MLSP Project: CNN network to synthesize 7T MRI images from 3T MRI images

Amal Chaoui, Ayoub Youssoufi, Pierre Bourse, Yassin El Hajj Chehade
Univ. Lille - Centrale Lille

Abstract

In this project, we aim to evaluate the performance of the WATNet (Wavelet-based Affine Transformation Network) proposed by Qu et al. [2020], a deep learning network for enhancing the resolution of Magnetic Resonance Imaging (MRI) images. The WATNet leverages both the spatial and frequency content of images using wavelets to learn the low-frequency information while preserving the high-frequency details. Our goal is to reproduce the results of this previous study, in which the WATNet was used to recreate 7T MRI images from corresponding 3T MRI images. Additionally, we will compare the performance of the WATNet to a simple CNN architecture. Finally, we open room to improve the architecture in future works.

1 Introduction

Magnetic Resonance Imaging (MRI) has been widely used in the medical field for the last decades. The resolution of MRI images has increased a lot since its creation, moving from 0.5T for the first MRI in the 1980s, to 7T MRI. However, such precision has a cost. Scanners that are able to produce 7T MRI are very costly and quite few worldwide.

One idea to gain access to higher quality data, while still using available scanners, is to use learning-based methods to recreate 7T MRI images from corresponding 3T MRI images. Previous work has proven its efficiency in this field, using different learning methods, but their performance is highly influenced by the quality of the hand-crafted features.

In their paper, Qu et al. [2020] introduced a deep learning network *WATNet (Wavelet-based Affine Transformation Network)* that leverages both the spatial and frequency content of images using wavelets to learn the low-frequency information (i.e. contrast) while preserving the high-frequency details.

In this project, we aim at reproducing the method and results of the paper and comparing the performance of the WATNet to that of a basic CNN architecture (BasicCNN) composed of Conv2D + BatchNorm + ReLU/sigmoid blocks without the WAT layers.

This report is organised as follows. In section 2, we present the architectures of the 2 networks. Section 3 is dedicated to describing the experimental setting used to produce the results. We compare the performance of the BasicCNN and WATNet both qualitatively and quantitatively in section 4. Finally, we wrap up with a conclusion and improvement ideas in section 5. Source code and data are available at this Github repository.

2 Method

In this section, we present the different neural networks implemented in order to synthesize the 7T MRI images using the 3T MRI images. We first describe the pre-processing steps that allow

synthesising the 3T MRI images from the 7T images (cf. section 2.1). Afterwards, we describe the architectures of 2 neural networks implemented in this project: the BasicCNN and the WATNet (cf. sections 2.2, 2.3).

2.1 Pre-processing

To train and test the neural networks, we want to create a dataset of 3T-7T images. Initially, we have access to 10 7T MRI images associated with 10 different patients. All the 7T MRI images have a shape of $182 \times 218 \times 182$. The goal of the pre-processing step is to generate the synthesized 3T MRI images from the 7T images. For that purpose, we perform the following steps:

1. normalise the 7T images between $[0, 1]$ using the min-max normalisation:

$$I_{\text{norm}} = \frac{I - \min(I)}{\max(I) - \min(I)}.$$

2. artificially generate the 3T images from the 7T images.
3. perform a histogram matching on both the 3T and 7T images using one 7T reference image.

The normalization step is essential for building a consistent dataset for the training process, as it scales the intensity values between 0 and 1. For this, we used the min-max normalization technique, so that the CNN only learns the relationship between the resolution of the images (3T and 7T MRI) and avoids issues related to large intensity values.

The second step is for creating the 3T MRI images that we will use as inputs for the model. In the paper, the authors used real 3T and 7T images. As we only have access to 7T MRI, we artificially create 3T images by adding gaussian noise of variance $\sigma = 0.003$ to the 7T images. We take care of choosing small noise variance in order to limit the change in magnitude of the 7T MRI pixels as they lie in [0,1].

The last step is histogram matching of the two MRI images, 7T and 3T. The idea is to have a common intensity distribution for all MRI images. The main importance of this processing is to align the intensity distribution of both images, thus improving the quality of the images and allowing stable and efficient learning for CNN networks.

2.2 BasicCNN

The architecture of the BasicCNN model consists of 4 stacks of a 2D convolutional layer, followed by batch normalization and activation layer. The first 3 layers have respectively 64, 128, 256 filters and a kernel size of (4x4) with a stride of (1, 1). A padding is added to make the output have the same dimensions as the input. The last convolutional layer has 3 filters and the same kernel size and padding as the previous layers, but with *sigmoid* activation function, used to produce pixel values in $[0, 1]$. The model has relatively a total number of parameters equal to 672 963. Shown in figure 1 is the BasicCNN architecture. Conv2D(F, K, S, P) denotes a 2D convolutional layer with F filters of size K, a stride of S and a padding size P. BatchNorm(m) denotes a Batch Normalisation layer with momentum m.

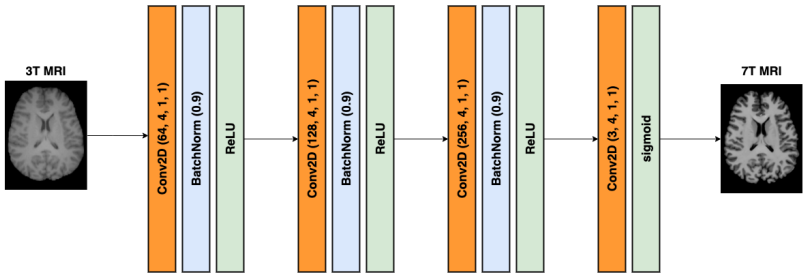
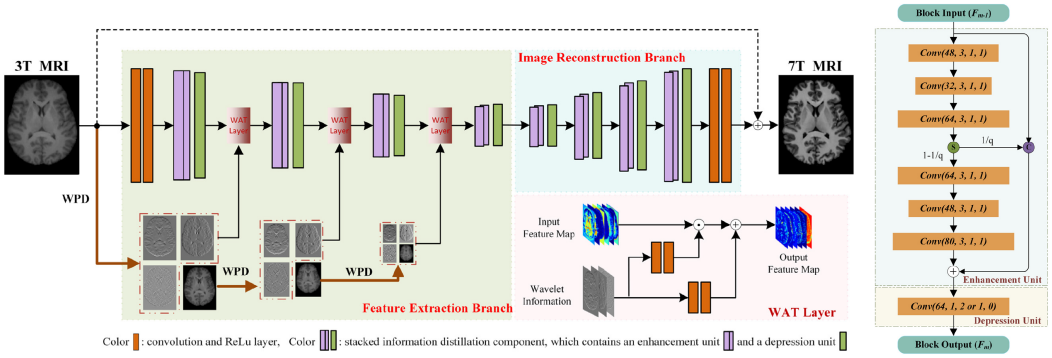


Figure 1: Architecture of the BasicCNN model.

2.3 WATNet

In this section, we provide the implementation details of the WATNet architecture, which is the same as in the original paper. Figure 2a illustrates the global architecture of the WATNet composed of encoding and decoding branches: feature extraction branch and image reconstruction branch.



(a) WATNet (b) DBlock

Figure 2: (a) WATNet architecture combining spatial and wavelet contents to synthesise 7T from 3T MRI images; (b) Architecture of the DBlocks used in the WATNet to suppress useless information in the learned feature maps.

Feature extraction branch. It consists of 2 Conv2D(64, 3, 1, 1) + ReLU blocks that generate feature maps (Conv2D(F, K, S, P) denotes a convolutional layer with F:#filters, K:kernel-size, S:strides, P:padding size), followed by 4 distillation blocks (*DBlocks*) used to extract hierarchical features. As their name suggests, these blocks serve the purpose of suppressing unnecessary information. This operation is performed by the depression unit. But prior to this step, the enhancement unit helps to extract more informative features before distillation. The architecture of the DBlocks is shown in figure 2b. The enhancement unit has a common architecture across DBlocks. The architecture of the depression unit for the first three DBlocks is Conv2D(64, 1, 2, 0) + ReLu and Conv2D(64, 1, 1, 0) + ReLu for the last DBlock.

Finally, inserted in between the DBlocks are 3 Wavelet-based Affine Transformation (WAT) layers. They allow for the refinement of the learned feature maps by incorporating high-frequency details extracted from the wavelet transform, first applied to the 3T input image and then iteratively to the approximation coefficients.

Image reconstruction branch. This sub-network receives the modulated feature maps obtained from the feature extraction branch and learns the *residual image* that is added afterwards to the 3T input images. The architecture inversely replicates the feature extraction branch without including the WAT layers and with Conv2D(64, 1, 2, 0) + ReLu in the depression unit replaced by Conv2D(64, 3, 1, 0) + ReLu + DeConv2D.

Finally, we add Conv2D(64, 3, 1, 1) + ReLU and Conv2D(64, 3, 1, 1) + tanh to reconstruct an output of the same dimensions as the input. The tanh activation is used instead of the sigmoid to make faster convergence. In fact, as the input values are in the range [0,1] and the CNN is large, we are likely to face vanishing gradients, hence slow convergence. The values of the tanh's derivative around 0 are larger that the sigmoid's, allowing for strong gradients and big learning steps.

3 Experimental setting

Loss function. As advised in the original paper, we use the Mean Absolute Error (MAE) as the loss function for the 2 networks. In fact, since the images used are normalised between $[0, 1]$, the error values are too small (between -1 and 1 at most). Thus, there is no need to use the MSE which penalizes large errors. Using MAE allows even better convergence.

	Worst case	Basic CNN	WATNet
	mean ± *standard deviation*		
MAE	**0.012** ± 0.004	0.014 ± 0.004	**0.012** ± 0.004
PSNR	28.759 ± 2.887	28.028 ± 2.020	**28.890** ± 2.882
SSIM	0.949 ± 0.017	**0.953** ± 0.016	0.949 ± 0.018

Table 1: Mean and standard deviation of the MAE, PSNR, and SSIM on the 10 7T original/synthesized MRI images. The worst case corresponds to directly comparing the 3T to the 7T images.

Training and validation. Due to the scarcity of the data available to train and test the networks (10 MRI images), we, therefore, opt for the *leave-one-out* cross-validation. In practice, we perform 10 trainings of each model; at each one, the network is trained on 9 subject images and tested on the MRI images of the remaining subject. Furthermore, it is noteworthy to mention that the 2 networks are trained on $64 \times 64 \times 3$ patches at a time. This is equivalent to constraining the receptive field of the networks to focus on learning localized relevant information at a time and thus achieving better performance.

During the validation phase, however, the inference is made on the whole input images instead of small patches as in the training. Otherwise, patch-by-patch inference is time-consuming and does not guarantee to have consistency in contrast between synthesized patches.

Parameters setting. For the sake of legitimate comparison, we set the parameters of the 2 networks (BasicCNN and WATNet) to the same values. Namely, we train the models on 10 epochs (empirically observed to be sufficient), with a linear learning rate decay from 0.001 to 0.0001 and the number of decay steps set to 10. We choose *Adam* as the optimizer algorithm for the 2 networks.

It is important to mention that, since the WATNet is relatively large and complex, the vanishing/exploding gradients might be an issue. The filter weights in the convolutional layers of the WATNet are therefore initialized with the *HeNormal* initializer that draws random samples from a normal distribution with variance inversely proportional to the number of incoming neurons. In doing so, the initialization is done keeping into account the size of the previous layer which helps in attaining a global minimum of the cost function faster and more efficiently.

Performance metrics. We use the Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) metrics to report the performance of the models on the 10 MRI images when used in the validation set.

4 Results and discussion

In this section, we present the performance of the BasicCNN and WATNet models under the experimental setting described in section 3. As a baseline for comparison, we also compare the results to those of the worst-case scenario; that is when the synthesized 7T images are the original 3T images. The results are shown in table 1.

For the quantitative evaluation, the WATNet slightly outperforms the BasicCNN in terms of MAE and signal-to-noise ratio, but with a larger standard deviation for the PSNR. However, in terms of structural similarity, BasicCNN achieves better results than the WATNet, although the latter incorporates high-frequency details from the wavelet transform to reconstruct the 7T images.

However, it is insufficient to judge the models merely in terms of the quality metrics. We thus make a qualitative comparison of the obtained 7T images in figures 3 and 4 (all synthesized 7T images are provided in figures 5, 6 in the appendix A).

The BasicCNN model succeeds at removing the noise and smoothes out the borders of the brain regions, unlike the WATNet, hence the good SSIM metric value. Yet, this also results in blurring the fine details in the brain images. Although WATNet achieves good results in terms of MAE and PSNR, figure 4 shows that there is bearly a difference between the original 3T images and the 7T images synthesized using the WATNet. It is therefore not surprising that it results in almost the same metric values as in the worst case (cf. table 1), which is inconsistent with the results outlined in the original

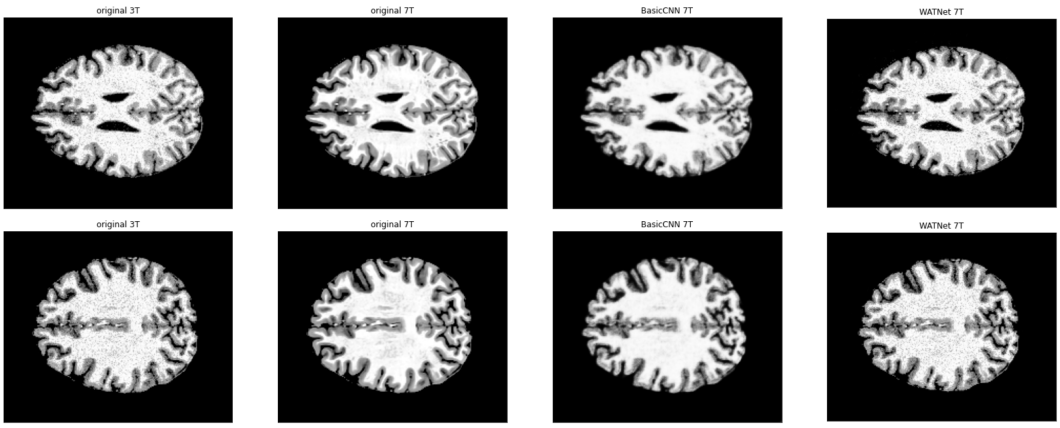


Figure 3: Comparison of the 100th slice of 2 synthesized 7T images using BasicCNN and WATNet.

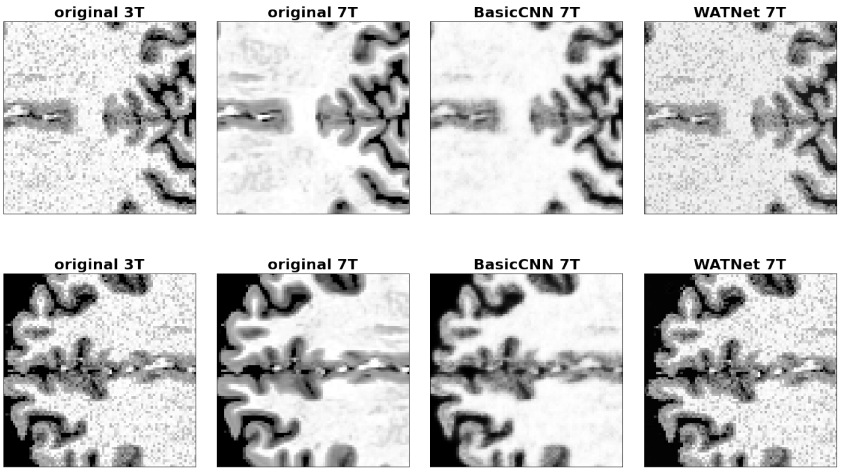


Figure 4: Comparison of 2 patches from the synthesized 7T MRI images.

paper and the efficiency of the WAT layers is not manifest. By contrast, the BasicCNN performs better in that it succeeds in removing the added Gaussian noise.

This discrepancy in performance may be highly attributed to the difference in the training datasets. Whereas the authors used real 3T and 7T images, we used artificially generated 3T images through the addition of random gaussian noise to the 7T images, which does not incorporate any prior/background information about the structure of the brain tissues.

5 Conclusion and discussion

This project aims at reproducing the work of Qu et al. [2020] in synthesizing 7T MRI from 3T MRI images. The novelty of their work is manifested in the insertion of WAT layers that allow for extracting relevant high-frequency information obtained through the wavelet transform and used to modulate the spatial content.

However, the WATNet did not result in the same performance as described in the original paper, especially in terms of the quality of synthesized 7T images. In fact, the BasicCNN consisting of simple blocks of Conv2D + BatchNorm + ReLU/sigmoid performed better. One of the reasons for this can be linked to the fact that the 3T images used by Qu et al. [2020] are real ones, whereas we only generated them from the 7T images through the addition of random noise. Moreover, it is unfortunate

that such proposed work in medical Machine Learning lacks accurate details of implementation that might be crucial for results reproduction.

To make room for improving the WATNet model, it might be interesting to apply a mask to the inferred 7T images before computing the loss. In fact, since we are only interested in reproducing the brain voxels (not the dark background), the loss will no longer take into account errors due to discrepancies in the dark background pixels, stimulating the model to only focus on the important regions.

Finally, it can also be interesting to use the WATNet as a generator of a GAN network while considering a discriminator with a PatchGAN architecture. Unlike usual discriminator architectures, PatchGAN tries to classify each patch in an image as real or fake. In doing so, the discriminator captures fine details by focusing on small receptive fields at a time and only penalizes structure at the scale of local image patches, thus hoping to achieve even better results than the WATNet alone.

References

Liangqiong Qu, Yongqin Zhang, Shuai Wang, Pew-Thian Yap, and Dinggang Shen. Synthesized 7t mri from 3t mri via deep learning in spatial and wavelet domains. *Medical Image Analysis*, 62:101663, 2020. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101663. URL `https://www.sciencedirect.com/science/article/pii/S1361841520300293`.

A Synthesized 7T images of all subjects

We provide the synthesized 7T images from the 3T images using BasicCNN and WATNet in figures 5 and 6.

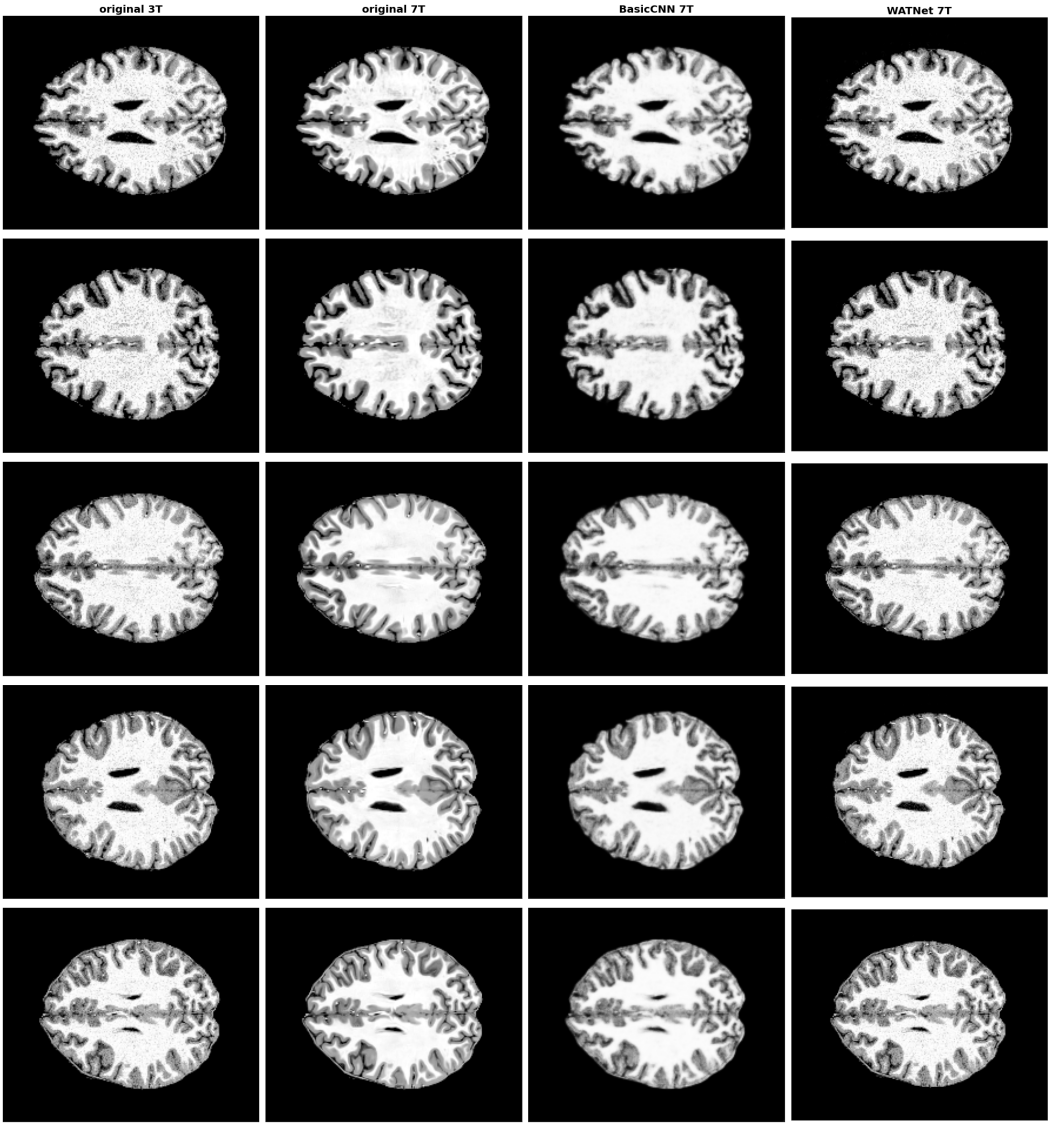


Figure 5: Comparison of synthesized 7T images using BasicCNN and WATNet for the first 5 subjects.

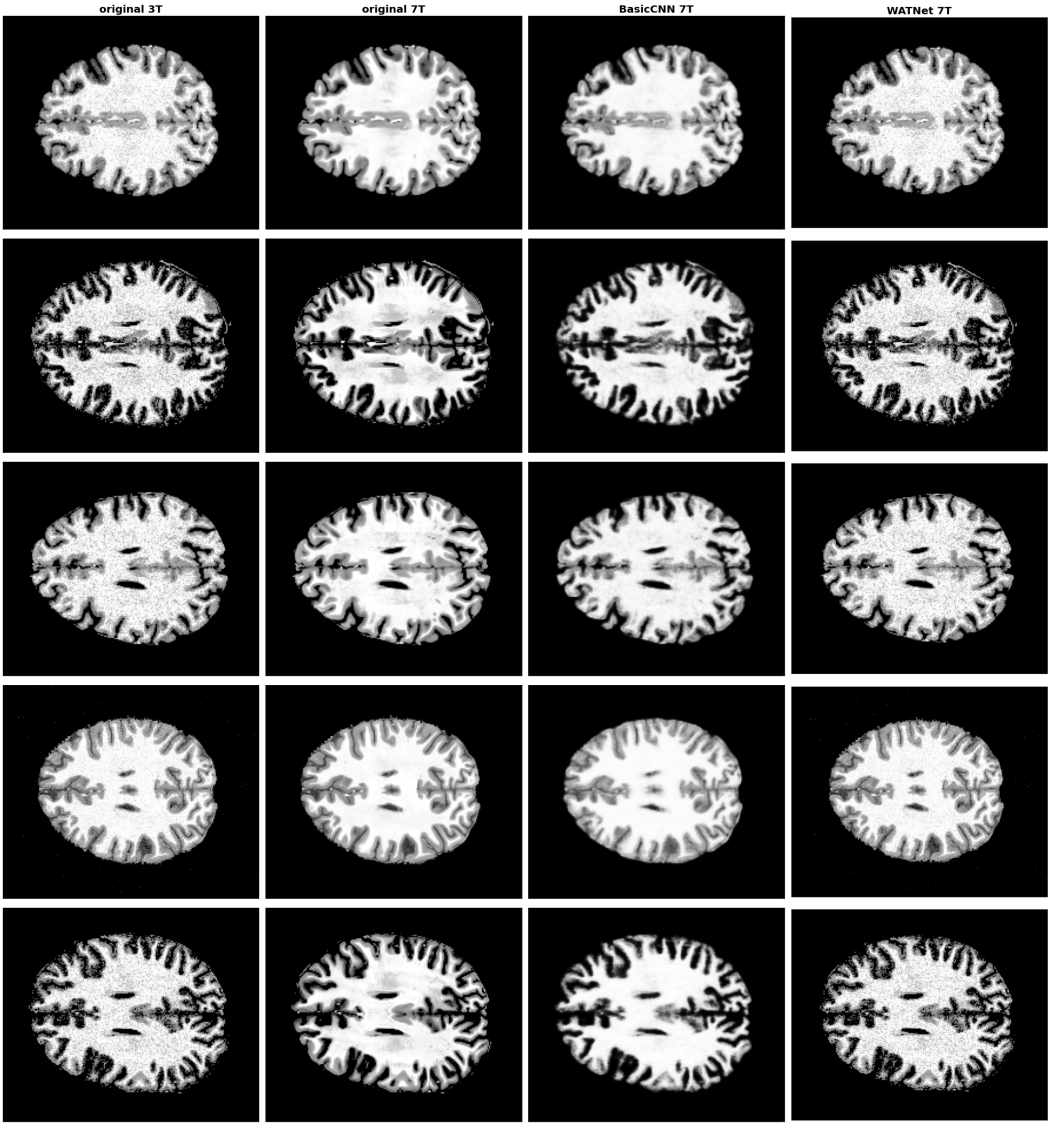


Figure 6: Comparison of synthesized 7T images using BasicCNN and WATNet for the last 5 subjects.