December 2019

Scatter Reduction By Exploiting Behaviour of Convolutional Neural Networks in Frequency Domain

Carlos Ivan Jerez Gonzalez
University of Wisconsin-Milwaukee

SCATTER REDUCTION BY EXPLOITING BEHAVIOUR OF CONVOLUTIONAL NEURAL NETWORKS IN FREQUENCY DOMAIN

by

Carlos Ivan Jerez Gonzalez

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

in ENGINEERING

at

The University of Wisconsin-Milwaukee

December 2019

ABSTRACT

SCATTER REDUCTION BY EXPLOITING BEHAVIOUR OF CONVOLUTIONAL NEURAL NETWORKS IN FREQUENCY DOMAIN

by

Carlos Ivan Jerez Gonzalez

The University of Wisconsin-Milwaukee, 2019
Under the Supervision of Professor Jun Zhang

In X-ray imaging, scattered radiation can produce a number of artifacts that greatly undermine the image quality. There are hardware solutions, such as anti-scatter grids. However, they are costly. A software-based solution is a better option because it is cheaper and can achieve a higher scatter reduction. Most of the current software-based approaches are model-based. The main issues with them are the lack of flexibility, expressivity, and the requirement of a model. In consideration of this, we decided to apply Convolutional Neural Networks (CNNs), since they do not have any of the previously mentioned issues.

In our approach we split the image into three frequency bands: low, high low and high high and process each of them separately with a CNN. Then, we downsample the low frequency band and upsample the high frequency band, so that the frequency is increased and decreased respectively. Finally, we train three CNNs with each of the components and put them back together to have the reconstruction of the image. We demonstrate theoretically that doing this leads to better results, and provide comprehensive empirical evidence of the capability of our algorithm for doing scatter correction.

TABLE OF CONTENTS

1 Introduction **1**

2 The problem and current solutions **4**

2.1 The problem: scatter radiation 4

2.2 Current solutions: hardware 5

2.2.1 Anti-scatter grid 5

2.2.2 Air gap and scan slot 6

2.3 Current solutions: software 7

3 Theoretical Analysis **8**

3.1 Scatter radiation from a mathematical perspective 8

3.2 Decomposition of the signal in frequency bands 8

3.3 Upsampling and downsampling of the signal 10

4 Approach **11**

4.1 Convolutional Neural Network 11

4.2 Splitting the signal in frequency bands 12

4.3 Downsampling and upsampling 13

4.3.1 Downsampling 14

4.3.2 Upsampling 14

5 Experimental setup **15**

5.1 Data 15

5.1.1 Description of the data 15

5.1.2 Difficulties to obtain more data 16

5.2 CNN architecture 16

5.2.1 Coordconv 18

5.3 Training 18

5.4 Frequency splitting 20

5.5 Downsampling and upsampling 21

6 Results **22**

6.1 CNN vs Bayesian optimization 22

6.2 Analyzing loss curves of different CNNs 23

6.3 Analyzing output images of different CNNs 24

6.4 Our method applied to realistic images 25

7 Conclusions and open problems **35**

8 References **36**

LIST OF FIGURES

1 Scatter radiation vs volume of patient 5

2 Anti-scatter grid 6

3 CNN structure 11

4 Frequency components 13

5 Downsampling and upsampling 14

6 Loss for different CNNs 17

7 Implemented CNN architecture 18

8 Frequency components using realistic image 21

9 CNN vs Bayesian optimization 22

10 Loss for different frequencies 23

11 Upsampling signal for reduced loss 24

12 Splitting and upsampling CNN vs vanilla CNN 25

13 Splitting and upsampling CNN vs vanilla CNN. Zoomed in 26

14 CNN reconstruction for chest phantom 27

15 CNN reconstruction for chest phantom. Zoomed in. 29

16 Profile curves for rib cage region 30

17 Zoomed-in CNN reconstruction for chest phantom 31

18 Profile curves for spinal region 32

19 CNN reconstruction for abdominal phantom 33

20 Zoomed-in CNN reconstruction for abdominal phantom 33

21 Profile curves for abdominal phantom 34

LIST OF TABLES

1 CIF factor for ground truth and CNN reconstruction 29

1 Introduction

In X-ray imaging, scattered radiation can produce noise and a number of image artifacts, such as cupping, shadows, and decreased soft-tissue contrast [1, 2]. In practice, hardware solutions such as anti-scatter grids are often used to reduce scatter. However, the remaining scatter can still be significant and additional software-based corrections are often needed. Furthermore, good software solutions can potentially reduce the amount of anti-scatter hardware needed, thereby reducing cost.

So far, many of the approaches implemented to tackle this problem were model-based such as [3, 4] , which used Bayesian Optimization as the underlying technique. The problem with this kind of approaches is the requirement of a prior. Even though some priors such as edge-preserving smoothness, Total Variation (TV) and Markov Random Fields (MRF) have proven to be useful, they have not been enough to capture the complexity of scatter radiation. In addition, scatter radiation is not spatially invariant, for instance, in soft tissue scatter effects are less noticeable than in hard tissue (such as bones). This phenomenon is disregarded in some of the current approaches, e.g. [3, 4]. In [5], they estimate the X-ray scatter signals using Maximum Likelihood Estimation (MLE) method and kernel modeling with Monte Carlo simulation. However, their approach falls short in that, on one hand they are reducing scatter on soft tissue, which is easier and on the other hand they do not achieve a perfect reconstruction and artifacts such beam hardening are still present in their results. Additionally, they did not test their approach on realistic patient data, which could substantially affect the results. In other words, the problem was overly simplified and yet there were persistent scatter artifacts.

CNNs (convolutional neural networks) have been widely used as the-state-of-the-art in the last few years for many tasks involving images, from classification [6] to image segmentation [7]. This is mainly due to CNN's power of expressivity and due to its locally connected structure (filters), which allows to reduce the search space. As a result, it is easier to find a good solution. Then, in order to improve the results produced by the previous approaches we decided to use CNNs.

The advantages of using a CNN as opposed to the aforementioned approaches are

that, on one hand, it is highly non-linear because of the activation function, therefore they are able to model very complex mappings. On the other hand, CNNs do not require a prior. Moreover, the learning process is done with data corresponding to scatter radiation. Thus, CNNs can directly learn the underlying distribution. These two sum up the main reason the reasons why neural networks, in particular CNNs, are so powerful. Additionally, the implementation and training is relatively easy with the libraries currently available which makes CNNs even more appealing.

In our work, we apply CNNs to X-ray scatter correction like done in [7, 8]. Nonetheless, we provide additional insight and ways to obtain a better result than just applying the CNN. We notice that CNNs can manage more accurately information that is within a smaller frequency range. Based on this, we decide to split the image into different frequency bands and process each band independently, so that, the CNN can focus on a certain frequency band at a time and not the whole frequency spectrum.

In this work our main contributions are:

- Splitting the signal into different frequency bands and processing each band independently yields better results than processing the signal as a whole when using CNNs.

- For high frequency bands, one can improve the results by dilating the signal before passing it through the CNN and for low frequency bands one can subsample and then pass the subsampled version of the signal through the neural network.

A similar idea to our first contribution was implemented in [9], except they used wavelets for the frequency band splitting and they did not present any theoretical reasoning justifying why his approach worked the way it did. They only provided one plot which shows that using wavelet components produces better results than processing the entire signal as a whole. Additionally, some questions such as: is it more difficult to train on certain components of the wavelet transform or is it better if you keep breaking up the signal? remained unanswered. In our work, we go beyond just splitting in frequency domain, in that we also incorporate other techniques, such as downsampling, upmsapling,

Coordconv [10], and masking the loss.

In this paper, we present a theoretical framework in which is shown that splitting a signal into frequency bands leads to better results and upsampling the high frequency component and downsampling the frequency of such signal also leads to better results in terms of the error, that is to say, the error is further reduced. Subsequently, we describe the method where apply and demonstrate our claims in a thorough fashion, then we show experimental results and analysis confirming what was predicted by the theory.

2 The problem and current solutions

In this section we describe the problem of scatter radiation, why it occurs, how grave it can be and why it is important. Then, we provide the current solutions, which can be divided in hardware and software. We discuss for each of them why they are not viable, what their problems are and implicitly how our proposed solution is potentially better than any of the current ones.

2.1 The problem: scatter radiation

The basic principle of projection x-ray imaging is that x-rays travel in straight lines [11]. However, when x-rays interact with a material, such as the organs inside a patient, scatter radiation is produced [12], that is, some rays will be scattered, hence the straight-line assumption is violated. The scattered radiation that strikes the detector will significantly degrade the image, reduce the contrast, reduce the signal-to-noise ratio and artifacts such cupping and shadow will appear. [1].

The amount of scatter detected in an image is characterized by the scatter-to-primary ratio (SPR). The SPR is defined as the amount of energy deposited in a specific location in the detector by scattered photons, S, divided by the amount of energy deposited by primary (non-scattered) photons in that same location, P [11]. Formally put

$$\text{SPR} = \frac{S}{P} \tag{1}$$

For an SPR of 1, half of the energy on the detector at that location is from scatter, which makes 50% of the information useless. Additionally, the amount of scatter depends on the location. For denser areas, such as bones, the amount of scatter will be significantly higher than for less dense areas, such as soft-tissue. If left uncorrected, the amount of scatter on an image can be dominant. The SPR increases typically as the volume of tissues that are irradiated increases. Figure 1 illustrates the SPR for three patients with different thicknesses (10 cm, 20 cm and 30 cm). The field of view refers to the size of object that is being observed, namely a field of view of 20 cm means that an object of 20

cm X 20 cm is being viewed.

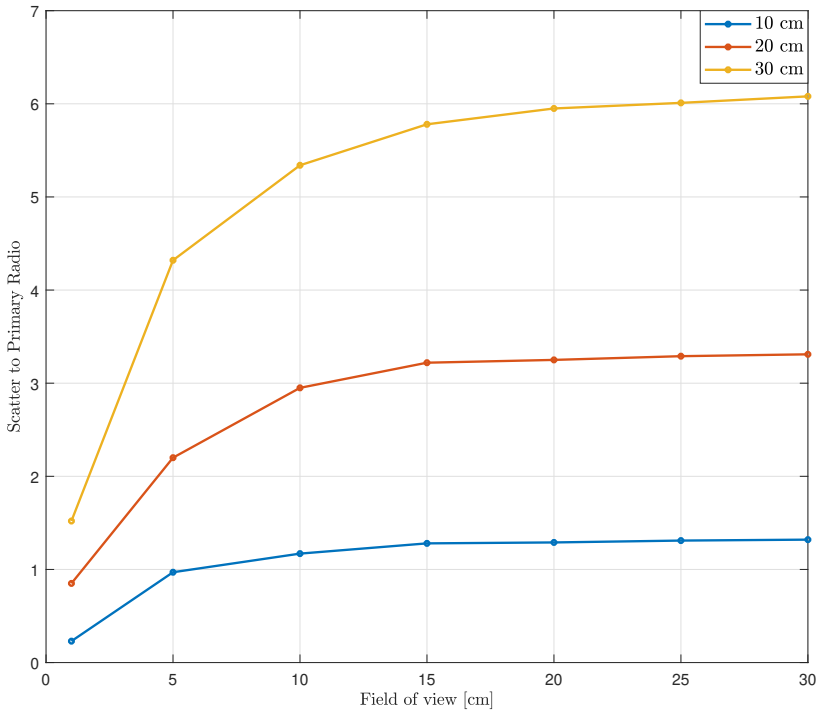


Figure 1: Scatter radiation vs volume of patient and field of view. Image taken from [11]

Portable bedside chest radiography is one of the most frequent x-ray examinations in hospitals [13]. As such, it is an important and well-established diagnostic tool for the examination of critically ill patients and indispensable for verifying correct positioning of catheters, tubes and lines, and to avoid complication due to misplacements [14]. Therefore it is of vital importance to have methods that can do scatter reduction.

2.2 Current solutions: hardware

2.2.1 Anti-scatter grid

The most widespread technology to reduce scatter radiation is the anti-scatter grid. Its functioning principle is extremely simple: Have parallel plates between patient and detector, so that only rays travelling straight will reach the detector and the scattered rays will be absorbed by the plates. Figure 2 shows the principle aforementioned.

As simple as it is, the anti-scatter grid displays a number of problems:

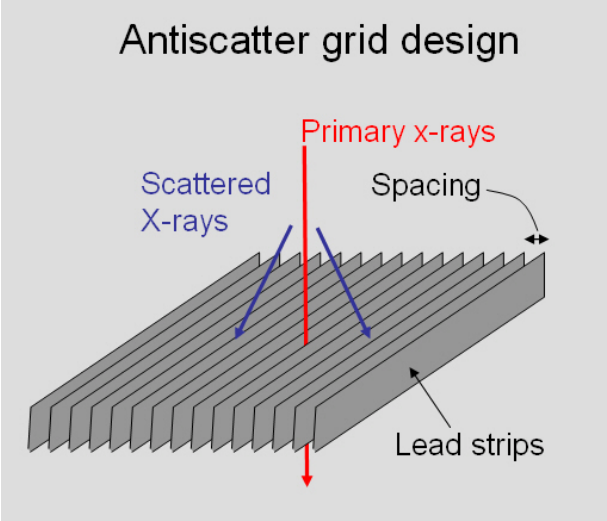


Figure 2: Anti-scatter grid design

- The grid has to be perfectly aligned with the beam, which can be troublesome to achieve, hence time-consuming.
- The interspace material would ideally be air, so that the primary rays can freely travel, but the lead septa require support for structural integrity. Hence a solid material is used, which, as a side-effect, will decrease the energy of the primary rays.
- Designing and manufacturing grids is hard because the width of each septum has to be very small. A 2048 X 1680 chest radiographic system has approximately 200 μm detector elements, and thus a grid with 45 μm wide grid septa [11].
- The above problems translate into high costs for anti-scatter grids.

2.2.2 Air gap and scan slot

The air gap technique consists of separating the patient farther away from the detector, so that scattered rays can spread away and do not reach the detector, since they are are deflected from the primary rays. However, there is an undesired magnification effect of the patient's anatomy. Practical factors limit the usefulness of the air gap method. As magnification of the patient anatomy increases, the coverage of a given detector dimension is reduced, and there is a loss in spatial resolution due to the increased blurring of the

finite focal spot with magnification.

The scan slot technique consists of imaging a very small field at a time by using a slit and a slot and then going across the image. If the field of view is small, then a small deviation of a ray will result in that ray not reaching the detector, that is the main idea. The problem with this approach is the long acquisition time. Owing to the fact that X-rays are ionizing radiation, patients should be exposed for as little time as possible and with lowest possible doses.

2.3 Current solutions: software

Many of the issues that arise from using the current software solutions were already discussed in the introduction. Here, we will discuss some of the issues that arise in the approach presented in [13], because that approach has been widely used to do scatter correction.

The approach in [13] is carried out in three steps. In the first step they use a physical model to estimate the scatter signal. Then they subtract that estimate from the detector image, where they estimate the scatter signal again. This is done until successive estimates of the scatter signal do not differ significantly. In the second step, the estimated scatter signal is matched to the scatter signal that a grid would physically remove. This is done to ensure that the physical model has the desired effect. In the third step, the estimated scatter signal from the second step is subtracted from the detector image.

The problem with the aforementioned approach is the requirement of the physical model, because the physical models available are not flawless. In others words, the approach is limited by the quality of the physical model. Therefore, relaxing this limitation could potentially lead to far superior results.

3 Theoretical Analysis

3.1 Scatter radiation from a mathematical perspective

Let x = $\{x_1, x_2, .., x_n\}$ and s = $\{s_1, s_2, ..., s_n\}$, respectively, be the "true" image and scatter, where i denotes a pixel location. In X-ray imaging, x and s are not directly observable; instead, the observed image y = y_i is a Poisson random field with x + s as mean. Specifically, each observed pixel y_i is

$$y_i \sim Poisson(x_i + s_i) \tag{2}$$

Where $\sim$ means "is a sample" from a Poisson distribution with mean $x_i + s_i$.

The goal is then to find a mapping f from y to x or at least an approximation to such mapping. When using CNNs we do not utilize the fact there is an underlying Poisson distribution, therefore we will not discuss this mathematical model further.

3.2 Decomposition of the signal in frequency bands

Here we present the theoretical foundation of the main contribution of this work. We show that the mean squared error (MSE), when estimating one random variable y, from another random variable x, is reduced if we further decompose x and disregard the components that are independent with y. This is written mathematically in proposition 1.

The proof we present is limited, in that we require the different frequency bands to be independent variables, which is not necessarily the case in practice. However, our empirical results seem to support our assumption about independence, at least, to a certain degree.

Proposition 1. *Let x_1, x_2 and y_1 be random variables with zero mean, where x_1, x_2 are independent and so y_1 and x_2. Let $\mathbb{E}[y_1|x_1] = \varphi_1(x_1)$, and $\mathbb{E}[y_1|x_1 + x_2] = \varphi(x_1 + x_2)$. Let the mean squared error (MSE) between a random variable y and an estimate of y ($\hat{y}$) be $\mathbb{E}[(y - \hat{y})^2]$. Then $\mathbb{E}[(y_1 - \varphi_1(x_1))^2] > \mathbb{E}[(y_1 - \varphi_1(x_1))^2]$.*

Proof. We will apply the orthogonality principle, which states the following: Let $\psi(u) =$

$\mathbb{E}[v|u]$, then for any "reasonable" g(u) we have:

$$\mathbb{E}[(v - \psi(u))g(u)] = 0 \tag{3}$$

First, let us consider:

$$\begin{aligned}\mathbb{E}[(y_1 - \varphi(x_1 + x_2))^2] = \mathbb{E}[(y_1 - \varphi_1(x_1))^2] + \mathbb{E}[(\varphi_1(x_1) - \varphi(x_1 + x_2))^2] \\ - 2\mathbb{E}[(y_1 - \varphi_1(x_1))(\varphi_1(x_1) - \varphi(x_1 + x_2))]\end{aligned} \tag{4}$$

Then we consider:

$$\mathbb{E}[(y_1 - \varphi(x_1))(\varphi_1(x_1) - \varphi(x_1 + x_2))] = \cancel{\mathbb{E}[(y_1 - \varphi(x_1))\varphi_1(x_1)]} - \mathbb{E}[(y_1 - \varphi(x_1))\varphi(x_1 + x_2)] \tag{5}$$

Where the first term goes to 0 by the orthogonality principle. Let's now consider the second term

$$\begin{aligned}\mathbb{E}[(y_1 - \varphi(x_1))\varphi(x_1 + x_2)] &= \mathbb{E}_{x_2}[E_{x_1,y_1|x_2}[(y_1 - \varphi_1(x_1))\varphi(x_1 + x_2)]] \\ &= E_{x_2}[\mathbb{E}_{x_1,y_1}[(y_1 - \varphi_1(x_1))\varphi(x_1 + x_2)]]\end{aligned} \tag{6}$$

x_2 is like a constraint inside the equation and $\mathbb{E}_{x_1,y_1|x_2} = \mathbb{E}_{x_1,y_1}$ because x_1 and x_2, y_1 and x_2 are independent. As a consequence, the term inside goes to 0 by the orthogonality principle and the whole expectation goes to 0.

Equation 4 becomes:

$$\mathbb{E}[(y_1 - \varphi(x_1 + x_2))^2] = \mathbb{E}[(y_1 - \varphi_1(x_1))^2] + \mathbb{E}[(\varphi_1(x_1) - \varphi(x_1 + x_2))^2] \tag{7}$$

If we assume $P[\varphi_1(x_1) \neq \varphi(x_1 + x_2)] > 0$, then

$$\mathbb{E}[\varphi_1(x_1) - \varphi(x_1 + x_2))^2] \neq 0 \tag{8}$$

Hence,

$$\mathbb{E}[(y_1 - \varphi(x_1 + x_2))^2] > \mathbb{E}[(y_1 - \varphi(x_1))^2] \tag{9}$$

□

This result is quite general as we did not make any assumption in the distribution of the random variables or the estimator. Hence, it can be applied to estimators that are not necessarily neural networks.

3.3 Upsampling and downsampling of the signal

Multigrid methods refer to a set of numerical methods whose key idea is discretization over a predefined grid. These methods are used to solve ordinary or partial differential equations, which are ubiquitous in physics and engineering. It turns out that many standard iterative methods posses the smoothing property, meaning that high frequency error is efficiently eliminated but low frequency error persists [15]. To overcome this problem one proceeds to use a coarser grid, increasing thereby the frequency of the signal and then returning to the original grid. As a result, one can reduce error in all frequencies.

We borrow this idea from multigrid methods. We observed, CNN does not perform perform as well in high frequencies as it does in low frequencies, as long as the frequency is not very low. This agrees with what was found in [16]. So, we simply upsample the image via linear interpolation, so that its frequency decreases and the CNN can output a better reconstruction. The downside, though, is that the image is bigger after interpolation, which translates to more memory required to train the CNN.

Conversely, if the image contains only low frequency information, one can downsample, the frequency is thus increased slightly. In summary, our observations were that the CNN performs the best when the frequency is neither too high nor too low. The upside of downsampling is that the image is smaller, hence less memory is required to train.

4 Approach

In this section we provide a description of three main elements of our approach. In the experiments and results section, we will provide details regarding the architecture of the CNN, the training, the splitting done in the frequency, subsampling and downsampling among other aspects.

4.1 Convolutional Neural Network

A convolutional neural network is a special case of fully connected network. Whilst in fully connected neural networks, connections are made among all neurons, in CNN connections are made only among spatially close neurons, thus enforcing the use of the correlation among nearby elements. CNNs are especially suitable for applications involving image processing or computer vision. Simply because in an image, there is a strong correlation among pixels located nearby to each other and no correlation or negligible correlation among pixels located far away from each other. Figure 3 illustrates the structure of a CNN. In that particular case, five filters are applied to the input, which is padded, so that the resulting image after applying the filter has the same size. Then the resulting image (in this case 5 images) goes through a predefined activation function. Subsequently, ten filters are applied and continues in that manner until the last layer, where one filter with a linear activation function is applied, thus generating the output. One sets arbitrary layers and arbitrary number of filters as well as filter size (3x3 or 5x5, 7x7. In practice 3x3 is typically used).

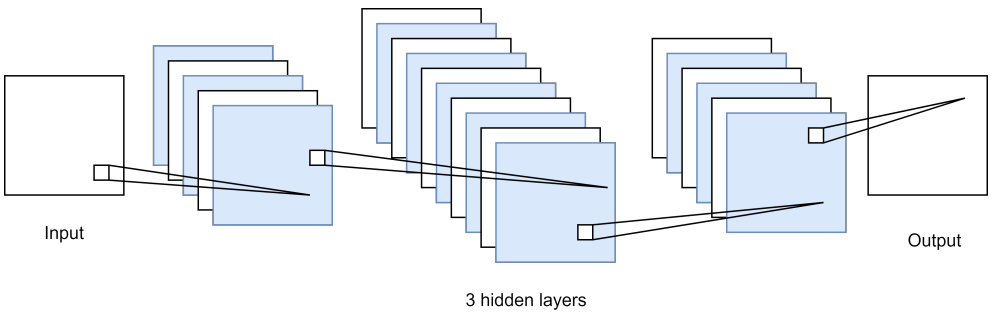


Figure 3: Convolutional neural network structure

The MSE (Mean Square Error) is used as a metric and it is minimized by learning

the right filters to map from input to output. This minimization is done by using some variant of gradient descent.

4.2 Splitting the signal in frequency bands

To separate the image into frequency bands we convolve the image with a Gaussian kernel. The low component is obtained from the convolution and the high frequency component can be obtained by:

$$H_f = I - L_f \tag{10}$$

Where H_f is the high frequency component, I is the image and L_f is the low frequency component, obtained from the convolution. The cutoff frequency can be adjusted by modifying the standard deviation in the Gaussian kernel. One can perform further splits, for instance, one can take the high frequency component and split it into two more, ending up with three frequency components.

The energy of an image is defined as:

$$E = \langle x(i,j), x(i,j) \rangle = \sum_{i,j} |x_{i,j}|^2 \tag{11}$$

When doing the split, it is important to find a cutoff frequency, such that H_f and L_f have comparable energies, otherwise, one component will overly dominate, as a result the splitting becomes essentially useless. This is hard to achieve in the first split because L_f contains the mean (DC part) of the image, and that is where a majority of the energy resides.

Figure (4) illustrates an image that has been split to three frequency components. As we can observe, the low frequency component contains more of the background details, including the mean, and it is blurry. The high low frequency and high frequency component are zero mean (the mean is contained in the low frequency component). High low frequency component contains some edges and some prominent details. On the other hand, high high frequency component contains mostly texture and any noise there is in

the image.

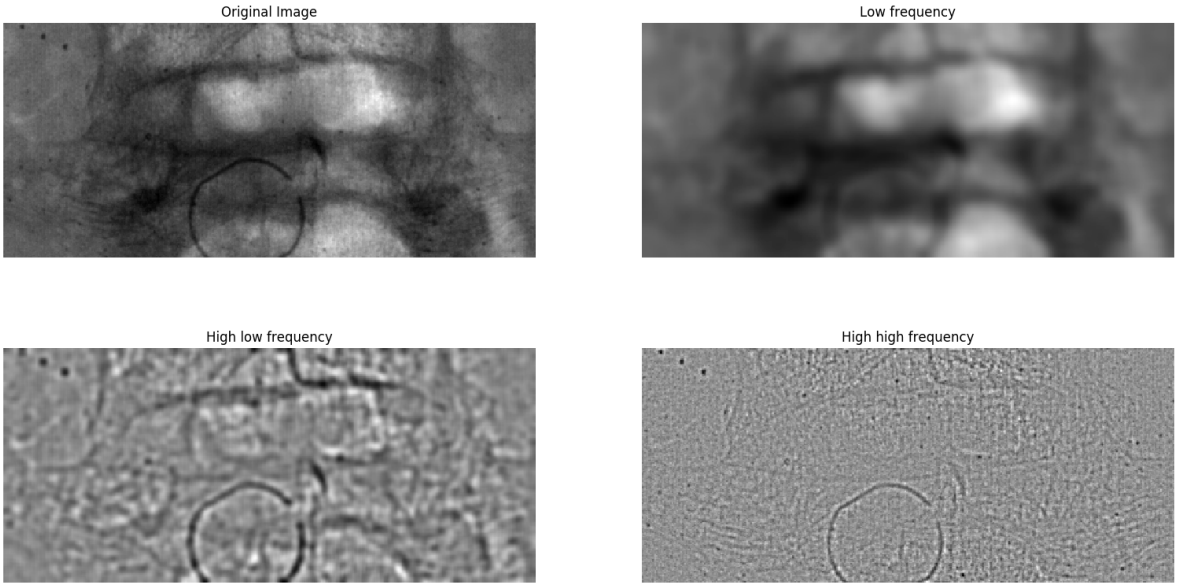


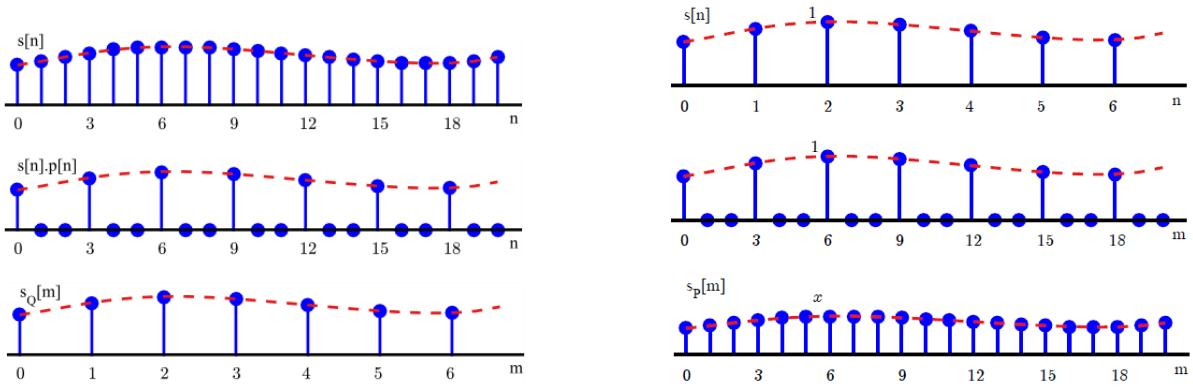
Figure 4: Image split in three frequency components

To reconstruct the original image, one adds up the frequency components. That can be easily seen from equation 10.

4.3 Downsampling and upsampling

We observed that CNN does not perform very well when the frequency of the signal is too low or too high. Since we are dealing with discrete signals, one simple method to modify the frequency is by downsampling, where one takes a low frequency signal and reduces the size of the signal, so that the frequency is increased. The other method is upsampling, where one increases the size of the signal by filling in with pixels that have "smooth" transitions in intensities for neighboring existing pixels. As a result, the frequency is reduced. Figure 5 illustrates the process of downsampling and upsampling.

These two techniques go hand in hand. If the image is downsampled and processed, it has to be subsequently upsampled, so that we recover back the original image size. The other way around is also true, that is, if the image upsampled and processed, it has to be subsequently downsampled.



(a) Downsampling

(b) Upsampling

Figure 5: Altering frequency of a discrete signal

4.3.1 Downsampling

One just takes one pixel every n pixels to downsample by a factor of n. This is done along the vertical and horizontal direction. The downsampling can nonetheless be arbitrary, namely, one can downsample along the horizontal direction by a factor of m and the vertical direction by a factor of n.

4.3.2 Upsampling

Upsampling, in this context, refers to increasing the image size. If for every two contiguous pixels one adds other $n-1$ pixels in between, the image will be upsampled by a factor of n. This is achieved through linear interpolation. So for 1D, given two contiguous pixels, we can find a line that connects them as follows:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0} \tag{12}$$

Where y refers to the intensity and x refers to the pixel location. If we want to upsample by a factor of 3, then x will be $1/3$ and $2/3$ and we can find the corresponding intensities for those locations using equation 12. Just as before, we can upsample along the vertical direction by a factor of m and along the horizontal direction by a factor of n.

5 Experimental setup

In this section, we describe the different aspects of setting up the performed experiments. This section is complemented with the results section. The code was written in python and the neural network was implemented in pytorch.

5.1 Data

The key to success for any deep learning or machine learning algorithm is the data. In [17] that is thoroughly discussed. Moreover, many of the greatest accomplishments in deep learning were partly possible due to the large amount of data available, e.g. in [18] or [19].

We acquired the data from GE Healthcare. For this project they provided us with images of four different objects. This amount of data happened to be very limited, yet our algorithm was able to produce outstanding results.

5.1.1 Description of the data

As previously mentioned, the data consisted of four objects. For each object, they provided us with a pair of images; one image was obtained using the anti-scatter grid (ground truth or target) and the other was obtained without the anti-scatter grid (scattered image or input). We use a patch-based approach, that is, we separate the image into patches and consider each of them as one image. This allows us to overcome memory issues. The four objects are described as follows:

- A circular object. It contains mostly high frequency information, especially near the center. The right half of the circle is used as training and the left half is used as testing. In addition, we can make a direct comparison with the algorithm used in [20], because they used the same image.

- A section of the spine. One part is shown in figure 4, the testing. The training was the section that was right above. As we can see, the two objects are very different. In practice, we found that despite both images from both objects were affected by

scatter radiation, it did not yield good results to train on one kind of image and test on the other.

- Thoracic region of a phantom. This was by far the most comprehensive and realistic object of the four, in that it was a bigger image and it contained different textures, details and it resembles the human thorax. We were given 12 ground truth images and 12 scattered images that were taken at different doses: 2 mAs, 3.2 mAs and 4 mAs at 110 kVp, that is 36 scattered images total. We use the image taken at 2 mAs because on one hand, it means the patient is exposed to less radiation and on the other hand, the images taken at 3.2 mAs and 4 mAs presented saturation in many regions.

- Abdominal region of a phantom. The previous description fits this image, except we were only given ground truth and scattered image with a dose of 3.2 mAs. Hence, the image is saturated in some regions. Figure 1 showed the relationship between the level of scatter the volume of the object. We observed that here, in that the scatter effects are not as noticeable as they are in the thoracic image. Therefore, this image is only used for testing.

5.1.2 Difficulties to obtain more data

The difficulty to obtain more data or more realistic data stems from the fact that any person willing to help has to be exposed to radiation multiple times so that the scattered image and the ground truth image can be obtained. That can be highly harmful to the body.

5.2 CNN architecture

[21] and [22] are rather general purpose architectures, that even though have proven to provide improvement over a vanilla neural, their performance is still surpassed by architectures designed to tackle specific problems such as [7]. [23] provides a new architecture that has state-of-the-art results for CT-reconstruction and image super-resolution.

We implement and run different architectures. We find that, at least for this problem and the data we had at our disposal, the architectures do not seem to make a difference in terms of the MSE or the images quality. This is illustrated in figure 6, where the x-axis is epochs and y-axis is MSE. We can observe how for different architectures the same minimum testing loss is achieved. In fact, the only difference is the level of overfitting, where bigger networks tend to overfit more. This finding agrees with [24]. In order to make a strong claim in this direction more research needs to be done. It is possible that we have not yet found the architecture that works for this problem.

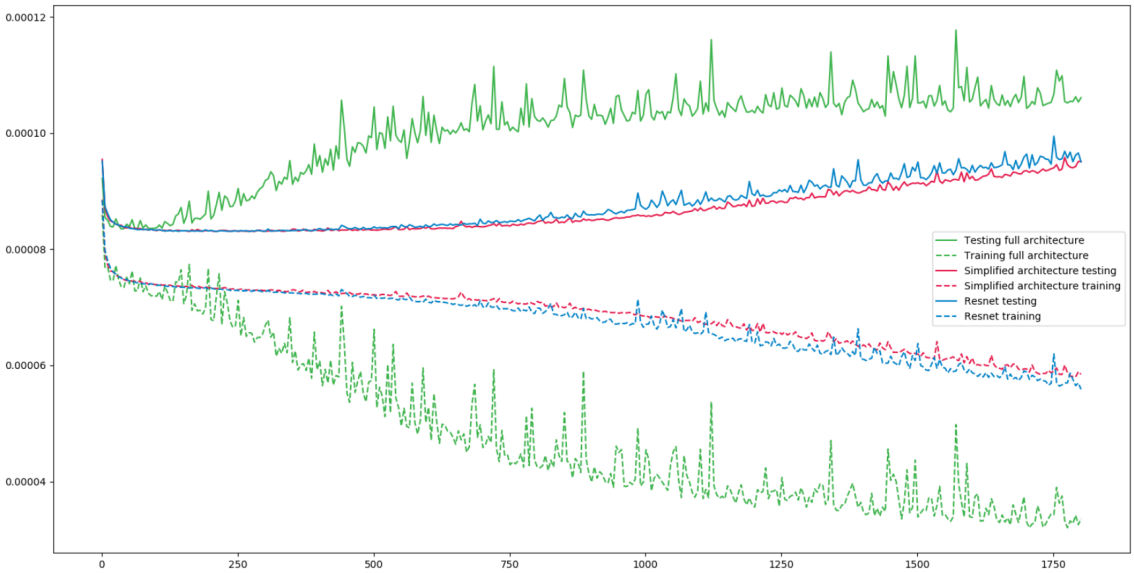


Figure 6: Training and testing loss for different CNNs

We then arbitrarily decided to use the architecture described in [23]. The basic cell of that architecture is illustrated in figure 7, where x_n is the reconstruction that starts as initial guess, and y is the scattered image. That architecture is an RNN-like architecture, in that it has cells (figure 7) that iteratively do the reconstruction, namely, the next cell is will be with x_{n+1} and y. The number of cells is a hyperparameter. For the model block, we implemented a gaussian that did not lead to reduction on the error and a more ad-hoc model that was too computationally expensive and therefore we discarded it.

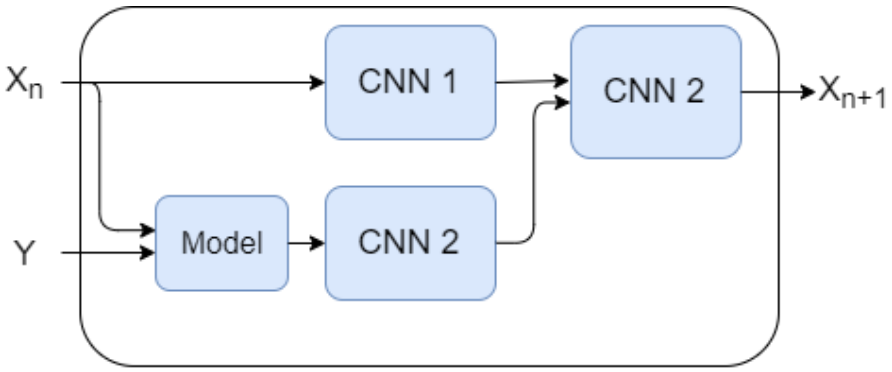


Figure 7: Implemented CNN architecture

5.2.1 Coordconv

The level of scatter radiation varies with location. That is not captured by CNNs, because CNNs apply the same filter to all locations across the image. We enforce CNN to use the location by adding the coordinates as inputs. That had been previously done in [10], where they named it Coordconv. In our implementation we include the coordinates exactly the same it was done in [10].

Thus we have a CNN that is location variant. Empirically, including the coordinates did signify a big improvement in the result. What had been impossible for a myriad of architectures, was easily accomplished by using CoordConv.

5.3 Training

Some of the main aspects of training are optimizer, learning rate, batch size, initialization, normalization, regularization (batch normalization, dropout), masking the loss. We will describe what each of them was in our case.

- **Optimizer**: When Adam [25] was developed in 2015, there was much hype about it. However, after extensive use in many different projects, people have tended to go to SGD (Stochastic Gradient Descent). In [26], they discuss some of the generalization problems associated with Adam and how they are not present in SGD. Therefore, in our experiments we use mostly SGD momentum.

- **Learning rate**: In our observations, most of the time when the neural does not converge, it is due to a high learning rate. This can be easily solved by reducing

it. Furthermore, once the neural network converges, if trained long enough, the learning rate becomes somewhat irrelevant. In other words, one achieves practically the same result with different learning rates.

- **Batch size**: In [27], they discuss the effects of using different batch sizes. Their conclusion is that large batch size (greater than 512 data points) leads to poor generalization. In our case, we are also limited by our GPU memory, so the batch size we used was 4.

- **Initialization**: We use the default initialization in pytorch, which is the He initialization [28]. We also tried Xavier-glorot initialization [29]. Both initialization methods yielded essentially the same results.

- **Normalization**: Before passing the data through the neural network, we always normalized between 0-1 and then we denormalize to the original range. We also tried normalizing to zero mean and variance one, but that did not work as well.

- **Regularization**: Dropout and CNN do not seem to work well together. Dropout is usually implemented in fully connected layers. On the other hand, batch normalization [21] seems to very effective at helping optimization, as it is extensively discussed in [30]. Unfortunately, having batch normalization occupies a significant amount of GPU memory, therefore we opted for not using it.

- **Masking the loss**: Any newcomer to deep learning is welcomed with MNIST data. Even at that level of expertise, one realizes that the neural network can recognize certain digits more easily than others e.g. 8 and 6 get misclassified but 0 is often correctly classified. Masking the loss is putting special emphasis in those patterns or parts that seem to be harder for the neural network to learn. In this particular case we implement that idea as follows:

  The loss we utilized was MSE (Mean Squared Error). That is, the error at every pixel is calculated, squared and then the mean over all pixels is calculated.

  Let $y_{i,j}$ be the pixel at location (i,j) of the ground truth, and $x_{i,j}$ be the pixel at

location (i,j) of the reconstruction, where i goes from 1 to n and j goes from i to m. MSE is:

$$MSE = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} (y_{i,j} - x_{i,j})^2 \tag{13}$$

The mask is applied as follows:

Let $m_{i,j}$ be the pixel of the mask, where i goes from 1 to n and j from 1 to m. Then:

$$MSE_{masked} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} m_{i,j}(y_{i,j} - x_{i,j})^2 \tag{14}$$

The mask is designed in such a way that assigns higher weights to the regions that are harder to learn. In other words, pixels that are harder to learn are multiplied with a higher weight, so that the contribution to the loss is higher.

This idea, however as sound as it seems, did not have the expected effect when applied. It only contributed to reduce some artifacts that were appearing. This could have been due to the fact we designed a suboptimal mask. In [23], masking seems to work pretty well. More research is needed to fully address why in our case did not seem to work but in [23] it did.

5.4 Frequency splitting

In figure 4, there was already an illustration of what splitting in frequency components looks like. In figure 8 we illustrate the three components that were actually used to train three neural network i.e. one neural network for each component. To obtain this splitting, we take the original image and apply a Gaussian filter with $\sigma = 50$, that produces Low frequency, then by equation 10 we obtain High frequency. Then we apply a Gaussian filter with $\sigma = 10$ on High frequency, that produces High low frequency and by using equation (10), we obtain High high frequency.

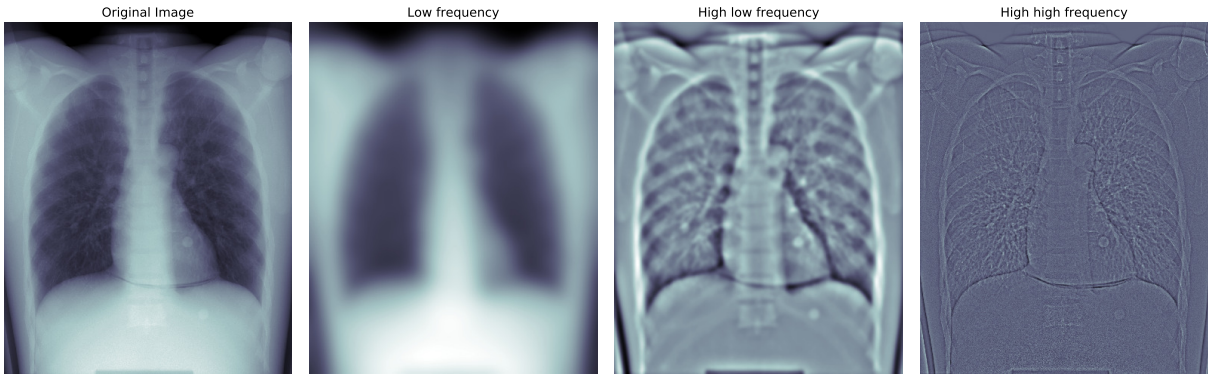


Figure 8: Frequency components used for training

5.5 Downsampling and upsampling

Low frequency in figure 8 is downsampled as described in section 4.3.1 by a factor of 8 in both directions i.e. vertically and horizontally. Then it is processed by the neural network and upsampled as described in section 4.3.2, so that the original size is recovered. On the other hand, High high frequency was not upsampled, because the image was already too big and even the original size barely fitted the memory available in the computer.

6 Results

In this section we show the experimental results that validate our claims. Firstly, we compare our CNN approach against a Bayesian optimization approach. Secondly, we empirically confirm that splitting the signal does indeed lead to lower mean squared error. We do this by showing the loss curves across epochs. Thirdly, we provide images that were obtained using the different methods we described, so that not only we have the MSE as a measure of performance but also the image itself. Finally, we show our results on more realistic images, where we compare our results against the scattered image. We provide some numerical analysis and we show and discuss some of the properties/effects that CNN has.

6.1 CNN vs Bayesian optimization

In figure 9 we can compare how CNN performs versus a Bayesian approach implemented in [4]. High-frequency information is prevalent in the middle of the semi-circle, which is recovered by the CNN approach but not by [4], which just blurs the middle. When looking at other performance features of the algorithms such as ability of denoising and contrast enhancement, we can see that both algorithms are comparable.

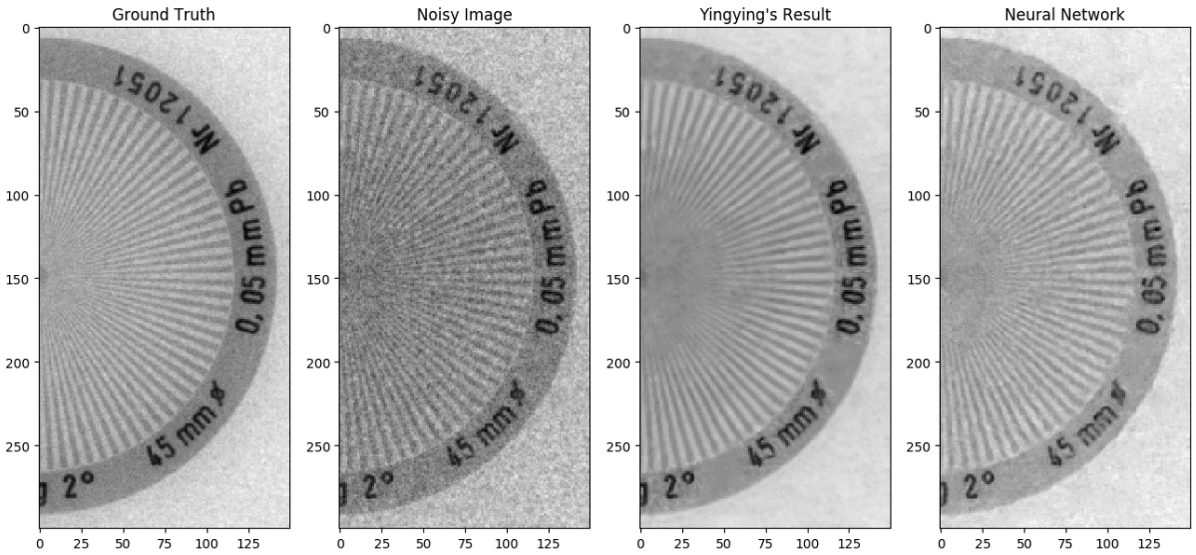


Figure 9: Scatter reduction done by CNN vs Bayesian optimization

The comparison is unfair, in that there are not complicated patterns or structures

and it does not look like a real-life X-ray image, obviously. With images with more complicated structures, the superiority of CNN becomes much more noticeable.

6.2 Analyzing loss curves of different CNNs

In figure 10, we show the loss (MSE) in the y-axis and number of epochs in the x-axis. We can see that if we add the error from the frequency components (red curve and blue curve), we still obtain less error than the error produced by the image as a whole (green curve). These curves were obtained using exactly the same architecture with the same hyperparameters. Therefore, we can conclude that the reason for the improvement was decomposing the signal into frequency bands.

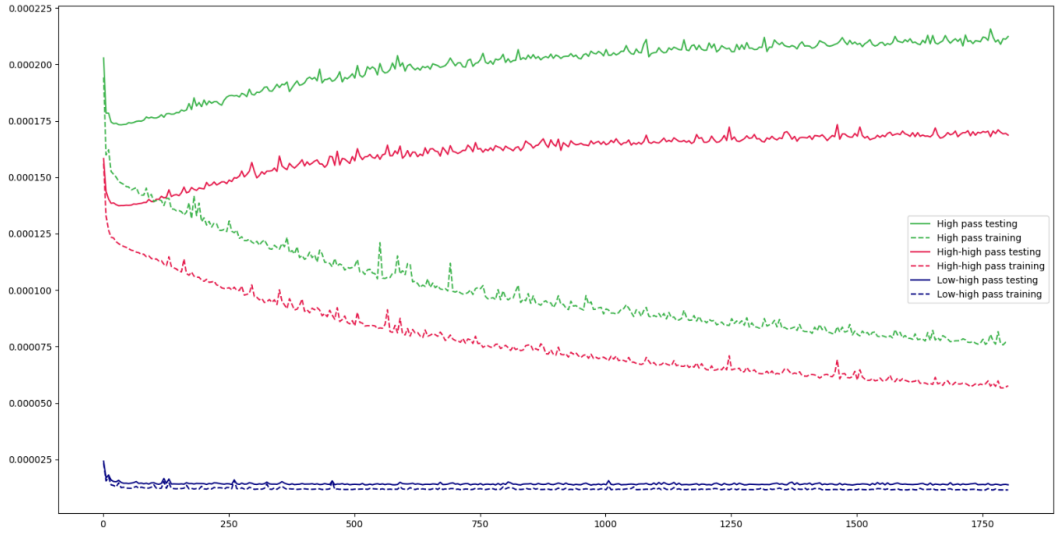


Figure 10: Loss for different frequencies

Likewise, in figure 11 we show the loss curve obtained after training on the high frequency component for an upsampled (dilated) and non-upsampled (non-dilated) signal. As previously, the same architecture and hyperparameters are maintained. Hence, we can conclude that upsampling does indeed contribute to finding smaller error.

In figure 10 and 11 we can see that the models are clearly overfitting, but nonetheless our claims still hold, namely, despite the overfitting, splitting the signal into frequency components and upsampling leads to better results.

This result is remarkable, in that if we look at figure 6, we observe that changes in architecture did not lead to a noticeable change in performance if at all but a change in

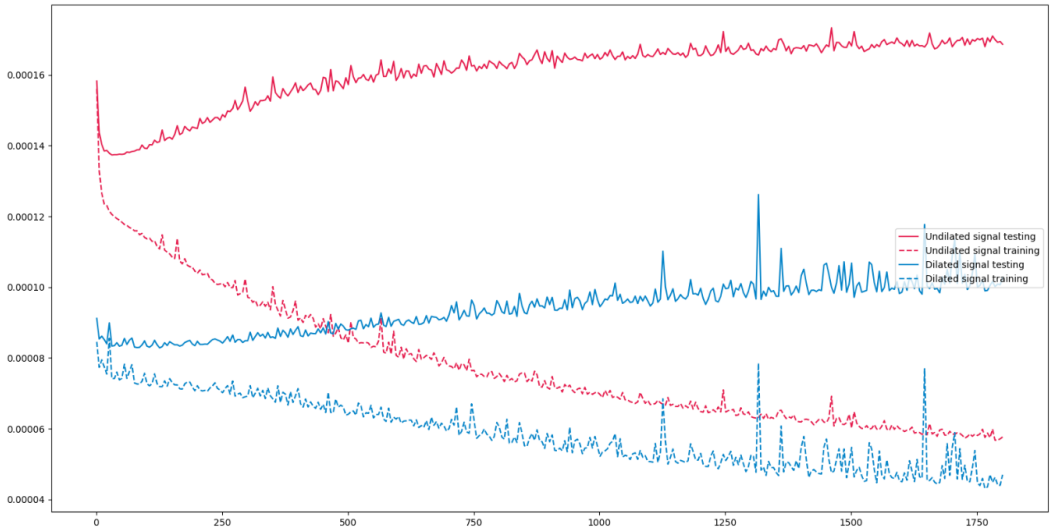


Figure 11: Loss for upsampled signal vs non-upsampled signal

pre-processing had such a strong impact on CNN performance.

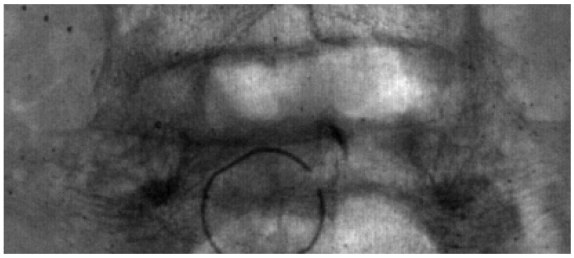
6.3 Analyzing output images of different CNNs

In figure 12 we can compare ground truth, scattered image, vanilla CNN result, and, what we shall call "splitting result". To better perceive the differences, zooming in is encouraged. That is why we have included figure 13, where we zoomed in on the region located in the lower left part of figure 12. If necessary, we recommend even zooming within the document to better observe the differences between the images shown.

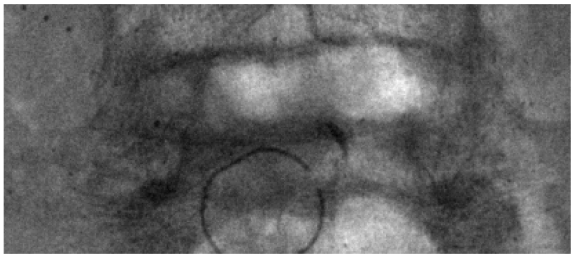
Especially in figure 13, we can observe how 13d is much more noise-free than 13c. Furthermore, 13d is even more noise-free than 13a, which is technically a mistake by the algorithm, because in theory the algorithm should reconstruct the ground truth faithfully. But practically it is beneficial, in that it is has a smoothing property than allows to generate non-noisy images even if the ground truth is somewhat noisy. We will refer to this as the smoothing property of CNNs.

Another main observation is that, partly because 13d has essentially no noise, some of the details are sharper in 13d. For instance, within the big red ellipse in 13, in 13d, it is evident there is a pattern that almost follows a line. In 13c that is not so evident. On the other hand, in 13a that line-pattern is obvious.

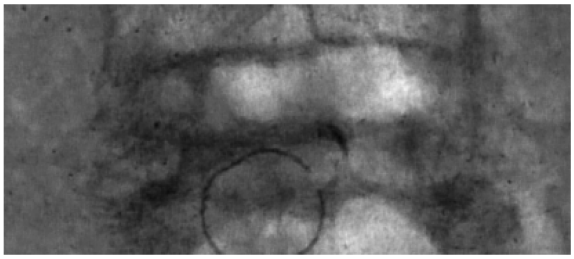
As aforementioned, the data we have used so far was toy data. That is reflected in



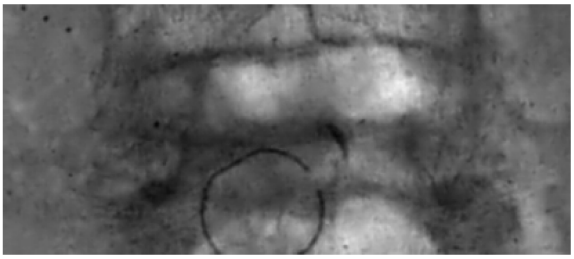
(a) Ground Truth



(b) Scattered Image



(c) Vanilla Neural Net Result



(d) Result with splitting and upsampling

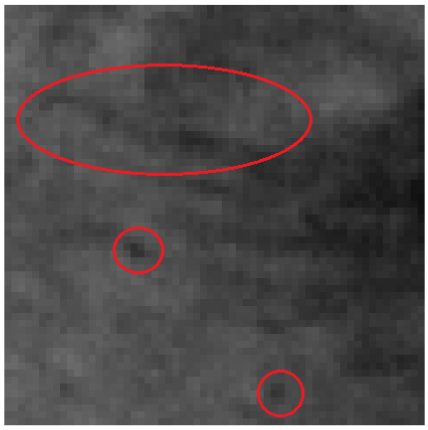
Figure 12: Our approach with vanilla CNN

the fact that if we compare 12a and 12b, we see that the main difference is noise and some very slight blurring effect. We can notice the latter effect more clearly in 13. In other terms, the scatter effects in figure 12 are not very prominent.

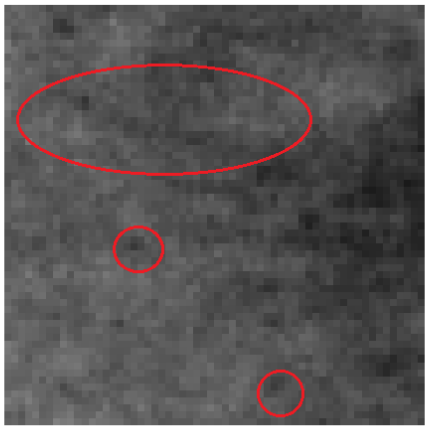
To sum up, we use this image to validate our approach but the image does not fully display the power of our approach.

6.4 Our method applied to realistic images

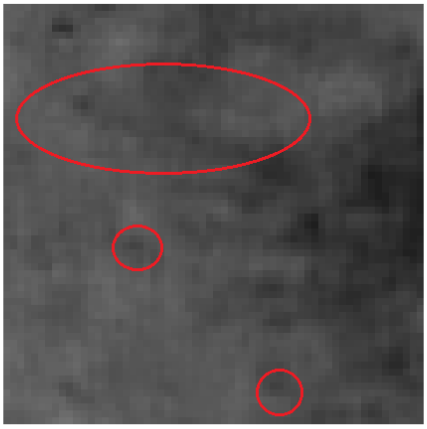
To rigorously evaluate our method, we need to use images that are as close as possible to human images obtained in chest bedside radiography. We then use the two last objects described in section 5.1.1. "Thoracic region of a phantom 2 mAs" has no saturation and the scatter effects are noticeable. "Thoracic region of phantom" taken with 3.2 mAs and 4 mAs were discarded because, even though scatter effects are present, the level of saturation was too high and using them would have undermined the learning by CNN. Since we have 12 images of the thoracic region, we take 11 to train and 1 to test. To further validate our results, we use "Abdominal region of phantom" as testing as well.



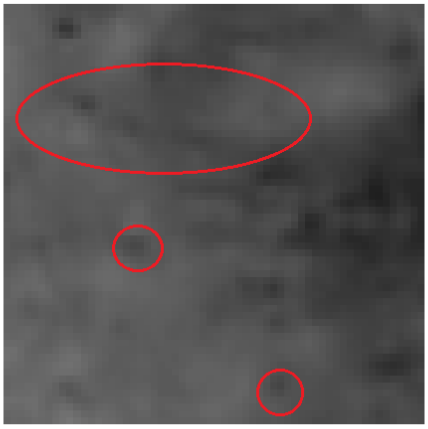
(a) Ground truth



(b) Scattered image



(c) Vanilla neural net result



(d) Result with splitting and upsampling

Figure 13: Our approach compared with vanilla CNN. Zoomed in.

Note that it could have not been done the other way around, because "Abdominal region of phantom" does not have as much scatter effect. Hence if we had trained on that data, CNN would not have been able to learn how to reduce scatter. It would have been ideal to have many more images from different objects.

Figure 14 illustrates the CNN reconstruction for the chest phantom image. To better view please use a screen. At a glance, we notice that at the edge of the rib cage there is a white strip. In the scattered image that is barely noticeable. On the other hand, in the CNN reconstruction that edge is standing out just as much as it is in the ground truth. Another region that presented a major difference among the images was the lower spine.

Figure 14: CNN reconstruction for chest phantom

In figure 15 we zoom in on the rib cage region. The improvement from scattered image to CNN reconstruction becomes evident. Then, in figure 16 we plot a horizontal slice taken across figure 15, where the x-axis represents pixel location across the slice and y-axis represents intensity. We can clearly observe the smoothing property of CNN. In addition, CNN significantly enhances the edge between pixel 75 and 130, that is in fact the white strip that stands out in 15 for Ground truth and CNN reconstruction but not for scattered image. There is an undesired effect however. After the "bump" the neural network curve does not decrease as fast as ground truth and the scattered image. We see that at pixel 130 (figure 16) ground truth's curve and scattered image's curve meet and

continue together, whilst neural network is still above them. In terms of the image, that means there is a local increase in the mean.

In figure 17 we zoom in on the spine region. Again, the improvement is evident. The spinal structure is hard to detect in the scattered image, whereas in ground truth and CNN reconstruction is not. Figure 18 shows the profile along a horizontal slice. We can observe the smoothing property again, and that the CNN takes the scattered image that is basically flat (excluding the noise) and outputs some edges. In figure 16 the problem of the locally increased mean persists nonetheless, which in this case is reflected in figure 17 as having excessive brightness.

Locally increased mean does not substantially diminish the image quality. In figure 14, there is no sign of that effect. Furthermore, radiologists look at the image globally, so the locally increased mean is not a major issue.

The MSE between the scattered image and ground truth and ground truth and reconstruction is 1,191,021 and 1,0545,587, respectively. That is a reduction of roughly 12%, which does not seem significant at all. We need to recall nonetheless that the neural network locally increased mean will contribute to the MSE but does not substantially affect the image itself because the mean is irrelevant to the screen when the image is displayed. If we remove the global mean from all the three images and calculate the MSE again, we obtain that the MSE between scattered and ground truth and ground truth and reconstruction is 1095051 and 764332, respectively. That is a reduction of roughly 31 %.

In [13], they define the Contrast improvement factor (CIF), which is the only factor they used to assess their results. This factor is defined over a disk, so that there is contrast between the disk and background. It is defined as CIF $= \frac{C_N}{C_0}$, where C_0 are the contrast improvements from the reference image (C_0), which in our case is the scattered image, and C_N which is the image to be tested and C is defined as:

$$C = \frac{X_{out} - X_{in}}{X_{out}} \tag{15}$$

Where C is the contrast coefficient, X_{out} is the average around the disk and X_{in} is

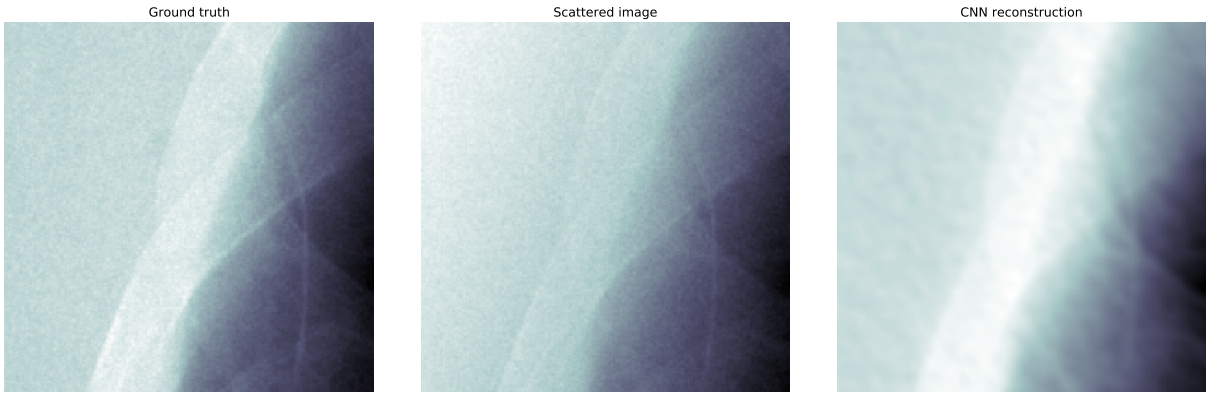


Figure 15: CNN reconstruction for chest phantom

the average inside the disk.

We calculate that factor for our results and ground truth to make our results comparable to those in [13]. The CIFs are shown in table 1. For the top left and bottom left disks, our result's CIF is higher than ground truth itself but for the top right and bottom right disks our results falls short, that is, our contrast improvement was lower than ground truth's. Note, though, that every CIF was greater than 1. That means that for each of the disk there was contrast improvement according the CIF.

Disk	Top left	Bottom left	Top right	Bottom right
Ground truth	1.3041	1.2359	1.5956	2.4800
CNN result	1.3524	1.3399	1.1787	1.6479

Table 1: CIF factor for ground truth and CNN reconstruction

This index, however, is flawed. For that if one linearly normalizes or changes the global mean, then C changes (increases or decreases, depending whether a mean is added or subtracted). But recall that a screen is not affected by changing the mean, hence C as a measure of contrast should not change if the image is normalized but nevertheless it is changed. We show that as follows:

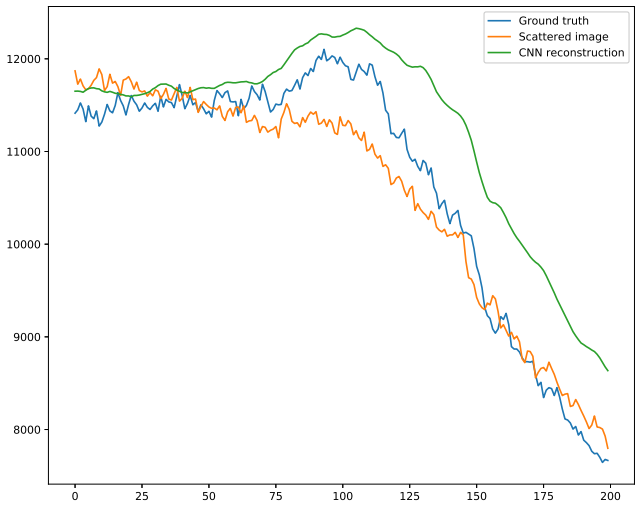


Figure 16: Profile curves for rib cage region

Let us normalize as:

$$Xn_i = \frac{X_i - X.min}{X.max - X.min} \tag{16}$$

Where Xn_i is the normalized pixel, X_i is the image pixel, X.min is the minimum value of image X and X.max is the maximum value of image X.

We can calculate C again after normalization and it becomes:

$$C_{Norm} = \frac{\frac{X_{out}-X.min}{X.max-X.min} - \frac{X_{in}-X.min}{X.max-X.min}}{\frac{X_{out}-X.min}{X.max-X.min}} \tag{17}$$

We simplify equation 17 and get:

$$C_{Norm} = \frac{X_{out} - X_{in}}{X_{out} - X.min} \tag{18}$$

As we can see after normalization $C > C_{norm}$, namely, the contrast was improved by just linearly normalizing, even though that has not real effect on the contrast when the image is displayed.

In figure 19 we show the other test image. As previously described: this is image is not as useful or insightful as the thoracic image because the scatter effects are not as

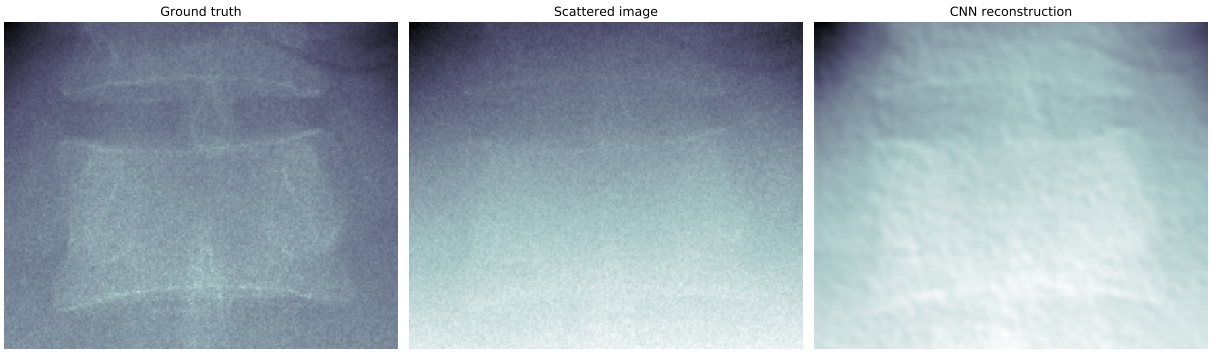


Figure 17: Zoomed-in CNN reconstruction for chest phantom

prevalent. Ground truth and the scattered look very similar. That image also has some saturated regions, where the information is totally gone and therefore irrecoverable.

An unexpected "artifact" that we can observe in this image is that the neural network does not seem to differentiate between foreground and background. The tube-like structure that possibly represents the corpus cavernosum in the phantom is much brighter in ground truth than the scattered image. The CNN brightens that region but as an unexpected consequence, it also brightens the background. This is not a major issue, because when displayed in the clinic, the image is masked and the background is ignored.

In figure 20 we zoom in on a region of figure 19. In there, we see one of the disks that were previously mentioned. There is no major difference among the three images. In particular, the contrast between the disk and background appears to be the same for three images; no perceptible difference is found. There is only some more texture in ground truth, which does not carry vital information about the image. Furthermore, once it is zoomed out, that texture is not perceived anymore. Likewise, in figure 21 we show the profile of a slice of figure 20. We observe the smoothing property of CNN. In addition, we observe that the behavior of the three curves is similar. The only difference is in the local mean, which does not play a major role, as we previously determined.

With this image we omit the numerical analysis (MSE, CIF). MSE: Because, on one

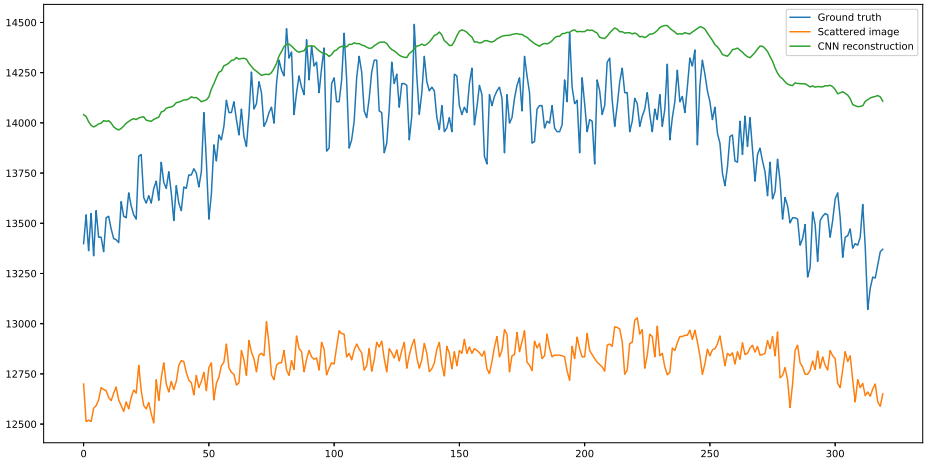


Figure 18: Profile curves for spinal region

hand, MSE does not fully capture the quality of an image, i.e. having lower MSE does not necessarily mean perceived image quality is better and on the hand, for images whose quality is so visually similar, MSE can be misleading for the reason that we previously stated. CIF: because as we shown before, CIF is a highly flawed measure of contrast.

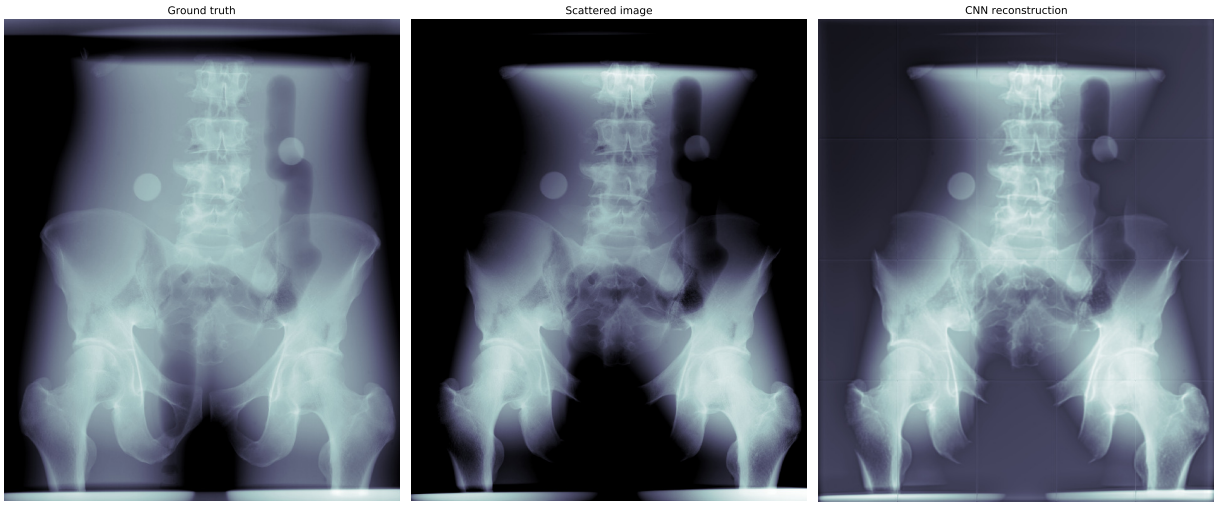


Figure 19: CNN reconstruction for abdominal phantom

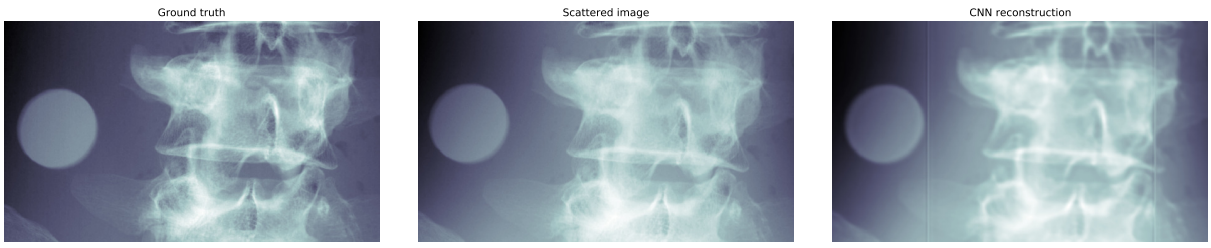


Figure 20: Zoomed-in CNN reconstruction for abdominal phantom

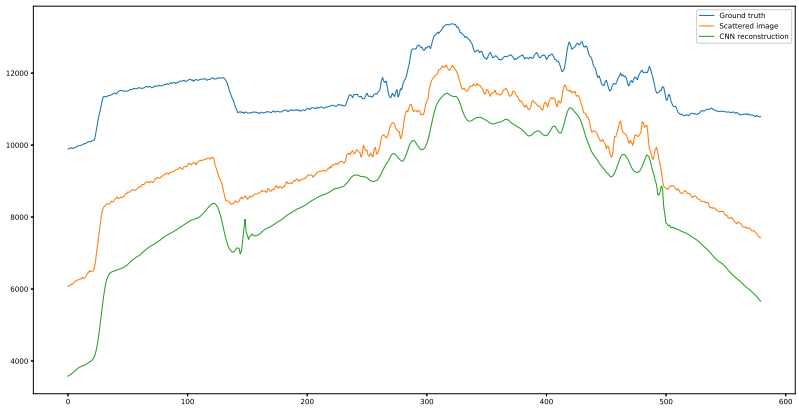


Figure 21: Profile curves for abdominal phantom

7 Conclusions and open problems

In this work, we have proposed a deep learning solution for the problem of scatter reduction. We have investigated and validated the effectiveness of splitting the data in frequency components as well as upsampling high frequency information and downsampling low frequency information. We empirically found that CNNs are better when dealing with signals whose frequency is not too low or too high. We believe that is reason why downsampling and upsampling help. On the other hand, to our surprise, architecture did not seem to play a major role in the learning ability of CNN, that is, for very different architectures we obtained very similar, if not equal, results.

Instead, adding coordinates as additional channels in the input provided a more significant improvement in results than any change in architecture. Moreover, modifying other hyperparameters such as learning rate, batch size, numbers of layers/channels did not seem to have much effect in the learning ability. This opens up three possible avenues for further investigation: One should focus more on pre-processing (frequency splitting, upsampling, downsampling, Coordconv are all pre-processing) and less in architecture when designing CNNs. We were never able to find the right set of hyperparameters and that is why they did not seem to substantially influence the result. If this were the case, then a natural question would be: how to find the right hyperparameters? Research is being done on that topic but so far no conclusive have been generated. The data was just not enough and that is why despite having sophisticated architectures, the results were not optimal.

8 References

[1] Ruhrnschopf Peter and Klingenbeck Klaus. A general framework and review of scatter correction methods in x-ray cone beam computerized tomography. part 1: scatter compensation approaches. In *Medical Physics, Volume 38, Issue 7*, pages 4296–4311. American Association of Physicists in Medicine, 2011.

[2] Ruhrnschopf Peter and Klingenbeck Klaus. A general framework and review of scatter correction methods in x-ray cone beam computerized tomography. part 2: scatter estimation approaches. In *Medical Physics, Volume 38, Issue 9*, pages 5186–5199. American Association of Physicists in Medicine, 2011.

[3] Ping Xue Yingying Gu, Jun Zhang. Scatter correction by non-local techniques. volume 10133, pages 10133 – 10133 – 9, 2017.

[4] Y. Gu, J. Zhang, and P. Xue. Multi-grid nonlocal techniques for x-ray scatter correction. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 10574 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 105741O, March 2018.

[5] Jartuwat Rajruangrabin Chalinee Thanasupsombat Tanapon Srivongsa Sorapong Aootaphao, Saowapak S. Thongvigitmanee and Pairash Thajchayapong. "x-ray scatter correction on soft tissue images for portable cone beam ct," biomed research international, vol. 2016, article id 3262795, 12 pages,. pages 1–5, 10 2016.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[8] Hua Qian, Xue Rui, and Sangtae Ahn. Deep learning models for pet scatter estimations. pages 1–5, 10 2017.

[9] Junhong Min Eunhee Kang and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *American Association of Physicists in Medicine*, 2017.

[10] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *CoRR*, abs/1807.03247, 2018.

[11] The essential physics of medical imaging. *European Journal of Nuclear Medicine and Molecular Imaging*, 30(3):456–456, Mar 2003.

[12] E. Rehn. Modeling of scatter radiation during interventional x-ray procedures. 2015.

[13] Detlef Mentrup, Sascha Jockel, Bernd Menser, and Ulrich Neitzel. ITERATIVE SCATTER CORRECTION FOR GRID-LESS BEDSIDE CHEST RADIOGRAPHY: PERFORMANCE FOR A CHEST PHANTOM. *Radiation Protection Dosimetry*, 169(1-4):308–312, 06 2016.

[14] J C Wandtke. Bedside chest radiography. *Radiology*, 190(1):1–10, 1994. PMID: 8043058.

[15] William Briggs, Van Henson, and Steve McCormick. *A Multigrid Tutorial, 2nd Edition*. 01 2000.

[16] Zhi-Qin J. Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. *CoRR*, abs/1807.01251, 2018.

[17] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24:8 – 12, 05 2009.

[18] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian

Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft II: A new challenge for reinforcement learning. *CoRR*, abs/1708.04782, 2017.

[19] Realtalk: How we recreated joe rogan's voice using ai. `https://medium.com/dessa-news/realtalk-how-it-works-94c1afda62f0`. Accessed: 2019-11-17.

[20] Yingying Hu and Jun Zhang. Multi-grid nonlocal tehcniques for x-ray scatter correction. *American Association of Physicists in Medicine*, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[22] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[23] Hongquan Zuo. *Model Augmented Deep Neural Network for Medical Image Reconstruction Problem.* PhD thesis, uwm, 2019. Theses and Dissertations. 2278.

[24] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *CoRR*, abs/1803.08450, 2018.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to SGD. *CoRR*, abs/1712.07628, 2017.

[27] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[29] Xavier Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 01 2010.

[30] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2483–2493. Curran Associates, Inc., 2018.