Sparse reconstruction for CT imaging with deep convolution neural network: an investigation study

Cédric Bélanger
Département de physique, de génie physique et d'optique
Université Laval
Québec, Canada
cedric.belanger.2@ulaval.ca

Maxence Larose
Département de physique, de génie physique et d'optique
Université Laval
Québec, Canada
maxence.larose.1@ulaval.ca

Rémy Bédard
Département d'informatique et de génie logiciel
Université Laval
Québec, Canada
remy.bedard.1@ulaval.ca

Abstract

Sparse-view computed tomography (CT) is increasingly used in the medical field to help reduce radiation doses given to patients needing tissue imaging. Since the reconstruction of projection images from the sparse data with conventional algorithms creates artifacts in the resulting images, deep learning image processing methods seem interesting to potentially replace these algorithms or complement them by removing the artifacts they produce. This paper examines several deep convolutional network architectures, all U-Net variants, on the task of retrieving phantom images of breast tissue from 2D slices reconstructed by a filtered back projection (FBP) algorithm. Of the studied models, Breast U-Netv2 performed the best in testing, but the traditional U-Net achieved comparable results (0.5% decrease in accuracy) at much lower computational cost (about haft the training and prediction time). The best performing architectures were also combined in ensemble methods, improving the accuracy by at least 7% compared to all single models. The model's performance was comparable to the published literature.

1 Introduction

Computed tomography (CT) imaging modalities have evolved and gained popularity in the medical field since the 1980s because of many technical advances in CT scanner design and performance [14]. CT imaging is also widely used in medicine because of its relatively low cost, versatility and good soft tissue imaging [1]. One drawback of CT imaging is the radiation dose that is given to the patient to acquire images of their tissues. For instance, the typical effective dose for a chest CT is $6.2\,\mathrm{mSv}$ [2]. Motivated by the ALARA (As Low As Reasonably Achievable) principle [9], it is always desirable to decrease the radiation dose when the image quality is not compromised. While a recent study argued that this principle is outdated and erroneous [12], the ALARA principle is still of interest and widely used in medical imaging. One way to address this issue is to reduce the number of projections during the CT scan. However, with less projections, recovering the images with traditional algorithms such as filtered back projection (FBP) lead to streaking artifacts in the images. Therefore, numerous studies have investigated numerical solutions to recover high quality images from fewer projections.

Sparse CT reconstruction is an active field of research which aims to recover dense CT scan images from sparse representation of the data. Compressed sensing [5] and iterative methods were studied to solve this problem by attempting to minimize the total variation (TV) [4]. The minimization of the TV, essentially a sum of coefficients from a discrete gradient transform, was often used in few-view or dose reduction efforts in CT reconstruction [4]. Traditionally, these methods are slow when implemented on CPU, as they require tens of minutes to hours for the reconstruction [4].

Preprint. Under review.

When implemented on graphics processing units (GPU), the reconstruction can be achieved within one minute [4]. Recently, deep convolution neural network (CNN) approaches have shown great success in image processing, such as classification [11, 16, 8], segmentation [13], and image denoising [19]. In the literature, various studies successfully implemented CNNs to solve the sparse CT reconstruction problem [10, 7]. The main advantage of CNNs for sparse CT reconstruction is that, when executed on GPU, they are sufficiently fast to be useful in medicine. However, a recent study argued that CNNs could not achieve the accuracy of the TV approach [15].

In this work, we aim to address the question of whether the sparse CT reconstruction problem can be solved using deep CNN architectures. First, a brief overview of CT imaging is made. Then, the problem to solve is briefly described and the network architectures and experimental setup used in this study are presented. At the end, metric comparisons between the different CNNs approaches are made.

2 Computed tomography

CT imaging is based on the acquisition of different projections (or beam's eye views) of the same object (in this case the patient) with a poly-energetic continuous X-ray photon beam [14]. To acquire an image of a patient, an X-ray tube and a photon detector are rotated around them to acquire different projections (or views). The attenuation of the photons, which follows the Beer-Lambert law, depends on the energy of the photons, the composition of the tissues and the electron density of the tissues [14]. Therefore, different tissue compositions will yield more or less photon attenuation. The attenuation information (or the signal) of each projection is measured by a detector, which is composed of an array of semiconductors. Hence, one projection corresponds to the measurement of the photon attenuation given a single view. The signal of each projection is collected by the array of semiconductors and stored in an image called a sinogram (see Figures 1a and 1b). The tissue composition of the patient can be reconstructed in a 2D or 3D volumetric image from the sinogram by using the filtered back projection algorithm (FBP) [14] (see Figures 1c and 1d). The quality of a CT image depends on the tube current, tube voltage, filter composition and number of projections [2]. This study focuses on reducing the number of projections to reduce the radiation dose given to the patient while maintaining high quality images.

3 CNN-based sparse CT reconstruction

The problem to solve is to find a function $f(x)$ that output an image prediction $\hat{y}$ of the ground truth image y given an image x with sparse representation of y. In a CNN approach, a convolutional network is trained to learn $f(x)$ by minimizing the differences between $\hat{y}$ and y, given thousands of images. In this study, the ground truth images consisted of 2D single slices of breast CT phantoms as illustrated in Figure 1e. The breast phantoms (BP) consisted of modeled fat, fibroglandular tissue, skin tissue, and microcalcification-like objects [15]. The phantoms were generated numerically as described in Sidky et al. [15]. The size of the BP images was 512×512 pixels. The sparse sinogram of the BP were obtained by simulating 128 projections of a divergent fan beam CT geometry with a single row detector of 512 pixels (Figure 1a). A dense sinogram would require at least 512 projections (Figure 1d) to achieve high quality FBP reconstructions (Figure 1d), which is 4 times more projections than the sparse sinogram (Figure 1a). The images x that were given as input to the networks were obtained by FBP reconstruction (512×512 pixels as shown in Figure 1c) using the sparse sinograms. For simplicity, these images will be referred to as FBP-128 for the rest of this article. The aim of the CNNs is to recover the breast phantom image (Figure 1e) from the FBP-128 image (Figure 1c).

4 Materials and methods

4.1 Network architectures

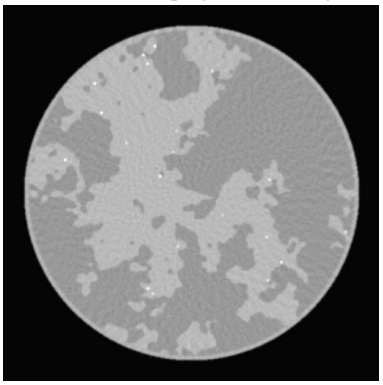
All network architectures investigated have the same base of encoder and decoder blocks with skip connections between corresponding encoder and decoder layers, as shown in Figure 2. The following sub-sections present the details of each model's encoder and decoder components. It should be noted that for each implementation, the input channel size was one, since the BP and FBP-128 images were encoded in grayscale.

4.1.1 U-Net

A U-Net is a well-known network architecture used for image segmentation tasks [13]. It has also shown great potential in solving the sparse CT reconstruction problem [15]. In this work, a U-Net (referred to as U-Net Original in this work) as described in Ronneberger et al. [13] was implemented. The code was borrowed from a Github

(a) 128 projections sinogram (sparse)

(b) 512 projections sinogram (dense)



(c) FBP-128 (sparse)

(d) FBP-512 (dense)

(e) Breast phantom (GT)

Figure 1: Illustration of the sparse and dense sinograms (a and b respectively) of a breast phantom (e) and the corresponding FBP reconstructions (c and d). The images were generated by Leonardo Di Schiavi Trotta.

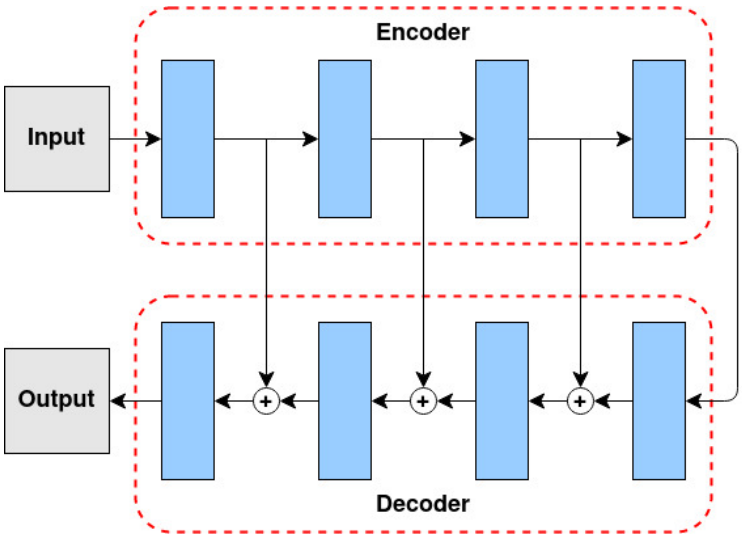


Figure 2: Base network architecture of a U-Net model.

repository[1]. The architecture of the U-Net Original, similar to the diagram in Figure 2, consisted of an encoder with five downsampling layers (using max pooling) with skip connections and a decoder of five upsampling layers (using bilinear interpolation). Each layer consisted of two 3×3 convolutions. After each convolution, a batch normalisation layer and a rectified linear unit (ReLU) activation were used. Each 3×3 convolution were zero padded to preserve the image size. The number of channels at each encoder and decoder depth were 32, 64, 128, 256, and 512. For more details on the implementation, we refer the reader to the U-Net paper [13].

[1]https://github.com/milesial/Pytorch-UNet

For the purpose of investigation, this network was trained in different ways to compare the achievable performances. Three networks were trained, namely the U-Net Original, the U-Net AUG and the U-Net DIFF. The U-Net Original uses the base images whereas the U-Net AUG was trained with augmented images additionally (see section 4.5). The U-Net DIFF used the differences between the FBP-128 images and the breast phantom BP images as targets (see Figure 9). This network must therefore learn the map of noise and artefacts present in the image, which could prove to be easier to do than to predict the breast phantom directly.

As a comparison to this U-Net, a pretrained U-Net was also investigated. The U-Net model from the Segmentation Models Pytorch (SMP) library was used because this library offers many encoders and encoder initialization weight options. [2]. The chosen encoder was the resnet34 and the encoder weights were pretrained using imagenet. As the imagenet data consists of RGB images, preprocessing must be applied to the grayscale images that we have. The preprocessing is directly implemented in the SMP library and allows to extend the gray image on 3 channels. Two networks were trained using SMP library, namely SMP U-Net and Pretrained SMP U-Net. The only difference between these networks is that the encoder parameters were frozen with the pre-trained SMP U-Net while they were updated during training with the SMP U-Net.

4.1.2 Dense U-Net

Based on a study on sparse-view CT reconstruction [21], a Dense U-Net is a type of U-Net of which the main feature is the use of Dense Blocks in the encoder. A Dense Block is a group of convolutional layers, the ones used here consisting of two sets of 3×3 convolution, batch normalisation and ReLU activation, one after the other. The main difference between the Dense U-Net and other similar architectures (such as a standard U-Net or Residual U-Net [20]) is the connections between the layers in the Dense Blocks. While a Residual network propagates previous information with identity connections between the input of a layer and the input of the subsequent one [8], which are then added together, the inputs of all previous layers of a Dense Block are concatenated with the input of the next layer. This approach permits the reuse of previously discovered features. Figure 3 shows an example of a Dense Block with three layers. The network architecture used in this work had an encoder of four Dense Blocks separated by 2×2 max pooling operations. The decoder had four layers consisting of two sets of 3×3 convolution, batch normalisation and Leaky ReLU [18], applied to the concatenation (instead of addition) of the upsampled output of the previous layer and of the output of the corresponding Dense Block from the encoder. The upsampling was done by nearest neighbor and the number of channels at each encoder and decoder layer were 32, 64, 128, 256, and 512. The code for this architecture was borrowed from a Github repository[3].

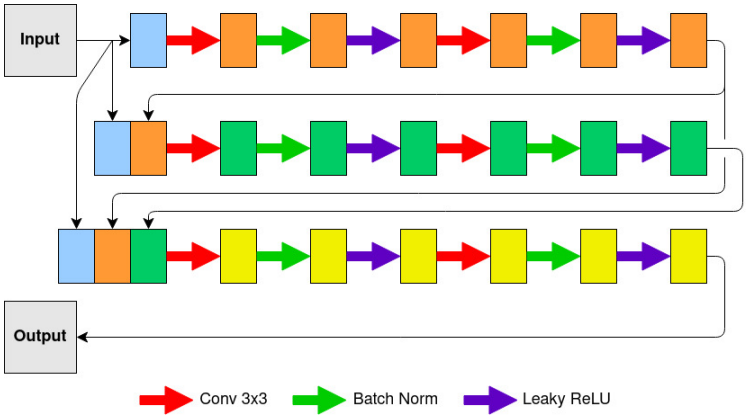


Figure 3: Example of a Dense Block with 3 layers.

4.1.3 Inception U-Net

Inspired by GoogLeNet's inception modules [16], an inception U-Net was also investigated. The inception U-Net follows a similar architecture as described in section 4.1.1. Instead of using 3×3 convolutions for each convolution block, as used in the traditional U-Net, the convolution blocks were customized to allow 3×3 and 5×5 convolutions, followed by a concatenation. In addition, each double convolution block used a residual 3×3 convolution as it showed great empirical results in a U-Net architecture [20]. Hence, we make the assumption that the network can learn by itself

[2]https://smp.read thedocs.io/en/latest/

[3]https://github.com/4uiiurz1/pytorch-nested-unet/

which convolution operations are the most relevant to minimize the loss. The illustration of an inception block in the encoder part of the Inception U-Net is shown in Figure 4. In the Inception U-Net, the upsampling was done by nearest neighbor and the number of channels at each encoder and decoder layer were 32, 64, 128, 256, and 512.

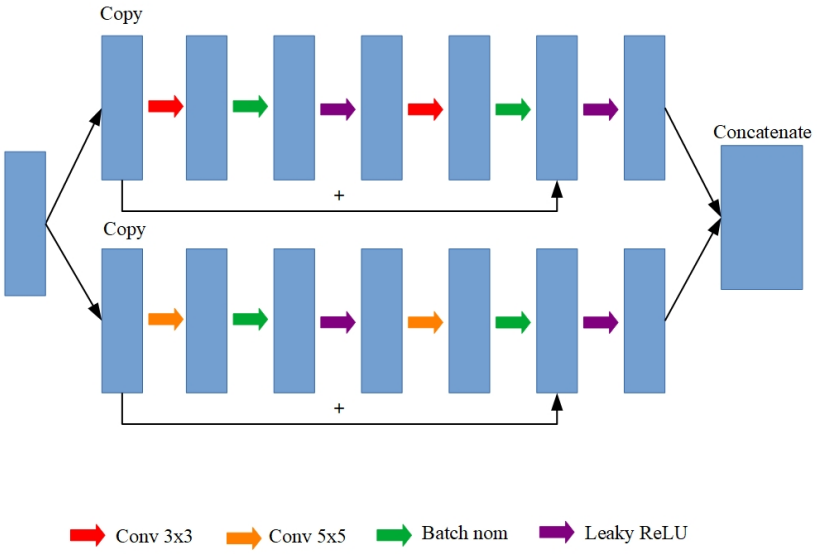


Figure 4: Illustration of an inception block in the encoder part of the Inception UNet.

4.1.4 Pretrained RED-CNN

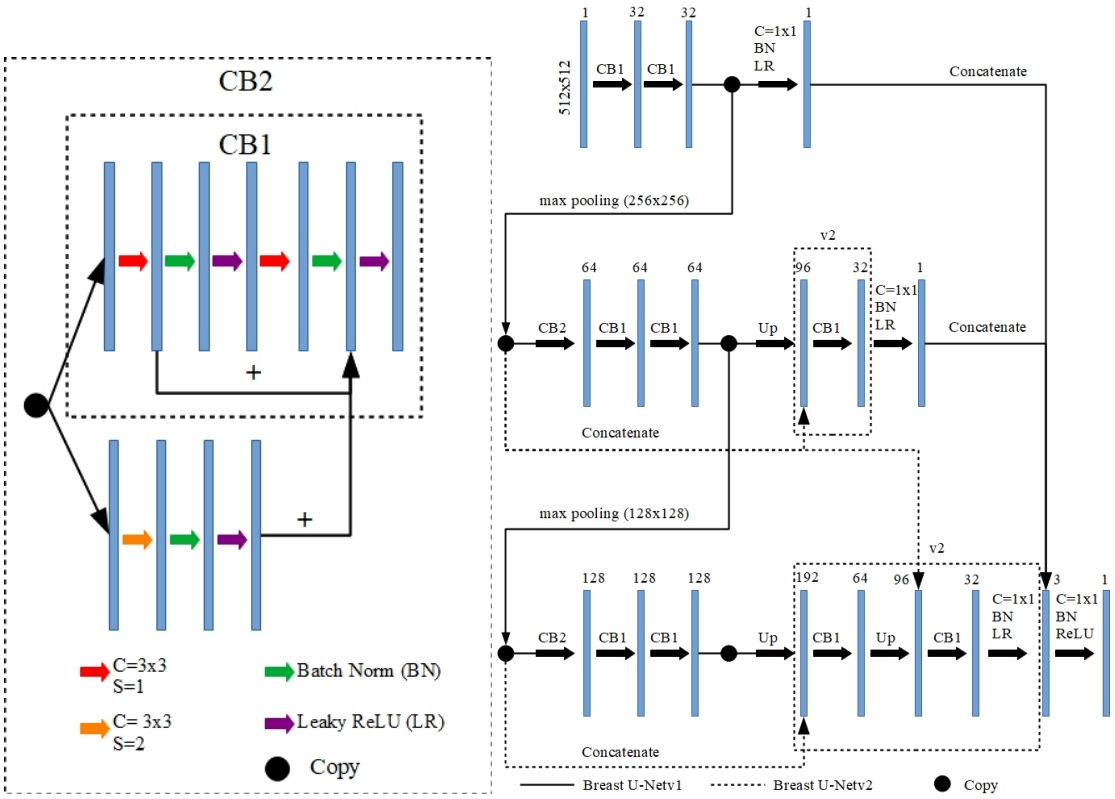
A RED-CNN, or Residual Encoder-Decoder CNN, introduced by Chen et al. [3], has the same architecture as a regular U-Net, but without max pooling operations reducing feature sizes. This network was studied since it was used in [3] for image denoising for low-dose CT imaging, which is a problem similar to ours. The architecture we used consisted of an encoder with five layers of 5×5 convolutions followed by a regular ReLU activation and a decoder with five layers of 5×5 transposed convolutions followed by a ReLU. Skip connections after each two layers of the encoder propagated the outputs, which were added to the corresponding decoder layers' inputs. The number of channels in the encoder and decoder sections was 96. Pretrained parameters obtained from CT images in Chen et al. Github repository were used[4], and the model was fine-tuned during training. While this network was used on patch CT images rather than on full CT images [3], we hypothesised that the pretrained parameters should be close to a good solution and that the architecture is also relevant when using full CT images as it was done in this study.

4.1.5 Breast U-Net

The Breast U-Netv2 implementation is based on a network (referred to as Breast U-Netv1 here) used for artifact removal in breast CT imaging [6]. In this article, the authors claimed that adding more layers to the encoder would result in disappearing features representing fibroglandular tissue composition. Furthermore, the authors claimed that adding convolutional layers in the decoder would reduce the impact of sparse fibroglandular tissue, thus resulting in an over-estimation of dense fibroglandular tissues. Preliminary results showed that their method could be improved by adding convolutional layers in the decoder. Therefore, our Breast U-Netv2 used additional convolutional layers with skip connections in the decoder as shown in the dashed v2 box in Figure 5b. Two residual convolutional blocks (Figure 5a) were used in the Breast U-Net architecture (Figure 5b). The first one (CB1 in Figure 5a) is similar to the convolutional blocks used in the Inception U-Net. In the second block (CB2 in Figure 5a), a 3x3 convolution with a stride of two was added to CB1 before the final Leaky ReLU activation in CB1.

The Breast U-Netv1 described in Ghazi et al. [6] was also implemented to test the hypothesis of the effect of the convolutions in the decoder part. This architecture did not use the v2 convolutional blocks in the decoder. Instead,

[4]https://github.com/SSinyu/RED-CNN



(a) (b)

Figure 5: Illustration of the (a) convolution blocks used in the (b) Breast U-Netv2 architecture.

the architecture used an upsampling factor of 2 (bilinear interpolation) after second encoder block and an upsampling factor of 4 (bilinear interpolation) after the third encoder block [6]. The representation of the three encoder blocks were concatenated before the final convolution.

4.2 Ensemble methods

The goal of ensemble methods is to combine the predictions of several models trained for the same task by the same learning algorithm in order to improve predictions' robustness over a single model. Ensemble methods are often used to slightly increase the accuracy of an already performing model for tasks such as image classification or object detection [8, 16]. The ensemble algorithm presented in this study used different models trained in parallel and combined their predicted images. It is hypothesized that the use of different models rather than the same model trained with different hyper-parameters will improve the generalization of the ensemble model, and therefore its performance in test. In fact, 4 different models are used in the ensemble and, to obtain the best performance, the 4 best models are chosen based on their validation loss.

Several ways of combining the images predicted by the models have been implemented. A simple method of weighted vote based on the loss in validation of the models was used to get the benchmark for the other combination methods. A more complex method that has been implemented is trained weighted voting, which uses a single layer of 1x1 convolution to optimize the weights used for the combination. Finally, the last implemented method used an inception block consisting of the concatenation of 1x1, 3x3, and 5x5 convolutions, max pooling, and input channels followed by a 1x1 convolution. In both networks, a ReLU was used after the 1x1 convolution.

4.3 Metrics evaluation

The metrics used in this study are the root mean squared error (RMSE) and the peak signal-to-noise ratio (PSNR). The RMSE is the primary metric that determines the model's ranking for the 2021 American Association of Physicists

in Medicine (AAPM) DL-sparse-view CT Grand Challenge [5], and was therefore used in our study. It is calculated by equation 2, where N is the size of the dataset evaluated by the metric.

$$MSE(\hat{y}, y) = \frac{\sum_{i=0}^{511} \sum_{j=0}^{511} \hat{y}_{i,j} - y_{i,j}}{512^2} . \tag{1}$$

$$RMSE(\hat{Y}, Y) = \frac{1}{N} \sum_{k}^{N} \sqrt{MSE(\hat{Y}_k, Y_k)} . \tag{2}$$

We also use the PSNR as this metric is often used in the literature to compare noisy sparse reconstruction CT images with ground truth images [7]. It is obtained by equation 3, where PR is the pixel range, meaning the distance between the maximum and minimum possible values of pixels in the images. It is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

$$PSNR(\hat{Y}, Y) = \frac{1}{N} \sum_{k}^{N} \left(20 \log_{10}(PR) - 10 \log_{10}(MSE(\hat{Y}_k, Y_k) \right) . \tag{3}$$

4.4 Experimental setup

To train and measure the networks' performance, we used the dataset of CT images distributed by the organizers of the AAPM's 2021 DL-sparse-view CT challenge. This dataset contains 4000 pairs of FBP-128 and BP images. To train the networks, the data were split in 3240 training images, 360 validation images and 400 tests images. The Adam optimizer was used with a learning rate of 0.0001 and a weight decay of 0.0001. Since some networks had large memory consumption, and to allow a fair comparison between all networks, a batch size of 1 was used. The learning rate was decreased by a factor of 3.33 when a plateau was observed for 3 consecutive epochs. The training of each network was done over 20 epochs. The networks' parameters were saved during the training phase each time the validation loss was improved. During the testing phase, the weights corresponding to the best validation loss were used. The loss function used for the training was the RMSE. All networks were implemented using PyTorch v1.8, and the training was executed on a Windows 10 workstation using an Intel(R) Core(TM) i9-10920X CPU (128 Go and 24 cores @ 3.5 GHz) and a NVIDIA GeForce RTX 2080-Ti NVIDIA GeForc GPU (11 GB of memory).

For the ensemble method, the training of the networks was done over 100 epochs with a batch size of 1, using the Adam optimizer with a weight decay of 0.0001 and an initial learning rate of 0.01 which was reduced with a factor 10 when a plateau was observed for 3 consecutive epochs. The training was executed using the predicted images of the U-Net Original, the Breast U-Netv2, the Dense U-Net, and the inception U-Net. Hence, 4 channels were used at the input of the ensemble networks. The number of parameters to learn was 5 for the 1x1 convolution, and 192 for the inception block.

4.5 Data augmentation

In order to increase the amount of data available for training, the images were augmented by conducting random rotations. All the images and their targets were only rotated once and, as a result, the total number of available images was doubled to 8000 image pairs of FBP-128 and BP. However, it is important to note that only one model was trained with the augmented data in order to measure the influence on the performance from using the augmentations. Models generally do not use augmentation and the abbreviation AUG in a model's name means that it was trained with the augmented images in addition to the initial images of the AAPM's dataset.

5 Results

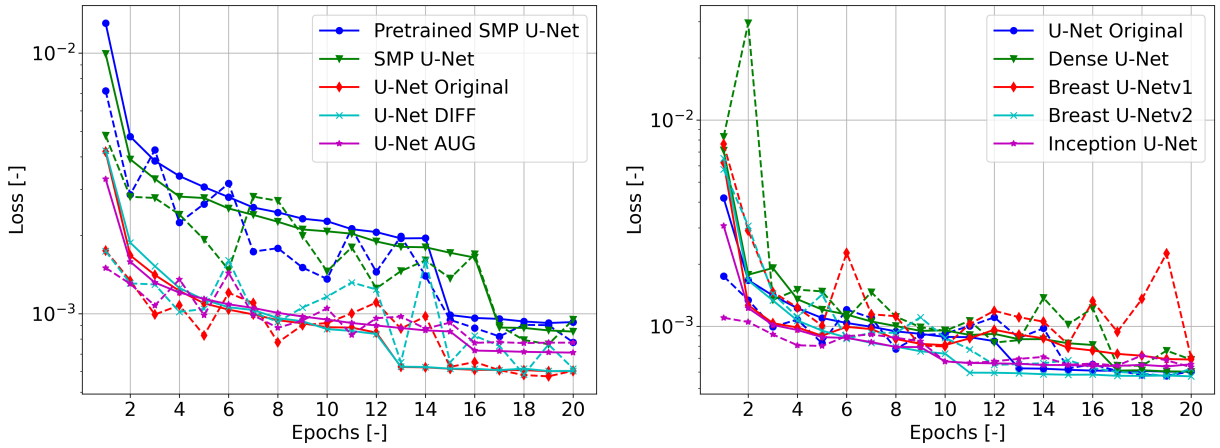
The different results related to the training phase are presented in Table 1. The best performances achieved for each parameter are shown in bold, and it can be noticed that the Breast U-Netv1 is the network which contains the fewest parameters to train, with only 1 122 127 parameters. When it comes to average training time (per epoch), the U-Net Original network is the least time-consuming, averaging 180 seconds per epoch. Finally, the network with the smallest training and validation RMSE loss is Breast U-Netv2, with 5.71×10^{-4} for both losses. Regarding the RED-CNN architecture, preliminary results were unimpressive (validation loss between 8×10^{-4} and 9×10^{-4}), so the model was not presented here nor further developed. The learning curves for all models are shown in Figure 6. As we can see,

[5] https://www.aapm.org/GrandChallenge/DL-sparse-view-CT/

Table 1: Training results. The train loss corresponds to the loss at the 20$^{\text{th}}$ epoch, and the validation loss corresponds to the best validation loss reached during training.

Model	Number of parameters	Average epoch training time [s]	Train loss (RMSE)	Validation loss (RMSE)
Pretrained SMP U-Net	29 942 753	222	9.28×10^{-4}	7.76×10^{-4}
SMP U-Net	29 936 481	255	8.53×10^{-4}	7.63×10^{-4}
U-Net Original	4 320 033	**180**	6.01×10^{-4}	5.75×10^{-4}
U-Net AUG	4 320 033	360	7.08×10^{-4}	7.71×10^{-4}
U-Net DIFF	4 320 033	181	6.03×10^{-4}	5.84×10^{-4}
Dense U-Net	9 162 753	564	7.08×10^{-4}	6.42×10^{-4}
Breast U-Netv1	**1 122 127**	220	6.89×10^{-4}	7.01×10^{-4}
Breast U-Netv2	1 343 951	387	$\mathbf{5.71 \times 10^{-4}}$	$\mathbf{5.71 \times 10^{-4}}$
Inception U-Net	18 516 131	980	6.53×10^{-4}	6.32×10^{-4}

all the networks managed to learn and reduce their initial loss. We also notice in Figure 6a that the networks trained with SMP U-Net obtained higher loss in training and in validation than the networks trained with the U-Net Original. In figure 6b, the cluster of points rather condensed towards the last epoch shows that the networks achieved similar performance after 20 epochs.



(a) (b)

Figure 6: Illustration of (a) the learning curves for U-Net model trained with different images or weight parameters and (b) the learning curves for different models. The solid and dotted lines correspond to the training and validation loss, respectively.

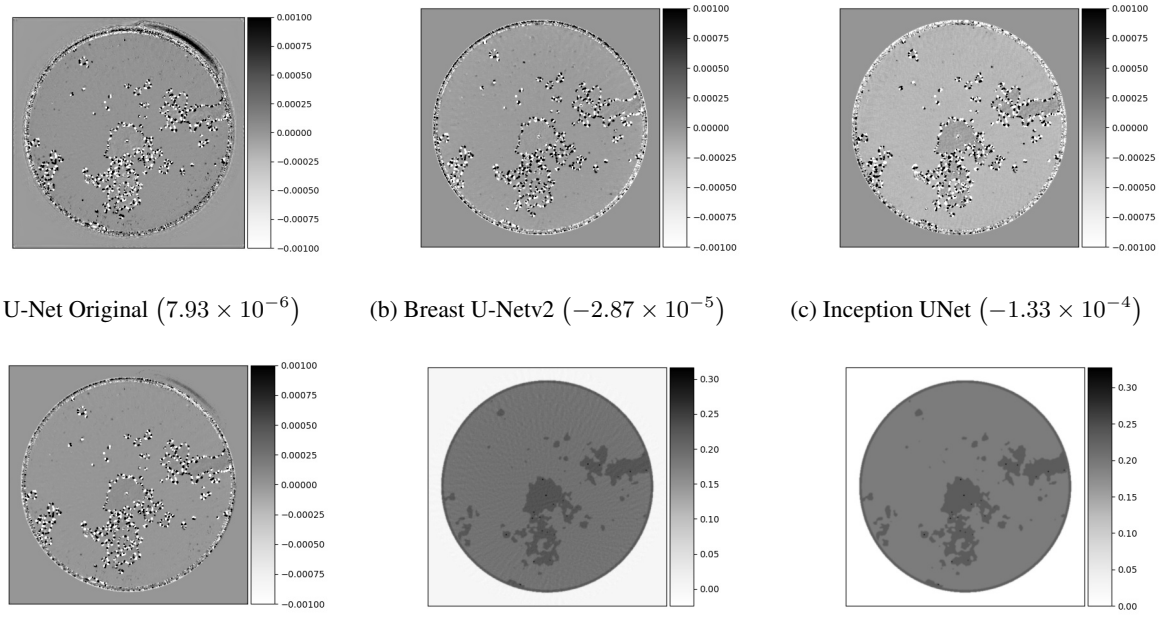
The results on the test set obtained with the different networks are included in Table 2. All the networks achieved an RMSE lower than 8×10^{-4} and, in the single model testing phase, Breast U-Netv2 achieved the lowest RMSE value of 5.70×10^{-4}. The U-Net Original was second, and the Inception U-Net was third. The Dense U-Net achieved the lowest standard deviation on the RMSE with 1.01×10^{-4}. The highest PSNR observed was $70.851\,\text{dB}$ with U-Net DIFF. The Pretrained SMP U-Net had the lowest standard deviation on the PSNR with $1.364\,\text{dB}$. The 4 models used in ensemble methods are U-Net Original, Dense U-Net, Breast U-Netv2 and Inception U-Net. The weights attributed to each of these single models for the vote in Weighted Vote and CNN Vote are presented in the Appendix in Table 3. The results show that the trained weighted average using a CNN achieved the lowest RMSE value among all models and ensemble methods with 5.32×10^{-4}. The highest PSNR observed among ensemble methods was $65.662\,\text{dB}$ with CNN vote.

The differences between prediction obtained with the best models and ground truth for a random test case are presented in Figure 7. The U-Net Original predicted image shown in Figure 7a contains an observable artifact at the top right of the delimiting circle. Multiple predicted images were observed and the artifact seems to be systematically present in the predictions made by this network. Also, we notice that the errors made by the different models are not exactly the same. For example, the Inception U-Net predicted image shown in Figure 7c seems to give pixel values that are often lower than expected ($\mu = -1.33 \times 10^{-4}$) while the U-Net Original tends to overshoot the pixel values

Table 2: Quantitative evaluation of test results for all metrics. Both the metrics' values and standard deviations (σ) are presented.

	Average image prediction time [ms]	RMSE	σ_{RMSE}	PSNR [dB]	σ_{PSNR}
Single model					
Pretrained SMP U-Net	37.2	7.76×10^{-4}	1.20×10^{-4}	62.304	**1.364**
SMP U-Net	26.1	7.62×10^{-4}	1.21×10^{-4}	62.475	1.413
U-Net Original	**14.7**	5.73×10^{-4}	1.02×10^{-4}	64.974	1.598
U-Net AUG	**14.7**	6.16×10^{-4}	1.08×10^{-4}	64.346	1.562
U-Net DIFF	**14.7**	5.83×10^{-4}	1.04×10^{-4}	**70.851**	1.599
Dense U-Net	42.8	6.41×10^{-4}	$\mathbf{1.01 \times 10^{-4}}$	63.976	1.387
Breast U-Netv1	17.0	7.00×10^{-4}	1.21×10^{-4}	63.229	1.545
Breast U-Netv2	26.9	$\mathbf{5.70 \times 10^{-4}}$	1.05×10^{-4}	65.030	1.662
Inception U-Net	46.3	6.31×10^{-4}	1.09×10^{-4}	64.138	1.550
Ensemble method					
Weighted Vote	8.82*	5.41×10^{-4}	$\mathbf{1.04 \times 10^{-4}}$	65.496	**1.729**
CNN Vote	**8.78***	$\mathbf{5.32 \times 10^{-4}}$	$\mathbf{1.04 \times 10^{-4}}$	**65.662**	1.766
Inception Block	29.56*	5.36×10^{-4}	$\mathbf{1.04 \times 10^{-4}}$	65.585	1.746

*The time needed for the evaluation of all single models used in the ensemble is not taken into account here.



(a) U-Net Original $\left(7.93 \times 10^{-6}\right)$ (b) Breast U-Netv2 $\left(-2.87 \times 10^{-5}\right)$ (c) Inception UNet $\left(-1.33 \times 10^{-4}\right)$

(d) CNN Vote $\left(-2.67 \times 10^{-6}\right)$ (e) FBP Image (f) Breast phantom (GT)

Figure 7: Differences between prediction obtained with the best models and ground truth for a random test case. The mean of the prediction difference images are also presented in parenthesis for each image. The differences image between the FBP and the breast phantom is presented in the Appendix in Figure 9.

in general ($\mu = 7.93 \times 10^{-6}$). Finally, the U-Net Original network has also been trained for 50 epochs to compare its performance with other models trained with only 20 epochs. The results are presented in the Appendix section in Table 4 and the learning curve is shown in Figure 8. This network achieved an RMSE loss of 5.39×10^{-4} and a PSNR of 65.522 dB, which is better than every single model presented previously, excluding ensemble methods.

6 Discussion

Even if the U-Net Original is a rather simple network compared to most of the other implemented networks, it achieved the second best RMSE accuracy with 5.73×10^{-4} on the test set. This network is also the least time-consuming with an average of 180 s per epoch in training and only 14.7 ms average prediction time per image. It was hypothesized that using differences between the FBP-128 images and the breast phantom BP images as targets might ease the learning process for the U-Net Original. The results tend to show that there is no improvement regarding the RMSE as the U-Net DIFF network obtained an RMSE 1% greater than the U-Net Original's one. However, the U-Net DIFF obtained the highest PSNR (70.851 dB) among all models, 9% higher than the PSNR of the U-Net Original. Therefore, we suppose that learning noisy images might help to reduce the noise in the predicted images, but further investigations are needed.

Pre-trained networks on ImageNet were also investigated, but the results were not conclusive as the Pretrained SMP U-Net and the SMP U-Net obtained an accuracy of RMSE on the test set inferior by 35% and 33% respectively compared to the U-Net Original accuracy. Those poor results might be explained by the fact that these pre-trained networks were trained on RGB images (3 channels) of real life objects or animals, while our images are limited to a single channel and are far from resembling to everyday objects. Furthermore, the SMP pre-trained networks have 27 times more parameters than the U-Net Original network which makes them difficult to train. In fact, it was hard to find pre-trained networks on CT images, and the only network we found was RED-CNN, whose preliminary results were unimpressive.

When looking into the variants of the U-Net Original architecture, only the Breast U-Netv2 had a better RMSE accuracy (slight improvement of 0.5%) with four times less parameters. However, the training time is twice of the training time with U-Net Original, because of the large number of copy in the Breast U-Net architecture. When testing the hypothesis of the effect of the convolution blocks in the decoder of the Breast U-Net architecture, it was found that the Breast U-Netv2 performed better than the Breast U-Netv1 (increase of 22.8% in RMSE accuracy). This result suggests that adding convolution blocks in the decoder helped the training, which is the opposite of what the authors of the Breast U-Netv1 claimed [6]. However, the authors did not give details on the implementation of their upsampling method, and thus our method could differ from theirs. Nevertheless, we argue that adding convolution blocks with skip connections in the decoder is helpful with our dataset. Since the Dense U-Net and the Inception U-Net also performed worse than the Breast U-Netv2 (decrease of 11.1% and 9.5% in the RMSE accuracy), this suggests that increasing the depth of the encoder (5 layers with the Dense/Inception/Original U-Net vs 3 layers with the Breast U-Netv2) might not help the training phase, as argued by Ghazi et al. [6]. Considering the complex shape of fibroglandular tissue, small images at depth 4 and 5 in the encoder (64×64 or 32×32 pixels, respectively) might not be able to preserve useful representation of the tissue. Hence, the Breast U-Net might have an edge over the other U-Net methods by learning the useful features earlier in the training phase.

All implemented ensemble methods have achieved better performance than the best single model. In fact, the RMSE accuracy on the test set was improved by 7% with the best ensemble method, which is the CNN voting method, compared to the accuracy of Breast U-Netv2. These results prove that combining multiple models' predictions helps performance in generalization. The decrease in lost can be explained by the fact that the errors made by the single models used in the ensemble are different from each other as we can see in Figure 7. The mean values μ of the predicted differences images of each single model are slightly different while some are positive and some are negative. The ensemble method can therefore average these estimates and obtain better results on new images. For example, the artifact observed in the top right of the predicted images with U-Net Original (Figure 7a) has diminished in importance in the image predicted with CNN vote prediction (Figure 7d). Moreover, since the voting method uses only four weights and one bias, it can easily be interpreted. Table 3 in the Appendix presents the weights and biases of the two simplest ensemble methods. The weights of the Weighted Voting method act as a reference for the other methods' weights since they were simply determined by the loss in validation of each single model. In the CNN Vote method, the weight for Breast U-Netv2 has increased by 62% while the weight for Dense U-Net has decreased by 82%. The ensemble method seems to have learned to ignore the predicted images of Dense U-Net.

Limitations in this study are largely centered on the small data set used to train the networks. While five different architectures were investigated, the achieved RMSE is of the same order of magnitude (10^{-4}) for all networks. This suggests that the number of training examples is an issue for the sparse-view reconstruction. To address this issue, 4000 additional image pairs were generated from random rotations of the existing pairs. However, the results did not seem conclusive, since the accuracy of RMSE obtained on the test set with U-Net AUG is inferior by 7.5% than with U-Net Original, but with twice the training time and number of examples. An alternative to rotations as a data augmentation method would be to generate similar BP (see Figure 1) with in-house algorithms. With the help of colleagues, new BP were generated, and this approach is still under investigation for the AAPM Grand Challenge.

Another limitation is that the networks were trained for 20 epochs only. Therefore, it is unknown which network would be the best under hundreds of epochs. For instance, the RMSE obtained with U-Net Original trained for 50 epochs

compared to U-Net Original trained for 20 epochs on the test set was improved by 7%. However, this improvement was achieved by increasing the total training time by 250%. Hence, depending on the available resources at hand (such as access to GPUs), it might not be necessary to train as much as hundreds of epochs to achieve a satisfactory level of RMSE accuracy. Furthermore, the method used to decrease the learning rate might not be optimal for 20 epochs. As an alternative, it could be interesting to gradually reduce the learning rate from 10^-4 to 10^{-5} or 10^{-6}. However, with a learning rate below 10^{-6}, preliminary results showed that no further improvements could be obtained in the validation loss. Using a batch size of 1 might not be optimal as well, since it is more time consuming to compute the gradient for each example, and the batch normalisation layers might have worsening effects for a small batch size [17]. Two ways to overcome this issue would be to use patch CT images instead of using full CT images [3] or training with group normalization layers instead of batch normalization layers [17].

Since the geometry of the CT beam used to generate the sinograms for the FBP reconstruction of the FB phantom was unknown, we could not reconstruct the sparse-view CT images using the state-of-the-art TV method [4]. Therefore, we could not compare our results with non CNN-based methods. However, in Sidky et al. [15], the TV method could achieve a RMSE accuracy ranging from 10^{-6} to 10^{-8}. Hence, they concluded that CNN methods (accuracy of the order of 10^{-4}) could not compete with the TV method [15]. However, their comparison was based on 10 test images only. While we did not approach an accuracy of 10^{-6}, our results confirm their observations on 400 test images, thus making our results more robust. Furthermore, we achieved a similar level of RMSE accuracy (10^{-4}) with less than three hours of training using the Breast U-Net architecture, compared to the 450 epochs (few days) of training in Sidky et al. [15] (AAPM challenge organizers). Also, by looking at the AAPM Grand Challenge ranking (on May 3rd 2021), the best RMSE achieved by the competitors is 2.34×10^{-4} (based on 10 validation images). However, the methods used by the best team are still unknown, so does not allow further comparisons.

In other studies, CNNs tend to compete with the TV method [10, 7, 3, 21]. Hence, this suggests that the achieved accuracy might be dependent on the data set for both CNNs and TV approaches. Nevertheless, the achieved accuracy in both RMSE and PSNR with all the networks are comparable [21] or superior [3] to previous models published in the literature. It could be interesting to train our models on different datasets, such as real chest CT images or head and neck CT images rather than using numerically-generated phantoms to further compare with the state-of-the-art TV method.

7 Conclusion

Different deep CNN architectures were trained and tested on sparse-view CT images of 2D breast phantoms. All the networks had a U-Net style architecture. The architecture that achieved the best RMSE accuracy after 20 epochs was the Breast U-Netv2 with 5.70×10^{-4}. We also tested ensemble methods combining the best performing architectures, which improved the single models' performance, the best being CNN Vote with 5.32×10^{-4} RMSE accuracy. Considering the accuracy and the low prediction time (lower than 100 ms) of the studied models, we conclude that U-Net type deep convolutional networks are appropriate for sparse CT reconstruction.

Acknowledgments and Disclosure of Funding

The authors would like to acknowledge the support of Leonardo Di Schiavi Trotta for his help with data augmentation and Daniel Gourdeau for fruitful discussions.

References

[1] Susovan Banerjee, Tejinder Kataria, Deepak Gupta, Shikha Goyal, Shyam Singh Bisht, Trinanjan Basu, and Ashu Abhishek. Use of ultrasound in image-guided high-dose-rate brachytherapy: enumerations and arguments. *Journal of Contemporary Brachytherapy*, 9(2):146–150, April 2017. ISSN 1689-832X. doi: 10.5114/jcb.2017.67456. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5437087/.

[2] Jerrold T Bushberg and Inc Ovid Technologies. *The essential physics of medical imaging*. Wolters Kluwer Health/Lippincott Williams & Wilkins, Philadelphia, 3rd ed edition, 2011. ISBN 9780781780575.

[3] Hu Chen, Yi Zhang, Mannudeep K. Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE Transactions on Medical Imaging*, 36(12): 2524–2535, 2017. doi: 10.1109/TMI.2017.2715284.

[4] Philippe Després and Xun Jia. A review of GPU-based medical image reconstruction. *Physica Medica*, 42:76–92, October 2017. ISSN 1120-1797. doi: 10.1016/j.ejmp.2017.07.024. URL http://www.sciencedirect.com/science/article/pii/S1120179717302417.

[5] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. doi: 10.1109/TIT.2006.871582.

[6] Peymon Ghazi, Andrew M. Hernandez, Craig Abbey, Kai Yang, and John M. Boone. Shading artifact correction in breast ct using an interleaved deep learning segmentation and maximum-likelihood polynomial fitting approach. *Medical Physics*, 46(8):3414–3430, 2019. doi: https://doi.org/10.1002/mp.13599. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mp.13599.

[7] Yoseob Han and Jong Chul Ye. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE Transactions on Medical Imaging*, 37(6):1418–1429, 2018. doi: 10.1109/TMI.2018.2823768.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[9] William R. Hendee and F. Marc Edwards. ALARA and an integrated approach to radiation protection. *Seminars in Nuclear Medicine*, 16(2):142–150, 1986. ISSN 0001-2998. doi: https://doi.org/10.1016/S0001-2998(86)80027-7. URL https://www.sciencedirect.com/science/article/pii/S0001299886800277.

[10] Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017. doi: 10.1109/TIP.2017.2713099.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

[12] Paul A. Oakley and Deed E. Harrison. Death of the alara radiation protection principle as used in the medical sector. *Dose-Response*, 18(2):1559325820921641, 2020. doi: 10.1177/1559325820921641. URL https://doi.org/10.1177/1559325820921641. PMID: 32425724.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[14] Seeram E. Computed Tomography: A Technical Review. *Radiologic technology*, 89(3):279CT–302CT, 2018. ISSN 0033-8397. 279.

[15] Emil Sidky, Iris Lorente, Jovan G. Brankov, and Xiaochuan Pan. Do cnns solve the ct inverse problem. *IEEE Transactions on Biomedical Engineering*, pages 1–1, 2020. doi: 10.1109/TBME.2020.3020741.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL http://arxiv.org/abs/1409.4842.

[17] Yuxin Wu and Kaiming He. Group Normalization. *International Journal of Computer Vision*, 128(3):742–755, March 2020. ISSN 1573-1405. doi: 10.1007/s11263-019-01198-w. URL https://doi.org/10.1007/s11263-019-01198-w.

[18] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. 05 2015.

[19] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. doi: 10.1109/TIP.2017.2662206.

[20] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018. doi: 10.1109/LGRS.2018.2802944.

[21] Zhicheng Zhang, Xiaokun Liang, Xu Dong, Yaoqin Xie, and Guohua Cao. A sparse-view ct reconstruction method based on combination of densenet and deconvolution. *IEEE Transactions on Medical Imaging*, 37(6):1407–1417, 2018. doi: 10.1109/TMI.2018.2823338.

8 Appendix

Table 3: Weights used for the vote in ensemble methods.

Models	Weighted Vote	CNN Vote
U-Net Original	0.2387	0.2983
Dense U-Net	0.2344	0.0431
Breast U-Netv2	0.2628	0.4268
Inception U-Net	0.2641	0.2329
Bias	0.0	-0.00002
Weights summation	1.0	1.0011

Table 4: Quantitative evaluation of the U-Net Original model performances trained with 50 epochs.

Model	Train loss (RMSE)	Validation loss (RMSE)	Test loss (RMSE)	$\sigma_{\text{test loss}}$ (RMSE)	PSNR [dB]	$\sigma_{\text{test loss}}$ (PSNR)
U-Net Original	5.36×10^{-4}	5.40×10^{-4}	5.39×10^{-4}	1.01×10^{-4}	65.522	1.691

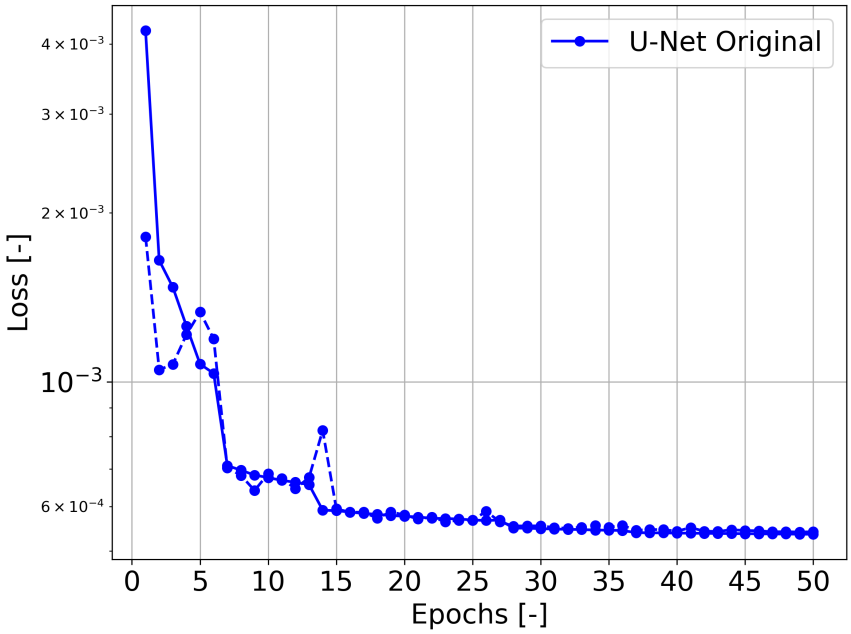


Figure 8: Learning curve for the U-Net Original model trained with 50 epochs.

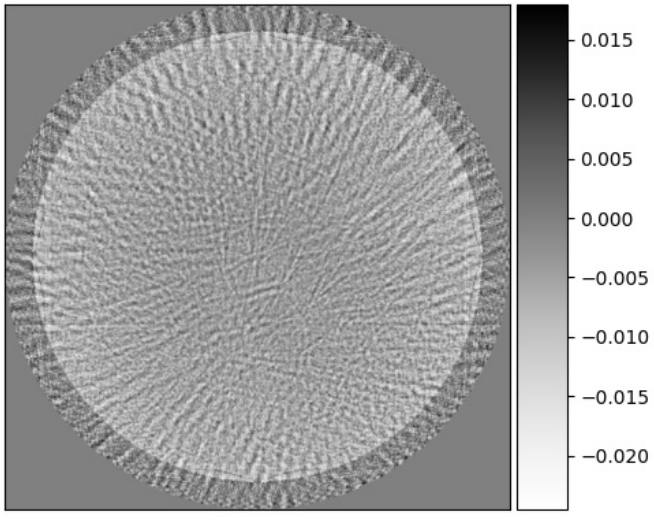


Figure 9: DIFF image for the random test case.