Evaluating U-Net for Low-View CT Reconstruction

Erik Clemens
May 12, 2022

***Abstract*—In this paper, I re-implement the FBPConvNet presented in [1] to further study the impact of training data on model performance. While FBPConvNet offers some computational advantages over iterative reconstruction techniques, I find that the complexity of the training and test data have a significant impact on model performance. Size of the training dataset is also found to influence model performance.**

I. Introduction

As we discussed in this course, the measurements obtained from Computed Tomography (CT) are described by the Radon transform of the volume being imaged. If Radon transform data is available for all possible lines through the volume, then the attenuation coefficients of the volume can be reconstructed exactly using filtered back-projection (FBP) [2]. It is both impractical and undesirable to collect an infinite number of measurements, as increasing the number of measurement angles exposes the subject to higher levels of ionizing radiation. When the number of measurement angles (typically called views in the literature) is low, the quality of the FBP degrades. Thus, much ongoing research seeks to find enhancements or alternatives to the FBP that provide high quality reconstructions with a low number of views.

For my project, I wrote a PyTorch implementation of the low-view CT reconstruction method published by Jin *et al.* [1]. This particular approach was fascinating to me, as it utilized techniques discussed in this course, as well as allowing me to leverage my previous research experience in deep learning. I also thought this valuable to prepare me for a potential career in the medical imaging industry, where deep learning reconstruction techniques have improved image quality, lowered radiation doses, and achieved FDA approval [3].

A. Existing Methods

Several existing methods attempt to invert the Radon transform. The simplest is the FBP, but, as mentioned above, it suffers from low image quality in the case of low-views.

As we discussed in class, it is also possible to reconstruct the attenuation image by modeling the discrete Radon transform as a system of linear equations. $A\vec{c} = \vec{b}$, where $\vec{c}$ is the vectorized attenuation image and $\vec{b}$ is the vectorized sinogram measurements. In this case, the matrix A is $m \times n$, where m is the total number of pixels in the sinogram and n is the total number of pixels in the reconstructed image. To have a medically valuable image, n must be large. When the number of views is limited for patient safety, however, m is low, leading to a heavily underdetermined system with either no solution or an infinite number of solutions [2]. Thus, even effective methods for solving this system of linear equations, such as conjugate gradients least-squares algorithms, struggle to reproduce a quality image.

Iterative approaches have also been developed for low-view CT reconstruction. For example, the authors of [1] refer to the ISTA algorithm presented in [4]. This algorithm iteratively filters the estimated image, adds a bias term, and passes the resulting points through a non-linear function. Within these iterative frameworks, novel regularization methods such as [5] are often developed to improve reconstruction quality. While such methods have produced quality reconstructions, they require hand-tuning and require time-consuming computation.

II. Methods

In their paper, Jin *et al.* observe that the structure of state-of-the-art iterative reconstruction algorithms (filter, adding a bias, and then applying a pointwise non-linearity) is similar to that seen in Convolutional Neural Networks. Thus, they seek to apply an existing neural network to the problem of image reconstruction. To do this, they modify the U-Net architecture originally presented in [6]. This modified architecture, which they call FBPConvNet, is detailed in Figure 1.

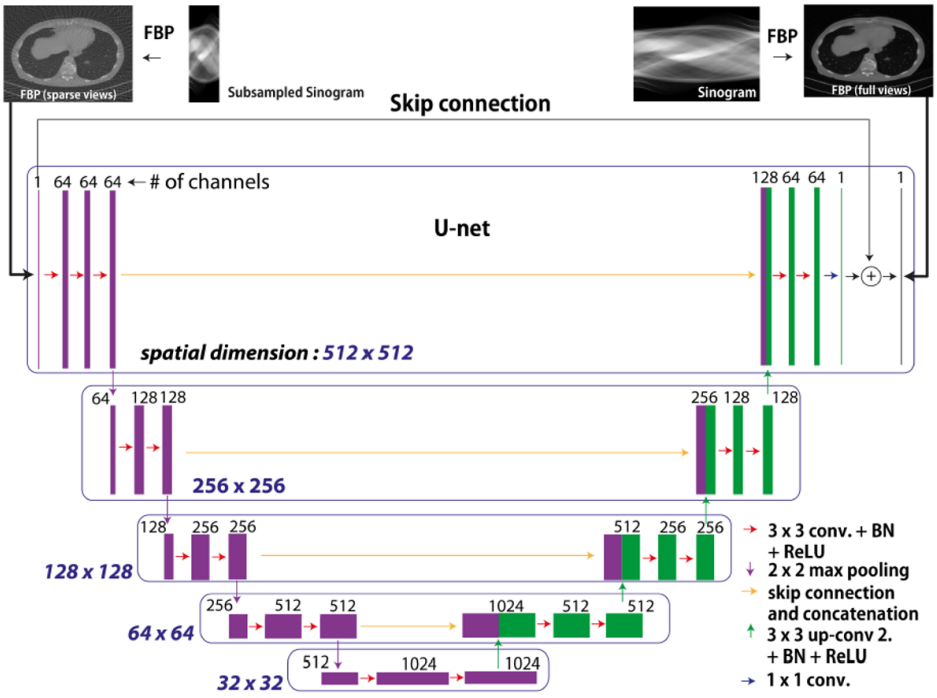


Fig. 1: Modified U-Net architecture presented in [1].

As seen in Figure 1, the FBPConvNet takes in the low-view FBP image as input and seeks to emulate the high-view FBP ground truth. This noise reduction to match ground truth is performed through the network layers. At each stage of the network, 3×3 convolutional filters are applied to the network, producing an equivalent number of channels. Thus, the first convolutional layer takes in the single channel grayscale image and applies 64 different convolutional filters to it. The output from each filter is passed to the next layer, which also applies convolutional filters matching the number of channels, and so on.

The convolutional layers are paired with max pooling layers, which form the downward connections on the left side of Figure 1. These 2×2 max pooling layers subsample the image by looking at a 2×2 grid and keeping only the pixel with the maximum intensity. By combining convolutional filters with max pooling, the network is able to learn progressively more complex features within the input. At the lowest layer of Figure 1, the input image has been reduced to a 32×32-pixel, highly compact, abstract representation of the input image.

The left side of the architecture in Figure 1 performs upsampling to convert the highly compact representation back into a 512×512-pixel image. At each stage, the yellow lines represent a skip connection that carries some information from earlier in the network. These skip connections help ensure that no information is lost during the downsampling/abstraction process. The final layer in this modified U-Net combines the 64-channel image and combines the channels to produce a final grayscale image. Lastly, the U-Net is modified to have a skip connection between the input and the final output. Thus, the network is only tasked with learning the difference between the noisy FBP image and the quality FBP image, greatly simplifying the process. It would be possible to have a network reconstruct directly from a low-view sinogram, but the skip connection and use of the FBP to generate a low-view reconstruction incorporate the known physics of the situation and simplify model training.

The weights in all the convolutional filters represented in Figure 1 are tuned through a process known as training, which allows the network to ”learn” about the data. Model training is a non-convex optimization problem that seeks model weights such that some loss criterion is optimized. A stochastic version of gradient descent, similar to the gradient descent methods we discussed in class, is used to find an optimal solution. Including the batch normalization layers, the FBPConvNet has $31.1M$

parameters to tune.

In their paper, Jin *et al.* use a simulated dataset consisting of random images of ellipses to train and evaluate the FBPConvNet [1]. They take the radon transform of these images using 1000, 143, and 50 views and then reconstruct images from the sinogram. The 1000-view images represent the ground truth, that is, what the reconstruction would look like if it were safe to subject the patient to this level of radiation. The 143- and 50-view reconstructions represent the noisy, low-view data.

A. My Contribution

One major disadvantage of applying convolutional neural networks to a problem is that the quality of the solution is heavily dependent on the data used for model training. In their paper, Jin *et al.* note that if a model is trained on high-view data, it performs poorly when reconstructing from low-view sinograms [1]. I wanted to further explore the effects of training data on testing image quality, so I explore two dimensions: dataset size and data complexity.

In their paper, Jin *et al.* use a dataset consisting of 500 images to train their model [1]. 475 are used for training, while 25 images are used to test the final result. Since a larger dataset often allows the model to capture a more realistic input distribution, I performed experiments with a 500-sample dataset and compared them with results from a 1000-sample dataset. For each dataset size, I use 90% of the data for training, 5% for validation, and 5% for testing.

In addition to the size of the dataset, the complexity of the data can have an effect on model performance. While the original authors do not provide their dataset nor explain the complexity of their data, having a different number of objects in the training images can strongly impact model performance. To explore this, I generated data with three levels of complexity: images with a number of ellipses randomly uniformly sampled between 5 and 14, images with a number of ellipses randomly uniformly sampled between 15 and 24, and images with a number of ellipses randomly uniformly sampled between 25 and 34.

I generated my images using Python's `numpy` and `scikit-image` libraries to handle ellipse generation and rotation, and used the `radon()` and `iradon()` methods of `scikit-image` to perform the reconstruction with different view levels. My model was implemented in PyTorch and trained on a single NVIDIA RTX 3090 GPU. Training took 45 minutes for the 500-sample datasets and 90 minutes for the 1000-sample datasets. I followed the hyper-parameter settings used in [1], although I acknowledge that hyper-parameter optimization could lead to improved results. Since I could not find the loss function used by the original authors, I used the $L1$ loss for training, as recommended in [7]. For quantitative results on my testing data, I used the Peak Signal-to-Noise Ratio, as defined in [7]. Code and TensorBoard log files for my dataset generation, model training, and evaluation are available at https://github.com/Erik-C-55/FBPConvNet_Recreation. Updates for readability and usability of this repository are ongoing.

III. Main Results

For my experiments, I trained the model on various combinations of input views (143 or 50), different dataset sizes (500 or 1000 samples) and different ellipse quantities ($5-14$, $15-24$, or $25-34$). This resulted in training 12 different models. As the model trained, I saved the weights from the epochs with the best validation accuracy. The progression of the reconstruction quality from an early epoch to the epoch with the best validation accuracy (in this case, epoch 90) can be seen in Figure 2. The model used in Figure 2 was trained and using the $15-24$ ellipse, 50-view, 1000-sample dataset. The images come from the test portion of this dataset, meaning the model has never seen them before. By saving the weights periodically during training, I was able to generate these test reconstructions afterward. Notice how the model quickly learns to reduce streaking artifacts, but it requires more training to begin to reconstruct the faint, upright ellipse in the center of Figure 2b.

After training the 12 models, I evaluated them on data from the other datasets. Since the images from the 500-sample datasets were included in the corresponding 1000-sample datasets, I do not test models trained on 500-sample datasets on test sets from the 1000-sample datasets, or vice-versa. I do however, test across all the other combinations of image complexity and view numbers. This results in 6 test evaluations for each of the 12 models, as seen in Tables I and II.

There are several key findings in Tables I and II. First, in every case, the modified U-Net outperforms

Train & Val \ *Test*	*5+ ellipse 50 views*	*15+ ellipse 50 views*	*25+ ellipse 50 views*	*5+ ellipse 143 views*	*15+ ellipse 143 views*	*25+ ellipse 143 views*
5-14 ellipses, 50 views	22.8/39.2	20.3/33.5	19.0/30.2	32.3/42.0	29.9/37.6	28.5/35.2
15-24 ellipses, 50 views	22.8/40.6	20.3/35.9	19.0/32.7	32.3/41.8	29.9/38.0	28.5/35.5
25-34 ellipses, 50 views	22.8/43.4	20.3/39.5	19.0/36.6	32.3/43.4	29.9/41.1	28.5/39.0
5-14 ellipses, 143 views	22.8/28.1	20.3/23.8	19.0/21.8	32.3/47.8	29.9/43.4	28.5/40.4
15-24 ellipses, 143 views	22.8/26.3	20.3/23.1	19.0/21.2	32.3/49.7	29.9/46.2	28.5/43.9
25-34 ellipses, 143 views	22.8/26.6	20.3/23.6	19.0/21.9	32.3/50.0	29.9/46.8	28.5/44.8

TABLE I: Mean Per-Image Peak Signal-to-Noise Ratio (PSNR) for 500-sample data. Written "FBP PSNR/UNet PSNR".

Train & Val \ *Test*	*5+ ellipse 50 views*	*15+ ellipse 50 views*	*25+ ellipse 50 views*	*5+ ellipse 143 views*	*15+ ellipse 143 views*	*25+ ellipse 143 views*
5-14 ellipses, 50 views	21.4/39.1	20.1/34.3	18.7/30.8	30.9/41.6	29.6/38.2	28.3/35.7
15-24 ellipses, 50 views	21.4/41.6	20.1/37.7	18.7/34.3	30.9/42.7	29.6/39.8	28.3/37.3
25-34 ellipses, 50 views	21.4/43.2	20.1/40.3	18.7/37.4	30.9/43.9	29.6/42.0	28.3/40.0
5-14 ellipses, 143 views	21.4/26.3	20.1/23.3	18.7/21.2	30.9/47.3	29.6/43.7	28.3/40.7
15-24 ellipses, 143 views	21.4/24.7	20.1/22.6	18.7/20.9	30.9/49.2	29.6/46.8	28.3/44.4
25-34 ellipses, 143 views	21.4/24.8	20.1/22.8	18.7/21.1	30.9/49.5	29.6/47.4	28.3/45.3

TABLE II: Mean Per-Image Peak Signal-to-Noise Ratio (PSNR) for 1k-sample data. Written "FBP PSNR/UNet PSNR".

the vanilla FBP. Secondly, as expected, the FBP and modified U-Net have higher PSNR with more views. As found in [1], the models trained on 143-view data perform much worse on the 50-view test data than on the 143-view test data. Interestingly, however, the models trained on the 50-view data tend to perform better on the 143-view data than on the 50-view data they were trained with, showing the strong impact of having more views.

My experiments find that image complexity has a significant impact on reconstruction quality. While the FBP PSNR does not decrease significantly as image complexity increases, the U-Net PSNR does. Even when the model is trained on complex data, it still performs better on the simpler test data. This shows a significant connection between data complexity and model performance. Lastly, dataset size does seem to play a small role in model performance. For the models trained and evaluated on 50-view data (the upper-left quadrant of Tables I and II), there is an average per-cell increase of 0.79 dB when the dataset size is increased from 500 to 1000. For the models trained and evaluated on 143-view data (the lower-right quadrant of Tables I and II), there is an average per-cell increase of 0.14 dB when the dataset size is increased from 500 to 1000.

REFERENCES

[1] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[2] T. G. Feeman, *The Mathematics of Medical Imaging: A Beginner's Guide*, 2nd ed., ser. Springer Undergraduate Texts in Mathematics and Technology. Springer International Publishing Switzerland, 2015.

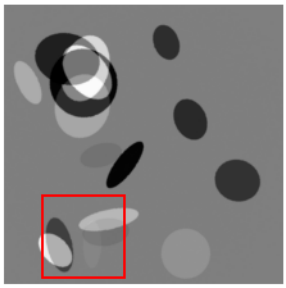
[3] J. Hsieh, E. Liu, B. Nett, J. Tang, J. Baptiste Thibault, and S. Sahney, "A new era of image reconstruction: Truefidelity™," General Electrical Company, Tech. Rep. JB68676XX, July 2019.

[4] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20042

[5] M. T. McCann, M. Nilchian, M. Stampanoni, and M. Unser, "Fast 3d reconstruction method for differential phase contrast x-ray ct," *Opt. Express*, vol. 24, no. 13, pp. 14 564–14 581, Jun 2016. [Online]. Available: http://opg.optica.org/oe/abstract.cfm?URI=oe-24-13-14564

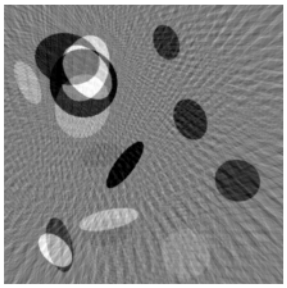
[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[7] J. Zbontar, F. Knoll, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzal, A. Romero, M. G. Rabbat, P. Vincent, J. Pinkerton, D. Wang, N. Yakubova, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, "fastMRI: An open dataset and benchmarks for accelerated MRI," *CoRR*, vol. abs/1811.08839, 2018. [Online]. Available: http://arxiv.org/abs/1811.08839

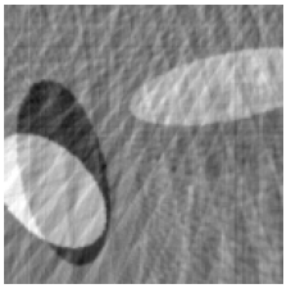


(a) Ground Truth

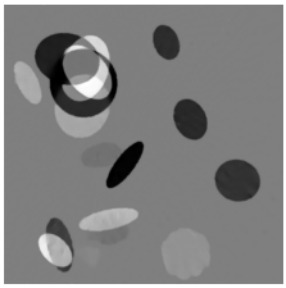
(b) Ground Truth Detail



(c) FBP. PSNR=19.5 dB

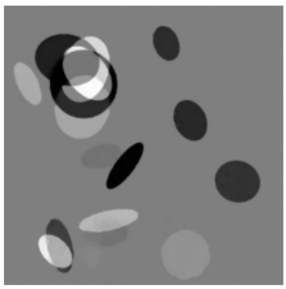


(d) FBP Detail



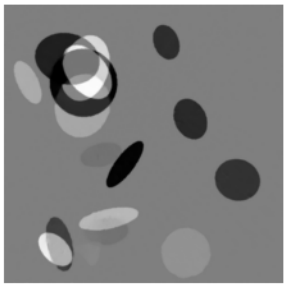
(e) Ep. 38. PSNR=35.2 dB

(f) Epoch 38 Detail



(g) Ep. 62. PSNR=36.5 dB

(h) Epoch 62 Detail



(i) Ep. 90. PSNR=37.7 dB

(j) Epoch 90 Detail

Fig. 2: Reconstruction improvement with training. Detail images show the ROI in sub-fig. (a)