjusted in each optimizing step. This phenomenon is called Internal Covariate Shift and can lead to instabilities, saturation of the nonlinear acitivation functions and the need for very fine tuning of the optimizer parameters as is described in [5]. As a solution, the authors propose *batch normalization*, in which additional learnable parameters are introduced, which normalize the data before each layer. This has the benefit of faster training and additional regularization.

3.5 Data Preprocessing and Augmentation

Data Preprocessing

In standard image processing, the images are often represented as arrays of floats in the range of 0 to 1 or integers between 0 and 255. The ranges of numbers in our data differ from this convention, see the histograms of examples for each image category in Figure 4.

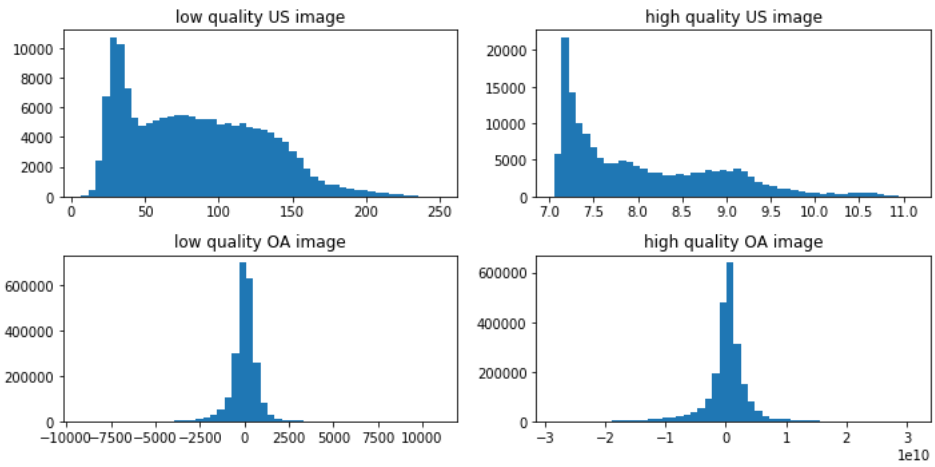

Figure 4: Histograms of raw ultrasound and optoacoustic images

One aspect is the ranges of input and output values differing quite a lot. This would create the additional challenge for a neural network to have to adjust the weights to bridge the divide between input and output before any real training can be done. Secondly, it is beneficial for the training to have data that is centered around 0 with variance of roughly 1. (See [11])
We solved these two problems by calculating the mean value and variance for the training data set (separately for input and target images) and scaling the data by subtracting the mean and dividing by the standard deviation. See the histograms of the scaled ul-

trasound examples in Figure 5

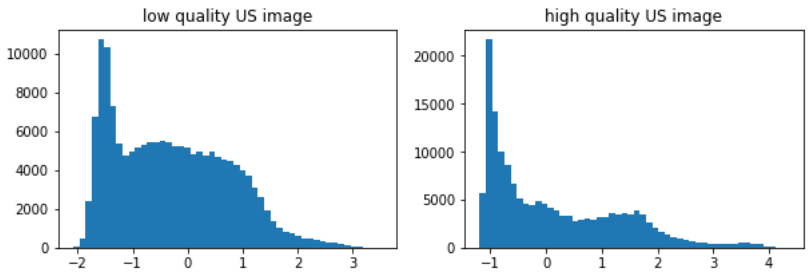

Figure 5: Histograms of scaled ultrasound images

Additionally, the optoacoustic data has severe outliers (most of all the low quality data) as can be seen in Figure 6, depicting the histogram with the vertical lines representing the 0.01 and 99.99 % quantiles. After consultation with experts on optoacoustic data we decided to truncate the data at these quantiles before the scaling because these values are likely to be erroneous and noisy.

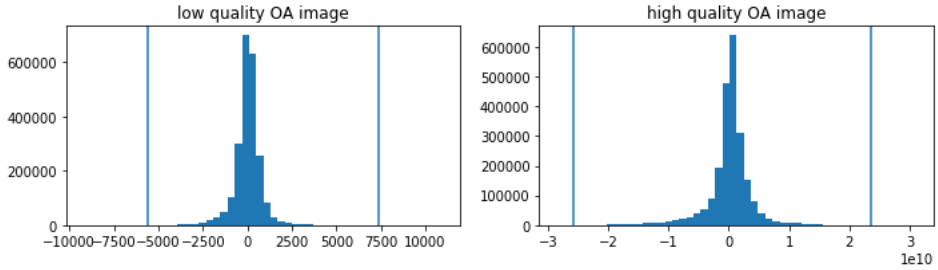

Figure 6: Histograms of raw optoacoustic images with 0.01 and 99.99 % quantiles

With these methods, we are not normalizing the data in a strict sense, which would involve scaling each pixel differently. This would enhance the importance of pixels with low variance, which in our images are of low importance.

Data Augmentations

In neural network training, good generalization performance is often dependent on a large number of training data samples. In the medical field, this is often hard to achieve because data collection is expensive. We also encountered this problem and had to train our models with about 100 base data samples each.
A way to artificially increase the number of training data samples without having to actually collect new data is *data augmentation*, which is most often done with image data. In data augmentation, the data pair is transformed via various techniques to produce a

"new" data pair. These techniques can for instance be flipping, translating or rotating the image.

We selected augmentation methods belonging to two groups: Methods that transform input and target image to simulate a new data pair (flipping, cropping and deforming) and methods that distort the input image to produce a data pair that is harder to learn (blurring and adding noise).

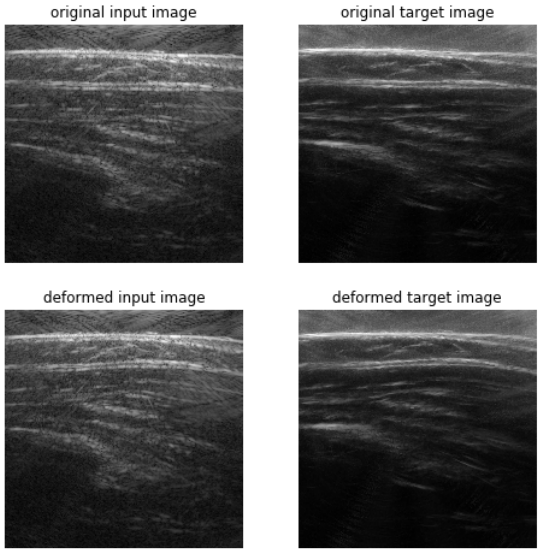

Figure 7: Original and deformed images

Flipping Both the input image and the target image are flipped horizontally (around their vertical axis). We did not flip the images vertically in order to preserve their geometric structure.

Elastic deformations We implemented elastic deformations similar to the approach described in [15]. In our version, a point in the interior of the image is chosen at random to be the deformation center, and random dislocation parameters are generated. Dislocation parameters of 0 are assigned to a few anchor points on the frame. The dislocations for the rest of the image are interpolated between the anchor points and the deformation center. See Figure 7 for an example.

Cropping The lower edge and either the left or the right edge are cropped by a few pixels. The cropping side and number of cropped pixels are chosen at random. The rest of the image is stretched to the original size.

Blurring The input image is blurred with a gaussian filter. This technique is supposed to simulate lower image quality in the input image and consequently forces the model to overcome a wider quality gap between input and target. See Figure 8 for an example.

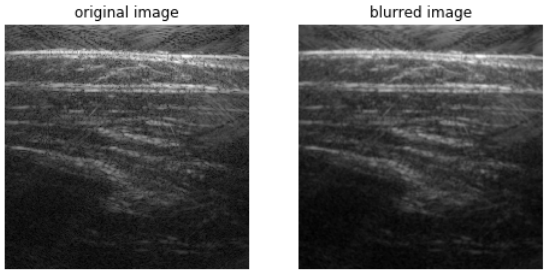

Figure 8: Original and blurred input images

Speckle noise Multiplicative Gaussian noise is applied to the input image. This

is a simple simulation of the speckle noise type that is typically seen in ultrasound images (see [18]). See Figure 9 for an example.

Adding more frames For some of the ultrasound data, we were provided with additional image frames that were reconstructed from signals captured shortly after one another, about 140ms apart. These can't be considered independent data samples because they are almost identical. That is why we treated the additional frames like augmented data and added them to the training data set.

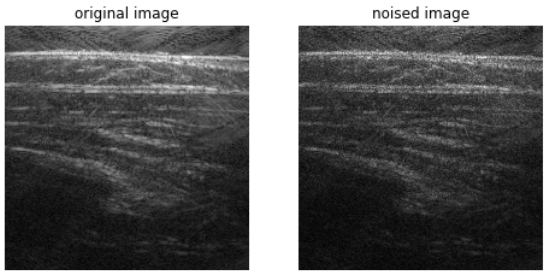

Figure 9: Original and noised input images

For each task and model we selected augmentations that are suitable for the specific problem. For Subproject 1 Ultrasound (see Section 4), we did an augmentation test series, testing the effect the different augmentations have for our result and discovered, that flipping, deforming and adding more frames have the biggest effect on decreasing the validation loss and getting the structure of the image, while blurring and speckle noise help to refine the image quality and produce less blurry predicted images.

3.6 Infrastructure

Training deep CNNs involves heavy computation that is best done on a general purpose graphics processing unit (GPGPU or GPU). Our code is written in Python with the models written in PyTorch, which allows fast computation on GPUs.
Early on in the project we reached the point of not being able to train or even test the models with our private machines. We are glad to have been able to use the infrastructure of the Leibniz-Rechenzentrum (LRZ), most importantly their GPUs. They consist of 4 single Nvidia P100 GPUs and 2 Nvidia DGX-1s with 8 GPUs each. Virtual Machines with access to this hardware can be booked for a certain time and can then be used to execute the training code.
We also used the LRZ Linux Cluster Storage and the LRZ Data Science Storage to store the data for the project and as mount devices for the GPU Servers.

4 Subproject 1: Ultrasound

The first task that was set for us is presented in this section. We start with a detailed problem statement, giving an outline of what our model has to be capable of. Afterwards some more insights and illustrations are added to this point with some examples from our data set. We follow up with the used methods, that were introduced in 3.5. After that we will present the used network architecture which will be the core of the following network architectures as well. Finally, some results are presented, showing the progress made as well as some thoughts about the remaining challenges.

4.1 Detailed problem statement

Our aim is to produce a deep learning solution which can generate high quality ultrasound images from corresponding low quality ultrasound images. The difference in quality of the input and target ultrasound images is due to the fact that they were obtained from different reconstructions of ultrasound signals (described in Section 3.2) For both images a homogeneous speed of sound model was used, using the same speed of sound for the reconstruction. That means, there are no deformations from low to high. The differences should consist of solely noise and some other artifacts such as reflections. Therefore, our model must perform a pixel wise mapping from reconstruction R_1 to reconstruction R_2, which denoises the image and deletes or creates artifacts depending on the target.

4.2 Data

For every measured ultrasound signal we obtain a low quality input image and corresponding high quality target image. Our data consists of 98 input and target ultrasound images each of dimensions 401x401. For training we use a train validation split of 90 to 10 percent of the total samples. Additionally, we set aside two samples as test data.

Data Exploration In Figure 10 in the top yellow outlined region, we see that in the low quality image the noise is more prominent compared to the high quality image. In the lower left region outlined in blue, one can see certain artifacts and noise in the input which are not present in the target. On the other hand in the green region in the upper left part, a clearly visible reflection artifact is present in the target, whereas it is not in the input. That shows the target image is not strictly improving in all aspects compared to the input. Nonetheless our network will try to map these artifacts correctly as well. As there are no visible information for this in the input, we expect that this will solely worsen the results.

4.3 Methods

Used Augmentation The number of total samples for such a complex task is small in comparison to similar tasks [13]. To enhance the generalization capacity and lower the estimation error (see Section 2), we perform the augmentations flipping, three different

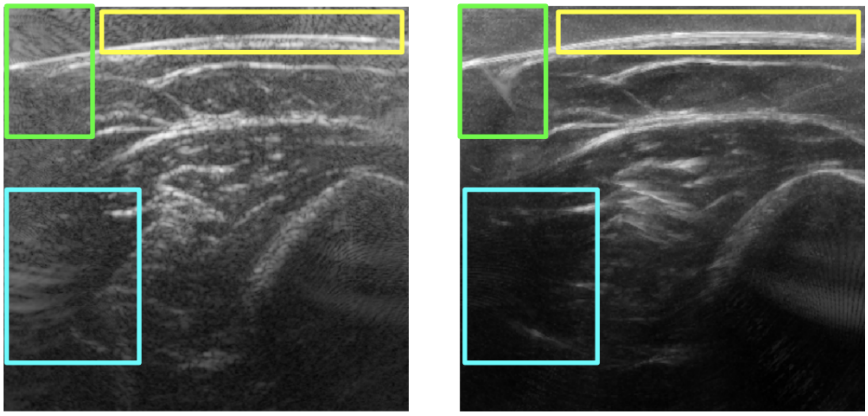

Figure 10: Difference between high and low quality images

elastic deformations, cropping, blurring and speckle noise (as described in Section 3.5). For each flipped image, we further perform the other augmentations mentioned on this image as well. Thus, we obtain an augmentation factor of 14 for each image present in the dataset. The validation set consists solely of original image pairs without any augmentations.

Scaling Since the range of numbers vary a lot in our dataset, we scale the training data roughly around zero mean and variance one as described in Section 3.5 and use the same scaling on the validation as well as the test set.

4.4 Model

The architecture used is a fully convolution encoder decoder with symmetrical skip connections as proposed by Mao et al [13]. Skip connections are added from the convolution layers to its corresponding deconvolution layer as shown in Figure11. The convolution layers capture the image features while the transpose convolution layers upsample these feature maps. Adding skip connections provides two benefits. They pass information of the convolutional feature maps to the corresponding transpose convolutional layers and as we are adding the skip connections, we are essentially modeling the error term. As we also have a skip connection from the input directly to the output of the network, our model is learning what needs to be deleted or added instead of completely reconstructing the image. Furthermore, skip connections tackle the problem of vanishing gradients as the gradient propagates back directly over the skip connection to the bottom layers.

The model we use consists of 7 convolution layers with (64,128,128,256,256,512,512) channels and 7 transpose convolution layers (with reversed order of channels) for upsampling with skip connections. All the kernel sizes of convolutions have been set to be (7,7) with stride (2,2). We also used padding of (2,2) and output padding of (1,1)

appropriately in order to ensure that we have same dimensions again for the feature maps. No pooling step was performed as these could discard useful image details.

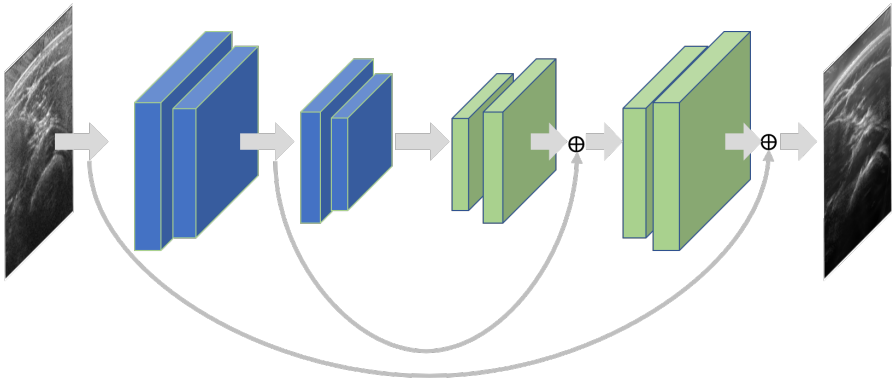

Figure 11: Architecture of proposed network

4.5 Result

We trained the network described on the P100 for 125 epochs with a batch size of 16. The loss function used was the Mean Squared Error (MSE) obtained pixel by pixel such that the same locations are always compared. The optimizer was Adam. We used a base learning rate of 1e-04 with a learning rate scheduler as described in Section 3.4.

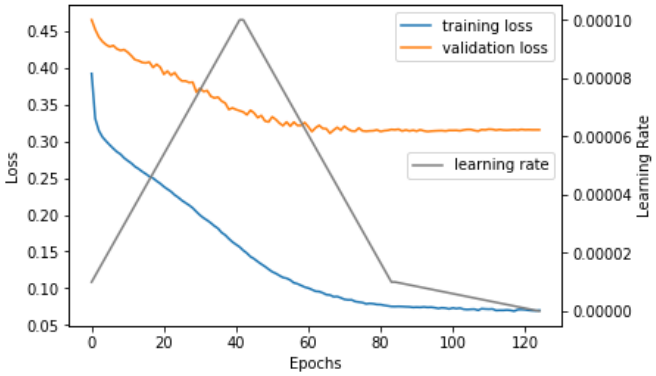

Figure 12: Training and Validation loss of SP 1 Ultrasound

The training and validation loss curve is shown in Figure 12. We see that the train loss decreases significantly and is quite low. In the training image in Figure 13, we see the predicted image is quite close in quality to that of the target. This shows that our model is capable of extracting fine features in the data.

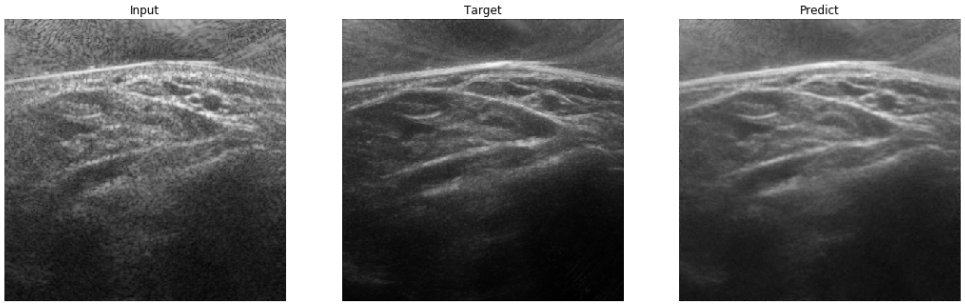

Figure 13: Training Sample: Input, target and predicted ultrasound images

However, we see the validation curve is higher and we are actually overfitting on the train image.Hence we can see in the test results in Figure 14 the predicted images are not as good in quality as the target and have deficits in producing fine lines. In the results we see that our network is capable of predicting all the important features and deleting most of the noise and is of much higher quality than the input images.

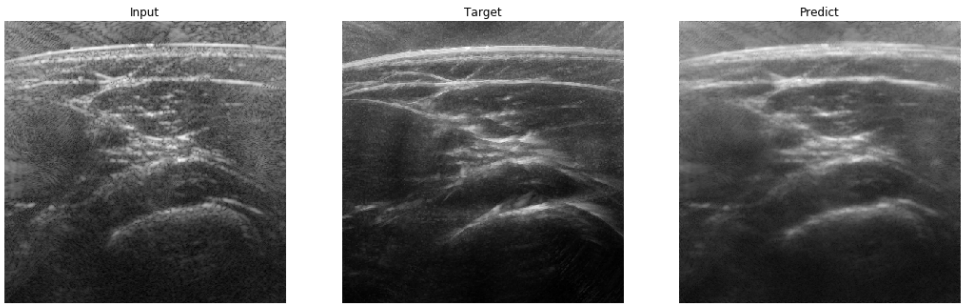

Figure 14: Test Sample: Input, target and predicted ultrasound images

5 Subproject 1: Optoacoustic

In this section the task concerning the optoacoustic part of the project is presented. At the beginning we will start with an in depth description of the task, followed by a detailed depiction of the used data. Afterwards, the methods that were used are presented and justified. This is then followed by the different chosen approaches to tackle this task, as well as the used network structure. The end of this chapter consists of the generated results for each of the approaches.

5.1 Detailed problem statement

The core problem that we have to solve in this subproject is similar to the one in 4. The retrieved signal at different wavelengths is processed by two reconstruction algorithms. For each pixel and wavelength we get the amount of absorption, which results, using 28 different wavelengths, in a data shape of (401, 401, 28). So, for every pixel we have an absorption spectrum, which has specific characteristics depending on the tissue type at this pixel. Therefore, instead of capturing the visual information that is displayed in the channels like in the ultrasound case, the relevant information, beside spatial dependencies, is also contained in the spectral information per pixel.
The first reconstruction algorithm can be used in production, as it produces single images in real time. Whereas the second one on the other hand takes about fifteen minutes, which is a substantial amount of time longer. Both use the same speed of sound for the reconstruction and therefore no deformations or shifts between pixels from one to the other are introduced. The differences are mostly due to several approximation errors in the faster reconstruction algorithm compared to the second. Our assumption is, that the offsets and fluctuations in the spectra can be seen as noise. Again we call the spectra reconstructed with the first algorithm low quality images and with the second one high quality images.

In Figure 15 one can see what our network has to be able to do. It takes the low

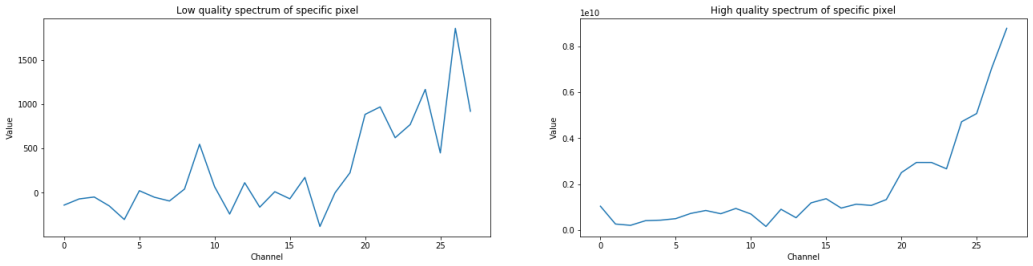

Figure 15: Spectra reconstructed from the same signal at the same pixel. The left shows the low whereas the right shows the high quality reconstruction.

quality spectral information and maps these on the high quality spectra. This problem is comparable to denoising high definition colored pictures and focus on keeping the color information. Neural networks have shown to be able to do that which can be seen

in [17]. Nevertheless instead of capturing the information given in three color channels, our case demands constructing information from 28 channels.

5.2 Data

One data sample consists of an image pair of low and high quality image. Whereas one image is represented as array of shape (401, 401, 28), which occupies saved in the pickle format about 36MB of storage. This results in about 72MB of storage for one data sample.
As the data size turned out to be a problem for computation time and capacity, we decided to restrict our studies to the upper part of the images, so the used data shape is (201, 401, 28). To test the capability of a network to denoise these kind of data, this is reasonable because most of the relevant information that can be captured, are in the upper part of the image and the number of artifacts increase with depth as well.
Over the course of the project the number of total data samples increased from originally 24 to 108 in the end. For the training process of our network we made a train-validation split of 90-10 percent of the samples and additionally took two more samples aside as 'test' set to have comparable images between models. This results in total in a split of 95 training, 11 validation and two test images.

5.3 Methods

Due to the strongly differing type of the optoacoustic data compared to the ultrasound, we need to consider which of the presented methods in 3.5 are sensible to use. This especially needed adjustments for the scaling as well as selecting only the suitable augmentations. Furthermore the evaluation of the results demanded new methods.

Used Scaling As described in 3.5 we want the data roughly centered around zero with variance being about one. The values operate on a very large scale with outliers skewing the mean or the variance of the data. Whereas we only had a single channel in Subproject 1 ultrasound, the important information is now contained in the spectra for each pixel. Therefore we are not only truncating the data to delete outliers, but are also using only single parameters for mean and variance for the whole training dataset.

Used Augmentations Augmentation methods like blur are motivated through experience of visual information in natural images. However, in this case we are not dealing with natural images but with a huge amount of absorption spectra. Therefore we only used augmentation methods that are sure to keep the spectral information as well as the pixel wise spatial information from input to target. Namely these were flipping and elastic deformations.

Evaluation and Visualization During the training process and for the network the used metric determines the quality of the results. It is known that for image translation using solely the metric to evaluate the model, one cannot be sure if the network learned to keep

the important information and therefore performs well or not. For this reason we used two ways to visualize the spectral information contained in input, target and prediction as further performance measure.

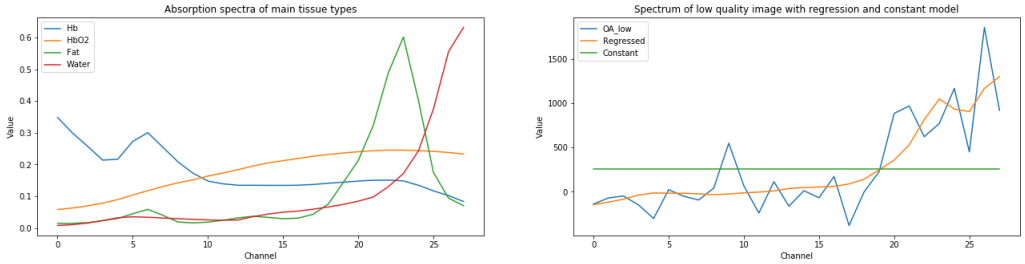

Figure 16: Left shows the base spectra of tissue types that are tested on. The right shows a low quality reconstructed spectra, with the result of the regression of the base spectra and that of a constant model.

One applicable use case of optoacoustic imaging is to display additional information of tissue types to the doctor during an ultrasound scan. We are going to focus on four possible types displayed in figure 16, namely hemoglobin (HB), oxygenated hemoglobin (HBO2), water and fat. For each of the pixels we perform a regression on the base spectra and get therefore four regression coefficients per pixel. Additionally we run an F-test on the null-hypothesis that the regression model is no better than any constant model (see [1] for details). If the resulting p-value lies below 0.05 we reject the hypothesis and the pixel is considered significant and will be taken into account for further visualizations. From this information we create the two visualizations mentioned before. We want to display and distinguish the main structures of different tissue types. Even though the absorption spectra of HB and HBO2 are different, they are displaying the same kind of structures, e.g. vessels. For this reason we add the two blood channels together. Afterwards we display each of the three resulting channels as a single color image, see figure 23. To combine this information into one image, we normalize each of the three channels and display them as an RGB image, which can be seen in figure 25. Insignificant pixels chosen through the F-Test and negative values are displayed as black in all of those plots.

Principal Component Analysis Due to the huge data size per sample we applied in one of our approaches the widely known principle component analysis (PCA) [7]. One essentially fits a subspace on the data such that projecting the data points onto it, the variance between the projected points is maximized. One wants to keep as much information about the structure of the original data as possible. Please note that in performing this, the data is centered around the origin. Which in our case means centering each channel individually. We fit the PCA on the target space, i.e. on the high quality images of our training data. As the low quality images operate with a different magnitude, we scale the data beforehand, as described above in 5.3. Afterwards we fit

the PCA on the scaled high quality training data, transform all input and target with it and before giving this data into our network we scale the transformed data again. Further details to this method is explained in the next subsection 5.4.

5.4 Approaches and used Model

The first tries to run models on the whole data quickly showed that this poses a harder problem than the denoising in the Subproject 1 ultrasound described in 4. To test our model capacity we began this task with trying to overfit on one sample. However, when using all 28 channels we did not even manage to overfit on one single sample properly. Therefore instead of using the whole data size, we focused on finding compression methods or taking parts of the data to extract and learn only the most significant features.

Sliced Channels The noise and the artifacts present in all of the 28 channels posed to be too much for our model to handle. As we are focusing on retrieving the information of the types of tissue per pixel, we decided to choose the most significant channels from the 28 to differentiate the different absorption spectra, shown in Figure 16. For this we chose the channels (1, 4, 7, 11, 16, 24, 28), such that the peaks and intersections between the different base spectra were kept, see Figure 17. By slicing the channels in

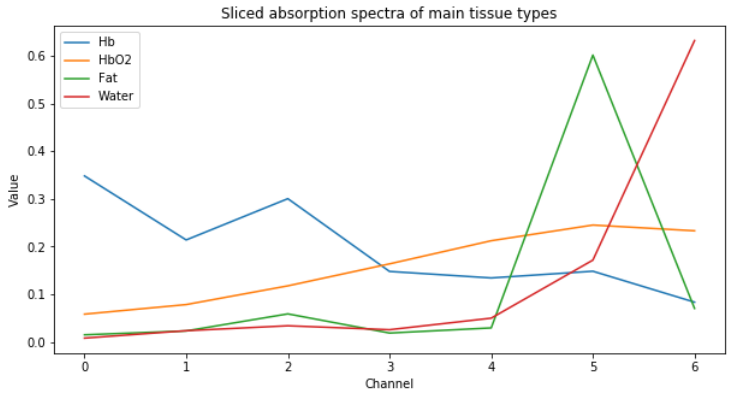

Figure 17: The base spectra with the chosen slice (1, 4, 7, 11, 16, 24, 28).

the described way, we lost a not to be neglected part of the information in one image, which can be seen as an example in Figure 18. However, with this we only need a fourth of the original data size and can still use the F-test to check for significance.

Regression Coefficients Whereas keeping as much information in the data as possible is very important, our evaluation methods and described goals, are focusing on differentiating between the different base tissue types. These are determined by the explained regression and the resulting regression coefficients. Even though a lot of information is lost performing the regression, one can assume that we are filtering out most of the noise with it. For our next approach we are therefore performing the regression on low and

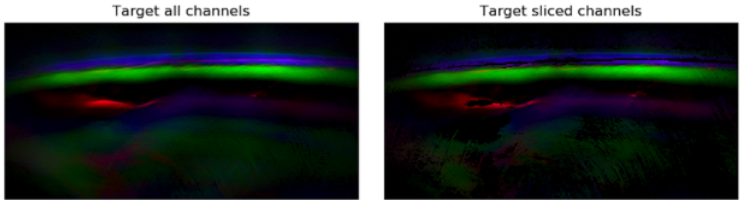

Figure 18: The left plot uses all 28 channels for the regression, whereas on the right only the seven sliced channels were used.

high quality images and use these as input and target of our network. Not only that we reduce the amount of channels from 28 to four, but by our assumption we filter out most of the noise, especially in the low quality images. This step should make it far easier for our network to extract the main features. However, we are losing the possibility to test for significance for each pixel. An example of the difference can be seen in Figure 19, where especially in the upper part the picture with significance test is clearly darker.

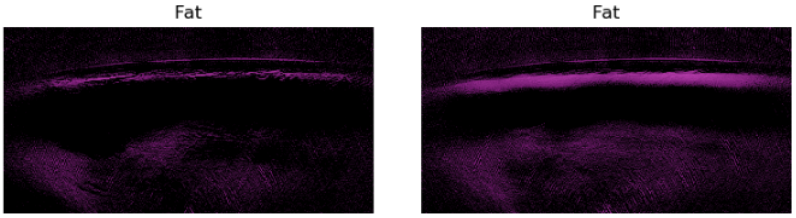

Figure 19: Left one can see the visualization of the fat regression coefficient of all 28 channels, with the test on significance. On the right the same image was used without the significance test.

Principal Components The first approach in using a slice of the possible channels unfortunately loses a big part of the information contained in the data. The second approach is highly promising for good results, as we are directly optimizing on the information to be displayed and are filtering out a lot of noise at the beginning. Unfortunately beside not being able to test on significance anymore, we are restricting ourselves to the main tissue types, given by our base spectra. So instead of taking just some slice of the data, we are trying to find a suitable subspace, keeping as much information as possible. The subspace spanned by the principal components of the high quality images, should be the space where the relevant information of the low quality images should lie. So fitting the PCA on the targets of the training samples and projecting the low quality images into this subspace, should delete most of the noise in the input and keep the relevant information to map to the target. The first four principle components already keep most of the variance in the high quality images, which can be seen in 20. However, we later want to test on a different subspace spanned by the base spectra and check for

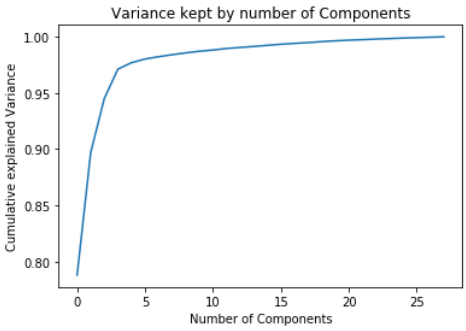

Figure 20: The preserved variance of high quality images by principal components.

significance. Therefore we decided to use the first seven principal components instead of only four, which are shown in Figure 21.

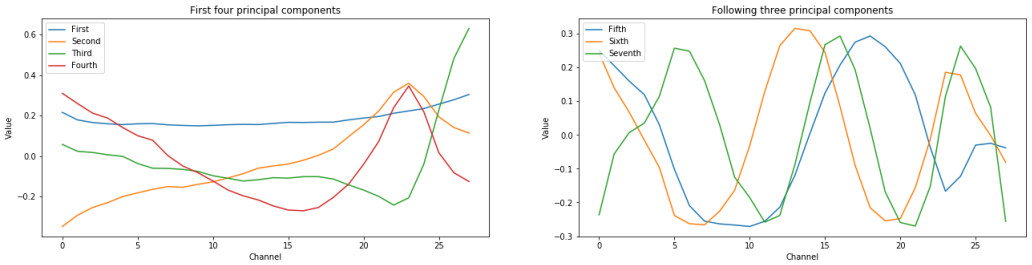

Figure 21: Left shows the four most important principal components and right the following three.

Used Model For all of the three described approaches, we essentially used the same model. It is a variant of the widely known U-Net with the changes described in 4.4. Again we use skip connections from the beginning to the end of the network, so it is essentially learning the error term between low and high quality. Furthermore we use five convolutional layers with (128, 256, 512, 1024, 2048) channels followed by five convolution transposed layers, with the reversed order of channels. All kernels are of size (7, 7) with stride (2, 2). Depending on the approach, we have four or seven input and output channels.

5.5 Results

For each of the three approaches, we trained a network with the previously described model for 300 epochs with a batch size of eight. All of the following approaches performed well on the training data, so we are going to focus on validation images as well as our fixed test set to easier compare the models among each other.

Sliced Channels The training process using the sliced channels as input shows a to be expected training and validation curve, see Figure 22.

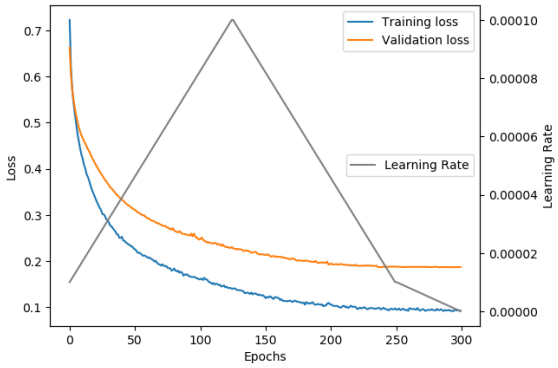

Figure 22: The training and validation loss of the sliced channel approach.

Both keep decreasing the entire training process, whereas the validation curve lies above the training. Taking a close look at some of the validation images, one can see, that we get visually closer to the target. Compare therefore Figure 23 and Figure 24. Especially the fat and total blood volume plots show a clear improvement from the input image. Whereas getting more significant information about the water spectra is not captured in Figure 24. On the other hand the target shows in all of the channels a more refined structure and sharper boundaries. The network can therefore not filter out all of the noise.

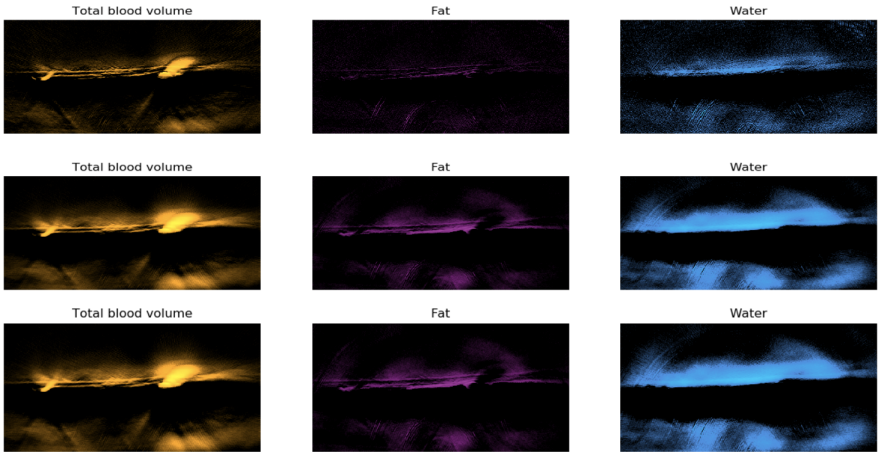

Figure 23: Sliced Channel Approach: From top to bottom we have the order Input, Target and Prediction of the single channel visualization of a validation image.

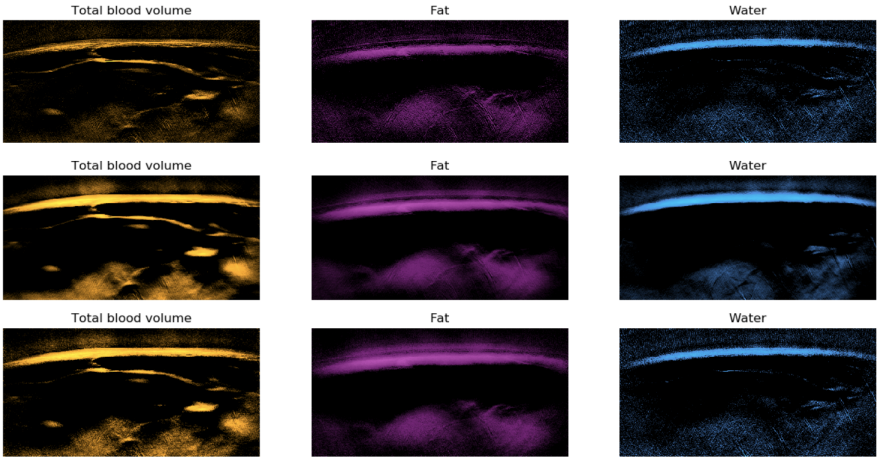

Figure 24: Sliced Channel Approach: From top to bottom we have the order Input, Target and Prediction of the single channel visualization of a validation image.

The same behavior can be observed in the RGB plot of one of the test images seen in Figure 25. In the upper part the water is hardly present in the predicted image, only about as much as in the Input.

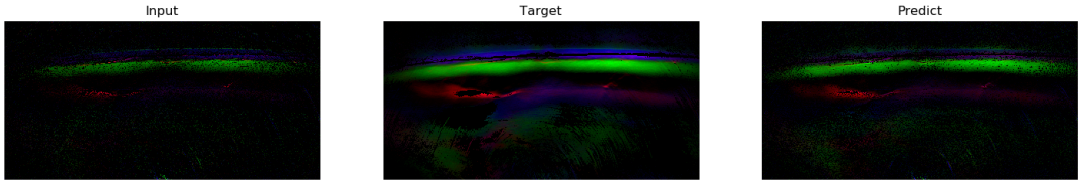

Figure 25: Sliced Channel Approach: The RGB visualization of a test image.

Furthermore in the clearly visible blood vessel located in the left middle part of the image, a hole is visible in the target, whereas it is not so much in the prediction. This is due to the target not being significant there anymore, whereas we get significant pixels in the prediction. In the constructed image, the network created something, that is not present in the target, or at least not significant. Nonetheless if one uses all 28 channels for this image, the remaining pixels of the blood vessel are all significant, again compare Figure 18.

Regression Coefficients Again the training and validation curves look as to be expected, see therefore Figure 26.

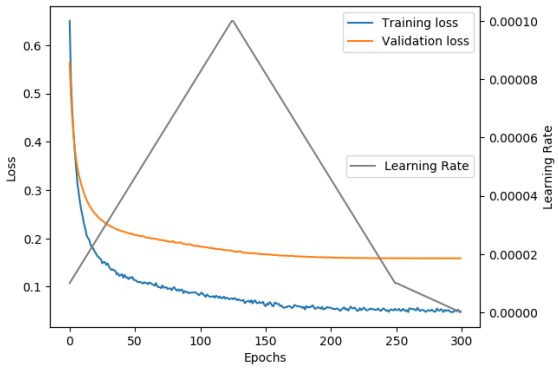

Figure 26: The training and validation loss of the regression coefficients approach.

The steep drop at the beginning hints at the learning rate being too high, which is undermined by the fact, that the validation curve stops decreasing at about 170 epochs. Now in all three plots one can see the improvement through the model, see Figure 27 and Figure 28. Even the water channel is visually very close to the target. Nevertheless the noise contained there is still visibly higher than in the blood volume or the fat channel.

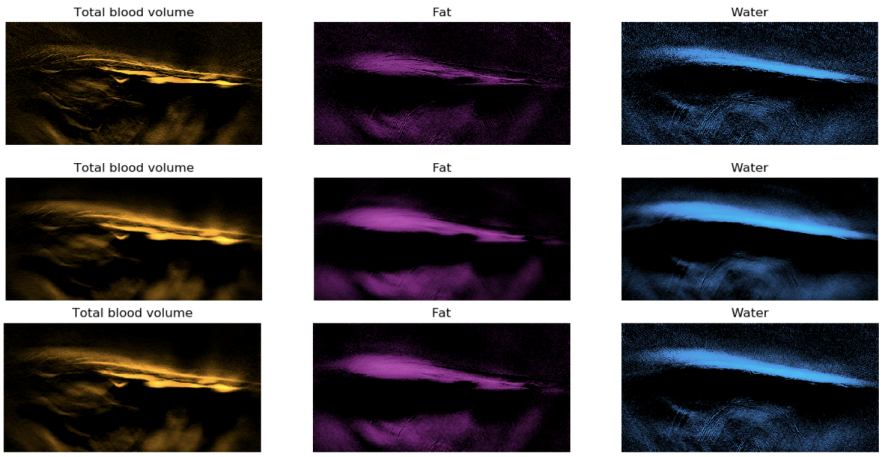

Figure 27: Regression Coefficients Approach: From top to bottom we have the order Input, Target and Prediction of the single channel visualization of a validation image.

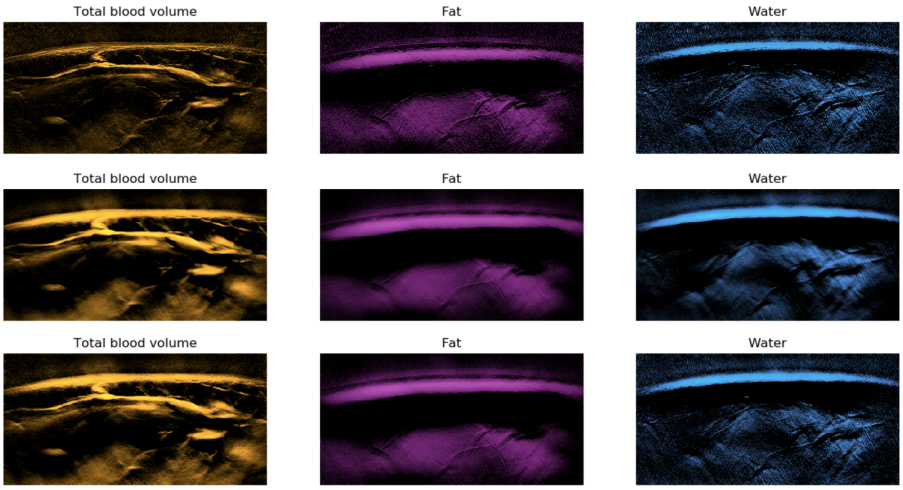

Figure 28: Regression Coefficients Approach: From top to bottom we have the order Input, Target and Prediction of the single channel visualization of a validation image.

In the RGB image, seen in Figure 29, overlaying the three channels, one can hardly see any differences at all. The brightness of the blue color, representing the water, is slightly lower in the prediction than in the target. Furthermore the structures themselves and their forms, are clearly defined in the prediction as well. In our testing and evaluation, we did not encounter an instance of our network creating something without it being there in the target.

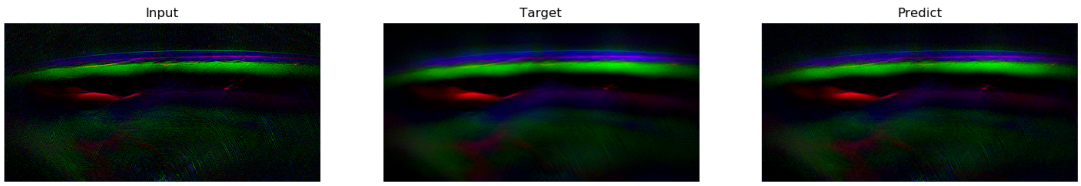

Figure 29: Regression Coefficients Approach: The RGB visualization of a test image.

Principal Component Analysis Compared to the other loss curves, it takes longer till the validation curve is no longer decreasing, but it is still converging. Also visible in Figure 30, is that the total loss in the end is lower than in the sliced channel case, which had the same number of channels in In- and Output. So we would expect the prediction to be closer to the target than in the sliced channel approach.

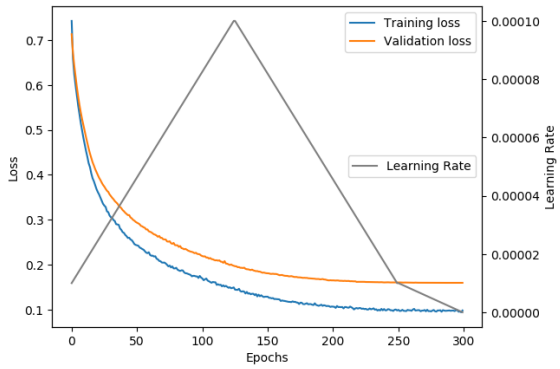

Figure 30: The training and validation loss of the principal components approach.

The enhancement of the network is clearly visible in Figure 31 and 32.

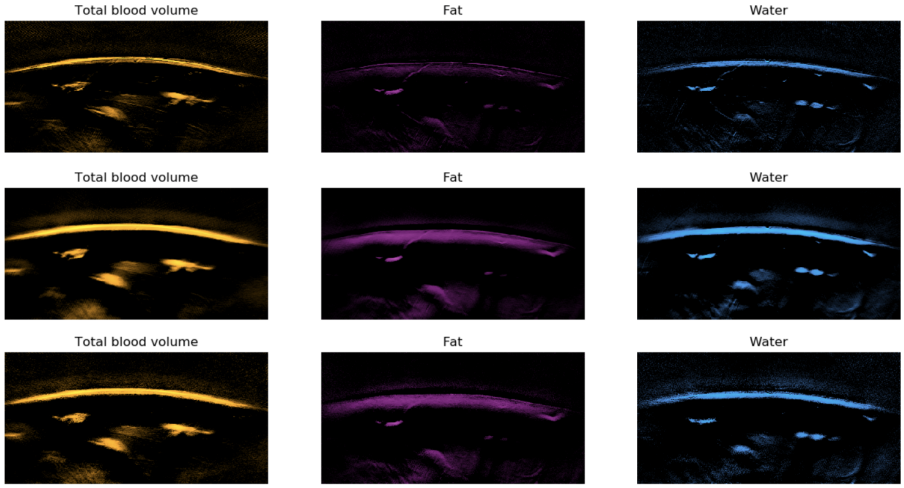

Figure 31: Principal Components Approach: From top to bottom we have the order Input, Target and Prediction of the single channel visualization of a validation image.

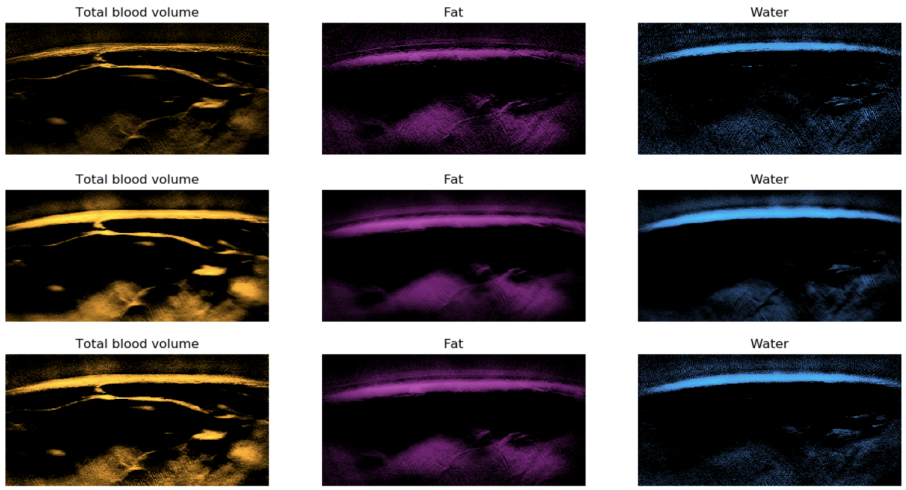

Figure 32: Principal Components Approach: From top to bottom we have the order Input, Target and Prediction of the single channel visualization of a validation image.

Nonetheless in all three of the channels the target image has brighter structures compared to the prediction. The difference in blood and water spectra is visually about the same magnitude, whereas the prediction of the fat channel is closer to the target. In the RGB images, here in Figure 33, one can only see minor differences.

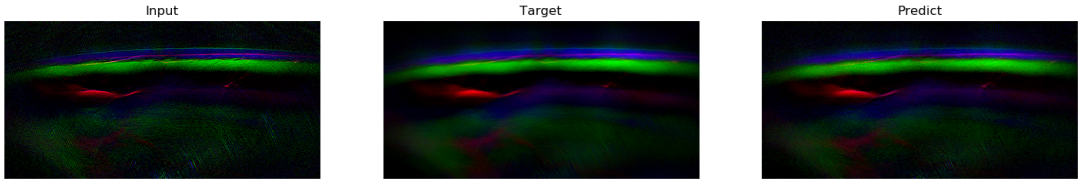

Figure 33: Principal Components Approach: The RGB visualization of a test image.

For example the target shows fat in the lower part of the image, which is smoother than in the prediction.

6 Subproject 2

6.1 Original SP2

6.1.1 Detailed problem statement

As discussed in Section 3.1, the solution to the acoustic inverse problem and hence the reconstructed ultrasound image depends on the speeds of sound along the way of the acoustic wave through the tissue and the ultrasound probe. Since it is impossible to know these speeds for each point, one has to assume a simplified speed of sound model. In this part of the project, additionally to denoising the image, the task is also to learn the translation and deformation between input and target images reconstructed with different speed of sound models.
The most basic speed of sound model is the homogeneous model, in which only one, constant speed is assumed. Under this assumption, the reconstruction is fast and simple, however suboptimal because the model is too simple. If we have a look at the area the acoustic wave has to propagate through in order to reach the detectors, we can divide it into two regions: the region inside the tissue, up until the membrane of the probe, and the region inside the probe. The tissue contains several main components like fat, muscle, water etc. But since water is the most prominent part, one approximate value for the speed of sound for tissue could be the one for water at around 37 degrees celsius which is roughly 1520 m/s, which can be estimated through the formula described in [3]. On the other hand the probe is filled with the coupling medium heavy water, which has an approximate speed of sound of 1397 m/s. Since in each of the two regions the speed of sound is approximately constant, but has a huge difference, it is reasonable differentiate between those two. This is captured in the dual speed of sound model, with one speed of sound value in the coupling medium (*coupling speed of sound*) and on value for the speed inside the tissue (*tissue speed of sound*).
In this part of the project we want our model to learn the mapping from low quality images reconstructed with the assumption of the simple, homogeneous speed os sound model to high quality images with the described dual speed of sound model. We can express our modeling task as learning the mapping:

$$p_1, c_{single}, c_{tissue}, c_{couplant} \mapsto p_2 \tag{1}$$

where p_1 is the reconstructed image with the acoustic homogeneous model, using c_{single} as speed of sound. p_2 is the image reconstructed with the dual speed of sound model with the tissue speed of sound c_{tissue} and the couplant speed of sound $c_{couplant}$.

6.1.2 Data

In this subproject we have an input image (low quality, c_{single}) and a target image (high quality, ($c_{couplant}$,c_{tissue})) for the training process. The low quality image is reconstructed with the reconstruction algorithm $R1$ which assumes a single speed of sound model. The speed of sound for low quality images in this subproject we call c_{single}. The high quality

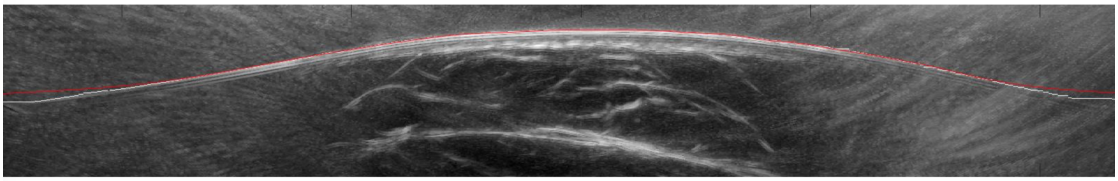

Figure 34: Detection of the surface with the homogeneous speed of sound model, figure taken from [8]

image is reconstructed with $R2$ which assumes a dual speed of sound model. The speeds of sound used are $c_{couplant}$ and c_{tissue}. From the recorded signal five images are reconstructed where c_{single} is drawn from a normal distribution with a mean of $1420\frac{m}{s}$ and a standard deviation of $30\frac{m}{s}$. Similarly, five target images are reconstructed from the signal but under the assumption of a heterogeneous speed of sound model. The speed of sound value $c_{couplant}$ is fixed and the values for c_{tissue} are drawn from a normal distribution with mean of $1520\frac{m}{s}$ and standard deviation of $30\frac{m}{s}$. Additionally we have one image representing the detected membrane of the measuring device which is detected from the homogeneous model with the speed of sound of the couplant as described in the previous section. The membrane is extracted through checking the gradient of the signal since the membrane results in the first strong increase in the values. An example of the detection can be seen in Figure 34 where the white line is the membrane and the red line is the extracted shape.

For every training pair we additionally need the speed of sound values used. These are encoded in an image. Thus, for c_{single} we have one image with the used speed of sound value in each pixel. To map to the target we need the speed of sounds and the location of the membrane to let our model learn this mapping. We have this information encoded in another image. In every pixel above the detected membrane we have the value $c_{couplant}$ and in each pixel below this membrane the value is c_{tissue}. An example of one data sample can be seen in Figure 35. The left image shows the input image, in the middle is the mask of the membrane with the corresponding speeds of sound and in the right we see the target image.

From every recorded signal we get five input images and five target images. We can use every input–target pair as training sample and get 25 combinations of input and target.

Additionally to the mapping from Section 4, the model now has to learn the deformation based on the speed of sound models. In Figure 36 we can see two images where the different reconstruction algorithms were used. In the left image we have the simple reconstruction while in the one on the right the deformation due to the refraction is accounted for. In general we can see the the structures of the target are slightly lower on in the vertical axis. We can observe that in the upper area the deformation is not

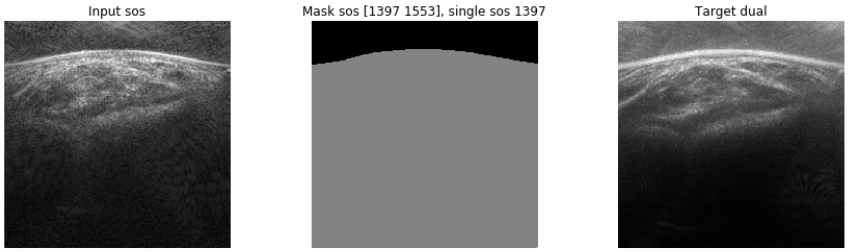

Figure 35: Training sample. Left: input image, middle: boundary mask, right: target image

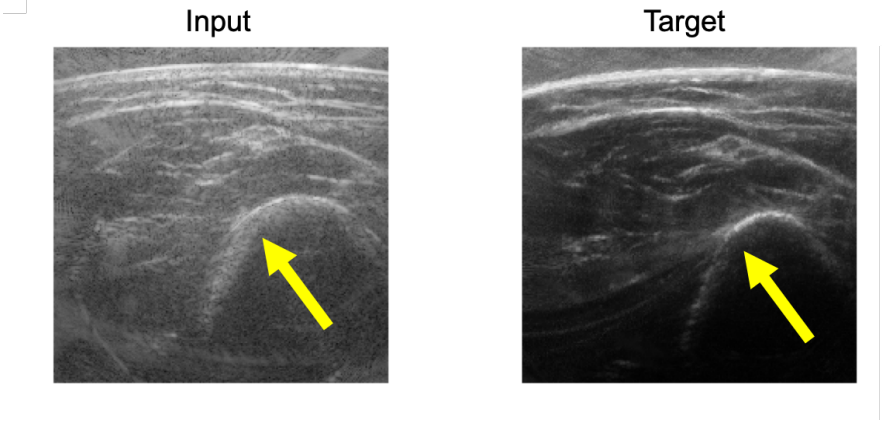

Figure 36: Images produced with different reconstruction algorithms. Left: input image $(R1, c_{single})$. Right: target image $(R2, (c_{couplant}, c_{tissue}))$.

as apparent as in the lower regions. The bone in the lower right of both images, on the other hand, was visibly more deformed. One can see that the curvature of this structure is stronger and the difference in location is greater than the differences in the upper areas. This is due to the different modelling processes of the wave propagation, the further the sound waves are propagated into the tissue, the higher the difference of the calculated paths.

In this subproject we also have the in Section 4 already described characteristics of noise and artifacts. Thus, the model has to learn the deformation on the noisy images while increasing the image quality.

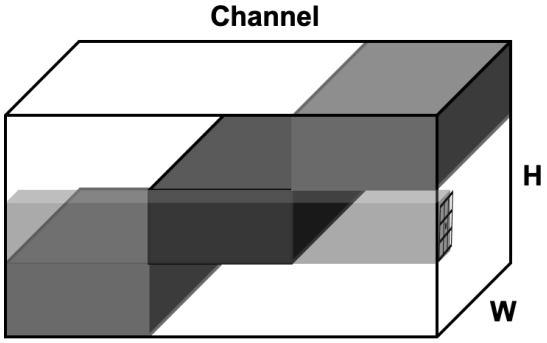

Figure 37: Schematic depiction of a convolution applied on the masked input image

6.1.3 Methods

The difficulty of this part of the project is training the neural network learn the denoising mapping as in Section 4 and additionally the deformation. As our target is already deformed and less noisy, we cannot decouple these tasks in the learning process. Therefore the noise in the input image hinders the learning process of the deformation mapping. The network has to extract the feature representation of the tissue and then use the information of the dual speed of sound mask to map the features onto the correct locations. These deformations could be learned by a fully-connected layer since they are linear and otherwise only depend on the refraction. Important factors are the distance traveled in the respective media and the neighbouring tissue information. However, in our case a fully-connected layer would have too many parameters since we would map from the vectorised input image to the vectorised target. Thus, we would map from 160,801 to 160,801 nodes which results in more than $1e10$ parameters which are too many to train. In literature these kind of deformations on images are often performed with spatial transformer networks [6]. In this framework the features of an image are extracted and localised. Then a transformation matrix is learned and then is applied to the original image. With our image content it could be hard to learn the deformation on a smaller convolved feature map or lower resolution since especially in the lower areas of the image, quite thin structures are present. These structures could prevent the network from learning the correct deformations in these areas since on the smaller feature maps the true local information could be lost already. Furthermore, in implementations of the network, the deformations matrix is also learned with several fully-connected layers which is still not quite feasible after two convolutions. Thus, the tradeoff between local information and number of parameters makes this approach not ideal for our task. Another approach called locally connected neural networks where there is no parameter sharing at all. So for each applied kernel we get a new set of weights. If we assume a kernel of size $7x7$ then this would already result in $7,879,249$ parameters for one layer. Since many deformations go further than 7 pixels in the vertical directions this kernel would still be too small. Thus, the high number of parameters also makes this approach

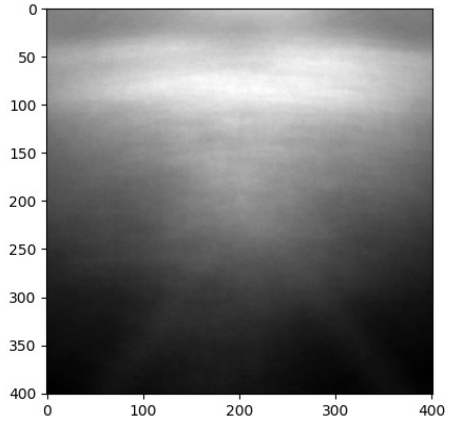

Figure 38: Mean image of training input images reconstructed with homogeneous speed of sound model

not feasible for the current problem.

To tackle the tradeoff between number of parameters in the network and quality of the learned deformation, we want to apply the same transformation over some parts of the image. Specifically, we want to use the same filter over the whole width of the image and up to a specific depth. To achieve this we duplicate the input image and stack the duplicates together in the channel dimension. Then we mask every channel to contain specific parts of the image and set the rest to zero. Thus, when we apply a convolution over this input we have an own parameter set per channel. As an example, if the convolution is at the location as shown in Figure 37, then the results would only depend on the convolution of the input image with the parameters of the kernel of channel two. All other channels are zero and not contributing to the result. If we now backpropagate the gradient, the parameters of the middle channel are only updated at the non-zero parts of the image since it did not contribute to the other areas. In this example case we could learn three different deformations which are applied over the whole width and one third of the height. After this convolution step we mask the resulting feature representation again but start overlapping the non-zero areas step by step to enable a smooth transition of the different deformations in our predicted image.

Based on the overlaid image of the training data, as shown in Figure 38, we chose four regions. The first one being the part of the image down to the membrane. In this area only the couplant medium was present and thus we have no important structures to learn. As second area we took the membrane since it is the transition of the media. Then we took the middle part of the image: starting below the membrane down to approximately 250 pixels. There most of the structures are present and we wanted to keep these in one slice. The rest of the image is in the last slice. Thus, in the beginning

we have four different parameter sets for the image to apply four different deformations. Throughout the network one can imagine the masking as having four groups of channels. In the beginning each contains a specific area of the image and with each convolutional layer this area is increased. These groups of channels do not have to be evenly distributed inside the network, which is why we also included a parameter how the distribution of the groups should be.

In this subproject we used only the flip, blur and speckle noise as augmentation techniques. Since we have to map a deformation already, introducing an additional deformation to the image would make it more difficult for our network to learn the original. Therefore, we only performed blurring and adding speckle noise on the original input as well as on the flipped version of the input resulting in an augmentation factor of six. Which results, with 100 original samples in training and six in validation, in a training set of 25*6*100 image pairs.

6.1.4 Model

The foundation of the model used for this project is the one introduced in Section 4. This fully convolutional encoder decoder framework with symmetric skip connections is the main part. However, this approach has drawbacks for the additional deformation task in this subproject. Forwarding and adding the feature maps with the symmetric skip connections onto the already upscaled transformed feature maps, introduces the original localizations of the non- or less deformed feature maps. Thus, instead of adding these together we stack the maps in the channel dimension to have the next transposed convolution fuse them. Thus, the transpose convolution kernels can learn to fuse the information of slightly different locations, depending on the kernel size. Special emphasis lies on the usage of the details of the first input image. The obstacle is that this image is never transformed and thus the difference due to the deformation between that image and the last feature map is greatest. Dealing with this issue we stack the duplicated and masked input image to the transformed feature map after the last transpose convolution and perform one additional convolution on to the result. We want to fuse the details of the forwarded map on the correct locations of the transformed one. We used five convolutional layer with (64, 128, 256, 512, 1024) channels, kernel size (7,7), stride (2,2) and padding (2,2). These layers are followed by the corresponding transpose convolutional counterparts to have the same image resolution as in the beginning. Additionally we used output padding of (1,1) at the third and sixth layer. In the end the last convolution had a kernel size of (7,7), stride of (1,1) and same padding to keep the resolution.

6.1.5 Result

The masks which are used for this run have the following fractions of the image starting from the top downwards: [0.125, 0.375, 0.375, 0.125] and cover the parts of the image as discussed previously. Furthermore, we used an exponential increase in the overlapping.

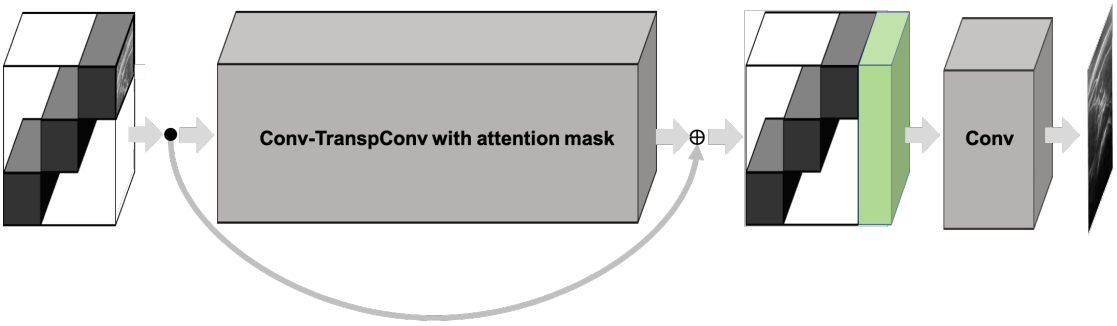

Figure 39: Model for Subproject 2. Base structure is the Conv-TranspConv model with attention masks. The input image and the transformed feature map are fused with another convolutional network

The overlapping ratio for the l'th layer is calculated as follows:

$$ratio = exp(l - N_{layer}) \tag{2}$$

Where N_{layer} denotes the number of consecutive convolutional layers. This ratio means that starting from the chosen slices from the beginning we expand those by the ratio of the remaining part of the image. This results in almost no overlapping in the first layer to complete overlapping after the last convolutional layer and thus before going into the first transpose convolution. To avoid overfitting we use weight regularization, see Section 3.4.

In Figure 40 we can see the loss curves. Furthermore, we can visually evaluate how our model performed on the test images in Figure 41. As we can see, the model has learned to apply the deformation. However, the quality of the predicted images is decreased compared to the results seen in Section 4.

There are many explanations on why we end up with these results. This task is more difficult than just the image translation from Subproject 1. Now the model additionally has to learn the deformation. Thus, some pixels are noise and have to be deleted while others are important structures and have to be shifted. Differentiating between those tasks is challenging for the model. The resulting image is very noisy and blurry. Losswise this result is preferably better since some pixels with high values are now in the same positions as the transformed ones.

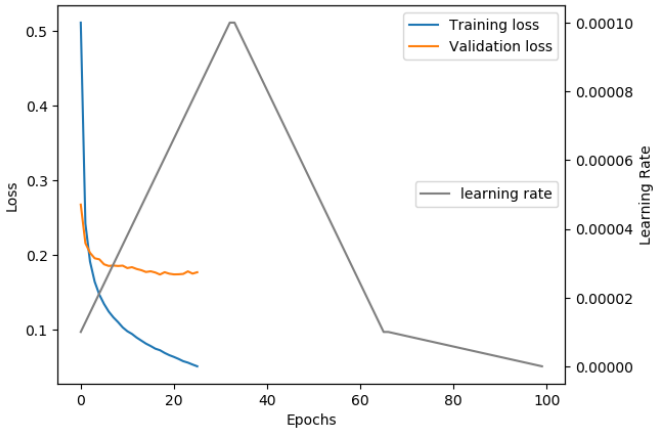

Figure 40: Trainings and validation loss of the run of Subproject 2 with varying learning rate. The left axis determining the values of the loss while the right axis shows the values of the learning rate for the respective epoch

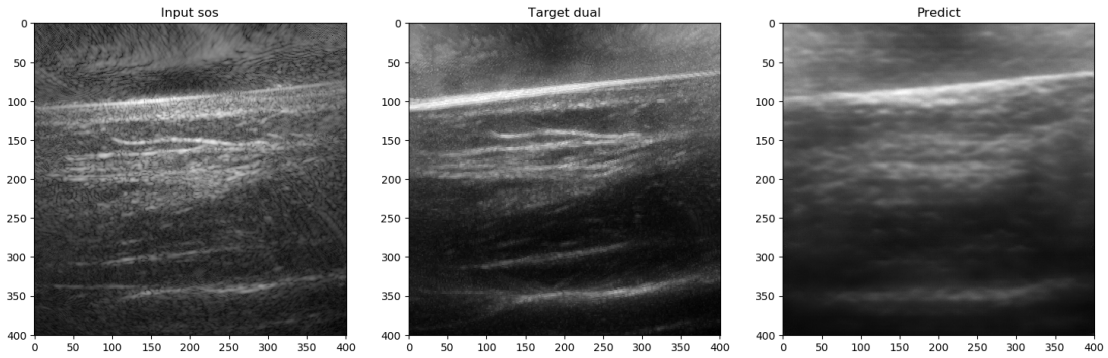

Figure 41: Results for the run. Left: input, middle: target, right: prediction

6.2 Relaxed SP2

6.2.1 Detailed problem statement

One could see in the previous subsection, that the learning a suitable mapping from a homogeneous to a dual speed of sound model is hard. As we could not achieve satisfying results with the original problem, we relaxed it, to possibly boost the performance of our network.
Before we had to learn the mapping of a reconstruction with some sampled single speed of sound model to one with a dual speed of sound forward model where the tissue speed of sound is sampled. To relax the problem we now use two input images reconstructed with R_1. The values for the reconstruction are the couplant and tissue speed of sound for the two input images respectively. With this input we can then let the network learn the mapping to the reconstruction with a dual speed of sound with the same values for couplant and tissue as in the input. The general applicability of the system however, is not decreased. While measuring with the iThera Medical MSOT Acuity device, the user can just set the values for the single speed of sound at will. Furthermore, for the extraction of the membrane the couplant speed of sound has to be used either way. Thus, the user only has to have another scan with the tissue speed and the necessary input data would be constructed.

The mapping that the network now has to learn can be expressed as:

$$p_{1,c_{couplant}}, p_{1,c_{tissue}}, c_{couplant}, c_{tissue} \mapsto p_2 \tag{3}$$

where $p_{1,c_{couplant}}$ is the image constructed with R_1 using couplant speed of sound, $p_{1,c_{tissue}}$ is the image reconstructed with R_1 using tissue speed of sound, $c_{couplant}$ is the couplant speed of sound, c_{tissue} is the tissue speed of sound and p_2 is the image reconstructed with R_2 and $c_{couplant}, c_{tissue}$.

6.2.2 Data

With the reformulated problem, we now have two single speed of sound images. One constructed with $c_{couplant}$ and one with c_{tissue}. For each of the single speed of sound values we have one image with the values encoded in each pixel. Furthermore, we have the dual speed of sound mask as described in section 6.1.2. These images are used as input data. The target is, as before, one image reconstructed with the given dual speed of sound mask.

In Figure 42 we can see the three different images. One reconstructed with $c_{couplant}$, one with c_{tissue} and one with both. In this sample we can see that the difference between both single speed of sound images is big. The shift in the upper and lower regions is multiple pixels. That would require a big receptive field for the network to fuse the details of the right areas of these images.

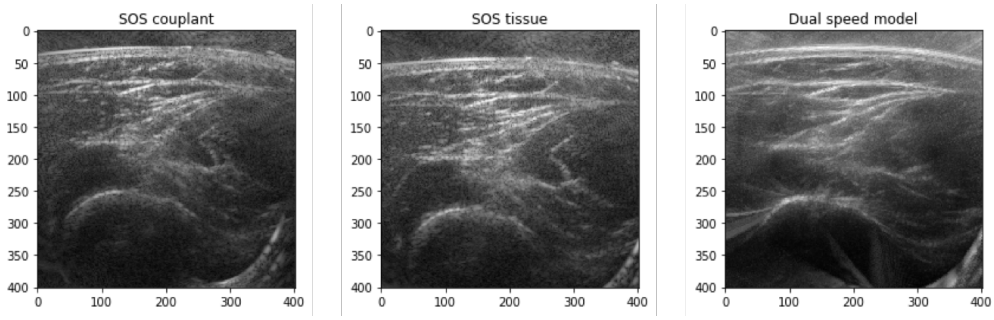

Figure 42: Left: input image with a couplant speed of sound. Middle: input image with a tissue speed of sound. Right: reconstructed image with dual speed of sound.

6.2.3 Methods

The analysis of the data shows that a big receptive field is necessary. Furthermore, the important information is possibly many pixels apart. To overcome this issue, we decided to apply a translation to the image reconstructed with c_{tissue}, such that the membrane is at the same location as in the one reconstructed with $c_{couplant}$. In Figure 43 we can see an image reconstructed with c_{tissue} and the dual speed of sound mask. We want to translate the image by an approximation of the difference between the membranes, denoted as y. To detect the boundary between couplant and tissue for the dual speed of sound model, it is detected using the couplant speed of sound $c_{couplant}$. Therefore the membrane of the dual speed of sound image is at the same location as in the reconstructed input image using $c_{couplant}$. The distance from the top to the membrane in the mask is denoted as d while the distance from the top to the membrane of the image with c_{tissue} is denoted as Y. We can express the distance y as the following:

$$y = Y - d \approx (c_{tissue} - c_{couplant}) \cdot \alpha \tag{4}$$

where we approximate α as a constant factor. We calculated the factor for 30 training images and used the mean as our fixed constant. With this information we can translate the images which are reconstructed with c_{tissue} for y pixels and add zeros at the missing end. In Figure 44 we can see the result of the translation. As one can see, the locations of the structures are approximately on the same level. Thus a smaller receptive field is sufficient to extract the information.

As an outlook, we will suggest a more sophisticated way to calculate the translation for each image. The difference y can be expressed as

$$y = d_{tissue} - d_{real} = t \cdot c_{tissue} - t \cdot c_{couplant} = t \cdot (c_{tissue} - c_{couplant}) \tag{5}$$

where d_{real} is the distance from the detector to the membrane. t is the time that the

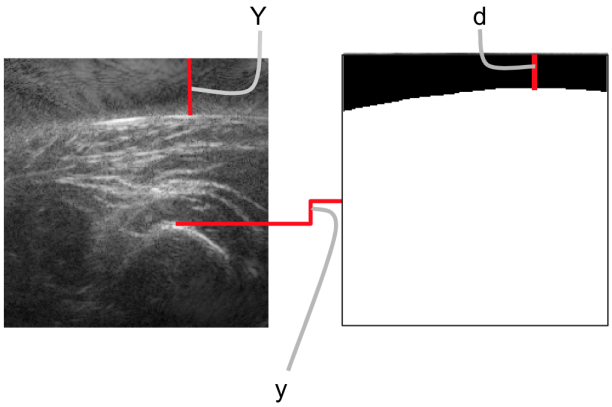

Figure 43: Left: image reconstructed with c_{tissue}. Right: dual speed of sound mask.

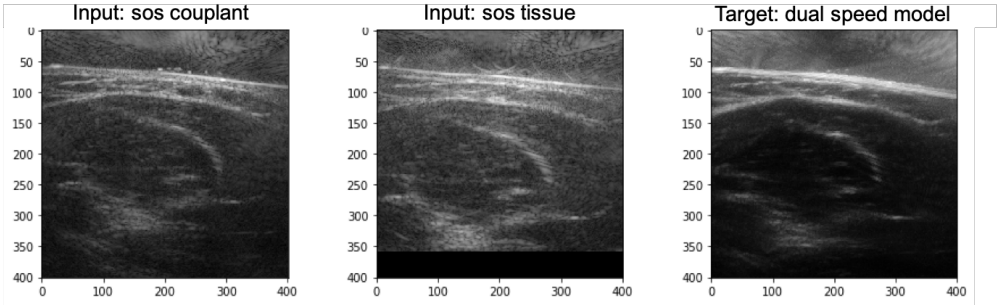

Figure 44: Processed sample. Left: input image reconstructed with $c_{couplant}$. Middle: input image reconstructed with c_{tissue} and shifted. Right: target image reconstructed with dual speed of sound.

sound wave needs to propagate to the membrane and is given by

$$t = \frac{d_{real}}{c_{couplant}} = \frac{0.04 + d}{c_{couplant}} \tag{6}$$

Since the detector is four centimeters above the image. The t is now the factor we approximated before, but now we have it with higher precision since the distance to the membrane d is extracted from the image with the couplant speed of sound model.

In this subproject we mask the images with attention masks as well, similar as in Section 6.1.3. However, in this case the first two slices were taken from the image reconstructed with the couplant speed of sound. The first one only contains the noise above the membrane, whereas the second is the area containing the membrane and skin

layer. The structures present up till the end of the second slice are closer to the target since we have the same speed of sound up to the membrane. The deeper we are into the tissue, the stronger the deformation compared to the dual speed of sound. That means, the shape of the membrane of the input image with couplant speed of sound is exactly the same as in the target and still close just below the skin. The last two slices are used from the image reconstructed with tissue speed of sound and take up about three quarters of the whole height of the image. The representation of the tissue should be closer to the one in the target since it has the same speed of sound for that area. We used the same augmentations as described in Section 6.1.3.

6.2.4 Model

For this relaxed problem we used a similar model as the one presented in Section 6.1.4. Since we shifted the image which is reconstructed with tissue speed of sound, a smaller receptive field is sufficient, see Section 6.2.3. Thus, we now alternated stride one and two for each layer. That way, we keep more details in the feature maps and have a smaller receptive field. The deformations can be up to around 40 pixels, thus we take a large kernel to enable the network getting the information over large distances. In order to preserve resolution we use stride one and same padding.

The details of the model we used are: five convolutional layers with (64, 128, 256, 512, 1024) channels, kernel size (7,7) and stride [(1,1), (2,2), (1,1), (2,2), (1,1)] for the respective convolutional layers. We used padding of (2,2) and output padding of (1,1) at the third and sixth layer. These layers are followed by the corresponding transpose convolutional counterparts to end with a feature map of the same output dimension. In the end we have two additional convolutional layers with (8, 1) channels and kernel size (41,11), stride (1,1) and same padding.

6.2.5 Result

We trained the described model for about 40 epochs. The results of the loss curves are shown in Figure 45. As one can see the losses are still strongly decreasing, even though the validation loss started to drift away from the training, one can expect better results if one continues the training. Unfortunately one epoch takes a bit more than two hours to train, that is the reason why we were not able to finish the full run until this point.

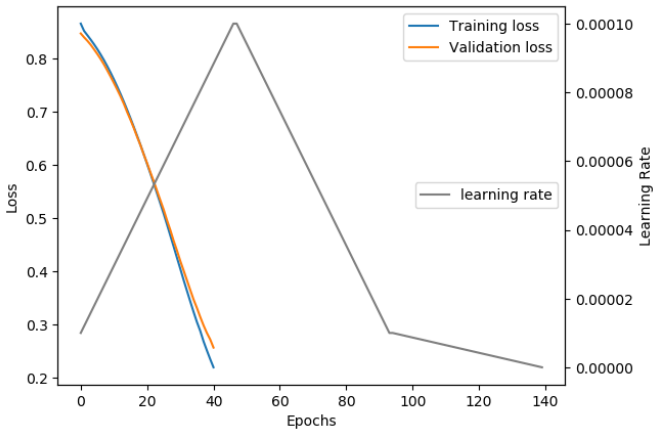

Figure 45: Training and validation loss of the run of the relaxed problem

Nevertheless, one can already see a significant improvement compared to the results of the original problem presented in Subsection 6.1.5. In Figure 46 one can see that not only the membrane at the top is at the correct position, but also the other main structures are clearly visible together with most of the details. Furthermore even a lot of the noise was filtered in our prediction, which can be seen especially in the center of the image.

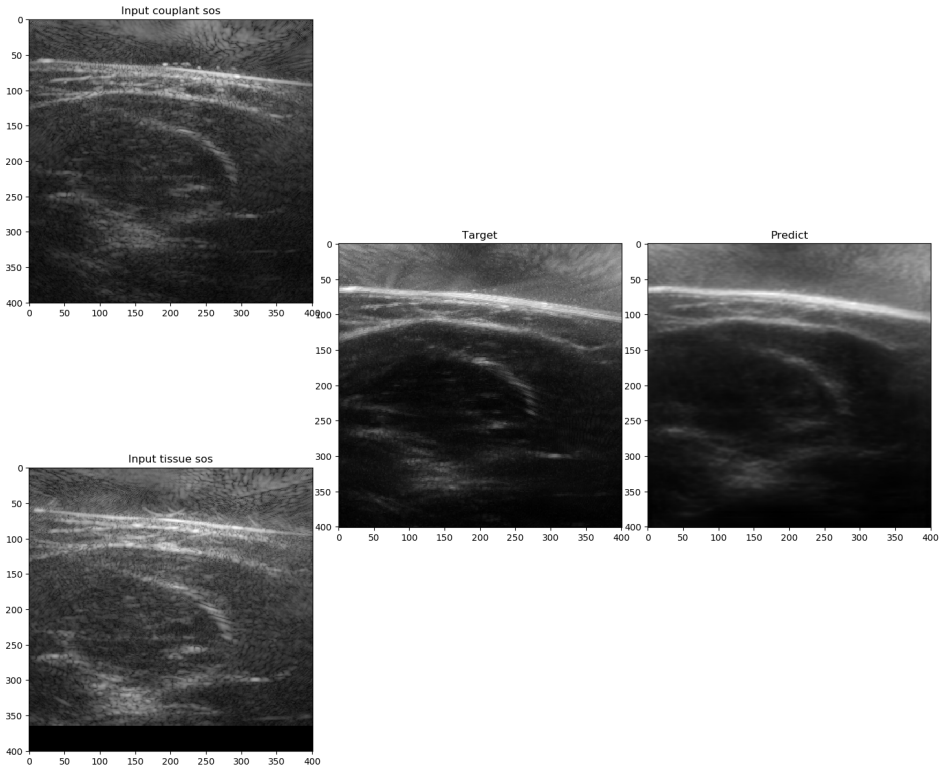

Figure 46: Result of a validation image. The two input images reconstructed with the couplant and tissue speed of sound are depicted on the left, followed by the target and prediction.

Another example is the test image shown in Figure 47. Here, even the very fine structures at the top of the target image, just below the membrane, is strikingly clear in the prediction. Which shows the great improvement compared to the results of the original problem of subproject 2.

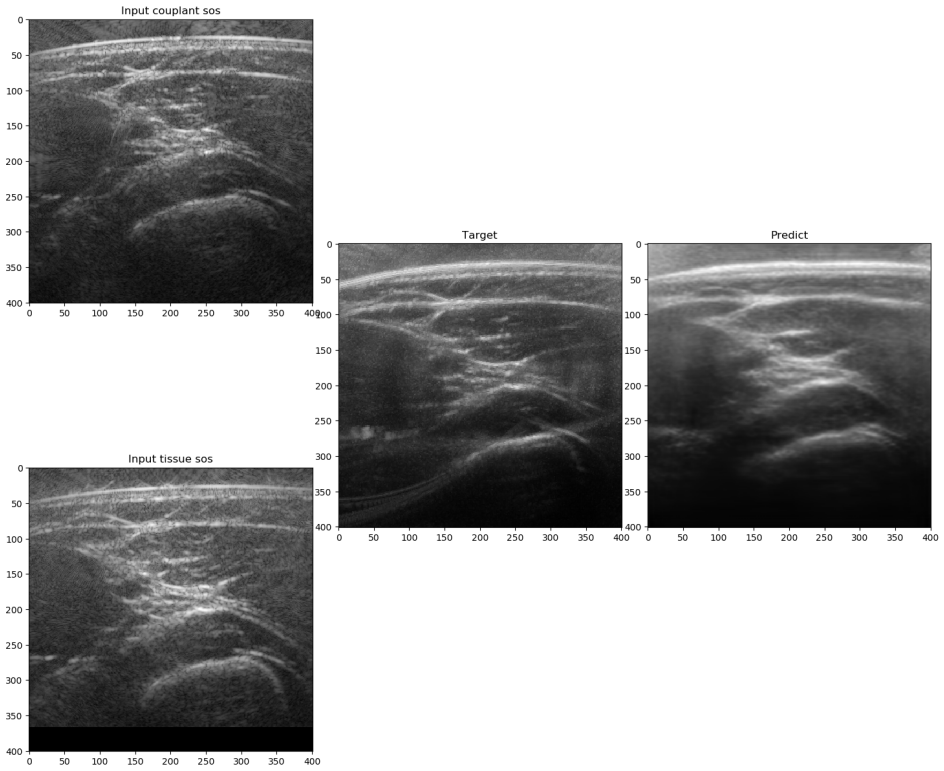

Figure 47: Result of a test image. The two input images reconstructed with the couplant and tissue speed of sound are depicted on the left, followed by the target and prediction.

7 Discussion

Subproject 1 Ultrasound

The task of denoising the ultrasound images in Subproject 1 should give us insights to the data and the general problem of image to image translation. Even here we could already see that the denoising is hard for our chosen network structure. Although our predicted results are visually better than the input, there is still an obvious gap in quality compared to the target.
The performance on the training data is really good, so we can assume that the optimization error, i.e. the error to the optimal empirical risk minimizer introduced in Section 2, is quite low. Still, the huge gap between the loss curves might result from a high estimation error. That means that the generalization capacity of our model is not sufficient. However, one has to keep in mind that the ability to generalize strongly depends on the characteristics of the training data in the overall feature space of the problem. The training data probably cannot cover much of that space, which means the validation set might lie outside of the covered feature space. Therefore our training set might not be rich enough for the trained model to be able to generalize well. To prevent biasing our network too strongly on the training set, regularization techniques might be useful.
This gap might shrink substantially with a growing data set, but to further boost the performance one would need to put more thought into the parameter choice and some structural adjustments as well. Also, acquiring more data does not always lead to better results. We tried to include an older study into our data set, but as there has been a former software update, the quality varied from our original data. The difference was so big, that including this additional data worsened our result instead of boosting it.
As one could observe in Figure 10, there is not only the difference in image quality between input and target data. There are also some artifacts in the target images that are not present in the input. Our model therefore tries to learn unwanted characteristics. This fact also hints that the approximation error, which encodes the error between the best model we can build and the true underlying distribution, is probably not negligible.
All in all this task was thought as a warm up, as we more or less could only focus on denoising the image and got reasonable results for that. Furthermore, the generalization capacity will probably increase with a larger data set, even using the same network, given that the data is retrieved under similar conditions as before. Even though more optimization would be needed to get better insights, we decided to focus our effort on the other subprojects, as these are of higher interest concerning the overall result of the project. Besides, we might not even be able to solve it completely due to the high approximation error.

Subproject 1 Optoacoustic

During the course of the optoacoustic subproject, we realized that our used structure is not capable of extracting and reconstructing the significant information when all 28 channels were used. More precisely, we did not even manage to properly overfit on a

single sample. That is the reason why we performed certain changes on the data, to be able to focus on special aspects. This has also led to different possible use cases and results for each of the approaches.
Firstly we took only a subset of the channels, which reduced the data size and complexity substantially. However, we lose a lot of information in the slicing step, which strongly limits the quality of the predictions. That is why we decided to learn on the regression coefficients as the second approach. This not only reduces the data size, but also deletes some of the noise in the input. In comparison to the sliced approach, we are therefore keeping all of the information in the images that is needed for the visualization. Unfortunately we lost the possibility to check each pixel for significance and also biased ourselves strongly on the four base tissue types, introduced in Figure 16. There might be some other tissues of interest that are completely neglected in this approach. These would be detected in the reconstruction of the signal, but deleted with extracting the regression coefficients in the beginning. Therefore we chose yet another approach. Instead of taking a fixed subset of the channels, we performed a PCA to extract the principal components in the target images, and projected input and target data onto this space. With this, we are keeping more of the data structure than in the sliced approach, but also possibly delete a lot of the noise in the input and some in the target. This showed much better results than just taking a slice of the channels and we can even back-project into the 28 dimensional space to check for significance. However, as the first four extracted principle components are close to the spectra of the main tissue types, almost all pixels are depicted as significant (compare Figure 16 and Figure 21). Hence this approach is also not suited to keep the information about significance. Furthermore as we are fitting the principal components to the training data, we are biased to the main tissue types in the data.
For all of the loss curves the difference in validation and training loss is relatively high. One explanation for that might be a high estimation error, again due to the small number of samples. Nonetheless, the approximation error is not negligible, especially not in the PCA approach, as we are deleting a lot of information to bias on the target images. On the other hand, the performance on the training data is reasonably good for all three approaches. Hence we can assume that the optimization error is quite low.
Another possible use case for optoacoustic imaging in general is detecting injected contrast agents, which can possibly have a completely different absorption spectrum than the base tissue types. If we are only using the regression coefficients of the base tissue types, we completely rule out the possibility to detect some contrast agents. On the other hand one can include the absorption spectrum of the contrast agent in the regression in the beginning. In the PCA approach, this would only be detected, if the contrast agents would be adequately represented in the training data, such that it is not neglected when fitting the PCA. To extract important features for this, this aspect would also be important for all approaches. However, we had no such data for our project and these ideas could not be verified.
All in all, we achieved reasonable results for the problem that is biased on the base spectra. Concerning the whole problem of optoacoustic imaging, it cannot be sufficiently covered with this amount of data and strong focus on the main tissue types. So, more

research has to be done in this aspect.

Subproject 2 Original Problem

At the beginning of this new task, besides denoising, we especially needed to figure out how to capture the additional deformation from the input to the target image. It became clear that our network used for the denoising in Subproject 1 was not sufficient to capture this, because the filter parameters in convolutional layers are shared and hence optimized over the whole image. Therefore, we duplicated our input and introduced masks to focus the attention on different areas of the images to enable extracting features without other areas interfering with the extraction process. Without this, the network was not able to capture any deformation at all, which is due to the varying kinds of deformations depending on the depth inside the tissue. Some of these deformations might be reversed for the upper and lower part respectively, erasing the learned deformations of the other. However, using the attention mask approach, we can now see that the needed deformation is being learned. This is visible through the main structures, especially the membrane, shifting accordingly to the target, as seen in Figure 41. Nonetheless, the fine structures and other details are lost in our prediction. That is due to the fact, that our idea of learning the error term from Subproject 1 is no longer valid. For this approach to work, one would need the comparison of the deformed input image and the target, to detect the noise inside the target. Unfortunately, we were not able to decouple those tasks accordingly, as described in Section 6.1.5.
Adding the last convolutional layer with stride one and convolving the masked input image with the deformed feature map increased the quality of the prediction visibly, but unfortunately not to a satisfying level. The problem with that layer is that the difference of the structures in the original image and the deformed map can very well be 40 pixels and thus we would need a bigger receptive field.
This is also clearly visible in the loss curves, showing that the network might capture specific shifts in the training data, but is not able to generalize to the validation set. That is probably the reason why the train and validation loss are drifting apart very early in the training process. Here we obviously cannot keep the estimation error under control, as the network is not able to extract the significant features.
From this we concluded that learning the deformation is possible with putting attention to different areas. But we were not able to keep the details in this step, as the spatial differences between input and target structures are too big. This gave rise to the idea of the relaxed problem.

Subproject 2 Relaxed Problem

In this last part of our project, we proposed a relaxation of the original problem of Subproject 2. Instead of trying to learn the mapping from any homogeneous speed of sound to any dual speed of sound, we are now only mapping from two homogeneous speed of sound input images to the dual speed of sound model, which uses the same speeds as the two input images. Even though there is still a non-trivial deformation of main structures

from input to target, these are now closer together. The assumption is that learning the deformation like in the original problem of Subproject 2 and then trying to extract the details from the input images, should now be a lot easier, because the structures are not so far apart.
And as a matter of fact, we can see that the results are way better, as the details are also kept in the validation and test images. The feature extraction of the training set seems to be vastly more significant for the validation set, than it was possible to achieve in the original problem. This can also be seen in the validation loss staying close to the training loss for longer in the training process.
Even though creating the input is computationally more complex than for the original problem of Subproject 2, as two homogeneous speed of sound images have to be produced, it is still feasible for the use case of achieving real time imaging. However, we not only changed the problem statement, but also used a different network. Compared to the model used in the original problem there are two major differences. The first is the usage of stride (1,1) for every second layer and the second is adding a last convolutional layer to the very end, again with stride (1,1) and same padding. This resulted in a computationally very costly training phase, which takes about two hours per epoch. Additionally, the size of the model is probably too large for real time imaging on a single GPU that is available in the machine.
Nevertheless, we did not perform any further tests or optimization of the network and code structure. To finalize the evaluation of this approach, this would be surely needed and it may very well be, that one could lower the computational time sufficiently to enable real time imaging.
Most likely, this network structure would also boost the results for the other tasks of this project, such as those of original problem of Subproject 2. However, due to the time constraints of this project we were not able to test this anymore.

8 Conclusion & Outlook

With optoacoustic and ultrasound imaging, it is possible to do fast and non invasive clinical imaging of stuctures and tissue up until a few centimeters below the surface of the skin. High quality image reconstruction in this field however is too slow for real time applications. In this project, we propose deep learning solutions to map low quality, easily accessible images to their high quality counterparts.
In Subproject 1, the task was mainly about denoising and we used the ultrasound part of this subproject as a warm-up phase and stepping stone to the other, more complex tasks. In being confronted with high-dimensional optoacoustic images, the limitations of the computational resources and the base model became apparent for the first time and we developed several approaches in order to alleviate these problems. In both the optoacoustic and the ultrasound part, we achieved improvement of image quality. However, we were not able capture all of the details in the ultrasound images and could not develop a generic approach to cover all possible use cases of the optoacoustic imaging.
In Subproject 2, additionally to denoising the images, we had to bridge the divide between different speed of sound models, which meant learning deformations. For standard convolutional networks, this is hardly possible because of the parameter sharing and the local focus of convolutional layers. However, introducing fully connected layers, or even locally connected layers, would increase the number of parameters to an unmanageable amount. As a compromise, in our approach the parameters of the convolution are only shared among certain regions and very large convolutional filters are applied at the end of the network. Even with these model adaptations, mapping between arbitrary speed of sound values proves to be very difficult: We were able to produce the rough deformations but lost most of the image content. In a relaxed version of the task, the model input includes images reconstructed with a homogeneous speed of sound model with the respective target speed of sound values. The deformation from input to target is still non-trivial, but it is locally closer together, which eases up the task significantly. Additionally we only have to learn the mapping from already matching single speed of sound images to the corresponding dual speed of sound and not the generalized form of any single to any dual speed of sound image. Performing small adjustments to the used network architecture resulted in a significant improvement of the results. We not only could learn the deformation but keep most of the content as well. The results in the discussed subprojects and tasks proved that enhancement of ultrasound and optoacoustic images w.r.t. image quality and speed of sound model with a deep learning architecture is possible. However, the task at hand has to be analyzed thoroughly and the model adjusted according to specific domain knowledge.
The domain of interest for further research concerning the applications for ultrasound imaging lies clearly in the tasks of Subproject 2. There are several points to improve our results of the relaxed problem. One was already mentioned in Section 6.2.3, where one can calculating an image specific translation and not the same approximated for all images. Additionally the used network architecture can be optimized further to enhance quality and lower computational complexity. Our results in optoacoustic imaging gave insights to the general task of multispectra image translation. However, one would need

to further specify use cases and additional suitable evaluation techniques to gain further insights to this problem. Especially to extend our work to targets reconstructed with a dual speed of sound model, similar to the task in Subproject 2.

References

[1] C. Chase. *Demand-Driven Forecasting : A Structured Approach to Forecasting.* Wiley and SAS Business Series. Wiley, 2013.

[2] B. Cox, J. G. Laufer, S. R. Arridge, and P. C. Beard. Quantitative spectroscopic photoacoustic imaging: a review. *Journal of Biomedical Optics*, 17(6)(061202), 2012.

[3] K. Fujii and R. Masui. Accurate measurements of the sound velocity in pure water by combining a coherent phase-detection technique and a variable path-length interferometer. *The Journal of the Acoustical Society of America*, 93(1):276–282, 1993.

[4] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. 2017. arXiv:1706.02677v2.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456, 2015.

[6] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. 2015. arXiv:1506.02025.

[7] I. T. Jollife. *Principal component analysis.* Springer series in statistics. Springer, 2002.

[8] D. Jüstel. TUM data innovation lab project: Enhancement of clinical optoacoustic and ultrasound images (internal presentation). IBMI/CBI, TUM, Helmholtz Zentrum München, 2018.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014. arXiv:1412.6980.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012.

[11] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller. *Efficient BackProp*, pages 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[12] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2017. arXiv:1711.05101.

[13] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. 2016. arXiv:1606.08921.

[14] M. Mohri, R. A., and T. A. *Foundations of Machine Learning.* The MIT Press, Cambridge, Massachusetts, 2012.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. arXiv:1505.04597.

[16] A. Rosenthal, V. Ntziachristos, and D. Razansky. Acoustic inversion in optoacoustic tomography: A review. *Current Medical Imaging Review*, 9:318–336, 2013.

[17] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1609.05158.

[18] B. Shruthi, S. M, and S. Renukalatha. Speckle noise reduction in ultrasound images - a review. *International Journal of Engineering Research & Technology*, 4(2):1042–1046, 2015.

[19] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. 2018. arXiv:1803.09820.

[20] S. Vedula, O. Senouf, A. M. Bronstein, O. V. Michailovich, and M. Zibulevsky. Towards CT-quality ultrasound imaging using deep learning. 2017. arXiv:1710.06304.