Date of acceptance Grade

Instructor

Reconstructing simulated breast phantoms using neural networks inspired by the problem geometry

Joel Enwald

Helsinki August 25, 2020

Master's degree

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Joel Enwald

Työn nimi — Arbetets titel — Title

Reconstructing simulated breast phantoms using neural networks inspired by the problem geometry

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's degree	August 25, 2020	36 pages + 2appendix pages

Tiivistelmä — Referat — Abstract

Mammography is used as an early detection system for breast cancer, which is one of the most common types of cancer, regardless of one's sex. Mammography uses specialised X-ray machines to look into the breast tissue for possible tumours. Due to the machine's set-up as well as to reduce the radiation patients are exposed to, the number of X-ray measurements collected is very restricted. Reconstructing the tissue from this limited information is referred to as limited-angle tomography. This is a complex mathematical problem and ordinarily leads to poor reconstruction results. The aim of this work is to investigate how well a neural network whose structure utilizes pre-existing models and known geometry of the problem performs at this task.

In this preliminary work, we demonstrate the results on simulated two-dimensional phantoms and discuss the extension of the results to 3-dimensional patient data.

Avainsanat — Nyckelord — Keywords

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Breast cancer is one of the most common forms of cancer, and the most prevalent one in women[1]. Existing cancer screening methods help lower the mortality rates of cancer [2, 3, 4, 5], but could be more accurate in terms of specificity. Overdiagnosis rates in randomized trials have been found to be between 11 and 22% [6]. These false positive results can cause additional anxiety and distress [7][8]. The additional screening caused by false positives have monetary costs associated to them and the additional radiation can even cause cancer, although this is rare [6]. There is thus motivation for finding more accurate methods of recognizing breast cancer, and ways of doing so while exposing the patient to the least amount of radiation possible.

There are several breast imaging modalities used to diagnose breast cancer, all falling under an umbrella term of *mammography*. Conventional 2D mammography is done by taking a measurement of a compressed breast directly from above (see figure 1) and another from the side [9], which results in two-dimensional images. In 3D computed tomography and tomosynthesis, the breast is measured from different angles (see figure 2) and then a 3-dimensional reconstruction is consctructed.

Tomosynthesis using a limited field of view for imaging has a lower radiation dose and cost than that of conventional 3D computed tomography (CT) [10], despite providing many of the same benefits such as better visibility of microcalcifications and low-contrast objects [11] as well as depth information [12]. However the problem of reconstruction from a limited field of view is very challenging, and a standard approach using filtered back-projection results in images with strong artifacts.

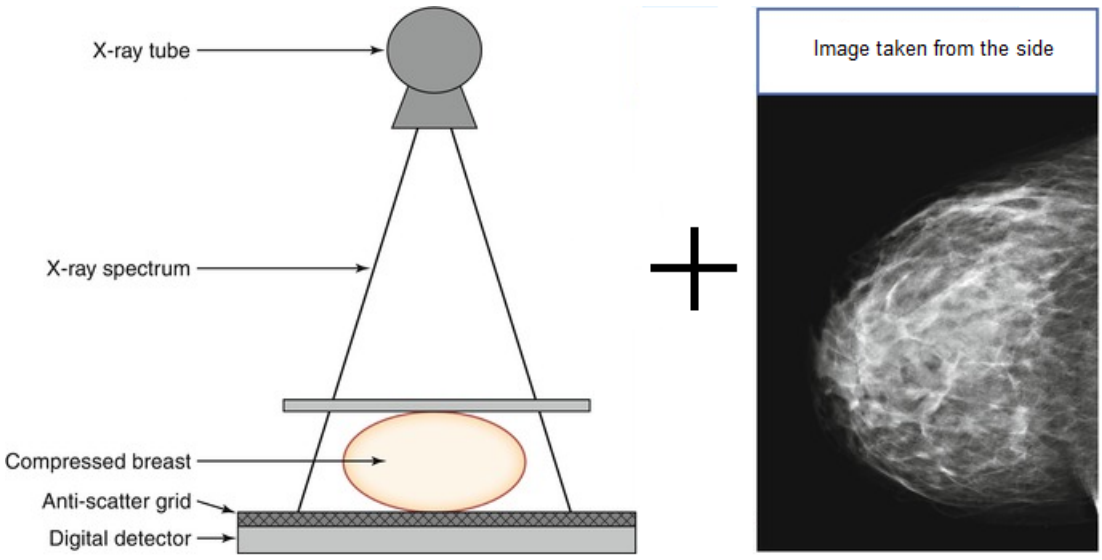

Figure 1: A conventional tomography setup[13]

In this paper is presented a two-part neural network that first uses a learned version of a basic algorithm called Shift-and-add, and then sharpens and denoises the preliminary reconstruction resulting from that. Due to the method being able to learn

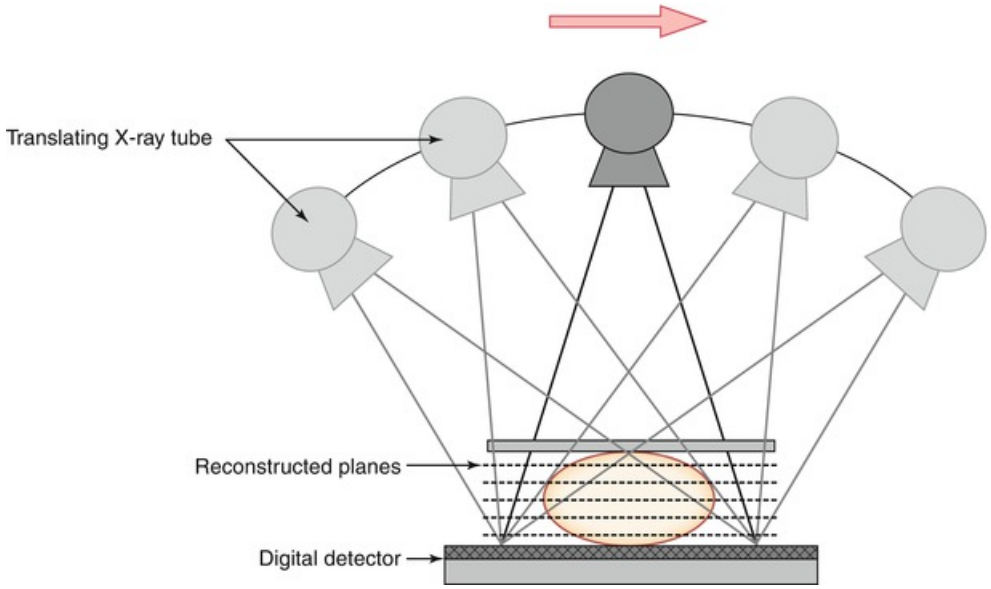

Figure 2: A conventional tomosynthesis setup[13]

features from data, it can adapt to specific inputs and produce good results while also being partially interpretable.

This work was originally part of a joint project between universities of Helsinki and Oulu called AIDMEI, which aims to develop reliable and understandable machine learning solutions to improve MRI and mammographic imaging methods.

2 Background concepts

2.1 X-ray measurements in mathematical form

When an object is imaged tomographically, penetrating waves are sent through it. They get weaker due to being absorbed by matter, and their intensity is measured on the other side of the object. Mathematically these waves are actually modelled as rays.

Consider that we are imaging a 2D object. Let L be an imaging ray passing through the object, and the attenuation (absorption rate) of the object at point (x, y) be $f(x, y)$. Let (x_0, y_0) be the point where the ray first enters the object, and (x_1, y_1) the point where the ray leaves it. When a ray is first shot, it has an intensity of $I(x_0, y_0)$ and when it is measured, an intensity of $I(x_1, y_1)$. See figure 3 for reference. From this we would like to find out the attenuation values of the object.

The attenuation function f at point (x, y) in the object defines the relative weakening of a ray of radiation at that point when it moves a small distance dt into some

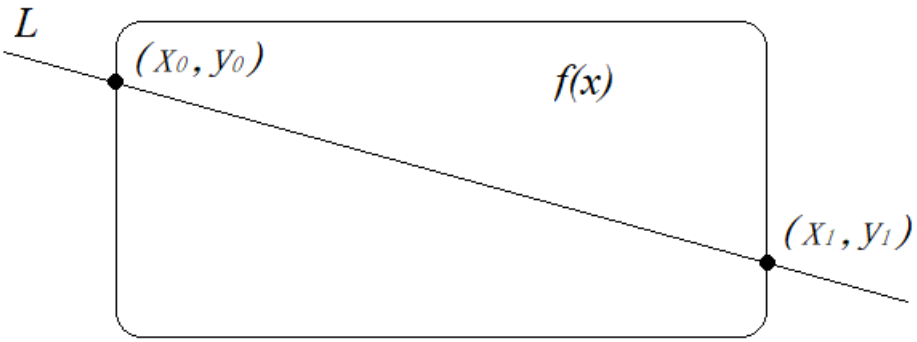

Figure 3: Ray L being shot through an object with attenuation function $f(x)$.

direction,

$$f(x,y)dt = -\frac{dI(x,y)}{I(x,y)}$$

Now let $m : \mathbb{R} \to \mathbb{R}^2, m(s) = (1-s)(x_0, y_0) + s(x_1, y_1)$ be a function mapping the interval $[0, 1]$ to the line segment between (x_0, y_0) and (x_1, y_1). We can then get a line integral of the function $f(x, y)$ from (x_0, y_0) to (x_1, y_1) as

$$\begin{aligned}
&\int_0^1 f(m(s))ds \\
=&-\int_0^1 \frac{dI(m(s))}{I(m(s))} \\
=&-\int_0^1 \frac{\frac{dI(m(s))}{dm(s)}}{I(m(s))}\frac{dm(s)}{ds}ds \\
=&-\int_0^1 \frac{I'(m(s))}{I(m(s))}m'(s)ds \\
=&-(\log(I(m(1))) - \log(I(m(0)))) \\
=&\log(I(x_0, y_0)) - \log(I(x_1, y_1))
\end{aligned}$$

This means that if we know the initial energy of the ray, and the final energy at the moment of measuring, we can calculate the integral of the attenuation over the line. Taking several measurements with different offsets from multiple angles around the object is called a radon transform, and the resulting data of measurements a sinogram.

2.2 Radon transform and back-projection

The Radon transform in 2D is an integral operator that maps a 2-parameter function into a 2-dimensional parameter space where the first parameter determines the angle of a line in the original space, and the second its distance from origin.

Let $A = (x, y)$ be a point in a phantom, α the angle at which a ray passes through the point (x, y), and s the shortest distance from origin o to that line. The origin is set to be in the centre of the layer of the phantom that is closest to the detector. The Radon transform is then defined as

$$R_f(\alpha, s) = \int_{-\infty}^{\infty} f(s\langle\cos(\alpha), \sin(\alpha)\rangle + t\langle\sin(\alpha), -\cos(\alpha)\rangle)dt.$$

Parameters α and s specify the direction of the line over which the integral is taken.Vector $s\langle\cos(\alpha), \sin(\alpha)\rangle$ is perpendicular to the line, and $t\langle\sin(\alpha), -\cos(\alpha)\rangle$ is parallel to it. Each point in the sinogram is given by individual line integrals.

Meanwhile, the value at each point in the back-projected reconstruction is the integral over the angles of a half-circle given by

$$\begin{aligned} f_{BP}(x, y) &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} R_f(\alpha, \langle x, y\rangle \cdot \langle\cos(\alpha), \sin(\alpha)\rangle d\alpha \\ &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} R_f(\alpha, (x\cos(\alpha) + y\sin(\alpha))d\alpha. \end{aligned}$$

Since only line integrals over the attenuation function can be measured and not the function's values themselves, a simple assumption is made in back-projection reconstruction. It assumes that along a given line, the function's values are constant. The measured values are thus projected to the image onto each line (see figure 4). These values are then averaged over the number of measurement angles, or an integral is taken as in the continuous case. Each reconstruction point's value will thus be an average of the line integrals of the lines passing through it (ignoring some constant multipliers).

2.3 Tomosynthesis and shift-and-add algorithm

The shift-and-add algorithm is a two-step process which allows reconstructing the 3D structure of an object, layer by layer. First, measurements from multiple angles are taken. The measurements are then aligned and summed together to form a reconstruction of a single layer. The same procedure is then repeated for each layer. A demonstration of the process can be found in figure 5.

Let p be a detail in a phantom at height y from the phantom's centre o. In figure 6, we can see how a point p in the phantom is imaged with two rays from differing angles. The intensities of the rays are measured by two different spots on the detector. The distance between these spots is Δx. We can calculate Δx from

$$\tan(\alpha) = \frac{\Delta x}{\frac{h}{2} + y},$$

giving us

$$\Delta x = \tan(\alpha)(\frac{h}{2} + y).$$

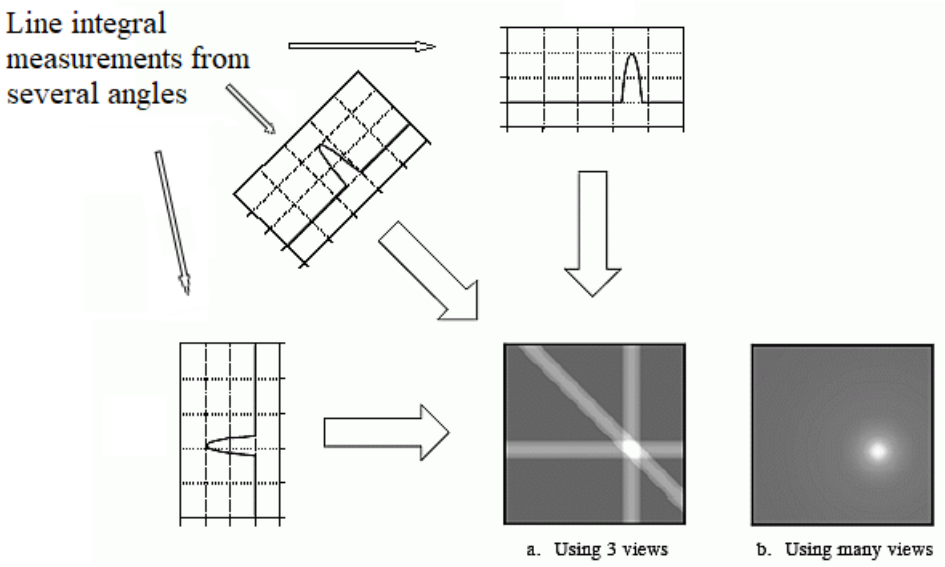

Figure 4: Line integrals being smeared across an image and added together to form a reconstruction. Original image source [14], modified for this work.

From this we see that the distance between the locations of two values in two different measurements corresponding to detail p in the phantom does not depend on the horizontal distance x of the point p from the centre. Therefore all values in a measurement corresponding to details at the same height as p in the phantom can be aligned with details in another measurement by shifting, and stretching them is not required.

2.4 Convolution

Convolution is a mathematical operation between two functions, with a third function as a result. It can be used to find shape similarities between the convolved functions.

Let f and g be integrable functions $\mathbb{R}^2 \to \mathbb{R}$. Now convolution between them is defined as

$$(f * g)(x, y) \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s, t) g((x, y) - (s, t)) ds dt. \tag{1}$$

$$\equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f((x, y) - (s, t)) g((s, t)) ds dt. \tag{2}$$

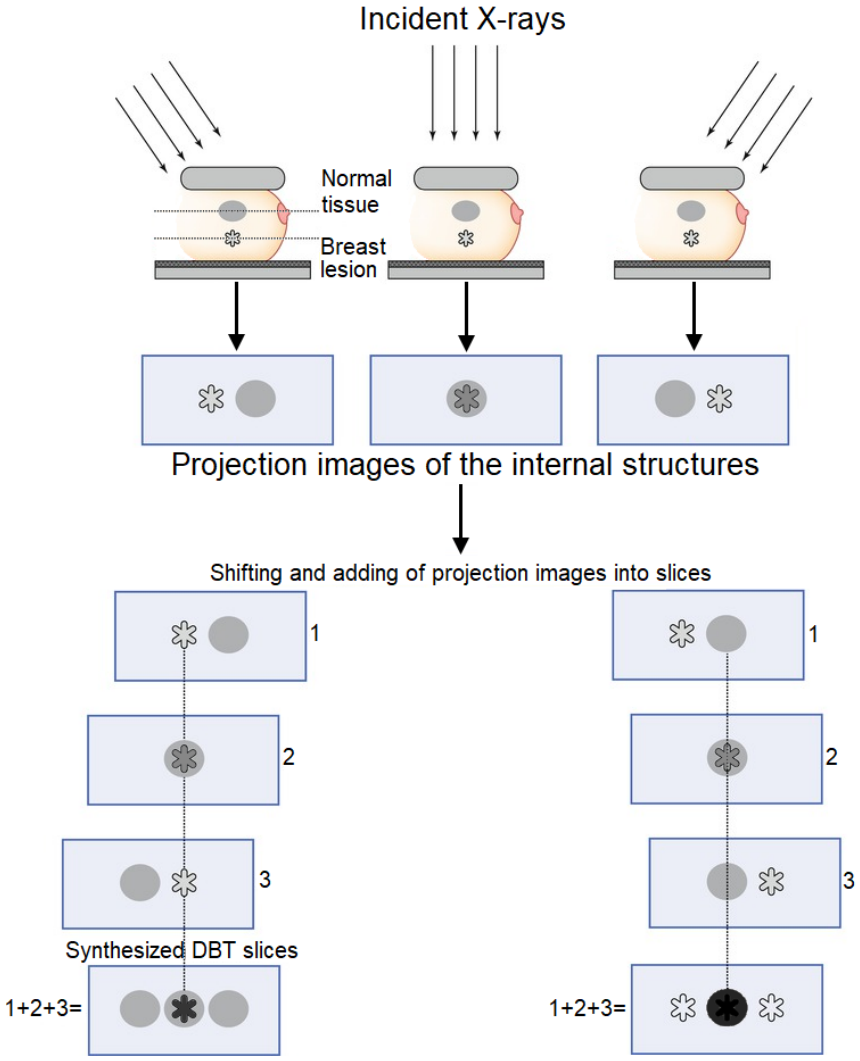

Figure 5: The shift-and-add algorithm. By shifting the measurements by specific amounts before overlaying them, features in different layers of the imaged object are brought into focus.[13]

If f and g are discrete functions $\mathbb{Z}^2 \to \mathbb{R}$, then it is defined as

$$(f * g)(n, m) \equiv \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(j, k) g((n, m) - (j, k)) \tag{3}$$

$$\equiv \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f((n, m) - (j, k)) g(j, k). \tag{4}$$

In the discrete 2D case with compactly supported functions, convolution becomes the operation between two matrices that outputs a third matrix. The smaller of the convolved matrices is sometimes referred to as a kernel. We will refer to the other convolved matrix as the input and the resulting matrix as the output. The convolution can then be thought of as the kernel moving over the input matrix

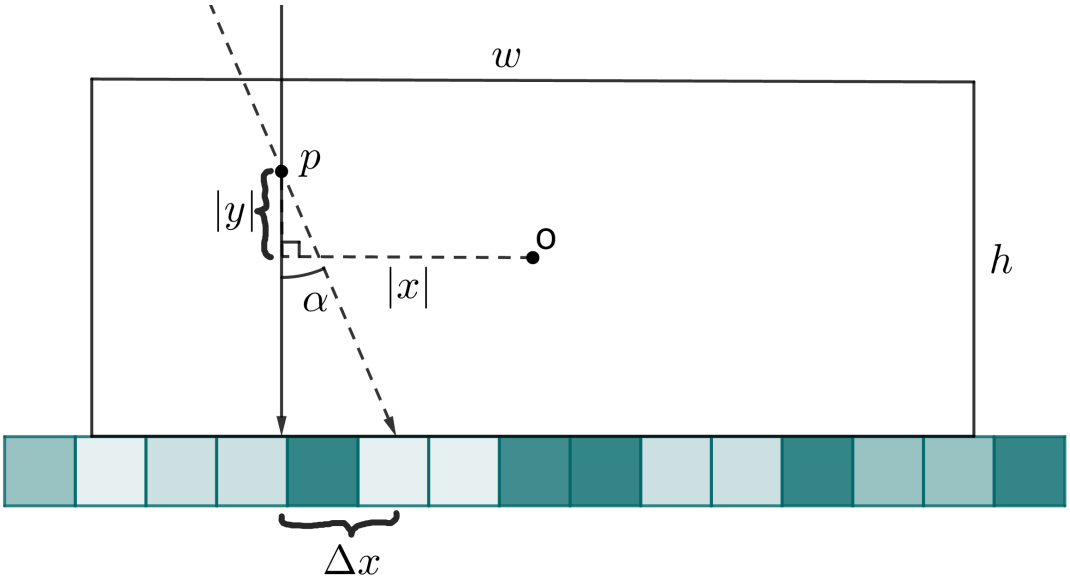

Figure 6: Measuring a detail in a phantom, using parallel-beam geometry.

in a sliding window fashion, and all the aligned values of the two matrices being multiplied and then summed up to get one element of the output matrix. See figure 7 for reference.

2.5 Neural networks

Artificial neural networks are a machine learning model that was originally based on the brain's neural connections. They are composed of connected neurons that form a network which can perform the job of an operator or even a full algorithm. The network is given an input and a target based on which the network adjusts the strength of the connections between its neurons in a way that more optimally converts the input to something approximating the desirable target. With more data instances, the network's connections are altered in a way that enables them to give desirable outputs for all the inputs in the data set.

Possibly the simplest form of a network is the so-called fully connected model. As the name suggests, it is composed of layers where all neurons in one layer connect to all neurons in the adjacent layers. A common model that requires less processing power is the convolutional network, where convolution is performed between layers, and parameters are the values in the convolution kernels. Using multiple convolution kernels in each layer and multiple layers enables the network to form complicated detection and modification mechanisms especially suited for 2D image data.

A complex neural network that consists of many layers and performs intricate tasks will often be referred to as a deep neural network (DNN). Such networks have been known to excel in different image processing tasks using photographs, x-ray scans, or other kind of data. This is most clear from the fact that starting as far as 2012

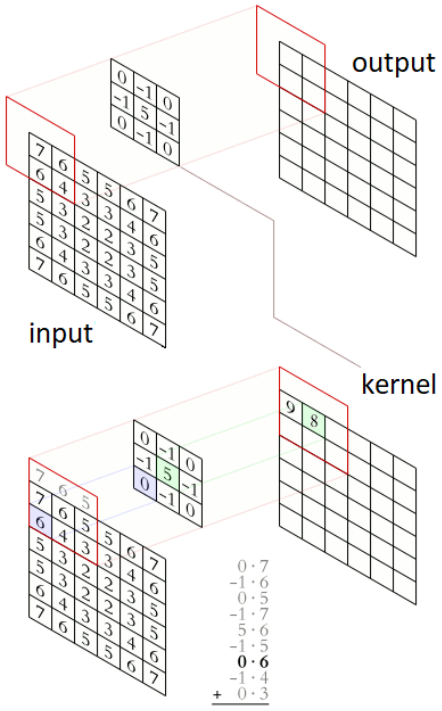

Figure 7: Two-dimensional discrete convolution with a 3x3 kernel. There are different solutions for when the kernel needs values outside of the other matrix. Often those values are simply said to be 0. In this case values on the edge are repeated over the edge. [15]

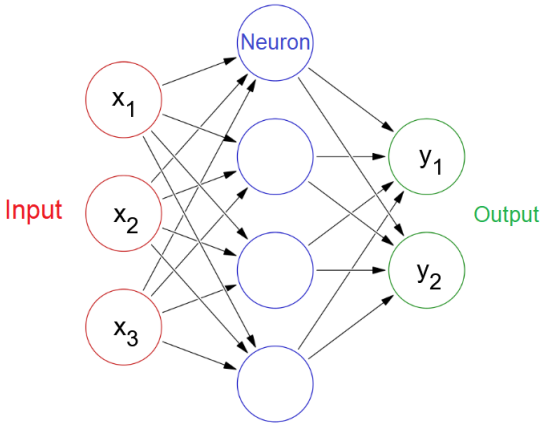

Figure 8: A simple fully connected neural network [16]

the winners of ImageNet Large-Scale Visual Recognition Challenge have all used a variety of DNNs to find solutions to the challenges. [18]. DNNs have also been successfully applied to tomosynthesis reconstruction tasks [19].

Most prevalent uses of neural networks in medical imaging are in the post-processing part of a reconstruction or detection task. This means that there is usually a preliminary reconstruction that uses traditional methods (such as (un)filtered backpro-

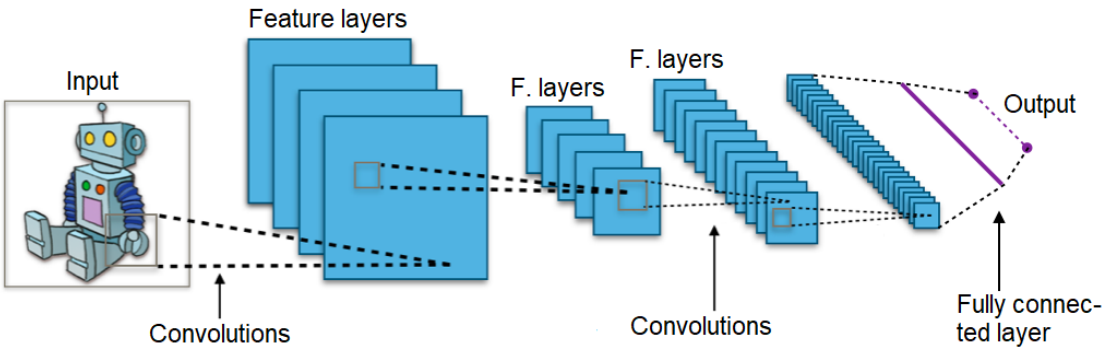

Figure 9: A typical convolutional neural network processing image data. Multiple convolution kernels result in multiple preliminary outputs called feature layers. [17]

jection or shift-and-add algorithm in the case of tomosynthesis). End-to-end neural networks, such as the one introduced in this paper, take measurements as input and output a reconstruction. This approach to neural networks is less common, but having a network that both reconstructs and post-processes the data is convenient and justifiable for our purposes. We create a neural network consisting of two main parts. The first part is a convolutional neural network (CNN) that mimics the shift-and-add algorithm to create an unfiltered backprojection reconstruction. Then, assuming that the unfiltered backprojection is a convolution on the original image, we deconvolve the reconstruction to get a better result.

3 Relationship between Shift-and-add and Convolution

3.1 Generalized Shift-and-add using a convolutional layer

Convolving a sinogram (fig.10) with a kernel such as the one in figure 11, gives us the equivalent of shifting the rows of the sinogram by amounts corresponding to the offsets of the pixels in the kernel from the central column, and then adding values in the same column together. In other words, the convolution performs a shift-and-add procedure, meaning that the shift-and-add algorithm can be implemented using a CNN. The convolution is demonstrated in figure 12, with the final result displayed in figure 13.

Figure 10: An example sinogram.

For testing purposes, we train a neural network to learn the shift-and-add reconstruction using the simulated 2D phantoms. We first train a network that learns

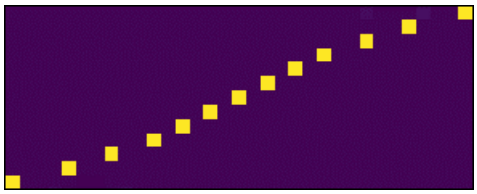

Figure 11: An example kernel used to mimic the Shift-and-add algorithm. Purple pixels correspond to 0's and yellow ones to 1's.

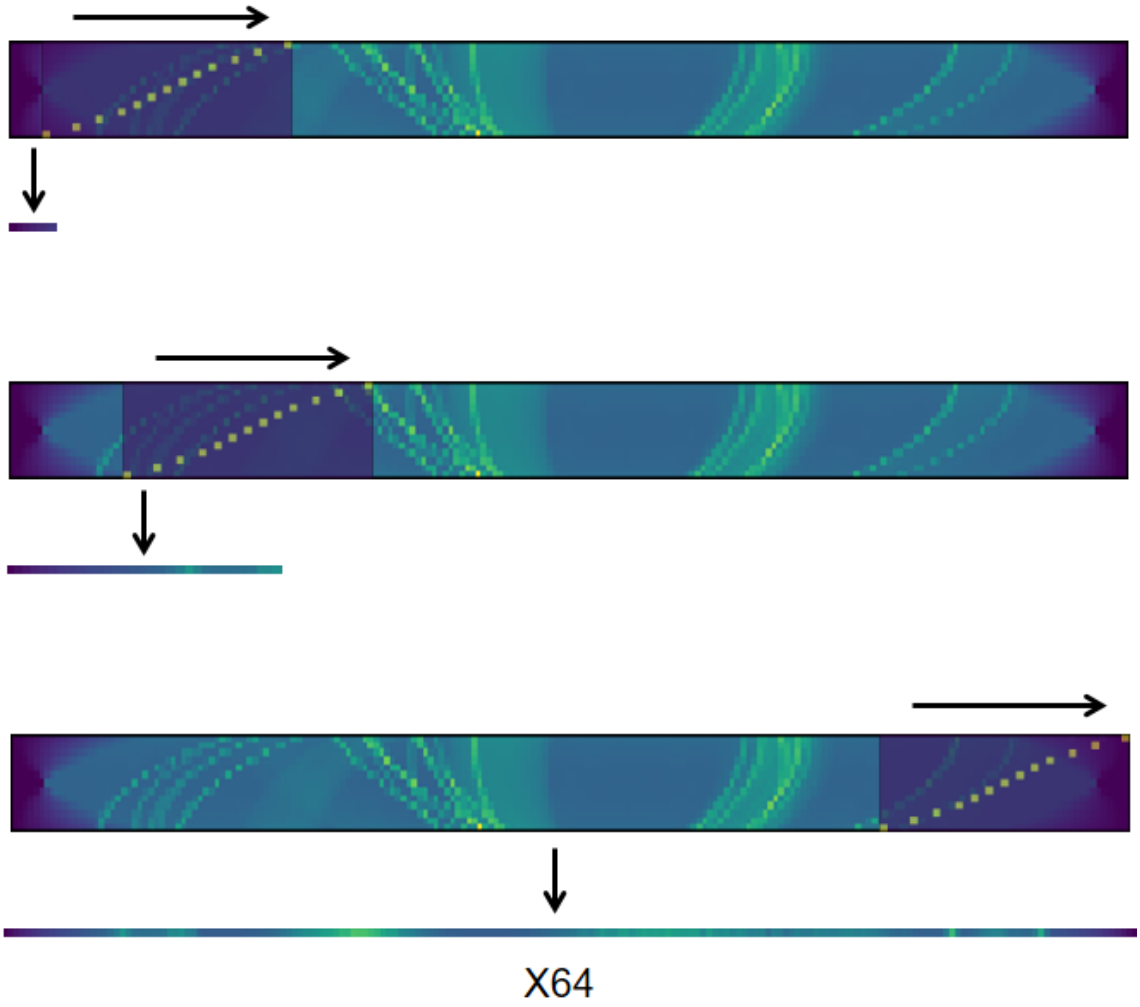

Figure 12: Demonstration of the shift and add algorithm as a convolution.

only a single kernel, using that to reconstruct the first row of the shift-and-add reconstruction. After training the network for 50 epochs, it learns a kernel that is pretty similar to the example kernel (figure 14). Excessive width of the kernel is due to the Python's scikit-image library's iradon-function that is used in this work.

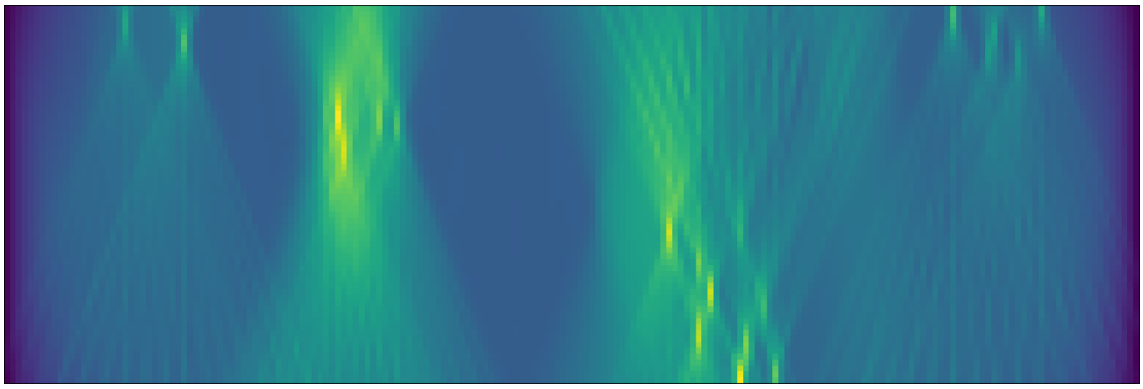

Figure 13: Reconstruction using the method demonstrated in figure 12

.

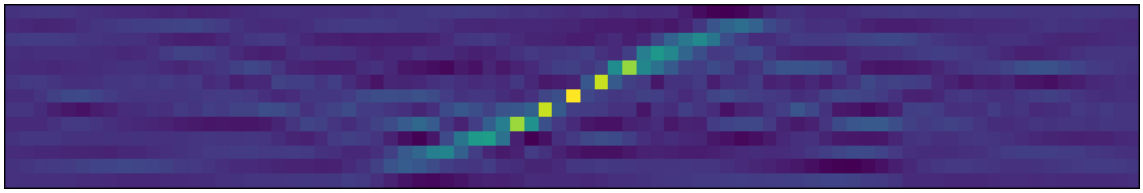

Figure 14: Learned kernel for the network learning the transformation for the first row of the shift-and-add reconstruction. Here using skimage-library's default geometry, hence the sinusoidal shape of the kernel.

3.2 Shift-and-add reconstruction as convolved image

In this section we demonstrate the use of the shift-and-add algorithm on idealised limited angle tomography, i.e. when an infinite number of projections can be collected from a limited range of angles. With such data the shift-and-add algorithm results in a reconstruction that is just the original image convolved by a specific kernel.

Let the image we want to reconstruct be f. In order to reconstruct it at point (x, y), we will take the Radon transform of f and back-project it back to the image space.

Let the function transforming polar coordinates to cartesian ones be $\sigma(t, \theta) = (t\cos(\theta), t\sin(\theta))$

We will show that in performing variable substitution while integrating σ, the differentials transform is $da_x da_y = tdtd\theta$.

The general form of the differentials transform for a function $(v_1, v_2) = \phi(u_1, u_2)$ is

$dv_1 dv_2 = |J(\phi)| du_1 du_2$, where $|J(\phi)|$ is the determinant of the Jacobian of ϕ.

$$\begin{aligned}
|J(\sigma)| &= \begin{vmatrix} \frac{\partial \sigma_1}{\partial t} & \frac{\partial \sigma_1}{\partial \theta} \\ \frac{\partial \sigma_2}{\partial t} & \frac{\partial \sigma_2}{\partial \theta} \end{vmatrix} \\
&= \begin{vmatrix} \cos(\theta) & -t\sin(\theta) \\ \sin(\theta) & t\cos(\theta) \end{vmatrix} \\
&= \cos(\theta) t\cos(\theta) - (-t\sin(\theta)\sin(\theta)) \\
&= t(\cos^2(\theta) + \sin^2(\theta)) \\
&= t
\end{aligned}$$

and thus we get

$$da_x da_y = t dt d\theta.$$

Now the back-projection is

$$f_{BP}(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} f((x,y) + (t\cos(\theta), t\sin(\theta))) dt d\theta \tag{5}$$

$$= \int_0^{2\pi} \int_0^{\infty} \frac{f(x,y) + (t\cos(\theta), t\sin(\theta)))}{t} t dt d\theta \tag{6}$$

$$= \int_{R^2} \frac{f((x,y) + (a_x, a_y))}{|(a_x, a_y)|} da_x da_y \quad || \ (a_x, a_y) = (a_x, a_y) - (x,y) \tag{7}$$

$$= \int_{R^2} \frac{f(a_x, a_y)}{|(a_x, a_y) - (x,y)|} da_x da_y \tag{8}$$

$$= \int_{R^2} f(a_x, a_y) \frac{1}{|(x,y) - (a_x, a_y)|} da_x da_y \tag{9}$$

$$= (f(a_x, a_y) * \frac{1}{|(a_x, a_y)|})(x,y), \tag{10}$$

where * stands for convolution. From line (2) to (3) we used a change of variables $a_x = t\cos(\theta), a_y = t\sin(\theta)$ which means that $t = \sqrt{a_x^2 + a_y^2} = |(a_x, a_y)|$.

From this we can see that the back-projection reconstruction (using infinite measurements) is the original function convolved with the function $\frac{1}{|(a_x, a_y)|}$.

If we only take measurements between angles ξ and $\xi + a$ the back-projection looks different. Let $\delta_{\xi,a}(t\cos(\theta), t\sin(\theta))$ be a characteristic function that has the value 1

when θ is between ξ and $\xi + a$ and 0 otherwise. Now

$$
\begin{aligned}
f_{BP(\xi,a)}f(x,y) &= \int_{\xi}^{\xi+a}\int_{-\infty}^{\infty} f((x,y)+(t\cos(\theta),t\sin(\theta)))dtd\theta \\
&= \int_{0}^{2\pi}\int_{-\infty}^{\infty} \delta_{\xi,a}(t\cos(\theta),t\sin(\theta))f((x,y)+(t\cos(\theta),t\sin(\theta)))dtd\theta \\
&= \int_{0}^{2\pi}\int_{0}^{\infty} \delta_{\xi,a}(t\cos(\theta),t\sin(\theta))\frac{f(x,y)+(t\cos(\theta),t\sin(\theta)))}{t}tdtd\theta \\
&\text{Change of variables: } a_x = t\cos(\theta), a_y = t\sin(\theta) \Longrightarrow t = |(a_x,a_y)| \\
&= \int_{R^2} \delta_{\xi,a}((a_x,a_y))\frac{f((x,y)+(a_x,a_y))}{|(a_x,a_y)|}da_xda_y \;||\; (a_x,a_y) = (a_x,a_y)-(x,y) \\
&= \int_{R^2} \delta_{\xi,a}((a_x,a_y)-(x,y))\frac{f(a_x,a_y)}{|(a_x,a_y)-(x,y)|}da_xda_y \\
&= \int_{R^2} \delta_{\xi,a}((x,y)-(a_x,a_y))\frac{f(a_x,a_y)}{|(x,y)-(a_x,a_y)|}da_xda_y \\
&= (f(a_x,a_y) * \frac{\delta_{\xi,a}(a_x,a_y))}{|(a_x,a_y)|})(x,y).
\end{aligned}
$$

In the limited angle case of tomosynthesis back-projection reconstruction is equivalent to the convolution of the original function with a sort of hourglass-shaped function. This function has values inversely proportional to the distance from the center in the hourglass-shaped area (see the left side of figure 15) and is 0-valued elsewhere. The presence of convolution makes a neural network capable of deconvolution appropriate for the task of deblurring the image. In practice with finite measurements the kernel consists of individual lines in the bowtie-shaped area (see the right side of figure 15).

4 GSAA-Deconvolution network

4.1 Generalized shift-and-add (GSAA) network

We recall that our aim is to construct an end-to-end network that consists of two main parts. A convolutional neural network and a deconvolutional neural network. The latter will be described in the following section.

The first part of the combined reconstruction network uses 64 13x41 convolution kernels to convert the sinogram data into an intermediary reconstruction. The sinogram is convolved with each kernel, and the results are concatenated. The structure of the network makes it possible for it to learn the shift-and-add reconstruction, but also something else if the optimization process so chooses. Hence we refer to this part of the full network as a generalized shift-and-add network or GSAA.

To test if the shift-and-add algorithm can be learned by this part of the network, it was trained independently with a goal of generating the shift-and-add reconstruc-

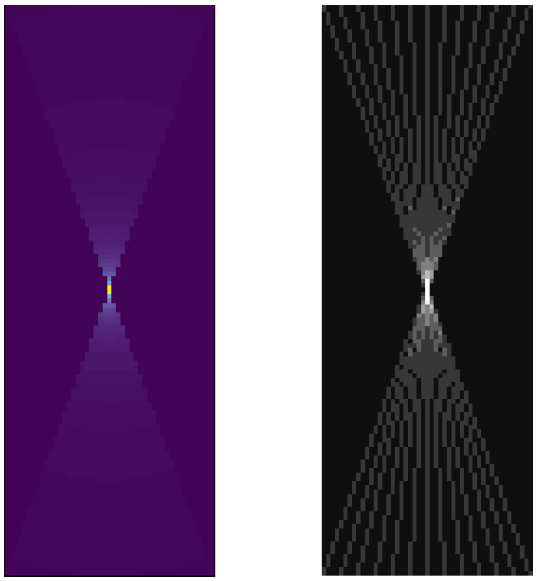

Figure 15: Left: hourglass kernel corresponding to continuous measurements. Enhanced for visual clarity. Right: discrete version

tions. The results replicate the target reconstruction very accurately. Some of the learned kernels can be seen in figure 16 and the results are compared in figure 17.

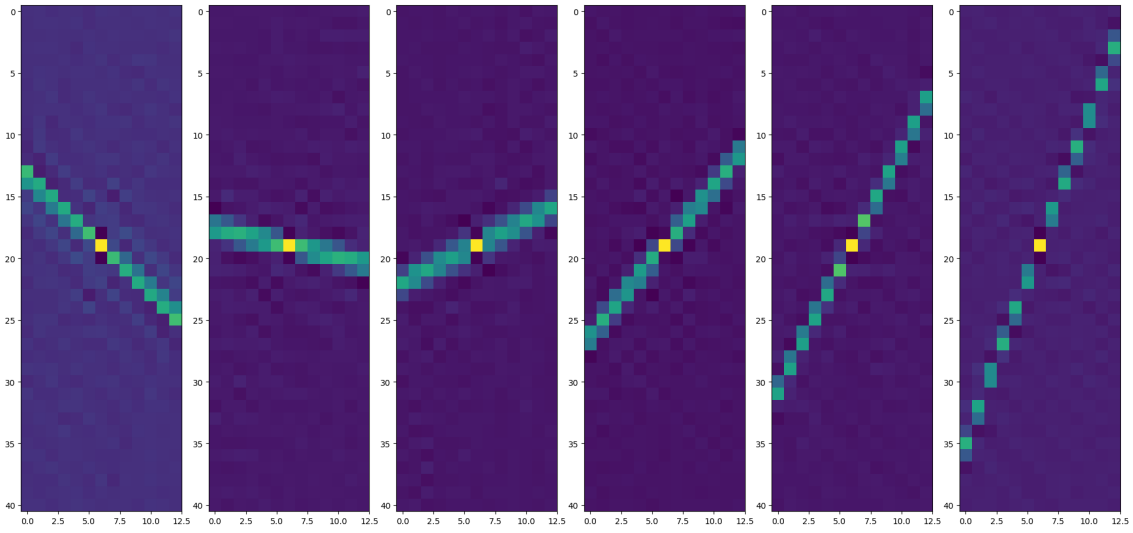

Figure 16: Kernels 1, 13, 25, 37, 49 and 61 out of the 64 kernels learned by the network after 20 epochs of training. Each one shifts and adds the layers of a sinogram by a different amount to form one layer of the reconstruction.

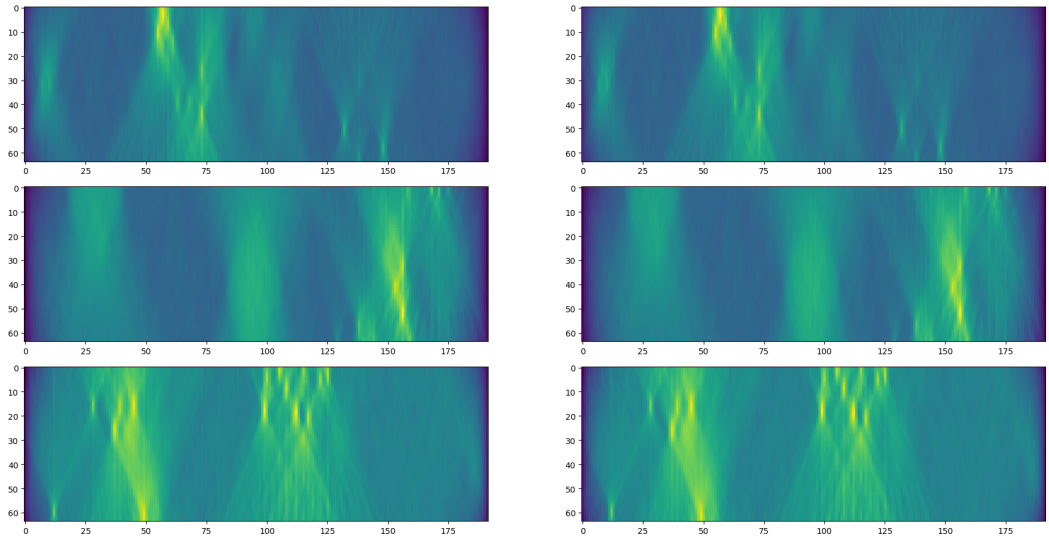

Figure 17: Left: Reconstructions generated by the GSAA network. Right: Target back-projection reconstructions

4.2 Deconvolution network

Since convolution with the hourglass kernel (fig. 15) spreads the information of a single pixel onto an area shaped like the kernel, it might be good for the deconvolution network's kernels to cover that same area in order to recover the information. For this purpose, either the individual kernels have to be big enough, or a combination of several layers' kernels' reach has to be big enough. We use multiple layers with 5, 20, 100 and 400 kernels respectively. The kernels' sizes alternate between 1x41 and 41x1. Horizontal and vertical one-dimensional kernels are used in successive layers in order to mimic the functionality of a 41x41 kernel, while requiring much fewer parameters. However, this comes at the cost of not being able to form every kind of kernel matrix by using the combinations. As the last layer is very large - 400 feature layers - we use depthwise separable convolution to combine all 400 results into one. This network was trained and tested by feeding it unfiltered back-projection reconstructions, which are equivalent to shift-and-add reconstructions (see Appendix 1). Targets were the original phantoms, and the network's outputs were quite close to them (see figure 18). ReLU, defined as

$$g(x) = \max(0, x),$$

was used as the activation function for layers and maxpooling was used between every other layer. Maxpooling is a downsampling method which reduces a dimensionality of an image by replacing a subregion of it with the maximum value in the region. The network was trained for 25 epochs.

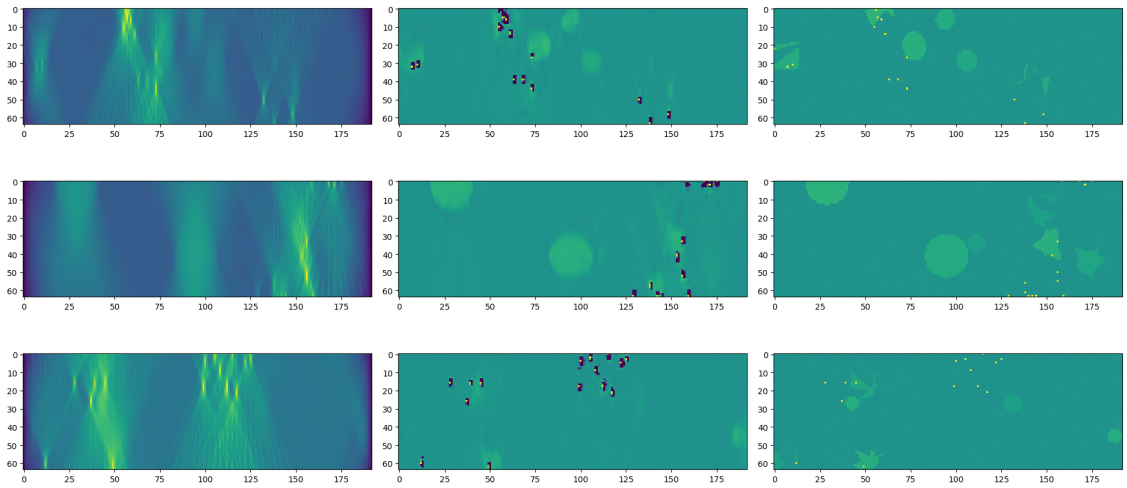

Figure 18: Results from the deconvolution network. Left: input shift-and-add reconstructions Middle: Reconstructions Right: Target phantoms. The black spots are zero-values, seemingly a side-effect of the network's focus on reconstructing microcalcifications.

4.3 Combined reconstruction network

The reconstruction network is composed of the GSAA and Deconvolution networks joined together sequentially. The intermediary output from GSAA-network is given as input to the deconvolution network, which outputs the final reconstruction. In addition there is a residual connection added: the shift-and-add reconstruction is added to the final output of the Deconvolutional part right before being fed to a last ReLU activation. In theory this should make it easier for the network to learn the transformation, since the shift-and-add reconstruction and target phantom are relatively close to each other and with this the network starts training from something resembling an identity mapping, instead of a random one.

All networks use a custom loss function called $f_{L1.5}$, with behavior resembling both L1 and L2 loss functions.

$$f_{L1.5}(x) = \sqrt{x^2 + 1} - 1$$

The function $f_{L1.5}(x)$ approaches $\frac{1}{2}x^2$ as x gets closer to 0. As x gets further from 0, $f_{L1.5}(x)$ approaches $|x| - 1$ (see figure 19 for reference). This function was chosen because similarly to L1 it de-emphasizes the large errors caused by calcifications, which were not as important to reconstruct in this work. In addition it acts similarly to L2 for small errors.

The network uses the Adam-optimizer [20] and it was trained for 60 epochs. Results can be seen in 27.

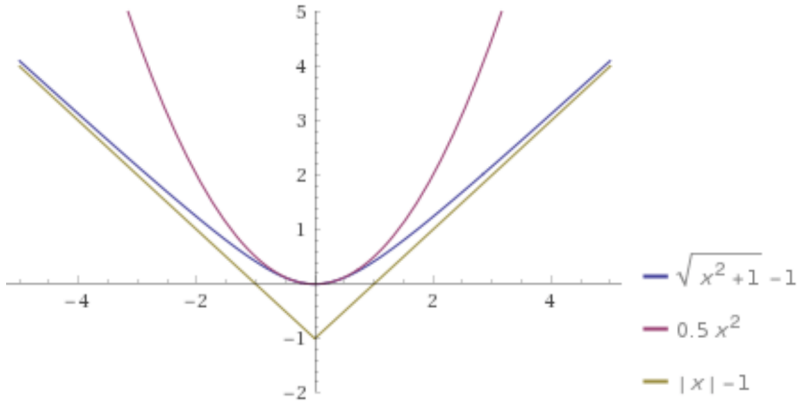

Figure 19: $f_{L1.5}$ compared to other functions. Image courtesy of Wolfram|Alpha.

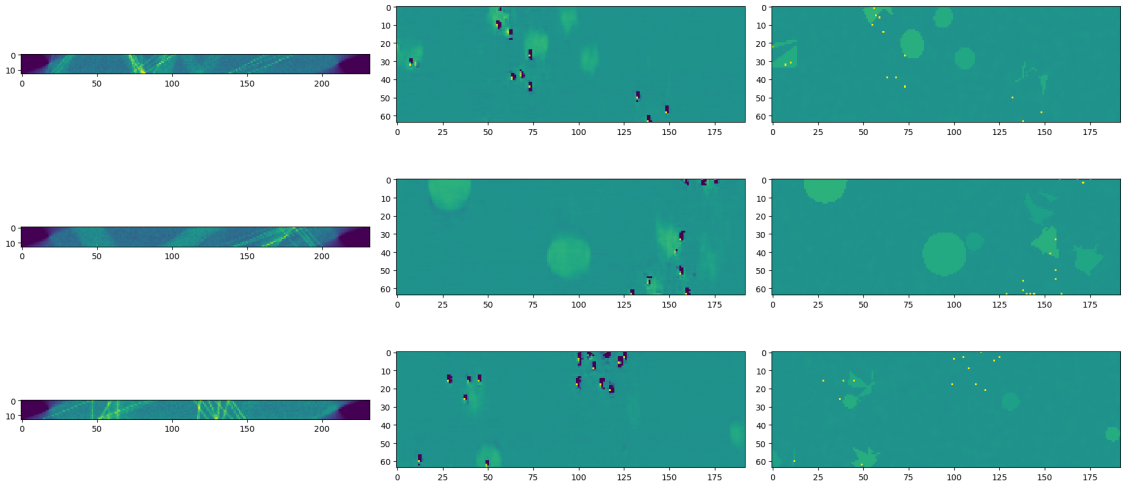

Figure 20: Results from the combined GSAA-Deconvolution network. Left: input measurements Middle: Reconstructions Right: Target phantoms

5 Acquiring data

5.1 Simulated breast phantoms

To test our reconstruciton algorithms we have constructed some simplified two dimensional breast phantoms. Their general rectangular share roughly approximates a slice of a compressed breast in a tomosynthesis mesurement device. The benign masses are represented by random and smooth ellipsoids, calcifications by cluster of individual bright pixels and finally the malignant masses generated by overlapping 3 random triangles. Otherwise the breast is assumed to be compose entirely of adipose tissue of constant attenuation. In total 1000 phantoms were generated for training the networks, and 200 for validating.

Appropriate photon energies used in tomosynthesis are 15-25 keV. At 20 keV, the

mass attenuation coefficient for adipose (fatty) tissue is 0.5677 cm^2/g [21]. Adipose tissue has a density of 0.9196 g/cm^3 [22]. With that we get that the attenuation coefficient for adipose tissue is **0.5221** cm^{-1}. According to another source ([23]) the attenuation coefficient for adipose tissue is 0.488 cm^{-1} at 19 keV. From the same source, tumor tissue's attenuation coefficient at 19 keV is **0.920** cm^{-1} and fibrous tissue's coefficient is not significantly different from that of tumor tissue. In order to simplify the problem we have opted to not include any fibrous tissue in the simulated phantoms. The mass attenuation coefficient of calcium at 20 keV is 13.06 cm^2/g [21] and its density 1.55 g/cm^3 so it has an attenuation coefficient of **20.243** cm^{-1}.

With these numbers in mind, we will use average attenuation coefficients of 0.5, 1, 1 and 20, for adipose tissue, benign masses, malignant masses, and calcifications respectively, to simulate each of the breast phantoms. Some examples of these phantoms can be seen in figure 21.

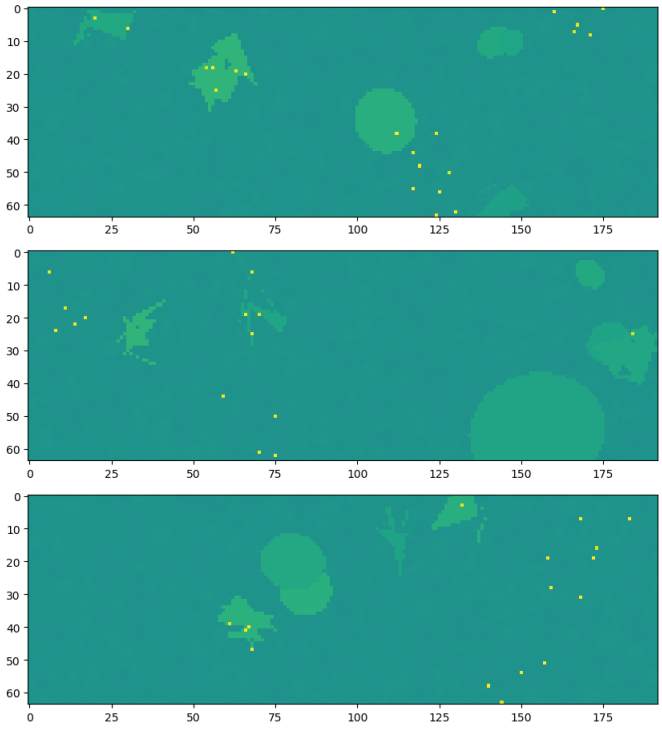

Figure 21: Some examples of generated 2D phantoms. Ellipsoidal benign masses, irregular malignant masses, and calcification clusters can be seen.

Figure 22: An example sinogram of a simulated phantom, with each row being one measurement from 13 evenly distributed angles within a field of view of 40 degrees.

5.2 Measuring the simulated phantoms

5.2.1 Changing the measurement geometry

By default, Python's scikit-image's radon-function measures an image by rotating it and taking line integrals (sums) along parallel lines, with the "detector" being perpendicular to the lines at all times. This results in a sinogram such as the one seen in figure 22. However, in most tomosynthesis setups the detector is instead fixed with respect to the phantom, and does not stay perpendicular to the rays.

To change this to the realistic geometry, we will calculate the differences between the measurements of radon-function and tomosynthesis geometry, and modify the measurements accordingly. In figure 23, $|x_p|$ is the horizontal distance from the center of the phantom to the point on the Python geometry's detector that the perpendicular ray through point p' intersects. Distance $|x_r|$ shows where the ray hits the realistic geometry's detector, when the detector moves with the phantom. By finding out the differences between these, we can modify the measurements.

From figure 23, we get

$$\begin{aligned} \sin(\alpha) &= \frac{|x_1|}{\frac{h}{2}} \\ \iff |x_1| &= \frac{h}{2}\sin(\alpha), \text{ and} \\ \cos(\alpha) &= \frac{|x_2|}{|x_r|} \\ \iff |x_2| &= |x_r|\cos(\alpha) \end{aligned}$$

and with that we get

$$\begin{aligned} |x_p| &= |x_1| + |x_2| = \frac{h}{2}\sin(\alpha) + |x_r|\cos(\alpha) \\ \iff |x_r| &= \frac{|x_p| - \frac{h}{2}\sin(\alpha)}{\cos(\alpha)}. \end{aligned}$$

With some inspection of the signs of the variables and different cases we get

$$x_r = \frac{x_p - \frac{h}{2}\sin(\alpha)}{\cos(\alpha)}.$$

Each phantom was measured from 13 different angles between 70 and 110 degrees. Before measuring, the phantoms were upscaled by a factor of two to increase accuracy, using nearest neighbour sampling. This also meant each measurement had twice as many values. The higher-detail measurements were then shifted to be equivalent with the correct geometry, and finally downscaled by two, using linear interpolation.

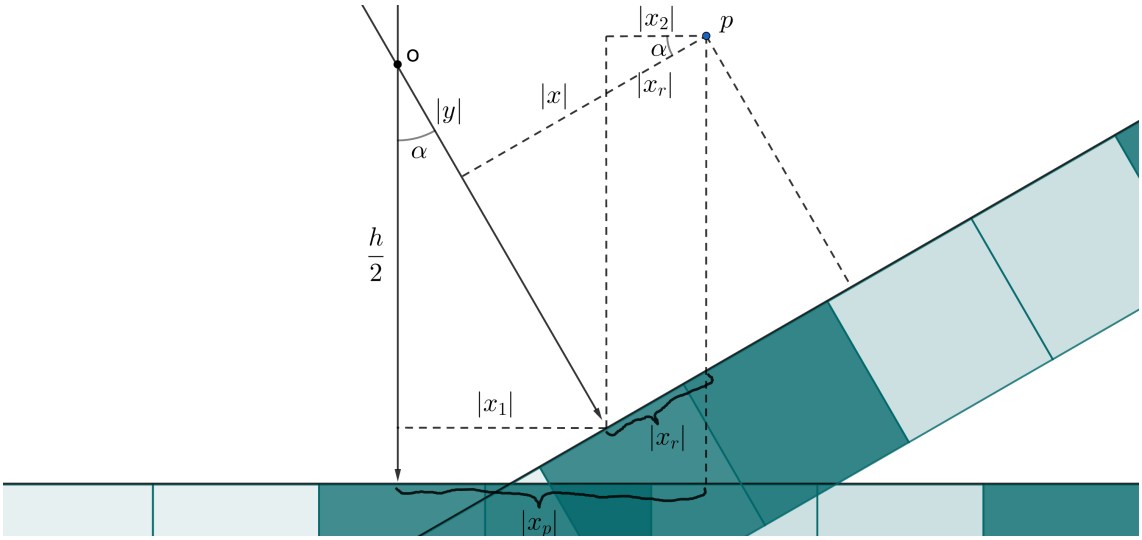

Figure 23: Finding out the conversion formula between x_r and x_p.

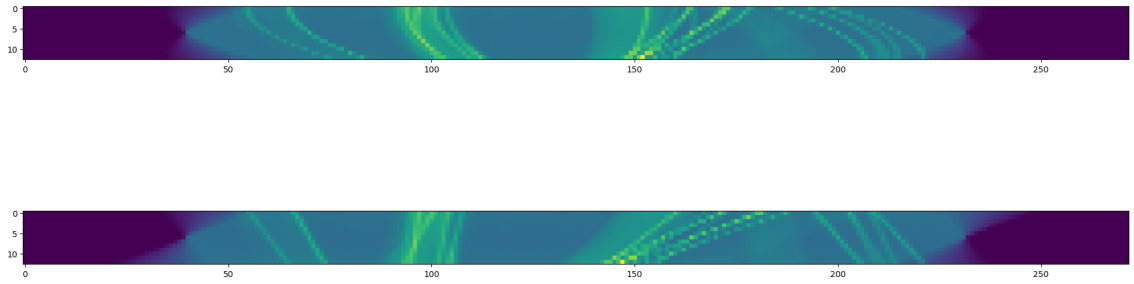

Figure 24: Comparison of sinograms with Python's and tomosynthesis geometries. The many curves transform into lines, which fits into how shift-and-add will shift the layers linearly in order to align the details.

6 Numerical results

The following results were obtained by running python 3.7 code on a laptop using the Community Edition of PyCharm2018.3.7. The laptop has a Windows 10 operating system and a built-in NVIDIA GeForce 940MX GPU with CUDA 10.0. We have already covered the specifics of how the data has been acquired in the previous section 5. In this section, we only present the main results.

Earlier in this thesis, we have already demonstrated results for the two parts of the network working independently, see figure 17 and 18.

The difference between the kernels generated by the GSAA part in the combined network (figure 25) and those generated by the independent GSAA network (figure 16) is curious: Kernel values for the combined network are much larger (roughly between 0.1 to 1 for the combined network compared to between 0 and 0.03 for GSAA network), but also less focused on the central line. Perhaps with more training the values would align more tightly on the central line, but the difference in magnitudes

might be explained simply by scaling of values inside the networks.

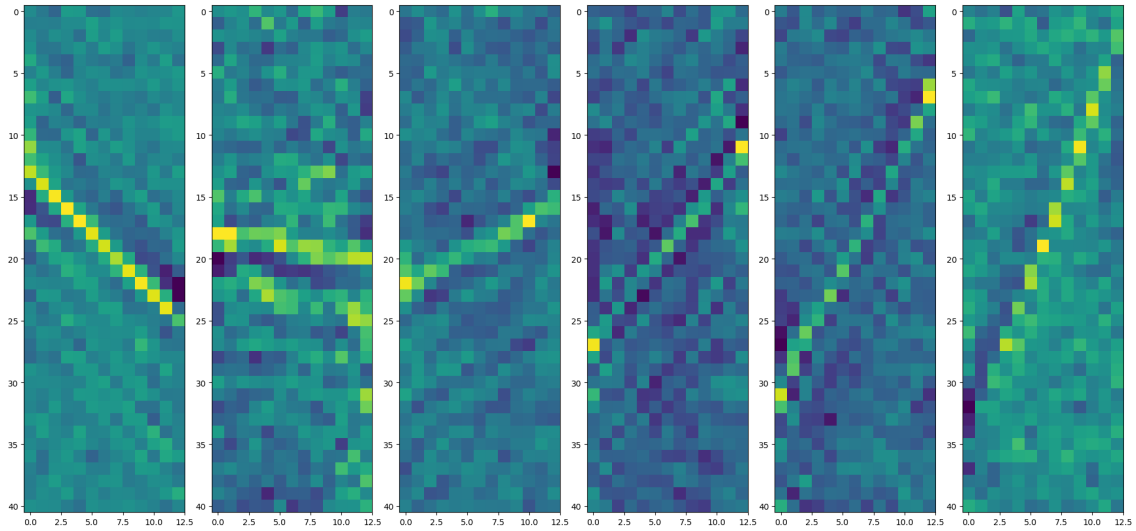

Figure 25: Kernels 1, 13, 25, 37, 49 and 61 of the 64 kernels learned by the GSAA part of the GSAA-Deconvolution network.

It seems that the combined network is able to capture the general shape of the kernels but it may be overwhelmed with the task of both reconstructing and post-processing the data. All of the reconstruction results are combined in figure 26.

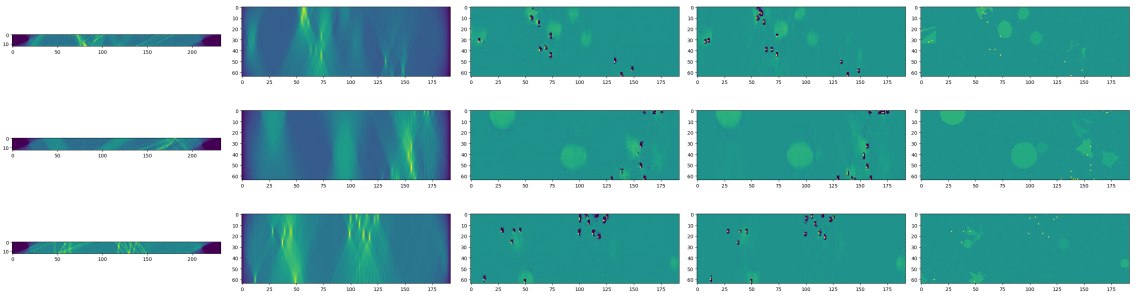

Figure 26: From left to right: 1) sinogram input into GSAA and the GSAA-Deconvolution network; 2) The shift-and-add reconstruction or the GSAA reconstruction (see figure 17); 3) The reconstruction output from the combined GSAA-D network; 4) The reconstruction output from the separately trained Deconvolution part and finally 5) the target phantom.

Results of only the combined network are visualised in figure 27. We can also observe the training and validation losses for this network in figure 28. For numerical comparison of the results, refer to table 1. Before comparisons, the phantoms' values were scaled logarithmically to be between 0 and 1 to reduce the effect of the high-attenuating calcifications on the errors. This is the same scaling as has been used in the visualizations. However since L1-error weighs calcifications less, and it was used to train the networks, scaling was not used before comparing phantoms using it.

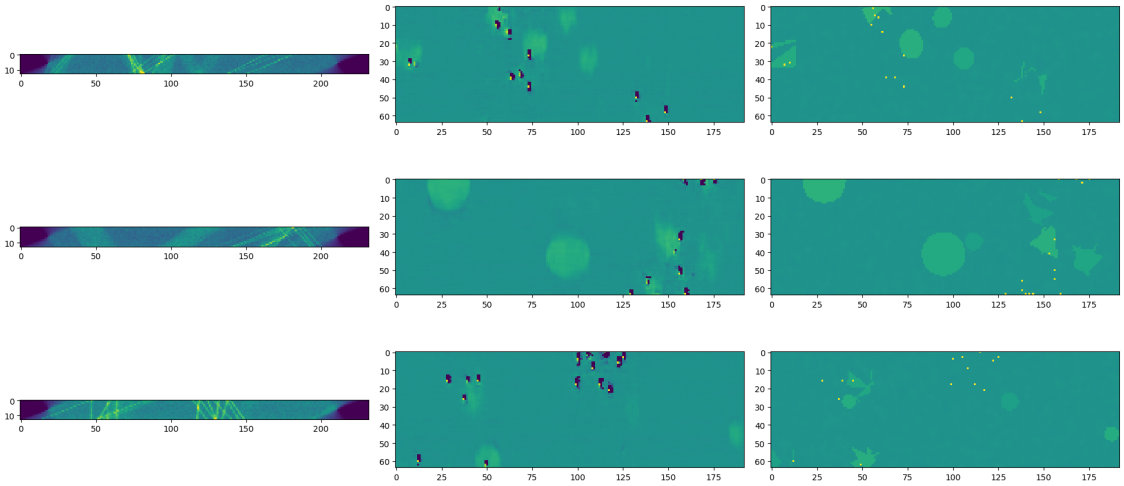

Figure 27: Left: input sinograms. Middle: reconstructions learned by the GSAA-Deconvolution network. Right: target phantoms.

Table 1: Comparing phantom reconstructions by various methods to target phantoms using different metrics. $L_{1.5}$ is the error metric used to train the networks. "Normalised SAA recon." refers to reconstructions obtained using the Shift-and-add algorithm. Due to their values being of different magnitude compared to the target phantoms, they were normalised to have the same mean and standard deviation as the phantoms on average before comparing. "Deconvolution" denotes the results by the Deconvolutional network introduced in chapter 4.2 and "GSAA-Deconv." refers to ones by the combined reconstruction network. SSIM is the Structural Similarity Index[24] and PSNR is the peak signal-to-noise ratio. Values are averages over the validation set. In blue are the best scores for each metric. In red are the worst.

	L1	L1.5	L2	SSIM	PSNR
Normalised SAA recon.	0.55278	0.21493	0.97354	0.20169	48.26550
Filtered back-proj. recon.	0.51314	0.18392	0.84422	0.06884	48.92250
Deconvolution	0.05860	0.01438	0.12474	0.80917	58.11822
GSAA-Deconv.	0.07113	0.02181	0.27037	0.78678	54.23360

We can see that while visually the separate networks may do an overall better job due to less artifacts, numerically the combined network fairs quite well. This is partially due to how well it reconstructs calcifications which are highly attenuating but do not give us too much useful information in terms of potential diagnosis

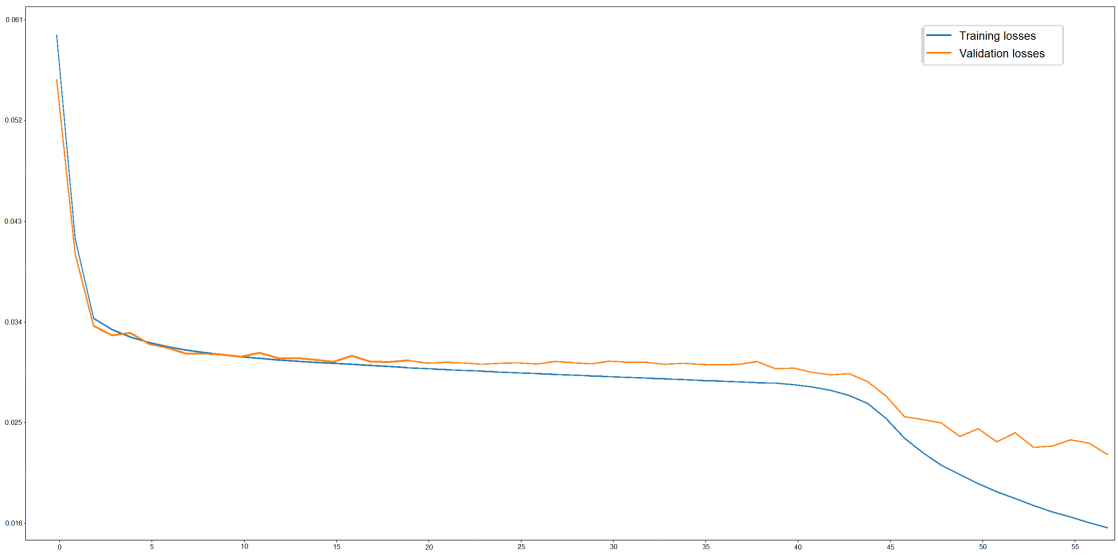

Figure 28: NEW PLOT! Training and validation losses for the GSAA-Deconvolution network. After 40 epochs of training, the step size was made bigger which seemed to enable gradient descent to find a better group of solutions. Some overfitting can be seen towards the end of the training.

7 Comparison to other reconstruction methods

What do I want in this section?? Classes of reconstruction techniques: - Back-projection based - Iterative Some comparison between the two, and similarities to my approach. Some simple implementations, comparison of results?

Most commonly used methods for reconstructing tomosynthesis images are either iterative schemes, or are based on back-projection. If back-projection is used, some kind of filtering is often done to the fourier transform of the data before or after back-projecting. Alternatively the back-projection may be post-processed using iterative or other methods. The GSAA-Deconvolution network essentially learned back-projection as a reconstruction method, and then some kind of post-processing using knowledge about the type of data it was taught to output.

7.1 Back-projection based techniques

High-pass filter is common, slightly different ones have been tried to force a uniform depth response over frequencies.

Trying to solve for and get rid of the blur from adjacent planes has been done.

7.1.1 Frequency filtering

Perhaps the most common approach to computed tomography is using the simple back-projection algorithm as basis for reconstruction. The algorithm is usually augmented with different filters, which can be applied to the fourier transform of the back-projected image. Although for greater computational speed and the same outcome, filters are instead often applied to the measurement data before back-projecting.

Since back-projection produces a blurred version of the original image, high-pass filters can be used to deblur the image by de-emphasizing low frequency components of the signal and emphasizing the high-frequency ones. However this often has issues, since realistically there is always noise in the signal that also gets amplified when using a high-pass filter. Any artifacts arising from incomplete data can be magnified too. The most basic, v-shaped "ramp" filter can be seen in figure 29, together with modified filters that try to reduce the amplification of noise by emphasizing the highest frequencies less (c-e), or try to retain more of the lower-frequency information in order for the image to be less "flat" (a) [25].

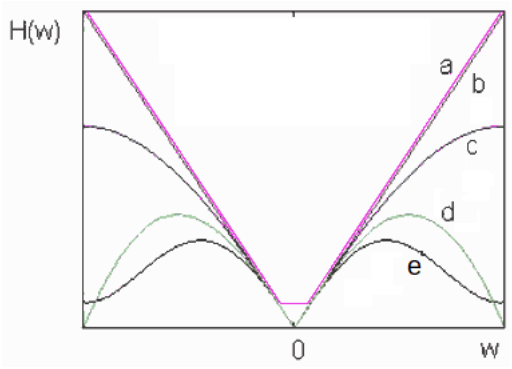

Figure 29: (a) Zero-avoidance filter (b) Ramp filter (c) Shepp-Logan Filter (d) Cosine Filter (e) Hamming filter. H(w) is the filter's value and w is the corresponding frequency. Image originally from [26], modified

7.1.2 Post-processing using Total Variation and Tikhonov regularisation

Another approach to processing measurement data after back-projecting it is regularization, which is used in underdetermined systems resulting from not having enough measurement data for a uniquely determined solution. It is also used when some knowledge about the solution is desired to be incorporated into the solving process. Total variation regularization attempts to minimize the L1-norm of the solution's gradient. This means that the result will have fewer jumps in values, which can be useful if the desired solution is piecewise constant. It can also be used to get rid of noise or artifacts. Basic Tikhonov regularisation (also called L2-regularization) gives a solution with a small L2-norm, but can also be modified to

for example give solutions with smoothly changing values. Examples of using these are shown in section 7.2.4

7.2 Iterative reconstruction methods

In this section we define some basic iterative reconstruction methods for solving tomographic problems. The following formulations have been largely borrowed from Colsher's article [27].

For these formulations, approximate a volume to be imaged as being represented by a density function $D(i, j, k)$, where $i = 1, ..., I$; $j = 1, ..., J$ and $k = -K, ..., K$. Each index corresponds to a cubical voxel within which the density is assumed to be constant, and the center plane has indices where $k = 0$ (see figure 31). By taking N measurements of size LxM we get a discrete function of projections $P(l, m, n)$ with $l = 1, .., L$, $m = 1, ..., M$ and $n = 1, ..., N$, with each pixel corresponding to a measurement ray. The projections are assumed to be parallel to the planes indicated by k. The indices of each projection are assumed to correspond to the middle layer of the object such that $i = l$ and $j = m$ for $k = 0$. The displacements of coordinates in the projections are determined by the following equations.

$$l = [i - k \tan\theta_n \cos\phi_n]$$
$$m = [j - k \tan\theta_n \sin\phi_n],$$

where [ ] signifies rounding to the nearest integer, and θ_n and ϕ_n are the tomographic and displacement angles, respectively (see figure 30). In tomosynthesis geometry the displacement angle $\phi_n = 0$ and tomographic angles θ_n are equally spaced between $-\theta_{max}$ and θ_{max}.

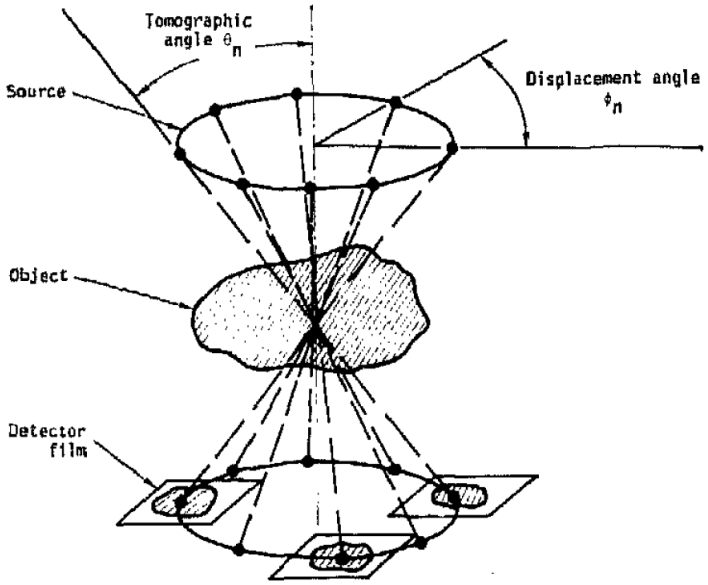

Figure 30: Displacement angle ϕ_n and tomographic angle θ_n visualized. [27]

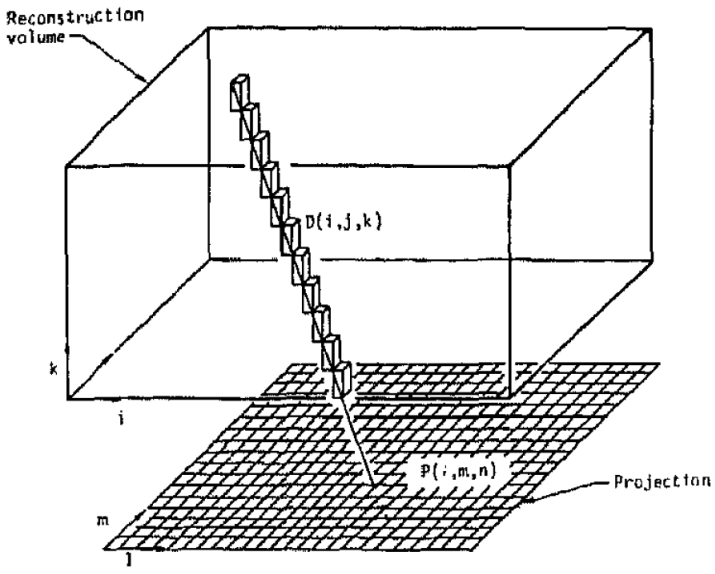

Figure 31: Representation of the discretized volume and a corresponding projection. [27]

The projection pixel values are given by

$$P(l,m,n) = \sum_i \sum_j \sum_k W(i,j,k,l,m,n) D(i,j,k),$$

where W denotes the weights assigned to each intersection of a given ray and voxel. Instead of calculating them exactly they are approximated by

$$W(i,j,k,l,m,n) = \delta(l, [i - k\tan\theta_n \cos\phi_n])\delta(m, [j - k\tan\theta_n \sin\phi_n]),$$

where δ is the Kronecker delta-function with $\delta(a,b) = 1$ if $a = b$ and which is zero-valued elsewhere. The [ ] again denotes integer rounding. For a given ray, all voxels with their center close to it get a weight of 1, in such a way that no two horizontally adjacent voxels are assigned weight if the ray is roughly vertical. The row of voxels can be thought to have a "thickness" of 1 (see figure 31).

With these weights the projection pixel values become

$$P(l,m,n) = \sum_{k=-K}^{K} D([l + k\tan\theta_n \cos\phi_n], [m + k\tan\theta_n \cos\phi_n], k)$$

From this we see that $2K + 1$ cells contribute to each projection pixel value, except for those rays that penetrate the object from a side and thus have a shorter journey through it. Indices corresponding to areas outside the object are ignored. This formulation is now a group of l x m x n linear equations with i x j x k unknowns. The case of many high-resolution measurements can be too large to solve directly, and thus iterative methods for finding a solution are often applied. In cases of limited measurements the system can be undetermined, in which case some kind of regularisation such as norm minimisation should be utilised to find a unique solution.

7.2.1 Algebraic Reconstruction Technique (ART)

We initialise the voxel values with something, for example 0's. Then projection pixel values are calculated as before

$$P^{q,n}(l,m,n) = \sum_{k=-K}^{K} D^{q,n}([l + k\tan\theta_n \cos_n], [m + k\tan\theta_n \sin_n], k).$$

Here q refers to the q:th iteration. The error between the projected measurement and the observed one is

$$\begin{aligned} & E^{q,n}(l,m,n) = P(l,m,n) - P^{q,n}(l,m,n). \\ \iff\ & E^{q,n}(l,m,n) + P^{q,n}(l,m,n) = P(l,m,n) \end{aligned}$$

Now if corrections are added to each voxel along a given ray such that the sum of corrections will equal $E^{q,n}(l,m,n)$, then $P^{q,n}(l,m,n) = P(l,m,n)$. The total correction value needed is distributed evenly to each voxel along the ray in the absence of any a priori assumptions about the error. Therefore the density at each voxel is changed as

$$D^{q+1,n}(i,j,k) = D^{q,n}(i,j,k) + \frac{1}{2K+1} E^{q,n}([i - k\tan\theta_n \cos_n], [j - k\tan\theta_n \sin_n], n).$$

Note that the division by $2K+1$ means that the densities are undercorrected close to the edges of the object where its intersections with rays are shorter.

The algorithm calculates all projection values for a given projection, corrects the densities, then moves on to the next projection. Since each correction makes the projection values correct for a given projection, but at the same time changes them for other projections, the algorithm never fully converges and thus needs a stopping criteria. During the procedure it can go through all projections multiple times.

7.2.2 Simultaneous Iterative Reconstruction Technique (SIRT)

SIRT is similar to ART, but the values of the voxels are modified by the averaged error over all the projections

$$D^{q+1}(i,j,k) = D^{q}(i,j,k) + \frac{1}{N'(2K+1)} \sum_{n} E^{q,n}([i - k\tan\theta_n \cos\phi_n], [j - k\tan\theta_n \sin\phi_n], n).$$

Here N' is the total number of rays passing through a given pixel at (i,j,k) and can differ from N. Dividing by $N'(2K+1)$ distributes the error (approximately) equally to all voxels.

7.2.3 Iterative Least Squares Technique (ILST)

Solving the system of linear equations can be looked at as a minimisation problem. ILST minimises the mean squared error (residual) between the most recently

calculated raysums, and the measured projections, defined at each step as

$$\Re^q = \sum_l \sum_m \sum_n (P(l,m,n) - P^q(l,m,n))^2 / \sigma^2(l,m,n).$$

Here $\sigma^2(l,m,n)$ is the variance of the noise in l,m,nth projection pixel. Noise is assumed to be additive with uniform variance.

Changes in voxel densities at each step are defined as

$$\begin{aligned}\Delta D^q(i,j,k) &= D^{q+1}(i,j,k) - D^q(i,j,k) \\ &= \frac{1}{N'} \sum_n E^q([i - k\tan\theta_n \cos\phi_n], [j - k\tan\theta_n \sin\phi_n], n),\end{aligned}$$

where $E^q(l,m,n) = P(l,m,n) - P^q(l,m,n)$.

This algorithm assigns all the error from a given ray to each voxel along it, which overcorrects. Thus a damping factor β is used, which is chosen to minimise the mean squared error between the newly calculated raysums and the projections

$$\beta = \frac{\sum_l \sum_m \sum_n (E^q(l,m,n) \sum_k \Delta D^q([l + k\tan\theta_n \cos\phi_n], [m + k\tan\theta_n \sin\phi_n], k))}{\sum_l \sum_m \sum_n (\Delta D^q([l + k\tan\theta_n \cos\phi_n], [m + k\tan\theta_n \sin\phi_n], k))^2}.$$

If the density values are desired to be nonnegative, they can be restricted by setting them to

$$D^q(i,j,k) = \max(0, D^q(i,j,k))$$

after each iteration.

7.2.4 Comparison of GSAA-Deconvolution network to TV- and L2-regularized back-projections

Neural nets have a lot of "preloaded" computation. They initially need even more time than iterative methods, but are fast to use once trained. Perhaps the biggest advantage of machine learning methods is that they incorporate detailed knowledge about the characteristics of the data they are trained on, and thus adapt to efficiently solve the problem in question. This is in comparison to "data-agnostic" methods that at most incorporate very vague knowledge about the solution, often through regularization. A common downside of machine learning models is narrowness, meaning that a model trained on one type of data would give poor results on some other type, unless it is trained again on that data.

In the following tables and figures we will take a look at results achieved by regularizing frequency filtered and unfiltered back-projections of the simulated phantoms' sinograms, with varying regularization parameters, and we compare those results to ones achieved by the GSAA-Deconvolution network. Before regularization the unfiltered back-projections were normalized to have the same mean as the corresponding phantoms, due to the back-projections' large values. This is unrealistic in that it

could not be done in a real setting, but gives the best possible starting point for the regularization algorithm.

Let f_r be the original reconstruction, μ_{TV} and μ_{L2} regularization parameters and $|\cdot|$ and $||\cdot||$ denote L1- and L2-norms. Then the Total Variation regularization used has the formulation

$$\min_f |\nabla f| + \mu_{TV}||f - f_r||^2, \tag{11}$$

while the L2 regularization (closely related to Tikhonov) has the formulation

$$\min_f ||f - f_{0.5}||^2 + \mu_{L2}||f - f_r||^2, \tag{12}$$

where $f_{0.5} \equiv 0.5$ and ∇x is the finite difference matrix of x.

The optimization algorithm used to find the regularization minima was the so-called Primal-dual hybrid gradient method [28][29] implemented by the ODL python library [30]. Let us first take a look at regularized filtered back-projections.

Table 2: Mean difference and similarity scores over 20 randomly sampled validation set phantoms and their corresponding total-variation regularized filtered back-projections. Numbers in the top row refer to the regularization parameter used and thus the strength of the regularization, with 0 meaning no regularization. L2 is the mean squared error, SSIM is the Structural Similarity Index[24] and PSNR is the peak signal-to-noise ratio. In blue are the best scores for each regularization parameter. In red are the worst.

	0	0.0375	0.075	0.15	0.3	0.6
L2	0.84717	0.64520	0.58205	0.53444	0.54117	0.61407
SSIM	0.07042	0.32977	0.44518	0.52359	0.52635	0.49772
PSNR	48.88722	50.07479	50.52165	50.88897	50.83116	50.27965

As can be seen from table 2, the best scores 0.53444, 0.52635 and 50.88897 achieved by total-variation regularized back-projections are worse than the 0.27037, 0.78678 and 54.23360 achieved by the GSAA-Deconvolution network. Notice that for the first score (L2) lower is better, and the other way around for the rest. This is because the mean squared error measures difference and the others measure similarity.

In the L2-regularization used, the mean squared errors between the reconstruction and an image of constant value 0.5, and between the reconstruction and the corresponding filtered back-projection were minimized. A constant image of 0.5 was chosen because that is the average density of the adipose tissue which is prevalent in the simulated phantoms.

As can be seen in table 3 and figure 33, the SSIM metric thinks that regularizing a filtered back-projection reconstruction very close to the constant function 0.5 would be the best solution. This is due to most phantoms being quite close to a constant function, only slighly differing where there are masses, and largely differing where there are calcifications. That kind of solution is however not very useful for visual

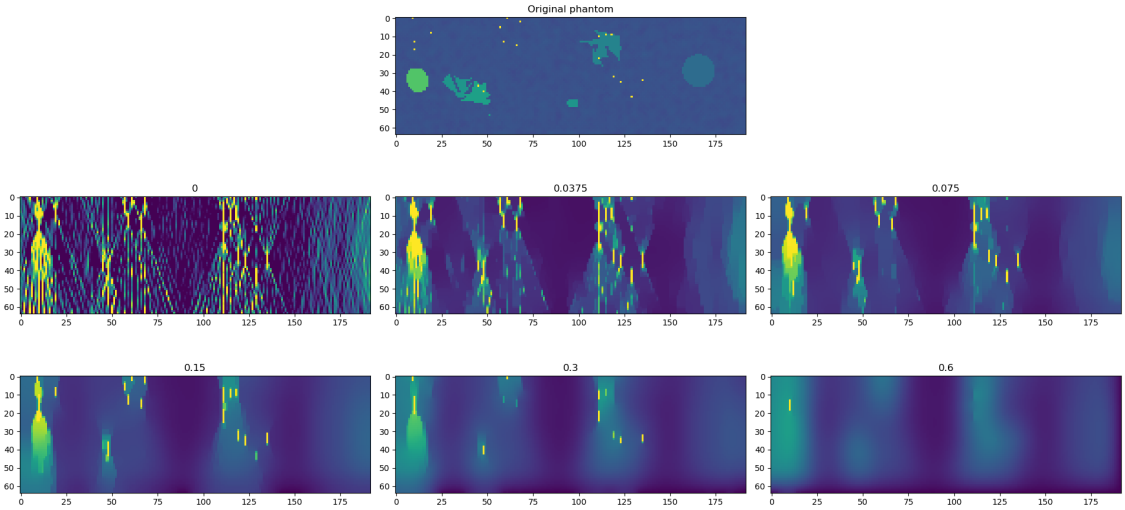

Figure 32: Visual comparison of filtered back-projections that have been TV-regularized according to equation (11). Top is the original phantom. Leftmost picture in middle row is the filtered back-projection without modifications. The last five are the back-projections with differing regularization parameter μ_{TV}. According to the numerical results, either 0.15 or 0.3 is the best parameter value.

Table 3: Mean difference and similarity scores over 20 randomly sampled validation set phantoms and their corresponding L2-regularized filtered back-projections. Numbers in the top row refer to the regularization parameter used and thus the strength of the regularization, with 0 meaning no regularization. L2 is the mean squared error, SSIM is the Structural Similarity Index[24] and PSNR is the peak signal-to-noise ratio. In blue are the best scores for each regularization parameter. In red are the worst.

	0	1.2	2.4	4.8	9.6	19.2	38.4	76.8
L2	0.8631	0.5174	0.5086	0.5301	0.5567	0.5765	0.5885	0.5952
SSIM	0.0692	0.2053	0.3158	0.4679	0.6182	0.7206	0.7690	0.7856
PSNR	48.8350	51.0633	51.1447	50.9676	50.7553	50.6032	50.5134	50.4647

determination of cancerous growths. The same can be seen happening in trying to L2-regularize unfiltered back-projections, in table 5 and figure 34.

Table 4: Mean difference and similarity scores over 20 randomly sampled validation set phantoms and their corresponding L2-regularized unfiltered back-projections.

	0	0.3	0.6	1.2	2.4	4.8	9.6	19.2
L2	0.5552	0.5576	0.5594	0.5620	0.5648	0.5672	0.5688	0.5698
SSIM	0.7328	0.7510	0.7612	0.7713	0.7780	0.7806	0.7810	0.7807
PSNR	50.6864	50.6678	50.6533	50.6333	50.6117	50.5935	50.5811	50.5737

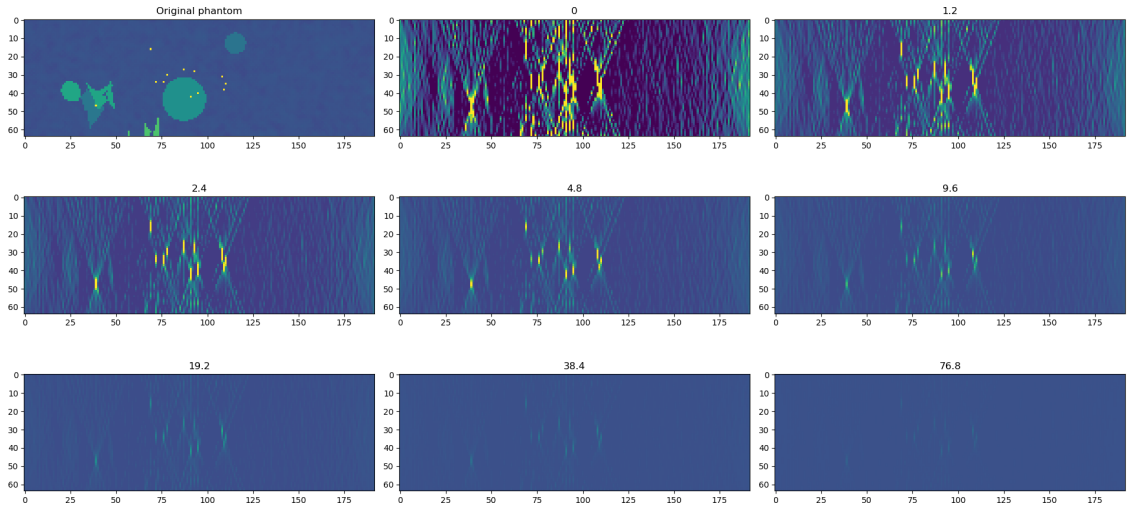

Figure 33: Visual comparison of filtered back-projections that have been L2-regularized according to equation (12). Top left is the original phantom. Top middle is the filtered back-projection without modifications. The rest are the back-projections with differing regularization parameter μ_{L2}. According to mean squared error and PSNR, 2.4 is the best parameter value.

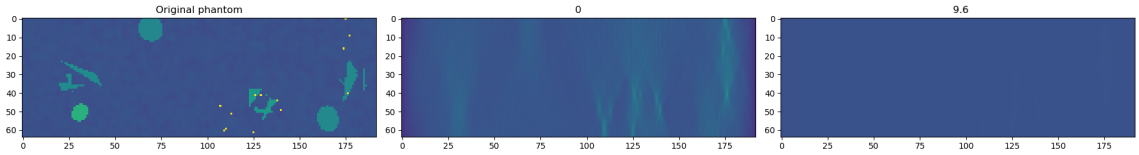

Figure 34: Visual comparison of unfiltered back-projections that have been L2-regularized according to equation (12). Left is the original phantom. Middle is the unfiltered back-projection without modifications. Right is the regularised reconstruction corresponding to the best regularisation parameter according to SSIM, 9.6.

Table 5: Mean difference and similarity scores over 20 randomly sampled validation set phantoms and their corresponding TV-regularized unfiltered back-projections.

	0	0.01875	0.0375	0.075	0.15	0.3	0.6	1.2
L2	0.5916	0.5931	0.5944	0.5968	0.5999	0.6021	0.6038	0.6056
SSIM	0.7272	0.7398	0.7430	0.7472	0.7523	0.7576	0.7622	0.7596
PSNR	50.4724	50.4615	50.4518	50.4346	50.4116	50.3960	50.3838	50.3707

In table 6 summarizing the results, it can be seen that according to SSIM the constant-valued function is an even better solution than what the GSAA-Deconvolution network produced and SSIM thus does not seem especially suitable for these comparisons. In terms of L2 error and PSNR however, the network's results are noticeably better than any of the regularizers'.

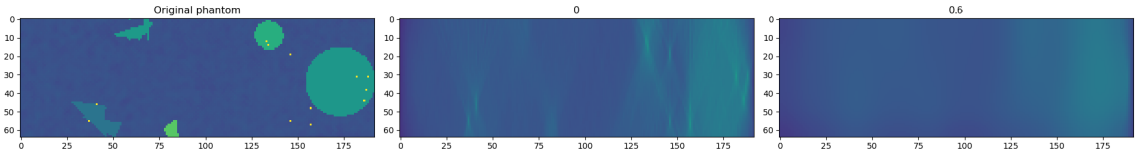

Figure 35: Visual comparison of filtered back-projections that have been TV-regularized according to equation (11). Left is the original phantom. Middle is the unfiltered back-projection without modifications. Right is the regularised reconstruction corresponding to the best regularisation parameter according to SSIM, 0.6.

Table 6: Best scores for different post-processing methods and the GSAA-Deconvolution network. Also scores for a constant 0.5-valued phantom.

	GSAA-Deconv	FBP TV	FBP L2	UFBP L2	UFBP TV	0.5-valued phantom
L2	0.2704	0.5344	0.5086	0.5552	0.5916	0.6092
SSIM	0.78678	0.5264	0.7856	0.7810	0.7622	0.7914
PSNR	54.23360	50.8890	51.1447	50.6864	50.4724	50.3545

8 Conclusion and Discussion

In this thesis we have presented an end-to-end trained neural network that takes simulated tomosynthetic data as input and outputs a full reconstruction. Our network consists of two main parts: convolutional and deconvolutional. The convolutional part of the network creates reconstructions similar to the ones generated using back-projection or shift-and-add algorithm. The second part of the network is the post-processing part that performs deconvolution, which is similar to sharpening the input. It also does some denoising. This method is chosen because the intermediary reconstruction is theoretically a convolved version of the original image (in our case, the breast phantom).

Overall, the final reconstructions of our algorithm look promising, especially considering the limited nature of the input data in both quantity and diversity. The GSAA and deconvolution networks have done great individually at reconstructing and post-processing data, as can be seen in figures 17 and 18. The GSAA-network, for example, learns kernels exactly as expected. Convolution with them essentially performs a shifting and adding operation on the images.

When combined together and trained end-to-end, the final reconstructions don't quite reach the quality of those of the deconvolution part alone. This might be because of a deeper network being harder to train due to how the weight changes propagate through the network layer by layer. However the overall results are better than what is achieved with simple back-projection and regularization.

The shift-and-add mimicking kernels learned in the combined network look like blurred versions of the kernels learned when the GSAA part was trained separately. Also some of the kernels' values are low next to the high values of the central "rays" (see figure 25). This shape resembles kernels used for image sharpening, which might indicate that as the latter part of the network tries to sharpen an image, some of that effort could be leaking into the GSAA-part.

Both the combined network and the Deconvolution network could be argued to overemphasise the reconstruction of the calcification clusters. The locations of the individual calcifications are found nearly exactly, and even their values are very close to true. However a lower level of accuracy could be sufficient for detecting breast cancer, so network capacity might be better utilized elsewhere. Especially since the process seems to generate areas of zero values around the calcifications.

What is most important for locating possible cancer is finding masses in breasts, determining their shapes and locating calcifications within masses, which can indicate malignant tumors. In terms of these goals, the network succeeds partially in determining the existence and shape of masses, struggling with small shapes but finding calcifications quite well. Since cancer is usually hoped to be found at an early stage, it is questionable how useful this method is. At least it should be a more powerful approach than non-data based ones, due to it adapting specifically to the properties of tomosynthesis measuremens.

Several adjustments can be made to improve the current results of the combined network. One obvious suggestion would be to try adding layers or kernels to the network. This could allow it to learn more varied shapes and details and lead to a better reconstruction, particularly for the deconvolution part. Another idea would be to add a residual connection to the deconvolution part, which adds the unmodified intermediary reconstruction to the final one. This makes it so the deconvolution part will need to learn the difference between the intermediary and final reconstructions, which could be easier than learning the full transformation.

The GSAA part, when trained individually, currently uses the shift-and-add reconstruction as its target. Replacing the target with a reconstruction from a different method could lead to different results. It may also be helpful to just train the two parts of the network in an alternating. Finally, it would help generalisability if more diverse data was included.

This work can also be expanded on in many different ways. As most modern X-ray CT machines image using cone-beam/fan-beam geometry rather than parallel-beam, so can our network be switched to one that takes fan-beam data as input instead of parallel-beam. It should also be fairly straightforward to scale this method to three dimensions and even try to use real patient data instead of simulated 2D slices. A potential limitation would be computational power, as the current network has been trained on a laptop's low-power GPU. With more computing capability, 3D end-to-end network for mammography could be quite achievable.

References

1 W. H. Organization, "Cancer today." http://gco.iarc.fr/today/home Accessed: 2019-11-26.

2 S. W. Duffy, L. Tabár, H.-H. Chen, M. Holmqvist, M.-F. Yen, S. Abdsalah, B. Epstein, E. Frodis, E. Ljungberg, C. Hedborg-Melander, A. Sundbom, M. Tholin, M. Wiege, A. Åkerlund, H.-M. Wu, T.-S. Tung, Y.-H. Chiu, C.-P. Chiu, C.-C. Huang, R. A. Smith, M. Rosén, M. Stenbeck, and L. Holmberg, "The impact of organized mammography service screening on breast carcinoma mortality in seven swedish counties," *Cancer*, vol. 95, no. 3, pp. 458–469, 2002.

3 L. Tabár, B. Vitak, H.-H. T. Chen, M.-F. Yen, S. W. Duffy, and R. A. Smith, "Beyond randomized controlled trials," *Cancer*, vol. 91, no. 9, pp. 1724–1731, 2001.

4 T. Sarkeala, S. Heinävaara, and A. Anttila, "Organised mammography screening reduces breast cancer mortality: A cohort study from finland," *International Journal of Cancer*, vol. 122, no. 3, pp. 614–619, 2008.

5 H. D. Nelson, R. Fu, A. Cantor, M. Pappas, M. Daeges, and L. Humphrey, "Effectiveness of Breast Cancer Screening: Systematic Review and Meta-analysis to Update the 2009 U.S. Preventive Services Task Force Recommendation," *Annals of Internal Medicine*, vol. 164, pp. 244–255, 02 2016.

6 H. D. Nelson, M. Pappas, A. Cantor, J. Griffin, M. Daeges, and L. Humphrey, "Harms of Breast Cancer Screening: Systematic Review to Update the 2009 U.S. Preventive Services Task Force Recommendation," *Annals of Internal Medicine*, vol. 164, pp. 256–267, 02 2016.

7 A. N. A. Tosteson, D. G. Fryback, C. S. Hammond, L. G. Hanna, M. R. Grove, M. Brown, Q. Wang, K. Lindfors, and E. D. Pisano, "Consequences of False-Positive Screening Mammograms," *JAMA Internal Medicine*, vol. 174, pp. 954–961, 06 2014.

8 M. Solbjør, S. Forsmo, J.-A. Skolbekken, V. Siersma, and J. Brodersen, "Psychosocial consequences among women with false-positive results after mammography screening in norway," *Scandinavian Journal of Primary Health Care*, vol. 36, pp. 1–10, 10 2018.

9 RSNA-ARC, "Mammography." https://www.radiologyinfo.org/en/info.cfm?pg=mammo Accessed: 2020-02-16.

10 J. T. Dobbins and H. P. McAdams, "Chest tomosynthesis: Technical principles and clinical update," *European Journal of Radiology*, vol. 72, no. 2, pp. 244 – 251, 2009. Digital Radiography.

11 A. P. Smith, L. Niklason, B. Ren, T. Wu, C. Ruth, and Z. Jing, "Lesion visibility in low dose tomosynthesis," in *Digital Mammography* (S. M. Astley, M. Brady, C. Rose, and R. Zwiggelaar, eds.), (Berlin, Heidelberg), pp. 160–166, Springer Berlin Heidelberg, 2006.

12 A. S. Ha, A. Y. Lee, D. S. Hippe, S.-H. S. Chou, and F. S. Chew, "Digital tomosynthesis to evaluate fracture healing: prospective comparison with radiography and ct," *American Journal of Roentgenology*, vol. 205, no. 1, pp. 136–141, 2015.

13 T. M. Svahn, *Digital Mammography and Digital Breast Tomosynthesis*, pp. 65–91. New York, NY: Springer New York, 2015.

14 S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. 1997.

15 M. Plotke, "File:3d convolution animation.gif." `https://commons.wikimedia.org/wiki/File:3D_Convolution_Animation.gif` Accessed: 2018-10-17.

16 Glosser.ca, "File:colored neural network.svg." `https://en.wikipedia.org/wiki/Artificial_neural_network` Accessed: 2018-10-17.

17 Aphex34, "File:typical cnn.png." `https://commons.wikimedia.org/wiki/File:Typical_cnn.png` Accessed: 2018-10-17.

18 S. Bazrafkan, V. V. Nieuwenhove, J. Soons, J. D. Beenhouwer, and J. Sijbers, "Deep learning based computed tomography whys and wherefores," 2019.

19 N. Moriakov, K. Michielsen, J. Adler, R. Mann, I. Sechopoulos, and J. Teuwen, "Deep learning framework for digital breast tomosynthesis reconstruction," 2018.

20 D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

21 J. H. Hubbell and S. M. Seltzer, "X-ray mass attenuation coefficients." `https://physics.nist.gov/PhysRefData/XrayMassCoef/ComTab/breast.html` Accessed: 2019-08-21.

22 M. S. Farvid, T. W. K. Ng, D. C. Chan, P. H. R. Barrett, and G. F. Watts, "Association of adiponectin and resistin with adipose tissue compartments, insulin resistance and dyslipidaemia," *Diabetes, Obesity and Metabolism*, vol. 7, no. 4, pp. 406–413, 2005.

23 R. C. Chen, R. Longo, L. Rigon, F. Zanconati, A. D. Pellegrin, F. Arfelli, D. Dreossi, R.-H. Menk, E. Vallazza, T. Q. Xiao, and E. Castelli, "Measurement of the linear attenuation coefficients of breast tissues by synchrotron radiation computed tomography," *Physics in Medicine and Biology*, vol. 55, pp. 4993–5005, aug 2010.

24 Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

25 J. Orman, T. Mertelmeier, and W. Haerer, "Adaptation of image quality using various filter setups in the filtered backprojection approach for digital breast tomosynthesis," in *Digital Mammography* (S. M. Astley, M. Brady, C. Rose, and R. Zwiggelaar, eds.), (Berlin, Heidelberg), pp. 175–182, Springer Berlin Heidelberg, 2006.

26 Y. Haider, M. Arif, N. Rahman, and M. Haseeb, "A prototype system for infrared computed tomography for image reconstruction," 12 2009.

27 J. G. Colsher, "Iterative three-dimensional image reconstruction from tomographic projections," *Comput. Graphics Image Process.; (United States)*.

28 T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the mumford-shah functional," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1133–1140, 2009.

29 A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, 05 2011.

30 "Operator discretization library documentation." `https://odlgroup.github.io/odl/` Accessed: 2020-07-29.

31 P. Sprawls, "Mammography physics and technology for effective clinical imaging." `http://www.sprawls.org/resources/MAMMO/module.htm#1` Accessed: 2019-08-21.

32 R. Diestel, *Graph Theory*. Berlin: Springer, 3rd ed., 2005.

Appendix 1. Equivalency of back-projection and shift-and-add reconstructions

Let $A = (x, y)$ be a point in a phantom, α the angle in which a ray passes through point (x, y), and s the shortest distance from origin o to that line. Origin will be in the middle of the bottom layer the phantom. The radon transform is then defined as

$$R_f(\alpha, s) = \int_{-\infty}^{\infty} f(s\langle\cos(\alpha), \sin(\alpha)\rangle + t\langle\sin(\alpha), -\cos(\alpha)\rangle)dt$$

Each value in the Radon transform space is the value of a line integral through a given measurement line specified by α and s. Vector $s\langle\cos(\alpha), \sin(\alpha)\rangle$ is perpendicular to the line, and $t\langle\sin(\alpha), -\cos(\alpha)\rangle$ is parallel to it.

In back-projection reconstruction the value of a reconstructed point is the integral over the angles of a half-circle (all points in the space)

$$\begin{aligned} f_{BP}(x, y) &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} R_f(\alpha, \langle x, y\rangle \cdot \langle\cos(\alpha) + \sin(\alpha)\rangle)d\alpha \\ &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} R_f(\alpha, (x\cos(\alpha) + y\sin(\alpha)))d\alpha \end{aligned}$$

In measuring an image using tomosynthesis geometry, the radon transform is slighly different. Let's look at a ray passing through point (x, y). If the ray is at an angle of $\alpha \in]-\frac{\pi}{2}, \frac{\pi}{2}[$, 0 being perpendicular to the detector, the point (x, y) is measured at $x_p = x + y\tan(\alpha)$ on the detector. The complete measurement is thus

$$T_f(\alpha, x_p) = \int_{-\infty}^{\infty} f(\langle x_p, 0\rangle + t\langle\sin(\alpha), -\cos(\alpha)\rangle)dt,$$

where the vector $\langle\sin(\alpha), -\cos(\alpha)\rangle$ is parallel to the measurement line.

We want to add together the points on the measurements corresponding to the same detail in the image. We now have a measurement space of points (α, x_p). By taking an integral over α in the corresponding points we get the shift-and-add reconstruction

$$f_{SAA}(x, y) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T_f(\alpha, x + y\tan(\alpha))d\alpha$$

This is equivalent to shifting the layers of the sinogram corresponding to different angles, and adding together all the overlapping points.

Now we see that

$$f_{SAA}(x,y) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T_f(\alpha, x_p) d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T_f(\alpha, x + y\tan(\alpha)) d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x + y\tan(\alpha), 0\rangle + t\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha \; ||t = t - \sin(\alpha)(x + y\tan(\alpha))$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x + y\tan(\alpha), 0\rangle + (t - \sin(\alpha)(x + y\tan(\alpha)))\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x + y\tan(\alpha), 0\rangle + (t - \sin(\alpha)x - \sin(\alpha)y\tan(\alpha))\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x + y\tan(\alpha) - x\sin^2(\alpha) - y\sin^2(\alpha)\tan(\alpha), x\sin(\alpha)\cos$$

$$+ y\sin(\alpha)\cos(\alpha)\tan(\alpha)\rangle + t\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x - x(1 - \cos^2(\alpha)) + y\tan(\alpha)(1 - \sin^2(\alpha)), x\sin(\alpha)\cos(\alpha)$$

$$+ y\sin(\alpha)\cos(\alpha)\frac{\sin(\alpha)}{\cos(\alpha)}\rangle + t\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x\cos^2(\alpha) + y\frac{\sin(\alpha)}{\cos(\alpha)}\cos^2(\alpha),$$

$$x\sin(\alpha)\cos(\alpha) + y\sin^2(\alpha)\rangle + t\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(\langle x\cos^2(\alpha) + y\sin(\alpha)\cos(\alpha), x\sin(\alpha)\cos(\alpha)$$

$$+ y\sin^2(\alpha)\rangle + t\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$=$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\infty}^{\infty} f(((x\cos(\alpha) + y\sin(\alpha))\langle \cos(\alpha), \sin(\alpha)\rangle + t\langle \sin(\alpha), -\cos(\alpha)\rangle) dt d\alpha$$

$$= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} R_f(\alpha, (x\cos(\alpha) + y\sin(\alpha)) d\alpha$$

$$= f_{BP}(x,y)$$

So the back-projected reconstruction and the shift-and-add reconstruction using tomosynthesis geometry for the measurements are the same.