Super-resolution for Natural Images and Magnetic Resonance Images

by

Pak Lun Kevin Ding

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved November 2020 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Teresa Wu
Yalin Wang
Yezhou Yang

ARIZONA STATE UNIVERSITY

December 2020

ABSTRACT

Image super-resolution (SR) is a low-level image processing task, which has many applications such as medical imaging, satellite image processing, and video enhancement, etc. Given a low resolution image, it aims to reconstruct a high resolution image. The problem is ill-posed since there can be more than one high resolution image corresponding to the same low-resolution image. To address this problem, a number of machine learning-based approaches have been proposed.

In this dissertation, I present my works on single image super-resolution (SISR) and accelerated magnetic resonance imaging (MRI) (a.k.a. super-resolution on MR images), followed by the investigation on transfer learning for accelerated MRI reconstruction. For the SISR, a dictionary-based approach and two reconstruction-based approaches are presented. To be precise, a convex dictionary learning (CDL) algorithm is proposed by constraining the dictionary atoms to be formed by non-negative linear combination of the training data, which is a natural, desired property. Also, two reconstruction-based single methods are presented, which make use of (i)the joint regularization, where a group-residual-based regularization (GRR) and a ridge-regression-based regularization (3R) are combined; (ii)the collaborative representation and non-local self-similarity. After that, two deep learning approaches are proposed, aiming at reconstructing high-quality images from accelerated MRI acquisition. Residual Dense Block (RDB) and feedback connection are introduced in the proposed models. In the last chapter, the feasibility of transfer learning for accelerated MRI reconstruction is discussed.

ACKNOWLEDGEMENTS

First, I would like to give thanks to my advisor, Dr. Baoxin Li. I would like to thank for his patience and guidance for these years. He always gives me his opinions and makes me look at things from a different perspective. Without his help, I won't be able to finish my PhD degree.

I would also like to thank Dr. Teresa Wu, Dr. Yalin Wang, Dr. Yezhou Yang for their advice on my dissertation, and for serving as my dissertation committee members.

In this journey, I have cooperated with a lot of people, including Dr. Kan Chang, Dr. Yikang Li, Dr. Parag Chandakkar, Dr. Xu Zhou, Riti Paul, Dr. Yuxiang Li, Dr. Zhiqiang Li, Jiuxu Chen, Sarah Martin. I had a great time while working with them, and I would like to thank them for their feedback. I also appreciate the support from my labmates, Tianshu Yu, Sachin Chhabra, Nupur Thakur, Yaoxin Zhuo, Yuzhen Ding, Vijetha Guttupalli, Steve Hu, Dr. Lin Chen, Jashmi Lgisetty, Dr. Qiongjie Tian, Dr. Ragav Venkatesan, Dr. Yilin Wang, Dr. Jun Cao.

Lastly, I would like to thank my fiance Betrica Fok, my family and my friends, Haiqin Lin, Rongyu Lin, Yiqun Dai, Enlai Zhang, Keyo Leung, Chase Cai, Oscar Jin, Chris Luo, Kelsey Wang, Dr. Bo Peng, Xiaorong Zhang, Zheng Zhu, Hazel Yang, Dunchuan Wu, Jackie Leung, Wenwan Zhang, Ke Liu, Daniel Yang, Barbara Luk and Hao Li. Their love and support encourage me to complete my PhD degree.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Image super-resolution (SR) is an inverse problem. It aims to reconstruct a high quality image while a low quality image is given. Solving such kind of problems is meaningful to many real world applications such as medical imaging, or satellite image processing, etc.

However, these problems are difficult due to their ill-posedness, since there can be more than one high resolution image corresponding to the same low-resolution image. In this dissertation, I present the proposed machine learning models to solve the super-resolution problem in natural image domain, and accelerated MRI reconstruction. I also analyze the flexibility of transferring knowledge learned from natural images to reconstruct high quality MR images.

1.1 Super-resolution for Natural Images

In recent years, dictionary learning approaches have been used in image super-resolution, achieving promising results. Such approaches train a dictionary from image patches and reconstruct a new patch by sparse combination of the atoms of the dictionary. Typical training methods do not constrain the dictionary atoms. In this dissertation, I propose a convex dictionary learning (CDL) (Ding *et al.*, 2017) algorithm by constraining the dictionary atoms to be formed by non-negative linear combination of the training data, which is a natural, desired property. We evaluate our approach by demonstrating its performance gain over typical approaches.

Also, I propose a reconstruction-based single image super resolution (SR) method by using joint regularization (Chang *et al.*, 2018b), where a group-residual-based reg-

ularization (GRR) and a ridge-regression-based regularization (3R) are combined. In GRR, non-local similar patches are grouped together, and the group weights are calculated so as to adaptively constrain the residual values in the gradient domain. In 3R, we adopt the ridge-regression-based method to establish the projection matrices from an external high-resolution (HR) training set, so that the external HR information can be utilized. To obtain an estimation of the targeted HR image, an efficient algorithm is designed for solving the joint formulation. Experimental results on different image datasets indicate that the proposed method is able to achieve the state-of-the-art performance.

I also proposed an effective single image SR algorithm by using collaborative representation and exploiting non-local self-similarity of natural images (Chang *et al.*, 2018a). In particular, the collaborative-representation-based method is applied to build the so-called self-projection matrices from a training set of HR images. Then the learned self-projection matrices are used to establish the collaborative-representation-based regularization (CRR), which is responsible for introducing the external HR information. Furthermore, to guarantee a reliable estimation of the HR image, the non-local low-rank regularization (NLR) which exploits internal prior of images is also taken into consideration. Since the CRR term and NLR term are complementary, they are assembled together to form a new reconstruction-based framework for SR recovery. Finally, to implement the proposed framework, an iterative algorithm is designed to gradually improve the quality of the SR results. Extensive experimental results indicate that the proposed approach is capable of delivering higher quality of SR results than several state-of-the-art SR methods.

1.2 Accelerated MRI Reconstruction

Typical Magnetic Resonance Imaging (MRI) scan may take 20 to 60 minutes. Reducing MRI scan time is beneficial for both patient experience and cost considerations. Accelerated MRI scan may be achieved by acquiring less amount of k-space data (down-sampling in the k-space). However, this leads to lower resolution and aliasing artifacts for the reconstructed images. There are many existing approaches for attempting to reconstruct high-quality images from down-sampled k-space data, with varying complexity and performance. In recent years, deep-learning approaches have been proposed for this task, and promising results have been reported. Still, the problem remains challenging especially because of the high fidelity requirement in most medical applications employing reconstructed MRI images. In this work, we propose a deep-learning approach (Ding *et al.*, 2019), aiming at reconstructing high-quality images from accelerated MRI acquisition. Specifically, we use Convolutional Neural Network (CNN) to learn the differences between the aliased images and the original images, employing a U-Net-like architecture. Further, a micro-architecture termed Residual Dense Block (RDB) is introduced for learning a better feature representation than the plain U-Net. Considering the peculiarity of the down-sampled k-space data, we introduce a new term to the loss function in learning, which effectively employs the given k-space data during training to provide additional regularization on the update of the network weights. To evaluate the proposed approach, we compare it with other state-of-the-art methods. In both visual inspection and evaluation using standard metrics, the proposed approach is able to deliver improved performance, demonstrating its potential for providing an effective solution.

Inspired by the recent findings on this regard, I note that the feedback connections can improve model performance by refining the low level features with the high level

features. Moreover, since some of the weights in the network are used multiple time, we are allowed to design a more complicated model with same number of parameters. In this paper, I propose a Variational Feedback Network (VFN) for accelerated MRI reconstruction. Specifically, we extend the previous proposed variational network with recurrent neural network (RNN), which is designed to achieve the feedback manner. We analyze the influence of our proposed model by comparing it to other state-of-the-art methods. Quantitative and qualitative evaluations demonstrate that our proposed model performs superiorly against other compared methods on accelerated MRI reconstruction.

1.3 Transfer Learning for Accelerated MRI Reconstruction

As training data are typically scarce in medical applications, deep learning techniques often employ transfer learning, which attempts to incorporate knowledge learned from abundant natural images. Nevertheless, we have yet to see a systematic study on this for MRI reconstruction. In this paper, we investigate the feasibility and potential effects of applying transfer learning in accelerated MRI reconstruction. First, we degrade the natural images in the frequency domain such that they have similar artifacts which are caused by the k-space undersampling scheme. After that, we use these image pairs to pretrain a neural network and analyze if it is possible to transfer the knowledge to improve the performance on accelerated MRI reconstruction. Experiments were designed to evaluate and compare reconstruction performance with and without transfer learning, using the popular UNet architecture as the building block. Our results suggest that transfer learning does not provide significant performance gain in this application. This is further analyzed by visualizing the learned features from different layers of the network, which appears to suggest that knowledge learned from natural images is eventually washed out during

the training using MRI data.

Chapter 2

SINGLE IMAGE SUPER-RESOLUTION

2.1 Convex Dictionary Learning

High-resolution images are useful in many real world applications, such as medical imaging and satellite imaging. When only low-resolution images are available, super-resolution is a common technique deployed for resolution enhancement. The task of single-image super-resolution, which is the focus of this paper, is to reconstruct a high-resolution version from only a low-resolution image. Conventional techniques for solving this problem can be categorized into interpolation-based (such as bilinear interpolation)(Hou and Andrews, 1978), reconstruction-based (Farsiu *et al.*, 2004; Tipping and Bishop, 2003), and example-based methods (Chang *et al.*, 2004; Sun *et al.*, 2008). Interpolation-based schemes tend to generate over-smoothed images, and thus complex details are often missing in the reconstructed high-resolution images. In the reconstruction-based methods, super-resolution is viewed as an inverse problem, which is typically ill-posed, and thus different regularization schemes have been introduced. Most existing efforts on this regard show that such approaches only work for small up-scaling factor. Example-based approaches use images in the training set as priors for reconstruction. Typically they require a huge dataset of high-resolution and low-resolution patch pairs as the examples.

In recent years, dictionary learning-based methods (Yang *et al.*, 2010b, 2012a; Mairal *et al.*, 2008) have been proposed and promising results have been reported. There are also studies (Kulkarni *et al.*, 2012) on relations between such approaches and the theory of compressed sensing (Donoho, 2006). In a dictionary-learning-based ap-

proach, a compact dictionary is learned from some training image patches and future super-resolution would be based on only this compact dictionary. In a representative algorithm of this kind (Yang *et al.*, 2010b), a pair of dictionaries are learned such that both high-resolution and low-resolution image patches can be transformed into the same sparse representation. Other examples of such approaches include (Wang *et al.*, 2012; Zhang *et al.*, 2012a; Jia *et al.*, 2013). Most existing such efforts base the learning on the basic K-SVD algorithm (Aharon *et al.*, 2006) or its variants. K-SVD, as a generalized K-means algorithm, is mainly concerned with minimum-mean-squared-error reconstruction, but less concerned with whether the learned dictionary is physically meaningful (and thus potentially optimal in some sense). For example, while the learned dictionary atoms are supposedly some de facto "basis" image patches, most existing approaches would allow such patches contain negative components and/or allow sparse codes under the dictionary to be negative.

Aiming at learning dictionaries whose atoms are closer to natural image patches as well as support non-negative combinations to form new image patches (both intuitive and desirable properties for a dictionary-based scheme), we propose a new algorithm for training a dictionary for super-resolution. The key idea and main contributions of this paper are thus:

- We introduce a convexity constraint to the dictionary, which forces the atoms of the dictionary to lie within the convex cone of the training patches, and thus formulate a new dictionary learning problem "convex dictionary learning" (CDL).

- We propose an optimization algorithm for solving the CDL problem to obtain a solution, and demonstrate with experiments the effectiveness of the proposed formulation and algorithm.

2.1.1 Super-resolution, Sparse Coding and Dictionary Learning

The problem of super-resolution can be described as follows: let $x_h \in \mathbb{R}^h$ be the vector denoting the original high-resolution image, $x_l \in \mathbb{R}^l$ be the vector denoting the low-resolution version of the original image, $H : \mathbb{R}^h \to \mathbb{R}^h$ and $S : \mathbb{R}^h \to \mathbb{R}^l$ be the blur and decimation operators respectively, then the relation between x_h and x_l can be written as:

$$x_l = SHx_h \tag{2.1}$$

Given x_l, the problem of super-resolution is to find $\hat{x_h} \in \mathbb{R}^h$ such that $\hat{x_h} \approx x_h$. Note that the number of columns is larger than the number of rows for SH, and thus the super-resolution problem is ill-posed.

Let X be the input data, which contains M-dimensional N input signals, *i.e.* $X = (x_1, ..., x_N) \in \mathbb{R}^{M \times N}$. Learning a reconstructive dictionary with K columns, is equivalent to the following optimization problem:

$$\begin{aligned} &< D, Y >= \arg\min_{D,Y} \|X - DY\|_F^2 \\ &s.t. \quad \|y_j\|_0 \leq T \quad \forall j \in \{1, ..., N\} \end{aligned} \tag{2.2}$$

where $D = (d_1, ..., d_n) \in \mathbb{R}^{M \times K}$ is a over-complete dictionary with $K > M$, $Y = (y_1, ...y_N) \in \mathbb{R}^{K \times N}$ is the sparse representation matrix of input data X, $\|y\|_0$ is the number of non-zero entries in y, T is the sparsity constraint constant, and $\|X - DY\|_F^2$ is the reconstruction error.

To construct D, we need to minimize the reconstruction error and fulfill the sparsity constraints in (2.2). The optimization problem in (2.2) can be split into the following two sub problems (2.3) and (2.4):

$$\min_Y \|X - DY\|_F^2 \quad s.t. \quad \forall j \in \{1, ..., N\}, \|y_j\|_0 \leq T \tag{2.3}$$

and

$$D = \arg\min_{D} \|X - DY\|_F^2 \tag{2.4}$$

The main idea of dictionary learning is to find a dictionary such that every input can be approximated as a linear combination of the columns of the dictionary. Several dictionary training methods have been proposed, such as method of optimal directions(MOD) (Engan *et al.*, 1999) and KSVD algorithm (Aharon *et al.*, 2006), the latter is widely used in many real world applications.

For dictionary-based super-resolution, Yang *etal.* (Yang *et al.*, 2012a) proposed a coupled dictionary training model:

$$\begin{aligned} &\min_{D^H, D^L, y_j} \sum_j \|x_j^H - D^H y_j\|_2^2 + \|x_j^L - D^L y_j\|_2^2 \\ &s.t. \quad \|y_j\|_0 \leq T \quad \forall j \in \{1, ..., N\} \end{aligned} \tag{2.5}$$

where x_j^H and x_j^L are the patches extracted from the high-resolution image and its corresponding feature vector in the low-resolution image at the same location respectively. y_j is the sparse representation for the jth pair of low-resolution and high-resolution patches. The core idea of this approach is, the sparse coefficient of the high-resolution patch, after the process described in Eq. (2.1), should remain unchanged.

	baby	bird	butterfly	head	woman	barbara	flowers	lenna	pepper	zebra
Bicubic	33.94	32.48	24.06	32.90	28..56	26.20	27.32	31.73	30.76	26.69
	0.9589	0.9753	0.9070	0.8897	0.9485	0.8109	0.8832	0.9730	0.9726	0.9249
KSVD	34.61	33.54	25.34	33.27	29.52	26.46	28.09	32.47	31.34	27.88
	0.9618	0.9782	0.9271	0.8938	0.9553	0.8191	0.8938	0.9753	0.9744	0.9406
CDL	**34.98**	**33.78**	**25.61**	**33.51**	**29.87**	**26.72**	**28.40**	**32.82**	**31.47**	**28.33**
	0.9649	**0.9786**	**0.9291**	**0.8981**	**0.9583**	**0.8300**	**0.8993**	**0.9768**	**0.9748**	**0.9460**

Table 2.1: The PSNR(dB) and SSIM Comparsion for Bicubic, KSVD, and CDL

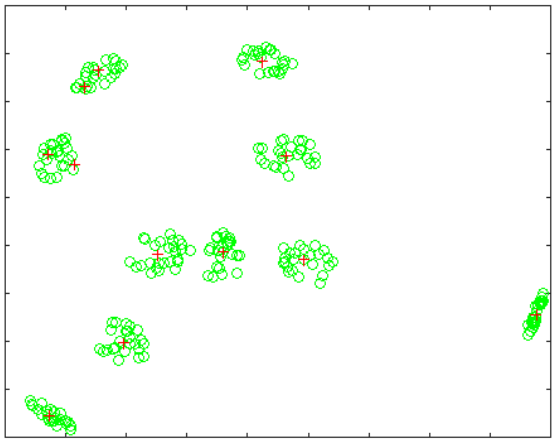


Figure 2.1: A Synthetic Data Sets with Several Clusters. The Red"+" Are the Columns of the Dictionary Constructed by CDL.

2.1.2 Convex Dictionary Learning

We now present the details of our proposed model, which is centered at imposing convexity to dictionary learning.

Problem Formulation

A baseline dictionary-learning approach like Eq. (2.2) does not constrain the columns of the dictionary $D = (d_1, ..., d_K)$. We impose a convex constraint on the dictionary such that the column vectors of D lie in the convex cone formed by the column vectors of the input data matrix X, that is:

$$d_j = Xa_j, \quad \|a_j\|_1 = 1, \quad a_j \geq 0 \quad \forall j \in \{1, ..., K\} \tag{2.6}$$

where $\|\cdot\|_1$ denotes the l_1 norm. We say that d_j is a convex combination of $x_1, ..., x_N$, if there exists $c_i \geq 0, \forall i \in \{1, ..., N\}$ such that $d_j = c_1x_1 + ... + c_Nx_N$ and $\sum_i c_i = 1$. In this case, d_j lies in the convex cone which is formed by some x_i, where c_1, ..., c_N is the barycentric coordinates.

After imposing this constraint, the objective function for dictionary construction

can be defined as:

$$\begin{aligned} &\min_{D,Y} \|X - DY\|_F^2 \quad s.t. \quad \forall j \in \{1, ...N\} \\ &D = XA, A \geq 0, \|a_j\|_1 = 1, \|y_j\|_0 \leq T \end{aligned} \tag{2.7}$$

where $A = (a_1, ..., a_K) \in \mathbb{R}_+^{N \times K}$ is the convex cone matrix. Since we can rewrite D into XA, Eq. (2.7) can be simplified to Eq. (2.8):

$$\begin{aligned} &\min_{A,Y} \|X - XAY\|_F^2 \quad s.t. \quad \forall j \in \{1, ...N\} \\ &A \geq 0, \quad \|a_j\|_1 = 1, \quad \|y_j\|_0 \leq T \end{aligned} \tag{2.8}$$

Adding the convexity constraint to the dictionary gains us some desired properties for the dictionary. Suppose the data are from some clusters, and the optimization is done correctly, the columns of the dictionary should be close to the centroids of the clusters. Getting those centroids is meaningful, since the training data is part of the population, we are trying to recover the original distribution of the data, where the means (centroids) serve as an important role. Another benefit is that, if x_i and x_j are close enough, they will be in the same cluster, which means their sparse code y_i and y_j will be similar. So, CDL preserves the distance among the data in the sparse representation. From this viewpoint, the convex constraint encourages that there are fewer dominating non-zero entries in each sparse representation. Fig. 2.1 illustrates these points for CDL on a synthetic data set.

Optimization

The initialization will be discussed in Sect. 2.1.3. After the initialization, we update Y and A alternatively until they converge. To approximate the optimal solution, we introduce a matrix multiplicative update rule as follows:

Step 1: Update Y (with others fixed): The minimization problem with respect to Y can be decomposed into N sub-problems. The sparse representation Y_j

Figure 2.2: Visual Comparison of Sr Results on "Lenna", "Barbara". The First, Second and Last Column Are from Bicubic, KSVD, CDL Respectively.

can then be computed by using any non-negative pursuit algorithm:

$$\forall j \in \{1, ..., N\}, \quad \min_{y_j} \|x_j - XAy_j\|_2^2 \quad s.t. \quad \|y_j\|_0 \leq T \tag{2.9}$$

Step 2: Update A (with others fixed): The minimization problem with respect to A can be solved by a multiplicative rule, which is stated as follows:

$$B_{ij} \leftarrow A_{ij} \sqrt{\frac{[(X^TX)^+Y^T + (X^TX)^-AYY^T]_{ij}}{[(X^TX)^-Y^T + (X^TX)^+AYY^T]_{ij}}} \tag{2.10}$$

$$A_{ij} \leftarrow \frac{B_{ij}}{\sum_i B_{ij}} \tag{2.11}$$

Here, M_{ij} is the element located in the ith row and the jth column in M, M^+ and M^- are the positive part and negative part of M respectively, *i.e.*

$$M^+ = (|M| + M)/2 \quad M^- = (|M| - M)/2 \tag{2.12}$$

Step 3: Increase the iteration counter: J = J + 1

Algorithm 1: Algorithm for CDL

Input: X,A_0,Y_0,T

while $J = 1 :$ *max iteration number* **do**

update Y using eqt. (2.9);

update A using eqt. (2.10)and (2.11);

update $J = J + 1$;

Calculate $D = XA$ and normalize it.

Output: D

Step 1, 2, 3, should be repeated until convergence. For the update of A, it satisfies the KKT condition and the proof can be found in (Ding *et al.*, 2010), and thus convergence is guaranteed. The procedure is summarized in Algorithm 1.

2.1.3 Experimental Results

We apply our CDL method to single image super-resolution, and compare our algorithm to others using the model described in (Yang *et al.*, 2012a). In our experiments, we use the same training images used in (Yang *et al.*, 2012a). The low-resolution images in the training set are generated by the high-resolution images using down-sampling with bicubic interpolation. Eight popular images for super-resolution are chosen as the testing set. We use 5×5 image patch pairs for a up-scaling factor 3, and the number of columns of the dictionary is 256. Part of the codes are provided by (Yang *et al.*, 2012a) and (Li and Ngom, 2013).

Experimental Setting

We describe how to initialize A and Y for CDL. We can pick a random vector for $Y^{(0)}$, uniformly distributed from 0 to 1, and then set $A^{(0)} = (Y^{(0)})^T P$, where b_j equals to the l_2 norm of the jth row of $Y^{(0)}$ and $P = diag(1/b_1, ..., 1/b_K)$. Another

way is to do a K-means clustering on X. Suppose we obtained the cluster indicator matrix $C = (c_1, ..., c_k)$, where the elements of C is either 0 or 1. Then, by setting $Y^{(0)} = C + \mu E$, where E is a matrix of all 1s, and μ is a small value (*e.g.* $\mu = 0.2$). The centroids of the clusters can be used to be the columns of dictionary D, then $\forall j$, $d_j = Xc_j/b_j$ or $D = XCB$ where $B = diag(1/b_1, ..., 1/b_K)$. Follwed by calculating B, we then have $A = CB$. This method is also used in (Ding *et al.*, 2010). We use the first method in our implementation.

Comparison

We compare our model with bicubic interpolation (Hou and Andrews, 1978), and KSVD (Aharon *et al.*, 2006) under the model described in (Yang *et al.*, 2012a) . In Table 2.1, we compare the peak signal to noise ratio (PSNR) and structural similarity index (SSIM) for the reconstructed high resolution images. The result shows that our approach is consistently better. As illustrated in Fig. 2.2, bicubic interpolation generates over-smoothed results, KSVD's results are a little bit blurry at the complex texture, and some artifacts appear near the edges. Our algorithm gives better results for not just recovering more texture details, but also presenting sharper edges with fewer artifacts.

2.1.4 Further Discussion

KSVD is a dictionary-learning algorithm using the singular value decomposition approach, which makes the dictionary built by KSVD captures as much of the energy of the training data. However, in many real world applications, the testing data may not be identical to the training data, which means KSVD may not be the best approach for reconstructing the testing data. Our approach practically attempts to capture the clusters of the data and thus is less affected by such mismatch between

training and testing data.

For time complexity, CDL uses a matrix multiplicative update rule for training, which makes its training time much longer than KSVD. However, since training is typically done offline, this is not a practical concern.

2.2 Single Image Super Resolution Using Joint Regularization

The task of single image super-resolution (SR) is to restore a high-resolution (HR) image from a given low-resolution (LR) image. Usually, the following degradation model is considered to relate the HR image to the observed LR image

$$\mathbf{Y} = \mathbf{SHX} + \mathbf{n} \tag{2.13}$$

where $\mathbf{Y}$ and $\mathbf{X}$ respectively stand for the LR image and its HR version, $\mathbf{H}$ is a blurring matrix, $\mathbf{S}$ represents the decimation operator, and $\mathbf{n}$ is the additive noise vector.

Among different types of SR methods, the learning-based methods are commonly studied. Its basic idea is to learn the mapping function from the LR space to the HR space. To this end, many methods train the basic models by using an external dataset, and the representative works include the nearest-neighbor-based methods Freeman *et al.* (2002); Chang *et al.* (2004); Gao *et al.* (2012), the sparse-representation-based methods Yang *et al.* (2010b); Zeyde *et al.* (2010); Kulkarni *et al.* (2012), the regression-based methods Kim and Kwon (2010); Yang and Yang (2013); Timofte *et al.* (2013, 2014); Jiang *et al.* (2017), the deep-learning-based methods Dong *et al.* (2016a,b); Wang *et al.* (2015a); Kim *et al.* (2016b), etc. To relax the dependence on the similarity between the training dataset and the test images, some approaches make use of the internal examples Yang *et al.* (2010a); Yang and Wang (2013); Huang and Ahuja (2015), and assume that the patches in a natural image frequently recur at various scales of the same image. As both the external- and the internal-example-driven methods suffer from their respective drawbacks, approaches which jointly take advantage of the external and the internal examples have also been proposed Wang *et al.* (2015b); Cheong and Park (2017).

Apart from the learning-based strategies, another important type of the SR meth-

ods is the reconstruction-based methods, where prior information is applied to regularize the solution spaces. There are different kinds of prior information, including local priors (Rudin *et al.*, 1992; Takeda *et al.*, 2007; Bredies *et al.*, 2010; Hu and Jacob, 2012; Lefkimmiatis *et al.*, 2015), non-local self-similarity (NLSS) priors (Chierchia *et al.*, 2014; Buades *et al.*, 2005; Gilboa and Osher, 2008; Zhang *et al.*, 2010; Maggioni *et al.*, 2013; Dong *et al.*, 2014), sparsity priors (Elad and Aharon, 2006; Dong *et al.*, 2011, 2013), etc. To obtain superior performance, many algorithms assemble complementary priors (Dong *et al.*, 2011, 2013; Zhang *et al.*, 2012b; Ren *et al.*, 2016, 2017; Chen *et al.*, 2017; Liu *et al.*, 2017). For instance, in (Zhang *et al.*, 2012b), steering kernel regression (Takeda *et al.*, 2007) and non-local means (Buades *et al.*, 2005) were combined, while in (Ren *et al.*, 2016, 2017), NLSS and local geometric duality were incorporated. Besides the above priors, some works assume that in some applications, an HR patch of the image is available (El Gheche *et al.*, 2017; Hidane *et al.*, 2016, 2014), and therefore it can be used to guide the reconstruction of the whole HR image.

we propose a reconstruction-based single image SR algorithm, and the main contributions are threefold: First, a group-residual-based regularization (GRR) approach is proposed, which can well characterize the local structure and the NLSS prior of natural images. Second, in order to integrate the advantages of the reconstruction-based and the learning-based SR methods, we present a ridge-regression-based regularization (3R), where the HR features from an external dataset are introduced. Finally, an optimization algorithm named JRSR (joint-regularization-based SR) is designed for solving the minimization problem which incorporates both GRR and 3R. To facilitate evaluation and further exploration, the implementation of JRSR will be available online [1] after this letter is formally published.

[1]http://www.public.asu.edu/%7ebli24/

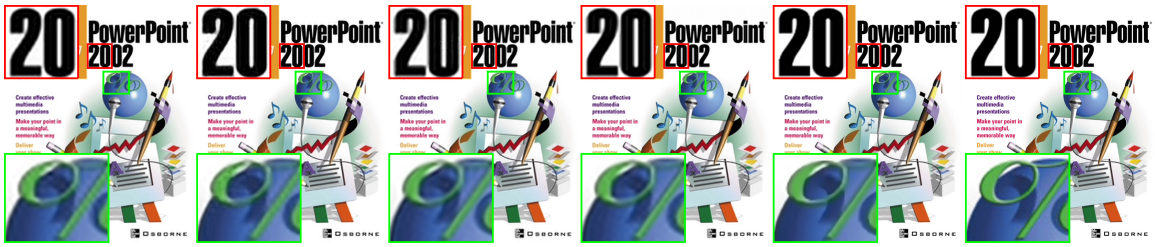


Figure 2.3: SR Results by Different Methods on *ppt3* (×3). From Left to Right: SelfExSR(Huang and Ahuja, 2015) (PSNR: 25.69dB, SSIM: 0.9179), SRCNN(Dong *et al.*, 2016a) (25.94dB, 0.9172), SKR-NLM(Zhang *et al.*, 2012b) (25.18dB, 0.9050), NCSR(Dong *et al.*, 2013) (25.90dB, 0.9246), JRSR (27.58dB, 0.9599), Ground Truth.

2.2.1 The Proposed Joint Regularization

We propose to reconstruct $\mathbf{X}$ from $\mathbf{Y}$ through solving the following minimization problem:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\frac{1}{2}}\|\mathbf{Y} - \mathbf{SHX}\|_2^2 + \alpha\mathbf{\Psi}_{\mathrm{GRR}}(\mathbf{X}) + \beta\mathbf{\Psi}_{\mathrm{3R}}(\mathbf{X}) \tag{2.14}$$

where α and β are trade-off parameters. In problem (2.47), besides the data fidelity term, two additional regularization terms are incorporated, where $\mathbf{\Psi}_{\mathrm{GRR}}(\mathbf{X})$ exploits the structural information in images, and $\mathbf{\Psi}_{\mathrm{3R}}(\mathbf{X})$ introduces the HR information from an external dataset. As the two priors are complementary, jointly considering them can lead to a better result. Now let us discuss them in detail.

2.2.2 Group-Residual-based Regularization (GRR)

As a traditional model, total variation (TV) regularization is good at describing piecewise smooth structures in natural images. It can be formulated by

$$\mathbf{\Psi}_{\mathrm{TV}}(\mathbf{X}) = \|\mathbf{DX}\|_1 \tag{2.15}$$

where $\mathbf{D} = [\mathbf{D}_{\mathrm{h}}^{1T}, \mathbf{D}_{\mathrm{v}}^{1T}]^T$, with $\mathbf{D}_{\mathrm{h}}^1$ and $\mathbf{D}_{\mathrm{v}}^1$ representing the matrices that generate the horizontal and the vertical first-order derivatives, respectively.

However, TV regularization is a global model, which tends to smear out the edges and the details in texture regions. To achieve a better result, one alternative way

is to adaptively regularize the gradient value at each pixel by exploiting non-local similarity within an image. In this letter, we propose the following framework

$$\mathbf{\Psi}_{\mathrm{GRR}}(\mathbf{X}) = \sum_i \|\mathbf{W}_i \circ (\mathbf{F}_i\mathbf{D}\mathbf{X} - \mathbf{E}_i)\|_1 \tag{2.16}$$

where $\mathbf{F}_i$ stands for the matrix which extracts the ith group of similar patches in the gradient domain; $\mathbf{E}_i$ denotes the estimation of the ith group of similar patches in the gradient domain; $(\mathbf{F}_i\mathbf{D}\mathbf{X} - \mathbf{E}_i)$ is the residual of a group of similar patches in the gradient domain. Motivated by our previous work (Chang *et al.*, 2016a), a weight matrix $\mathbf{W}_i$ is introduced to compensate the unreliable estimation for the ith group, so that the compensated result could be more sparse. $\circ$ in (2.16) is the Hadamard product.

To utilize model (2.16), an image is divided into overlapped squared image patches with size $p_\mathrm{s} \times p_\mathrm{s}$. For the ith exemplar patch, we calculate the l_2 distance between it and all the candidate patches in a search window with size $w_\mathrm{s} \times w_\mathrm{s}$. After the $(N_\mathrm{p}-1)$ nearest patches are located, the similar patches are combined as a group, and their positions are recorded so as to build $\mathbf{F}_i$.

Let $\mathbf{x}_{i,m}$ be the mth patch in the ith group in the gradient domain, where $m = 1, 2, ..., N_\mathrm{p}$. Since the patches in the same group are similar to each other, we can obtain an estimation for $\mathbf{x}_{i,m}$ by averaging the other patches in the group

$$\mathbf{e}_{i,m} = \frac{1}{N_\mathrm{p} - 1} \sum_{n \neq m} \mathbf{x}_{i,n} \tag{2.17}$$

Hence the prediction of the whole group can be formed by stacking all $\mathbf{e}_{i,m}$ into a matrix, i.e., $\mathbf{E}_i = [\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, ..., \mathbf{e}_{i,N_\mathrm{p}}]$.

Notice that $\mathbf{e}_{i,m}$ in Eq. (2.17) is actually the expectation of a group of similar patches except $\mathbf{x}_{i,m}$. Therefore, to appropriately establish $\mathbf{W}_i$, we make use of the

variance of these similar patches, which is computed by

$$\mathbf{v}_{i,m} = \left[\frac{1}{N_{\mathrm{p}} - 1} \sum_{n \neq m} (\mathbf{x}_{i,n} - \mathbf{e}_{i,m})^{\circ 2} \right]^{\circ \frac{1}{2}} \tag{2.18}$$

where $\circ^{\frac{1}{2}}$ and $\circ^2$ stand for the Hadamard root and power, respectively. According to the way we perform estimation, a larger variance could lead to a worse estimation. Consequently, our weight matrix is built by $\mathbf{W}_i = [\mathbf{w}_{i,1}, \mathbf{w}_{i,2}, ..., \mathbf{w}_{i,N_{\mathrm{p}}}]$, where $\mathbf{w}_{i,m} = \mathbf{v}_{i,m}^{\circ -1}$, and $\circ^{-1}$ is the Hadamard inverse.

Table 2.2: The PSNR(dB) and SSIM Results on Ten Test Images (×3)

Methods	*Butterfly*	*Parrots*	*Parthenon*	*Bike*	*Flower*	*Girl*	*Hat*	*Leaves*	*Plants*	*Raccoon*	Average
A+	26.89	29.54	26.84	24.30	28.81	33.43	30.99	25.91	33.32	28.92	28.90
	0.9018	0.9077	0.7343	0.7794	0.8377	0.8206	0.8722	0.8931	0.9103	0.7507	0.8408
SelfExSR	26.25	29.67	26.86	24.11	28.99	33.60	30.64	25.84	33.24	29.15	28.83
	0.8786	0.9102	0.7390	0.7774	0.8447	**0.8287**	0.8666	0.8827	0.9108	0.7679	0.8406
SRCNN	27.69	29.77	26.94	24.41	28.89	33.31	30.88	26.47	33.32	29.04	29.07
	0.8999	0.9072	0.7373	0.7812	0.8369	0.8181	0.8655	0.8998	0.9067	0.7570	0.8410
SKR-NLM	26.73	29.74	26.74	24.30	28.94	33.46	30.77	26.14	33.11	29.02	28.89
	0.8888	0.9055	0.7310	0.7843	0.8407	0.8225	0.8647	0.8887	0.9041	0.7595	0.8390
NCSR	28.15	30.26	27.12	24.69	29.35	**33.62**	31.26	27.57	33.99	**29.30**	29.53
	0.9170	0.9117	0.7471	0.8018	0.8529	0.8271	0.8742	0.9227	0.9173	**0.7702**	0.8542
JRSR	**29.74**	**30.51**	**27.34**	**25.09**	**29.84**	33.49	**31.81**	**28.55**	**34.57**	29.19	**30.01**
	0.9386	**0.9171**	**0.7565**	**0.8191**	**0.8664**	0.8233	**0.8849**	**0.9424**	**0.9247**	0.7624	**0.8635**

2.2.3 Ridge-Regression-based Regularization (3R)

Inspired by the success of ANR (Timofte *et al.*, 2013) and A+ (Timofte *et al.*, 2014), we propose a ridge-regression-based regularization (3R), so that the HR information of an external dataset can be effectively utilized. However, different from ANR and A+, we consider the ridge regression problem in the HR space.

Suppose that $\mathbf{N}_j$ is the jth neighborhood in the HR space, which is the most suitable one for representing the feature of the lth HR image patch $\mathbf{f}_l$. The ridge regression problem for finding the corresponding coefficient $\hat{\mathbf{c}}_l$ can be written as

$$\hat{\mathbf{c}}_l = \min_{\mathbf{c}_l} \|\mathbf{f}_l - \mathbf{N}_j\mathbf{c}_l\|_2^2 + \lambda\|\mathbf{c}_l\|_2^2 \tag{2.19}$$

with λ denoting the trade-off parameter. The solution of problem (2.39) is given by

$$\hat{\mathbf{c}}_l = (\mathbf{N}_j^T\mathbf{N}_j + \lambda\mathbf{I})^{-1}\mathbf{N}_j^T\mathbf{f}_l \tag{2.20}$$

By multiplying both sides of the equal sign in Eq. (2.40) with $\mathbf{N}_j$, we have

$$\mathbf{N}_j\hat{\mathbf{c}}_l = \mathbf{P}_j\mathbf{f}_l \tag{2.21}$$

where $\mathbf{P}_j = \mathbf{N}_j(\mathbf{N}_j^T\mathbf{N}_j + \lambda\mathbf{I})^{-1}\mathbf{N}_j^T$ is the projection matrix, which is computed offline and stored for online reconstruction.

With Eqs. (2.39) and (2.21), it is easy to know that $\mathbf{P}_j\mathbf{f}_l$ should be close to $\mathbf{f}_l$. Hence we can use this knowledge to build a new regularization framework, which is formulated by

$$\mathbf{\Psi}_{3\mathrm{R}}(\mathbf{X}) = \sum_l \|\mathbf{R}_l\mathbf{B}\mathbf{X} - \mathbf{P}_j\mathbf{R}_l\mathbf{B}\mathbf{X}\|_1 \tag{2.22}$$

where $\mathbf{R}_l$ denotes the matrix which extracts the feature of the lth image patch, and $\mathbf{B}$ is the feature transformation matrix. Following (Yang *et al.*, 2010b), the first-order and second-order derivatives of the patches are used as the representation. Thus

Algorithm 2: Learning $\mathbf{D}_{\mathrm{H}}$ and $\{\mathbf{P}_j\}$ (Offline Training)

Input: An external HR dataset

Extract features of image patches from the HR dataset;

Learn an HR dictionary $\mathbf{D}_{\mathrm{H}}$ with M atoms by K-SVD (Aharon *et al.*, 2006);

For each atom $\mathbf{a}_j$ in $\mathbf{D}_{\mathrm{H}}$

1. Locate the K nearest neighbours for $\mathbf{a}_j$ in N_{T} training samples and build $\mathbf{N}_j$;
2. Compute $\mathbf{P}_j = \mathbf{N}_j(\mathbf{N}_j^T\mathbf{N}_j + \lambda\mathbf{I})^{-1}\mathbf{N}_j^T$;

End for

Output: $\mathbf{D}_{\mathrm{H}}$, $\{\mathbf{P}_j\}$.

$\mathbf{B} = [\mathbf{D}_{\mathrm{h}}^{1T}, \mathbf{D}_{\mathrm{v}}^{1T}, \mathbf{D}_{\mathrm{h}}^{2T}, \mathbf{D}_{\mathrm{v}}^{2T}]^T$, with $\mathbf{D}_{\mathrm{h}}^2$ and $\mathbf{D}_{\mathrm{v}}^2$ denoting the matrices that produce the horizontal and the vertical second-order derivatives, respectively.

In order to provide enough diversity, the total number of the projection matrices should be large. Furthermore, for each image patch, it is necessary to locate the most suitable matrix $\mathbf{P}_j$. For these two reasons, at the stage of offline training, a dictionary $\mathbf{D}_{\mathrm{H}}$ in the HR space needs to be learned, and then the K nearest neighbours of the jth dictionary atom $\mathbf{a}_j$ are found and grouped into the matrix $\mathbf{N}_j$. The algorithm for learning $\mathbf{D}_{\mathrm{H}}$ and $\{\mathbf{P}_j\}$ is listed as Algorithm 4. When performing online reconstruction, the feature of each image patch is compared with all the atoms in $\mathbf{D}_{\mathrm{H}}$. After the nearest atom $\mathbf{a}_j$ is found, its index j can lead us to the related $\mathbf{P}_j$.

2.2.4 Optimization Algorithm for Single Image SR

To efficiently solve problem (2.47), we define $\mathbf{d}_{1,i} = \mathbf{F}_i\mathbf{D}\mathbf{X} - \mathbf{E}_i$ and $\mathbf{d}_{2,l} = (\mathbf{I} - \mathbf{P}_j)\mathbf{R}_l\mathbf{B}\mathbf{X}$. By applying split-Bregman method (Goldstein and Osher, 2009) and dividing the target problem (2.47) into a number of subproblems, our joint-

Algorithm 3: JRSR (Online Reconstruction)

Input: $\mathbf{S}$, $\mathbf{H}$, $\mathbf{Y}$, $\mathbf{D}_{\mathrm{H}}$, $\{\mathbf{P}_j\}$, α, β, γ, η

Use bicubic interpolation to obtain the initial result $\mathbf{X}^0$;

For $k = 0, 1, ..., N_{\mathrm{s}} - 1$

If $\mathrm{mod}(k, 30) == 0$

Use $\mathbf{X}^k$ to calculate $\{\mathbf{F}_i\}$, $\{\mathbf{W}_i\}$ and $\{\mathbf{E}_i\}$;

End if

Locate the suitable $\mathbf{P}_j$ for each image patch;

Compute $\mathbf{A}$ by (2.25) and $\mathbf{G}$ by (2.26);

Use the CG method to solve (2.24) and get $\mathbf{X}^{k+1}$;

If $k \geq 120$ and $\|\mathbf{X}^{k+1} - \mathbf{X}^k\|_2^2/N < e$

Break;

End if

Compute $\{\mathbf{d}_{1,i}^{k+1}\}$ by (2.27) and $\{\mathbf{d}_{2,l}^{k+1}\}$ by (2.29);

Update $\{\mathbf{b}_{1,i}^{k+1}\}$ and $\{\mathbf{b}_{2,l}^{k+1}\}$ according to (2.23);

End for

Output: $\hat{\mathbf{X}} = \mathbf{X}^{k+1}$.

regularization-driven iteration is formulated as follows

$$
\begin{cases}
\mathbf{X}^{k+1} = {}_{\mathbf{X}}\, \frac{1}{2}\|\mathbf{Y} - \mathbf{SHX}\|_2^2 \\
\qquad\qquad +\frac{\gamma}{2}\sum_i \|\mathbf{d}_{1,i}^k - (\mathbf{F}_i\mathbf{DX} - \mathbf{E}_i) - \mathbf{b}_{1,i}^k\|_2^2 \\
\qquad\qquad +\frac{\eta}{2}\sum_l \|\mathbf{d}_{2,l}^k - (\mathbf{I} - \mathbf{P}_j)\mathbf{R}_l\mathbf{BX} - \mathbf{b}_{2,l}^k\|_2^2 \\
\mathbf{d}_{1,i}^{k+1} = {}_{\mathbf{d}_{1,i}}\, \alpha\|\mathbf{W}_i \circ \mathbf{d}_{1,i}\|_1 \\
\qquad\qquad +\frac{\gamma}{2}\|\mathbf{d}_{1,i} - (\mathbf{F}_i\mathbf{DX}^{k+1} - \mathbf{E}_i) - \mathbf{b}_{1,i}^k\|_2^2 \\
\mathbf{d}_{2,l}^{k+1} = {}_{\mathbf{d}_{2,l}}\, \beta\|\mathbf{d}_{2,l}\|_1 \\
\qquad\qquad +\frac{\eta}{2}\|\mathbf{d}_{2,l} - (\mathbf{I} - \mathbf{P}_j)\mathbf{R}_l\mathbf{BX}^{k+1} - \mathbf{b}_{2,l}^k\|_2^2 \\
\mathbf{b}_{1,i}^{k+1} = \mathbf{b}_{1,i}^k + (\mathbf{F}_i\mathbf{DX}^{k+1} - \mathbf{E}_i) - \mathbf{d}_{1,i}^{k+1} \\
\mathbf{b}_{2,l}^{k+1} = \mathbf{b}_{2,l}^k + (\mathbf{I} - \mathbf{P}_j)\mathbf{R}_l\mathbf{BX}^{k+1} - \mathbf{d}_{2,l}^{k+1}
\end{cases}
\tag{2.23}
$$

where γ and η are two trade-off parameters, k is the iteration number, and the auxiliary variables $\{\mathbf{b}_{1,i}^{k+1}\}$ and $\{\mathbf{b}_{2,l}^{k+1}\}$ are updated according to the Bregman iteration (Goldstein and Osher, 2009).

Since the objective function with respect to $\mathbf{X}$ in Eq. (2.23) is quadratic, by setting its derivative to be zero, the closed form solution of $\mathbf{X}$ can be written as follows

$$\mathbf{X}^{k+1} = \mathbf{G}^{-1}\mathbf{A} \tag{2.24}$$

where

$$\begin{aligned}\mathbf{A} =&(\mathbf{SH})^T\mathbf{Y} + \gamma\mathbf{D}^T\left[\sum_i \mathbf{F}_i^T(\mathbf{E}_i + \mathbf{d}_{1,i}^k - \mathbf{b}_{1,i}^k)\right] \\ &+ \eta\mathbf{B}^T\left[\sum_l \mathbf{R}_l^T(\mathbf{I} - \mathbf{P}_j)^T(\mathbf{d}_{2,l}^k - \mathbf{b}_{2,l}^k)\right]\end{aligned} \tag{2.25}$$

$$\begin{aligned}\mathbf{G} =&(\mathbf{SH})^T(\mathbf{SH}) + \gamma\mathbf{D}^T(\sum_i \mathbf{F}_i^T\mathbf{F}_i)\mathbf{D} \\ &+ \eta\mathbf{B}^T\left[\sum_l \mathbf{R}_l^T(\mathbf{I} - \mathbf{P}_j)^T(\mathbf{I} - \mathbf{P}_j)\mathbf{R}_l\right]\mathbf{B}\end{aligned} \tag{2.26}$$

However, directly calculating Eq. (2.24) requires large computational complexity. Therefore, we use the conjugate gradient (CG) method to solve the linear system problem $\mathbf{G}\mathbf{X}^{k+1} = \mathbf{A}$ and obtain $\mathbf{X}^{k+1}$.

To solve the subproblem on $\mathbf{d}_{1,i}$ in Eq. (2.23), the shrinkage formula is used to get the closed-form solution (Chang *et al.*, 2016a)

$$\mathbf{d}_{1,i}^{k+1} = \text{shrink}(\mathbf{F}_i\mathbf{D}\mathbf{X}^{k+1} - \mathbf{E}_i + \mathbf{b}_{1,i}^k, \alpha\mathbf{W}_i/\gamma) \tag{2.27}$$

With a vector $\mathbf{X}$ and a threshold ϵ, the shrinkage operation is defined by

$$\text{shrink}(\mathbf{X}, \epsilon) = \max(|\mathbf{X}| - \epsilon, 0) \circ \text{sgn}(\mathbf{X}) \tag{2.28}$$

For the subproblem on $\mathbf{d}_{2,l}$ in Eq. (2.23), the solution is

$$\mathbf{d}_{2,l}^{k+1} = \text{shrink}((\mathbf{I} - \mathbf{P}_j)\mathbf{R}_l\mathbf{B}\mathbf{X}^{k+1} + \mathbf{b}_{2,l}^k, \beta/\eta) \tag{2.29}$$

As all the subproblems have been addressed, JRSR is summarized as Algorithm 5, where N_s is the maximum number of iterations, e is a scalar which controls the convergence of JRSR, and N is the number of pixels in an image. In practice, for several images it was observed that the PSNR does not vary much above 120 iterations. Therefore, after 120 iterations, if the relative change of $\mathbf{X}^k$ is too small, JRSR will terminate.

2.2.5 Experimental Results

To evaluate the effectiveness of JRSR, five state-of-the-art SR algorithms are compared, including three learning-based methods (A+ (Timofte *et al.*, 2014), SelfExSR (Huang and Ahuja, 2015), and SRCNN (Dong *et al.*, 2016a)) and two reconstruction-based methods (SKR-NLM (Zhang *et al.*, 2012b) and NCSR (Dong *et al.*, 2013)). For simplicity, different SR methods are only applied on the luminance channel, while bicubic interpolation is used for the chromatic channels. The experiments are carried out under MATLAB 2014b environment and on a PC with Intel(R) Core(TM) i7-4790 CPU and 8G RAM.

In order to generate the LR images, we first blur the original images by a 7×7 Gaussian filter with a standard deviation of 1.6, and then down-sample the blurred images by a factor of 3. For a fair comparison, the models of A+ (Timofte *et al.*, 2014) and SRCNN (Dong *et al.*, 2016a) are re-trained according to the particular blur kernel used in this letter, and the degradation model used in the back-projection step of SelfExSR (Huang and Ahuja, 2015) is also modified.

To learn our $\mathbf{D}_{\mathrm{H}}$ and $\{\mathbf{P}_j\}$, the commonly used 91 training images (Yang *et al.*, 2010b) are selected as the external dataset. The settings of offline learning are: $N_{\mathrm{T}} = 5000000$, $M = 1024$, $K = 2048$, and $\lambda = 1.2$. When performing online reconstruction, we set $p_{\mathrm{s}} = 5$ for both of the GRR and 3R terms. To group similar

patches, the search window size $w_{\mathrm{s}} = 25$, and the group size $N_{\mathrm{p}} = 10$. The other settings of JRSR are: $N_{\mathrm{s}} = 150$, $\alpha = 0.006$, $\beta = 0.00025$, $\gamma = \alpha/20.0$, $\eta = \beta/20.0$, and $e = 0.001$.

The peak signal to noise ratio (PSNR) and the structure similarity (SSIM) results of the luminance channel on ten commonly used test images (Dong *et al.*, 2011) are shown in Table 2.2, where the best methods are highlighted in bold. It can be observed that the proposed JRSR outperforms the other methods in most cases, and the average PSNR and SSIM gains over the second best method (i.e., NCSR (Dong *et al.*, 2013)) are 0.48dB and 0.0093, respectively. To comprehensively test JRSR, we further carry out experiments on other three image datasets, including *Set5* (Bevilacqua *et al.*, 2012), *Set14* (Zeyde *et al.*, 2010) and *BSD100* (Timofte *et al.*, 2014), and the average PSNR and SSIM results are given in Table 2.3.

For visual comparison, the SR results of *ppt3* from *Set14* are presented in Fig.1. We can see that JRSR not only produces sharper edges, but also suppresses the artifacts better than the other methods. Due to length limitation, we report more visual comparisons in the *supplementary material*.

The average running time for super-resolving a 256×256 test image is shown in Table 2.4. The running time of JRSR is comparable to NCSR, but is much longer than the learning-based methods. The computational burden mainly focuses on: 1) computing $\{\mathbf{F}_i\}$ and $\{\mathbf{W}_i\}$ cost $O(p_{\mathrm{s}}^2 w_{\mathrm{s}}^2 N_{\mathrm{e}})$ and $O(p_{\mathrm{s}}^2 N_{\mathrm{p}} N_{\mathrm{e}})$, respectively, where N_{e} denotes the number of exemplar patches in an image; 2) locating the suitable $\mathbf{P}_j$ for all patches takes $O(p_{\mathrm{s}}^2 M N_{\mathrm{e}})$; 3) updating $\mathbf{X}^k$ by the CG method needs $O(N^2 N_{\mathrm{g}})$, where N_{g} is the number of iterations of the CG method. More discussions about the implementation of JRSR can be found in the *supplementary material*.

Among the tested methods, JRSR, A+ (Timofte *et al.*, 2014) and SRCNN (Dong *et al.*, 2016a) need offline training. It takes $O(p_{\mathrm{s}}^2 M N_{\mathrm{T}})$ to train the model of JRSR,

Table 2.3: The Average PSNR(dB) and SSIM Results on *Set5*, *Set14* and *BSD100* (×3)

Methods	*Set5*		*Set14*		*BSD100*	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
A+	32.30	0.9027	28.88	0.8089	28.01	0.7703
SelfExSR	32.33	0.9021	28.89	0.8155	28.06	0.7792
SRCNN	32.37	0.9010	28.97	0.8107	28.07	0.7739
SKR-NLM	32.34	0.9007	28.82	0.8102	28.08	0.7754
NCSR	33.01	0.9100	29.27	0.8197	28.35	0.7837
JRSR	**33.44**	**0.9156**	**29.57**	**0.8239**	**28.56**	**0.7875**

Table 2.4: Average Running Time(s) on 256 × 256 Test Images

A+	SelfExSR	SRCNN	SKR-NLM	NCSR	JRSR
0.19	18.13	1.67	95.58	192.45	197.34

leading to a training time of 5639.1 seconds. Compared with SRCNN, the training phase of which lasts for days, JRSR is much faster and does not require GPU. Since A+ conducts dimensionality reduction for the features, training the model of A+ costs 1328.6 seconds, which is less than JRSR. It should be pointed out that the basic models of both A+ and SRCNN are learned according to a given degradation model. When the model changes (i.e., the magnification factor or the blur kernel changes), it is necessary to re-train their models, which is inconvenient in real applications. On the contrary, the training phase of JRSR is independent of the degradation model. Therefore, after one-time training, JRSR is able to work with different degradation models.

2.3 Single Image Super-resolution Using Collaborative Representation and Non-local Self-similarity

Many multimedia applications involve the task of resolution-enhancement for images or videos as a post-processing technique for overcoming the limitation of the acquisition camera's sensor resolution (Yue *et al.*, 2016). Single image super-resolution (SR) is such a technique that aims to estimate a high-resolution (HR) output from a given low-resolution (LR) image. Generally, the observed LR image is regarded as a degraded version of the inaccessible, true HR image, and the relationship between them can be modeled by the following equation

$$\mathbf{Y} = \mathbf{SHX} + \mathbf{n} \tag{2.30}$$

where $\mathbf{Y}$ and $\mathbf{X}$ are the observed LR image and the corresponding HR image, respectively, $\mathbf{S}$ denotes the down-sampling matrix, $\mathbf{H}$ stands for the blurring operator, and $\mathbf{n}$ represents the additive Gaussian white noise. The existing SR techniques can be roughly divided into three categories: the interpolation-based methods, the learning-based methods and the reconstruction-based methods.

The interpolation-based SR methods predict the unknown pixels in a HR grid. Among them, the linear kernels, such as Bicubic and Lanczos, are widely used due to low complexity. Nevertheless, the linear kernels tend to generate noticeable artifacts such as ringing, aliasing, and blurring in images full of edges and details. To address this problem, some locally adaptive kernels have been proposed, such as (Li and Orchard, 2001; Zhang and Wu, 2006). The locally adaptive kernels are helpful to approximate local structures, but details in fine structure regions still cannot be preserved well.

In the learning-based methods, usually it is assumed that there exists a mapping relationship between the LR space and the HR space. To learn the basic models, many

methods require an external set of HR images, and the corresponding LR images are generated according to the degradation model (2.30). In the nearest-neighbor-based approaches (Freeman *et al.*, 2002; Chang *et al.*, 2004; Gao *et al.*, 2012), the LR and HR image patches are supposed to form low-dimensional nonlinear manifolds with similar local geometry. Through learning an LR and HR dictionary pair, the sparse-representation-based methods (Yang *et al.*, 2010b; Zeyde *et al.*, 2010; Kulkarni *et al.*, 2012; Peleg and Elad, 2014) enforce the corresponding signals in the LR space and the HR space to have the same sparse representations. In (Wang *et al.*, 2017b) and (Wang *et al.*, 2017a), the non-linear mapping relationship from the observed LR space to the latent HR space is learned by Gaussian process regression. To ensure a fast and accurate SR, collaborative representation is applied to compute the mapping matrices in (Timofte *et al.*, 2013, 2014; Zhang *et al.*, 2016b). Owing to the requirement of external data, the performance of the above learning-based methods highly depends on the similarity between the images for training and for testing. To break this dependence, some self-similarity-based SR algorithms have been proposed (Zhu *et al.*, 2014; Yang *et al.*, 2010a; Huang and Ahuja, 2015). In very recent years, deep learning techniques have attracted substantial academic interests in their usage for image SR, such as (Dong *et al.*, 2016a,b; Liu *et al.*, 2016). However, the training time of the deep-learning-based methods is very long.

To make the SR problem well-posed, the reconstruction-based methods adopt the prior knowledge of natural images to regularize the solution spaces. The typical categories include local priors, non-local self-similarity (NLSS) priors and sparsity priors. Among them, the local priors exploit local statistics of images in the spatial domain. As classic models, the total variation (TV) regularization (Rudin *et al.*, 1992) is effective in restoring smooth regions but tend to smear out the edges and details. As an alternative, the piecewise autoregressive (AR) model (Dong *et al.*, 2011) can

provide better results. Due to the fact that natural images usually contain much non-local redundancy, many NLSS priors have been studied. Representative works include the non-local means (NLM)-based regularization (Dong *et al.*, 2011; Zhang *et al.*, 2012b, 2016a, 2018; Chang and Li, 2015; Li *et al.*, 2016), the non-local TV (NLTV)-based regularization (Gilboa and Osher, 2008; Zhang *et al.*, 2010), the non-local low-rank regularization (NLR) (Dong *et al.*, 2014; Chang *et al.*, 2015, 2016a), etc. With the development of sparse representation techniques (Aharon *et al.*, 2006), several sparsity-based regularization models have also been proposed (Dong *et al.*, 2011, 2013; Elad and Aharon, 2006). To achieve the superior performance, many reconstruction-based SR methods assemble different kinds of regularization models. For example, the local geometric duality and the NLSS of images are jointly exploited in (Ren *et al.*, 2017) and (Ren *et al.*, 2016); in (Zhang *et al.*, 2012b), steering kernel regression (Takeda *et al.*, 2007) is combined with the NLM model; AR, NLM and sparsity models are incorporated together in (Dong *et al.*, 2011). Note that some methods utilize the complementary properties of both the learning-based and the reconstruction-based methods. For instance, Zhang *et al.* (Zhang *et al.*, 2015) and Jiang *et al.* (Jiang *et al.*, 2017) take advantage of NLM (Buades *et al.*, 2005) to improve the outcomes of the learning-based methods; in (Zhang *et al.*, 2017), local structural, NLSS and collaborative representation are simultaneously applied to refine the initial estimate obtained by the learning-based method; in (Chen *et al.*, 2017), the gradient of the final reconstructed SR image is required to be similar to that of the result generated by the learning-based method.

It is well known that the performance of the reconstruction-based SR methods highly relies on the reasonability of the incorporated prior information. In order to preserve fine details and suppress annoying artifacts in the resultant images, we propose a novel reconstruction-based single image SR method which takes both the

external and internal priors into consideration. Inspired by the success of the learning-based SR methods, we design a new regularization model to introduce HR information from an external dataset. Furthermore, aiming at increasing the reliability of the reconstruction-based framework, the internal structures of images are also characterized. In summary, the main contributions of this paper are threefold:

1. A new collaborative-representation-based regularization (CRR) model which exploits external prior is developed. Similar to anchored neighborhood regression (ANR) (Timofte *et al.*, 2013) and A+ (Timofte *et al.*, 2014), our model learns the projection matrices from an external set of HR images offline. Nonetheless, the learning process is independent of the degradation model. Therefore, when the degradation model changes, there is no need to re-train our model.

2. A powerful joint regularization term which combines CRR and NLR is presented, where CRR is responsible for introducing the external HR information, and NLR exploits the internal prior (to be more specific, the NLSS property) of natural images. As the two priors are complementary, combing them together could well pose the SR problem.

3. An iterative algorithm is proposed to solve the optimization problem that assembles the global reconstruction constraint and the joint regularization. The proposed algorithm, which is called super-resolution by collaborative representation and non-local self-similarity (CRNS), solves each sub-problem iteratively and finally outputs the best SR results. To facilitate evaluation and further exploration of the proposed approach, the source code will be available on the corresponding author's webpage [2] after this paper is published.

[2]http://www.public.asu.edu/%7ebli24/

The remainder of the paper is organized as follows. The related background is briefly reviewed in Section 2.3.1. The proposed CRR model, the joint regularization model and the CRNS algorithm are detailed in Section 2.3.2. Extensive experimental results and discussions are provided in Section 2.3.3.

2.3.1 Related Background

Collaborative-representation-based image SR

In the collaborative-representation-based methods (Timofte *et al.*, 2013, 2014), the following collaborative representation (also known as ridge regression) problem is considered

$$\hat{\mathbf{w}}_i = \underset{\mathbf{w}_i}{} \|\mathbf{y}_i - \mathbf{N}_j^{\mathrm{L}}\mathbf{w}_i\|_2^2 + \lambda\|\mathbf{w}_i\|_2^2 \tag{2.31}$$

where $\mathbf{y}_i$ is the ith LR image patch, $\mathbf{w}_i$ is the coefficient of $\mathbf{y}_i$ over an LR neighbourhood $\mathbf{N}_j^{\mathrm{L}}$, λ is a trade-off parameter. The closed-form solution of (2.31) is given by

$$\hat{\mathbf{w}}_i = (\mathbf{N}_j^{\mathrm{L}T}\mathbf{N}_j^{\mathrm{L}} + \lambda\mathbf{I})^{-1}\mathbf{N}_j^{\mathrm{L}T}\mathbf{y}_i \tag{2.32}$$

With $\hat{\mathbf{w}}_i$, the HR image patch $\mathbf{x}_i$ is calculated by

$$\mathbf{x}_i = \mathbf{N}_j^{\mathrm{H}}\hat{\mathbf{w}}_i \tag{2.33}$$

where $\mathbf{N}_j^{\mathrm{H}}$ is the related HR neighborhood.

In the training phase, an LR dictionary $\mathbf{D}_{\mathrm{L}} = [\mathbf{d}_1^{\mathrm{L}}, \mathbf{d}_2^{\mathrm{L}}, ..., \mathbf{d}_M^{\mathrm{L}}]$ in the LR space needs to be trained, where M stands for the size of the dictionary, $\mathbf{d}_i^{\mathrm{L}}$ denotes the ith atom. In addition, the neighborhood $\mathbf{N}_j^{\mathrm{L}}$ is built by grouping the K nearest neighbors of the jth dictionary atom $\mathbf{d}_j^{\mathrm{L}}$, which is the closest atom to $\mathbf{y}_i$. In (Timofte *et al.*, 2013), $\mathbf{N}_j^{\mathrm{L}}$ is learned on $\mathbf{D}_{\mathrm{L}}$, while in (Timofte *et al.*, 2014), the whole training material in the LR space is used. To increase the computing speed, the corresponding

projection matrix $\mathbf{P}_j$ is computed as

$$\mathbf{P}_j = \mathbf{N}_j^{\mathrm{H}}(\mathbf{N}_j^{\mathrm{L}^T}\mathbf{N}_j^{\mathrm{L}} + \lambda\mathbf{I})^{-1}\mathbf{N}_j^{\mathrm{L}^T} \tag{2.34}$$

In the testing phase, it is needed to search over $\mathbf{D}_{\mathrm{L}}$ and find the matched atom $\mathbf{d}_j^{\mathrm{L}}$ for each LR image patch, so that the suitable $\mathbf{P}_j$ can be located. Finally, the ith HR image patch is computed by $\mathbf{x}_i = \mathbf{P}_j\mathbf{y}_i$.

Non-local self-similarity (NLSS) -based regularization

In the reconstruction-based methods, the following framework is frequently considered

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{} \mathbf{\Psi}(\mathbf{X}) \qquad s.t. \quad \mathbf{Y} = \mathbf{SHX} \tag{2.35}$$

where $\mathbf{\Psi}(\mathbf{X})$ denotes the regularization term which incorporates the prior information.

Many kinds of regularization exploit NLSS within natural images. For instance, the regularization term based on NLM filter (Buades *et al.*, 2005) is expressed as (Dong *et al.*, 2011; Chang and Li, 2015)

$$\mathbf{\Psi}_{\mathrm{NLM}}(\mathbf{X}) = \sum_i \|x_i - \mathbf{b}_i^T\mathbf{p}_i^{\mathrm{N}}\|_2^2 \tag{2.36}$$

where $\mathbf{p}_i^{\mathrm{N}}$ denotes the non-local pixels of the ith pixel x_i in an image, $\mathbf{b}_i$ contains the corresponding weights.

Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be two image patches whose central pixels are x_i and x_j, respectively. The weight for pixel x_j is calculated by

$$b_{i,j} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/h)/c_i \tag{2.37}$$

where h stands for a scalar, and c_i denotes the normalization factor.

Apart from the NLM-based regularization, NLR is another successful NLSS-based regularization method (Dong *et al.*, 2014). To apply NLR, an image should be divided

into overlapped patches with a size of $\sqrt{P_{\mathrm{s}}} \times \sqrt{P_{\mathrm{s}}}$. For the ith exemplar patch, the N_{p1} nearest similar patches are found in a search window and grouped into a matrix $\mathbf{x}_i^{\mathrm{G}}$. With $\mathbf{x}_i^{\mathrm{G}}$, the NLR term can be expressed as

$$\mathbf{\Psi}_{\mathrm{NLR}}(\mathbf{X}) = \sum_i (\frac{1}{2}\|\mathbf{x}_i^{\mathrm{G}} - \mathbf{L}_i\|_2^2 + \gamma \mathrm{Rank}(\mathbf{L}_i)) \tag{2.38}$$

where γ is the trade-off parameter, and $\mathbf{L}_i$ is the low rank matrix which is close to $\mathbf{x}_i^{\mathrm{G}}$.

2.3.2 The Proposed Method

Collaborative-representation-based regularization (CRR)

In the collaborative-representation-based methods, the projection matrices $\{\mathbf{P}_j\}$, which map LR image patches to the HR space, are trained according to the degradation model (2.30). When the magnification factor or the blur kernel in model (2.30) changes, $\{\mathbf{P}_j\}$ have to be re-learned. This characteristic makes the collaborative-representation-based methods difficult to implement in practice.

To address this problem, we consider the collaborative representation problem in the HR space, i.e.,

$$\hat{\mathbf{w}}_i = \underset{\mathbf{w}_i}{} \|\mathbf{x}_i - \mathbf{N}_j^{\mathrm{H}}\mathbf{w}_i\|_2^2 + \lambda\|\mathbf{w}_i\|_2^2 \tag{2.39}$$

where $\mathbf{x}_i$ denotes the ith HR image patch. The closed-form solution of (2.39) is

$$\hat{\mathbf{w}}_i = (\mathbf{N}_j^{\mathrm{H}T}\mathbf{N}_j^{\mathrm{H}} + \lambda\mathbf{I})^{-1}\mathbf{N}_j^{\mathrm{H}T}\mathbf{x}_i \tag{2.40}$$

It is noteworthy that in (2.31), the neighborhood $\mathbf{N}_j^{\mathrm{L}}$ in the LR space is applied to represent the ith LR image patch $\mathbf{y}_i$, while in (2.39), the HR neighborhood $\mathbf{N}_j^{\mathrm{H}}$ is utilized to represent the ith HR image patch $\mathbf{x}_i$. To establish $\{\mathbf{N}_j^{\mathrm{H}}\}$, an HR dictionary $\mathbf{D}_{\mathrm{H}}$ should be learned beforehand, and the matrix $\mathbf{N}_j^{\mathrm{H}}$ is built by grouping the K

training samples which are the closest to the jth dictionary atom $\mathbf{d}_j^{\mathrm{H}}$ in the HR space.

As $\hat{\mathbf{w}}_i$ is the solution of problem (2.39), the distance between $\mathbf{N}_j^{\mathrm{H}}\hat{\mathbf{w}}_i$ and $\mathbf{x}_i$ should be small. Therefore, this prior knowledge can be used to develop a new regularization term called CRR. Using the way of expression in Section 2.3.1, CRR is defined as

$$\mathbf{\Psi}_{\mathrm{CRR}}(\mathbf{X}) = \sum_i \frac{1}{2}\|\mathbf{x}_i - \mathbf{P}_j\mathbf{x}_i\|_2^2 \tag{2.41}$$

where $\mathbf{P}_j$ is built by

$$\mathbf{P}_j = \mathbf{N}_j^{\mathrm{H}}(\mathbf{N}_j^{\mathrm{H}T}\mathbf{N}_j^{\mathrm{H}} + \lambda\mathbf{I})^{-1}\mathbf{N}_j^{\mathrm{H}T} \tag{2.42}$$

Similar to (2.34), our projection matrices $\{\mathbf{P}_j\}$ can also be learned offline, as detailed in Algorithm 4. It should be pointed out that:

[1)]The training of our $\{\mathbf{P}_j\}$ only involves the images in the HR space, which means it is unrelated to the degradation model. $\{\mathbf{P}_j\}$ in (Timofte *et al.*, 2013) and (Timofte *et al.*, 2014) are used to map image patches from the LR space to the HR space, while our $\{\mathbf{P}_j\}$ connect HR image patches to themselves. To distinguish our $\{\mathbf{P}_j\}$ from the other works, our $\{\mathbf{P}_j\}$ are named as self-projection matrices. In order to find the most suitable $\mathbf{P}_j$ for the ith image patch $\mathbf{x}_i$, we need to search for the dictionary atom to which $\mathbf{x}_i$ is matched. After the matched atom $\mathbf{d}_j^{\mathrm{H}}$ is found, the index j can be used to locate $\mathbf{P}_j$. Therefore, the trained dictionary $\mathbf{D}_{\mathrm{H}}$ should also be stored for the further SR reconstruction.

Nonetheless, empirically it is found that directly minimizing (2.41) does not lead to a satisfactory result. First of all, since $\mathbf{P}_j$ is trained by using the external data, there might exist some patches which cannot be well represented in the absence of suitable examples. Secondly, as $\|\mathbf{x}_i - \mathbf{N}_j\mathbf{w}_i\|_2^2$ is not the only regularization term in (2.39), if we multiply $\mathbf{P}_j$ with $\mathbf{x}_i$, the result is just close, but not equal to $\mathbf{x}_i$.

Algorithm 4: Training of Self-Projection Matrices

Input: HR images, M, K, λ

Extract image patches from an external HR database;

Train a HR dictionary $\mathbf{D}_{\mathrm{H}}$ with size M via K-SVD (Aharon *et al.*, 2006);

For each atom $\mathbf{d}_j$ in $\mathbf{D}_{\mathrm{H}}$

Find the K nearest neighbours of $\mathbf{d}_j$ in the whole HR database and form $\mathbf{N}_j^{\mathrm{H}}$;

Calculate $\mathbf{P}_j$ via (2.42);

End for

return $\{\mathbf{P}_j\}$, $\mathbf{D}_{\mathrm{H}}$.

To improve the accuracy of our CRR model, let us assume that there is a precise estimation of $\mathbf{x}_i$ named $\mathbf{E}_i$. The original model (2.41) is modified as

$$\mathbf{\Psi}_{\mathrm{CRR}}(\mathbf{X}) = \sum_i \frac{1}{2}\|(\mathbf{x}_i - \mathbf{P}_j\mathbf{x}_i) - (\mathbf{E}_i - \mathbf{P}_j\mathbf{E}_i)\|_2^2 \tag{2.43}$$

We define $(\mathbf{x}_i - \mathbf{P}_j\mathbf{x}_i)$ in Eq. (2.43) as the projection error of $\mathbf{x}_i$. If $\mathbf{E}_i$ is very similar to $\mathbf{x}_i$, it is quite natural that the projection error of $\mathbf{E}_i$ should be close to that of $\mathbf{x}_i$. As a result, it is more reasonable to minimize (2.43) than to minimize (2.41).

Generally, there are many potential ways to compute $\{\mathbf{E}_i\}$. However, due to the fact that the original HR images are not available in practice, utilizing the NLSS property of natural images is a good solution. For a given $\mathbf{x}_i$, we find the N_{p2} nearest similar patches in a search window, and then a good estimation can be computed as

$$\mathbf{E}_i = \sum_q \omega_{i,q}\mathbf{x}_{i,q} \tag{2.44}$$

where $\mathbf{x}_{i,q}$ denotes the similar patches of $\mathbf{x}_i$. Inspired by (Buades *et al.*, 2005; Dong *et al.*, 2011), $\{\omega_{i,q}\}$ are set as

$$\omega_{i,q} = \exp(-\|\mathbf{x}_i - \mathbf{x}_{i,q}\|_2^2/h)/c_i \tag{2.45}$$

Note that the weight in (2.37) is pixel-wise, while the weight in (2.45) is patch-wise.

Owing to the fact that $\{\mathbf{x}_{i,q}\}$ can only be extracted from the latest established HR image, to guarantee the accuracy of $\{\mathbf{E}_i\}$, it is necessary to design a reconstruction algorithm which iteratively updates $\{\mathbf{E}_i\}$ and $\mathbf{X}$ multiple times. Such an algorithm will be presented in Section 2.3.2.

Joint regularization by using collaborative representation and non-local self-similarity (NLSS)

Since CRR requires a training set of HR images to establish both $\{\mathbf{P}_j\}$ and $\mathbf{D}_{\mathrm{H}}$, it can be regarded as an external prior. In addition to the external priors, the internal priors can also play an important role in effectively regularizing the solution spaces for the SR problem. Among different kinds of internal regularization, NLR is very effective and has been successfully utilized in previous works such as image demosaicking (Chang *et al.*, 2015) and compressive sensing recovery of images (Chang *et al.*, 2016a,b). For the expression of the NLR term, please refer to Eq. (2.38).

To achieve a more robust and reliable SR result, we propose to combine both the CRR and NLR terms together, leading to the following problem

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\frac{1}{2}}\|\mathbf{Y} - \mathbf{SHX}\|_2^2 + \alpha\mathbf{\Psi}_{\mathrm{CRR}}(\mathbf{X}) + \beta\mathbf{\Psi}_{\mathrm{NLR}}(\mathbf{X}) \tag{2.46}$$

Reconstruction framework (2.46) is an unconstraint problem, where the first term indicates that the resultant image needs to be consistent with the LR input. α and β are two trade-off parameters which are responsible for balancing the data fidelity term, the CRR term and the NLR term. It is worth noting that the values of α and β are very important to the effectiveness of the framework (2.46). We will illustrate how to choose the suitable α and β through experiment in Section 2.3.3.

Solving the optimization problem

For convenience of expression, we let $\mathbf{R}_i\mathbf{X} = \mathbf{x}_i$ and $\mathbf{F}_i\mathbf{X} = \mathbf{x}_i^{\mathrm{G}}$, where $\mathbf{R}_i$ denotes the matrix extracting patch $\mathbf{x}_i$ from $\mathbf{X}$, and $\mathbf{F}_i$ stands for the matrix extracting the similar patches group for the ith exemplar patch from $\mathbf{X}$, respectively. Moreover, it should be noticed that besides $\mathbf{X}$, the low rank matrices $\{\mathbf{L}_i\}$ are also unknown. Thus problem (2.46) can be rewritten as

$$\begin{aligned}\{\hat{\mathbf{X}}, \{\hat{\mathbf{L}}_i\}\} = &\underset{\mathbf{X},\{\mathbf{L}_i\}}{} \frac{1}{2}\|\mathbf{Y} - \mathbf{SHX}\|_2^2 \\ &+ \frac{\alpha}{2}\sum_i \|(\mathbf{R}_i\mathbf{X} - \mathbf{P}_j\mathbf{R}_i\mathbf{X}) - (\mathbf{E}_i - \mathbf{P}_j\mathbf{E}_i)\|_2^2 \\ &+ \beta\sum_i (\frac{1}{2}\|\mathbf{F}_i\mathbf{X} - \mathbf{L}_i\|_2^2 + \gamma \mathrm{Rank}(\mathbf{L}_i))\end{aligned} \tag{2.47}$$

To obtain the solution of (2.47), the original problem is divided into the following two sub-problems

$$\begin{cases} \{\hat{\mathbf{L}}_i\} = \underset{\{\mathbf{L}_i\}}{} \sum_i (\frac{1}{2}\|\mathbf{F}_i\mathbf{X} - \mathbf{L}_i\|_2^2 + \gamma \mathrm{Rank}(\mathbf{L}_i)) \\ \hat{\mathbf{X}} = \underset{\mathbf{X}}{} \|\mathbf{Y} - \mathbf{SHX}\|_2^2 + \beta\sum_i (\|\mathbf{F}_i\mathbf{X} - \mathbf{L}_i\|_2^2) \\ \qquad\qquad +\alpha\sum_i \|(\mathbf{R}_i\mathbf{X} - \mathbf{P}_j\mathbf{R}_i\mathbf{X}) - (\mathbf{E}_i - \mathbf{P}_j\mathbf{E}_i)\|_2^2 \end{cases} \tag{2.48}$$

The "$\{\mathbf{L}_i\}$" sub-problem is a rank minimization problem, which is NP-hard. The usual way to tackle this problem is to replace the matrix rank with the nuclear norm (Candès *et al.*, 2011). However, the non-convex surrogate is found to approximate the rank better than the nuclear norm (Dong *et al.*, 2014). Therefore, the $\log\det(\cdot)$ surrogate function is used here, and the "$\mathrm{Rank}(\mathbf{L}_i)$" in (2.48) is replaced with a newly defined function as below

$$L(\mathbf{L}_i, \varepsilon) = \log\det((\mathbf{L}_i\mathbf{L}_i^T)^{1/2} + \varepsilon\mathbf{I}) \tag{2.49}$$

where $\mathbf{I}$ is the identity matrix, and ε denotes a small constant. By treating each $\mathbf{L}_i$

separately, we resort to

$$\hat{\mathbf{L}}_i = \underset{\mathbf{L}_i}{} \frac{1}{2}\|\mathbf{F}_i\mathbf{X} - \mathbf{L}_i\|_2^2 + \gamma L(\mathbf{L}_i, \varepsilon) \tag{2.50}$$

Based on the proximal operator of weighted nuclear norm in (Dong *et al.*, 2014), the solution of (2.50) can be found through an iterative way, and the result at the kth iteration is calculated by (the interested readers are referred to (Dong *et al.*, 2014) for the proof)

$$\mathbf{L}_i^k = \mathbf{U}(\mathbf{\Sigma} - \gamma \ \mathrm{diag}(\mathbf{W}^{k-1}))_+\mathbf{V}^T \tag{2.51}$$

where the lth element in the vector $\mathbf{W}^{k-1}$ is calculated by $W_l^{k-1} = 1/(\sigma_l^{k-1} + \varepsilon)$, σ_l^{k-1} stands for the lth singular value of $\mathbf{L}_i^{k-1}$, $\mathbf{U\Sigma V}^T$ denotes the singular value decomposition (SVD) of $\mathbf{F}_i\mathbf{X}$, and $(x)_+ = \max(x, 0)$. In practice, executing Eq. (2.51) one time is enough. By setting $\mathbf{F}_i\mathbf{X}$ as the initial guess, the solution of (2.50) is

$$\mathbf{L}_i = \mathbf{U}(\mathbf{\Sigma} - \gamma \ \mathrm{diag}(\mathbf{Z}))_+\mathbf{V}^T \tag{2.52}$$

where $Z_l = 1/(\Sigma_l + \varepsilon)$.

Due to the reason that the objective function in the "$\mathbf{X}$" sub-problem is strictly convex, by setting the derivative of the objective function to be zero, we can find the closed form solution of $\mathbf{X}$ as follows

$$\hat{\mathbf{X}} = \mathbf{A}^{-1}\mathbf{V} \tag{2.53}$$

where

$$\begin{aligned}\mathbf{A} =&(\mathbf{SH})^T(\mathbf{SH}) + \beta\sum_i \mathbf{F}_i^T\mathbf{F}_i \\ &+ \alpha\sum_i \mathbf{R}_i^T(\mathbf{I} - \mathbf{P}_j)^T(\mathbf{I} - \mathbf{P}_j)\mathbf{R}_i\end{aligned} \tag{2.54}$$

$$\begin{aligned}\mathbf{V} =&(\mathbf{SH})^T\mathbf{Y} + \beta\sum_i \mathbf{F}_i^T\mathbf{L}_i \\ &+ \alpha\sum_i \mathbf{R}_i^T(\mathbf{I} - \mathbf{P}_j)^T(\mathbf{I} - \mathbf{P}_j)\mathbf{E}_i\end{aligned} \tag{2.55}$$

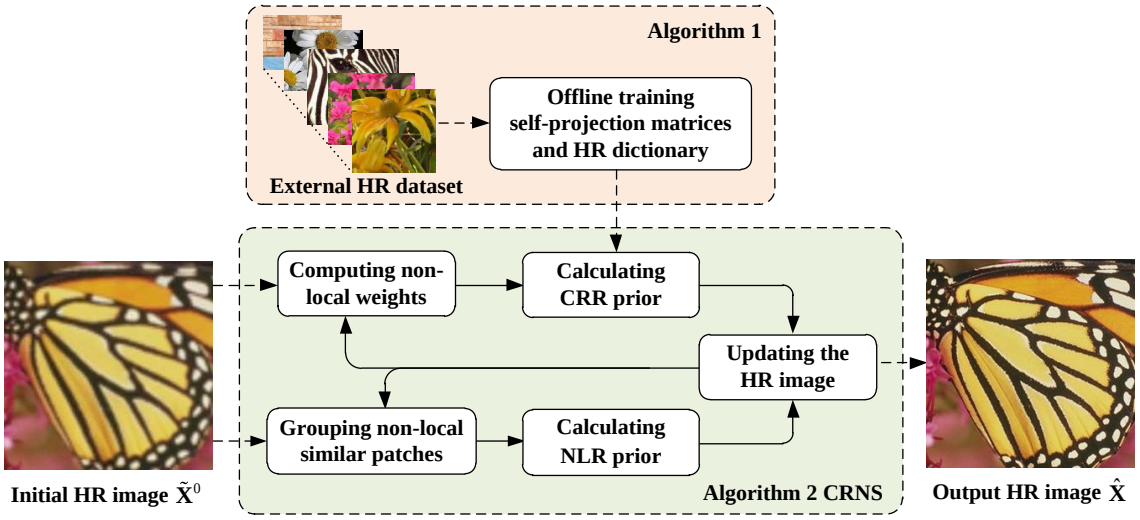


Figure 2.4: The Flowchart of the Proposed Method

However, the complexity of directly calculating $\hat{\mathbf{X}}$ via (2.53) can be high. Therefore, conjugate gradient (CG) method is used to obtain the result. Please see the Appendix for more details.

By iteratively solving the "$\{\mathbf{L}_i\}$" sub-problem and the "$\mathbf{X}$" sub-problem, the results of both $\{\mathbf{L}_i\}$ and $\mathbf{X}$ can be gradually improved until the variables converge.

Summary of the proposed method

The flowchart of our method is given in Fig. 2.4, and a detailed description of the proposed algorithm named CRNS is summarized as Algorithm 5, where $\mathbf{X}^k$ and $\tilde{\mathbf{X}}^t$ represent the results in the inner loop and the outer loop, respectively, K_{m} and T_{m} are the maximum iteration numbers of the inner loop and the outer loop, respectively. At each inner iteration of Algorithm 5, every newly updated HR image patch $\mathbf{x}_i$ is compared with all the atoms in $\mathbf{D}_{\mathrm{H}}$. After the nearest neighbour of $\mathbf{x}_i$ is found, the position j is used to locate the matched $\mathbf{P}_j$.

To guarantee a reliable result, $\{\omega_{i,q}\}$, $\{\mathbf{E}_i\}$ and $\{\mathbf{F}_i\}$ are updated multiple times in Algorithm 5. Note that although the searching of similar patches is required for

Algorithm 5: CRNS

Input: $\mathbf{S}$, $\mathbf{H}$, $\mathbf{Y}$, $\{\mathbf{P}_j\}$, $\mathbf{D}_{\mathrm{H}}$, α, β, γ

Using Bicubic interpolation to get the initial guess $\tilde{\mathbf{X}}^0$;

Outer loop for $t = 0, 1, ..., T_{\mathrm{m}} - 1$

Use $\tilde{\mathbf{X}}^t$ to compute $\{\omega_{i,q}\}$ according to (2.45);

Use $\tilde{\mathbf{X}}^t$ to update $\{\mathbf{F}_i\}$;

Set $\mathbf{X}^0 = \tilde{\mathbf{X}}^t$;

Inner loop for $k = 0, 1, ..., K_{\mathrm{m}} - 1$

Use $\mathbf{X}^k$ and $\{\omega_{i,q}\}$ to calculate $\{\mathbf{E}_i\}$;

For each image patch in $\mathbf{X}^k$, find the matched $\mathbf{P}_j$;

Calculate each $\mathbf{L}_i^{k+1}$ according to (2.52);

Use CG method to solve (2.53) and get $\mathbf{X}^{k+1}$;

End for

Set $\tilde{\mathbf{X}}^{t+1} = \mathbf{X}^{K_{\mathrm{m}}}$;

End for

return $\hat{\mathbf{X}} = \tilde{\mathbf{X}}^{T_{\mathrm{m}}}$.

constructing both $\{\mathbf{F}_i\}$ and $\{\mathbf{E}_i\}$, they should be performed separately. By doing so sufficient flexibility can be provided for CRR and NLM terms, as the two terms are allowed to have different patch sizes, different numbers of overlapping pixels and different numbers of similar patches in a group.

2.3.3 Experimental Results

In this section, we conduct estensive experiments to evaluate the effectiveness of the proposed CRNS algorithm. All the experiments are carried out on an Intel(R) Core (TM) i7-4790 PC under the MATLAB R2014b programming environment.

Ten methods are selected for comparison, including six learning-based methods (Zeyde's method (Zeyde *et al.*, 2010), ANR (Timofte *et al.*, 2013), A+ (Timofte *et al.*, 2014), LANR-NLM (Jiang *et al.*, 2017), SRCNN (Dong *et al.*, 2016a), FSRCNN (Dong *et al.*, 2016b)), and four reconstruction-based methods (ASDS (Dong *et al.*, 2011), ASDS-AR (Dong *et al.*, 2011), ASDS-AR-NL (Dong *et al.*, 2011), NCSR (Dong *et al.*, 2013)).

To evaluate the objective quality of the results, peak signal to noise ratio (PSNR), structure similarity (SSIM) index (Wang *et al.*, 2004) and feature similarity (FSIM) index (Zhang *et al.*, 2011) are applied. Due to the fact that human visual system is more sensitive to the luminance component, all the tested methods are only used to reconstruct the luminance component, while the Bicubic interpolation is applied for the chromatic components. Therefore, all the PSNR/SSIM/FSIM values only assess the quality of the luminance channel.

Experimental settings

Ten commonly used images shown in Fig. 2.5 (Dong *et al.*, 2011, 2013) are selected for testing. To generate the degraded LR images, all the test images are first blurred by a Gaussian filter with a size of 7×7 and a standard deviation of 1.6, and then down-sampled by a factor of 3. Note that Zeyde (Zeyde *et al.*, 2010), ANR (Timofte *et al.*, 2013), A+ (Timofte *et al.*, 2014), SRCNN (Dong *et al.*, 2016a) and FSRCNN (Dong *et al.*, 2016b) all use the Bicubic kernel as the blur kernel in their papers. Thus for a fair comparison, their models are re-trained with the particular blur kernels used in our experiments.

In our experiments, the 91 training images proposed in (Yang *et al.*, 2010b) are used to train the self-projection matrices $\{\mathbf{P}_j\}$ for the proposed CRR term. The same training set is also applied for the six benchmark learning-based methods. The

Figure 2.5: The Test Images. From Left to Right and Top to Bottom: *Butterfly*, *Parrots*, *Parthenon*, *Bike*, *Flower*, *Girl*, *Hat*, *Leaves*, *Plants*, *Raccoon*.

number of training samples, the dictionary size M and the neighborhood size K in Algorithm 4 are respectively set as 5000000, 1024 and 2048, which are the same as the standard settings of A+ (Timofte *et al.*, 2014). The trade-off parameter λ is equal to 0.9.

In Algorithm 5, the empirical settings of the trade-off parameters are: $\alpha = 0.0015$, $\beta = 0.005$ and $\gamma = 8.75$. The numbers of the inner iterations K_{m} and the outer iterations T_{m} are selected as 20 and 10, respectively. For CRR, we use 5×5 patches with an overlap of 3 pixels, and N_{p2} is 10. On the other hand, 5×5 patches with an overlap of 2 pixels, and $N_{\mathrm{p1}} = 12$ are chosen for NLR. h in Eq. (2.45) is 75, and the sizes of the search windows for CRR and NLR are both set as 25.

Table 2.5: The PSNR (dB), SSIM, and FSIM Results by Different Methods (×3)

Image	Zeyde	ANR	A+	LANR-NLM	SRCNN	FSRCNN	ASDS	ASDS-AR	ASDS-AR-NL	NCSR	CRNS
	25.21	25.37	26.89	26.01	27.69	27.81	26.70	26.60	27.33	28.15	**29.20**
Butterfly	0.8562	0.8561	0.9018	0.8713	0.8999	0.9108	0.8928	0.8898	0.9049	0.9170	**0.9322**
	0.8266	0.8253	0.8783	0.8367	0.8903	0.9018	0.8594	0.8553	0.8795	0.9029	**0.9308**
	28.56	28.90	29.54	29.46	29.77	30.09	29.85	29.89	30.19	30.26	**30.64**
Parrots	0.8894	0.8947	0.9077	0.9048	0.9072	0.9072	0.9084	0.9082	0.9110	0.9117	**0.9165**
	0.9239	0.9261	0.9372	0.9356	0.9395	0.9395	0.9379	0.9374	0.9399	0.9431	**0.9460**
	26.34	26.36	26.84	26.66	26.94	27.13	26.79	26.79	26.95	27.12	**27.22**
Parthenon	0.7089	0.7133	0.7343	0.7304	0.7373	0.7454	0.7351	0.7347	0.7397	0.7471	**0.7500**
	0.8010	0.8034	0.8184	0.8225	0.8271	0.8302	0.8210	0.8208	0.8252	**0.8365**	0.8343
	23.28	23.51	24.30	24.08	24.41	24.68	24.34	24.29	24.61	24.69	**25.00**
Bike	0.7266	0.7425	0.7794	0.7733	0.7812	0.7952	0.7866	0.7844	0.7970	0.8018	**0.8141**
	0.8150	0.8218	0.8461	0.8427	0.8524	0.8594	0.8474	0.8459	0.8548	0.8592	**0.8653**
	27.89	28.19	28.81	28.66	28.89	29.27	28.92	28.92	29.19	29.35	**29.64**
Flower	0.8021	0.8142	0.8377	0.8337	0.8369	0.8506	0.8394	0.8395	0.8472	0.8529	**0.8610**
	0.8623	0.8698	0.8827	0.8830	0.8844	0.8924	0.8867	0.8873	0.8939	0.8976	**0.9035**
	32.88	33.12	33.43	33.48	33.31	33.49	33.42	33.43	33.49	**33.62**	33.60
Girl	0.8028	0.8111	0.8206	0.8263	0.8181	0.8239	0.8190	0.8191	0.8200	**0.8271**	0.8255
	0.8958	0.9029	0.9096	0.9184	0.9085	0.9120	0.9129	0.9129	0.9129	**0.9211**	0.9173

Hat	29.96	29.98	30.99	30.58	30.88	31.16	30.85	30.78	30.98	31.26	**31.66**
	0.8494	0.8506	0.8722	0.8646	0.8655	0.8727	0.8669	0.8657	0.8697	0.8742	**0.8814**
	0.8772	0.8767	0.8982	0.8956	0.8958	0.9031	0.8983	0.8971	0.9012	0.9084	**0.9128**
Leaves	24.55	24.73	25.91	25.21	26.47	26.58	26.23	26.12	26.84	27.57	**28.41**
	0.8461	0.8489	0.8931	0.8679	0.8998	0.9048	0.8952	0.8917	0.9085	0.9227	**0.9360**
	0.8343	0.8296	0.8848	0.8474	0.8932	0.8962	0.8726	0.8672	0.8869	0.9083	**0.9269**
Plants	31.93	32.20	33.32	32.91	33.32	33.77	33.19	33.14	33.51	33.99	**34.57**
	0.8815	0.8897	0.9103	0.9051	0.9067	0.9154	0.9037	0.9030	0.9083	0.9173	**0.9241**
	0.9039	0.9075	0.9257	0.9221	0.9259	0.9332	0.9210	0.9204	0.9256	0.9360	**0.9405**
Raccoon	28.43	28.62	28.92	28.92	29.04	29.10	29.17	29.17	29.28	29.30	**29.39**
	0.7240	0.7364	0.7507	0.7640	0.7570	0.7638	0.7652	0.7651	0.7668	**0.7702**	0.7688
	0.8415	0.8503	0.8576	0.8746	0.8647	0.8665	0.8757	0.8757	0.8759	**0.8795**	0.8745
Average	27.90	28.10	28.90	28.60	29.07	29.31	28.95	28.91	29.24	29.53	**29.93**
	0.8087	0.8158	0.8408	0.8341	0.8410	0.8496	0.8412	0.8401	0.8473	0.8542	**0.8610**
	0.8581	0.8614	0.8839	0.8779	0.8882	0.8938	0.8833	0.8820	0.8896	0.8993	**0.9052**

Figure 2.6: SR Results of *Butterfly* (×3) Provided by Different Methods. From Left to Right and from Top to Bottom: Zeyde (Zeyde *et al.*, 2010) (PSNR: 25.21 dB), ANR (Timofte *et al.*, 2013) (25.37 dB), A+ (Timofte *et al.*, 2014) (26.89 dB), LANR-NLM (Jiang *et al.*, 2017) (26.01 dB), SRCNN (Dong *et al.*, 2016a) (27.69 dB), FSRCNN (Dong *et al.*, 2016b) (27.81 dB), ASDS-AR-NL (Dong *et al.*, 2011) (27.33 dB), NCSR (Dong *et al.*, 2013) (28.15 dB), CRNS (**29.20 dB**), Original Image.

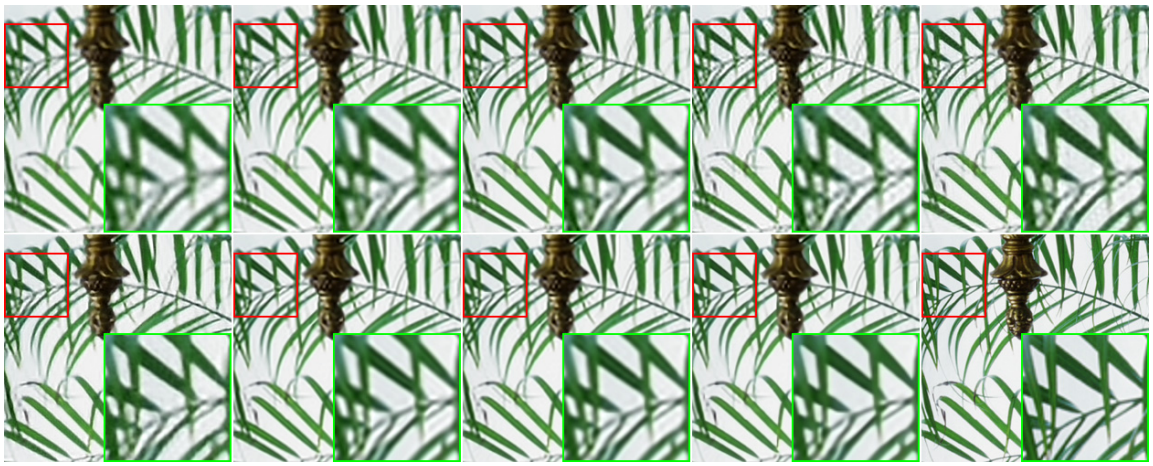


Figure 2.7: SR Results of *Leaves* (×3) Provided by Different Methods. From Left to Right and from Top to Bottom: Zeyde (Zeyde *et al.*, 2010) (PSNR: 24.55 dB), ANR (Timofte *et al.*, 2013) (24.73 dB), A+ (Timofte *et al.*, 2014) (25.91 dB), LANR-NLM (Jiang *et al.*, 2017) (25.21 dB), SRCNN (Dong *et al.*, 2016a) (26.47 dB), FSR-CNN (Dong *et al.*, 2016b) (26.58 dB), ASDS-AR-NL (Dong *et al.*, 2011) (26.84 dB), NCSR (Dong *et al.*, 2013) (27.57 dB), CRNS (**28.41 dB**), Original Image.

SR results (×3) on 10 test images

The PSNR, SSIM and FSIM obtained by different methods are listed in Table 2.5. The reconstructed images *Butterfly* and *Leaves* are given in Figs. 2.6 and 2.7, respectively.

It is obvious that Zeyde (Zeyde *et al.*, 2010) and ANR (Timofte *et al.*, 2013) have the worst performance and tend to generate blurry results. A+ (Timofte *et al.*, 2014), LANR-NLM (Jiang *et al.*, 2017), SRCNN (Dong *et al.*, 2016a) are more competitive than Zeyde (Zeyde *et al.*, 2010) and ANR (Timofte *et al.*, 2013) in terms of both objective and subjective quality. Nevertheless, fine details in the resultant images are still not well recovered. FSRCNN (Dong *et al.*, 2016b), ASDS-AR-NL (Dong *et al.*, 2011) and NCSR (Dong *et al.*, 2013) outperform the other comparison baselines. They are capable of well preserving the edge sharpness and image structures, but are still inferior to our CRNS algorithm. It can be seen in Table 2.5 that the proposed CRNS algorithm achieves the highest objective quality in most of the cases. The average PSNR, SSIM and FSIM gains over the second best method (i.e., NCSR (Dong *et al.*, 2013)) are 0.40dB, 0.0068 and 0.0059, respectively. In Figs. 2.6 and 2.7, both fine structures and sharp edges are well recovered by CRNS, leading to the highest subjective quality.

Effectiveness of different regularization priors

Table 2.6: The PSNR (dB), SSIM, and FSIM Results by Different Priors (×3)

Image	CRR			NLR			CRNS (CRR + NLR)		
	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Butterfly	28.77	0.9262	0.9270	28.97	0.9304	0.9258	**29.20**	**0.9322**	**0.9308**
Parrots	30.57	0.9145	0.9443	30.37	0.9151	0.9442	**30.64**	**0.9165**	**0.9460**
Parthenon	27.10	0.7457	0.8334	27.13	0.7478	0.8315	**27.22**	**0.7500**	**0.8343**
Bike	24.84	0.8074	0.8595	24.82	0.8108	0.8642	**25.00**	**0.8141**	**0.8653**
Flower	29.41	0.8543	0.8974	29.59	0.8603	0.9025	**29.64**	**0.8610**	**0.9035**
Girl	33.56	0.8247	0.9168	33.55	0.8241	0.9159	**33.60**	**0.8255**	**0.9173**
Hat	31.51	0.8782	0.9108	31.40	0.8792	0.9097	**31.66**	**0.8814**	**0.9128**
Leaves	28.06	0.9295	0.9245	28.02	0.9331	0.9213	**28.41**	**0.9360**	**0.9269**
Plants	34.39	0.9210	0.9400	34.29	0.9227	0.9378	**34.57**	**0.9241**	**0.9405**
Raccoon	29.31	0.7650	0.8716	29.31	0.7673	0.8735	**29.39**	**0.7688**	**0.8745**
Average	29.75	0.8567	0.9025	29.74	0.8591	0.9026	**29.93**	**0.8610**	**0.9052**

Figure 2.8: SR Results of *Hat* (×3) Produced by Different Regularization Terms. From Left to Right: CRR (PSNR: 31.51 dB, SSIM: 0.8782), NLR (31.40 dB, 0.8792), CRNS (**31.66 dB**, **0.8814**), Original Image.

In Eq. (2.46), besides the data fidelity term, both the CRR and NLR priors are involved. To validate the effectiveness of different regularization terms, we report the PSNR, SSIM and FSIM values of the SR results by different priors in Table 2.6.

According to Table 2.6, the following conclusions can be made: Firstly, separately incorporating the CRR prior or the NLR prior is able to achieve competitive output results. The average results obtained by CRR and NLR are both superior to the second best method, i.e., NCSR in Table 2.5. Secondly, by jointly incorporating the CRR prior and the NLR prior, the proposed algorithm CRNS constantly outperforms the two separated regularization priors.

The SR results of *Hat* by different regularization priors are shown in Fig. 2.8. It can be observed that the NCSR not only can provide sharp edges, but also can suppress noticeable artifacts in the resultant images. The reason lies in that the complementary advantages of both the external prior information and the internal NLSS of natural images are combined by the joint prior, resulting in a more faithful SR reconstruction.

Empirical study on parameters

In this subsection, the settings of the important parameters of CRNS are discussed. Note that all the discussions are based on a magnification factor of 3 and a Gaussian

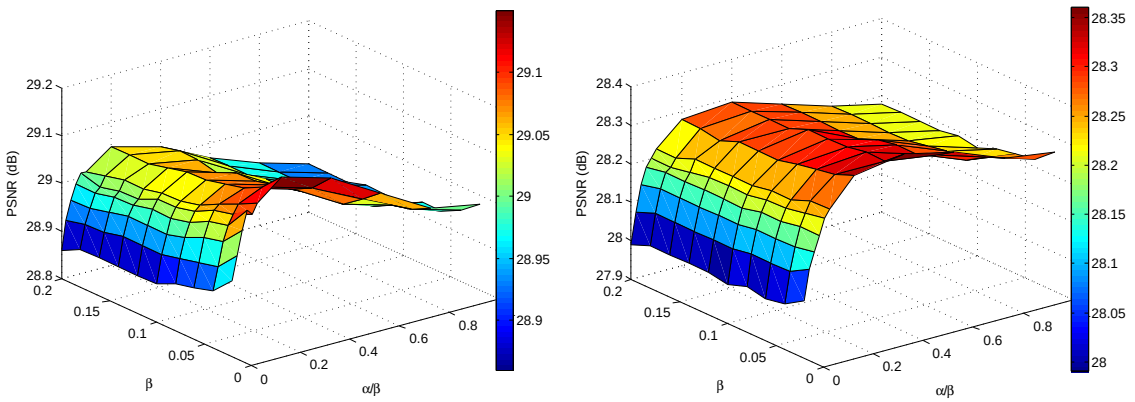


Figure 2.9: PSNR Surface Describing the Effects of α and β. The Left and Right Figures Are Based on the Experiments for Image: Butterfly and Image: Leaves.

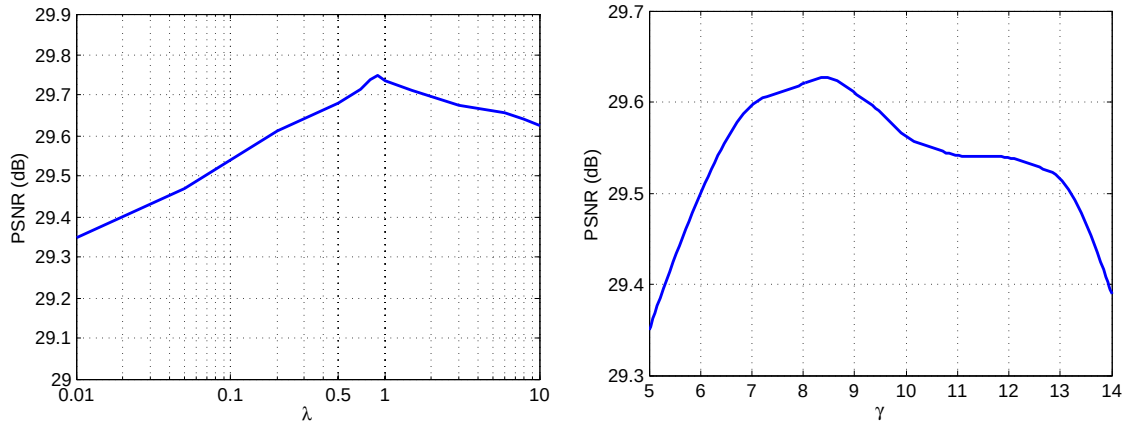


Figure 2.10: The Effects of λ in CRR and γ in NLR.

kernel with a size of 7×7 and a standard deviation of 1.6.

1) *Trade-off parameters α and β*: α and β indicate the importance of CRR term and NLR term in Eq. (2.46), respectively. We take *Butterfly* and *Leaves* as examples to show how α and β affect the performance of CRNS. The surfaces of PSNR for *Butterfly* and *Leaves* are presented in Fig. 2.9, where β varies from 0.02 to 0.2 and the ratio of α to β changes from 0 to 1.0. It is obvious that in both cases, the better results can be obtained when the value of β is lower than 0.04. In addition, the best ratio of α to β occurs around $0.20 \sim 0.40$. Therefore, after further exploring, we finally fix $\alpha = 0.0015$ and $\beta = 0.005$.

2) *Trade-off parameters λ and γ*: λ and γ are the trade-off parameters in (2.39)

Table 2.7: Effects of Patch Size (×3)

Patch Size	CRR		NLR	
	PSNR	SSIM	PSNR	SSIM
3×3	28.89	0.8402	28.98	0.8406
4×4	29.35	0.8496	29.63	0.8552
5×5	29.75	0.8567	29.74	0.8591
6×6	29.63	0.8558	29.56	0.8567
7×7	29.59	0.8543	29.40	0.8546

Table 2.8: Effects of pixel Overlapping (×3)

OverlappingPixels	CRR		NLR	
	PSNR	SSIM	PSNR	SSIM
1	29.22	0.8475	29.65	0.8583
2	29.43	0.8514	29.74	0.8591
3	29.75	0.8567	29.81	0.8598
4	29.83	0.8574	29.85	0.8602

and (2.38), respectively. Fig. 2.10 (a) shows the average PSNR obtained by CRR on the 10 test images in Fig. 2.5 with various λ, and Fig. 2.10 (b) presents the average PSNR obtained by NLR with different γ. According to Fig. 2.10, we choose $\lambda = 0.9$ and $\gamma = 8.75$, where the peak average PSNR values can be reached.

3) *Patch Sizes, Numbers of Overlapping Pixels, Number of Similar Patches for CRR and NLR*: The impacts of these parameters of CRR and NLR are evaluated and respectively listed in Tables 2.7, 2.8 and 2.9, where the average PSNR and SSIM results on the 10 test images in Fig. 2.5 are given. The results in Table 2.7 imply that it is better to set the patch size to 5×5 for both CRR and NLR. Table 2.8

Table 2.9: Effects of Number of Similar Patches (×3)

Number ofSimilar Patches	CRR		NLR	
	PSNR	SSIM	PSNR	SSIM
8	29.72	0.8560	29.64	0.8573
10	29.75	0.8567	29.71	0.8584
12	29.75	0.8567	29.74	0.8591
15	29.75	0.8567	29.77	0.8597
20	29.76	0.8568	29.79	0.8599

shows that more overlapping pixels lead to better performance of CRR and NLR. However, increasing the number of overlapping pixels results in more computational burden, especially for NLR. Therefore, we assign 2 overlapping pixels for NLR and 3 overlapping pixels for CRR. The results in Table 2.9 reveal that NLR is able to achieve better performance with a large number of similar patches. However, to balance between the performance and the complexity, we empirically set it to 12 for NLR. Due to the fact that the improvement of CRR with more than 10 similar patches is not obvious, this parameter is fixed as 10 for CRR.

Convergence of the CRNS algorithm

Fig. 2.11 shows how the PSNR varies with the number of iterations for different test images. We can see that all the PSNR curves increase monotonically with the growth of iteration number (note that CRNS contains an outer loop and an inner loop, and the iteration number here refers to the total number of updating $\mathbf{X}$ via Eq. (2.53)). In the first few iterations, the PSNR improvements grow rapidly. After about 10 iterations, the rate of change of PSNR decreases, but the curves still gradually rise. The PSNR curves reach (or almost reach) their peak values at an iteration number of

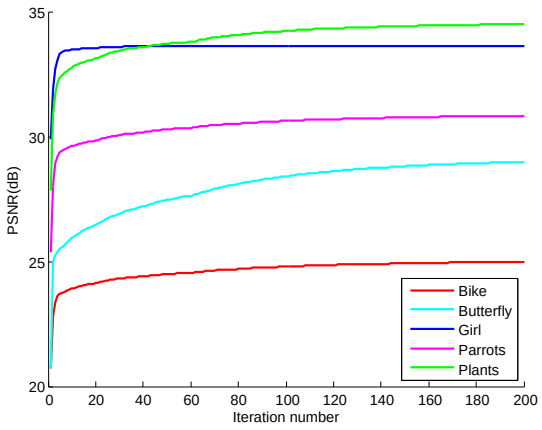


Figure 2.11: Verification of the Convergence of CRNS.

Table 2.10: Average PSNR (dB), SSIM and FSIM Results on Different Datasets ($\times 3$)

DatasetMethod	A+	SRCNN	FSRCNN	ASDS-AR-NL	NCSR	CRNS
Set5	32.30	32.37	32.60	32.62	33.01	**33.38**
	0.9027	0.9010	0.9068	0.9051	0.9100	**0.9149**
	0.9324	0.9344	0.9392	0.9349	0.9427	**0.9490**
Set14	28.88	28.97	28.95	29.10	29.27	**29.45**
	0.8089	0.8107	0.8137	0.8170	0.8197	**0.8226**
	0.9373	0.9379	0.9393	0.9412	0.9436	**0.9452**
BSD100	28.01	28.07	28.18	28.25	28.35	**28.46**
	0.7703	0.7739	0.7796	0.7803	0.7837	**0.7853**
	0.8619	0.8674	0.8715	0.8707	0.8741	**0.8744**

200. This phenomena also verifies that choosing the maximum numbers of the inner iterations K_{m} and the outer iterations T_{m} to be 20 and 10 is reasonable.

SR results ($\times 3$) on different datasets

In order to comprehensively verify the effectiveness of the CRNS algorithm and the robustness of the parameters selected by our empirical study, we conduct experiments

Figure 2.12: SR Results of *86000* from *BSD100* (×3), from Left to Right: A+ (Timofte *et al.*, 2014) (PSNR: 26.28 dB, SSIM: 0.8068), SRCNN (Dong *et al.*, 2016a) (26.33 dB, 0.8104), FSRCNN (Dong *et al.*, 2016b) (26.56 dB, 0.8199), ASDS-AR-NL (Dong *et al.*, 2011) (27.01 dB, 0.8304), NCSR (Dong *et al.*, 2013) (27.17 dB, 0.8358) , CRNS (**27.58 dB**, **0.8479**), Original Image.

on other three commonly used image datasets, including *Set5* (Bevilacqua *et al.*, 2012), *Set14* (Zeyde *et al.*, 2010) and *BSD100* (Timofte *et al.*, 2014). The degradation model used in this subsection is the same as that in Section 2.3.3. Among all the benchmark methods in Section 2.3.3, the best three learning-based methods (A+ (Timofte *et al.*, 2014), SRCNN (Dong *et al.*, 2016a) and FSRCNN (Dong *et al.*, 2016b)) and the best two reconstruction-based methods (ASDS-AR-NL (Dong *et al.*, 2011) and NCSR (Dong *et al.*, 2013)) are compared with CRNS. Due to the limitation of length, we only show the average PSNR/SSIM/FSIM values of the three image datasets in Table 2.10, and the super-resolved results of *86000* from dataset *BSD100* in Fig. 2.12. It can be observed from Table 2.10 that CRNS is superior to the other methods in all cases. In Fig. 2.12, the edges of the windows produced by CRNS are sharper and cleaner than the other methods.

Robustness to noise

In this subsection, we carry out experiments on noisy images, so as to validate the robustness of CRNS algorithm to noise. We use the same blur kernel, decimation factor and test images as that in Section 2.3.3. However, additive Gaussian noise with a standard deviation $\sigma_{\mathrm{n}} = 5$ is added to the LR images. Note that because of

Table 2.11: Average PSNR (dB), SSIM and FSIM Results on Noisy Images ($\times 3$, $\sigma_{\mathrm{n}} = 5$)

MetricMethod	A+	SRCNN	FSRCNN	ASDS-AR-NL	NCSR	CRNS
PSNR (dB)	27.19	26.93	26.33	27.80	28.19	**28.42**
SSIM	0.7260	0.7077	0.6992	0.7854	0.8002	**0.8020**
FSIM	0.8394	0.8324	0.8117	0.8525	0.8618	**0.8713**

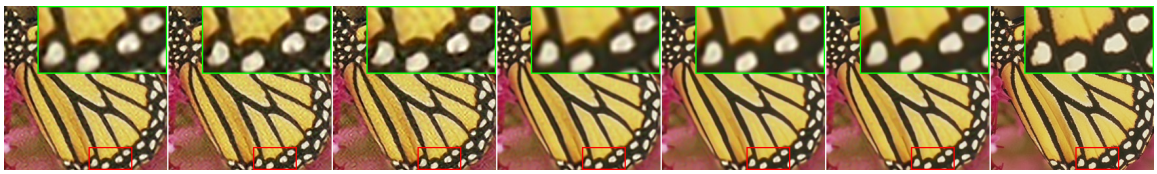


Figure 2.13: SR Results of Noisy *Butterfly* ($\times 3$, $\sigma_{\mathrm{n}} = 5$), from Left to Right: A+ (Timofte *et al.*, 2014) (PSNR: 25.87 dB, SSIM: 0.8109), SRCNN (Dong *et al.*, 2016a) (25.94 dB, 0.7850), FSRCNN (Dong *et al.*, 2016b) (25.56 dB, 0.7689), ASDS-AR-NL (Dong *et al.*, 2011) (25.96 dB, 0.8577), NCSR (Dong *et al.*, 2013)(26.80 dB, 0.8851) , CRNS (**27.54 dB**, **0.8918**), Original Image.

the presence of noise, the data fidelity term in problem (2.47) becomes less reliable. Therefore, to guarantee the quality of SR results, the values of α and β in (2.47) are enlarged to 0.0375 and 0.125, respectively. The average PSNR/SSIM/FSIM results obtained by different methods are listed in Table 2.11, and the visual SR results of *Butterfly* are presented in Fig. 2.13. Compared with the noiseless case, the performances of A+ (Timofte *et al.*, 2014), SRCNN (Dong *et al.*, 2016a) and FSRCNN (Dong *et al.*, 2016b) decrease significantly, which indicates that these methods are sensitive to noise. Among the tested methods, CRNS achieves the best objective results. Furthermore, the image reconstructed by CRNS is clean and with the least artifacts.

Table 2.12: Average PSNR (dB), SSIM and FSIM Results at Different Magnification Factors

Method	×2	×3	×4
ASDS-AR-NL	31.36	29.24	26.99
	0.8947	0.8473	0.7765
	0.9240	0.8896	0.8421
NCSR	31.70	29.53	27.27
	0.8993	0.8542	0.7902
	0.9301	0.8993	0.8535
CRNS	**32.14**	**29.93**	**27.52**
	0.9033	**0.8610**	**0.7965**
	0.9360	**0.9052**	**0.8599**

Figure 2.14: SR Results of *Flower* (Blur Kernel Size: 7×7, Standard Deviation: 1.6, Magnification Factor: ×4), from Left to Right: ASDS (Dong *et al.*, 2011) (PSNR: 26.70 dB, SSIM: 0.7512), NCSR (Dong *et al.*, 2013) (26.93 dB, 0.7685), CRNS (**27.16 dB**, **0.7765**), Original Image.

Experimental results for different degradation models

To fully assess the effectiveness of the proposed CRNS algorithm, different kinds of degradation models are considered. Note that the settings of CRNS are the same as described in Section 2.3.3. First of all, we conduct the experiments by applying the same blur kernel (blur kernel size: 7×7, standard deviation: 1.6) at different magnification factors 2, 3, and 4. Then we carry out experiments by applying the

Table 2.13: Average PSNR (dB), SSIM and FSIM Results at Different Blur Kernel Sizes

Method	3×3	5×5	7×7
ASDS-AR-NL	28.44	29.20	29.24
	0.8347	0.8480	0.8473
	0.8837	0.8903	0.8896
NCSR	29.00	29.55	29.53
	0.8513	0.8567	0.8542
	0.8989	0.9004	0.8993
CRNS	**29.35**	**29.96**	**29.93**
	0.8563	**0.8632**	**0.8610**
	0.9041	**0.9068**	**0.9052**

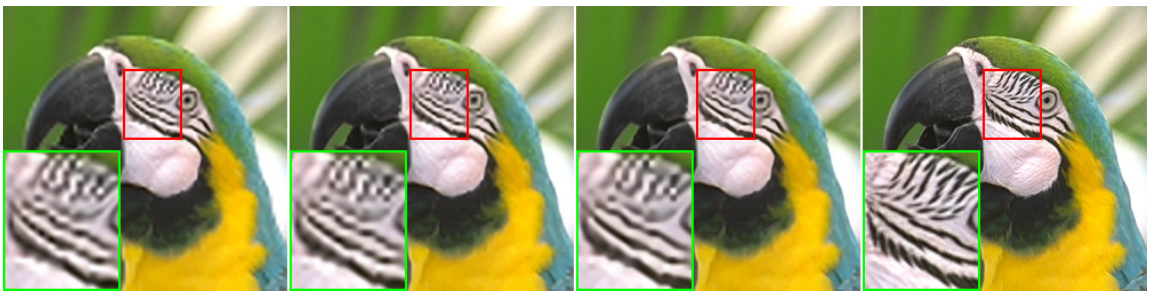


Figure 2.15: SR Results of *Parrots* (Blur Kernel Size: 3×3, Standard Deviation: 1.6, Magnification Factor: $\times 3$), from Left to Right: ASDS (Dong *et al.*, 2011) (PSNR: 29.25 dB, SSIM: 0.9012), NCSR (Dong *et al.*, 2013) (29.57 dB, 0.9129), CRNS (**30.10 dB**, **0.9168**), Original Image.

Gaussian kernel with the same standard deviation of 1.6 but different sizes 3×3, 5×5 and 7×7 at a magnification factor of 3. The reconstruction-based methods ASDS-AR-NL (Dong *et al.*, 2011) and NCSR (Dong *et al.*, 2013) are selected for comparison in this subsection.

The average values of PSNR, SSIM and FSIM obtained on the 10 test images in Fig. 2.5 at different magnification factors and different blur kernel sizes are re-

spectively given in Tables 2.12 and 2.13. For subjective quality comparisons, the SR results of *Flower* and *Parrots* are given in Figs. 2.14 and 2.15, respectively. We can find that CRNS is able to achieve the best objective results at different kinds of degradation models. In addition, the subjective results show that the proposed algorithm is cable of recovering much sharper edges than the other two baseline methods.

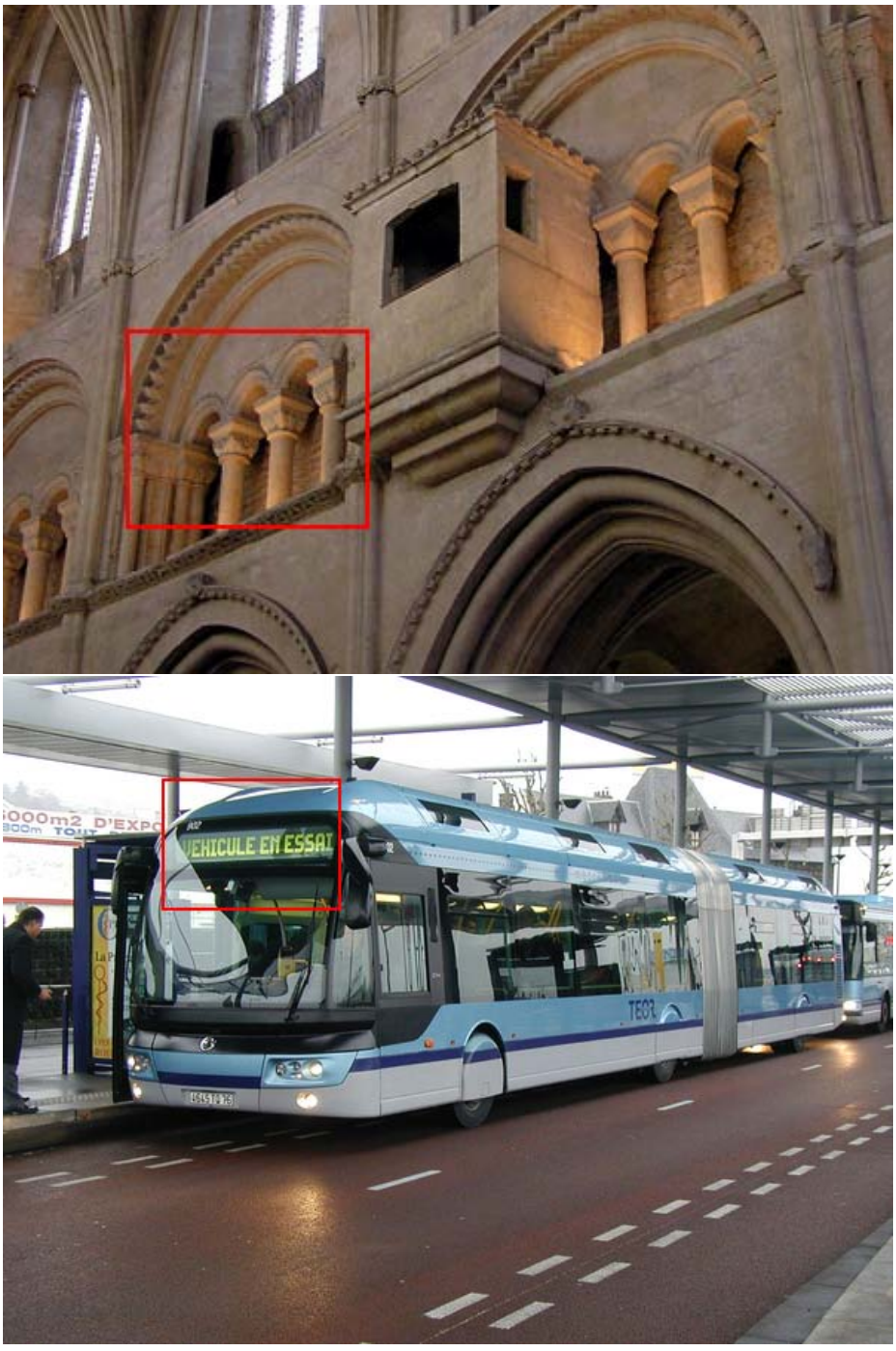


Figure 2.16: Actual Images and Their Corresponding ROI Areas for Testing.

Figure 2.17: SR Results of the ROI in *abbey* Image (×3), from Left to Right: Bicubic Interpolation, A+ (Timofte *et al.*, 2014), SRCNN (Dong *et al.*, 2016a), FSRCNN (Dong *et al.*, 2016b), ASDS-AR-NL (Dong *et al.*, 2011), NCSR (Dong *et al.*, 2013), CRNS.

Figure 2.18: SR Results of the ROI in *Bus* Image (×3), from Left to Right: Bicubic Interpolation, A+ (Timofte *et al.*, 2014), SRCNN (Dong *et al.*, 2016a), FSRCNN (Dong *et al.*, 2016b), ASDS-AR-NL (Dong *et al.*, 2011), NCSR (Dong *et al.*, 2013), CRNS.

Table 2.14: No-reference IQA (Using the Metric in (Ma *et al.*, 2017)) on Actual Images (×3)

Image	Bicubic	A+	SRCNN	FSRCNN	ASDS-AR-NL	NCSR	CRNS
Abbey	4.24	7.33	7.05	7.38	7.23	7.27	**7.57**
Bus	5.10	6.48	6.79	7.02	6.46	6.45	**7.14**
Average	4.67	6.91	6.92	7.20	6.85	6.86	**7.36**

SR results on actual images

In addition to the previous experiments on simulated LR images, we further carry out experiment on actual LR images (available from the sun database (Xiao *et al.*, 2010)), so as to validate the feasibility and robustness of the CRNS algorithm in a real acquisition environment. The selected two actual images and the corresponding region of interest (ROI) are shown in Fig. 2.16. The SR results of the ROI areas in *Abbey* and *Bus* by different SR methods with a magnification factor of 3 are presented in Figs. 2.17 and 2.18, respectively. In the two figures, we can find that: firstly, the bicubic interpolation generates the blurriest images; secondly, the other five baseline methods are able to produce sharper edges than bicubic interpolation, but they are not good at suppressing artifacts and noises; finally, the images reconstructed by the CRNS algorithm have the sharpest edges and the least artifacts/noises. Moreover, the no-reference image quality assessment (IQA) metric in (Ma *et al.*, 2017), which is designed to evaluate the quality of the SR results based on human perception, is adopted here to quantitatively compare different SR methods. The no-reference IQA values are given in Table 2.14, where we can see that CRNS obtains the highest scores. Overall, the experimental results in this subsection indicate that the proposed CRNS algorithm is also effective on actual images.

Table 2.15: Offline Training Time of Different Methods

Method	Zeyde	ANR	A+	LANR-NLM
Time	102.9 s	103.2 s	1328.6 s	104.4 s
Method	SRCNN	FSRCNN	ASDS-AR-NL	CRNS
Time	days	days	330.7 s	2687.3 s

Table 2.16: Average SR Reconstruction Time on 256×256 Test Images ($\times 3$)

Zeyde	ANR	A+	LANR-NLM	SRCNN	FSRCNN	ASDS
0.6 s	0.2 s	0.2 s	38.3 s	1.7 s	1.9 s	120.5 s
ASDS-AR	ASDS-AR-NL	NCSR	CRR	NLR	CRNS	CRNS-fast
126.8 s	146.3 s	191.3 s	142.9 s	259.1 s	337.9 s	197.5 s

Analysis of complexity

The computational burden of Algorithm 4 mainly focuses on finding the K nearest neighbours for the atoms in the dictionary $\mathbf{D}_{\mathrm{H}}$. Assume that N_{T} is the number of training samples, and the size of an atom is P_{s}. Thus for each dictionary atom, locating the K nearest neighbours costs $O(N_{\mathrm{T}}P_{\mathrm{s}})$. Table 2.15 shows the offline training time of different methods. Note that the training procedure of the proposed method is only carried out in the HR space. Therefore, when the degradation model changes, there is no need to re-train the model of CRNS.

The most complex parts of Algorithm 5 are analyzed below:

1. *Finding similar patches for each exemplar patch at each outer iteration*: This procedure costs $O(N_{\mathrm{e}}W_{\mathrm{s}}^2P_{\mathrm{s}})$, where $\sqrt{P_{\mathrm{s}}} \times \sqrt{P_{\mathrm{s}}}$ is the patch size, $W_{\mathrm{s}} \times W_{\mathrm{s}}$ is the size of the search window, N_{e} denotes the number of exemplar patches in an image.

2. *Updating each* $\mathbf{L}_i$ *at each inner iteration*: SVD is needed to calculate each low rank matrix $\mathbf{L}_i$. The complexity of thin SVD is $O(P_{\mathrm{s}}N_{\mathrm{p2}}r)$, where N_{p2} stands for the number of similar patches in a group, r is the rank of matrix $\mathbf{F}_i\mathbf{X}$.

3. *Locating the matched* $\mathbf{P}_j$ *for each exemplar patch at each inner iteration*: Each exemplar patch should be compared with the M atoms in dictionary $\mathbf{D}_{\mathrm{H}}$. Thus it takes $O(N_{\mathrm{e}}P_{\mathrm{s}}M)$ to locate $\mathbf{P}_j$ for all the N_{e} exemplar patches.

4. *Using CG method to update* $\mathbf{X}$ *at each inner iteration*: Because the matrix-vector product is the dominating part in CG method, the complexity of one CG iteration is $O(N^2)$, where N stands for the number of pixels in an image.

Table 2.16 shows the average running times of different methods spent on 256×256 test images [3] . It can be seen that the learning-based methods (Zeyde *et al.*, 2010; Timofte *et al.*, 2013, 2014; Jiang *et al.*, 2017; Dong *et al.*, 2016a,b) are much faster than the reconstruction-based methods. On the other hand, combing both the CRR and NLR priors makes our CRNS algorithm hard to compute.

To speed up the CRNS algorithm, one possible way is to cluster the image patches and assign the same self-projection matrix to all the patches in a cluster. However, this approach suffers from performance degradation, especially when the number of clusters is small. By analyzing the CRNS algorithm, we know that finding similar patches, locating the matched $\mathbf{P}_j$, and updating $\mathbf{L}_i$ for an exemplar patch are independent from that for the other exemplar patches. Therefore, utilizing the parallelization techniques is another potential solution to accelerate the above procedures, and this solution does not need to sacrifice any performance. The 'CRNS-fast' in Table 2.16 denotes the implementation of CRNS with the parallel loops in Matlab

[3]Note that the Matlab implementation of FSRCNN used in this paper (available: http://mmlab.ie.cuhk.edu.hk/projects/FSRCNN.html) is slower than the one reported in (Dong *et al.*, 2016b).

(connecting to 4 workers). It is obvious that CRNS-fast is much faster than the original implementation of CRNS without using any parallelization technique.

2.3.4 Appendix: Using CG Method to Calculate (2.53)

Consider Eq. (2.53) as the solution of the following linear system problem

$$\mathbf{A}\hat{\mathbf{X}} = \mathbf{V} \tag{2.56}$$

We can easily find that

$$\begin{aligned} \mathbf{A}^T = &\left[(\mathbf{SH})^T(\mathbf{SH})\right]^T + \beta \sum_i (\mathbf{F}_i^T \mathbf{F}_i)^T \\ &+ \alpha \sum_i \left[\mathbf{R}_i^T (\mathbf{I} - \mathbf{P}_j)^T (\mathbf{I} - \mathbf{P}_j)\mathbf{R}_i\right]^T \\ =&\mathbf{A} \end{aligned} \tag{2.57}$$

So matrix $\mathbf{A}$ is symmetric. Also, $\forall \mathbf{X} \neq 0$, we have

$$\begin{aligned} \mathbf{X}^T\mathbf{A}\mathbf{X} =&\mathbf{X}^T(\mathbf{SH})^T(\mathbf{SH})\mathbf{X} + \beta \sum_i \mathbf{X}^T \mathbf{F}_i^T \mathbf{F}_i \mathbf{X} \\ &+ \alpha \sum_i \mathbf{X}^T \mathbf{R}_i^T (\mathbf{I} - \mathbf{P}_j)^T (\mathbf{I} - \mathbf{P}_j)\mathbf{R}_i \mathbf{X} \\ =&||\mathbf{SHX}||_2^2 + \alpha \sum_i ||(\mathbf{I} - \mathbf{P}_j)\mathbf{R}_i\mathbf{X}||_2^2 + \beta \sum_i ||\mathbf{F}_i\mathbf{X}||_2^2 \\ >&0 \end{aligned} \tag{2.58}$$

So matrix $\mathbf{A}$ is positive definite. Therefore, it is beneficial to apply any algorithm in the family of CG methods to calculate Eq. (2.53). In our implementation, the Matlab built-in function *pcg* is applied.

Chapter 3

DEEP LEARNING FOR ACCELERATED MRI RECONSTRUCTION

3.1 Deep Residual Dense U-net for Resolution Enhancement in Accelerated MRI Acquisition

Magnetic resonance imaging (MRI) is among the most important imaging methods for medical diagnosis. Obtaining fully sampled MRI data requires relatively long scan time. To shorten MRI scan time for improved patient experience and reduced cost, researchers have investigated accelerated MRI acquisition. One basic idea for acceleration is to under-sample in k-space, which may cause aliasing in the reconstructed images. Parallel MRI (Pruessmann *et al.*, 1999; Griswold *et al.*, 2002) and compressed sensing (CS) MRI (Donoho, 2006; Lustig *et al.*, 2008) are two popular techniques. A representative technique of the parallel MRI, generalized auto-calibrating partial parallel acquisition (GRAPPA) (Griswold *et al.*, 2002), uses interpolation to fill the missing k-space data with the surrounding data from all the coils, while CS-MRI randomly samples the k-space data for approximating the original image.

Approaches using low-rank matrix completion technique to solve the CS-MRI / parallel MRI problem were also investigated. Representatives include SAKE (Shin *et al.*, 2014) and the annihilating filter based low-rank Hankel matrix approach (ALOHA) (Lee *et al.*, 2016). However, these algorithms have high complexity and k-space data are required during the reconstruction, making them impossible for cases with only image domain inputs.

In recent years, deep learning has become one of the most important tools for visual computing research (Ren *et al.*, 2015; He *et al.*, 2017), with great performance

in image classification (Krizhevsky *et al.*, 2012), segmentation (Ronneberger *et al.*, 2015), recognition (He *et al.*, 2016), super resolution (Tong *et al.*, 2017), etc. Therefore, some researcher started to utilize deep-learning techniques for medical image reconstruction. Wang et al (Wang *et al.*, 2016) trained a convolutional neural network (CNN) to learn the mapping from the aliased image to the original fully sampled reconstruction. The output of the network can be used as an initial guess or regularization term in conventional CS approaches. In ref.70, the authors proposed a multilayer perceptron for parallel MRI, and in ref.48 the researchers applied CNN on CS algorithm. Kang et al (Kang *et al.*, 2017) applied the CNN technique on computed tomography (CT), etc.

The authors of a recent paper (Lee *et al.*, 2017) used U-Net (Ronneberger *et al.*, 2015) with residual learning to learn the relationship between the aliased and original images, and the proposed framework outperforms traditional methods like SENSE and ALOHA. However, since U-Net is originally developed for medical image segmentation, it is likely that directly using it on image reconstruction may not give us the best performance. For this reason, we propose Residual Dense U-Net (RD-U-Net), a U-Net based deep neural network to further improve the quality of the reconstructed image. Inspired by DenseNet (Huang *et al.*, 2017), a residual dense block for refinement is introduced to improve the quality of the reconstructed image. Furthermore, we impose the Fourier constraint into the loss function. Experimental results show that, both visually and numerically, our proposed architecture has a better performance.

3.1.1 Related Works

In this section, we review the works that related to our proposed RD-U-Net model for accelerated MRI reconstruction. We note that there is a line of work on super-

resolution using sparse represent ion (e.g., (Kulkarni *et al.*, 2012)), which has delivered superior performance for super-resolution in natural images. However, such work is not amenable for the task of MR image reconstruction in this paper since the degradation involves structured aliasing.

U-Net

U-Net (Ronneberger *et al.*, 2015) is first proposed for biomedical image segmentation, which incorporates skip connection and downsampling/upsampling layers. These skip connection intend to extract local information, while the encoding/decoding procedure provide global information. While obtaining state-of-the-art results, U-Net is applied to other visual computing field like accelerated MRI reconstruction (Lee *et al.*, 2017) and pansharpening (Yao *et al.*, 2018).

Residual learning

Optimization is an important procedure to deep learning. A major challenge for the optimization is the gradient vanishing problem. To overcome this issue, the concept of residual learning is introduced in the residual network (ResNet) (He *et al.*, 2016). In this model, a shortcut connection (skip connection) is used in every basic residual block, which makes the gradient flow in the networks is relatively stable. It is also shown that in ref.74, adding skip connection leads to the flatness of loss surfaces. ResNet has provided very promising results on many applications.

Densely Connected Convolutional Networks

Many works have shown that the networks perform better if there are connections between layers close to the input and the ones close to the output. The authors in ref.55 propose Dense Convolutional Network (DenseNet), which connects each layer to

every other layer by using skip connections. Differ from the traditional convolutional network, every layer in DenseNet takes the feature maps from all preceding layers as inputs, and its output feature maps are served as inputs for the subsequent layers. DenseNet achieves state-of-the-art performance in a lot of real world problems.

3.1.2 Proposed Approach

For traditional super resolution, patch-based (Tong *et al.*, 2017; Ding *et al.*, 2017) approaches are usually used. However, for accelerated MRI reconstruction, the aliasing artifacts are of global nature. To diminish the global artifacts, we can use the whole image as an input. The authors of the ref.72 use U-Net with residual learning to handle this problem. To be precise, let x be the input image, and $\tilde{f}$ be the function represented by the U-Net, the output of the model can be defined as:

$$y = f(x) = x + \tilde{f}(x) \tag{3.1}$$

where f is the function representing the whole network. In this case $\tilde{f}(x) = y - x$, and since x and y are the low resolution and high resolution image respectively, $\tilde{f}$ maps x to the residual. Learning such $\tilde{f}$ leads to faster convergence(Lee *et al.*, 2017).

The proposed network architecture is illustrated in Fig. 3.1. Similar to the U-Net-based approach (Lee *et al.*, 2017), it consists of an encoding path and a decoding path. The encoding path is a traditional convolutional neural network, which consists of two 3 x 3 convolutions, each followed by a batch normalization (BN) layer and a nonlinear activation layer. After that, a 2 x 2 max pooling layer with stride 2 is applied for down sampling, and the number of channels is doubled after the down sampling. For the decoding path, at every stage it consists of a 2 x 2 deconvolution which up sample the feature map and reduce the number of channels by half. After up-sampling, feature maps from the same level in the encoding path are fed to the

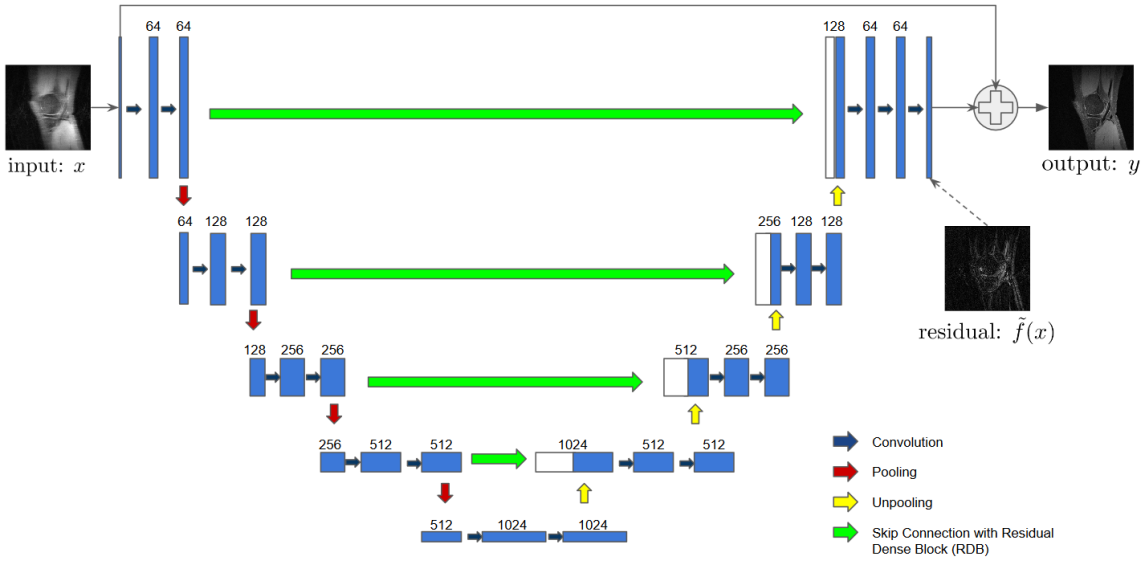


Figure 3.1: The Illustration of Our Proposed RD-U-Net. Blue Arrow: 3x3 Convolution + Batch Normalization + Nonlinear Activation; Red Arrow: 2x2 Max Pooling with Stride 2; Yellow Arrow: 2 X 2 Up-convolution; Green Arrow: Skip Connection with Residual Dense Block.

Residual Dense Block, and the corresponding output is concatenated, followed by two 3 x 3 convolutional layers, a BN layer and a nonlinear activation layer. A 1 x 1 convolutional layer is used at the final to map all the features to a single channel.

Refinement Using Residual Dense Block

U-Net has its limitations for extracting high frequency data. Since the high frequency content are only contained in the upper part of the network (the earlier stages of the encoding part, and the later stages of the decoding part), The network is not deep enough to extract the high frequency features. Inspired by DenseNet (Huang *et al.*, 2017), we introduce Residual Dense Block (RDB) to refine the feature map. RDB is formed by the "dense" part and "residual" part. For the "dense" part, the input is first passed through a convolutional layer, a batch normalization layer and a nonlinear activation layer, where the number of filters used in the convolutional layer is the same

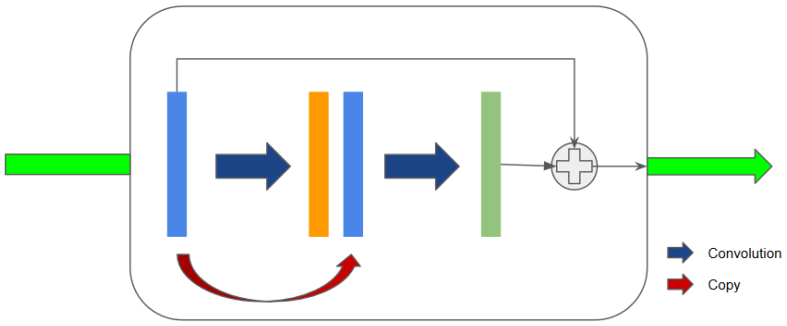


Figure 3.2: The Illustration of the Proposed Residual Dense Block (RDB). Blue Arrow: 3x3 Convolution + Batch Normalization + Nonlinear Activation; Red Arrow: Skip Connection; "Dense" Part: The Input (Blue) Is First Convolved, And Concatenate to Itself, And Then the Second Convolution Is Applied; "residual" Part: The Input and Output of the Dense Part Are Summed Together to Learn the Residual.

GRAPPA	U-Net	RD-U-Net($\alpha = 0$)	RD-U-Net($\alpha = 0.01$)
0.1483 ± 0.0052	0.0338 ± 0.0003	0.0326 ± 0.0002	0.0319 ± 0.0002

Table 3.1: The Mse (Mean and Standard Deviation) for Different Models on 5 Trials: Grappa, U-net, Rd-u-net Without Fourier Constraint (Denoted by $\alpha = 0$), Rd-u-net with Fourier Constraint (Denoted by $\alpha = 0.01$).

as the number of channels of its input. The output is then concatenated with the input of the RDB, and passed through another convolutional layer again, and reduce the number of channels by half. At the end the input of the RDB is added to the output of the "dense" part, to form the "residual" part. Fig. 3.2(a) is an illustration.

Instead of using the plain skip connection (copying the feature maps from the encoding part to the decoding part), adding a refinement is a more reasonable choice, as theoretically, the plain skip connection is a special case of the RDB (by setting all the weights of the second convolution in RDB to be zero).

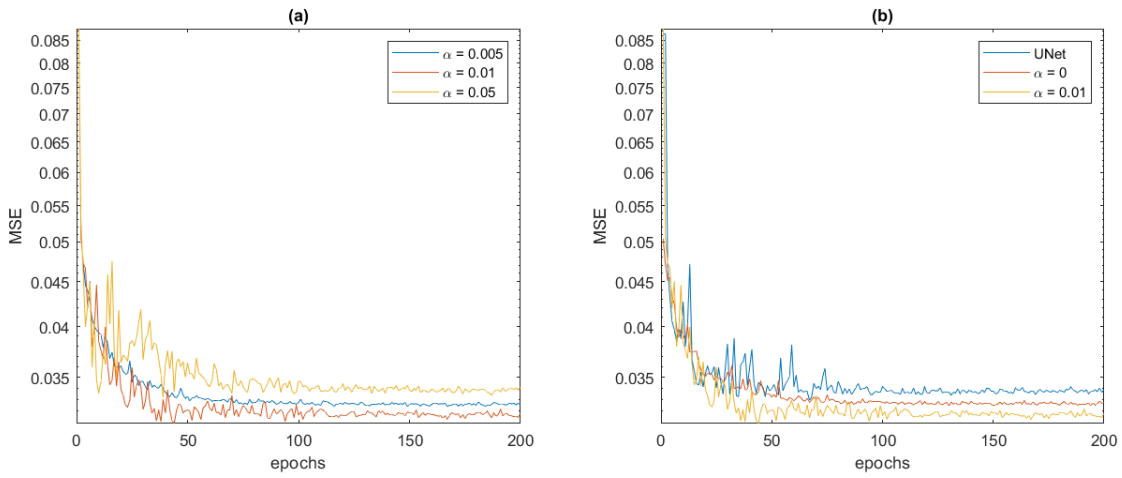


Figure 3.3: The Plots for the Testing Loss Versus Number of Epochs Trained in a Trial. (A) the Testing Loss for Different Value of α; (B) the Testing Loss for U-Net, RD-U-Net Without Fourier Constraint (Denoted by $\alpha = 0$) and with Fourier Constraint (Denoted by $\alpha = .01$).

Employing Fourier Constraints

L_2 loss is usually used in image reconstruction tasks, which is defined as follows:

$$\min \|y - f(x)\|_2 \tag{3.2}$$

where f is the mapping represented by the neural network. However, given that the degradation of the images are come from the missing of columns (or rows) of in the K-space data, we can use this prior to improve the performance, by using the following loss:

$$\min \|y - f(x)\|_2 + \alpha \|F(y) - F(f(x))\|_1 \tag{3.3}$$

where y and x represent the ground truth and the degraded image, respectively, f is the mapping represented by the neural network, F is the inverse Fourier transform, and α is a constant. As we expect the lost information only comes from part of the rows in the Fourier domain, we use L_1 norm for the Fourier regularization term. With this loss function we can effectively regularize the network learning by the error coming from those missing k-space data.

3.1.3 Experiments

In this section, we introduce the dataset we use, followed by the network settings, and comparative studies.

Magnetic Resonance Dataset

We use the fully sampled knee datasets from mridata.org to evaluate our RD-U-Net. There are 20 datasets, and the data were acquired in Cartesian coordinate on a GE clinical 3T scanner, with the following parameters: Receiver Bandwidth = 50.0, number of coils = 8, acquisition matrix size = 320 × 320, FOV = 160mm × 160mm × 153.6mm, number of slices = 256, TR = 1550ms, TE = 25ms, FA = 90, sequence type = SE. For each dataset, we only use the 150 slices from the central part (the 51st to 200th slices). We randomly pick 1 dataset for training, and 1 dataset for testing. For data augmentation, we generate 8 times more training samples by rotating and reflecting the images. The original k-space data are retrospectively down sampled by 4 times with 16 ACS (auto-calibration signal, 5% of total PE). The sampling pattern, a high resolution reconstructed image and a low resolution reconstructed image (by zero-filled) are shown in Fig. 3.4.

As the images are acquired under different scan conditions, normalization is necessary for the preprocessing of the data. We applied the following transformation for every single images:

$$x \leftarrow \frac{x - mean(x)}{std(x)} \tag{3.4}$$

After the transformation, the pixels in an image will have zero mean and unit variance.

Experimental Protocols and Results

We train the network for 200 epochs with batch size = 3. Stochastic gradient descent (SGD) is used with initial learning rate = 0.02 and momentum = 0.5. For every 20 epochs the learning rate is decreased by half. We use PoLU (Li *et al.*, 2018b) as our activation function. For calculating the error, we use mean square error (MSE), which can be defined as:

$$MSE = \frac{\|y - f(x)\|_F^2}{N} \tag{3.5}$$

where y and x represent the ground truth and degraded images, N represent the number of pixels, f denotes the function represented by the learned neural network, $\|\cdot\|_F$ represents the Frobenius Norm.

To determine the value of α, we first test the our model with $\alpha = 0.05, 0.01, 0.005$ with 1 trial for each value, using the loss function stated in Eq. 3.3. We use the α with the least MSE for the remaining experiments. The plot of the testing loss for different α is in Fig. 3.3(a). To evaluate the performance of the RD-U-Net model, we compared the reconstruction results with GRAPPA (Griswold *et al.*, 2002) and U-Net (Lee *et al.*, 2017). Tab. 3.1 report the results for 5 different runs, and Fig. 3.5 illustrates sample visual results. For the zero-filled reconstruction, there exists a lot of aliasing artifacts (see Fig. 3.4(c)). Although GRAPPA removes the aliasing artifacts, the reconstructed image is still with a lot of noise (Fig.3.5(a)). The U-Net approach is better than GRAPPA, but the result is still lack of details (Fig. 3.5(b)). The proposed RD-U-Net provides a better reconstructed image with the presentation of residual dense block. In Fig.3.5(c), we can see that the result is sharper at the edges, and clearer for the details. The Fourier constraint is also useful for obtaining a better reconstruction, as shown in Fig. 3.3(b), adding the such regularization leads to lower MSE.

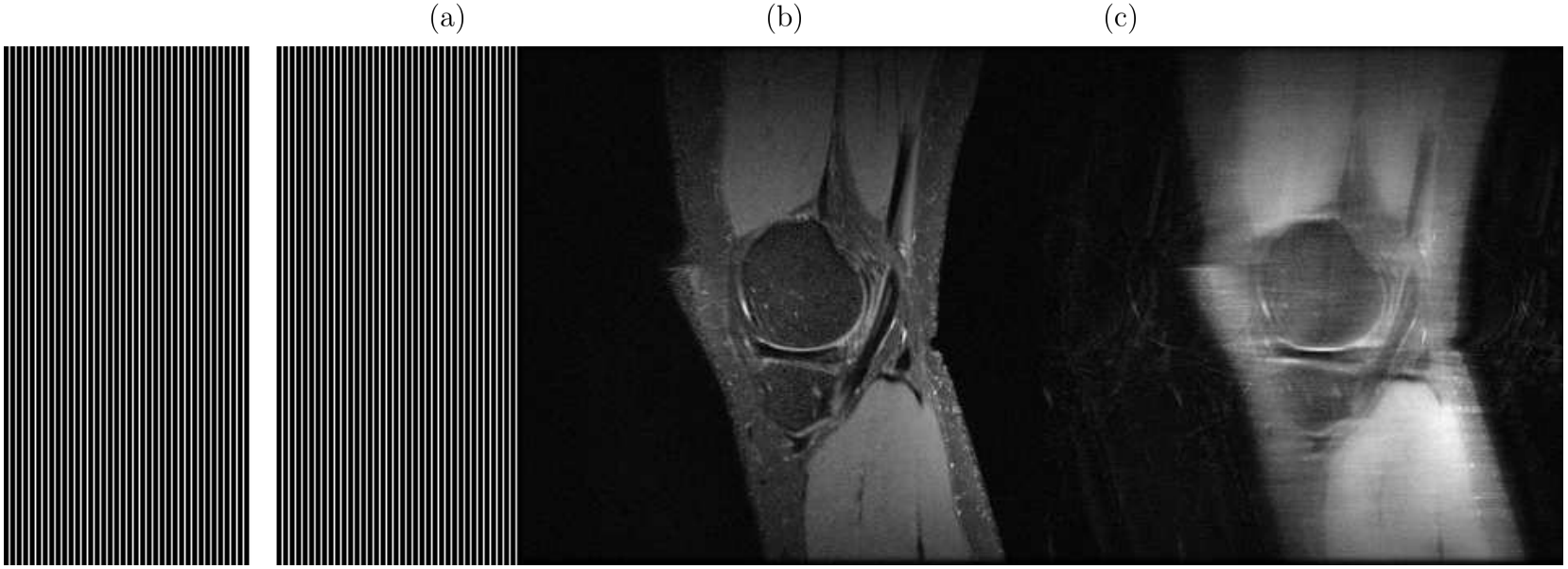


Figure 3.4: The Figures For (A) the Sampling Pattern (4x Accelration, with 16 Acs, 5% of Total Pe); (B) the Reconstructed Image from Fully Sampled K-space Data; (C) the Reconstructed Image from Zero-filled under Sampled K-space Data.

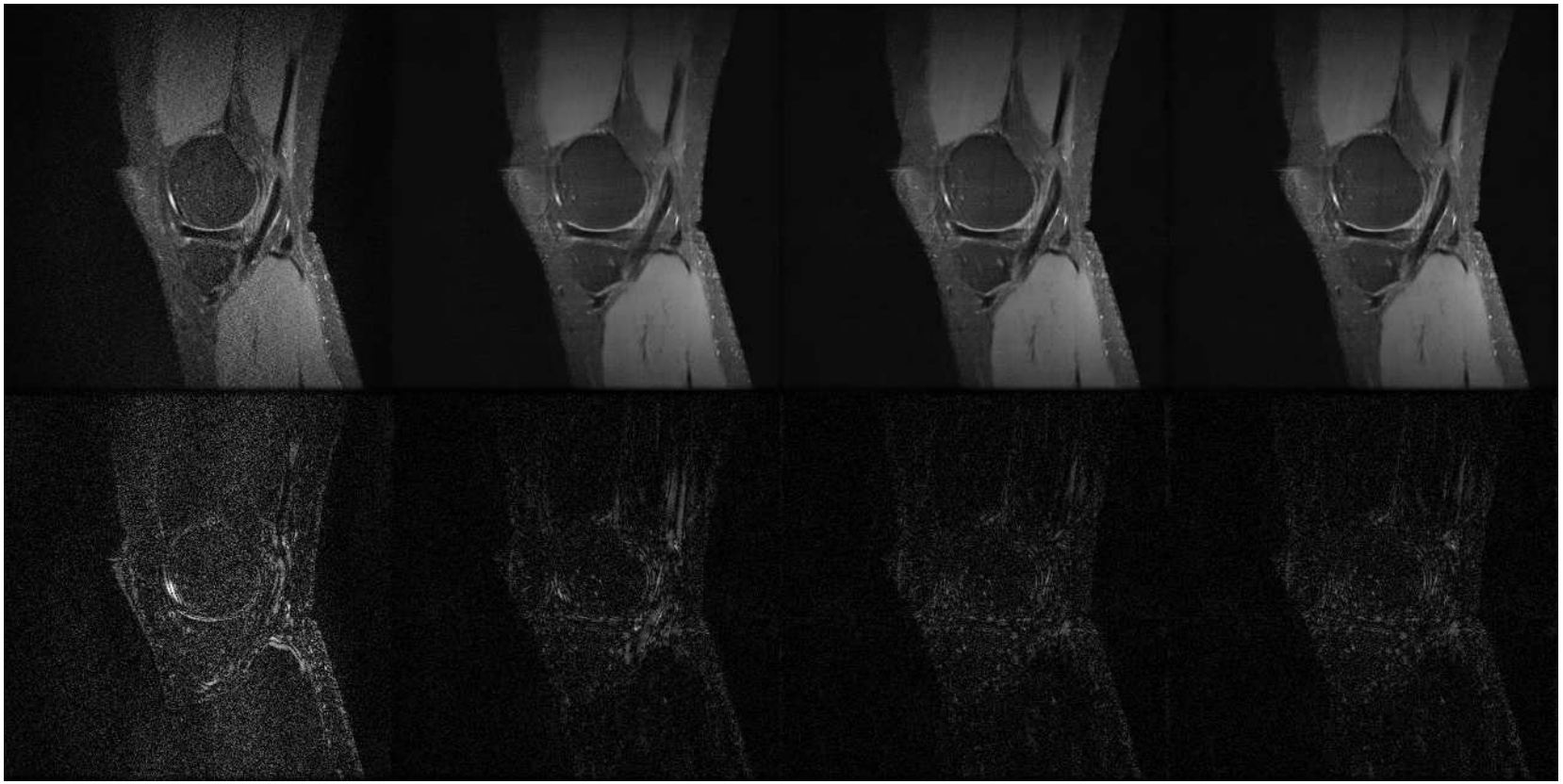


Figure 3.5: The Figures For (A) the Results from Grappa; (B) the Results from U-net (Lee *et al.*, 2017); (C) the Results from RD-U-Net; (D) the Results from RD-U-Net with Fourier Constraint. Row 1: The Reconstructed Images; Row 2: The Difference Between the Reconstructed Images and the Ground Truth.

The network was implemented using Pytorch 0.4.0 with python 3.6.3 on Ubuntu 16.04. All the experiments were performed on a computer with nVidia GTX 1080 GPU and Intel Xeon E5-2603 CPU, although the code was not optimzed to with respect to the particular computer hardware.

The reconstruction time for GRAPPA is about 20 seconds. The training time for the U-Net, RD-U-Net(without/with Fourier constraint) are about 13, 15 and 15 hours respectively. The reconstruction time for all the three compared neural networks are less than 1 second.

3.2 Variational Feedback Network for Accelerated MRI Reconstruction

Magnetic resonance imaging (MRI) is an important diagnostic tool for a lot of diseases. However, compared to other imaging techniques like CT or X-Rays, the scan time of MRI is relatively longer (can sometimes take up to 2 hours), which leads to poor patient experience and expansive cost. In order to improve the situation, it is meaningful to investigate if it is possible to decrease the scan time, while preserving the quality of the reconstructed images.

Parallel Imaging (PI) (Sodickson and Manning, 1997; Pruessmann *et al.*, 1999; Griswold *et al.*, 2002) is one of the most important techniques. Instead of sampling the data points sequentially, PI samples multiple data points at the same time, and combines them together in software. Compressed sensing (CS) (Donoho, 2006) is another important technique, which allows the MRI acquisition to be speeded up by sampling only a subset of k-space data. However, by the Nyquist-Shannon sampling theorem, the reconstruction under the down-sampling scheme contains aliasing artifacts. To address this problem, an approach called Generalized Auto-calibrating Partial Parallel Acquisition (GRAPPA) (Griswold *et al.*, 2002), fills the missing k-space measurements by applying interpolation on the surrounding data from every coils; SAKE (Shin *et al.*, 2014) and ALOHA (Lee *et al.*, 2016) use low-rank matrix completion to approximate the original k-space data. However, these algorithms have high complexity and take significant time to reconstruct the images, making them less practical.

Recently, deep learning is used for many tasks in artificial intelligence, such as image classification (He *et al.*, 2016; Huang *et al.*, 2017), image super resolution (Ding *et al.*, 2017; Yang *et al.*, 2012b), natural language processing (Kumar *et al.*, 2016; Conneau *et al.*, 2016), and video analysis (Feichtenhofer *et al.*, 2019; Wu *et al.*,

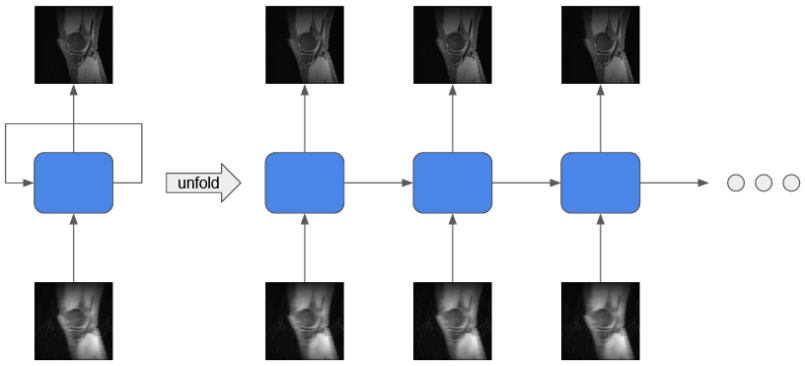


Figure 3.6: The Illustration of Our Feedback Mechanism. Similar to Rnn, The Output of the Network Is Used as an Input to the Network in the next Fold.

2019), and providing promising results. Researchers then started investigation on how to utilize deep learning techniques for accelerated MRI reconstruction. The authors of (Wang *et al.*, 2016) proposed a convolutional neural network (CNN) architecture to map the aliased images to the original high quality reconstruction. In (Kwon *et al.*, 2017), a multilayer perceptron was trained to learn the reconstruction for parallel MRI, and Hammernik et al. (Hammernik *et al.*, 2018) applied a variational network on CS-MRI algorithm.

The superiority of deep learning based approach mainly comes from the nonlinearity capacity of the neural network (Delalleau and Bengio, 2011; Pascanu *et al.*, 2013; Montufar *et al.*, 2014), which allows the network to learn a more complicated down-sampled/fully sampled mapping. In order to increase the complexity of the network, researchers usually increase the number of layers, which also increases the number of parameters. However, having a large network not just requires a lot of storage resources, but also makes the model suffer from the overfitting problem. Recurrent structure is one of the solutions to reduce the number of parameters, as it reuse part of the parameters. It has been shown its effectiveness in some recent studies (Liao and Poggio, 2016; Kim *et al.*, 2016a).

Compared to traditional CNN, another advantage of using RNN structure is that

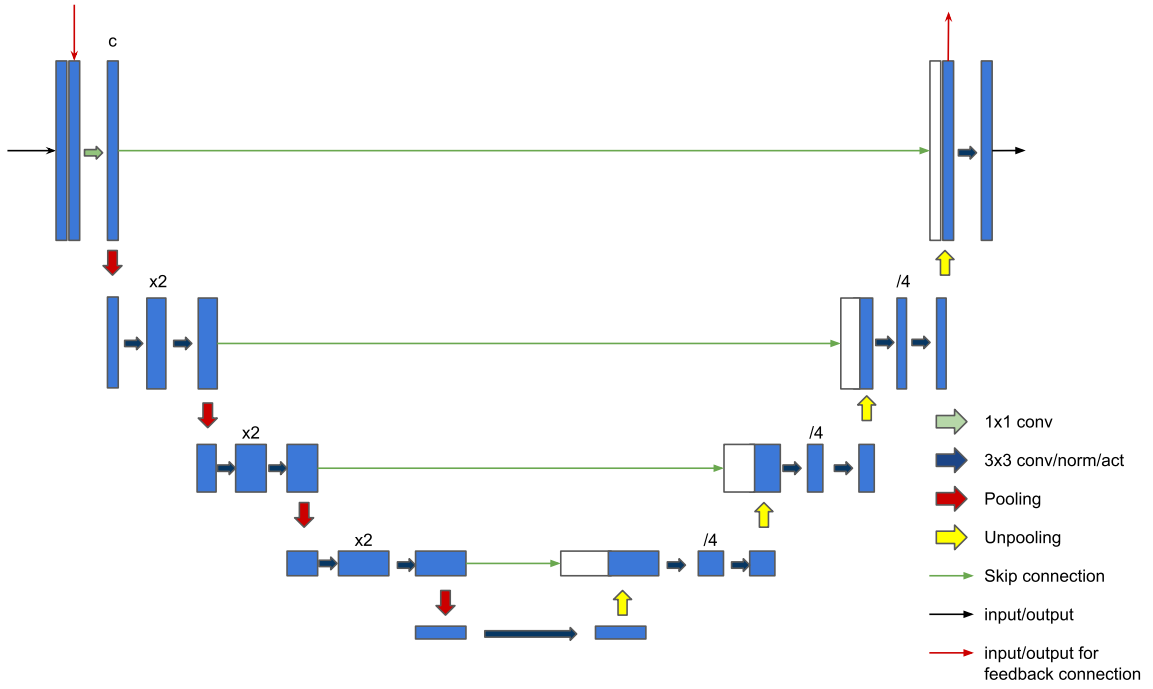


Figure 3.7: An Illustration of Our Feedback Block. In the Figure, Green Thick Arrows Represents 1×1 Convolutional Layers; Blue Thick Arrows Denote 3×3 Convolutional Layers, Each of Them Is Followed by a Normalization Layer and a Nonlinear Activation Layer; Red and Yellow Thick Arrows Represent the Pooling Layers and Unpooling Layers Respectively; The Skip Connections Are Represented by the Green Thin Arrows; The Black Thin Arrows Are the Input/Output for the Feedback Block, While the Red Thin Arrows Represent the Input/Output for the Feedback Connections.

the high level information is able to be passed back to the lower layers to refine the low level features. Theoretical studies in cognition theory (Hupé *et al.*, 1998; Gilbert and Sigman, 2007) state that the response signals can be transmitted from higher order areas to lower order areas by the feedback connections. Some recent works in image processing and computer vision are inspired by this observation (Stollenga *et al.*, 2014; Zamir *et al.*, 2017; Li *et al.*, 2019).

In this paper, we propose a Variational Feedback Network (VFN) for accelerated MRI reconstruction, which is an extension to a previously proposed variational model (Chen *et al.*, 2020) with feedback connections and recurrent structure. In order to achieve the feedback manner, we connect the output of a U-Net-like network to its

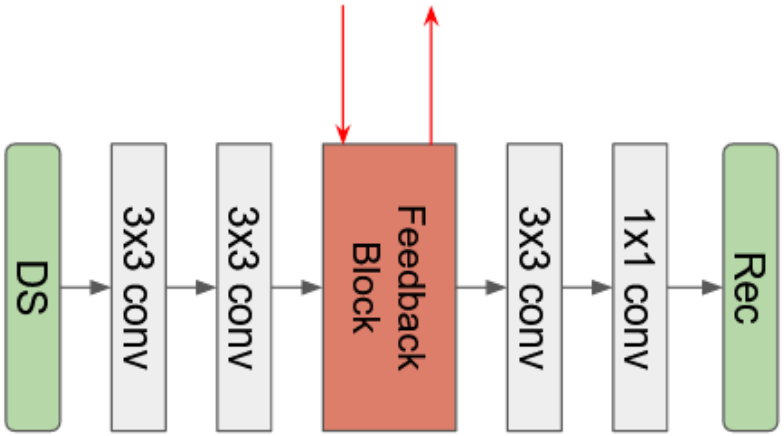


Figure 3.8: The Figure Illustrates Our Feedback Network (FN). The Red Arrows Indicates the Inputs/Outputs for the Feedback Connections. They Are Indicated as Red Thin Arrows in Fig. 3.7. See Fig. 3.7 for the Details of the Feedback Network.

input by using skip connection (see Fig. 3.6). We first conduct analysis on the number of folds of the RNN in our model, followed by the comparisons among different models. Experimental results demonstrate that our proposed model outperforms other leading neural networks for accelerated MRI reconstruction.

3.2.1 Background and Related Works

In this section, we review the background of MRI, followed by related works to our proposed model for accelerated MRI reconstruction.

Accelerated MRI Acquisition

To obtain MR images, MR scanner is used to scan a patient by acquiring measurements. The acquisition is done in the frequency domain, which is also called k-space. The acquired data is called k-space data, and can be used to obtain the images by applying an inverse Fourier transform $\mathcal{F}^{-1}$ to them. The relation between a given k-space data k and the corresponding image can be represented by the following

equation:

$$k = \mathcal{F}(x) \tag{3.6}$$

where $\mathcal{F}$ is an operator denoting Fourier transform.

To accelerate the acquisition of MRI, we can obtain only part of the k-space data. Consider a binary mask matrix M, the under-sampled k-space data $\tilde{k}_i$ for the ith coil can be described as:

$$\tilde{k}_i = M \circ k_i \tag{3.7}$$

where $\circ$ represents the Hadamard product.

To obtain the fully sampled and under-sampled images, the root-sum-of-squares reconstruction approach (Roemer *et al.*, 1990) can be applied to the data in the spatial domain:

$$X = \left(\sum_{i=0}^{n_c} |x_i|^2 \right)^{1/2} \tag{3.8}$$

$$\tilde{X} = \left(\sum_{i=0}^{n_c} |\tilde{x}_i|^2 \right)^{1/2} \tag{3.9}$$

where X and $\tilde{X}$ represent the reconstruction images from the fully sampled and under-sampled k-space data respectively.

Deep Learning for Accelerated MRI Reconstruction

Compressed Sensing MRI is well explored by the researchers in the last decade. A variety of methods have been proposed to help solve the problem. Most of them are regularized least squares approaches, which treat the reconstruction problem as an optimization problem. Nonetheless, these methods suffer from the shortcoming of limited MRI diagnostic information preservation, which is still yet to be overcome. In view of that, researchers then resort to deep learning techniques and investigate ways to apply them on accelerated MRI reconstruction. One of the popular choices

for this task is U-Net (Ronneberger *et al.*, 2015), which is originally proposed for medical image segmentation. Its architecture contains an encoder and a decoder, incorporated with several skip connections. The skip connections link the features from the same levels in the encoder and decoder together. Together with residual learning, the researchers in (Lee *et al.*, 2017) present a U-Net model to learn the residual of the reconstruction. In (Hyun *et al.*, 2018), the authors present a deep learning architecture to handle the reconstruction problem by considering k-space constraint. The authors of (Ding *et al.*, 2019), instead of using skip connection, introduce a residual dense block to refine the features.

A variational approach considering sensitivity of the coil is proposed in (Hammernik *et al.*, 2018). As most of the modern scanners contain multiple receiver coils, and their sensitivity affect the acquisition of the k-space samples, the authors of (Hammernik *et al.*, 2018) use ESPIRiT (Uecker *et al.*, 2014) to estimate the sensitivity maps of the coils and use them for the reconstruction of the MR images. In (Chen *et al.*, 2020), the authors proposed an end-to-end version for the variational network. In their work, the sensitivity maps estimation become part of the network.

3.2.2 Variational Feedback Network for Accelerated MRI Reconstruction

We propose our Variational Feedback Network (VFN) in this section. The basic block of our network is a U-Net-like feedback network, which helps extract high level features. We then use it as part of the recurrent network, which is employed in the variational network.

Feedback Block

We start from the feedback block. Similar to the traditional U-Net, it consists of an encoding part and a decoding part. For the encoder, it takes two inputs, one from

the degraded images after some convolutions, and one from the feedback block in the previous fold. The inputs are then fused together to become a set of features having c channels by a 1×1 convolutional layer. After that, a 2×2 pooling layer with stride $= 2$ is applied. The features are then passed through two 3×3 convolutional layers, each of them is followed by a normalization layer and a nonlinear activation layer. Also, the number of channels is multiplied by two after the first convolution. Once the second activation layer is reached, another pooling layer with stride $= 2$ is applied.

This conv-conv-pool combination is applied several times, depends on the depth of the feedback block. Note that their weights are not shared. At the end of the encoder, a 2×2 pooling layer with stride $= 2$ is applied, followed by a 3×3 convolutional layer, where the number of channels of the output remains unchanged.

For the encoding part, the features are first input to an unpooling layer, and then two 3×3 convolutional layers are applied. Similar to the encoder, each of them is followed by a normalization layer and nonlinear activation layer, and the number of unpool-conv-conv combination is the same as the number of conv-conv-pool combination in the encoder. There are skip connections from the corresponding level in the encoding part, and the first convolution in every level decreases the number of channel to 1/4. An unpooling layer is applied to the feature at the end of the decoder, and is output as one of the input for the feedback block in the next fold. Also, it is convolved with the skip connection that comes from the encoder by a 3×3 convolutional layer, followed by a normalization layer and activation layer. Fig. 3.7 is an illustration for the feedback block.

Feedback Network

As illustrated in Fig.3.8, the subnetwork can be divided into three parts: feature extraction, feedback block, and reconstruction. The first part contains two 3×3 convolutional layers, A normalization layer and an activation layer are applied instantly right after each convolutional layer. The output of this part in the t-th fold can be expressed as:

$$x_{in}^{t} = f_{FE}(I_{DS}) \tag{3.10}$$

where I_{DS} is the down-sampled data.

The second part is the feedback block, it takes two inputs (one is optional), one is the output from the feature extraction part in the same fold, and one is the output from the feedback block from the previous fold, which is optional. If there is no input from the previous fold, the 1×1 convolutional layer will no be applied. The structure of the feedback block is described in Sec. 3.2.2. There are two outputs for the this part. Mathematically, they can be described as:

$$[x_{out}^{t}, F^{t}] = f_{FB}(x_{in}^{t}, F^{t-1}) \tag{3.11}$$

where F^{t} is the output for the feedback connection in the t-th fold (*i.e.* it is used as an input for the feedback block in the next fold).

The last part is about reconstruction. The input is convolved by a 3×3 convolutional layer. After that, the output is normalized and passed through a nonlinear activation function, followed by a 1×1 convolutional layer. Together with the skip connection, the mathematical formulation for this part is:

$$I_{REC} = f_{R}(x_{out}) + I_{DS} \tag{3.12}$$

We name this model Feedback Network (FN).

Variational Network

We employ the variational network structure in (Chen *et al.*, 2020) by using our feedback mechnism, and call it Variational Feedback Network (VFN). This structure introduces the estimation of sensitivity maps to help refining the reconstruction. For our proposed FN, there are multiple inputs and outputs. To fit it into the refinement modules, we use the same image for all the inputs in the same refinement module. Only the last refinement module returns all the outputs of the FN, the remaining modules only return the one in the last fold ($t = T$). The illustration of the modified network is shown in Fig. 3.9,

Network settings

We want to maximize the Structural Similarity (Wang *et al.*, 2003) between the reconstruction and the ground truth. Therefore, we want to minimize the following value:

$$L(x, x^*) = -SSIM(x, x^*) \tag{3.13}$$

where $SSIM$ is the Structural Similarity Index, x and x^* represent the reconstruction and the ground truth respectively. As there are T outputs for our model, we use the following as our loss function:

$$L(\{x^t\}_{t=1}^T, x^*) = -\frac{1}{T}\sum_{t=1}^{T} SSIM(x^t, x^*) \tag{3.14}$$

where $x^1, ..., x^T$ are the outputs of our model. We use x^T as our final output.

We use instance normalization (Ulyanov *et al.*, 2016) and leakly ReLU (Maas *et al.*, 2013) for the normalization layer and activation layer respectively. For the feedback block, max pooling and bilinear interpolation are used for pooling and unpooling, number of channel of the features after the first convolution (denoted by c) is set

	T=1	T=2	T=3
NMSE	0.0373	0.0369	0.0367
SSIM	0.7280	0.7305	0.7311
PSNR	31.4851	31.5625	31.5931

Table 3.2: The Normalised Mean Square Error (NMSE), Structural Similarity Index Measure (SSIM), Peak Signal-to-noise Ratio (PSNR) of the VFN Model for $T = 1, 2, 3$.

	U-Net	E2EVN	VFN
NMSE	0.0387	0.0373	0.0369
SSIM	0.7244	0.7280	0.7305
PSNR	31.1783	31.4851	31.5625

Table 3.3: The Normalised Mean Square Error (NMSE), structural Similarity Index Measure (SSIM), Peak Signal-to-Noise Ratio (PSNR) for U-Net, E2EVN and Our Proposed VFN on 4x Acceleration.

to 18. The depth of the block is 4 (*i.e.* there are 4 pooling layers and 4 unpooling layers). Two cascades are used for the variational network.

3.2.3 Experiments

In this section, we first introduce the dataset we use. After that, we describe the network settings, followed by the comparative studies.

Magnetic Resonance Dataset

The single coil knee datasets from fastMRI are used to evaluate our proposed VFN. For the experiments described in this section, we run 5 trails for each them, every time we randomly pick 10 dataset for training and the other 10 for testing. To obtain the under-sampled data, the original fully-sampled data are retrospectively down-sampled by 4 times with 25 ACS(auto-calibration signal, about 7.8% of total PE).

Analysis of the Number of Folds

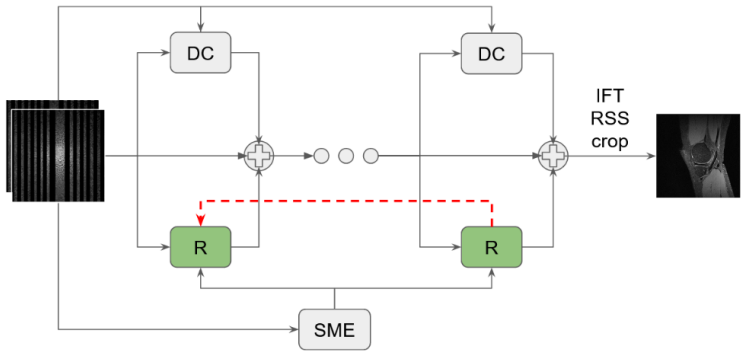
In this subsection, we analyze the impact of the number of folds of the subnetwork in our model by comparing the the final output of FN for $T = 1, 2, 3$. Tab. 3.2 shows that, the models with feedback connections ($T > 1$) are better than the one without feedback connection ($T = 1$). In addition, we can observe that, when T is increasing, the performance is also becoming better. This suggests that this feedback-RNN structure is beneficial to the performance.

Experimental Settings and Results

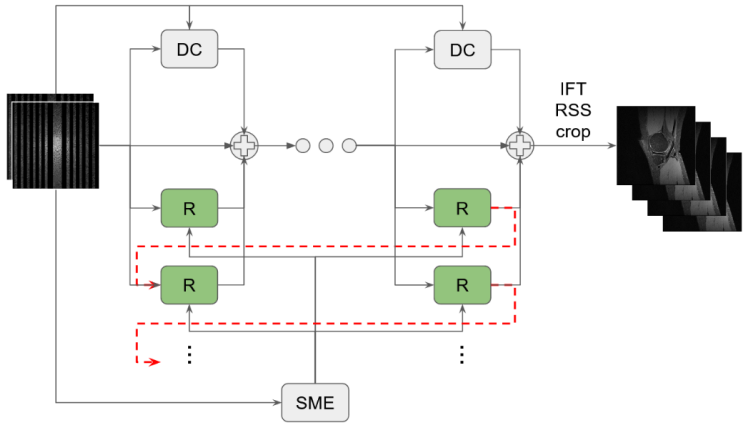
We compare our VFN model with U-Net (Ronneberger *et al.*, 2015), E2EVN (Chen *et al.*, 2020). We train the networks for 50 epochs with batch size = 8. Adam algorithm (Kingma and Ba, 2014) is used for with initial learning rate = 0.0001. The learning rate is multiplied by 0.1 at the $40th$ epoch. The number of folds T is set to 2.

Tab. 3.3 shows the results from different models under different metrics.

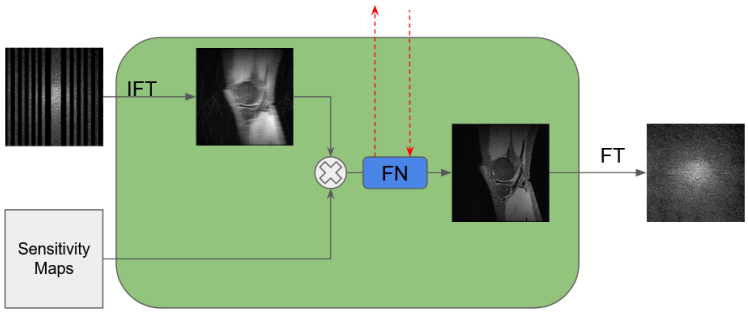
The network was implemented using Pytorch 1.5 (Paszke *et al.*, 2017) with python 3.7 on Ubuntu 18.04. A computer with nVidia GTX 1080GPU and Intel Xeon E5-2603 CPU is used for the experiments.



(a)



(b)



(c)

Figure 3.9: The Figure Shows (a) Our Proposed Network. Similar to (Chen *et al.*, 2020), DC, R, SME Represent the Data Consistency, Refinement and the Sensitivity Map Estimation Modules Respectively; (b) the Unfolded View of (a); (c) Our Modified Refinement Module. Fourier Transform, Inverse Fourier Transform and Our Proposed Feedback Network Are Denoted by IFT, FT and FN Respectively.

Chapter 4

TRANSFER LEARNING FOR ACCELERATED MRI

Magnetic resonance imaging (MRI) reconstruction is an important technique for medical imaging that has been used in the diagnosis of a wide range of diseases. To obtain high quality MRI images, fully sampled MRI data is required, which usually demands a long scan time (easily exceeding 30 minutes). This leads to high cost and poor patient experience. Many efforts have been done to cut the time of MRI data acquisition through both hardware and software developments. Examples include parallel imaging (Sodickson and Manning, 1997) and compressed sensing (CS) (Donoho, 2006; Lustig *et al.*, 2008). It is worth to mention that, among these approaches, CS techniques were a breakthrough in the reduction of the MR acquisition time. Comparing to other approaches, it speeds up the scanning by acquiring less measurement data to reconstruct high quality images. By the Nyquist-Shannon sampling theorem, such undersampling scheme leads to aliasing artifacts. To eliminate the artifacts, researchers incorporated additional prior knowledge during the reconstruction. Representative approaches include generalized auto-calibrating partial parallel acquisition (GRAPPA) (Griswold *et al.*, 2002), SAKE (Shin *et al.*, 2014), and the annihilating filter based low-rank Hankel matrix approach (ALOHA) (Lee *et al.*, 2016).

Recently, deep learning has become one of the most popular tools for machine learning research, achieving promising result in image classification (He *et al.*, 2016; Huang *et al.*, 2017), natural language processing (Kumar *et al.*, 2016; Conneau *et al.*, 2016), image super resolution (Tong *et al.*, 2017; Shocher *et al.*, 2018), and style transfer (Gatys *et al.*, 2016), etc. Researchers also started to investigate the use of deep learning to make MRI scans faster. In (Wang *et al.*, 2016), a convolutional neural

network (CNN) is deployed to map the aliased image to the fully sampled image. In (Kwon *et al.*, 2017), a multilayer perceptron is learned for parallel MRI, and in (Lee *et al.*, 2017; Hyun *et al.*, 2018; Ding *et al.*, 2019), the researchers applied UNet (Ronneberger *et al.*, 2015) or its variants to achieve accelerated MRI reconstruction.

Since deep networks require a lot of data to train, transfer learning is often employed to improve the performance when only limited data are available in a target domain. This technique has seen applications in image classification (Ganin and Lempitsky, 2015; Ganin *et al.*, 2016) and computer-aided diagnosis (Maqsood *et al.*, 2019; Byra *et al.*, 2020), etc., where the focus has been on improving the network architecture. Given typical scarcity of labeled data in the medical domain and the abundance of natural images, an interesting transfer learning problem is to consider how to to improve deep learning approaches for a medical imaging task through utilization of natural images like ImageNet (Russakovsky *et al.*, 2015). While some recent efforts have studied this problem to some extent (Raghu *et al.*, 2019; Zhu *et al.*, 2018; Knoll *et al.*, 2019), there is not much direct research on this for accelerated MRI reconstruction. In this paper we design research and experiments to answer the following question: is it feasible and beneficial to transfer knowledge learned from natural images for improving deep-learning-based MRI reconstruction?

Following typical protocols in current practice, the procedure of applying transfer learning on accelerated MRI reconstruction would be as follows. First, we train a neural network using a large dataset of natural images, optimizing the performance by capitalizing the size of the dataset. Then, we fine-tune the network through retraining with the target data (typically of a much smaller size). To achieve these steps in this study, we design a way to down-sample the natural images in the sense of k-space degradation. This enables us to perform systematic evaluation and comparison of the performance using an exemplar architecture based on the UNet, which has been

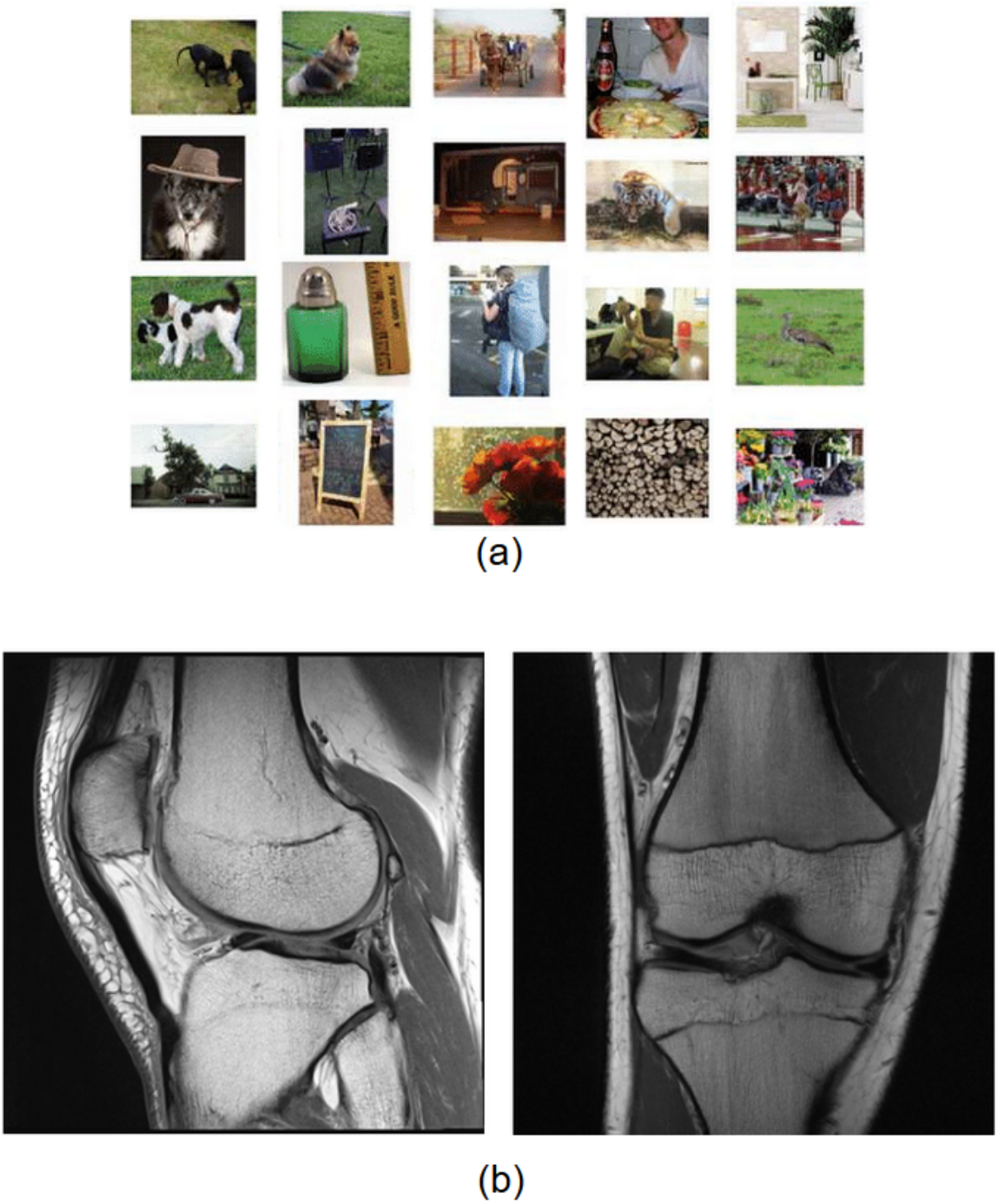


Figure 4.1: Example Images from (a) ImageNet Dataset; (b) FastMRI Dataset (Knee).

frequently used for MRI reconstruction. We analyze the weights of the network by visualizing them (Erhan *et al.*, 2009), and compare their similarity between the model before and after fine-tuning by CCA (Raghu *et al.*, 2017). While the experimental results suggest that the above typical transfer learning protocol does not lead to obvious performance gain, our analysis provides some insights as to why it is the case and hence points to potential future directions for improved transfer learning for this domain.

4.1 Related Works

In this section, we review the works which are related to our investigation. It is worth to mention that there is a lot of works performing state-of-the-art on super-resolution in natural images. However, such work is not that related to the accelerated MRI reconstruction as the degradation models are different, and it is clear that the low resolution images in the problem we are facing involve structured aliasing.

4.1.1 Deep Learning for Accelerated MRI Reconstruction

Over the past decade, researchers have explored the field of Compressive Sensing MRI deeply. Some works have been done upon the regularized least-squares approaches, which are set of model-fitting methods aiming to reconstruct MRI image by introducing a regularization term. However, when using these approaches, it is difficult to overcome the limitations in preserving diagnostic information in MRI. As a result, researchers investigate how to apply deep learning techniques to reconstruct the under-sampled MRI. UNet (Ronneberger *et al.*, 2015), which is first proposed for medical image segmentation, has soon become a popular choice in this task. In (Lee *et al.*, 2017), researchers present a UNet model the residual learning. The network estimates aliasing artifacts from distorted images of undersampled data. The authors

of (Hyun *et al.*, 2018), present a deep learning framework by imposing k-space constraint to tackle the reconstruction problem. In (Ding *et al.*, 2019), in order to extract better features, a new component which called residual dense block is used to replace the skip connection in UNet.

4.1.2 Transfer Learning

Transfer learning is a machine learning technique, which is usually used when the data in the target domain has little data to train. Depends on the goal, there are different ways to perform transfer learning. For most of the medical imaging task, a popular standard of transfer learning is to train a network using a natural image dataset, and then fine-tune it on the medical image dataset. Transfer learning technique has been employed to several medical applications like chest x-rays (Wang *et al.*, 2017c; Rajpurkar *et al.*, 2017), and ophthalmology (Abràmoff *et al.*, 2016; Gulshan *et al.*, 2016) A recent paper (Raghu *et al.*, 2019) investigates the possibility of transferring learned knowledge from natural images to medical images. However, most of the previous works are evaluated only on the image classification task, it is not clear that if such conclusions can also be drawn on image reconstruction task, as they are very different.

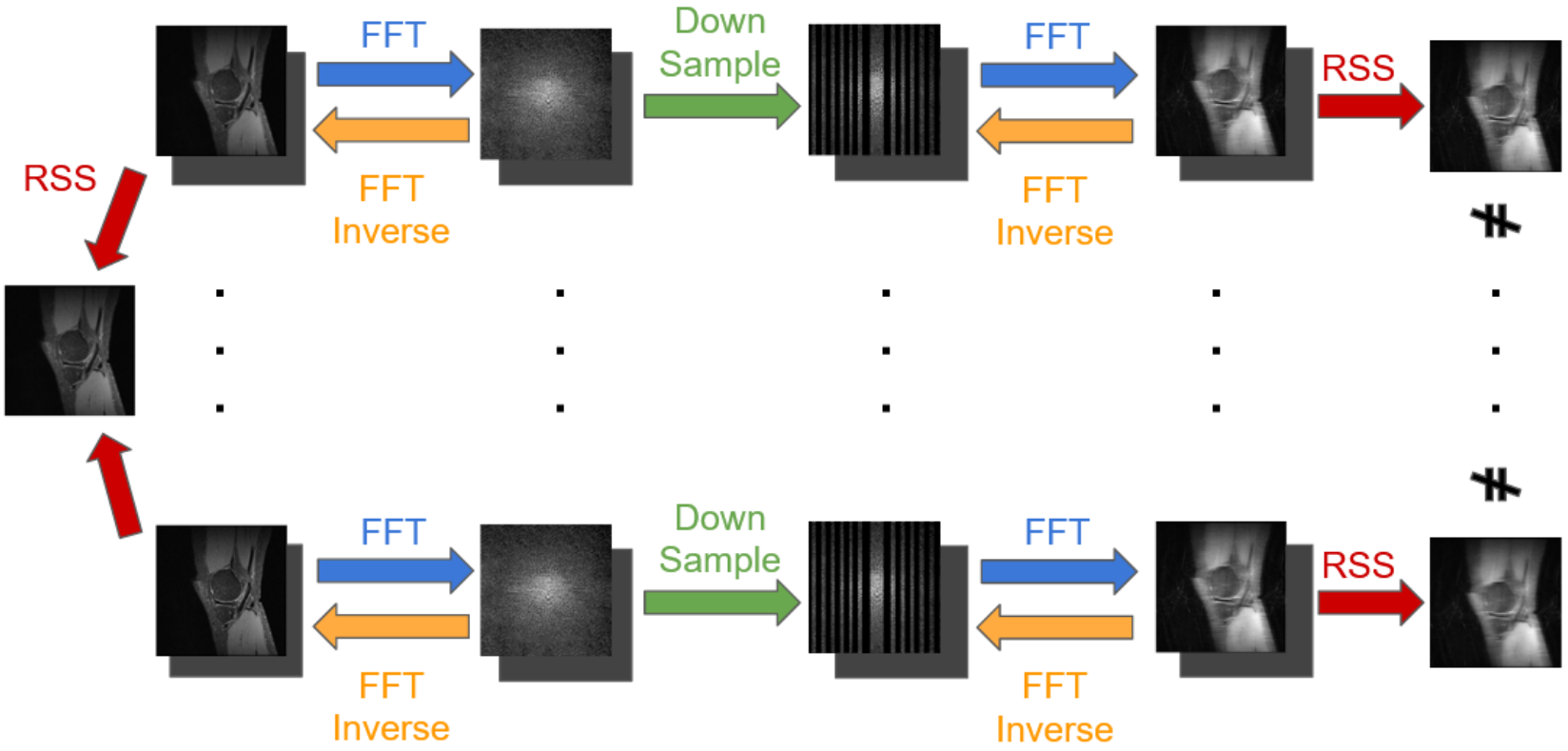


Figure 4.2: The Figure Illustrates That There Are a Lot of Different Ways to Construct a Low Resolution Version of an Image (in the Sense of K-space Degradation). The Images Except the Leftmost and Rightmost Column Have Two Channels. The Second Channel (Gray Square) Represents the Phase Part, While the First Channel Represents the Magnitude Part. For the Data in the Second Column, The Root-sum-of-squares (RSS) Are the Same, As Their Magnitude Are Equal. However, The Magnitude of Their Corresponding Down-sampled Data (the 5th Column) Are Different, Hence after Taking the RSS, The Images Are Different.

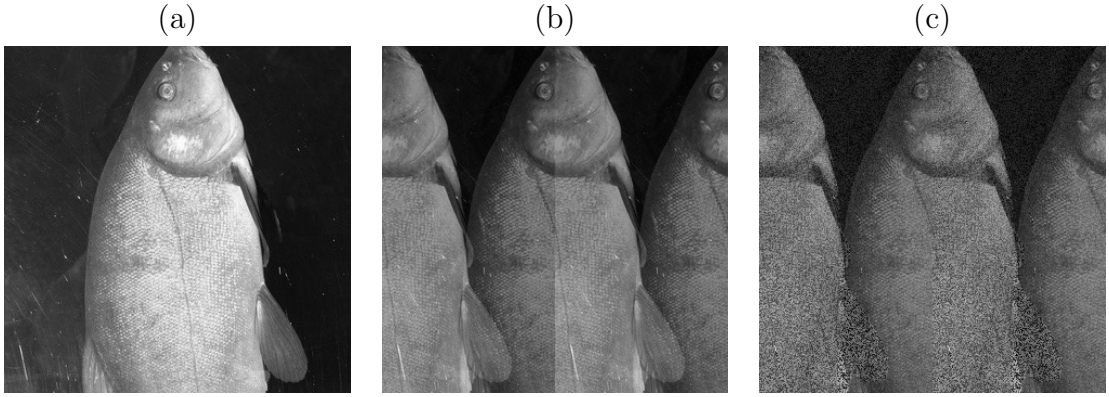


Figure 4.3: The Figures Illustrate the Effect of Using Different Phase Part for Undersampling. (A) Is the Original Image, (B) and (C) Are the Degraded Images under 2x Acceleration While Using Different Phase Part. We Can Observe That Although the Images Are Different, The Aliasing Effect Is Still Preserved.

4.2 Evaluating Transfer Learning for Accelerated MRI Reconstruction

We first introduce the datasets, and then present a scheme for making natural images (which are in general not from the k-space as MRI images do) useful for our reconstruction task, followed by the experiments and results.

4.2.1 Datasets

For the source of natural images, we use the ImageNet classification dataset (Russakovsky *et al.*, 2015). The dataset contains 1000 classes, and more than 1.2 million images. For the MRI data, we use the knee single-coil k-space data in the fastMRI dataset (Zbontar *et al.*, 2018). The dataset contains 973 volumes, in total 34,742 slices in the training set, and 199 volumes, in total 7,135 slices in the validation set. Fig.4.1 shows some examples from both datasets.

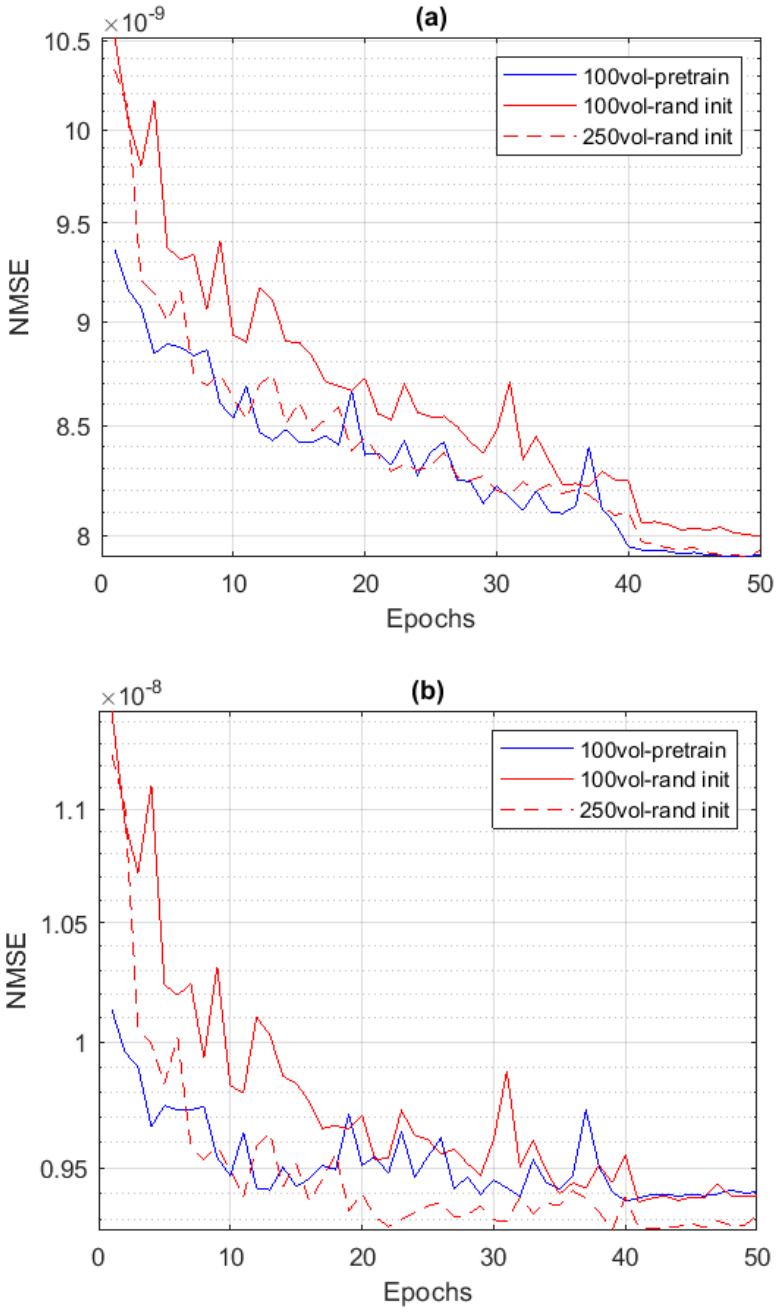


Figure 4.4: The Nmse of the (a) Training Set; (b) Testing Set During the Training Stage. We Compare 3 Settings: (1) Pretrained on Imagenet, Fine-tune on 100 Volumes of MRI Data; (2) Training on 100 Volumes of MRI Data from Scratch; (3) Training on 250 Volumes of MRI Data from Scratch.

4.2.2 Utilizing Natural Images

To create the down-sampled k-space data, under-sampling is performed by retrospectively setting part of the columns (phase encoding direction) of the k-space data to zero, from a fully sampled acquisition. To generate the under-sampling masks, we first keep a portion of k-space columns, which represent the lowest frequency of the corresponding signal. After that, the remaining columns are sampled equidistant. As 4x acceleration factor is used in our case, we only keep 1 out of every 4 columns. The down-sampled k-space data is obtained by simply applying the mask on the fully acquired k-space data. That is,

$$\tilde{y_i} = M \circ y_i \tag{4.1}$$

where $\circ$ represents the Hadamard product, y_i and $\tilde{y_i}$ represent the ith coil of the original and down-sampled k-space data respectively, for $i \in \{1, 2, ..., n_c\}$, n_c is the number of coils.

When both fully acquired and down-sampled k-space data are obtained, the root-sum-of-squares reconstruction method (Roemer *et al.*, 1990), which is one of the most popular coil combination methods, can be used to visualize the images. We first apply the inverse Fourier Transform to the k-space data on each coil,

$$m_i = \mathcal{F}^{-1}(y_i) \tag{4.2}$$

$$\tilde{m}_i = \mathcal{F}^{-1}(\tilde{y}_i) \tag{4.3}$$

where $\mathcal{F}$ is the function representing Fourier Transform, m_i and $\tilde{m}_i$ denote the fully acquired and down-sampled image representation of the ith coil respectively.

At this step, m_i and $\tilde{m}_i$ are still in complex domain. To visualize the corresponding

final images the root-sum-of-squares image can be calculated as:

$$x = \left(\sum_{i=0}^{n_c} |m_i|^2 \right)^{1/2} \tag{4.4}$$

$$\tilde{x} = \left(\sum_{i=0}^{n_c} |\tilde{m}_i|^2 \right)^{1/2} \tag{4.5}$$

For our case, since we are using single-coil data, we have $n_c = 1$, which leads to

$$x = |m_1| \tag{4.6}$$

$$\tilde{x} = |\tilde{m}_1| \tag{4.7}$$

For the natural images in the ImageNet dataset, usually they have 3 channels (for RGB images). We convert all the RGB images to grayscale by using the '*convert('LA')*' command in the python package Pillow. In order to down-sample the natural images, we create the phase part for them as it is required in the Fourier/inverse Fourier Transform. The simplest way is to fill the phase part with zero. By Eq.4.4, no matter what we choose, we can obtain the original natural images from these emulated k-space data. However, the down-sampled images obtained by using different phase part can be different. Mathematically, for an image x, we can always find $m^{(1)}$ and $m^{(2)}$ in complex domain such that

$$m^{(1)} \neq m^{(2)} \tag{4.8}$$

$$x = \left(\sum_{i=0}^{n_c} |m_i^{(1)}|^2 \right)^{1/2} = \left(\sum_{i=0}^{n_c} |m_i^{(2)}|^2 \right)^{1/2} \tag{4.9}$$

and

$$\begin{aligned} &\left(\sum_{i=0}^{n_c} |\mathcal{F}^{-1}(M \circ \mathcal{F}(m_i^{(1)}))|^2 \right)^{1/2} \\ \neq &\left(\sum_{i=0}^{n_c} |\mathcal{F}^{-1}(M \circ \mathcal{F}(m_i^{(2)}))|^2 \right)^{1/2} \end{aligned} \tag{4.10}$$

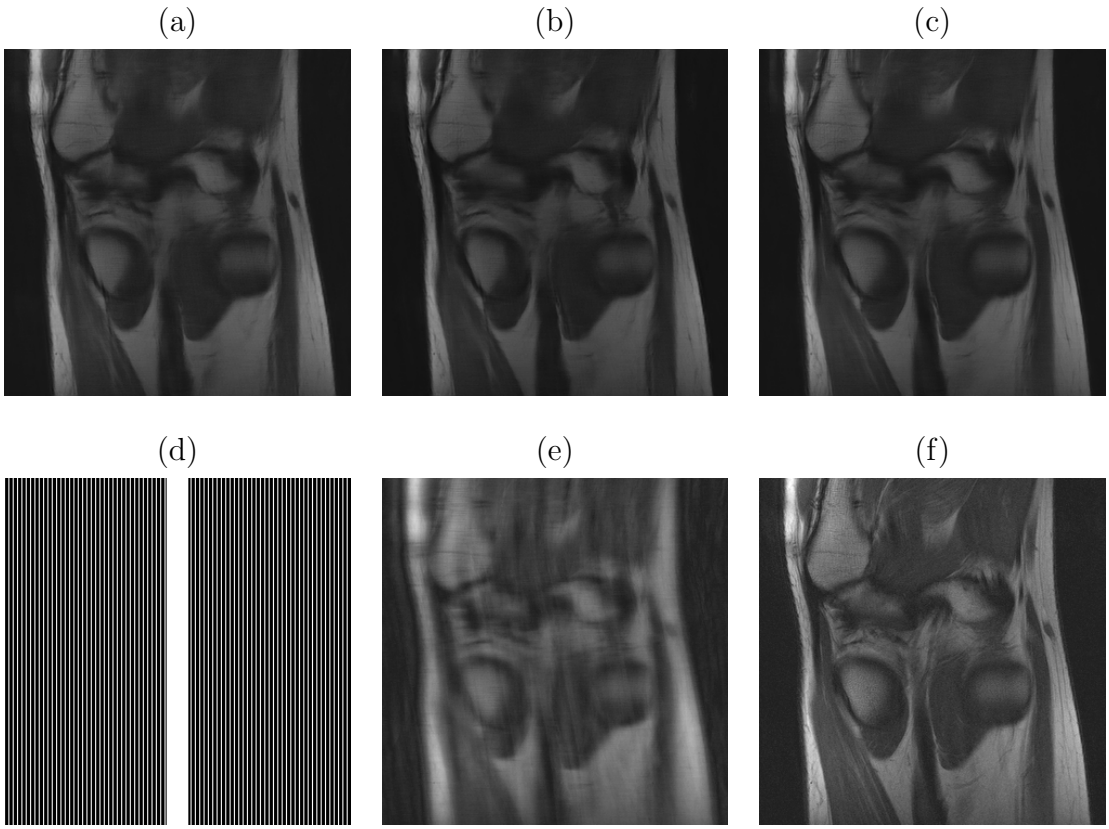


Figure 4.5: The Figure Shows (a)-(c) the Reconstructed Image by Different Model. (a) Pretrained on Imagenet, Fine-tune on 100 Volumes of MRI Data; (b) Training on 100 Volumes of MRI Data from Scratch; (c) Training on 250 Volumes of MRI Data from Scratch. For the Remaining Sub-figures, (d) the Mask We Used for Down-sampling K-space Data; (e) the Degraded Image; (f) the Ground Truth.

Due to this reason, there are a lot of way to do the under-sampling. Fortunately, as illustrated in Fig.4.3, the degraded images obtained by using different are not dramatically different. In our case, we randomly choose the phase part from the real MRI data, and use them as the phase part for the natural images. Fig.4.2 illustrates why they are different.

All data (for both MRI data and natural images) are cropped to 320×320, the input (low resolution) and output (high resolution) are normalized to have zero mean and unit variance. which means, we apply the following transformation for every

100vol-pretrain	100vol-rand init	250vol-rand init
9.3954×10^{-9}	9.3879×10^{-9}	9.3156×10^{-9}

Table 4.1: The Final Nmse of Loss of Different Settings: (1) Pretrained on Imagenet, Fine-tuned on 100 Volumes of MRI Data; (2) Training on 100 Volumes of MRI Data from Scratch; (3) Training on 250 Volumes of MRI Data from Scratch.

single image x:

$$x \leftarrow \frac{x - mean(x)}{std(x)} \tag{4.11}$$

The model settings and the evaluation metric will be discussed in the next section.

4.2.3 Experimental Results

To compare the feasibility of the transfer learning on accelerated MRI reconstruction, we evaluate the performance in the following two settings: (1) training from random initialization; (2) transfer learning from ImageNet. We compare the result by comparing the reconstruction error, and the visualization of the layer features.

We use the UNet architecture in (Zbontar *et al.*, 2018) to train a mapping from low resolution image to the residual. The high resolution image can be computed by adding up the input and the output of the network. This kind of residual learning architecture has been used in the past(Lee *et al.*, 2017; Ding *et al.*, 2019), and provides state-of-the-art result. An illustration of the network can be found in Fig.4.8(a). Number of epochs is 50 with batch size = 8. SGD is used with initial learning rate = 0.001. At the $40th$ epoch the learning rate is decreased to 0.0001. For calculating the error, we use the normalized mean square error (NMSE) as our metric of measurement, which can be defined as

$$NMSE(x, \hat{x}) = \frac{\|x - \hat{x}\|^2}{\|x\|^2} \tag{4.12}$$

where x is a reference image, $\hat{x}$ is a reconstructed image, and $\| \cdot \|^2$ is the squared Euclidean norm.

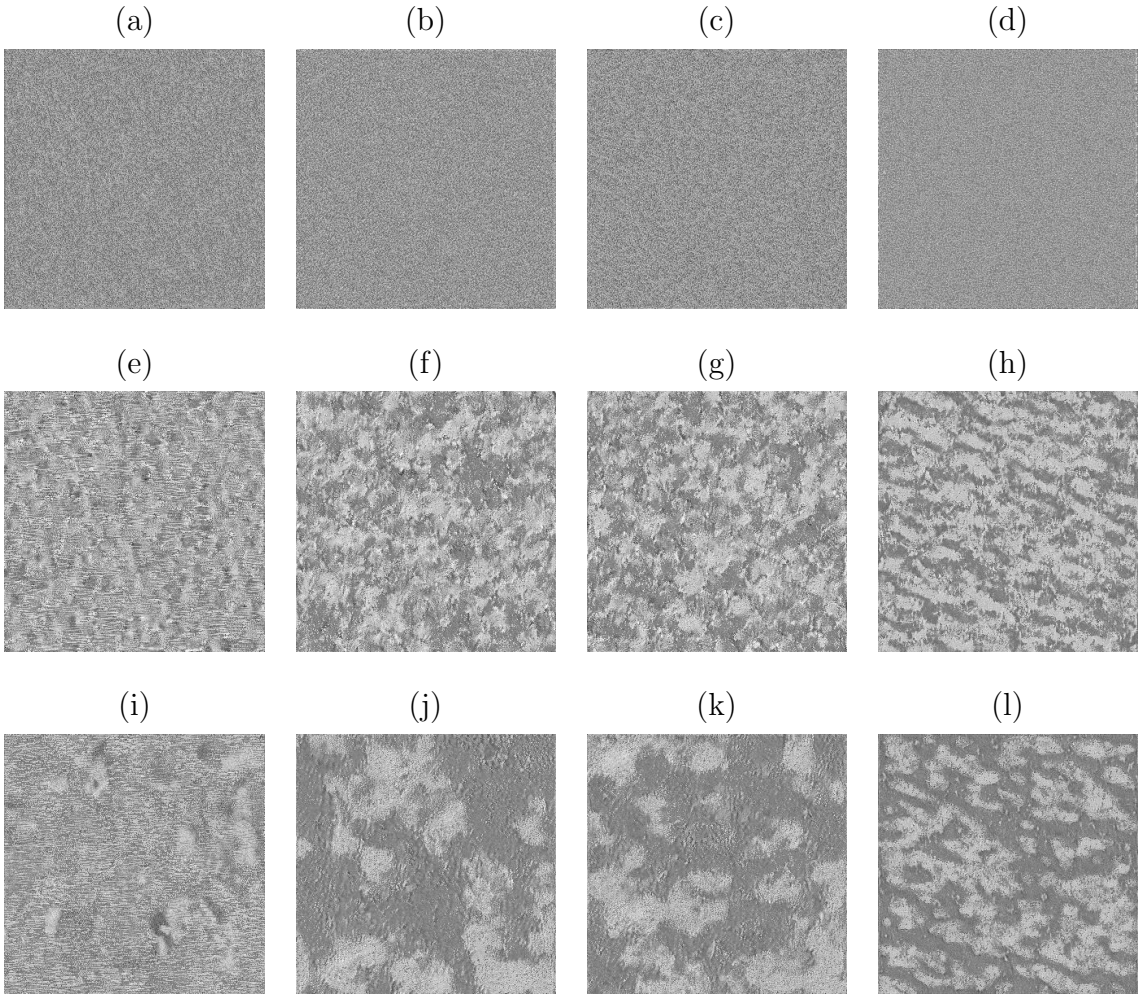


Figure 4.6: The Visualization of the Layer Features. The 1st, 2nd, 3rd and 4th Column Represent the Results from (1) Training on Imagenet; (2) Pretrained on Imagenet, Fine-tuned on 100 Volumes of MRI Data; (3) Training on Another 100 Volumes of MRI Data from Scratch; (4) Training on 250 Volumes of MRI Data from Scratch. The Images in the First Row, Second Row and Third Row Are Generated from a Channel of the (I) Input of the First Pooling Layer; (Ii) Output of the Encoding Part (the Output of the First Convolution Layer after the Last Pooling Layer) And; (Iii) Output of the Decoding Part (the Output of the First Convolution Layer after the Last Unpooling Layer) of the Unet.

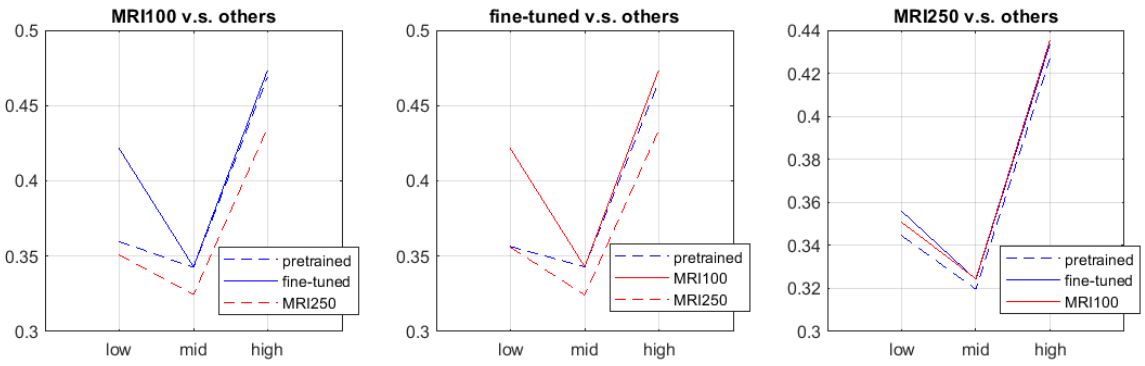


Figure 4.7: The CCA Similarities among Different Models on the Accelerated MRI Reconstruction Task. (Left) Pretrained on Imagenet + Fine-tune on 100 Volumes of MRI Data Vs Others; (Middle) Training on Another 100 Volumes of MRI Data from Scratch Vs Others; (Right) Training on 250 Volumes of MRI Data from Scratch Vs Others.

We compare the performance of using the following 3 settings:

1. pretrained on ImageNet, fine-tune on 100 volumes of MRI data;

2. training on another 100 volumes of MRI data from scratch;

3. training on 250 volumes of MRI data from scratch.

Fig.4.4 shows the plots of the NMSE of the training and validation set during the training process, and Tab.4.1 shows the final NMSE. It is natural to see that the result for setting (3) is obtaining better result than the one from setting (2), it proves that there is still some room to improve the result of setting (2) If the pretraining on ImageNet helps, the performance of setting (1) should be better than the one of (2) (or even better than (3)). However, the result shows that there is no significant improvement while transfer learning technique is used. Fig.4.5 shows some outputs from different models.

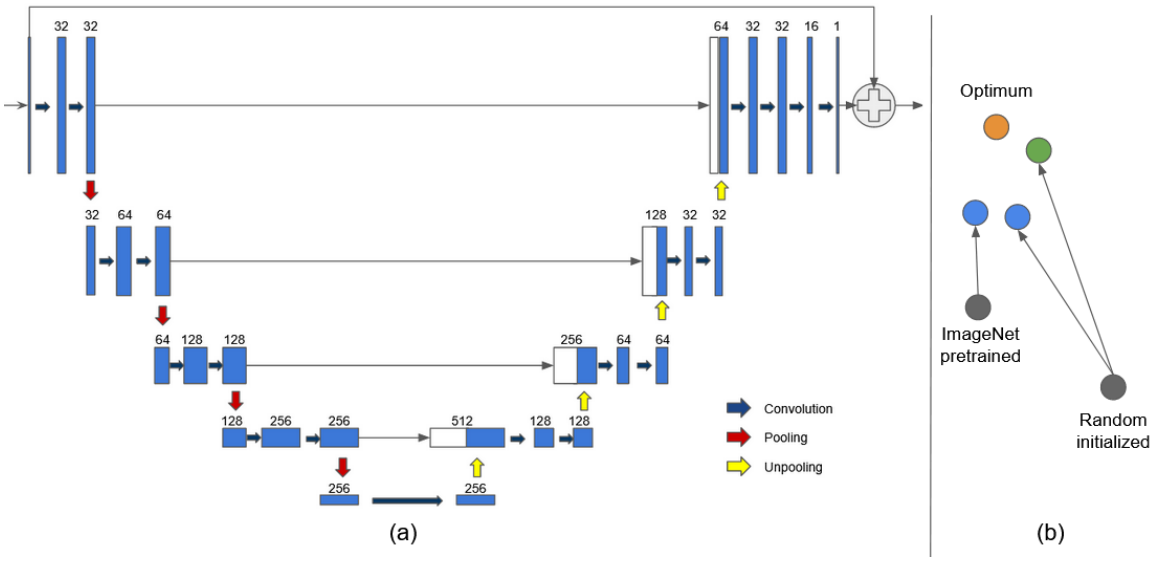


Figure 4.8: The Figure Shows (a) the Unet Architecture Used in the Experiment; (b) the Diagram of Our Intuition for Optimization. The Blue Nodes Represent the Weights That Are Trained/Fine-tuned by 100 Volumes of MRI Data, While the Green Node Represents the Weights That Is Trained by 250 Volumes of MRI Data. The Orange Node Is the Optimal Weights of the Network, The Closer to It, the Lower the Reconstruction Error.

4.3 Analysis of the Effectiveness of Transfer Learning

In Sec.4.2.3 the result shows that there are limited differences between training from scratch and fine-tuning from a pretrained model. This phenomenon is counter intuitive as it suggests that the knowledge learned from the dataset for pretraining does limited help. In general, pretraining provides us with a model with better "starting point", as the model has already gained some knowledge. In this case, it is likely that the model would converge to some point which is closer to the global optimum.

In this section, we try to explain this phenomenon by analyzing the learned weights, which serve as the representation of the neural network. CCA (Raghu *et al.*, 2017) and the technique of visualizing the weights of the layer(Erhan *et al.*, 2009) are used for comparing the filters of the networks in different training settings.

4.3.1 *Visualization of the Layer Features of the Network*

We investigate the layer features by using the technique proposed in (Erhan *et al.*, 2009). It allows us to visualize the representation of a selected filter in a specific layer. To be precise, for a network with fixed weight, we select a channel of the output of a specific layer, and plot the input (of the network) which maximize the average of the elements of this channel. This can be done by using gradient descent to update the input. Fig.4.6 shows the layer features at the output of a lower layer (the 1st row), a middle layer (the 2nd row) and a higher layer (the 3rd row).

For the images in the second row and the third row, we can see that, the images obtained from setting (1) and (2) (*i.e.* the second and the third column of the figure) are similar, which means that no matter what the initial weights of the network are, they become similar after some epochs of training. In other words, the knowledge learned from ImageNet is washed out after fine-tuning on MRI data. Also, comparing to setting (1), (2) and (3), the pattern in the images obtained from training on only ImageNet (the first column in the figure) looks quite different, this may imply that, even if we have a down-sampling scheme to produce aliasing artifacts for the natural images, the mapping learned from them is still quite different from the one learned from MRI data. Fig.4.8(b) illustrates our intuition of the learned weights under different settings.

For the images in the first row, we do not observe any significant difference like the one we observe for the second row and third row. This may due to the small receptive field of the lower layer.

4.3.2 Canonical Correlation Analysis (CCA) for the Layer Features of the Network

In order to investigate how the pretrained weights affects the performance, it is useful to analyze the representations in different layers. Investigating these latent layers is difficult because of the curse of high dimensionality. We compare the model with the 3 settings that we mentioned in Sec.4.2.3 , by using a recent tool CCA (Raghu *et al.*, 2017), which has been applied to some research on latent representations study (Saphra and Lopez, 2019; Magill *et al.*, 2018; Gotmare *et al.*, 2018), to evaluate the similarity of their weights.

Instead of comparing the weights of the model directly, CCA compares the activations of the neurons. To be more specific, it compares the outputs of the neuron on a sequence of inputs by searching correlated linear combinations. The higher score, the higher probability that the two mappings have similar functionality. Fig.4.7 shows the score in different layers for setting (1) and (2) versus other settings. If pretraining is helping, the score between setting (1) and (3) should be significantly higher than the one between setting (2) and (3). However, we do not observe it from the rightmost figure. Also, the first two figures are very similar, suggesting the pretraining weights are washed out or do not have a noticeable impact.

4.3.3 Experimental Settings

All the experiments were implemented using Pytorch 1.3, with python 3.7 on Ubuntu 18.04. They were performed on a computer with nVidia GTX 1080 GPU and Intel Xeon E5-2603 CPU.

Chapter 5

CONCLUSION

Super-resolution is an important field of study. Such technique can help increase the resolution for natural images, and can improve patient experience while doing the MRI scanning. In this dissertation, I present three works on super-resolution for natural images, two works on acceleration MRI reconstruction and the feasibility of transferring knowledge learned from natural images to reconstruct MR images. It is worth mentioning that some of my other accepted papers are also related to these topics indirectly. In (Li *et al.*, 2018b), I proposed a new activation function for increasing the number of response region, and in (Ding *et al.*, 2020), I proposed a new batch normalization technique, which makes the network to converge better and faster. Both works help improving the performance of neural network, and thus can be used for either natural image super-resolution or accelerated MRI reconstruction.

In the following, I summarize my major contributions for each topic, and provides some potential future works.

5.1 Super-resolution for Natural Images

I proposed a novel dictionary-learning approach to single-image super-resolution. Adding the convex constraint, which forces the columns of the dictionary to be formed by the columns of the input data, makes the columns of the dictionary close to the centroids of the input clusters. We also presented an iterative update algorithm for finding a solution. Experiments on commonly-used testing images showed that our approach is able to better reconstruct high-resolution images. Future work includes applying our proposed approach to multi-frame image super-resolution while consid-

ering joint regularization (Chang *et al.*, 2016a).

For the first regularization-based model, two new regularization models are proposed for single image SR, including GRR and 3R. GRR exploits both local and non-local self-similarity of images, while 3R introduces HR information from an external dataset. In addition, an algorithm called JRSR is designed to solve the optimization problem. Experiments demonstrated that our algorithm is able to deliver the best performance among all the tested methods. Moreover, the training phase of JRSR is independent of the degradation model. This independence is an advantage of JRSR over the learning-based methods.

For the second regularization-based model, I consider a competitive reconstruction-based framework which simultaneously utilizes both the external and internal priors of natural images. First, the suitable local neighbours are collected from external data so as to establish the CRR prior. After that, to yield a better SR result, the NLR prior which exploits NLSS of natural images is also incorporated. The effectiveness of the proposed CRNS algorithm is demonstrated by the experimental results with comparisons against different SR methods. Particularly, the average improvement of CRNS over NCSR (Dong *et al.*, 2013) (which provides the second best results among all the tested methods) is 0.40 dB in terms of PSNR, 0.0068 in terms of SSIM and 0.0059 in terms of FSIM. However, the computational burden of CRNS is higher than the other benchmark methods. In the future, we will try to extend CRNS to other applications, such as image deblurring and video SR.

5.2 Accelerated MRI Reconstruction

I proposed a new architecture to approximate the fully sampled MR images from the down sampled MR ones. The architecture is based on U-Net, and it achieves low NMSE during the reconstruction of MR images, due to the participation of the

residual dense refinement and the Fourier regularization. The visual results show that our proposed model is able to reduce more aliasing artifacts.

Also, I proposed a new network architecture - Variational Feedback Network (VFN) for Accelerated MRI reconstruction. The feedback connections and the recurrent U-Net structure can transmit the high level features back to the lower layers and refine the low level features, while reusing a lot of parameters. The experimental results have demonstrated that, our proposed VFN outperforms other state-of-the-art methods.

5.3 Transfer Learning for Accelerated MRI Reconstruction

I investigated how critical transfer learning is to reconstructing high quality images in accelerated MRI acquisition. To support the analysis, we designed a scheme to generate the phase part of the natural images, such that the images can be degraded similar to real MRI data. Through comparing the reconstruction error and the similarity of the weights, as well as visualizing the layer features, we found that transfer learning (using a typical pretraining-finetuning protocal) offers little benefit to performance. Future works could study if it is possible to learn a mapping such that the natural images can be mapped to a domain which is more correlated to the MRI data.

REFERENCES

Abràmoff, M. D., Y. Lou, A. Erginay, W. Clarida, R. Amelon, J. C. Folk and M. Niemeijer, "Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning", Investigative ophthalmology & visual science **57**, 13, 5200–5206 (2016).

Aharon, M., M. Elad and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation", Signal Processing, IEEE Transactions on **54**, 11, 4311–4322 (2006).

Bevilacqua, M., A. Roumy, C. Guillemot and A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding", in "British Machine Vision Conference (BMVC)", pp. 1–12 (Springer, Avignon, France, 2012).

Bredies, K., K. Kunisch and T. Pock, "Total generalized variation", SIAM Journal on Imaging Sciences **3**, 3, 492–526 (2010).

Buades, A., B. Coll and J. M. Morel, "A review of image denoising algorithms, with a new one", SIAM Multiscale Modeling and Simulation **4**, 2, 490–530 (2005).

Byra, M., M. Wu, X. Zhang, H. Jang, Y.-J. Ma, E. Y. Chang, S. Shah and J. Du, "Knee menisci segmentation and relaxometry of 3d ultrashort echo time cones mr imaging using attention u-net with transfer learning", Magnetic resonance in medicine **83**, 3, 1109–1122 (2020).

Candès, E. J., X. Li, Y. Ma and J. Wright, "Robust principal component analysis?", Journal of the ACM (JACM) **58**, 3, 11 (2011).

Chang, H., D.-Y. Yeung and Y. Xiong, "Super-resolution through neighbor embedding", in "Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on", vol. 1, pp. I–I (2004).

Chang, K., P. L. K. Ding and B. Li, "Color image demosaicking using inter-channel correlation and nonlocal self-similarity", Signal Processing: Image Communication **39**, 264–279 (2015).

Chang, K., P. L. K. Ding and B. Li, "Compressive sensing reconstruction of correlated images using joint regularization", IEEE Signal Processing Letters **23**, 4, 449–453 (2016a).

Chang, K., P. L. K. Ding and B. Li, "Single image super-resolution using collaborative representation and non-local self-similarity", Signal Processing **149**, 49–61 (2018a).

Chang, K., P. L. K. Ding and B. Li, "Single image super resolution using joint regularization", IEEE Signal Processing Letters **25**, 4, 596–600 (2018b).

Chang, K. and B. Li, "Joint modeling and reconstruction of a compressively-sensed set of correlated images", Journal of Visual Communication and Image Representation **33**, 286–300 (2015).

Chang, K., Y. Liang, C. Chen, Z. Tang and T. Qin, "Color image compressive sensing reconstruction by using inter-channel correlation", in "IEEE International Conference on Visual Communications and Image Processing (VCIP)", pp. 1–4 (IEEE, Chengdu, China, 2016b).

Chen, E. Z., T. Chen and S. Sun, "Mri image reconstruction via learning optimization using neural odes", arXiv preprint arXiv:2006.13825 (2020).

Chen, H., X. He, L. Qing and Q. Teng, "Single image super-resolution via adaptive transform-based nonlocal self-similarity modeling and learning-based gradient regularization", IEEE Transactions on Multimedia **19**, 8, 1702–1717 (2017).

Cheong, J. Y. and I. K. Park, "Deep CNN-based super-resolution using external and internal examples", IEEE Signal Processing Letters **24**, 8, 1252–1256 (2017).

Chierchia, G., N. Pustelnik, B. Pesquet-Popescu and J.-C. Pesquet, "A nonlocal structure tensor-based approach for multicomponent image recovery problems", IEEE Transactions on Image Processing **23**, 12, 5531–5544 (2014).

Conneau, A., H. Schwenk, L. Barrault and Y. Lecun, "Very deep convolutional networks for natural language processing", arXiv preprint arXiv:1606.01781 **2** (2016).

Delalleau, O. and Y. Bengio, "Shallow vs. deep sum-product networks", in "Advances in Neural Information Processing Systems", pp. 666–674 (2011).

Ding, C., T. Li and M. Jordan, "Convex and semi-nonnegative matrix factorizations", Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**, 1, 45–55 (2010).

Ding, P. L. K., B. Li and K. Chang, "Convex dictionary learning for single image super-resolution", in "2017 IEEE International Conference on Image Processing (ICIP)", pp. 4058–4062 (2017).

Ding, P. L. K., Z. Li, Y. Zhou and B. Li, "Deep residual dense u-net for resolution enhancement in accelerated mri acquisition", in "Medical Imaging 2019: Image Processing", vol. 10949, p. 109490F (International Society for Optics and Photonics, 2019).

Ding, P. L. K., S. Martin and B. Li, "Improving batch normalization with skewness reduction for deep neural networks", 25rd International Conference on Pattern Recognition (ICPR) (2020).

Dong, C., C. C. Loy, K. He and X. Tang, "Image super-resolution using deep convolutional networks", IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 2, 295–307 (2016a).

Dong, C., C. C. Loy and X. Tang, "Accelerating the super-resolution convolutional neural network", in "European Conference on Computer Vision (ECCV)", pp. 391–407 (Springer, Amsterdam, The Netherlands, 2016b).

Dong, W., G. Shi, X. Li, Y. Ma and F. Huang, "Comressive sensing via nonlocal low-rank regularization", IEEE Transactions on Image Processing **23**, 8, 3618–3612 (2014).

Dong, W., L. Zhang, G. Shi and X. Li, "Nonlocally centralized sparse representation for image restoration", IEEE Transactions on Image Processing **22**, 4, 1620–1630 (2013).

Dong, W., L. Zhang, G. Shi and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization", IEEE Transactions on Image Processing **20**, 7, 1838–1857 (2011).

Donoho, D., "Compressed sensing", Information Theory, IEEE Transactions on **52**, 4, 1289–1306 (2006).

El Gheche, M., J.-F. Aujol, Y. Berthoumieu and C.-A. Deledalle, "Texture reconstruction guided by a high-resolution patch", IEEE Transactions on Image Processing **26**, 2, 549–560 (2017).

Elad, M. and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries", IEEE Transactions on Image Processing **15**, 12, 3736–3745 (2006).

Engan, K., S. Aase and J. Husoy, "Frame based signal compression using method of optimal directions (mod)", in "Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on", vol. 4, pp. 1–4 vol.4 (1999).

Erhan, D., Y. Bengio, A. Courville and P. Vincent, "Visualizing higher-layer features of a deep network", University of Montreal **1341**, 3, 1 (2009).

Farsiu, S., M. Robinson, M. Elad and P. Milanfar, "Fast and robust multiframe super resolution", Image Processing, IEEE Transactions on **13**, 10, 1327–1344 (2004).

Feichtenhofer, C., H. Fan, J. Malik and K. He, "Slowfast networks for video recognition", in "Proceedings of the IEEE international conference on computer vision", pp. 6202–6211 (2019).

Freeman, W. T., T. R. Jones and E. C. Pasztor, "Example-based super-resolution", IEEE Comput. Graph. Appl. **22**, 2, 56–65 (2002).

Ganin, Y. and V. Lempitsky, "Unsupervised domain adaptation by backpropagation", in "International Conference on Machine Learning", pp. 1180–1189 (2015).

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand and V. Lempitsky, "Domain-adversarial training of neural networks", The Journal of Machine Learning Research **17**, 1, 2096–2030 (2016).

Gao, X., K. Zhang, D. Tao and X. Li, "Image super-resolution with sparse neighbor embedding", IEEE Transactions on Image Processing **21**, 7, 3194–3205 (2012).

Gatys, L. A., A. S. Ecker and M. Bethge, "Image style transfer using convolutional neural networks", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 2414–2423 (2016).

Gilbert, C. D. and M. Sigman, "Brain states: top-down influences in sensory processing", Neuron **54**, 5, 677–696 (2007).

Gilboa, G. and S. Osher, "Nonlocal operators with applications to image processing", Multiscale Modeling and Simulation **7**, 3, 1005–1028 (2008).

Goldstein, T. and S. Osher, "The split bregman method for $l1$ regularized problems", SIAM Journal on Imaging Sciences **2**, 2, 323–343 (2009).

Gotmare, A., N. S. Keskar, C. Xiong and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation", arXiv preprint arXiv:1810.13243 (2018).

Griswold, M. A., P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer and A. Haase, "Generalized autocalibrating partially parallel acquisitions (grappa)", Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **47**, 6, 1202–1210 (2002).

Gulshan, V., L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs", Jama **316**, 22, 2402–2410 (2016).

Hammernik, K., T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock and F. Knoll, "Learning a variational network for reconstruction of accelerated mri data", Magnetic resonance in medicine **79**, 6, 3055–3071 (2018).

He, K., G. Gkioxari, P. Dollár and R. Girshick, "Mask r-cnn", in "Computer Vision (ICCV), 2017 IEEE International Conference on", pp. 2980–2988 (IEEE, 2017).

He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 770–778 (2016).

Hidane, M., J.-F. Aujol, Y. Berthoumieu and C.-A. Deledalle, "Super-resolution from a low- and partial high- resolution image pair", in "IEEE International Conference on Image Processing (ICIP)", pp. 2145–2149 (IEEE, Paris, France, 2014).

Hidane, M., M. El Gheche, J.-F. Aujol, Y. Berthoumieu and C.-A. Deledalle, "Image zoom completion", IEEE Transactions on Image Processing **25**, 8, 3505–3517 (2016).

Hou, H. and H. Andrews, "Cubic splines for image interpolation and digital filtering", Acoustics, Speech and Signal Processing, IEEE Transactions on **26**, 6, 508–517 (1978).

Hu, Y. and M. Jacob, "Higher degree total variation (HDTV) regularization for image recovery", IEEE Transactions on Image Processing **21**, 5, 2559–2571 (2012).

Huang, G., Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks", in "CVPR", vol. 1, p. 3 (2017).

Huang, J. and A. S. N. Ahuja, "Single image super-resolution from transformed self-exemplars", in "IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 5197–5206 (IEEE, Boston, USA, 2015).

Hupé, J., A. James, B. Payne, S. Lomber, P. Girard and J. Bullier, "Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons", Nature **394**, 6695, 784–787 (1998).

Hyun, C. M., H. P. Kim, S. M. Lee, S. Lee and J. K. Seo, "Deep learning for undersampled mri reconstruction", Physics in Medicine & Biology **63**, 13, 135007 (2018).

Jia, K., X. Wang and X. Tang, "Image transformation based on learning dictionaries across image spaces", Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**, 2, 367–380 (2013).

Jiang, J., X. Ma, C. Chen, T. Lu, Z. Wang and J. Ma, "Single image super-resolution via locally regularized anchored neighborhood regression and nonlocal means", IEEE Transactions on Multimedia **19**, 1, 15–26 (2017).

Kang, E., J. Min and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction", Medical physics **44**, 10 (2017).

Kim, J., J. Kwon Lee and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 1637–1645 (2016a).

Kim, J., J. Lee and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks", in "IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 1646–1654 (IEEE, Las Vegas, USA, 2016b).

Kim, K. and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior", IEEE Transactions on Pattern Analysis and Machine Intelligence **32**, 6, 1127–1133 (2010).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

Knoll, F., K. Hammernik, E. Kobler, T. Pock, M. P. Recht and D. K. Sodickson, "Assessment of the generalization of learned image reconstruction and the potential for transfer learning", Magnetic resonance in medicine **81**, 1, 116–128 (2019).

Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in "Advances in neural information processing systems", pp. 1097–1105 (2012).

Kulkarni, N., P. Nagesh, R. Gowda and B. Li, "Understanding compressive sensing and sparse representation-based super-resolution", Circuits and Systems for Video Technology, IEEE Transactions on **22**, 5, 778–789 (2012).

Kumar, A., O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing", in "International conference on machine learning", pp. 1378–1387 (2016).

Kwon, K., D. Kim and H. Park, "A parallel mr imaging method using multilayer perceptron", Medical physics **44**, 12, 6209–6224 (2017).

Lee, D., K. H. Jin, E. Y. Kim, S.-H. Park and J. C. Ye, "Acceleration of mr parameter mapping using annihilating filter-based low rank hankel matrix (aloha)", Magnetic resonance in medicine **76**, 6, 1848–1864 (2016).

Lee, D., J. Yoo and J. C. Ye, "Deep residual learning for compressed sensing mri", pp. 15–18 (2017).

Lefkimmiatis, S., A. Roussos, P. Maragos and M. Unser, "Structure tensor total variation", SIAM Journal on Imaging Sciences **8**, 2, 1090–1122 (2015).

Li, H., Z. Xu, G. Taylor, C. Studer and T. Goldstein, "Visualizing the loss landscape of neural nets", in "Advances in Neural Information Processing Systems", pp. 6391–6401 (2018a).

Li, X., H. He, R. Wang and J. Cheng, "Superpixel-guided nonlocal means for image denoising and super-resolution", Signal Processing **124**, 173–183 (2016).

Li, X. and M. T. Orchard, "New edge-directed interpolation", IEEE Transactions on Image Processing **10**, 10, 1521–1527 (2001).

Li, Y., P. L. K. Ding and B. Li, "Training neural networks by using power linear units (polus)", arXiv preprint arXiv:1802.00212 (2018b).

Li, Y. and A. Ngom, "The non-negative matrix factorization toolbox for biological data mining", Source Code for Biology and Medicine **8**, 1, 1–15, URL `http://dx.doi.org/10.1186/1751-0473-8-10` (2013).

Li, Z., J. Yang, Z. Liu, X. Yang, G. Jeon and W. Wu, "Feedback network for image super-resolution", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 3867–3876 (2019).

Liao, Q. and T. Poggio, "Bridging the gaps between residual learning, recurrent neural networks and visual cortex", arXiv preprint arXiv:1604.03640 (2016).

Liu, D., Z. Wang and B. Wen, "Robust single image super-resolution via deep networks with sparse prior", IEEE Transactions on Image Processing **25**, 7, 3194–3207 (2016).

Liu, J., W. Yang, X. Zhang and Z. Guo, "Retrieval compensated group structured sparsity for image super-resolution", IEEE Transactions on Multimedia **19**, 2, 302–316 (2017).

Lustig, M., D. L. Donoho, J. M. Santos and J. M. Pauly, "Compressed sensing mri", IEEE signal processing magazine **25**, 2, 72–82 (2008).

Ma, C., C. Y. Yang, X. Yang and M. H. Yang, "Learning a no-reference quality metric for single-image super-resolution", Computer Vision and Image Understanding **158**, 1–16 (2017).

Maas, A. L., A. Y. Hannun and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models", in "Proc. icml", vol. 30, p. 3 (2013).

Maggioni, M., V. Katkovnik, K. Egiazarian and A. Foi, "A nonlocal transform-domain filter for volumetric data denoising and reconstruction", IEEE Transactions on Image Processing **22**, 1, 119–133 (2013).

Magill, M., F. Qureshi and H. de Haan, "Neural networks trained to solve differential equations learn general representations", in "Advances in Neural Information Processing Systems", pp. 4071–4081 (2018).

Mairal, J., G. Sapiro and M. Elad, "Learning multiscale sparse representations for image and video restoration", Multiscale Modeling & Simulation **7**, 1, 214–241 (2008).

Maqsood, M., F. Nazir, U. Khan, F. Aadil, H. Jamal, I. Mehmood and O.-y. Song, "Transfer learning assisted classification and detection of alzheimer's disease stages using 3d mri scans", Sensors **19**, 2645 (2019).

Montufar, G. F., R. Pascanu, K. Cho and Y. Bengio, "On the number of linear regions of deep neural networks", in "Advances in neural information processing systems", pp. 2924–2932 (2014).

Pascanu, R., G. Montúfar and Y. Bengio, "On the number of inference regions of deep feed forward networks with piece-wise linear activations", CoRR **abs/1312.6098**, URL `http://arxiv.org/abs/1312.6098` (2013).

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, "Automatic differentiation in pytorch", (2017).

Peleg, T. and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution", IEEE Transactions on Image Processing **23**, 6, 2569–2582 (2014).

Pruessmann, K. P., M. Weiger, M. B. Scheidegger and P. Boesiger, "Sense: sensitivity encoding for fast mri", Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **42**, 5, 952–962 (1999).

Raghu, M., J. Gilmer, J. Yosinski and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability", in "Advances in Neural Information Processing Systems", pp. 6076–6085 (2017).

Raghu, M., C. Zhang, J. Kleinberg and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging", in "Advances in Neural Information Processing Systems", pp. 3342–3352 (2019).

Rajpurkar, P., J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning", arXiv preprint arXiv:1711.05225 (2017).

Ren, C., X. He and T. Q. Nguyen, "Single image super-resolution via adaptive high-dimensional non-local total variation and adaptive geometric feature", IEEE Transactions on Image Processing **26**, 1, 90–106 (2017).

Ren, C., X. He, Q. Teng, Y. Wu and T. Q. Nguyen, "Single image super-resolution using local geometric duality and non-local similarity", IEEE Transactions on Image Processing **25**, 5, 2168–2183 (2016).

Ren, S., K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", in "Advances in neural information processing systems", pp. 91–99 (2015).

Roemer, P. B., W. A. Edelstein, C. E. Hayes, S. P. Souza and O. M. Mueller, "The nmr phased array", Magnetic resonance in medicine **16**, 2, 192–225 (1990).

Ronneberger, O., P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in "International Conference on Medical image computing and computer-assisted intervention", pp. 234–241 (Springer, 2015).

Rudin, L., S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms", Physica D: Nonlinear Phenomena **60**, 1, 259–268 (1992).

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge", International Journal of Computer Vision **115**, 3, 211–252 (2015).

Saphra, N. and A. Lopez, "Understanding learning dynamics of language models with svcca", in "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", pp. 3257–3267 (2019).

Shin, P. J., P. E. Larson, M. A. Ohliger, M. Elad, J. M. Pauly, D. B. Vigneron and M. Lustig, "Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion", Magnetic resonance in medicine **72**, 4, 959–970 (2014).

Shocher, A., N. Cohen and M. Irani, ""zero-shot" super-resolution using deep internal learning", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 3118–3126 (2018).

Sodickson, D. K. and W. J. Manning, "Simultaneous acquisition of spatial harmonics (smash): fast imaging with radiofrequency coil arrays", Magnetic resonance in medicine **38**, 4, 591–603 (1997).

Stollenga, M. F., J. Masci, F. Gomez and J. Schmidhuber, "Deep networks with internal selective attention through feedback connections", in "Advances in neural information processing systems", pp. 3545–3553 (2014).

Sun, J., J. Sun, Z. Xu and H.-Y. Shum, "Image super-resolution using gradient profile prior", in "Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on", pp. 1–8 (2008).

Takeda, H., S. Farsiu and P. Milanfar, "Kernel regression for image processing and reconstruction", IEEE Transactions on Image Processing **16**, 2, 349–366 (2007).

Timofte, R., V. D. Smet and L. V. Gool, "Anchored neighborhood regression for fast example-based super resolution", in "IEEE International Conference on Computer Vision (ICCV)", pp. 1920–1927 (IEEE, Sydney, Australia, 2013).

Timofte, R., V. D. Smet and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution", in "Asian Conference on Computer Vision (ACCV)", pp. 111–126 (Springer, Singapore, 2014).

Tipping, M. E. and C. M. Bishop, "Bayesian image super-resolution", in "Advances in Neural Information Processing Systems 15", edited by S. Becker, S. Thrun and K. Obermayer, pp. 1303–1310 (MIT Press, 2003), URL `http://papers.nips.cc/paper/2315-bayesian-image-super-resolution.pdf`.

Tong, T., G. Li, X. Liu and Q. Gao, "Image super-resolution using dense skip connections", in "Computer Vision (ICCV), 2017 IEEE International Conference on", pp. 4809–4817 (IEEE, 2017).

Uecker, M., P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala and M. Lustig, "Espirit—an eigenvalue approach to autocalibrating parallel mri: where sense meets grappa", Magnetic resonance in medicine **71**, 3, 990–1001 (2014).

Ulyanov, D., A. Vedaldi and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization", arXiv preprint arXiv:1607.08022 (2016).

Wang, H., X. Gao, K. Zhang and J. Li, "Fast single image super-resolution using sparse Gaussian process regression", Signal Processing **134**, 52–62 (2017a).

Wang, H., X. Gao, K. Zhang and J. Li, "Single image super-resolution using Gaussian process regression with dictionary-based sampling and student-t likelihood", IEEE Transactions on Image Processing **26**, 7, 3556–3568 (2017b).

Wang, S., Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng and D. Liang, "Accelerating magnetic resonance imaging via deep learning", in "Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on", pp. 514–517 (IEEE, 2016).

Wang, S., D. Zhang, Y. Liang and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis", in "Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on", pp. 2216–2223 (2012).

Wang, X., Y. Peng, L. Lu, Z. Lu, M. Bagheri and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 2097–2106 (2017c).

Wang, Z., A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", IEEE Transactions on Image Processing **13**, 4, 600–612 (2004).

Wang, Z., D. Liu, J. Yang, W. Han and T. Huang, "Deep networks for image super-resolution with sparse prior", in "IEEE International Conference on Computer Vision (ICCV)", pp. 370–378 (IEEE, Santiago, Chile, 2015a).

Wang, Z., E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment", in "The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003", vol. 2, pp. 1398–1402 (Ieee, 2003).

Wang, Z., Y. Yang, Z. Wang, S. Chang, J. Yang and T. S. Huang, "Learning super-resolution jointly from external and internal examples", IEEE Transactions on Image Processing **24**, 11, 4359–4371 (2015b).

Wu, C.-Y., C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl and R. Girshick, "Long-term feature banks for detailed video understanding", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 284–293 (2019).

Xiao, J., J. Hays, K. A. Ehinger and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo", in "Computer vision and pattern recognition (CVPR)", pp. 3485–3492 (IEEE, San Francisco, CA, USA, 2010).

Yang, C., J. Huang and M. Yang, "Exploiting self-similarities for single frame super-resolution", in "Asian Conference on Computer Vision (ACCV)", pp. 497–510 (Springer, Queenstown, New Zealand, 2010a).

Yang, C. Y. and M. H. Yang, "Fast direct super-resolution by simple functions", in "IEEE International Conference on Computer Vision (ICCV)", pp. 561–568 (IEEE, Sydney, Australia, 2013).

Yang, J., Z. Wang, Z. Lin, S. Cohen and T. Huang, "Coupled dictionary training for image super-resolution", Image Processing, IEEE Transactions on **21**, 8, 3467–3478 (2012a).

Yang, J., Z. Wang, Z. Lin, S. Cohen and T. Huang, "Coupled dictionary training for image super-resolution", IEEE Transactions on Image Processing **21**, 8, 3467–3478 (2012b).

Yang, J., J. Wright, T. Huang and Y. Ma, "Image super-resolution via sparse representation", Image Processing, IEEE Transactions on **19**, 11, 2861–2873 (2010b).

Yang, M. and Y. Wang, "A self-learning approach to single image super-resolution", IEEE Transactions on Multimedia **15**, 3, 498–508 (2013).

Yao, W., Z. Zeng, C. Lian and H. Tang, "Pixel-wise regression using u-net and its application on pansharpening", Neurocomputing (2018).

Yue, L., H. Shen, J. Li, Q. Yuan and H. Zhang, "Image super-resolution: The techniques, applications, and future", Signal Processing **128**, 389–408 (2016).

Zamir, A. R., T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik and S. Savarese, "Feedback networks", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 1308–1317 (2017).

Zbontar, J., F. Knoll, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana *et al.*, "fastmri: An open dataset and benchmarks for accelerated mri", arXiv preprint arXiv:1811.08839 (2018).

Zeyde, R., M. Elad and M. Protter, "On single image scale-up using sparse-representations", in "International Conference on Curves and Surfaces", pp. 711–730 (Springer, Avignon, France, 2010).

Zhang, K., X. Gao, J. Li and H. Xia, "Single image super-resolution using regularization of non-local steering kernel regression", Signal Processing **123**, 53–63 (2016a).

Zhang, K., X. Gao, D. Tao and X. Li, "Multi-scale dictionary for single image super-resolution", in "Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on", pp. 1114–1121 (2012a).

Zhang, K., X. Gao, D. Tao and X. Li, "Single image super-resolution with non-local means and steering kernel regression", IEEE Transactions on Image Processing **21**, 11, 4544–4556 (2012b).

Zhang, K., J. Li, H. Wang, X. Liu and X. Gao, "Learning local dictionaries and similarity structures for single image super-resolution", Signal Processing **142**, 231–243 (2018).

Zhang, K., D. Tao, X. Gao, X. Li and J. Li, "Coarse-to-fine learning for single-image super-resolution", IEEE Transactions on Neural Networks and Learning System **28**, 5, 1109–1122 (2017).

Zhang, K., D. Tao, X. Gao, X. Li and Z. Xiong, "Learning multiple linear mappings for efficient single image super-resolution", IEEE Transactions on Image Processing **24**, 3, 846–861 (2015).

Zhang, L. and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion", IEEE Transactions on Image Processing **15**, 8, 2226–2238 (2006).

Zhang, L., L. Zhang, X. Mou and D. Zhang, "FSIM: A feature similarity index for image quality assessment", IEEE Transactions on Image Processing **20**, 8, 2378–2386 (2011).

Zhang, X., M. Burger, X. Bresson and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction", SIAM Journal on Imaging Sciences **3**, 3, 253–276 (2010).

Zhang, Y., Y. Zhang, J. Zhang and Q. Dai, "CCR: Clustering and collaborative representation for fast single image super-resolution", IEEE Transactions on Multimedia **18**, 3, 405–417 (2016b).

Zhu, B., J. Z. Liu, S. F. Cauley, B. R. Rosen and M. S. Rosen, "Image reconstruction by domain-transform manifold learning", Nature **555**, 7697, 487–492 (2018).

Zhu, Z., F. Guo, H. Yu and C. Chen, "Fast single image super-resolution via self-example learning and sparse representation", IEEE Transactions on Multimedia **16**, 8, 2178–2190 (2014).