

PAPER • OPEN ACCESS

## Framelet pooling aided deep learning network: the method to process high dimensional medical data

To cite this article: Chang Min Hyun *et al* 2020 *Mach. Learn.: Sci. Technol.* 1 015009

View the [article online](#) for updates and enhancements.

You may also like

- [Theoretical and computational aspects of scattering from periodic surfaces: two-dimensional perfectly reflecting surfaces using the spectral-coordinate method](#)  
J DeSanto, G Erdmann, W Hereman et al.
- [Online prediction of respiratory motion: multidimensional processing with low-dimensional feature learning](#)  
Dan Ruan and Paul Keall
- [Summary of Papers](#)  
Serge Gauthier, Snezhana I Abarzhi and Katepalli R Sreenivasan



## PAPER

# Framelet pooling aided deep learning network: the method to process high dimensional medical data

## OPEN ACCESS

RECEIVED  
21 July 2019

REVISED  
8 October 2019

ACCEPTED FOR PUBLICATION  
19 November 2019

PUBLISHED  
25 February 2020

Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Chang Min Hyun<sup>1</sup> , Kang Cheol Kim<sup>1</sup>, Hyun Cheol Cho<sup>1</sup>, Jae Kyu Choi<sup>2,3</sup> and Jin Keun Seo<sup>1</sup>

<sup>1</sup> Department of Computational Science and Engineering, Yonsei University, Seoul, Republic of Korea

<sup>2</sup> School of Mathematical Sciences, Tongji University, Shanghai, 200092, People's Republic of China

<sup>3</sup> Author to whom any correspondence should be addressed.

E-mail: [jajcjk@tongji.edu.cn](mailto:jajcjk@tongji.edu.cn)

**Keywords:** deep learning, CNN, framelet, sparse-view CT, undersampled MRI

## Abstract

Machine learning-based analysis of medical images often faces several hurdles, such as the lack of training data, the curse of the dimensionality problem, and generalization issues. One of the main difficulties is that there exists a computational cost problem in dealing with input data of large size matrices which represent medical images. The purpose of this paper is to introduce a framelet-pooling aided deep learning method for mitigating computational bundles caused by large dimensionality. By transforming high dimensional data into low dimensional components by filter banks and preserving detailed information, the proposed method aims to reduce the complexity of the neural network and computational costs significantly during the learning process. Various experiments show that our method is comparable to the standard unreduced learning method, while reducing computational burdens by decomposing large-sized learning tasks into several small-scale learning tasks.

## 1. Introduction

Recently, medical imaging is experiencing a paradigm shift due to a remarkable and rapid advance in deep learning techniques. Deep learning techniques have expanded our ability via sophisticated disentangled representation learning through training data, and appear to show superiority of performance in various medical imaging problems including undersampled magnetic resonance imaging (MRI), sparse-view computed tomography (CT), artifact reduction, organ segmentation, and automated disease detection. In particular, U-net (Ronneberger *et al* 2015), a kind of convolutional neural network, seems to show remarkable capability of learning image representations. However, there are some hurdles to overcome, one of which comes from the high dimensionality, i.e., the high pixel dimension in 2D or 3D, of medical images. This paper addresses a way to resolve this issue through a so-called *framelet pooling aided deep learning network*.

Machine learning performance is closely related to the number, the quality, and the pixel dimensionality of the sampled data. For ease of explanation, let us consider a simple question to learn an unknown function  $f : [0, 1]^d \mapsto [0, 1]$  from a given sample  $(\mathbf{x}, y)$ , where  $\mathbf{x}$  is an input gray scale image lying in  $[0, 1]^d$  and  $y = f(\mathbf{x})$  is the corresponding output on the interval  $[0, 1]$ . Then one can ask how many training samples are needed to approximate  $f$  with a given tolerance  $\epsilon > 0$ . It is well-known that for Lipschitz continuous function  $f$ , we need to sample  $O(\epsilon^{-d})$  points (Mallat 2016). In addition, the author in Barron (1994) observed that the estimation error of the function  $f$  by 1 hidden layer neural networks is given by  $O\left(\frac{c_f}{m}\right) + O\left(\frac{md}{n_{\text{data}}}\log n_{\text{data}}\right)$ , where  $n_{\text{data}}$  is the number of training data,  $m$  is the number of neurons in the hidden layer, and  $c_f$  is a constant depending on the regularity of  $f$ . This means that in the case of  $d = 512^2$  (i.e. considering  $512 \times 512$  images) and  $m = d$ , we roughly need huge training data  $n_{\text{data}} = O(10^{12})$  to achieve the error of  $O(10^{-1})$ . This high number of required training data makes the problem intractable, especially when data lies in the high dimensional space. Such a phenomenon is referred as the *curse-of-dimensionality* in approximation sense. Even though the effect of dimensionality on deep networks is relatively weaker than shallow ones (Bruna and Mallat 2013, Pascanu *et al* 2013,

Mhaskar and Poggio (2016) in approximation sense, deep learning requires huge computational scale for training process. Thus, deep networks with high dimensional data also experience the curse-of-dimensionality in terms of computational burden.

In the literature, framelets are known to be effective in capturing key information of images. This is due to the multiscale structure of the framelet systems, and the presence of both low pass and high pass filters in the filter banks, which are desirable in sparsely approximating images without loss of information (Dong *et al* 2017). In this work, we propose a framelet-based deep learning method to reduce computational burdens for dealing with high dimensional data in the learning process. This method, called a *framelet pooling*, is based on the decomposition of a  $d$ -dimensional input-output pair  $(\mathbf{x}, \mathbf{y})$  into several  $d/2^{2k}$ -dimensional pairs  $\{(\mathcal{W}_{k,\alpha}\mathbf{x}, \mathcal{W}_{k,\alpha}\mathbf{y}) : \alpha = 1, \dots, r\}$ , where each  $\mathcal{W}_{k,\alpha}$  and  $\mathcal{W}_{k,\alpha}$  are  $d/2^{2k} \times d$  matrices corresponding to  $k$ th level framelet packet transform (Mallat 2009). Instead of learning the pair of high dimensional original data  $(\mathbf{x}, \mathbf{y})$ , the proposed method tries to learn much lower dimensional pairs  $(\mathcal{W}_{k,\alpha}\mathbf{x}, \mathcal{W}_{k,\alpha}\mathbf{y})$  in parallel fashion, so that we can achieve the computational efficiency in dealing with the large size images.

As an application of our proposed method, we deal with the undersampled MRI (Hyun *et al* 2018) and the sparse-view CT problem (Jin *et al* 2017), where huge memory problems may arise in recovering high resolution images. Experiments on undersampled MRI and sparse-view CT show that our framelet pooling aided reduced method provides very similar performance to the standard unreduced method, while reducing the computation time greatly by reducing the dimension of inputs and learning parameters in neural networks.

## 2. Method

Both undersampled MRI and sparse-view CT problem aim to find a reconstruction function  $f$ , which maps from an undersampled data  $\mathbf{P}^\sharp$  (violating Nyquist criteria) to a clinically meaningful tomographic image  $\mathbf{y}$ . Here, the undersampled data  $\mathbf{P}^\sharp$  can be expressed as the subsampling of the fully-sampled data  $\mathbf{P}$  (satisfying the Nyquist criterion)

$$\mathbf{P}^\sharp = \mathcal{S}\mathbf{P}, \quad (1)$$

where  $\mathcal{S}$  is a subsampling operator. The standard MRI and CT use the fully-sampled data  $\mathbf{P}$  to provide tomographic images, where the reconstruction functions  $f$  in MRI and CT are the inverse Fourier transform and inverse Radon transform, respectively. However, when we use the undersampled data  $\mathbf{P}^\sharp$ , these standard methods do not work as the Nyquist criterion is not satisfied any more. (See figures 1 and 2.) For the sake of clarity, we shall briefly state the mathematical framework of undersampled MRI and sparse-view CT in the following subsections.

### 2.1. Undersampled MRI

Let  $\mathbf{y}(z)$  be a distribution of nuclear spin density at the position  $z = (z_1, z_2)$ . The measured k-space data  $\mathbf{P}$  is governed by the Fourier relation

$$\mathbf{P} = \mathcal{F}\mathbf{y} = \int_{\mathbb{R}^2} \mathbf{y}(z) e^{-2\pi i z \cdot \xi} dz, \quad (2)$$

where  $\xi = (\xi_1, \xi_2)$  (Nishimura 2010). Therefore, with the fully-sampled data  $\mathbf{P}$ , the reconstruction image  $\mathbf{y}$  can be obtained by taking the inverse Fourier transform to the measured data  $\mathbf{P}$ ,

$$\mathbf{y} = \mathcal{F}^{-1}\mathbf{P}. \quad (3)$$

Note that the direct inversion method (3) can also be applied to the undersampled data  $\mathbf{P}^\sharp$ ,

$$\mathbf{y}^\sharp = \mathcal{F}^{-1}\mathcal{S}^*\mathbf{P}^\sharp. \quad (4)$$

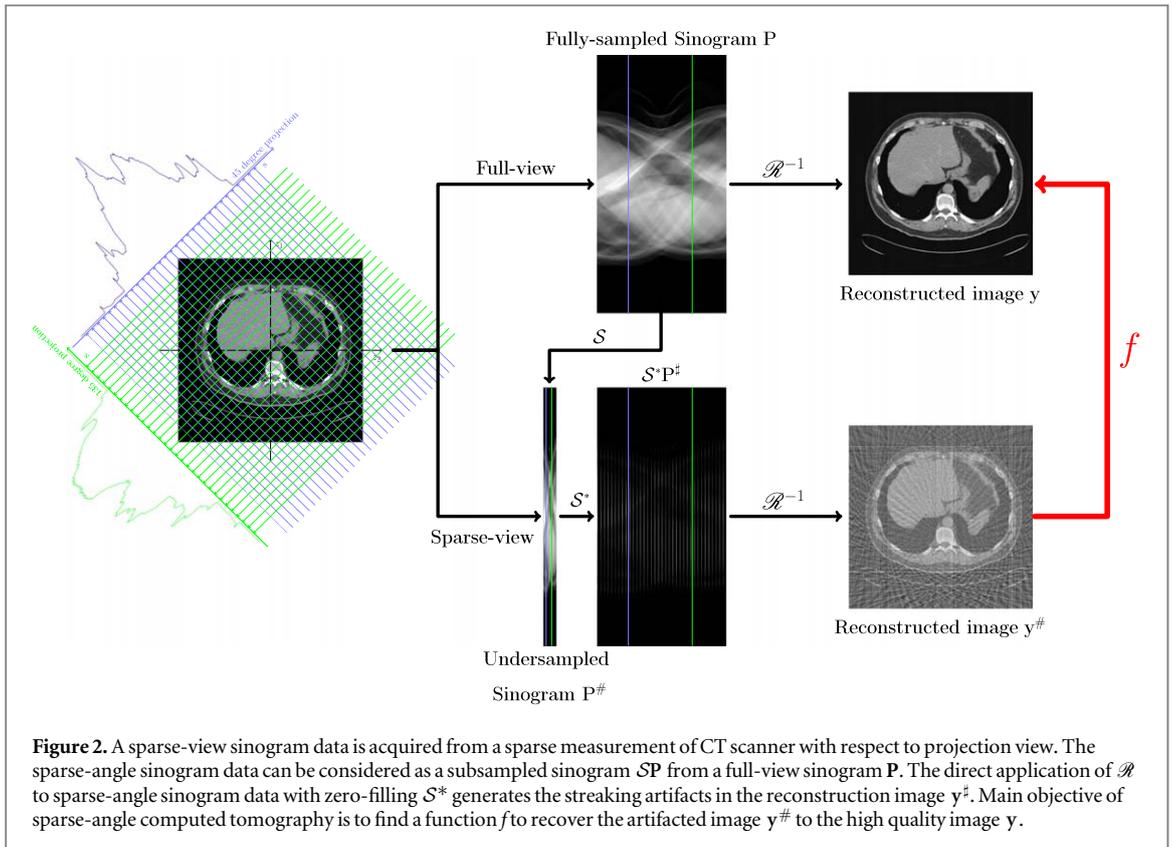
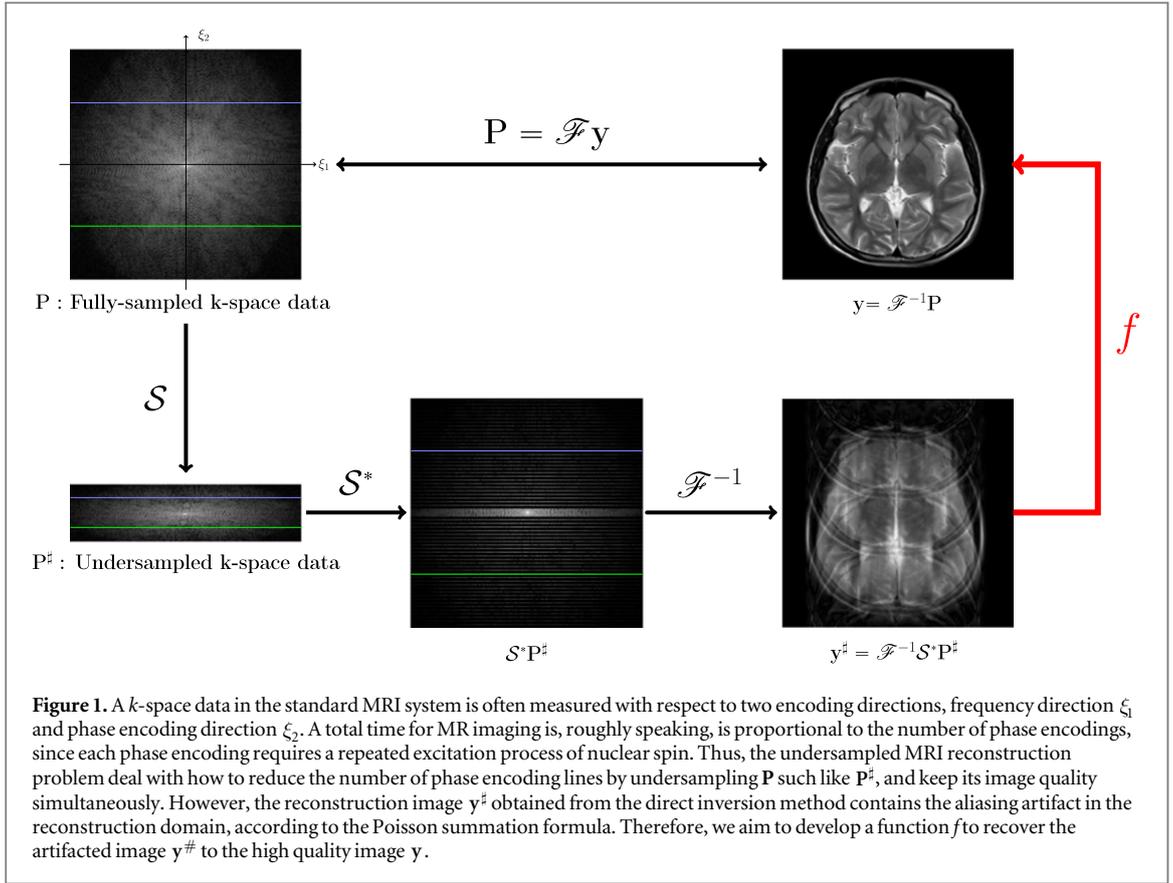
Here,  $\mathcal{S}^*$  is an adjoint operator of  $\mathcal{S}$  in the  $\ell^2$  space. However, the image  $\mathbf{y}^\sharp$  obtained from (4) contains aliasing artifacts as  $\mathbf{P}^\sharp$  violates the Nyquist criterion (see figure 1).

### 2.2. Sparse-view CT

In CT, the tomographic image  $\mathbf{y}(z)$  can be regarded as the distribution of linear attenuation coefficients at the position  $z = (z_1, z_2)$ . For CT data acquisition, x-ray beams are transmitted at various directions  $\theta := (\cos \varphi, \sin \varphi)$ ,  $0 \leq \varphi \leq 2\pi$ . Under the assumption of monochromatic x-ray generation, the projection data  $\mathbf{P}$  at the direction  $\theta$  is dictated by the following Radon transform

$$\mathbf{P} = \mathcal{R}\mathbf{y} = \int_{L_{\theta,s}} \mathbf{y}(z) d\ell_z, \quad (5)$$

where  $L_{\theta,s}$  is the projection line  $L_{\theta,s} := \{z \in \mathbb{R}^2 : \theta \cdot z = s\}$  (Seo and Woo 2013). With the fully-sampled data  $\mathbf{P}$  satisfying the Nyquist criterion,  $\mathbf{y}$  can be reconstructed by the inverse Radon transform



$$\mathbf{y} = \mathcal{R}^{-1}\mathbf{P}. \tag{6}$$

For the undersampled data  $\mathbf{P}^\sharp$ , which is measured with the low sampling frequency along the projection-view, we can apply the direct inversion formula (6) by filling zeros to unmeasured parts of undersampled data

$$\mathbf{y}^\sharp = \mathcal{R}^{-1} \mathcal{S}^* \mathbf{P}^\sharp. \quad (7)$$

However, the reconstruction image  $\mathbf{y}^\sharp$  contains streaking artifacts, which result from the violation of Nyquist criterion. Figure 2 shows the schematic and visual descriptions of the sparse-view CT problem.

### 2.3. Preliminaries on framelets

Provided here is a brief introduction on tight frames and framelets. Interested readers may consult e.g. (Ron and Shen 1997, Dong and Shen 2012, Shen and Xu 2013) for the detailed surveys. In this paper, we only consider the 2-dimensional case as we focus on the undersampled 2-dimensional image reconstruction. Note, however, that it is not hard to generalize into  $d$ -dimensional cases with  $d \geq 3$ .

For a given  $\Psi = \{\psi_1, \dots, \psi_r\} \subseteq L_2(\mathbb{R}^2)$ , an affine system  $X(\Psi)$  is the collection of the dilations and the shifts of the elements in  $\Psi$ :

$$X(\Psi) = \{\psi_{\alpha,n,\mathbf{k}} := 2^n \psi_\alpha(2^n \cdot -\mathbf{k}) : 1 \leq \alpha \leq r, n \in \mathbb{Z}, \mathbf{k} \in \mathbb{Z}^2\}. \quad (8)$$

We say that  $X(\Psi)$  is a tight wavelet frame on  $L_2(\mathbb{R}^2)$  if we have

$$\|f\|_{L_2(\mathbb{R}^2)}^2 = \sum_{\alpha=1}^r \sum_{n \in \mathbb{Z}} \sum_{\mathbf{k} \in \mathbb{Z}^2} |\langle f, \psi_{\alpha,n,\mathbf{k}} \rangle|^2 \quad (9)$$

for every  $f \in L_2(\mathbb{R}^2)$ . In this case, each  $\psi_\alpha$  is called a framelet, and  $\langle f, \psi_{\alpha,n,\mathbf{k}} \rangle$  is called the canonical coefficient of  $f$ .

The constructions of (anti-)symmetric and compactly supported framelets  $\Psi$  are usually based on a multiresolution analysis (MRA); we first find some compactly supported refinable function  $\phi$  with a refinement mask  $\mathbf{q}_0$  such that

$$\phi = 2^2 \sum_{\mathbf{k} \in \mathbb{Z}^2} \mathbf{q}_0(\mathbf{k}) \phi(2 \cdot -\mathbf{k}). \quad (10)$$

Then the MRA based construction of  $\Psi = \{\psi_1, \dots, \psi_r\} \subseteq L_2(\mathbb{R}^2)$  is to find finitely supported masks  $\mathbf{q}_\alpha$  such that

$$\psi_\alpha = 2^2 \sum_{\mathbf{k} \in \mathbb{Z}^2} \mathbf{q}_\alpha(\mathbf{k}) \phi(2 \cdot -\mathbf{k}), \quad \alpha = 1, \dots, r. \quad (11)$$

The sequences  $\mathbf{q}_1, \dots, \mathbf{q}_r$  are called wavelet frame mask or the high pass filters of the system, and the refinement mask  $\mathbf{q}_0$  is also called the low pass filter.

The unitary extension principle of (Ron and Shen 1997) provides a general theory of the construction of MRA based tight wavelet frames. Briefly speaking, as long as  $\{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_r\}$  are compactly supported and their Fourier series  $\hat{\mathbf{q}}_\alpha(\boldsymbol{\xi}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} \mathbf{q}_\alpha(\mathbf{k}) e^{-i\boldsymbol{\xi} \cdot \mathbf{k}}$  satisfy

$$\sum_{\alpha=0}^r |\hat{\mathbf{q}}_\alpha(\boldsymbol{\xi})|^2 = 1 \quad \text{and} \quad \sum_{\alpha=0}^r \hat{\mathbf{q}}_\alpha(\boldsymbol{\xi}) \overline{\hat{\mathbf{q}}_\alpha(\boldsymbol{\xi} + \boldsymbol{\nu})} = 0 \quad (12)$$

for all  $\boldsymbol{\nu} \in \{0, \pi\}^2 \setminus \{0\}$  and  $\boldsymbol{\xi} \in [-\pi, \pi]^2$ , the affine system  $X(\Psi)$  with  $\Psi = \{\psi_1, \dots, \psi_r\}$  defined by (11) forms a tight frame of  $L_2(\mathbb{R}^2)$ , and the filters  $\{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_r\}$  form a discrete tight frame on  $\ell_2(\mathbb{Z}^2)$ .

In the discrete setting, the first level framelet decomposition operator  $\mathcal{W}^{(1)}$  is defined as

$$\mathcal{W}^{(1)} = [\mathcal{W}_{0,0}^T, \mathcal{W}_{0,1}^T, \dots, \mathcal{W}_{0,r}^T]^T \quad (13)$$

where  $\mathcal{W}_{0,\alpha}$  is the  $d^2/2^{-2} \times d^2$  matrix given by

$$\mathcal{W}_{0,\alpha} \mathbf{x} = \downarrow (\mathbf{x} \otimes \mathbf{q}_\alpha(-\cdot)), \quad \forall \mathbf{x} \in \mathbb{R}^{d^2}.$$

Here,  $\downarrow$  stands for 2-dimensional down-sampling operator and  $\otimes$  is convolution operator with stride 1. Likewise, we can define the second level framelet decomposition  $\mathcal{W}^{(2)}$  by

$$\begin{aligned} \mathcal{W}^{(2)} = & [(\mathcal{W}_{1,0} \mathcal{W}_{0,0})^T, \dots, (\mathcal{W}_{1,0} \mathcal{W}_{0,r})^T, \\ & \dots, (\mathcal{W}_{1,r} \mathcal{W}_{0,0})^T, \dots, (\mathcal{W}_{1,r} \mathcal{W}_{0,r})^T]^T, \end{aligned} \quad (14)$$

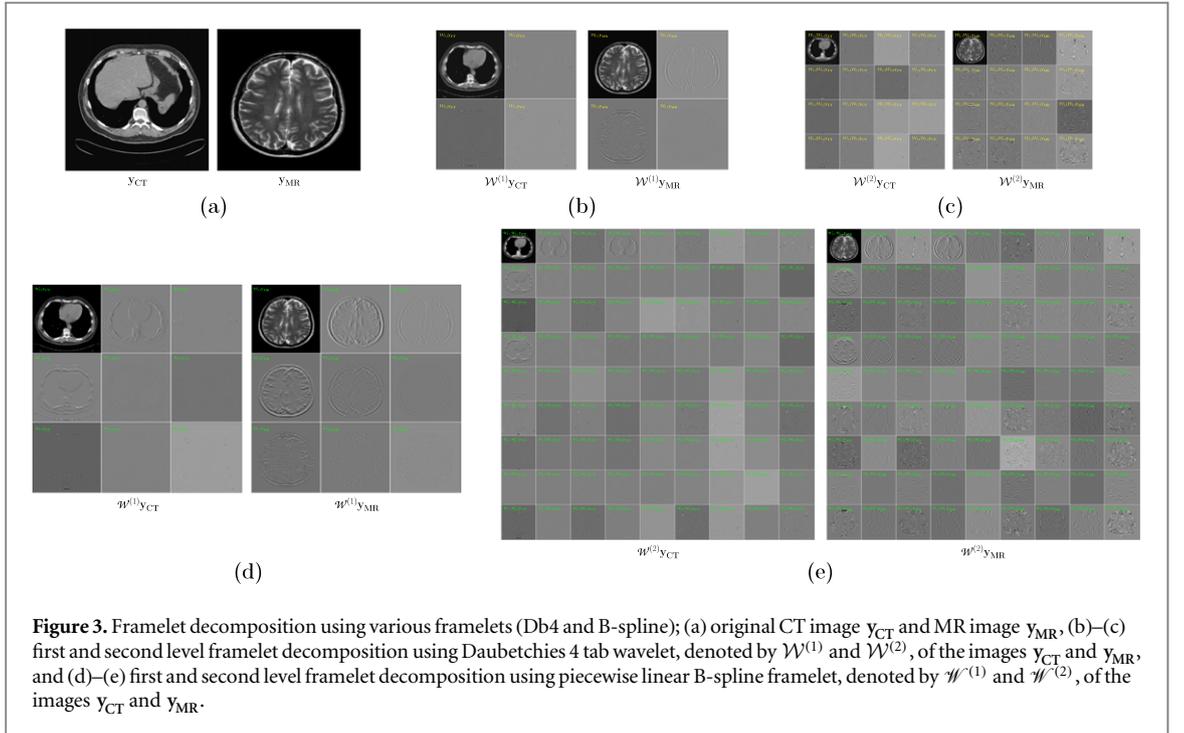
where  $\mathcal{W}_{1,\alpha}$  is the  $d^2/2^{-4} \times d^2/2^{-2}$  matrix given by

$$\mathcal{W}_{1,\alpha} \tilde{\mathbf{x}} = \downarrow (\tilde{\mathbf{x}} \otimes \mathbf{q}_\alpha(-\cdot)), \quad \forall \tilde{\mathbf{x}} \in \mathbb{R}^{d^2/2^{-2}}.$$

We can continue the above process to define the  $k$ th level framelet decomposition operator  $\mathcal{W}^{(k)}$ . Then since the filter  $\{\mathbf{q}_\alpha\}$  satisfies (12), we have  $(\mathcal{W}^{(l)})^T \mathcal{W}^{(l)} = \mathcal{I}$  for each  $l = 1, 2, \dots$ . Finally, figure 3 illustrates two examples of framelet decompositions using Daubechies wavelet (db4) (Daubechies 1988) and piecewise linear B-spline frame (Shen and Xu 2013).

### 2.4. Proposed deep learning approach for undersampled reconstruction

The objective of the undersampled reconstruction problem is to develop a deartifacting map  $f$ , which converts  $\mathbf{y}^\sharp \in \mathbb{R}^{d^2}$  (artifactual image) to  $\mathbf{y} \in \mathbb{R}^{d^2}$  (artifact removed image) with  $d^2$  being a pixel dimension of reconstructed



**Figure 3.** Framelet decomposition using various framelets (Db4 and B-spline); (a) original CT image  $y_{CT}$  and MR image  $y_{MR}$ , (b)–(c) first and second level framelet decomposition using Daubechies 4 tab wavelet, denoted by  $\mathcal{W}^{(1)}$  and  $\mathcal{W}^{(2)}$ , of the images  $y_{CT}$  and  $y_{MR}$ , and (d)–(e) first and second level framelet decomposition using piecewise linear B-spline framelet, denoted by  $\mathcal{W}^{(1)}$  and  $\mathcal{W}^{(2)}$ , of the images  $y_{CT}$  and  $y_{MR}$ .

image. In particular, deep learning techniques, such as U-net, infer  $f$  by minimizing training data-fidelity :

$$f = \operatorname{argmin}_{f \in \mathbb{D}_{L_{net}}} \sum_{i=1}^N \mathcal{L}(f(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}) \quad (15)$$

using a set of training data  $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})_{i=1}^N$ . Here,  $N$  is the number of training data,  $\mathbf{x}^{(i)}$  denotes the artifact image instead of  $(\mathbf{y}^\sharp)^{(i)}$ ,  $\mathbb{D}_{L_{net}}$  is a set of all learnable functions from a user-defined deep learning network architecture, and  $\mathcal{L}$  is a user-defined energy-loss function to evaluate the metric between deep learning output  $f(\mathbf{x}^{(i)})$  and label  $\mathbf{y}^{(i)}$ . However, if the pixel dimension of input increases, the total computational complexity in the training process increase largely. To address this curse-of-dimensionality issue, we propose the framelet pooling aided deep learning method to learn the deartifacting map  $f$  indirectly.

Let  $\mathcal{W}$  and  $\mathcal{W}$  be framelet decomposition operators defined as in subsection 2.3. The proposed framelet pooling deep learning network aims to infer the relation between  $\mathcal{W}^{(k_1)}\mathbf{x}$  and  $\mathcal{W}^{(k_2)}\mathbf{y}$  in the following least-squared minimization sense:

$$\mathbf{f} = \operatorname{argmin}_{\mathbf{f} \in \mathbb{D}_{L_{net}}} \sum_{i=1}^N \mathcal{L}(\mathbf{f}(\mathcal{W}^{(k_1)}\mathbf{x}^{(i)}), \mathcal{W}^{(k_2)}\mathbf{y}^{(i)}). \quad (16)$$

Here, each  $(\mathcal{W}^{(k_1)}\mathbf{x}^{(i)})_{\alpha_1}$  and  $(\mathcal{W}^{(k_2)}\mathbf{y}^{(i)})_{\alpha_2}$  are images with  $d^2/2^{-2k_1}$  and  $d^2/2^{-2k_2}$  pixel dimension, respectively. For example, let  $\mathcal{W}^{(2)}$  be the second level Daubechies 4 tab wavelet decomposition. If the second level Daubechies 4 tab wavelet decomposition is taken for  $\mathcal{W}^{(k_1)}$  and  $\mathcal{W}^{(k_2)}$  in the equation (16), the proposed deep learning method tries to find the function  $\mathbf{f}$  satisfying  $\mathbf{f}(\mathcal{W}^{(2)}\mathbf{x}) = \mathcal{W}^{(2)}\mathbf{y}$  in the sense of (16), as shown in figure 4.

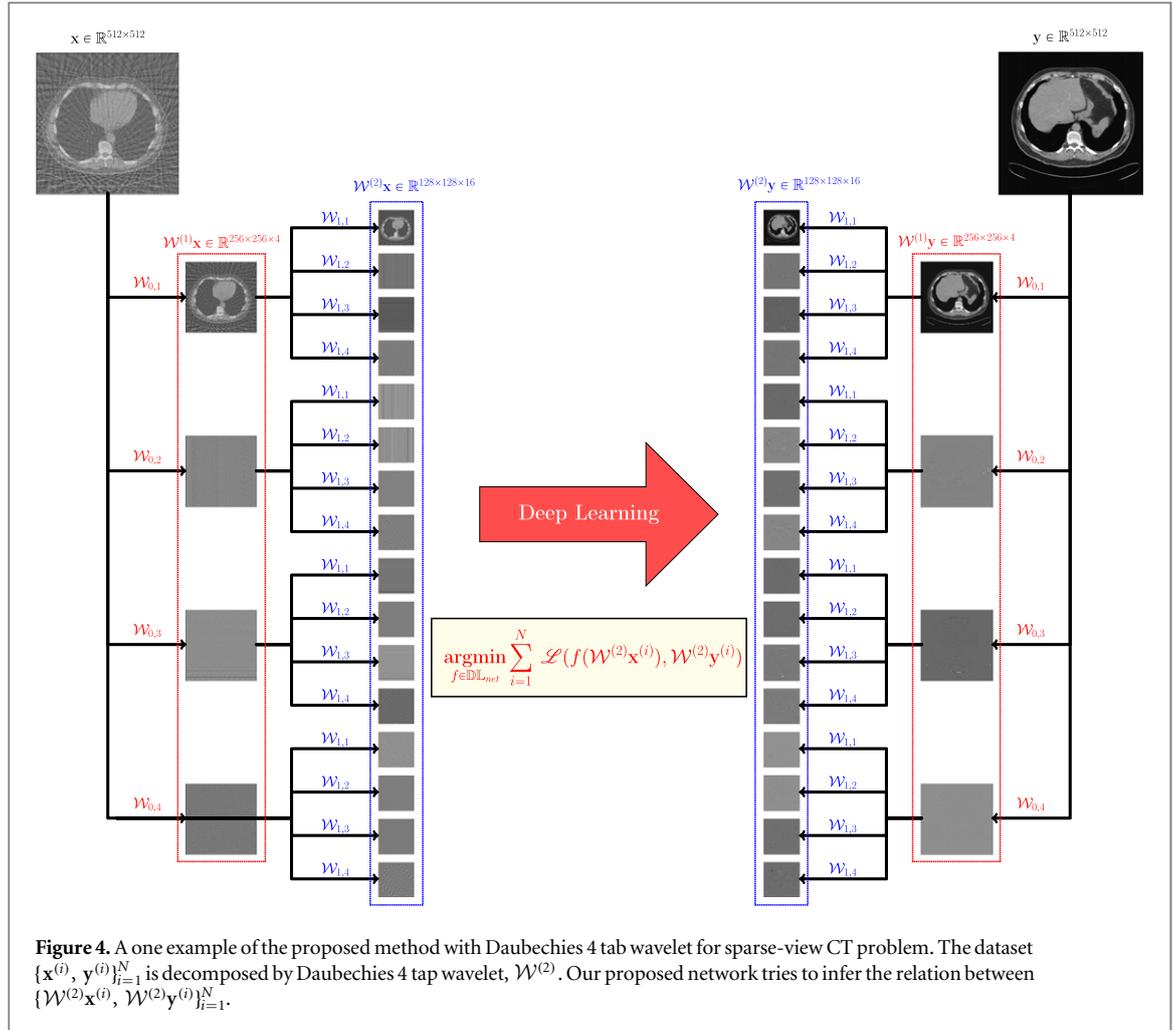
Compared to the direct deep learning scheme (15), the framelet-pooling aided deep learning method (16) is expected to mitigate the total computational complexity and time caused by high dimensional data in the learning process. In this paper, we test only the case that training inputs and labels are decomposed using same framelet decomposition  $\mathcal{W}^{(k)}$ . However, our method is not restricted only in this specific case.

### 3. Experiments and results

#### 3.1. Experimental set-up for undersampled MRI

Let  $\{\mathbf{y}_{MR}^{(i)} \in \mathbb{R}^{256 \times 256}\}_{i=1}^N$  denote the set of MR images reconstructed with the Nyquist sampling. Using  $\{\mathbf{y}_{MR}^{(i)}\}$ , we compute the training input  $\{\mathbf{x}_{MR}^{(i)} \in \mathbb{R}^{256 \times 256}\}_{i=1}^N$  by

$$\mathbf{x}_{MR}^{(i)} = \mathcal{F}^{-1} \mathcal{S}^* \underbrace{\mathcal{S} \mathcal{F} \mathbf{y}_{MR}^{(i)}}_{\mathbf{p}^\sharp}, \quad (17)$$



where  $\mathcal{F}$  is the 2D discrete Fourier transform,  $\mathcal{F}^{-1}$  is the 2-dimensional discrete inverse Fourier transform, and  $\mathcal{S}$  is a specifically user-chosen subsampling operator. In our experiments, we use the MR images  $\mathbf{y}_{\text{MR}}^{(i)}$  obtained from T2-weighted turbo spin-echo pulse sequence with 4408 ms repetition time, 100 ms echo time, and 10.8 ms echo spacing (Loizou *et al* 2011). The Fourier transform and its inverse are computed via `fft2` and `ifft2` in the Python package `numpy.fft`. Finally, for the sampling strategy, we choose the uniform subsampling with factor 4 and 12 additional low frequency sampling among total 256 lines (Hyun *et al* 2018).

In order to test our proposed method, we decompose dataset using  $k$  level framelet decomposition  $\mathcal{W}^{(k)}$  with various filter banks. We obtain

$$\{\mathcal{W}^{(k)}\mathbf{x}_{\text{MR}}^{(i)}, \mathcal{W}^{(k)}\mathbf{y}_{\text{MR}}^{(i)}\}_{i=1}^N, \quad (18)$$

where both  $\mathcal{W}^{(k)}\mathbf{x}_{\text{MR}}^{(i)}$  and  $\mathcal{W}^{(k)}\mathbf{y}_{\text{MR}}^{(i)}$  contains  $r^k$  pairs of  $256/2^{2k} \times 256/2^{2k}$  image. Here,  $k$  is the decomposition level and  $r$  is the number of filter  $\mathbf{q}_\alpha$ .

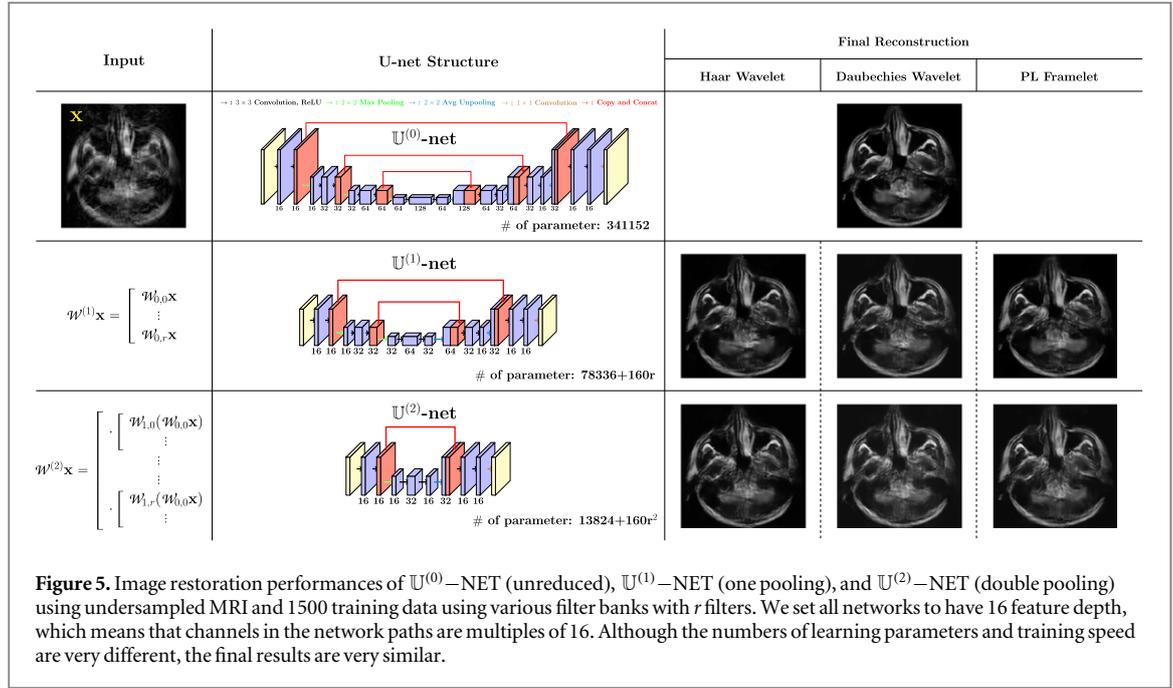
### 3.2. Experimental set-up for sparse-view CT

Let  $\{\mathbf{y}_{\text{CT}}^{(i)} \in \mathbb{R}^{512 \times 512}\}_{i=1}^N$  be a set of CT images reconstructed with the Nyquist sampling. The corresponding deep learning training inputs are computed in the following sense;

$$\mathbf{x}_{\text{CT}}^{(i)} = \mathcal{R}^{-1} \mathcal{S}^* \underbrace{\mathcal{S} \mathcal{R} \mathbf{y}_{\text{CT}}^{(i)}}_{\mathbf{p}^2}, \quad (19)$$

where  $\mathcal{R}$  is the discrete Radon transform,  $\mathcal{R}^{-1}$  is the filtered-back projection algorithm, and  $\mathcal{S}$  is a user-defined sampling operator. In our implementations, we use the projection algorithm `radon` and filtered back-projection algorithm `iradon` in the Python package `skimage.transform` for computing  $\mathcal{R}$  and its inverse  $\mathcal{R}^{-1}$ , respectively. Uniform subsampling with factor 6 in terms of projection-view is also used for  $\mathcal{S}$  in (19).

Applying the same process used to generate a dataset (18) for undersampled MRI experiments, we obtain the following decomposed dataset for sparse-view CT problem;



$$\{\mathcal{W}^{(k)}\mathbf{x}_{CT}^{(i)}, \mathcal{W}^{(k)}\mathbf{y}_{CT}^{(i)}\}_{i=1}^N, \quad (20)$$

where  $\mathcal{W}^{(k)}$  is a  $k$  level framelet decomposition.

In our whole experiments, we use a first and second level framelet decomposition ( $k = 1, 2$ ) with three different framelets (Haar wavelet(Haar), Daubechies 4 tap wavelet(Db4), and piecewise linear B-spline framelet(PL)).

### 3.3. Network configuration

To test our proposed method, we adapt the U-net architecture (Ronneberger *et al* 2015), as shown in figure 5, where the first half of network is the contracting path and the last half is the expansive path. At the first layer in U-net in figure 5, the input  $\mathcal{W}^{(k)}\mathbf{x}$  is convolved with the set of convolution filters  $\mathbf{C}^{(1)}$  so that it generates a set of feature maps  $\mathbf{h}^{(1)}$ , given by

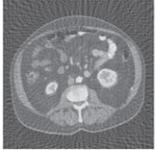
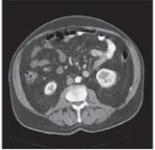
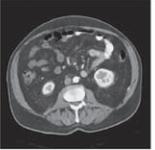
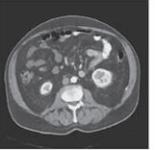
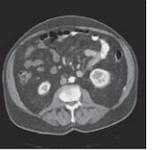
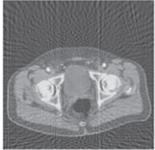
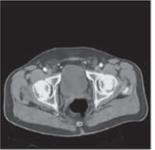
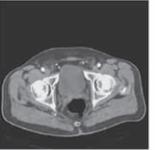
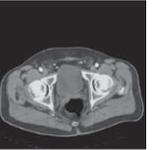
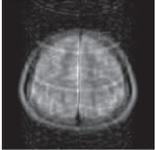
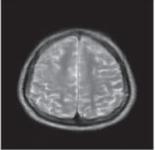
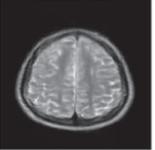
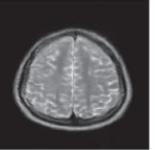
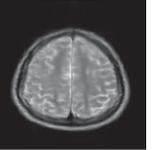
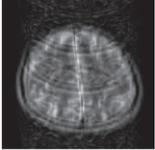
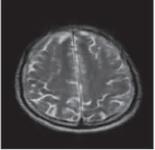
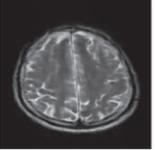
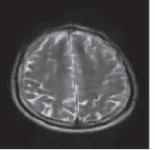
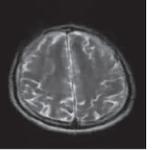
$$\mathbf{h}^{(1)} = \text{ReLU}(\mathbf{C}^{(1)} \otimes_1 \mathcal{W}^{(k)}\mathbf{x}),$$

where  $\text{ReLU}$  is the rectified linear unit  $\text{ReLU}(x) = \max\{x, 0\}$  and  $\otimes_1$  stands for the convolution with stride 1. We repeat this process to get  $\mathbf{h}^{(2)} = \text{ReLU}(\mathbf{C}^{(2)} \otimes_1 \mathbf{h}^{(1)})$  and apply max pooling to get  $\mathbf{h}^{(3)}$ . Through this contracting path, we can obtain low dimensional feature maps by applying either convolution or max pooling. In the expansive path, we use the  $2 \times 2$  average unpooling instead of max-pooling to restore the size of the output. To restore details in image, the upsampled output is concatenated with the correspondingly feature from the contracting path. At the last layer a  $1 \times 1$  convolution is used to combine each feature with one integrated feature (Ronneberger *et al* 2015).

The U-net in the top row of figure 5 will be denoted by  $\mathbb{U}^{(0)}$ -NET. The U-net in the middle row, denoted by  $\mathbb{U}^{(1)}$ -NET, is the reduced network by eliminating two  $3 \times 3$  convolution layers and one pooling/unpooling layer in the first and last part of  $\mathbb{U}^{(0)}$ -NET. Similarly,  $\mathbb{U}^{(2)}$ -NET is the reduced network by eliminating  $3 \times 3$  convolution layers and pooling/unpooling layer in the first and last part of  $\mathbb{U}^{(1)}$ -NET. Thus, this process can be viewed as the replacement of operations with unknown and trainable parameters into framelet operations with known and fixed parameters. In our experiments,  $\mathbb{U}^{(0)}$ -NET is used to learn  $f$  in the sense of direct learning (15). The reduced  $\mathbb{U}^{(k)}$ -NET ( $k = 1, 2$ ) is trained with  $k$  level framelet decomposed dataset in the sense of (16).

### 3.4. Experimental result

All training processes are performed in two Intel(R) Xeon(R) CPU E5-2630 v4, 2.20 GHz, 128 GB DDR4 RAM, and four NVIDIA GTX-1080ti computer system. We initialize all weights by a normal distribution with zero-centered and 0.01 standard deviation, under the Tensorflow environment (Abadi *et al* 2015). We use the  $\ell^2$  loss for the loss function  $\mathcal{L}$ . The loss function is minimized using the Adam Optimizer and the batch normalization for fast convergence (Kingma and Ba 2014, Ioffe and Szegedy 2015). For stability on training, the small learning rate  $10^{-6}$  is used. In order to guarantee the convergence of loss function, the network is trained until the training loss seems to converge sufficiently.

# of data (N=1500)	Input	$\mathbb{U}^{(0)}$ -net	$\mathbb{U}^{(1)}$ -net		
			Haar Wavelet	Daubechies Wavelet	PL Framelet
Sparse-view CT Example 1					
Sparse-view CT Example 2					
Undersampled MRI Example 1					
Undersampled MRI Example 2					

**Figure 6.** Image restoration performances of  $\mathbb{U}^{(0)}$ -NET (unreduced),  $\mathbb{U}^{(1)}$ -NET (one framelet pooling), and  $\mathbb{U}^{(2)}$ -NET (double framelet pooling) using undersampled MRI and 1500 training data.

**Table 1.** Table of the average computational time per epoch(second/epoch) in the undersampled MRI problem.

Average computational time per epoch (second/epoch)							
# of training data	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET		
		Haar	Db4	PL	Haar	Db4	PL
1500	11.798 03	4.664 32	4.430 48	8.323 20	3.636 46	3.757 58	12.988 98
1000	7.650 65	3.431 65	3.258 52	5.616 59	2.441 17	2.532 21	8.312 133
500	3.763 03	1.704 58	1.753 37	2.654 04	1.273 35	1.311 74	4.675 96
100	0.718 29	0.335 26	0.334 08	0.546 98	0.223 54	0.257 95	0.940 70

To avoid overfitting issue, we use the lightened U-net network  $\mathbb{U}^{(0)}$ -net, compared to the original deep U-net structure suggested in the paper (Ronneberger *et al* 2015), and apply the batch normalization technique, known as one efficient way to mitigate the overfitting phenomenon (Ioffe and Szegedy 2015). Noting that test errors, as shown in tables 2 and 4, are approximately zero, there is no overfitting or underfitting.

Figures 5 and 6 show reconstruction results from  $\mathbb{U}^{(0)}$ -NET,  $\mathbb{U}^{(1)}$ -NET, and  $\mathbb{U}^{(2)}$ -NET. Three models show similar reconstruction performances, regardless of their originated problem and their original data dimension. Quantitative evaluations and comparisons for the application on the undersampled MRI problem are summarized in tables 1, 2, and figure 7. For the sparse-view CT application, evaluations and comparisons are given in tables 3, 4, and figure 8. Tables 1 and 3 shows comparisons of average computational time per epoch among  $\mathbb{U}^{(0)}$ -NET,  $\mathbb{U}^{(1)}$ -NET, and  $\mathbb{U}^{(2)}$ -NET. The average computational time is computed by dividing the total computational time by the total number of epoch. Tables 2 and 4 contains test error evaluations and comparisons using two different metrics; mean square error (MSE) and structure similarity (SSIM) (Wang *et al* 2004).

These experimental results support the fact that the proposed method reduces the total computational time efficiently and provides competitive results compared to the direct learning algorithm using high dimensional

**Table 2.** Quantitative test error evaluations for undersampled MRI problem using two different metrics.

MSE( $\sim 10^{-8}$ )							
# of training data	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET		
		Haar	Db4	PL	Haar	Db4	PL
1500	3.427 66	3.734 20	3.716 04	3.613 57	4.752 85	4.962 53	4.725 60
1000	3.723 76	3.998 06	3.975 93	3.931 89	5.030 00	5.032 32	4.917 54
500	4.415 41	4.477 17	4.622 86	4.374 41	5.382 41	5.474 81	5.252 60
100	6.088 67	6.667 04	6.714 41	6.474 48	7.131 72	7.141 36	6.916 11

SSIM							
# of training data	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET		
		Haar	Db4	PL	Haar	Db4	PL
1500	0.812 43	0.792 51	0.798 06	0.799 25	0.746 33	0.737 04	0.736 48
1000	0.801 22	0.787 88	0.788 61	0.790 88	0.735 94	0.735 46	0.744 40
500	0.774 55	0.772 60	0.772 19	0.769 96	0.713 41	0.714 64	0.719 55
100	0.709 40	0.698 64	0.708 57	0.720 80	0.648 15	0.647 67	0.673 78

**Table 3.** Table of the average computational time per epoch in the sparse-view CT problem.

Average computational time per epoch (second/epoch)						
# of training data	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET	
		Haar	Db4	PL	Haar	Db4
1500	39.476 37	17.292 48	18.342 27	31.422 37	12.099 81	11.755 51
1000	26.425 27	11.803 29	12.341 89	20.916 42	8.276 48	8.214 92
500	13.006 67	6.000 89	6.063 89	10.544 22	4.102 79	4.035 06
100	2.465 22	1.040 74	1.124 65	1.953 24	0.752 94	0.811 97

images. Namely, our reduced method provides very similar performance to the standard unreduced method ( $\mathbb{U}^{(0)}$ -NET), while reducing the computation time greatly by reducing the input dimension.

We also test our proposed method with three different framelets and compare performances, as shown in tables 2 and 4 for the quantitative evaluation and tables 1 and 3 for the computational time. Experimental results report that Haar and Db4 Wavelet reduce the computational time more efficiently than PL framelet, but PL framelet exhibits the better performance than Haar and Db4 Wavelet. Compared to Haar and Db4 consisting of 4 filter banks, PL framelet has 9 filter banks (i.e. the number of filter banks equals the size of filters), which can increase the computational time. However, it should be noted that Haar and Db4 are orthonormal bases while PL framelet is a redundant tight frame system. This means that, thanks to the redundancy, it is likely that the error generated by the nonlinear deep learning process can lie in the nontrivial null space of the reconstruction operator, which can make the PL framelet yield better results than the orthonormal basis (Haar and Db4) (Dong *et al* 2017). We would like to mention that the computational time increases in the case of  $\mathbb{U}^{(2)}$ -NET with PL framelet in the undersampled MRI problem, compared to the original network  $\mathbb{U}^{(0)}$ -NET. We can observe that the reduction of computational time depends on the feature depth of network. In order to reduce total computational complexities of experiments as possible, our networks are set to have 16 feature depth, as described in figure 5. However, when the feature depth increases,  $\mathbb{U}^{(2)}$ -NET with PL framelet also exhibits the computational time reduction ability, as shown in the table 5.

Let us compare a memory requirement on a training process of each model, based on the estimation of memory occupation in the paper (Schlemper *et al* 2018). The  $\mathbb{U}^{(0)}$ -net has 341 152 trainable parameters, which

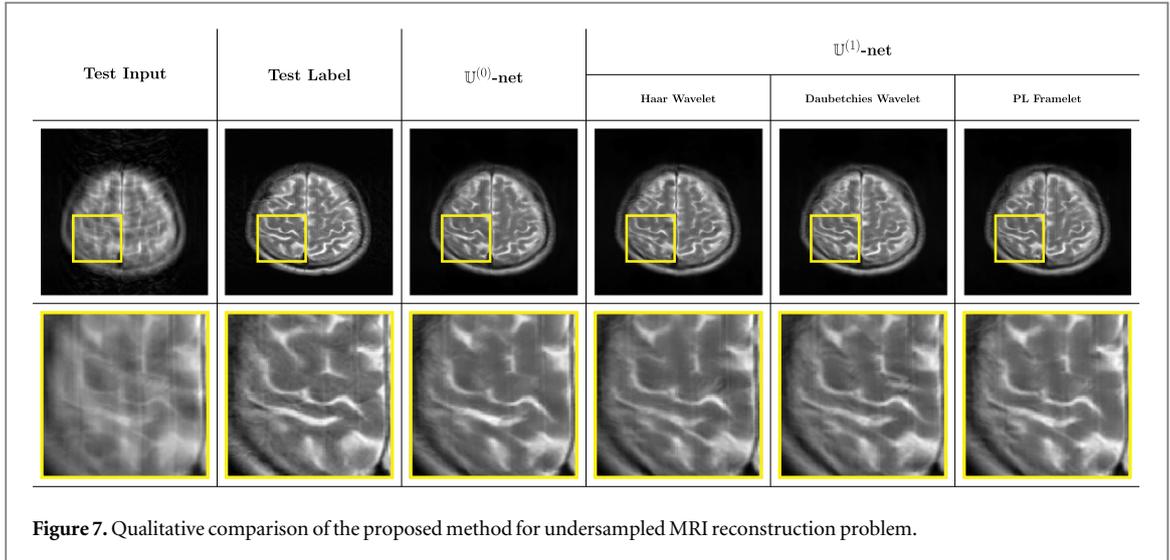


Figure 7. Qualitative comparison of the proposed method for undersampled MRI reconstruction problem.

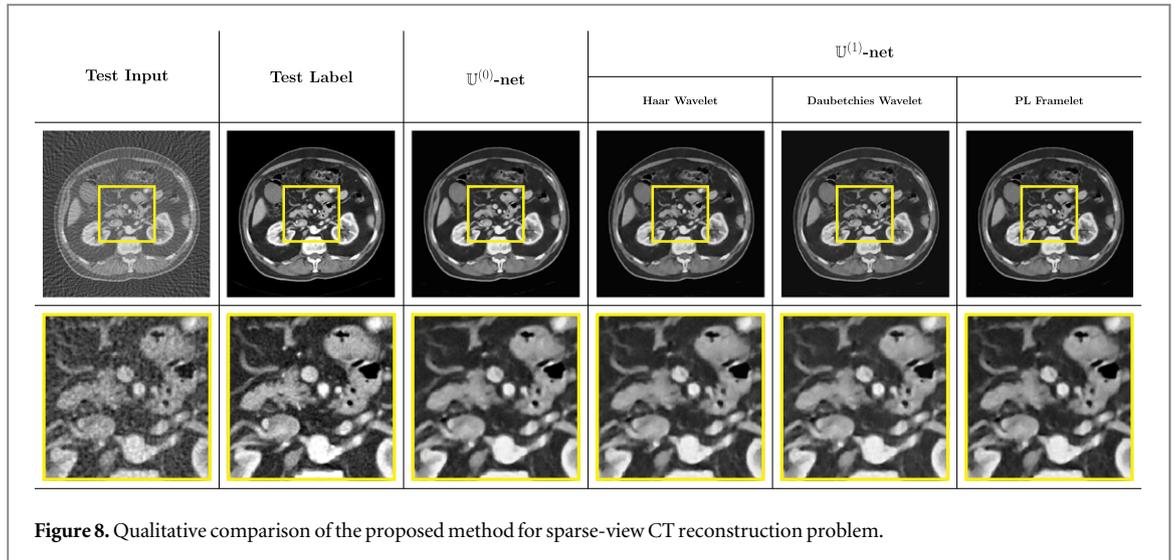
Table 4. Quantitative error evaluations for sparse-view CT problem using two different metrics.

MSE( $\sim 10^{-8}$ )						
# of training data	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET	
		Haar	Db4	PL	Haar	Db4
1500	2.620 30	2.920 02	2.956 05	2.935 51	3.875 06	4.285 39
1000	2.694 98	3.031 15	3.101 20	2.973 95	4.200 75	4.427 56
500	2.803 45	3.213 23	3.326 94	3.093 97	4.420 89	5.402 28
100	3.814 32	4.321 76	4.725 59	4.101 38	5.657 74	7.582 66
SSIM						
# of training data	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET	
		Haar	Db4	PL	Haar	Db4
1500	0.876 31	0.869 37	0.866 60	0.869 59	0.847 95	0.828 65
1000	0.875 33	0.867 61	0.863 57	0.869 25	0.844 44	0.823 83
500	0.874 51	0.864 93	0.857 50	0.867 28	0.839 32	0.809 27
100	0.860 39	0.848 86	0.840 55	0.848 63	0.828 47	0.804 64

can occupy 1.36MB memory in 32-bits single precision system. On the other hands, the  $\mathbb{U}^{(1)}$ -net and  $\mathbb{U}^{(2)}$ -net contain 78 976 and 16 384 unknown variables occupying 0.30 MB and 0.06 MB memory, respectively. Here, the number of filter  $r$  is assumed to be 4. In a training stage, an efficient back-propagation requires storage of all the intermediate activation maps. In our networks, activation maps in each convolution layer should be stored. Thus, a memory requirement can be estimated roughly by

$$\sum_{\text{convolution layer}} N_{\text{batchsize}} \times d_{\text{size of feature}} \times N_{\text{feature depth}} \times 4 \text{ bytes} \times 2, \quad (21)$$

where  $N_{\text{batchsize}}$  is a mini-batch size,  $d_{\text{size of feature}}$  is a pixel dimension of feature, and  $N_{\text{feature depth}}$  is a depth of feature in each convolution layer. Here, we multiply 2, because a forward computation and backward gradient updating exist in a training procedure. In our experiments, we used a mini-batch size 8 and set feature depths to be multiples of 16. In the case of CT image with 512 by 512 pixel resolution, the  $\mathbb{U}^{(0)}$ -net,  $\mathbb{U}^{(1)}$ -net, and  $\mathbb{U}^{(2)}$ -net require approximately 1.34 GB, 0.40 GB, and 0.10 GB memory, respectively. Thus, the proposed method can reduce significantly the total occupied memory in a training stage of deep neural networks.



**Figure 8.** Qualitative comparison of the proposed method for sparse-view CT reconstruction problem.

**Table 5.** Table of the average computational time per epoch in undersampled MRI problem, when using the proposed method with PL framelet and 1000 training data ( $N = 1000$ ).

Average computational time per epoch (second/epoch)			
Feature depth	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET	$\mathbb{U}^{(2)}$ -NET
16	7.650 65	5.616 59	8.312 133
32	12.780 46	6.186 188	8.634 091
64	26.208 05	8.805 362	8.979 816

**Table 6.** Performance evaluations (average computational time per epoch (second/epoch) and mean squared error (MSE)) of the proposed method for undersampled MRI and sparse-view CT problem, when the original deep U-net structure is used for  $\mathbb{U}^{(0)}$ -net and each network is trained from 1500 training data ( $N = 1500$ ).

Performance evaluations on undersampled MRI problem							
Performance	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET		
		Haar	Db4	PL	Haar	Db4	PL
Computation time	52.705 35	15.297 09	15.568 55	20.259 11	7.758 19	7.621 06	22.347 41
MSE ( $\sim 10^{-8}$ )	3.742 12	4.208 49	4.192 82	4.090 67	4.812 88	4.645 87	4.108 83
Performance evaluations on sparse-view CT problem							
Performance	$\mathbb{U}^{(0)}$ -NET	$\mathbb{U}^{(1)}$ -NET			$\mathbb{U}^{(2)}$ -NET		
		Haar	Db4	PL	Haar	Db4	
Computation Time	184.496 52	53.632 90	53.265 28	70.731 39	20.255 99	21.359 46	
MSE ( $\sim 10^{-8}$ )	0.261 50	0.270 65	0.272 49	0.269 25	0.294 37	0.303 45	

Lastly, we also test our proposed method in the case when the original deep U-net structure, described in the paper (Ronneberger *et al* 2015), is used for  $\mathbb{U}^{(0)}$ -net. As shown in table 6, we can observe that the proposed method also shows a powerful ability on reducing computational burdens and provides competitive reconstruction results.

## 4. Conclusion and discussion

In this paper, we proposed the framelet pooling aided deep learning network to reduce computational burdens in the training process. The proposed method decomposes large-scale learning tasks into several small-scale learning tasks through the framelet packet transformation so that we can handle large-scale medical imaging in a limited computing environment. Experimental results on undersampled MRI and sparse-view CT reconstruction problems show that our framelet pooling method is at least comparable to the standard deep learning based method, but is able to reduce total computational time in the training process significantly. Hence, we expect that our method is not limited to the 2-dimensional medical imaging problem. It seems possible that the framelet pooling method can be extended to deep learning problems with large-scale 3-dimensional medical imaging, which inevitably suffers from high computational complexity due to the high dimensionality of dataset.

Deep convolutional neural network with a large size of receptive field seems to be advantageous in capturing global features and properties, compared to patch-wise learning approaches. For instance, the local patch-based approach can solve denoising problems efficiently, since noise patterns are quite similar in each local patch. However, this approach may not deal with aliasing artifacts, which occur in the undersampled MRI problem with uniform subsampling described in the section 2.1 and figure 1. It is because aliasing patterns in figure 1 are generated by the global image domain. To deal with global aliasing artifacts, U-nets without framelet pooling possess high capacity, which requires huge computation power and a large number of data for training the network. Our method is designed to enlarge the receptive field size without increasing a total capacity by using the concept of framelet pooling.

In the experiments, we can see that the choice of filter banks indeed affects the performance of the proposed method. The use of tight frame can increase the reconstruction accuracy thanks to rich representation under the redundant system, but the computational time reduction ability can be marginal due to the increasing number of convolutions. In contrast, the orthogonal wavelet representation provides high computational time reduction by only using 4 filters, but generates less accurate results. Hence, the future work will focus on the construction of framelet transformation which is both adaptive to a given task (Cai *et al* 2014) and computationally efficient. It would also be interesting to provide a theoretical analysis on the approximation property of our deep learning network.

Compared to the recent paper (Ye *et al* 2018) on framelet-based deep learning, we focus on dimensionality reduction of data, where analytic framelet filter banks allow us to drop data dimensionality without loss of data information and to reduce a total capacity of deep neural network simultaneously. The proposed method is expected to be advantageous in dealing with a computational burden and data insufficient problem. In this paper, we only verify the advantage in terms of computational cost by significantly reducing the total number of parameters in deep neural networks.

The proposed method seems to have potential to be applied in other medical image analysis problems suffering from high dimensionality of data, which include super-resolution problems, medical image segmentation tasks, and anomaly detection. Our future research will cover applications of the proposed method into other medical image problems and tasks.

## Acknowledgments

Hyun, Kim, Cho and Seo are supported by Samsung Science & Technology Foundation SSTF-BA1402-01, Hyun is supported by the National Research Foundation of Korea grants NRF-2018H1A2A1062505, Kim is supported by NRF grant NRF-2017R1E1A1A03070653, and Choi is supported in part by the NSFC Youth Program 11901436.

## Data availability statement

Any data that support the findings of this study are included within the article.

## ORCID iDs

Chang Min Hyun  <https://orcid.org/0000-0002-7072-7489>

Jae Kyu Choi  <https://orcid.org/0000-0002-9546-753X>

## References

- Abadi M et al 2016 TensorFlow: large-scale machine learning on heterogeneous systems arXiv:1603.04467
- Barron A R 1994 Approximation and estimation bounds for artificial neural networks *Mach. Learn.* **14** 115–33
- Bruna J and Mallat S G 2013 Invariant scattering convolution networks *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1872–86
- Cai J F, Ji H, Shen Z and Ye G B 2014 Data-driven tight frame construction and image denoising *Appl. Comput. Harmon. Anal.* **37** 89–105
- Daubechies I 1988 Orthonormal bases of compactly supported wavelets *Commun. Pure Appl. Math.* **41** 909–96
- Dong B and Shen Z 2012 MAR-based wavelet frames and applications: image segmentation and surface reconstruction *Proc. SPIE* **8401** 13–28
- Dong B, Shen Z and Xie P 2017 Image restoration: A general wavelet frame based model and its asymptotic analysis *SIAM J. Math. Anal.* **49** 421–45
- Hyun C M, Kim H P, Lee S M, Lee S and Seo J K 2018 Deep learning for undersampled MRI reconstruction *Phys. Med. Biol.* **63** 135007
- Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift arXiv:1502.03167
- Jin K H, McCann M T, Froustey E and Unser M 2017 Deep convolutional neural network for inverse problems in imaging *IEEE Trans. Image Process.* **26** 4509–22
- Kingma D and Ba J 2014 Adam: a method for stochastic optimization arXiv:1412.6980
- Loizou C P, Murray V, Pattichis M S, Seimenis I, Pantziaris M and Pattichis C S 2011 Multi-scale amplitude modulation-frequency modulation (AM-FM) texture analysis of multiple sclerosis in brain MRI images, *IEEE Trans. Inf. Technol. Biomed.* **15** 119–29
- Mallat S 2009 *A Wavelet Tour of Signal Processing* 3rd edn (Amsterdam: Elsevier) (<https://doi.org/10.1016/B978-0-12-374370-1.X0001-8>)
- Mallat S G 2016 Understanding deep convolutional networks *Philos. Trans. R. Soc. A* **374** 20150203
- Mhaskar H N and Poggio T 2016 Deep versus Shallow networks: an approximation theory perspective *Anal. Appl.* **14** 829–48
- Nishimura D G 2010 *Principles of Magnetic Resonance Imaging* (Stanford, CA: Stanford University)
- Pascanu R, Montufar G and Bengio Y 2013 On the number of response regions of deep feedforward networks with piecewise linear activations arXiv:1312.6098
- Ron A and Shen Z 1997 Affine System in  $L^2(\mathbb{R}^d)$ : the analysis of the analysis operator *J. Fourier Anal. Appl.* **148** 408–47
- Ronneberger O, Fischer P and Brox T 2015 U-Net: convolutional networks for biomedical image segmentation *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Lecture Notes in Computer Science* **9351**
- Schlemper J, Caballero J, Hajnal J V, Price A N and Rueckert D 2018 A deep cascade of convolutional neural networks for dynamic MR image reconstruction *IEEE Trans. Med. Imaging* **37** 491–503
- Seo J K and Woo E J 2013 *Nonlinear Inverse Problems in Imaging* (New York: Wiley) (<https://doi.org/10.1002/9781118478141>)
- Shen Z and Xu Z 2013 B-spline framelets derived from the unitary extension principle *Soc. Ind. Appl. Math.* **45** 127–51
- Wang Z, Bovik A C, Sheikh H R and Simoncelli E P 2004 Image quality assessment: from error visibility to structural similarity *IEEE Trans. Image Process.* **13** 600–12
- Ye J C, Han Y and Cha E 2018 Deep convolutional framelets: a general deep learning framework for inverse problems *SIAM J. Imaging Sci.* **11** 991–1048