

Article

Improving Tomographic Reconstruction from Limited Data Using Mixed-Scale Dense Convolutional Neural Networks

Daniël M. Pelt ^{1,*} , Kees Joost Batenburg ¹, James A. Sethian ^{2,3}

¹ Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands; joost.batenburg@cwi.nl

² Department of Mathematics, University of California, Berkeley, CA 94720, USA; sethian@math.berkeley.edu

³ Center for Applied Mathematics for Energy Research Applications, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

* Correspondence: d.m.pelt@cwi.nl; Tel.: +31-(0)20-592-9333

Received: 1 September 2018; Accepted: 10 October 2018; Published: 30 October 2018



Abstract: In many applications of tomography, the acquired data are limited in one or more ways due to unavoidable experimental constraints. In such cases, popular direct reconstruction algorithms tend to produce inaccurate images, and more accurate iterative algorithms often have prohibitively high computational costs. Using machine learning to improve the image quality of direct algorithms is a recently proposed alternative, for which promising results have been shown. However, previous attempts have focused on using encoder–decoder networks, which have several disadvantages when applied to large tomographic images, preventing wide application in practice. Here, we propose the use of the Mixed-Scale Dense convolutional neural network architecture, which was specifically designed to avoid these disadvantages, to improve tomographic reconstruction from limited data. Results are shown for various types of data limitations and object types, for both simulated data and large-scale real-world experimental data. The results are compared with popular tomographic reconstruction algorithms and machine learning algorithms, showing that Mixed-Scale Dense networks are able to significantly improve reconstruction quality even with severely limited data, and produce more accurate results than existing algorithms.

Keywords: machine learning; deep learning; image reconstruction; tomography

1. Introduction

Tomography is widely used to nondestructively study the internal structure of various types of objects, for example using synchrotron radiation [1], laboratory X-ray scanners [2], or electron microscopes [3]. Typically, multiple 2D projection images are acquired while rotating an object. A *tomographic reconstruction algorithm* is then used to compute a 3D image of the internal structure of the scanned object using the acquired projection images. Because of the practical relevance of tomography, tomographic reconstruction has been extensively studied in the past, and a wide range of reconstruction algorithms have been developed [4]. Generally, popular existing algorithms can be divided into two groups: direct algorithms and iterative algorithms. Direct algorithms, e.g., filtered backprojection (FBP), are computationally efficient and produce accurate results if many artifact-free projections are available. Iterative algorithms, e.g., SIRT and Total-Variation minimization [5], can produce more accurate reconstructions than direct methods by exploiting prior knowledge about the scanned object and the experimental setup, but typically have high computational costs.

In many applications of tomography, the acquired projection images are limited in one or more ways. Common data limitations include: (1) acquiring only a limited number of projection images;

(2) having a significant amount of noise in the projection images; and (3) acquiring projections only for a limited angular range. Such limitations are often caused by unavoidable constraints of the experiment, such as a limit on the total X-ray dose that can be deposited on dose-sensitive objects [6], time constraints when scanning a large number of objects [7], or when the experimental setup blocks part of the imaging beam [8]. In problems with limited data, direct algorithms often produce reconstructions with insufficient quality for further analysis [9]. Iterative algorithms are typically able to produce more accurate results for limited data [9], but their high computational costs can prohibit their application in practice [10]. Furthermore, the type of prior knowledge that is exploited by an iterative algorithm limits the type of objects the algorithm can be successfully applied to.

In recent years, machine learning algorithms have proved successful in many image processing problems, e.g., image classification [11], semantic segmentation [12], and image enhancement [13]. These algorithms are based on deep convolutional neural networks (CNNs) that successively convolve images using *learned* convolution parameters. The parameters are learned by presenting the network with a large number of input images and corresponding “ground truth” target images. Because of its success in other imaging fields, it is likely that machine learning algorithms can be used to improve tomographic reconstruction, and, in earlier work, multiple authors have shown promising results [14–22]. For non-standard tomographic acquisition schemes in which data from only a few rays can be detected, e.g., plasma tomography, CNNs can be used to directly compute reconstructed images from the acquired data [20]. For the more standard tomographic acquisition scheme described above, several other approaches have been proposed. One approach is to include machine learning in existing tomographic reconstruction algorithms, thereby improving reconstruction quality. In [14,15], filters of the popular direct FBP algorithm are learned by a machine learning algorithm, optimizing them for each experiment. The resulting algorithm is computationally efficient, but, because the underlying neural network is relatively shallow, the improvement in image quality is relatively small. In [23], a deep neural network is included in an iterative reconstruction algorithm, allowing prior knowledge to be learned by the algorithm. Because the approach is based on an iterative algorithm, however, its computational costs are relatively high, making application to problems with real-world image sizes difficult.

Another approach is to use machine learning as a *post-processing* operation after reconstruction with a standard reconstruction algorithm. The advantage of this approach is that it is computationally efficient, widely applicable, and that existing software and hardware infrastructure can be used at experimental facilities to compute initial reconstructions. In many cases, *encoder–decoder networks*, a specific type of deep network of which U-Net is a popular example [12], are used to process tomographic images. For example, in [19], an encoder–decoder network is used to improve reconstructed images computed with direct algorithms for a single type of data limitation (a limited number of available projections), resulting in significant improvements in image quality. In [21], an encoder–decoder network is used to pre-process noisy projection images to improve the image quality of the resulting reconstruction. The encoder–decoder network architecture, however, includes a large number of parameters that have to be learned (often several million), which increases the risk of overfitting [24] and requires a large amount of training images (often several thousand) to produce accurate results. Furthermore, encoder–decoder networks require many intermediate images to compute accurate output images, resulting in prohibitively high computer memory requirements when presented with the large image sizes that are common in tomographic imaging. Finally, the accuracy of encoder–decoder networks depends on several hyperparameters that are problem-dependent and difficult to choose a priori, requiring time-consuming trial-and-error approaches to find reasonable values for each problem.

Recently, a new neural network architecture was proposed, the Mixed-Scale Dense architecture [22], which was designed to avoid the disadvantages of the encoder–decoder architecture given above. Specifically, Mixed-Scale Dense (MS-D) networks typically require significantly fewer

trainable parameters and intermediate images than encoder–decoder networks to obtain accurate results, thereby reducing the amount of training images that are needed and enabling application to significantly larger images. Furthermore, MS-D networks are able to automatically adapt to new problems, removing the need to choose sensitive hyperparameters in time-consuming ways. These properties make the MS-D network architecture well suited for processing tomographic images, which are often relatively large and are used in a wide variety of application fields, with each application requiring the training of a new network. The goal of this paper is to present the application of MS-D networks for improving the image quality of large tomographic reconstruction images. Here, we show results for a wide variety of common data limitations, including a limited number of projections, a limited angular range, and a limited exposure time per projection. We compare results with popular tomographic reconstruction algorithms and machine learning approaches.

This paper is structured as follows: in Section 2, the problem of tomographic reconstruction is defined and standard algorithm approaches are described. In addition, the encoder–decoder network architecture and MS-D network architecture are discussed. In Section 3, the experiments that were performed in this paper are discussed and results are shown comparing the proposed MS-D network with popular existing approaches. Finally, possibilities for future research are given in Section 4, along with some final remarks.

2. Materials and Methods

2.1. Problem Definition

In this paper, we focus on parallel-beam tomography problems, although the presented post-processing approach is applicable to other tomography geometries as well, such as cone-beam geometries. In parallel-beam geometries, the problem of reconstructing the 3D internal structure of the scanned object can be viewed as a collection of independent 2D reconstruction problems, which can be stacked to form a 3D image. In the following, we therefore focus on reconstructing and processing a single 2D image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. The formation of a projection image $P_\theta : \mathbb{R} \rightarrow \mathbb{R}$ at angle $\theta \in \mathbb{R}$ is given by the Radon transform:

$$P_\theta(u) = \iint_{\mathbb{R}^2} f(x, y) \delta(x \cos \theta + y \sin \theta - u) dx dy \tag{1}$$

where $u \in \mathbb{R}$ denotes the position on the detector and $\delta : \mathbb{R} \rightarrow \mathbb{R}$ is the Dirac delta function. In practice, projection data are acquired only for a finite set of N_θ projections and N_d detector elements, and the scanned object is represented by a pixel grid of $N \times N$ pixels. In this case, the acquired projection data can be described by a vector $\mathbf{y} \in \mathbb{R}^{N_\theta \times N_d}$, the reconstructed image by a vector $\mathbf{x} \in \mathbb{R}^{N \times N}$, and the formation of the projection data by a linear system:

$$\mathbf{y} = \mathbf{A} \mathbf{x} \tag{2}$$

with element $a_{ij} \in \mathbb{R}$ of $\mathbf{A} \in \mathbb{R}^{N_\theta N_d \times N^2}$ equal to the contribution of image pixel j to detector pixel i . The tomographic reconstruction problem is then to recover the unknown object \mathbf{x} from the acquired projection data \mathbf{y} .

2.2. Tomographic Reconstruction

As discussed above, popular tomographic reconstruction algorithms can be generally divided into two groups: direct algorithms and iterative algorithms. Direct algorithms are based on finding a continuous inversion formula of the continuous forward model (Equation (1)) and discretizing the result. For parallel-beam geometries, the result of this approach is the popular Filtered Backprojection (FBP) algorithm [4], which can be written as:

$$\mathbf{x}_{FBP} = \mathbf{A}^T \mathbf{C}_h \mathbf{y} \tag{3}$$

where C_h is a 1D convolution operation that convolves each detector row in \mathbf{y} with a filter $\mathbf{h} \in \mathbb{R}^{N_d}$. Typically, a fixed standard filter is used for each reconstruction, which can include additional low-pass filtering to reduce noise in the reconstructed image [25]. Recently, it was shown that more accurate results can be obtained when optimizing the used filter for each reconstruction, for example by using machine learning algorithms to learn accurate filters [14]. The advantage of direct algorithms is that they are usually computationally efficient, and produce accurate results when a large number of high-quality projection images are available. However, when presented with limited data, e.g., a limited number of projections, reconstruction images computed by direct algorithms often have insufficient image quality for the required analyses [9]. In these cases, iterative algorithms tend to produce more accurate results.

Iterative reconstruction algorithms are based on iteratively solving the discrete linear system given in Equation (2). A popular class of algorithms tries to find images that minimize the ℓ^2 -norm of the *residual error*, i.e., the difference between the acquired data and simulated projections of the reconstruction image, and an additional term $g : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}$ that penalizes images that do not fit with the chosen prior knowledge about the scanned object. The solution of such *regularized* iterative algorithms can be written as:

$$\mathbf{x}_{iter} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda g(\mathbf{x}) \tag{4}$$

where $\lambda \in \mathbb{R}$ controls the relative weighting between the residual error and the prior knowledge penalty. Note that similar regularization is also used in machine learning to improve the generalization of trained networks [26]. A popular choice of $g(\mathbf{x})$ is Total-Variation Minimization (TV-MIN) [5], i.e., $g(\mathbf{x}) = \|\nabla \mathbf{x}\|_1$ with ∇ a discrete gradient operation. If the scanned object obeys the requirements of the chosen penalty function $g(\mathbf{x})$, iterative reconstruction algorithms can produce significantly more accurate results than direct methods when reconstructing from limited data [9]. One of the main disadvantages of iterative algorithms, however, is their high computational costs, which can make it difficult to apply them in practice to real-world tomographic data. Furthermore, the correct value for the weighting parameter λ is highly problem-dependent and difficult to find a priori, requiring time-consuming trial-and-error approaches to determine. Finally, the chosen penalty function limits the type of objects that can be accurately reconstructed to those that obey that penalty function. As a result, direct algorithms are still the most popular reconstruction algorithms in many fields [27].

2.3. Deep Neural Networks for Improving Reconstructed Images

A recently proposed alternative to obtain accurate reconstructions from limited data is to use machine learning algorithms to improve reconstruction images computed by direct algorithms. Formally, we can define such approaches as:

$$\mathbf{x}_{ML} = ML(\mathbf{x}_{FBP}) = ML\left(\mathbf{A}^T \mathbf{C}_h \mathbf{y}\right) \tag{5}$$

where $ML : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$ represents the chosen machine learning algorithm that takes an FBP reconstruction and produces an improved reconstruction image. Here, we focus on Deep Convolutional Neural Networks (CNNs), which are a class of machine learning algorithms that have proved successful for analyzing and improving image data. CNNs work by processing images in successive layers, with each layer consisting of multiple images. The images of each layer are computed by convolving the images of the previous layer with learned filters, and applying a non-linear activation function to each pixel of the resulting image. Formally, we can describe image j in layer i of the network by $\mathbf{z}_i^j \in \mathbb{R}^{N \times N}$ and the number of images in layer i by n_i . We can then write the computation of \mathbf{z}_i^j in a typical CNN as:

$$\mathbf{z}_i^j = \sigma \left(\sum_{k=0}^{n_{i-1}-1} \mathbf{H}_{q_{ijk}} \mathbf{z}_{i-1}^k + b_{ij} \right) \tag{6}$$

where $H_{q_{ijk}}$ is a 2D convolution with a learned 3×3 pixel filter $q_{ijk} \in \mathbb{R}^{3 \times 3}$, $b_{ij} \in \mathbb{R}$ is a learned bias of each layer image, and $\sigma : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$ is a non-linear pixel-wise activation function, e.g., the ReLU function [28]. The first layer of the network consists of the input image to the network, and in the final layer the output image of the network is computed, for which a different activation function σ_o is often used. In the rest of this paper, we use the identity function as the final activation function ($\sigma_o(z) = z$) and the ReLU function for all other layers. To find values for the trainable parameters q_{ijk} and b_{ij} such that the CNN performs the task that is required, supervised learning can be used. In supervised learning, a large set of training image pairs are used, each consisting of an input image and a corresponding target image that should represent what the network output should be for that input image. After randomly initializing the network parameters, the training images are used to iteratively minimize a chosen objective function that measures the difference between the network output and target images.

Many previous attempts (e.g., [19,21]) to apply CNNs to tomographic images have used an encoder–decoder network architecture [12]. In addition to the standard convolutional operations described in Equation (6), encoder–decoder networks introduce three additional operations between layers: (1) downscaling, in which the images of the previous layer are reduced in size with a fixed downscaling operation; (2) upscaling, in which the images of the previous layer are increased in size with a learned upscaling operation; and (3) skip connections, which allow deep layers to use early layers as input. Typically, images are incrementally scaled down in the first half of the network layers, the encoder part, and subsequently scaled up in the second half of the layers, the decoder part. In Figure 1, a schematic of a typical encoder–decoder network is shown.

Although encoder–decoder networks are often able to produce accurate results in various fields, several problems prevent a wide application in processing real-world tomographic images. First, even though encoder–decoder networks typically have fewer trainable parameters than CNNs that include fully connected layers [29], they often still require a large number of trainable parameters (e.g., millions) to produce accurate results. As a result, to accurately train encoder–decoder networks, a large amount of training images are needed to prevent overfitting the network to the training data [22,24]. Second, encoder–decoder networks often require a large number of intermediate images (e.g., hundreds) to produce accurate results as well. Because of the large amount of required intermediate images, encoder–decoder networks require a large amount of computer memory, which can prohibit their application to tomographic images that often contain several million pixels. Finally, before applying an encoder–decoder network to a certain problem, several hyperparameters have to be chosen: how many downscaling and upscaling operations to use, how many images to use in each layer, and between which layers to add skip connections. These choices have a significant effect on the accuracy of the resulting network and are highly problem-dependent [22], and it is difficult to pick reasonable parameter values a priori and know which specific hyperparameter to change to improve results. As a result, a separate time-consuming trial-and-error search is required to find good values for each new application, making it difficult to apply these networks to tomographic problems in practice.

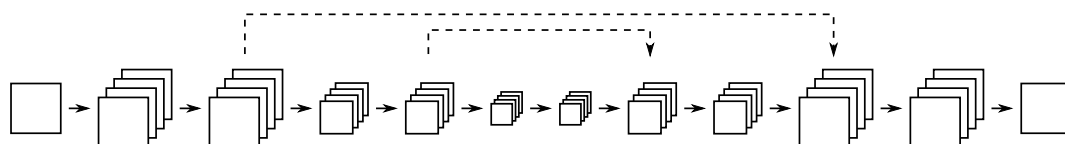


Figure 1. Schematic of a typical encoder–decoder CNN. The input image is shown on the left, and the network output on the right. In between, the networks consists of multiple layers, with arrows depicting convolutional operations, downscaling operations, and upscaling operations between layers. Skip connections are depicted by dashed arrows.

2.4. Mixed-Scale Dense Convolutional Neural Networks

In recent work [22], a new type of neural network, the Mixed-Scale Dense neural network (MS-D network), was proposed to avoid the disadvantages of encoder–decoder networks. To explain

MS-D networks, we first make two observations about encoder–decoder networks: (1) if a useful image feature is found in a certain layer, it has to be copied to deeper layers in order to be used, wasting operations and intermediate images; and (2) by having the encoder part before the decoder part, it is impossible for information learned by the decoder part to be used by the encoder part. In MS-D networks, all layers are *densely* connected: to compute an image of a certain layer, all previous layer images are used as input instead of only those of the previous layer as in standard CNNs. These dense connections prevent networks from having to copy useful features throughout the network, allowing MS-D networks to achieve accurate results with fewer intermediate images and trainable parameters than encoder–decoder networks. Because of the dense connections, we can choose to have each layer contain only a single image, increasing network depth without increasing the number of intermediate images. In addition, MS-D networks do not include any downscaling or upscaling operations, but use *dilated convolutions* [30] for capturing image features at various scales instead. By assigning a specific dilation $d_i \in \mathbb{Z}^+$ to each layer, each layer captures features at a certain image scale. If these scales are mixed within the network by choosing certain dilation distributions, we effectively mix the encoder and decoder parts within the network, allowing information from each to be used by the other. Formally, we can describe the computation of the image of layer i , $m_i \in \mathbb{R}^{N \times N}$, as:

$$m_i = \sigma \left(\sum_{k=0}^{i-1} D_{q_{ik}}^{d_i} m_k + b_i \right) \tag{7}$$

where $D_{q_{ik}}^{d_i}$ is a dilated convolution operation with dilation d_i and learned 3×3 pixel filter $q_{ik} \in \mathbb{R}^{3 \times 3}$, and $b_i \in \mathbb{R}$ is the learned bias of layer i . In Figure 2, a schematic of a MS-D network with eight layers is shown.

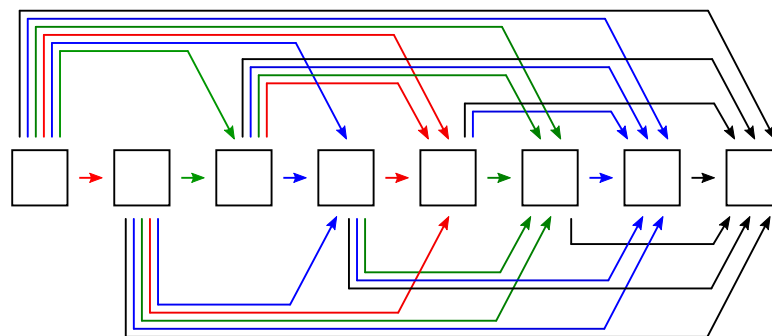


Figure 2. Schematic of an MS-D network. Arrows depict dilated convolution operations, with each color representing a different dilation factor. Note that all intermediate images are the same size, all layers consist of only a single image, and each layer (including the output layer) takes all previous layer images as input (including the input image).

Compared with encoder–decoder networks, MS-D networks have several advantages. First, MS-D networks can produce accurate results with relatively few intermediate images and trainable parameters, enabling application to large images and effective training with relatively small training sets. Second, although a set of dilations d_i have to be chosen in advance, the network can learn which combinations of dilations to use during training, automatically adapting the network to each specific problem. As a result, the same hyperparameters can be used for a wide variety of problems, removing the need to perform a time-consuming trial-and-error search for hyperparameter values. Finally, all layers of an MS-D network are computed in the same way using the same set of standard machine learning operations, making MS-D networks easy to implement, train, and use in practice. These advantages make MS-D networks especially well suited for use in tomography, in which large images from a wide variety of application fields have to be processed, and where it is difficult to acquire large sets of training images. For a more detailed explanation of the MS-D network architecture,

verification of the above statements, and examples of other applications of MS-D networks, we refer to [22].

3. Results and Discussion

3.1. Setup

To investigate the performance of the Mixed-Scale Dense convolutional neural network architecture for improving reconstructed images of tomographic data with various types of limitations, we implemented the architecture in Python, using PyCUDA [31] to accelerate computationally costly operations by running them on Graphic Programming Units (GPUs). This implementation is similar to the one presented in [22], and has similar computational costs (shown in Figure S5 of [22]). As an example, processing a reconstructed image of 1024×1024 pixels with a 100-layer MS-D network takes around 200 ms with an NVidia GTX 1080 GPU (NVidia, Santa Clara, CA, USA). Since MS-D networks are able to automatically adapt to each problem, we are able to use the same network hyperparameters in all experiments: each network is 100 layers deep (excluding the input and output layer), and we use equally distributed dilations $d_i \in [1, 10]$ by setting the dilation of layer i to $d_i = 1 + (i \bmod 10)$. The resulting MS-D network has around 46 thousand trainable parameters, which are initialized in the same way as described in [22]. Networks are trained using the ADAM algorithm [32] with a batch size of a single 2D image, i.e., each batch consists of 1024^2 training pixels if $N = 1024$. The algorithm minimizes the ℓ_2 -norm between the network output and target images of the training set. An independent set of image pairs is used as a *validation set* to monitor network quality during training and provide a stopping criterion. In all experiments, the ℓ_2 -norm over the validation set, i.e., the validation error, is computed after every 100 gradient steps, and the training is stopped once no improvement to the validation error is found for 10,000 gradient steps. The network parameters that yielded the lowest validation error are saved as output of the training procedure. All computations were performed on either a workstation with a single NVidia GTX 1070 GPU (NVidia, Santa Clara, CA, USA) running CUDA 9.2 or a server with four NVidia GTX 1080 GPUs running CUDA 9.1.

3.2. Simulations

First, we studied the performance of the proposed MS-D network approach using simulated tomographic data to systematically study a wide variety of data limitations. We used two different types of randomly generated 3D objects: foam phantoms and rock phantoms. To generate a foam phantom, 100,000 randomly-placed non-overlapping spheres with varying sizes were removed from a cylinder of a single material. The resulting object, shown in Figure 3, is similar to real-world foams, which are typically difficult to accurately reconstruct from limited data. To generate a rock phantom, 5000 randomly-placed non-overlapping tetrahedrons with varying sizes were placed inside a cube of a single material, and each tetrahedron was randomly assigned one of four materials. An example of a rock phantom is shown in Figure 3. Three different objects were generated for both phantom types by choosing different random seeds. For each type, one object was strictly used for producing the *training set*, one object was strictly used for producing the *validation set*, and the final object was used as an independent test object. Tomographic projections of the objects were simulated using the ASTRA toolbox [33]. Each computed projection image consists of 1024×1024 pixels, and was generated by discretizing the phantom object on a $4096 \times 4096 \times 4096$ pixel grid, simulating projections on a 4096×4096 pixel detector, and downsampling each projection to a 1024×1024 pixel grid by averaging over 4×4 pixel blocks. Reconstructions were computed on a $1024 \times 1024 \times 1024$ pixel grid, from which individual 2D slices were used as input to the MS-D networks.

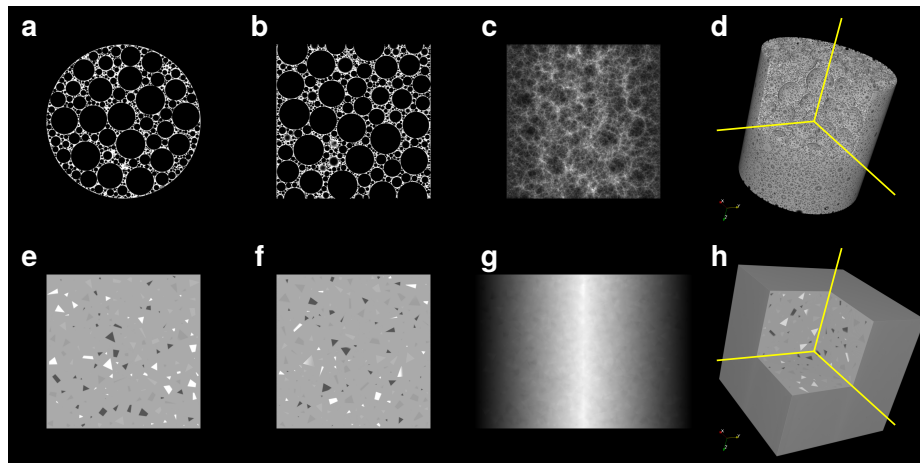


Figure 3. The two phantom types used for the simulations: the foam phantom (a–d); and the rock phantom (e–h), with: axial slices (a,e); sagittal slices (b,f); projection images at 45° (c,g); and 3D renderings (d,h), with box-shaped cutouts indicated by yellow lines.

In the following, we compare results of MS-D networks with results generated by popular tomographic reconstruction algorithms: the Filtered Backprojection algorithm (FBP) with a hann filter [4], the Simultaneous Iterative Reconstruction Technique (SIRT) [4], the SIRT algorithm with additional box constraints (i.e., pixel values are constraint between the minimum and maximum pixel values of the phantom), and Total Variation Minimization (TV-MIN) using the FISTA algorithm [5]. For each algorithm, we compare the algorithm output image with the FBP-hann reconstruction from 1024 noise-free projection images for the middle slice of the test object, and report two error metrics: the structural similarity index (SSIM) [34] and the root-mean-square error (RMSE). In the case of the SIRT reconstructions, we report results for the number of SIRT iterations that resulted in the smallest RMSE value, with a maximum of 2000 iterations. Similarly, for each TV-MIN reconstruction, we report results for the λ parameter (i.e., strength of the total variation term) and number of iterations that resulted in the smallest RMSE value, with a maximum of 500 iterations, and using the Nelder–Mead method to find the optimal λ value.

3.2.1. Limited Number of Projections

First, we investigated the performance of the MS-D network when only a limited number of projections are available. In each case, we trained an MS-D network to improve FBP-hann reconstructions from a limited number of noise-free projections over 180°, using FBP-hann reconstructions from 1024 noise-free projections over 180° as target images during training. In Figure 4, the RMSE and SSIM metrics are shown as a function of the number of available projections. For both phantom types, the MS-D networks produce images with significantly better error metrics compared with popular tomographic reconstruction algorithms, including Total-Variation minimization, an advanced regularized iterative algorithm. Compared with the FBP algorithm, post-processing images by using MS-D networks is able to reduce the number of projections eight-fold without a significant decline in error metric values. In Figure 5, reconstructed images of the middle slice of the foam test object are shown for 8, 16, 32, 64, and 128 projections. Note that, even with a highly limited number of projections, the MS-D network is able to reproduce the general shape of the scanned object, while traditional algorithms tend to produce blurry or streaky images that are not suitable for further analysis. In Figure A1, similar images for the rock test object are given, showing similar results.

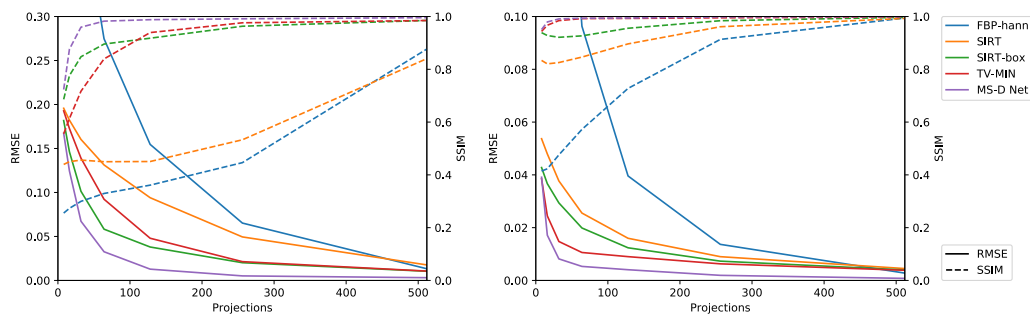


Figure 4. The RMSE (solid) and SSIM (dashed) error measure as a function of the number of projections for: the foam phantom (left); and the rock phantom (right). Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN).

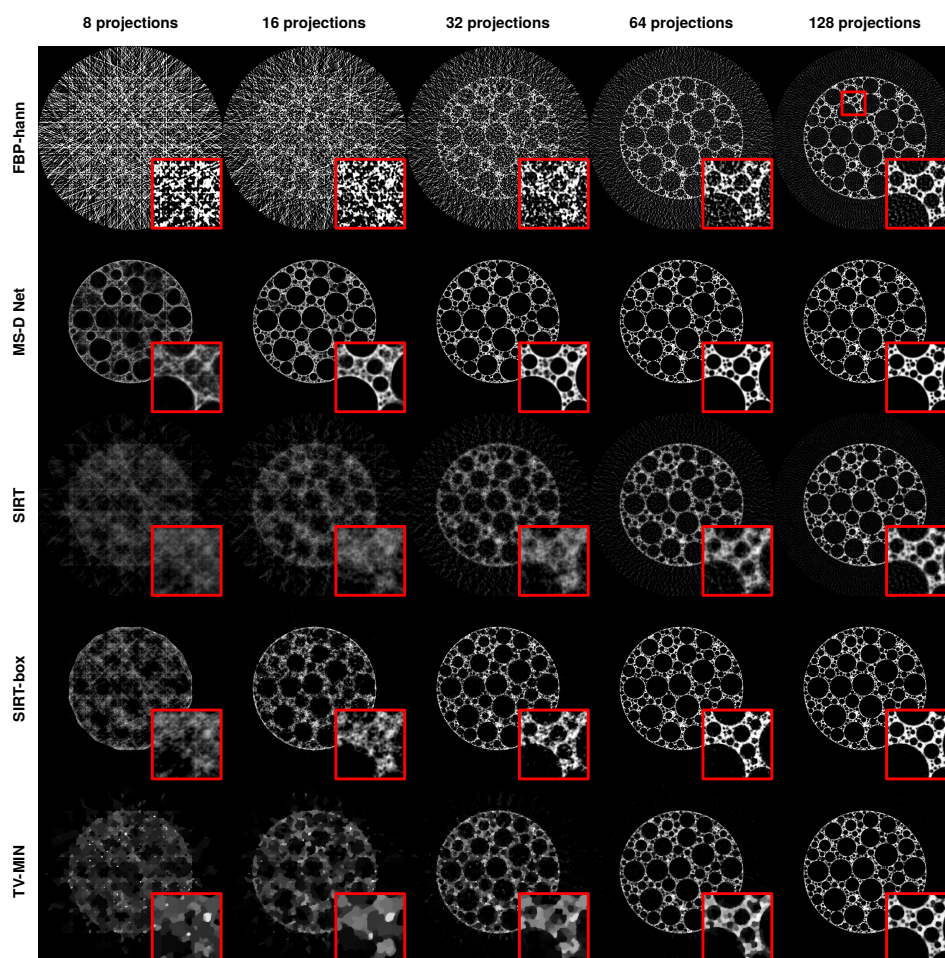


Figure 5. Reconstructions from a limited number of projections for the foam phantom. Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN). The input image of each MS-D network is the corresponding FBP-hann reconstruction shown in the top row. A small region indicated by the red square in the FBP-hann image is shown enlarged for each image.

3.2.2. Limited Exposure Time

Here, we investigated the performance of the MS-D network when the available projection data contains noise. In each case, we trained an MS-D network to improve FBP-hann reconstructions from

1024 noisy projections over 180° , using FBP-hann reconstructions from 1024 noise-free projections over 180° as target images during training. To generate noisy projections, Poisson noise was applied to the noise-free projections, with a parameter I_0 controlling the amount of noise, and a parameter γ controlling the amount of radiation absorbed by the sample. Specifically, the Poisson noise was applied in the following way: first, the noise-free projections were transformed to virtual photon counts using the Beer–Lambert law, with the background photon count set to I_0 . For each detector pixel, a new photon count was sampled from a Poisson distribution with the original photon count as the expected value. Finally, the resulting noisy photon counts were transformed back to noisy line integrals of the phantom.

We considered three cases for the γ parameter: one case in which the sample is highly absorbing (absorbing, on average, 97.5% of the incoming photons), one case in which the sample is low absorbing (absorbing, on average, 2.5% of the incoming photons), and one intermediate case (absorbing, on average, 50% of the incoming photons). In low absorbing samples, the applied Poisson noise is similar to Gaussian noise, while, in highly absorbing samples, non-linear effects are more pronounced. For each case and phantom type, we empirically chose a range of I_0 values that exhibited interesting results. The RMSE and SSIM metrics are given in Figure 6, showing similar results to Figure 4. In all cases, the MS-D networks are able to produce images with significantly better error metrics compared with traditional algorithms, especially for highly absorbing samples. Reconstructed images of the foam phantom are given for a few selected cases in Figure 7, showing that the MS-D network is able to produce accurate results, even from highly noisy data. Reconstructed images of the rock phantom are given in Figure A2.

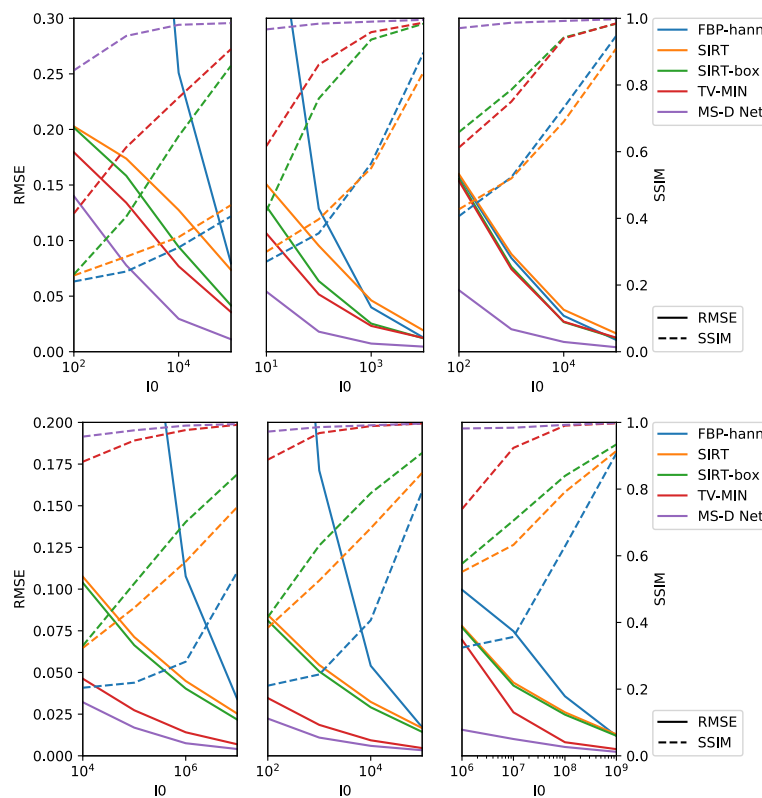


Figure 6. The RMSE (solid) and SSIM (dashed) error measure as a function of the amount of Poisson noise (I_0) for: the foam phantom (**top**); and the rock phantom (**bottom**). Given are results for 2.5% absorption (left), 50% absorption (middle), and 97.5% absorption (right), for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN).

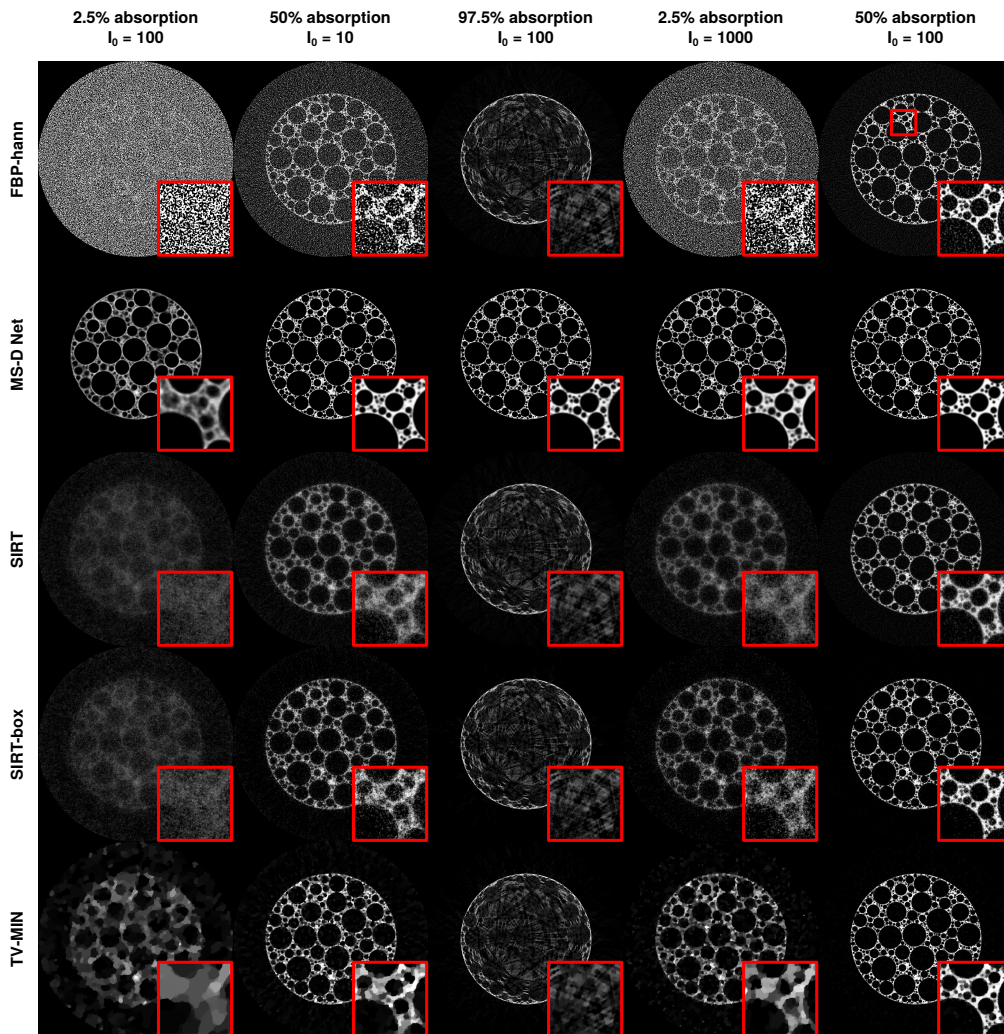


Figure 7. Reconstructions from projections with a limited exposure time for the foam phantom. Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN). The input image of each MS-D network is the corresponding FBP-hann reconstruction shown in the top row. A small region indicated by the red square in the FBP-hann image is shown enlarged for each image.

3.2.3. Limited Angular Range

Here, we investigated the performance of the MS-D network when projections are only available for a limited angular range (i.e., less than 180°). In each case, we trained an MS-D network to improve FBP-hann reconstructions from 256 noise-free projections over a limited angular range, using FBP-hann reconstructions from 1024 noise-free projections over 180° as target images during training. In Figure 8, the RMSE and SSIM metrics are shown as a function of the angular range of projections. Again, for both phantom types, the MS-D networks produce images with significantly better error metrics compared with popular tomographic reconstruction algorithms. In Figure 9, reconstructed images of the middle slices of both test objects are shown for a few selected angular ranges. Note that, even with a highly limited angular range, the MS-D network is able to reproduce the scanned object relatively well, while traditional algorithms tend to produce strong so-called missing wedge artifacts.

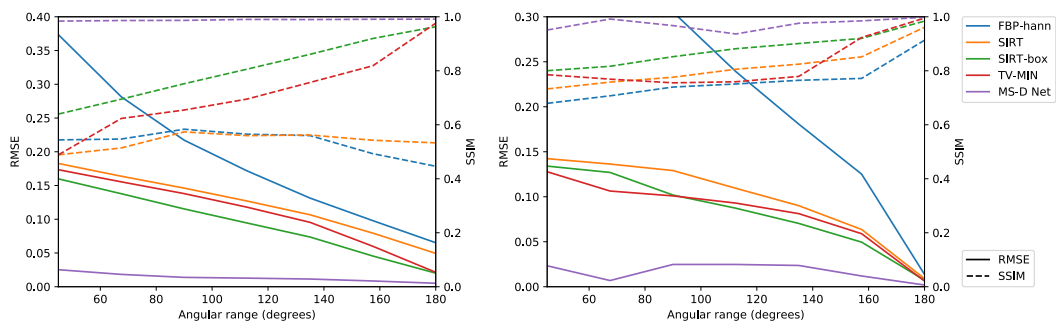


Figure 8. The RMSE (solid) and SSIM (dashed) error measure as a function of the angular range for: the foam phantom (**left**); and the rock phantom (**right**). Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN).

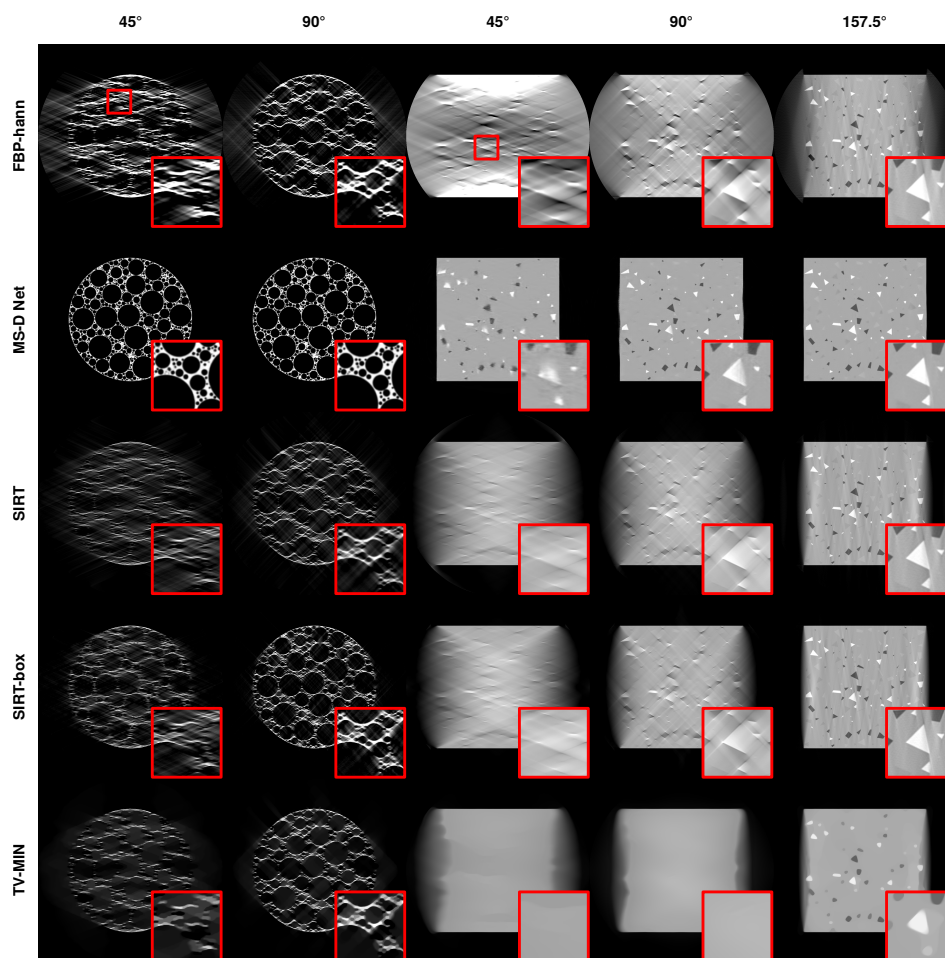


Figure 9. Reconstructions from a limited angular range for the foam and rock phantoms. Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN). The input image of each MS-D network is the corresponding FBP-hann reconstruction shown in the top row. A small region indicated by the red square in the FBP-hann images is shown enlarged for each image.

3.2.4. Quality of Training Images

In previous sections, we have assumed that high-quality images (i.e., FBP reconstructions using 1024 noise-free projections over 180°) are available for training the MS-D networks. In practice,

however, high-quality images to be used for training are likely to themselves contain some artifacts as well. In this section, we investigate the effect of the quality of the target images on the quality of the network outputs after training. We show the results of training networks to improve reconstructions from 32 projections (the middle column of Figure 5), using FBP reconstructions from 1024 noisy projections as target images. Specifically, we used data of the 2.5% absorption case of Section 3.2.2, of which reconstructions are shown in Figure 7, and varied the amount of noise to investigate the effect of the amount of noise in the target images on the quality of the network outputs. Results are shown in Figure 10. Note that, even with large amounts of noise in the target images, the MS-D network is still able to produce images that are similar to those obtained when training with noise-free target images. This suggests that, in practice, noise in the high-quality reconstructions is not likely to be an issue.

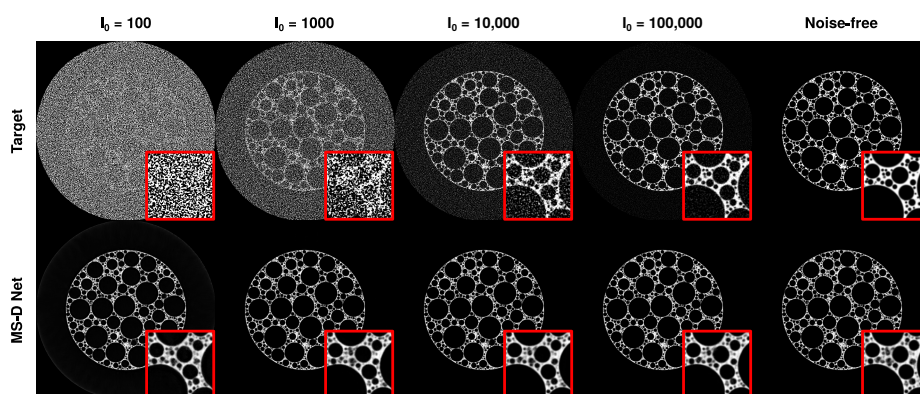


Figure 10. MS-D network output for reconstructions with 32 noise-free projections of the foam phantom. The networks were trained with noisy target images with various amounts of Poisson noise (I_0) and 2.5% absorption, of which representative images are shown in the top row. For comparison, the network output for noise-free target images is given as well.

3.2.5. Comparison with Other Networks

In this section, we compare results of the proposed MS-D network with other popular approaches for applying machine learning to tomographic images: the NN-FBP algorithm [14], and the FBPCConvNet network [19]. In both cases, we used code provided by the authors, which for the FBPCConvNet network consisted of an implementation in TensorFlow [35]. Furthermore, we used a similar training approach as in previous sections to train the different networks. In Figure 11, results are shown for improving reconstructions from 32 noise-free projections, for MS-D networks and the NN-FBP algorithm. Even though the NN-FBP reconstructions are more accurate than the FBP reconstructions, the MS-D network output is significantly more accurate than both. A possible reason for this is that the neural network underlying the NN-FBP algorithm only consists of a single layer, which prevents it from learning highly non-linear features.

For the FBPCConvNet results, we downsampled all input and target images to 512×512 pixels for both FBPCConvNet and MS-D networks, to make the comparison as fair as possible to FBPCConvNet, which was designed for processing images of that size. The hyperparameters for FBPCConvNet, e.g., depth and number of scaling operations, were chosen to be identical to those presented in [19], resulting in a network with around 31 million parameters that have to be learned, while the MS-D network has around 46 thousand learnable parameters. Using an NVidia GTX 1070 GPU, processing a single 512×512 pixel image with the FBPCConvNet network takes around 114 ms, and processing a single image with the MS-D network takes around 66 ms. We compare results for various data limitations: reconstructing with data from 16 projections, from 256 projections over 45° range, from 1024 noisy projections (with 2.5% absorption and $I_0 = 100$), and from 32 projections with noisy targets during training (1024 projections with 50% absorption and $I_0 = 10$ for foam phantoms and $I_0 = 100$ for rock phantoms). RMSE and SSIM metrics of the middle slice of the test object are given in Table 1. The results show that, in some cases, the FBPCConvNet network and MS-D network produce images

with similar error metrics. In other cases, however, the MS-D network is able to produce images with significantly better error metrics than the FBPCnvNet network, especially when noisy projections are involved. This suggests that the FBPCnvNet networks have a higher risk of overfitting to the training set due to their large amount of trainable parameters, which was also observed in [22]. Even in cases where FBPCnvNet networks produce equivalent results to MS-D networks, the FBPCnvNet networks require significantly larger amounts of computer memory, which can prohibit their use in practice. In Figure 12, the amount of computer memory required to store all intermediate images is shown as a function of the input image size for both FBPCnvNet networks and MS-D networks. These results show that, for typical image sizes in tomography, FBPCnvNet networks can be difficult to fit in GPU memory, making practical and efficient implementation difficult.

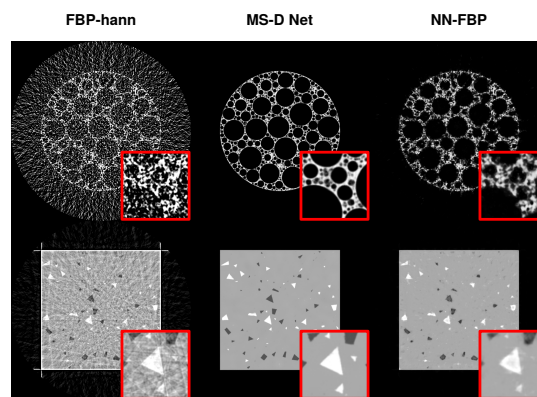


Figure 11. Reconstructions from 32 noise-free projections, using FBP with the hann filter, the proposed MS-D network, and the NN-FBP algorithm.

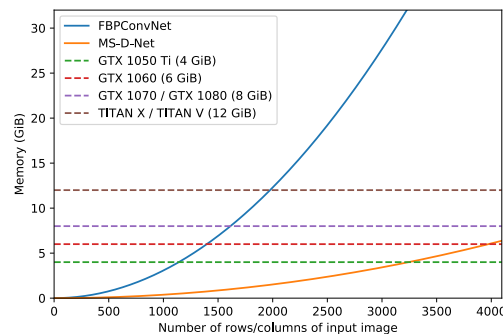


Figure 12. The required memory to store all intermediate images of a FBPCnvNet network [19] and a MS-D network with 100 layers [22] as a function of the size of the input image, shown with the maximum usable memory of several NVidia GPUs that are popular for machine learning. Note that, in practice, training these networks typically requires a multiple of the given memory, further limiting the size of images that can be successfully trained.

Table 1. Comparison between the RMSE (top) and SSIM (bottom) of images of FBPCnvNet and the proposed MS-D Network architecture. For each column, the best results are shown in bold.

Method	Metric	16 Projections		45° Range		Noisy Target		Noisy Input
		FOAM	ROCK	FOAM	ROCK	FOAM	ROCK	FOAM
FBPCnvNet	RMSE	0.110	0.017	0.046	0.012	0.072	0.011	0.103
	SSIM	0.734	0.963	0.982	0.981	0.630	0.920	0.782
MS-D-Net	RMSE	0.100	0.019	0.039	0.013	0.065	0.011	0.099
	SSIM	0.910	0.957	0.985	0.980	0.726	0.930	0.907

3.3. Experimental Data

As a final test, we applied the proposed MS-D network to a real-world experimental dataset. We used the fatigue-corrosion data sets from TomoBank [36], which were acquired at the Advanced Photon Source synchrotron at Argonne National Laboratory. To acquire this data, Peak-aged Al 7075 samples were corrosion-pitted by soaking in exposed 3.5 wt. % NaCl solution for fifteen days (360 h). The samples were fatigue tested in situ in solution using synchrotron X-ray tomography to analyze the fatigue crack initiation and growth characteristics. 25 tomographic scans were acquired after increasing amounts of fatigue cycles from 750 to 14,346. In each scan, 1500 projections were acquired with a 2160×2560 pixel detector. The acquired data was processed with the TomoPy software package [37] and reconstructed using the gridrec algorithm [38], which is a computationally efficient approximation of the FBP algorithm, resulting in 2160 2D slices of 2560×2560 pixels per scan. For training, we used reconstructions of two tomographic scans, one after few fatigue cycles (750) and one after many fatigue cycles (14,300), using reconstructions from 150 projections over 180° as network inputs, and reconstructions from 1500 projections over 180° as training targets. To produce a validation set, 100 slices out of the 4320 available slices were randomly selected and removed from the training set. In Figure 13 results are shown for the middle slice of a tomographic scan after an intermediate amount of fatigue cycles (8500). Note that the standard gridrec reconstruction from 150 projections contains severe noise which prohibits further analysis. On the other hand, the MS-D network is able to produce a reconstructed image from 150 projections that has an equivalent quality to a standard reconstruction from 1500 projections. In other words, during the experiment, the intermediate scans could have been acquired with 10 times fewer projections, reducing acquisition time and X-ray dose, without a significant loss in reconstruction quality. In fact, the noise in the MS-D network output is less severe than the noise in the gridrec reconstruction from 1500 projections, which is also observed in Section 3.2.4.

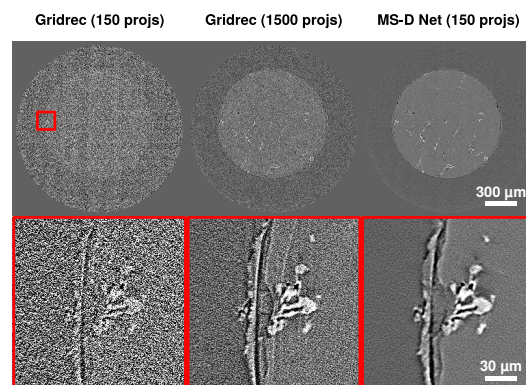


Figure 13. Reconstructions of an Al 7075 sample [36] using: the gridrec algorithm with 150 projections, the gridrec algorithm with 1500 projections, and the MS-D network with 150 projections. The input to the MS-D network is the gridrec reconstruction shown in the left column, and the network was trained with data from the same sample at earlier and later fatigue cycles. A small region indicated by the red square is shown enlarged in the bottom row.

4. Conclusions

In this paper, we present the application of the Mixed-Scale Dense neural network architecture [22] to improving tomographic reconstructions from limited data. Various types of data limitations were investigated, including a low number of projections and a limited angular range. The proposed approach is computationally efficient, achieves accurate results even with severely limited data, and can be applied to the large images that are common to tomographic imaging. Compared with existing tomographic reconstruction algorithms, MS-D networks produce more accurate reconstruction images than both direct and iterative algorithms, and are significantly faster at producing them than iterative algorithms. Compared with existing machine learning approaches based on encoder–decoder networks, the MS-D network architecture requires fewer trainable parameters, leading to more accurate results for several types of limited data. Furthermore, MS-D networks required fewer intermediate images, enabling the processing of significantly larger images. Finally, MS-D networks are able to automatically adapt to different problems, allowing the same network to be successfully applied to widely different data limitations and object types.

In the experiments for this paper, we only show results for parallel-beam tomographic data. However, since the approach is based on post-processing reconstructed images, it can be directly applied to reconstructions from other geometries, e.g., cone-beam tomography. How well the MS-D network architecture performs in such cases is subject to further research. In addition, the proposed approach uses a 2D neural network to improve 2D reconstructed slices. A full 3D network that improves entire 3D reconstruction volumes can be expected to further improve reconstruction quality by enabling capturing 3D object features. To achieve this goal, new 3D networks have to be developed that can efficiently process the large 3D images that are produced by tomographic scanners. On the other hand, the results of this paper show that, with 2D Mixed-Scale Dense convolutional neural networks, a significant improvement over existing methods can be achieved.

Author Contributions: Conceptualization, D.M.P., K.J.B., and J.A.S.; Methodology, D.M.P. and J.A.S.; Software, D.M.P.; Writing—Original Draft Preparation, D.M.P.; and Writing—Review and Editing, K.J.B. and J.A.S.

Funding: Financial support was provided by The Netherlands Organisation for Scientific Research (NWO), project numbers 639.073.506 and 016.Veni.192.235; and the Center for Advanced Mathematics for Energy Research Applications (CAMERA), jointly funded by Advanced Scientific Computing Research (ASCR) and Basic Energy Sciences (BES) branches of the Office of Science, US Department of Energy.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

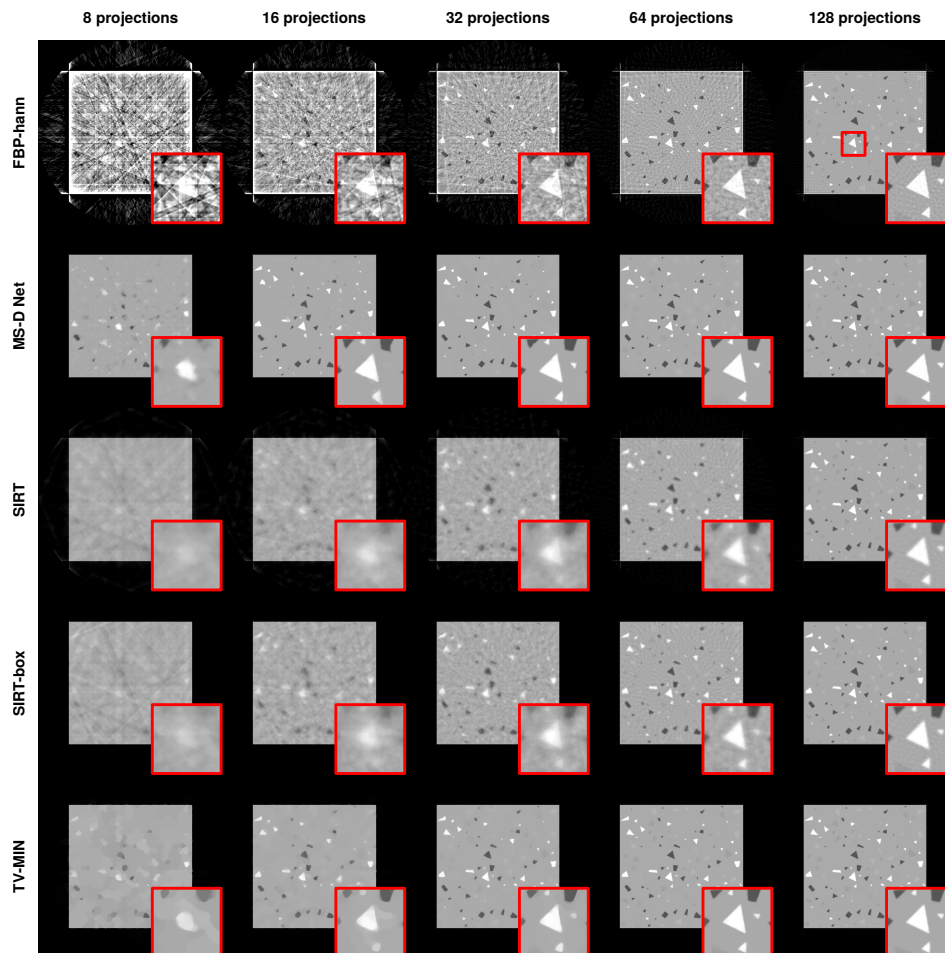


Figure A1. Reconstructions from a limited number of projections for the rock phantom. Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN). The input image of each MS-D network is the corresponding FBP-hann reconstruction shown in the top row. A small region indicated by the red square in the FBP-hann image is shown enlarged for each image.

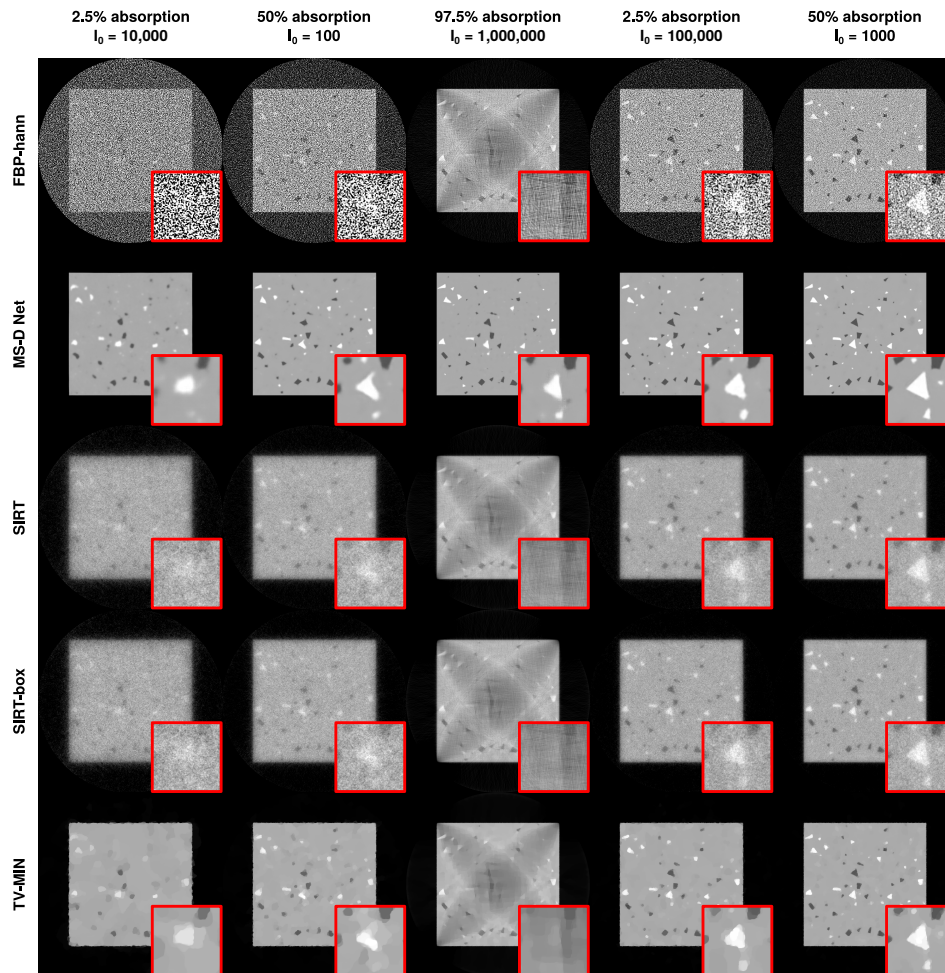


Figure A2. Reconstructions from projections with a limited exposure time for the rock phantom. Results are shown for FBP with the hann filter (FBP-hann), the proposed MS-D network (MS-D Net), the SIRT method, the SIRT method with box constraints (SIRT-box), and Total Variation minimization using the FISTA method (TV-MIN). The input image of each MS-D network is the corresponding FBP-hann reconstruction shown in the top row. A small region indicated by the red square in the FBP-hann image is shown enlarged for each image.

References

1. Donoghue, P.C.; Bengtson, S.; Dong, X.P.; Gostling, N.J.; Hultgren, T.; Cunningham, J.A.; Yin, C.; Yue, Z.; Peng, F.; Stampanoni, M. Synchrotron X-ray tomographic microscopy of fossil embryos. *Nature* **2006**, *442*, 680–683. [[CrossRef](#)] [[PubMed](#)]
2. Metscher, B.D. MicroCT for comparative morphology: Simple staining methods allow high-contrast 3D imaging of diverse non-mineralized animal tissues. *BMC Physiol.* **2009**, *9*, 11. [[CrossRef](#)] [[PubMed](#)]
3. Midgley, P.A.; Dunin-Borkowski, R.E. Electron tomography and holography in materials science. *Nat. Mater.* **2009**, *8*, 271. [[CrossRef](#)] [[PubMed](#)]
4. Kak, A.C.; Slaney, M. *Principles of Computerized Tomographic Imaging*; Society of Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001.
5. Beck, A.; Teboulle, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* **2009**, *18*, 2419–2434. [[CrossRef](#)] [[PubMed](#)]
6. Lovric, G.; Barré, S.F.; Schittny, J.C.; Roth-Kleiner, M.; Stampanoni, M.; Mokso, R. Dose optimization approach to fast X-ray microtomography of the lung alveoli. *J. Appl. Crystallogr.* **2013**, *46*, 856–860. [[CrossRef](#)] [[PubMed](#)]

7. Sipila, H. Moving-object computer tomography for luggage inspection. In Proceedings of the International Society for Optics and Photonics, Applications of Signal and Image Processing in Explosives Detection Systems, Boston, MA, USA, 1 April 1993; Volume 1824, pp. 39–41.
8. Xiao, X.; Liu, H.; Wang, L.; De Carlo, F. Density measurement of samples under high pressure using synchrotron microtomography and diamond anvil cell techniques. *J. Synchrotron Radiat.* **2010**, *17*, 360–366. [[CrossRef](#)] [[PubMed](#)]
9. Ritschl, L.; Bergner, F.; Fleischmann, C.; Kachelrieß, M. Improved total variation-based CT image reconstruction applied to clinical data. *Phys. Med. Biol.* **2011**, *56*, 1545. [[CrossRef](#)] [[PubMed](#)]
10. Bicer, T.; Gürsoy, D.; De Andrade, V.; Kettimuthu, R.; Scullin, W.; De Carlo, F.; Foster, I.T. Trace: A high-throughput tomographic reconstruction engine for large-scale datasets. *Adv. Struct. Chem. Imaging* **2017**, *3*, 6. [[CrossRef](#)] [[PubMed](#)]
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
12. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin, Germany, 2015; pp. 234–241.
13. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [[CrossRef](#)] [[PubMed](#)]
14. Pelt, D.M.; Batenburg, K.J. Fast Tomographic Reconstruction From Limited Data Using Artificial Neural Networks. *IEEE Trans. Image Process.* **2013**, *22*, 5238–5251. [[CrossRef](#)] [[PubMed](#)]
15. Bladt, E.; Pelt, D.M.; Bals, S.; Batenburg, K.J. Electron tomography based on highly limited data using a neural network reconstruction technique. *Ultramicroscopy* **2015**, *158*, 81–88. [[CrossRef](#)] [[PubMed](#)]
16. Boubilil, D.; Elad, M.; Shtok, J.; Zibulevsky, M. Spatially-adaptive reconstruction in computed tomography using neural networks. *IEEE Trans. Med. Imaging* **2015**, *34*, 1474–1485. [[CrossRef](#)] [[PubMed](#)]
17. Yang, X.; De Carlo, F.; Phatak, C.; Gürsoy, D. A convolutional neural network approach to calibrating the rotation axis for X-ray computed tomography. *J. Synchrotron Radiat.* **2017**, *24*, 469–475. [[CrossRef](#)] [[PubMed](#)]
18. Chen, H.; Zhang, Y.; Zhang, W.; Liao, P.; Li, K.; Zhou, J.; Wang, G. Low-dose CT via convolutional neural network. *Biomed. Opt. Express* **2017**, *8*, 679–694. [[CrossRef](#)] [[PubMed](#)]
19. Jin, K.H.; McCann, M.T.; Froustey, E.; Unser, M. Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Trans. Image Process.* **2017**, *26*, 4509–4522. [[CrossRef](#)] [[PubMed](#)]
20. Matos, F.A.; Ferreira, D.R.; Carvalho, P.J.; Contributors, J. Deep learning for plasma tomography using the bolometer system at JET. *Fusion Eng. Des.* **2017**, *114*, 18–25. [[CrossRef](#)]
21. Yang, X.; De Andrade, V.; Scullin, W.; Dyer, E.L.; Kasthuri, N.; De Carlo, F.; Gürsoy, D. Low-dose X-ray tomography through a deep convolutional neural network. *Sci. Rep.* **2018**, *8*, 2575. [[CrossRef](#)] [[PubMed](#)]
22. Pelt, D.M.; Sethian, J.A. A mixed-scale dense convolutional neural network for image analysis. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 254–259. [[CrossRef](#)] [[PubMed](#)]
23. Adler, J.; Öktem, O. Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **2018**, *37*, 1322–1332. [[CrossRef](#)] [[PubMed](#)]
24. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
25. Farquhar, T.; Chatziioannou, A.; Chinn, G.; Dahlbom, M.; Hoffman, E. An investigation of filter choice for filtered back-projection reconstruction in PET. *IEEE Trans. Nucl. Sci.* **1998**, *45*, 1133–1137. [[CrossRef](#)]
26. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
27. Pan, X.; Sidky, E.Y.; Vannier, M. Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Probl.* **2009**, *25*, 123009. [[CrossRef](#)] [[PubMed](#)]
28. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
29. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
30. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.

31. Klöckner, A.; Pinto, N.; Lee, Y.; Catanzaro, B.; Ivanov, P.; Fasih, A. PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation. *Parallel Comput.* **2012**, *38*, 157–174. [[CrossRef](#)]
32. Kingma, D.P.; Ba, L. ADAM: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
33. van Aarle, W.; Palenstijn, W.J.; De Beenhouwer, J.; Altantzis, T.; Bals, S.; Batenburg, K.J.; Sijbers, J. The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy* **2015**, *157*, 35–47. [[CrossRef](#)] [[PubMed](#)]
34. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
35. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
36. De Carlo, F.; Gürsoy, D.; Ching, D.J.; Batenburg, K.J.; Ludwig, W.; Mancini, L.; Marone, F.; Mokso, R.; Pelt, D.M.; Sijbers, J.; et al. TomoBank: A tomographic data repository for computational X-ray science. *Meas. Sci. Technol.* **2018**, *29*, 034004. [[CrossRef](#)]
37. Gürsoy, D.; De Carlo, F.; Xiao, X.; Jacobsen, C. TomoPy: A framework for the analysis of synchrotron tomographic data. *J. Synchrotron Radiat.* **2014**, *21*, 1188–1193. [[CrossRef](#)] [[PubMed](#)]
38. Marone, F.; Stampanoni, M. Regridding reconstruction algorithm for real-time tomographic imaging. *J. Synchrotron Radiat.* **2012**, *19*, 1029–1037. [[CrossRef](#)] [[PubMed](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).