

August 2019

Model Augmented Deep Neural Networks for Medical Image Reconstruction Problems

Hongquan Zuo
University of Wisconsin-Milwaukee

Follow this and additional works at: <https://dc.uwm.edu/etd>



Part of the [Computer Sciences Commons](#), and the [Electrical and Electronics Commons](#)

Recommended Citation

Zuo, Hongquan, "Model Augmented Deep Neural Networks for Medical Image Reconstruction Problems" (2019). *Theses and Dissertations*. 2278.

<https://dc.uwm.edu/etd/2278>

This Dissertation is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UWM Digital Commons. For more information, please contact open-access@uwm.edu.

MODEL AUGMENTED DEEP NEURAL NETWORKS FOR MEDICAL
IMAGE RECONSTRUCTION PROBLEMS

by

Hongquan Zuo

A Dissertation Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
in Engineering

at

The University of Wisconsin-Milwaukee
August 2019

ABSTRACT

MODEL AUGMENTED DEEP NEURAL NETWORKS FOR MEDICAL IMAGE RECONSTRUCTION PROBLEMS

by

Hongquan Zuo

The University of Wisconsin-Milwaukee, 2019

Under the Supervision of Professor Jun Zhang

Solving an ill-posed inverse problem is difficult because it doesn't have a unique solution. In practice, for some important inverse problems, the conventional methods, e.g. ordinary least squares and iterative methods, cannot provide a good estimate. For example, for single image super-resolution and CT reconstruction, the results of these conventional methods cannot satisfy the requirements of these applications. While having more computational resources and high-quality data, researchers try to use machine-learning-based methods, especially deep learning to solve these ill-posed problems. In this dissertation, a model augmented recursive neural network is proposed as a general inverse problem method to solve these difficult problems. In the dissertation, experiments show the satisfactory performance of the proposed method for single image super-resolution, CT reconstruction, and metal artifact reduction.

TABLE OF CONTENTS

List of Figures	vi
List of Tables	vii
Acknowledgements	viii
1 Introduction	1
1.1 Inverse Problems	1
1.2 Inverse Problem Methods	2
1.3 Deep Learning for Image Reconstructions	4
1.4 Contributions	6
2 Model Augmented RecurSive Neural Networks for Ill-posed Inverse Problems	8
2.1 Motivation of Model Augmented RecurSive Neural Networks	8
2.1.1 MAP Optimization	8
2.1.2 Gradient Descent for MAP	10
2.1.3 Projection and Back-projection Models	11
2.1.4 Deep Convolutional Neural Network Based Residual Estimator	13
2.1.5 Residual Learning in MARS	13
2.2 Structure of MARS	14
2.2.1 Bypass Branch	15
2.2.2 Likelihood Block	15
2.2.3 Prior Block	16
2.2.4 Reconstruction Block	16
2.3 Summary	19
3 Single Image Super-Resolution For Medical Images	21
3.1 Introduction	21
3.1.1 What is Super-Resolution	21
3.1.2 Why Use SISR for Medical Images	22
3.1.3 Organization of Chapter 3	23
3.2 Related Work	24

3.2.1	Before Deep Learning	24
3.2.2	Deep Learning	25
3.3	Contributions and Proposed Methods	27
3.3.1	Contributions	27
3.3.2	MARS for Medical Image SISR	28
3.3.2.1	Bypass branch	28
3.3.2.2	Likelihood Block	28
3.3.2.3	Prior Block	29
3.3.2.4	Reconstruction Block	29
3.3.3	Multi-channel MARS for SISR	30
3.3.4	Progressive Learning Structure of MARS	32
3.3.5	Special Problems for Medical Image SISR	34
3.3.5.1	Class Imbalance Problem	34
3.3.5.2	Should Use Training Data Augmentation?	35
3.3.5.3	Use L1-norm Loss Function for Medical Image SISR	36
3.4	Experimental Results	37
3.4.1	A preliminary Experiment	37
3.4.2	Natural Image SISR	38
3.4.2.1	Comparisons with state-of-the-art SISR methods	40
3.4.3	Chest X-ray image	42
3.4.3.1	Data Set and Platform	42
3.4.4	Results and Analysis	42
3.4.4.1	Limitations	45
3.4.5	conclusion	47
4	CT Reconstruction	48
4.1	Introduction	48
4.2	Related Works	50
4.2.1	Filtered Back-Projection	50
4.2.2	Iterative Methods	50
4.2.2.1	Statistical Model	51
4.2.2.2	Prior	52
4.2.2.3	Optimization	53
4.3	Contributions and Proposed Methods	53
4.3.1	Contributions	53
4.3.2	MARS for CT Reconstructions	54
4.3.2.1	Bypass Branch	54
4.3.2.2	Likelihood Feature Block	54
4.3.2.3	Prior Feature Block	55
4.3.3	Experimental Results of MARS for CT Reconstructions	55
4.3.4	Deep Learning Optimization Algorithms for Iterative Methods	56
4.3.5	Experimental Results of Deep Learning Optimization Algorithms for Iterative Methods	62

4.4 Conclusions	64
5 Summary, Limitations and Future Works	65
Curriculum Vitae	76

LIST OF FIGURES

1.1	Difficulty of inverse problems and model complexity of their methods	2
1.2	A comparison of inverse problem methods.	3
2.1	Iterative method structure	10
2.2	The structure of gradient descent.	12
2.3	The structure of naive MARS.	17
2.4	The structure of MARS.	18
3.1	SISR problem.	22
3.2	The structure of MMARS.	31
3.3	The structure of progressive learning.	33
3.4	Compare high-frequency patterns in medical images and natural images.	36
3.5	A preliminary Phantom X-ray image SISR	39
3.6	SISR results.	41
3.7	Balancing mask in MARS SISR	44
3.8	Without a balancing mask vs with a balancing mask.	44
3.9	Chest X-ray SISR results.	45
3.10	Chest X-ray SISR results: absolute errors of reconstructions.	46
3.11	Chest X-ray 2x/4x/8x SISR results.	46
4.1	A simple parallel CT scan that is an 2D Radon transform of objects.	49
4.2	A sinogram of a phantom image.	49
4.3	CT reconstruction training and testing loss curves with 3 configurations	57
4.4	CT reconstruction training and testing loss curves of each stages.	57
4.5	CT reconstruction result while increasing Poisson noise.	58
4.6	CT reconstruction results at two noise levels.	59
4.7	CT reconstruction results with strong noise and metal artifact.	60
4.8	CT reconstruction results with iterative methods.	63
4.9	Convergence results with iterative methods.	64

LIST OF TABLES

1.1	List of important deep learning network structures	6
3.1	Configurations of natural image SISR methods.	40
3.2	Average PSNR and SSIM for natural image SISR methods.	42
3.3	Training/ testing time of LapSRN and MARS for SISR.	42
3.4	Average PSNR/ SSIM of X-ray image SISR	43
3.5	PSNR/ SSIM of Figure 3.9	43
4.1	Average PSNR/ SSIM of CT reconstruction results	56
4.2	Training/ testing time of MARS for CT reconstruction.	56

ACKNOWLEDGMENTS

First, I would like to express our gratitude to our Ph.D. advisor, Dr. Jun Zhang for all his academic guidance, support and encouragement he has given to me during our time at the University of Wisconsin-Milwaukee. I sincerely appreciate the phone discussions at night with him in these years.

Besides our advisor, I would like to thank the rest of our dissertation committees: Dr. Chiu Tai Law, Dr. Mahsa Ranji, Dr. Tian Zhao, and Dr. Ping Xue. Thank you for their encouragements, helps, and recommendations.

Furthermore, I would like to thank our friends and group members: Dr. Yingying Gu, Yu Lu, Carlos Ivan Jerez Gonzalez, and Yang Zhang. Thank you for your friendships, good advice and sharing ideas. You supported me greatly and were always willing to help me. I also appreciate GE Healthcare founded us and supported our research projects.

Lastly, I would like to thank all our family members: our wife Mei Zhang, daughter Zoe, parents and our twin sister, who have hugely supported me and gave me the love all these years. I am very grateful. Thank you.

Chapter 1

Introduction

1.1 Inverse Problems

Solving an inverse problem is a process that reconstructs un-observed variables from a set of observed data. Inverse problem methods are widely used in signal processing, medical imaging, computer vision, and many other applications. For example, cross-section images of a patient can help radiologists to diagnose disease and injury, but they cannot be observed directly. Computer tomography (CT) reconstruction is an inverse problem, which generates these cross-section images from X-ray projection data acquired at many different angles around the patient. Another example of inverse problems is image super-resolution techniques, which generate a high-resolution image from one or a few observed low-resolution images.

Mathematically, in an inverse problem, a unknown variable $x \in X \rightarrow \mathbb{R}^{d_1}$ is reconstructed from an observed variable $y \in Y \rightarrow \mathbb{R}^{d_2}$, where d_1 and d_2 are dimension of space X and Y . It is usually modeled as a linear system

$$y = Ax + \epsilon \tag{1.1}$$

where $A : X \rightarrow Y$ is a linear projection model, ϵ is a random variable of noise. A could be a human-designed projection matrix, or a learned function from a training data set.

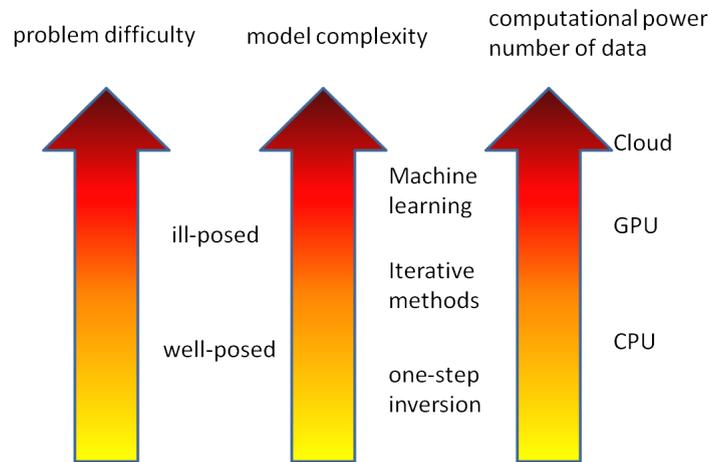


FIGURE 1.1: Difficulty of inverse problems and model complexity of their methods

1.2 Inverse Problem Methods

As shown in Figure 1.1, inverse problems have two difficulty categories: well-posed and ill-posed. A well-posed inverse problem has two features: 1) the number of unknown x is smaller than the number of observations y ; 2) the condition number of A is small. A well-posed problem has a unique and stable solution. A simple method, like the ordinary least squares (OLS), could generate a good result. On the contrary, in an ill-posed inverse problem, the number of unknowns is more than observations, or the condition number of A is large, which causes the numerical unstable problem or overfitting problem. “Unstable” means a small noise on y generates a large error on x . In practice, many important inverse problems are ill-posed. For example, single image super-resolution (SISR) and CT reconstruction with few views are two ill-posed inverse problems that are addressed in this dissertation.

Based on model complexity, there are three kinds of inverse problem methods:

1. one-step inversion, e.g. weighted least squares (WLS) in SISR and filtered back-projection (FBP) in CT reconstruction, see Appendix. The pros are simple and fast, the cons are low performance and sensitive to noise.
2. iterative methods, e.g. gradient descent and coordinate descent. The pros are tolerant of noise, no need to training data. The cons are slow convergence.

Method	Examples	Pros	Cons
One-step inversion	<ul style="list-style-type: none"> •weighted least squares (WLS) •filtered back-projection (FBP) 	<ul style="list-style-type: none"> •Simple •fast 	<ul style="list-style-type: none"> •sensitive to noise •manually-designed prior is weak
Iterative methods	<ul style="list-style-type: none"> •gradient descent (GD) •coordinate descent (CD) 	<ul style="list-style-type: none"> •noise tolerance 	<ul style="list-style-type: none"> •slow convergence •manually-designed prior is weak
Machine Learning	<ul style="list-style-type: none"> •deep learning (DL) in image reconstruction 	<ul style="list-style-type: none"> •superior results •data-driven prior is strong 	<ul style="list-style-type: none"> •training is time consuming •requires a large amount of training data

FIGURE 1.2: A comparison of inverse problem methods: one-step inversion methods, iterative methods and machine-learning-based methods. For CT reconstruction, FBP is an one-step inversion method.

3. machine-learning-based methods, especially deep learning for image reconstructions. The pros are superior performance and fast while testing. The cons are that it needs a large amount of training data and it is difficult to train a deep neural network.

An inverse problem and its method should have matched difficulty of the problem, model complexity of its method, and computational resources. The computational resources include computational power and training data sets. If the model complexity is higher than the difficulty, the method wastes computational resources and requires more training data. If the model complexity is lower than the difficulty, the result is not good.

Regularization functions is a popular tool to solve ill-posed inverse problems. A regularization function usually is an energy function, which has a sparse probability distribution. L1-norm [15], L2-norm or total variant (TV) of local gradients are common regularization functions. For example, TV regularization is used in CT reconstruction to preserve edges in CT images [65]. Mathematically, a regularization function reduces the condition number of A to reduce the ill-posedness of an inverse problem. In Bayesian statistics, maximum a posteriori probability (MAP) estimate is a popular method to estimate a solution of an inverse problem with regularization functions. Iterative methods, e.g. gradient descent or coordinate descent, are

used to solve MAP estimation and get close to the mode of this posterior distribution. If the problem is a convex problem [8], the mode of the posterior is the global optimum.

In practice, one-step inversion is too sensitive to noise ϵ . Meanwhile, iterative methods usually have a slow convergence and unsatisfactory results. For example, for low-dose CT reconstruction, the CT scan data has a strong Poisson noise. Then the reconstructed images have many artifacts, and it takes more iterations to get results. Now people can have more powerful computers and access more high-quality data, so they can choose machine-learning-based methods with a more complex model. deep learning, as a hot topic in machine learning, shows the superior performance upper-bound in many image reconstruction applications, e.g. image recognition [37], super-resolution [39] and segmentation [19].

1.3 Deep Learning for Image Reconstructions

deep learning is a family of machine learning methods to use artificial neural networks to learn the hierarchical representation of data. In recent years, deep learning has many success stories in inverse problems. Four main factors drive the growth of deep learning: structures, optimization algorithms, hardware and software platforms, and high-quality data sets.

Structures LeCun et al. [41] in 1989 proposed convolutional neural networks (CNN) and backpropagation algorithms to recognize hand write digit numbers. Alex et al. [37] used deep CNN AlexNet in ImageNet Large Scale Visual Recognition Challenge [12] and dramatically boosted the classification accuracy. People were excited by AlexNet's superior performance that reduced the classification error from 26.1% to 15.3%, also were inspired by how to use graphics processing unit (GPU) to accelerate deep neural network training. Simonyan et al. proposed VGGNet [49] to further improve the classification accuracy, which only stacks 3x3 CNN to build a very deep CNN. Because of the vanishing gradient problem (VGP), training a very deep neural network is very slow even stopped. He et al. [24] proposed ResNet structure, which contains bypass paths to do residual learning and directly backpropagation gradient to

parameters in front layers. Even more than 1000 layer CNNs can be trained with ResNet structure.

Optimization algorithms Deep learning optimization algorithms are also important to train a deep neural network. A typical deep neural network has millions trainable parameter, which are trained by Stochastic gradient descent (SGD) algorithm or Mini Batch gradient descent. Momentum [51] uses accumulated gradients and leads to faster and stable convergence. Nesterov accelerated gradient (NAG) [45] estimates new gradients to aggressively accelerate convergence. Adagrad [16] uses a adaptive learning Rate for every parameter. Adam [36] estimates mean and uncentered variance of gradients and reduce learning rate of the parameters, which have large variances of gradients.

Hardware and software platforms Besides many novel deep neural network structures, the developments of GPU and deep learning software platforms also support the growth of deep learning. Now the game-driven GPU industry has become Deep-Learning-driven. In our experiments, training a deep neural network with a GPU is 30 times faster than the training without GPU. The software platform like Caffe [32], Tensorflow [1], Pytorch [33], Theano [6], and Keras [21] reduce the workload to write a Deep Learning program, and lower the bar of studying deep learning.

High-quality data sets The MNIST database (Modified National Institute of Standards and Technology database) [41] is a large database of handwritten digits that is commonly used as a benchmark data set for training and testing of a deep neural network. It contains 60,000 training images and 10,000 testing images. ImageNet Large Scale Visual Recognition Challenge [12] data set contains more than 14 million hand-annotated images with more than 20,000 categories. DIV2K [4] is used in the SISR competitions NTIRE (CVPR 2017/2018) and PIRM (ECCV 2018).

Combine a model into a deep neural network An inverse problems usually have a projection matrix A to model the forward process from unknown x to observation y . Researchers

tried to combine A in to the structure of a deep neural network in order to provide more information and improve results. Weber et al. introduce Imagination-Augmented Agents (I2As) [60] to combine model-free and model-based aspects for deep reinforcement learning. Adler and Oktem [2] use iterative deep neural network as a gradient descent optimization process to reconstruct CT images. In their method, a deep CNN extracts features from gradient of likelihood function and prior function. The gradient of the likelihood function includes the projection matrix A . In this dissertation, we independently develop a similar structure as a general framework to solve ill-posed inverse problems. In our neural network, there are two feature extraction blocks, a model-related likelihood block and a model-free prior block. Comparing to Adler’s network, our network has more trainable parameters in the likelihood block to focus on the functionality of likelihood block. Adler didn’t study which branch gives more improvement. In this dissertation, an experiment in Chapter 4 proves that while working with a noised data, the likelihood block works better than the prior block.

TABLE 1.1: List of important deep learning network structures

Year	Structure
1989	LeNet [41]
2012	AlexNet [37]
2014	VGG [49]
2016	ResNet [24]
2017	DenseNet [27]

1.4 Contributions

The contributions of this dissertation are listed below:

1. Proposed a model augmented recursive neural network (MARS) structure as a framework to solve ill-posed inverse problems.
2. Used MARS in medical image Single Image Super-resolution (SISR). Address three special problems in medical image SISR to improve the resolution enhancement.

3. Used the proposed model augmented recursive neural network (MARS) in CT image reconstruction and metal artifact reduction (MAR)
4. Built a guideline of adjusting the distribution of trainable parameters in the proposed MARS neural networks: when the observed data contains strong noise, the likelihood block should have more trainable parameters. An experiment in this chapter demonstrated that when the initial solution, e.g. FBP, contained strong artifacts, the model-based likelihood block provided more improvements than the model-free prior block.
5. Used deep learning optimization algorithms, e.g. Momentum and Adam, for CT image iterative reconstructions.

This dissertation is organized as follows: Chapter 2 describes the motivation and variants of the proposed MARS neural network as a general framework to solve ill-posed inverse problems. Chapter 3 reviews SISR methods for natural and medical images. MARS is compared with a state-of-the-art SISR method for natural image SISR to demonstrate the functionality of MARS. Then MARS is used for medical image SISR. Chapter 4 reviews CT reconstruction methods and provides two experiments by using deep learning techniques for CT reconstruction: the first one is using two deep learning optimization algorithms to accelerate iterative CT reconstruction; the second experiment is using MARS for CT reconstruction and metal artifact reduction. Chapter 5 summarizes the main findings in this dissertation, and then discusses the limits of MARS and our future works.

Chapter 2

Model Augmented RecurSive Neural Networks for Ill-posed Inverse Problems

A model augmented recursive neural network (MARS) is proposed to solve ill-posed inverse problems. MARS combines human-designed projection and back-projection models into the structure of a deep neural network, to simplify the structure of a neural network and improve its performance. This chapter is organized as follows: Section 2.1 introduces the mathematics motivation of MARS, which is originally inspired by an iterative gradient descent algorithm to solve regularized inverse problems. Section 2.2 describes the structure of MARS and its main parts.

2.1 Motivation of Model Augmented RecurSive Neural Networks

2.1.1 MAP Optimization

A common procedure to solve an ill-posed inverse problem with some regularization is the following:

1. Build a projection model A while using the knowledge of generating y from x .

2. Apply a noise model of ϵ to build a probability likelihood function. The simplest case is to model ϵ as Gaussian random variables. Then the negative log-likelihood function is a weighted squared Euclidean distance between y and Ax .
3. In the Bayesian framework, add an extra constraint function $P(x)$ as a prior function to regularize x , then the inverse problem becomes a MAP optimization. In regularization theory, regularization can reduce the phenomenon of overfitting. total variation (TV) and generalized Gaussian Markov random field (GGMRF) are popular non-linear edge preserved smooth priors. Recently Data-driven priors [50], [55] give more competitive results.

$$\begin{aligned}
\hat{x} &= \arg \min_x Loss_{MAP}(x, y) \\
&= \arg \min_x (-\log P(y|x) - \log P(x)) \\
&= \arg \min_x (F_1(x, y) + F_2(x))
\end{aligned} \tag{2.1}$$

where $Loss_{MAP}(x, y) = -\log P(y|x) - \log P(x)$ is loss function of the MAP estimation. The first and second terms in Equation (2.1) are likelihood and prior functions, denoted by $F_1(x, y)$ and $F_2(x)$.

4. Solved the MAP estimation by iterative methods Equation (2.2).

$$x^{t+1} = x^t + R_\theta(x^t, y) \tag{2.2}$$

where t is the index of iterations, $R_\theta(x^t, y)$ a residual function to iteratively reduce reconstruction error, θ is fixed or learned parameters. $R_\theta(x^t, y)$ could be a human designed function or a machine learning based estimator. gradient descent (GD) or coordinate descent (CD) algorithms are first-order iterative methods. Second-order iterative methods, for instance Newton's method or quasi-Newton's method (BFGS), can improve optimization, but with intensive computation.

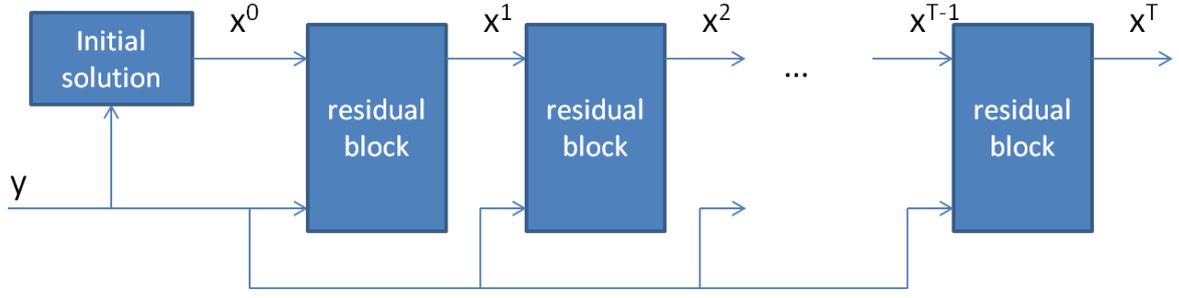


FIGURE 2.1: Iterative method structure

2.1.2 Gradient Descent for MAP

If both of the likelihood and prior functions have the first derivative, an iterative gradient descent solution at t iteration is

$$\begin{aligned}
 x^{t+1} &= x^t + R_{GD}(x^t, y) \\
 &= x^t - \alpha \nabla \text{Loss}_{MAP}(x^t, y) \\
 &= x^t - \alpha G(x^t, y) \\
 &= x^t - \alpha (\nabla_x F_1(x^t, y) + \nabla_x F_2(x^t))
 \end{aligned} \tag{2.3}$$

where α is a learning rate or step size of GD, $R_\theta(x^t, y) = R_{GD}(x^t, y)$ is a fixed function without trainable parameters, ∇_x is the gradient of a function with respect to x given y and $G(x^t, y) = \nabla \text{Loss}_{MAP}(x^t, y)$ is gradient at iteration t to compensate reconstruction error in x^t . The $-\alpha G(x^t, y)$ is a greedy estimate of the ground truth of residual $R_{gt} = x_{gt} - x^t$, where x_{gt} is the ground truth of x .

Based on a simple assumption that ϵ in Equation (1.1) is a Gaussian noise, the negative log likelihood function is

$$\begin{aligned}
 F_1(x) &= -\log P(y|x) \\
 &= -\log e^{-\frac{1}{2}(y-Ax)^T D(y-Ax)} \\
 &= \frac{1}{2}(y-Ax)^T D(y-Ax)
 \end{aligned} \tag{2.4}$$

where D is the inverse matrix of the covariance matrix of noise ϵ . If ϵ is a standardized independent and identically distributed (i.i.d.) Gaussian noise, D is the identity matrix, so

$$F_1(x) = \frac{1}{2}|y - Ax|^2 \quad (2.5)$$

In this case, the gradient from the likelihood function $F_1(x^t, y)$ is

$$\nabla_x F_1(x^t, y) = A^*(Ax^t - y) \quad (2.6)$$

where A^* is the adjoint operator of A . If A is a matrix, A^* is A^T , which is the transpose of A . A and A^T are called projections and back-projection models. In fact, $\nabla_x F_1(x^t) = A^T(Ax^t - y)$ has a nice interpretation of being the back-projected reconstruction error in the image domain from the reconstruction error $y - Ax^t$.

Traditionally prior term $F_2(x^t)$ is an human designed energy function of x^t , for example, total variation (TV) and generalized Gaussian Markov random field (GGMRF), as non-linear edge preserved priors. Now both of the likelihood and prior functions are human designed, so the gradient $G_{GD}(x^t, y)$ is a fixed function of x^t given y .

$$G_{GD}(x^t, y) = A^T(Ax^t - y) + \nabla_x F_2(x^t) \quad (2.7)$$

2.1.3 Projection and Back-projection Models

In iterative methods, both x^t and y are representations of the same image in latent space X and observation space Y . To measure inconsistency between x^t and y , there should be a projection model $A : X \rightarrow Y$ and a back-projection model $B : Y \rightarrow X$ to connect these two spaces. Because A is designed directly from the knowledge of an imaging system, A is accurate and reliable. Furthermore, a fixed A makes the problem simpler than a learned projection

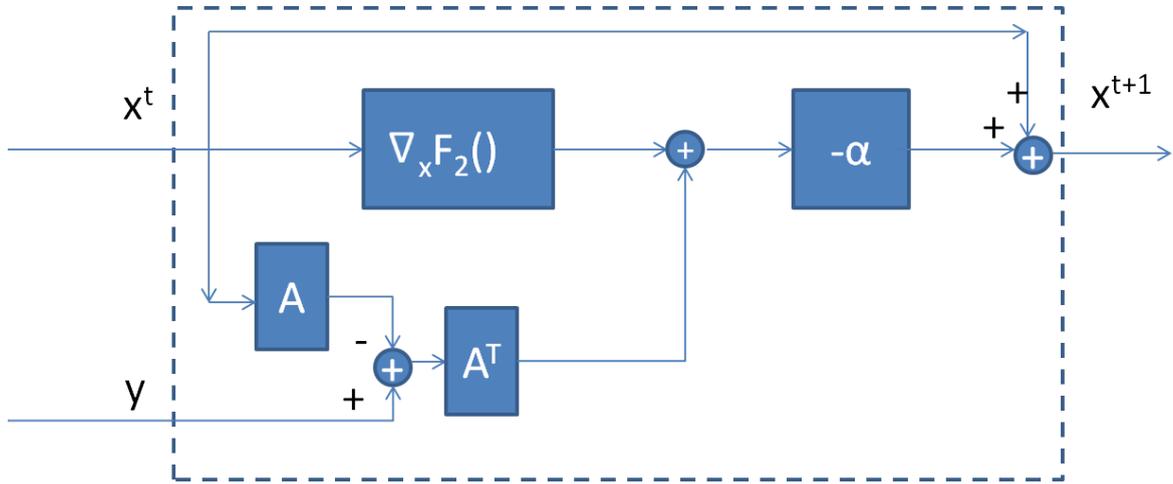


FIGURE 2.2: The structure of gradient descent.

model. So in this dissertation, A is a human-designed projection model. But there are two options for B : one is A^T , which is not the inversion of A and is just a back-projection with some statistic assumptions. A^T is a simple and low-cost back-projection model. Another is a machine-learning-based function B_{ML} , which is a sophisticated but high-cost back-projection model. The choice between A^T and B_{ML} depends on the cost-effectiveness for different applications.

In CT reconstructions, both of A and B are “global models”. X and Y are high-dimensional space. This high dimensionality causes two problems: first, it is a difficult task to train a model, e.g. Multilayer perceptrons (MLP), as a back-projection operator with a huge amount of parameters. Second, the number of parameters is too large to fit into a common computer system. For instance, x is a 1024×1024 CT image, and y is a 1000×1000 sinogram (CT data). The back-projection matrix is a $1024^2 \times 1000^2$ sparse matrix. If using 32-bit float, it needs almost 4 TB memory to store a full back-projection model B_{ML} . But after building a human-designed sparse matrix A^T [48] based on the geometric knowledge of a CT system, it only needs 2 GB to store A^T in a sparse matrix format. So A^T is used in CT reconstructions.

In SISR problems, both A and B are “local models”. A is a down-sample and blurry operator, and B should be an up-sample and sharpening operator. Because B has local support regions, the parameter of B_{ML} is sparse with fixed grids. The transpose convolutional neural network

can be trained as B_{ML} with an affordable computational cost. So B_{ML} is used in SR problems. In the sake of simplicity, B_{ML} is replaced by B .

2.1.4 Deep Convolutional Neural Network Based Residual Estimator

In conventional methods, both of likelihood and prior terms in equation (2.7) are human-designed. However, encouraged by the recent success of deep learning methods, especially in image reconstruction applications, a deep learning method is proposed to estimate the residual $R(x^t, y)$, from a large amount of training data. An MLP is a common neural network in deep learning. In MLP one neuron of one layer is fully connected to all neurons in the next layer. The amount of parameters is large, and it is easy to overfit. Some regularization on neural network parameters is needed to reduce the number of parameters and improve its generalization. Because images have localization and spatial invariant properties, convolutional neural network (CNN) is developed which adds a spatial parameter sharing regularization and learns a hierarchical feature bank with a local support region. In the proposed MARS method, a CNN based $R_\theta(x^t, y)$ is trained to approximate $R_{gt}(x^t, y)$, where θ is the set of all trainable parameters.

In $R_\theta(x^t, y)$, there are two inputs: previous solution x^t and observed y . Some researchers just use one, x^t or y , as the input of a CNN to estimate the residual $R_{gt}(x^t)$. Both of these networks produce competitive results. However, one intuitive question is: could we use both of x^t and y as inputs and how?

Some researchers apply the model augmented idea in [47] [2] to combine human-designed A and A^T into a neural network. Taking advantage of the known projection and back-projection models, the learning complexity of a neural network is reduced.

2.1.5 Residual Learning in MARS

The backpropagation is an efficient training algorithm in deep learning, which applies the chain rule of derivatives to layer-by-layer backward propagates the gradient of a loss function to

the trainable parameters in a gradient descent strategy. So a loss function is the source of informative feedbacks to adjust the parameters. Suppose in training data, ground truth x_{gt} is known, then the target residual at T iteration is also known, $R_{gt}^T = x_{gt} - x^{T-1}$, where T is the number of iterations. So the final target variable to learn is the residual R_{gt}^T , not the original target variable x_{gt} . By default, the loss function is an L2-norm,

$$\begin{aligned}
Loss_{ML}(x^{T-1}; y, \theta) &= |x_{gt} - (x^{T-1} + R_{\theta}(x^{T-1}, y))|^2 \\
&= |x_{gt} - x^{T-1} - R_{\theta}(x^{T-1}, y)|^2 \\
&= |R_{gt}^T - R_{\theta}(x^{T-1}, y)|^2
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
\theta^T &= \arg \min_{\theta} Loss_{ML}(x^{T-1}; y, \theta) \\
&= \arg \min_{\theta} |R_{gt}^T - R_{\theta}(x^{T-1}, y)|^2
\end{aligned} \tag{2.9}$$

Notice $x^T = x^{T-1} + R_{\theta}(x^{T-1}, y) = x^{T-1} + R_{\theta}(R_{\theta}(x^{T-2} + x^{T-2}, y), y) = \dots$ is a recursive function. The unrolled graph of the whole network is a deep feedforward network from y to x^T with shared parameters θ

2.2 Structure of MARS

MARS uses the iterative strategy of estimating residuals from a CNN based neural network. MARS has one bypass branch and three CNN blocks: likelihood, prior and reconstruction block. In the likelihood block, augmented with the human-designed A and A^* to project and back-project error between x^t and y .

2.2.1 Bypass Branch

A bypass branch directly sends x^t to x^{t+1} , which fits the equation of iterative methods equation (2.2). Furthermore, this bypass branch is also inspired by ResNets [24], which solves the vanishing gradient problem (VGP) while training a very deep neural network.

Vanishing Gradient Problem (VGP) Researchers demonstrated that compared with a shallow neural network, a deeper neural network has a higher representational capacity and potentially a higher-performance upper bound [40]. But simply stacking more layers does not improve even degrades the learning performance. The reason is that while doing backpropagation, the magnitude of gradients is reducing until smaller than the magnitude of noise. Then the training stops and no update for the parameters in front layers [25].

Solve VGP A solution of VGP is the multi-stage training method. Recently, ResNets showed that the bypass branches in neural networks are another solution of VGP. The bypass branches directly send the gradients into the parameters in front layers. Researchers [56] claim that ResNets is ensembles of relatively shallow networks and yields superior performance. For image reconstructions, usually x^t is a low frequency passed version of x_{gt} , and residual $x_{gt} - x^t$ is a high frequency passed version which has more high-frequency information.

$$\begin{aligned}x^t &= LPF_t(x_{gt}) \\x_{gt} - x^t &= HPF_t(x_{gt})\end{aligned}\tag{2.10}$$

where $LPF_t()$ is low-pass filters and $HPF_t()$ is high-pass filters.

2.2.2 Likelihood Block

A CNN $C_1(x^t, y)$ extracts non-linear features f_1^t from the back-projected reconstruction error $B(y - Ax^t)$ with a human-designed projection model A and a fixed or learned back-projection

models B . This block has two inputs, y , and x^t . Because y is directly observed from sensors or other data equipment, and x^t may have overfitting error, so y is much reliable than x^t . But x^t and the target x_{gt} are in the same space. The correction procedure is: first, project x^t to Ax^t ; second, measure error $y - Ax^t$; third, back-project/spread this error to all related unknown variables $B(y - Ax^t)$. Models A and B have augmented blocks in a neural network to provide refined features, so this structure is called model augmented. This branch is a model augmented likelihood branch or called the likelihood block in this dissertation.

2.2.3 Prior Block

A CNN $C_2(x^t)$ directly extracts features f_2^t from x^t . In some networks, the input of $C_2()$ is $\nabla_x F_2(x^t)$ [2], where $F_2(x^t)$ is a human-designed prior function, e.g. TV. Because the purpose of this dissertation is to demonstrate the functionality of likelihood block. We only use x as input of the prior block $C_2(x^t)$.

2.2.4 Reconstruction Block

The reconstruction block $C_3(f_1^t, f_2^t)$ is a multi-layer CNN with the likelihood and prior features f_1^t, f_2^t as inputs. There are two methods to combine these two groups of input features to a new feature f_3^t : one naive method is using the matrix sum operation directly add these two features, $f_3^t = f_1^t + f_2^t$; another is a matrix concatenation $[f_1^t, f_2^t]$ followed by a $C_{1 \times 1}()$, which is a 1×1 kernel size CNN layer. It is easy to show that the first operation is a special case of the second operation with fixed and sparse parameters in $C_{1 \times 1}()$. The MARS with the first operation is called Naive MARS, and the second operation is called MARS. The experiments are Chapters 3, 4 show that MARS has a better performance than Naive MARS.

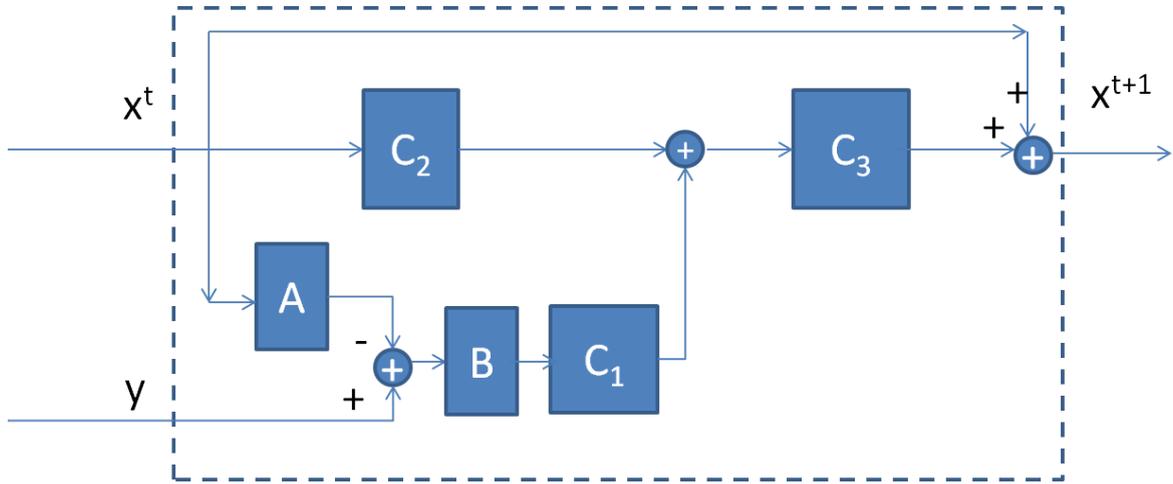


FIGURE 2.3: The structure of naive MARS.

$$\begin{aligned}
 f_1^t &= C_1(x^t, y) = C_1(A^T(y - Ax^t)) \\
 f_2^t &= C_2(x^t) \\
 f_3^t &= f_1^t + f_2^t \\
 f_4^t &= C_3(f_3^t) \\
 x^{t+1} &= x^t + f_4^t
 \end{aligned}
 \tag{2.11}$$

$$f_3^t = C_{1 \times 1}([f_1^t, f_2^t])
 \tag{2.12}$$

where $[\cdot]$ is the feature concatenate operation in neural network, and $C_{1 \times 1}(\cdot)$ is a 1×1 CNN layer with a linear activation function.

Algorithm 1: Naive MARS

Data: y, A, B **Result:** x^T

```
1 Initial solution  $x^0$ ;  
2 for  $t = 0, 1, \dots, T - 1$  do  
3   likelihood features:  $f_1^t = C_1(x^t, y) = C_1(B(y - Ax^t))$ ;  
4   prior features:  $f_2^t = C_2(x^t)$ ;  
5   combined features:  $f_3^t = f_1^t + f_2^t$ ;  
6   residual reconstruction:  $f_4^t = C_3(f_3^t)$ ;  
7   new solution:  $x^{t+1} = x^t + f_4^t$ ;  
8 end
```

Algorithm 2: MARS

Data: y, A, B **Result:** x^T

```
1 Initial solution  $x^0$ ;  
2 for  $t = 0, 1, \dots, T - 1$  do  
3   likelihood features:  $f_1^t = C_1(x^t, y) = C_1(B(y - Ax^t))$ ;  
4   prior features:  $f_2^t = C_2(x^t)$ ;  
5   combined features:  $f_3^t = C_{1 \times 1}([f_1^t, f_2^t])$ ;  
6   residual reconstruction:  $f_4^t = C_3(f_3^t)$ ;  
7   new solution:  $x^{t+1} = x^t + f_4^t$ ;  
8 end
```

2.3 Summary

This chapter introduced the proposed MARS neural network frame to iteratively solve ill-posed inverse problems. In inverse problems, there are two representations (x^t and y) in two different spaces (X and Y). MARS is trained to give an estimate x^T , which is close to ground truth

x_{gt} , meanwhile consistent with observation y . In MARS, a model augmented branch is designed to measure the difference between x^t and y and extract features from this difference. These reconstruction error features f_1^t is used to compensate the next estimate x^{t+1} . Chapters 3 and 4 describe the applications of MARS in medical image super-resolution and CT image reconstruction.

Chapter 3

Single Image Super-Resolution For Medical Images

3.1 Introduction

3.1.1 What is Super-Resolution

Super-resolution(SR) technique produces one high-resolution (HR) image from one or multi-frame of low-resolution (LR) images. Since the number of observations is much smaller than the number of unknown HR pixels, the SR problem is an ill-conditioned or ill-posed inverse problem, and it is hard to reconstruct a stable and visual pleased HR image.

In equation (1.1), the latent variable $x \in \mathbb{R}^{d_1}$ and the observation $y \in \mathbb{R}^{d_2}$, where $d_2 \ll d_1$. In the SR problem, A is generally modeled as a down-sample and blurry matrix, ϵ is an additive white Gaussian noise. The pseudoinverse inverse A^+ is unstable and generates artifacts in high-resolution results.

To solve the SR problem, there are two methods to add more constraints and reduce the ill-posedness of SR problems. One conventional method is multi-frame super-resolution (MFSR) which uses multi-frame of related LR images to increase the number of observed data d_2 and provide more constraints on HR image, [18] [17]. For instance, some recent flagship cell phones are equipped with two/three LR cameras, in order to simultaneously take multiple LR pictures at the same scene, and then fuse these LR pictures at a sub-pixel level. Another method is

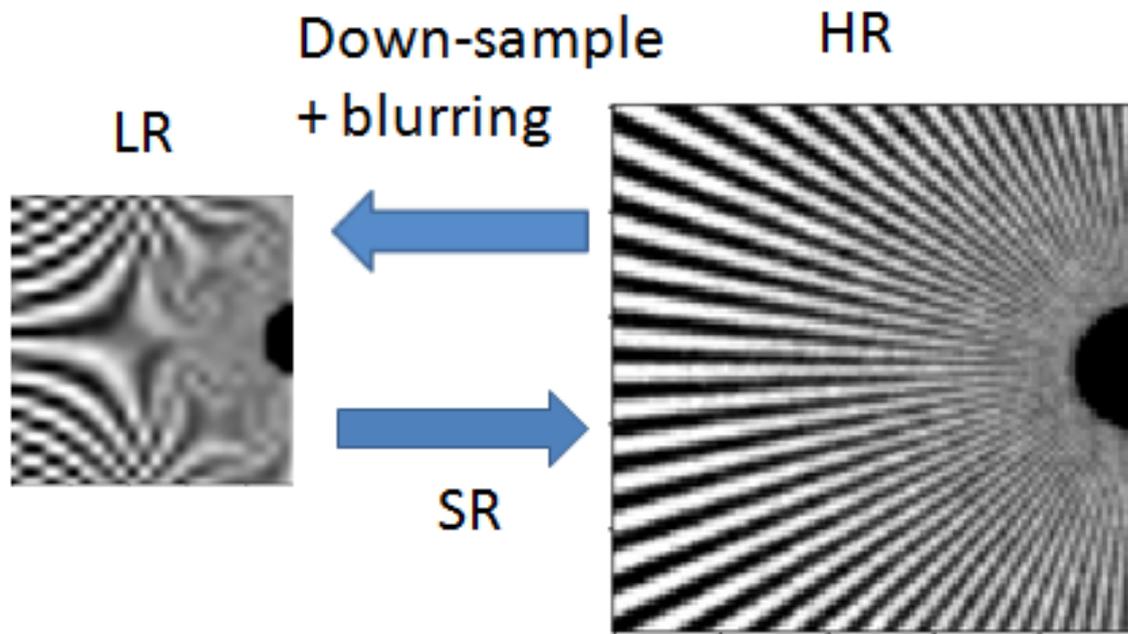


FIGURE 3.1: An illustration of a SISR problem. The forward process from HR to LR are down-sampling and blurring. Super-resolution is a inverse process to reconstruct HR from LR.

Single Image Super-Resolution (SISR) with some prior information from general models (like TV, L1-norm) or data-driven models (example based, deep learning). SISR is a hot topic in the computer vision community since for some situations, only one LR image can be provided to generate an HR image. For example, in many cases, only one copy is provided when applying SR to a historical picture. In another example, a low-end surveillance camera system can only give one observed LR image for the sake of cost-efficiency.

3.1.2 Why Use SISR for Medical Images

For medical imaging applications, HR images are desired to provide more information about patients to help doctors or radiologists to make decisions. Furthermore, machine learning-based medical diagnosis techniques have been emerging in recent years in medical imaging to increase diagnosis accuracy and reduce medical costs. As an important pre-processing step to

enhance the resolution of medical images, the SR technique is used to improve the performance of diagnosis machines [46] [57] [30].

When compared with MFSR, SISR is widely used in medical imaging, especially in X-ray imaging, due to the following two reasons: 1) SISR can be used to reduce the risks of cancer caused by x-radiation. For example, for X-ray imaging, some studies [43] [7] show a theoretical link between X-ray radiation and risk of cancers; therefore, the low dose of X-ray is often desired for X-ray imaging, which produces medical images with lower resolution. Similarly, Multi-frame X-ray imaging is usually restricted as it requires more X-ray exposure time and thus brings a higher risk of radiation-related diseases. SISR provides an effective way to obtain relatively high-quality medical images while reducing the patients' exposure to x-ray radiation. 2) SISR can be used to improve the cost-effectiveness of medical imaging. The image resolution and the cost of hardware in a medical imaging system are often in a nonlinear relationship. For example, machines that use 100-micron detector cell X-ray panels (higher-resolution or high-res) cost much higher than that use 200-micron detector cells (lower-resolution or low-res). Due to economic considerations, lower-res machines still have a lot of customers, especially in developing economies. In order to balance the quality of images and the cost of medical imaging hardware, a strategy is to use LR hardware with a lower cost to first obtain LR images and then enhance these low-resolution images by software, like SISR, to provide similar performance of HR hardware. This is the motivation of our SR project to apply SISR to increase the resolution and quality of LR X-ray images and approaching images from HR X-ray hardware.

3.1.3 Organization of Chapter 3

The rest of this chapter is organized as follows: Section 2.2 reviews related works of SISR, before and after deep learning; Section 2.3 describes how to use the proposed MARS method for medical image SISR, particularly for X-ray imaging; Section 2.4 has three SISR experiments: the first preliminary experiment shows the upper-bound performance of MARS SISR; the second natural image SISR experiment demonstrates that MARS has a comparable performance to state-of-the-art deep-learning-based SISR methods; the third chest X-ray image SISR

experiment shows the performance of MARS SISR, meanwhile emphasizing the contributions of the MARS method. Section 2.5 is the conclusion of medical image SISR with MARS.

3.2 Related Work

SISR techniques are driven by the development of deep learning techniques. Due to the superior performance of Deep learning, almost all winners of SISR competitions or tasks were using deep-learning-based methods after 2015 when Deep learning was [13]. Therefore, we divide the SISR literature into two categories based on whether deep-learning is applied or not: Non-deep-learning-based and deep-learning-based, or prior to deep learning and post deep learning. This review of related SISR literature primarily focuses on Deep learning-based methods, in order to guide the design of the proposed MARS method.

3.2.1 Before Deep Learning

Traditionally, interpolation techniques (e.g., cubic splines) have been used by many researchers to increase resolution. However, the resulting images are often blurry creating problems with accurate diagnoses. To increase sharpness, gradient enhancement techniques such as [23] were used to enhance edges in the increased-resolution image. Similarly, total variation (TV) and sparseness priors have also been used for the same purpose [44]. While useful in improving the sharpness of edges, these techniques are not very effective in improving textural details. To solve this problem, a number of learning based techniques were proposed, e.g. [14][10]. The learning-based techniques use a database of high-res images (and their low-res versions) to learn a predictive relationship between a low-res image patch and its corresponding high-res image patch; this relationship is used to generating high-res patches for low-res images whose high-res versions are not available. This approach can be quite effective but it generally requires a large database of high-res images whose low-res versions are very similar to the low-res images to be enhanced. The approach also raised concerns from some radiologists who

questioned whether high-res image patches from the patient database can or should be used to estimate high-res image patches for another patient.

Meanwhile, an effective alternative was proposed in [29]. This approach, known as self-exemplar SISR. The basic idea here is closely related to that of self-similarity or fractals: in a natural image, the same patterns can often be observed at different scales. The main advantage of this approach is that it does not require a database of training/example images, potentially increasing acceptance by radiologists. Based on this self-exemplar concept, Huang et al. [28] introduced more transformation to find potential similar patches from just one single image.

Yang et al. [62] applied coupled dictionary learning with sparse coding prior to SISR, based on an assumption that a pair of HR and LR patches should have the same sparse code. This assumption is used as a regularization while alternatively training a couple of HR dictionary and LR dictionary. From the view of deep learning, the LR dictionary is a linear coder with one-layer CNN, and the HR dictionary is a linear decoder with one-layer CNN. So this method can be implemented as a two-layer auto-encoder with sparse code. This auto-encoder is shallow and wide with 1024 codes. It has potential value to be combined with a deep neural network as a “wide and deep” structure [11].

3.2.2 Deep Learning

A deep CNN [41] learns hierarchical representations of images from a large-size training data set. Encouraged by many deep learning success stories, SRCNN [13] uses only 3 CNN layers and shows superior performance than previous non-deep learning method. SRCNN builds a benchmark framework for end-to-end style SISR methods: feature extraction, feature non-linear mapping, and reconstruction. The SISR methods based on this framework have four questions, arise and inspire improvements of SISR:

Where to obtain high-quality training data? The quality of the data set is crucial to the results of SISR. Researchers collect images from the Internet as a data set. Set5, Set14 [64], BSDS100 [5], Urban100 [28] are public high quality training data or benchmark testing data.

DIV2K [4] is used in the SISR competitions NTIRE (CVPR 2017/2018) and PIRM (ECCV 2018). Imagenet [12] is a large image data set, which has 14 million manually annotated images. Data augmentation is another strategy to increase the number of data set. Common data augmentation methods are random crops, rotation, and flip [37]. Data augmentation is one of the seven methods to improve SR in [54].

Use data augmentation for medical images? A simple answer is “No”. A deep neural network learns non-linear mapping (correlation) functions between input and the target x_{gt} . The input could be the original input or pre-processed input. For example, in SISR, SRCNN [13] uses interpolated HR images $x^0 = bicubic(y, 4)$ as the input, where $bicubic(\cdot, 4)$ is the 4x bicubic up-sample, and trained a CNN to learn the mapping functions between x^0 and x_{gt} . The advantage is that the mapping function is simpler because x^0 and x_{gt} are in the same space X . The disadvantage is information might be removed by the bicubic interpolation. LapSRN [38] directly learn the mapping functions between y and x_{gt} . The advantage is y has all the information. The disadvantage is a non-linear up-sample operate B should be learned from a training data set. DBPN [22] uses both of x^0 and y as inputs and iteratively reconstruct HR and LR images. This structure is similar to [3], which applies a deep neural network to iteratively solve a primal-dual problem. In the proposed MARS structure, both x^t and $y - Ax^t$ are inputs. So features in the spaces X and Y are extracted and mapped to x_{gt} .

The deeper, the better? One trend of deep learning is that researchers are applying deeper neural network to achieve superior performance. A review paper [63] goes through the development of deep learning SISR methods. This paper also compares the deep learning structures of recent SISR methods. One trend of deep learning SISR is the deeper, the better. Kim et al. proposed VDSR [34], which is a 20-layer deep VGG-style [49] SISR model. Kim et al. further introduce a deep 16x recursive neural network DRCN [35] which shares parameters among layers, in order to use a few parameters to build a deeper neural network. DRRN [52] and MS-LapSRN [39] build very deep neural networks on this recursive neural network structure. The proposed MARS is also a recursive neural network with shared parameters, see Figure 2.1.

How to train a deep neural network? Because of the vanishing gradient problem, it is difficult to train a plain VGG-style deep neural network. ResNet [24] and DenseNet [27] show that bypass links among CNN layers can speed up learning convergence and boost performance. One reason is that these bypass links directly back-propagate gradients to deeper layers. ResNet has shallow and deep these two kinds of paths. The shallow paths learn low-frequency information, and deep paths learn high-frequency residual information. So it is called Residual learning Network.

Progressive learning is one technique to train a deep neural network. The outputs of intermediate layers are raw estimate x^t of HR x_{gt} . The loss functions between x^t and x_{gt} are added to the total loss function, as regularizer on the intermediate representation of an image. These intermediate loss terms guide the reconstruction of x^t close to a reasonable solution x_{gt} . For example, an 8x MS-LapSRN generates 2x/4x/8x reconstructions progressively and uses all loss function at different scales (2x/4x/8x).

The proposed MARS uses residual learning and progressive learning techniques to train a very deep recursive neural network.

3.3 Contributions and Proposed Methods

3.3.1 Contributions

The contributions of this chapter are listed below:

1. Used the proposed model augmented recursive neural network (MARS) in medical image single image super-resolution (SISR).
2. Addressed a class imbalance problem in medical image SISR to improve the resolution enhancement.
3. Analyzed data augmentations for medical images. Used a merged training data set with 90% original data and 10% augmented data.

3.3.2 MARS for Medical Image SISR

The proposed model augmented recursive neural networks (MARS) in Algorithm 2 can be used for medical image SISR. The MARS is originally motivated by gradient descent (GD) algorithm to iteratively solve an ill-posed inverse problem Equation (2.3). As described in 2.2, MARS has one bypass branch and three CNN blocks (likelihood, prior and reconstruction).

3.3.2.1 Bypass branch

One bypass branch passes a current high-resolution estimate x^t to a next iteration/stage estimate x^{t+1} . The target output of MARS is the difference between the ground truth x_{gt} and x^t . This new target is called residual in boosting algorithms that convert a group of weak learners to a strong learner in order to reduce bias and variance of estimation. The bypassed signal x^t keeps low-frequency information and the residual R^t has high-frequency information which is the learning target of our neural network. The high-frequency information provides more detail information to help doctors to make medical decisions .

3.3.2.2 Likelihood Block

There are two image representations with inconsistency between them: an observed low-resolution image y , and an imperfect estimated high-resolution image x^t . They cannot be compared directly since they are at different resolution levels. A project model A is build to transform a high-resolution image to a low-resolution image, and a back-projection model B does the inverse/back projection.

Fixed Down-sample and Learned Up-sample. As mentioned in Chapter 2, both projection and back-projection models can be fixed human-designed models or learned from training data. The fixed models make methods simple and low-cost, and the learned models increase the learning capacity of methods with a higher computational cost. Considering A is directly build

from the knowledge of problems, and B is indirectly built from A and some statistic assumption, e.g. noise is i.i.d. Gaussian, A is more accurate than B . To balance the computational cost and performance of MARS, a fixed A and a learned back-projection model U is used in MARS SISR, where U is a one layer transpose CNN.

By using these two models A and U , MARS measures and reduces the inconsistency (reconstruction error) between y and x^t . Firstly, projects x^t to low-resolution Ax^t from a down-sample function A . Secondly, measures the low-resolution inconsistency $y - Ax^t$. Thirdly, back-projects the low-resolution inconsistency to high-resolution $U(y - Ax^t)$. Finally extracts features of the high-resolution inconsistency and trains a CNN as a regressor to map these features to the target residual R^t .

3.3.2.3 Prior Block

The traditional prior models are usually sparsity-driven energy functions of x , such as Tikhonov, total variation (TV) and generalized Gaussian Markov random field (GGMRF). These functions rely on some general sparse models of the image gradient. For example, the probability distribution of image gradients is Gaussian or Laplacian distribution, then the corresponding regularization/ prior model is L2-norm or L1-norm of image gradients. However these sparse prior models emphasize the generalization for all of the natural images, then they provide weak constraints to reconstruction. Data-driven models, e.g. deep learning, can learn more complex mapping functions in a large amount of training data [55] [13]. In proposed MARS, a CNN block is used to learn a prior model based on a natural or medical image data set.

3.3.2.4 Reconstruction Block

The reconstruction block in MARS is a CNN block, whose works are:

1. using a 1x1 CNN layer to compressed features $[f_1, f_2]$ from the likelihood feature block and the prior feature block. Comparing to a popular VGG-style [49] 3x3 CNN layer, this 1x1 CNN layer only has 1/9 trainable parameters.

2. doing feature embedding with a 5-layer CNN block.
3. the final reconstruction layer is a 5x5 CNN layer with the linear activation function. Actually, this layer learns a linear regression between the embedded features and residual R^t .

3.3.3 Multi-channel MARS for SISR

In the SISR problem, x is a high-resolution image and y is an observed low-resolution image. For medical images, x and y usually are grayscale images, which have only one intensity channel; for natural images, x and y are RGB three-channel images. To have the same structure for both medical and natural images, RGB images are transformed into YCbCr format, then MARS is used on the brightness Y channel and bicubic interpolations are used on the CbCr channels. After super-resolution, the resolution enhanced YCbCr images are converted back to RGB format. So for all SISR applications, x and y are one channel images.

The original MARS for SISR is applied to grayscale images which are a one-channel representation of images. The human-designed projection model A and back-projection model B connect one-channel images x and y . One intuitive question is could one-channel MARS be expanded to multi-channel MARS (MMARS). In other word, could these one-channel models A and B be used on multi-channel features? The idea of coupled features at low and high resolution is used in [62] that is if $f_x(x)$ and $f_y(y)$ are coupled filter banks of x and y which are trained together, these two multi-channel signals could have correlated representation coefficients. CNN could be used to learn these coupled linear or non-linear filter banks $f_x(x)$ and $f_y(y)$ from training data, in Equation (3.1). Two CNN blocks are trained as coupled feature extractors of x and y .

$$\begin{aligned}
 f_x^t &= C_x(x^t) \\
 f_y &= C_y(y)
 \end{aligned}
 \tag{3.1}$$

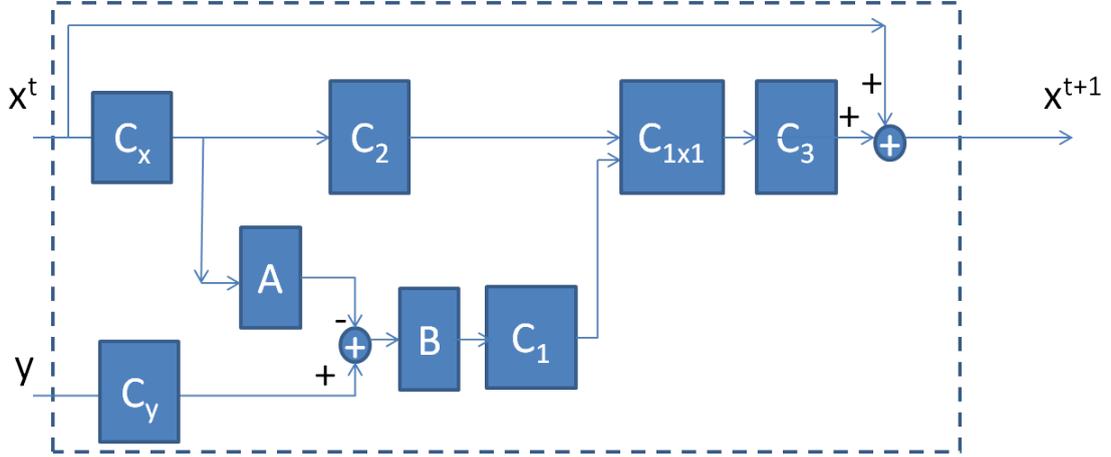


FIGURE 3.2: The structure of MMARS.

where $C_x(x^t)$ and $C_y(y)$ are CNN based feature extractors, f_x^t are features of high-resolution images, f_y are features of low-resolution images. These two features are coupled by A and B which means $B(Af_x(x) - f_y(y)) \approx 0$.

$$\begin{aligned}
 \tilde{f}_1^t &= C_1(f_x^t, f_y) = C_1(B(f_y - Af_x^t)) \\
 \tilde{f}_2^t &= C_2(f_x^t) \\
 \tilde{f}_3^t &= C_3([\tilde{f}_1^t, \tilde{f}_2^t]) \\
 x^{t+1} &= x^t + \tilde{f}_3^t
 \end{aligned} \tag{3.2}$$

where $C_1(), C_2(), C_3()$ are three CNN blocks, \tilde{f}_1^t are features from back-projected reconstruction features $B(f_y - Af_x^t)$ call likelihood features, \tilde{f}_2^t are features from previous solution f_x^t called prior features. \tilde{f}_3^t are residual reconstruction at iteration t .

Algorithm 3: MMARS

Data: y, A, B **Result:** x^T

- 1 Initial solution x^0 ;
 - 2 features of y : $f_y = C_y(y)$;
 - 3 **for** $t = 0, 1, \dots, T - 1$ **do**
 - 4 features of x^t : $f_x^t = C_x(x^t)$;
 - 5 likelihood features: $\tilde{f}_1^t = C_1(f_x^t, f_y) = C_1(B(f_y - Af_x^t))$;
 - 6 prior features: $\tilde{f}_2^t = C_2(f_x^t)$;
 - 7 residual reconstruction: $f_3^t = C_3([\tilde{f}_1^t, \tilde{f}_2^t])$;
 - 8 $x^{t+1} = x^t + f_3^t$;
 - 9 **end**
-

3.3.4 Progressive Learning Structure of MARS

Many SISR methods use one step up-sample to directly reconstruct y from x , which makes learning mapping function $R(x^t, y)$ for a large up-sample scale factor (e.g. 4 or 8) more difficult. The multilevel progressive learning structure is widely used in image reconstruction applications in a coarse-to-fine fashion. A MARS is designed to progressively upsample 2X higher resolution images at each resolution level from y to x . For example, if the scaling factor is 4, the intermediate images z_1, z_2 are generated with up-sample scaling factor 2 and 4.

$$\begin{aligned} z_1^{t+1} &= MARS_2(z_1^t, y) \\ z_2^{t+1} &= MARS_2(z_2^t, z_1^{t+1}) \\ x^{t+1} &= MARS_2(x^t, z_2^{t+1}) \end{aligned} \tag{3.3}$$

where $MARS_2$ is a proposed neural network with a constant up-sample factor 2. The initial solution of z_1, z_2 are bicubic interpolations from y . The synthetic ground truth of z_1, z_2 are

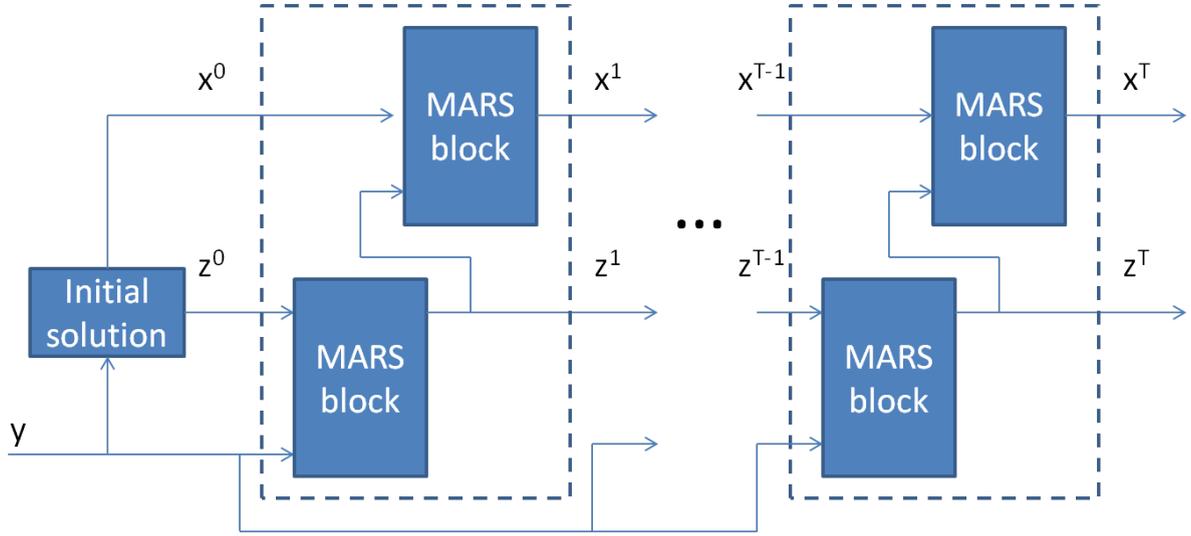


FIGURE 3.3: The structure of progressive learning. z is an intermediate resolution image.

bicubic down-samples from x_{gt} .

$$\begin{aligned}
 z_1^0 &= \text{bicubic}(y, 2) \\
 z_2^0 &= \text{bicubic}(y, 4) \\
 z_{1gt} &= \text{bicubic}(x_{gt}, 1/4) \\
 z_{2gt} &= \text{bicubic}(x_{gt}, 1/2)
 \end{aligned} \tag{3.4}$$

where $\text{bicubic}(a, b)$ is the bicubic function with input a and scaling factor b .

Multi-resolution Loss. The intermediate loss functions between reconstructions z_1^T, z_2^T and synthetic ground truth z_{1gt}, z_{2gt} can be added into the original loss function Equation (2.8).

$$\text{Loss}(\theta, x^{T-1}, y) = l(x_{gt} - x^T) + \gamma \sum_{j=1}^2 l(z_{jgt} - z_j^T) \tag{3.5}$$

where $l()$ is the squared L2-norm or Charbonnier penalty function Equation (3.6), γ is a balancing hyper-parameter. Notice the first term is the real target of the learning, other terms are just guides of intermediate representations, in order to speed up the learning convergency and

regularize parameters in the intermediate CNN layers. So $\gamma = 0.1$ forces the learning focus on the final result x^T .

There are two reasons to add the intermediate loss functions: first, without these intermediate loss functions, it is difficult to control the intermediate reconstruction results z_1^t, z_2^t . These intermediate loss functions serve as extra regularization to guide z_1^t, z_2^t . Second, while training a deep neural network, the vanishing gradient problem degrades the gain of simply increasing the number of layers. The intermediate loss functions directly go into the intermediate layers of a deep neural network and supply a stronger gradient to the parameters in front layers. For example, if the up-sample scaling factor is 8, the proposed neural network has MARS_2 blocks, and each MARS_2 has 10 CNN layers. The entrance of the first intermediate loss function $l(z_{1gt}, z_1^T)$ is the layer 10, and the entrance of the original loss function $l(x_{gt}, x^T)$ is the layer 30. Assume after passing a layer, the magnitude of a gradient is half of the previous one. So in layer 1, the magnitudes of the gradients from the original and the first intermediate loss functions are 2^{-29} and 2^{-9} which means the second gradients are 2^{20} larger than the first gradients.

3.3.5 Special Problems for Medical Image SISR

The proposed MARS SISR method can be used for natural and medical images. However, the difference between natural and medical images causes three special problems for medical image SISR should be considered in this dissertation. One contribution of this dissertation is solving these problems in MARS method and improving results.

3.3.5.1 Class Imbalance Problem

Compared to natural images, medical images have a class imbalance problem, since “smooth pixels” are much more than “texture pixels” in medical images. For example, in a chest X-ray image, the background is smooth scanning data of air. But in a picture of zebras in a wild environment, the background is grass or forest with textures. A SR neural network or regressor is trained as a “smooth” region regressor, not for texture regions. But texture regions provide more diagnosis information to doctors, like bone fractures and breast cancer.

The target of the SR regressor is to learn residuals between bicubic and target HR images, which is a high-pass version of target HR images. The signal to noise ratio (SNR) in smooth regions is lower than SNR in the texture region. After high pass filtering, the SNR gap between texture and smooth regions increases. This means it is worth to train a sophisticated regressor/neural network to learn mapping functions within texture regions. For smooth regions, the bicubic interpolation works well.

The mathematic reason for the classes-imbalance problem is that all pixels have equal weights in the loss function, so a larger class has a larger accumulative weights. An intuitive assumption is that each class should have equivalent accumulative weights. One re-balancing method is to sub-sample over-represented smooth class or over-sample under-represented but more important texture class. Another re-balancing method is each class multiply a re-balancing scalar in the loss function, focusing on errors from texture pixels, which have more interesting medical information.

3.3.5.2 Should Use Training Data Augmentation?

A simple answer is “No”. The training data augmentation is a popular technique to generate more synthetic training images by randomly shifting, rotation, scaling and flip. Many state-of-the-art deep learning methods for natural images tasks [37] [39] use the data augmentation to prevent overfitting and improve testing generalization. But in our medical image SISR experiments, a heavy performance drop is observed after using the data augmentation. The reason is that objects in natural images have highly random positions, directions, and scales. After the data augmentation, the synthetic training data can cover more patterns with different directions and scales. But it is a different story for medical images. For example, the design of X-ray equipment and the procedure manuals in clinics force the chest X-ray images are pre-aligned with a similar location, direction, and scale. To reconstruct patches of rib and shoulder bones, similar patterns can be found around the same location in other original training images. This pre-registration of medical images makes the original medical images are better than the augmented medical images for SISR applications. In addition, image rotation and resizing degrade

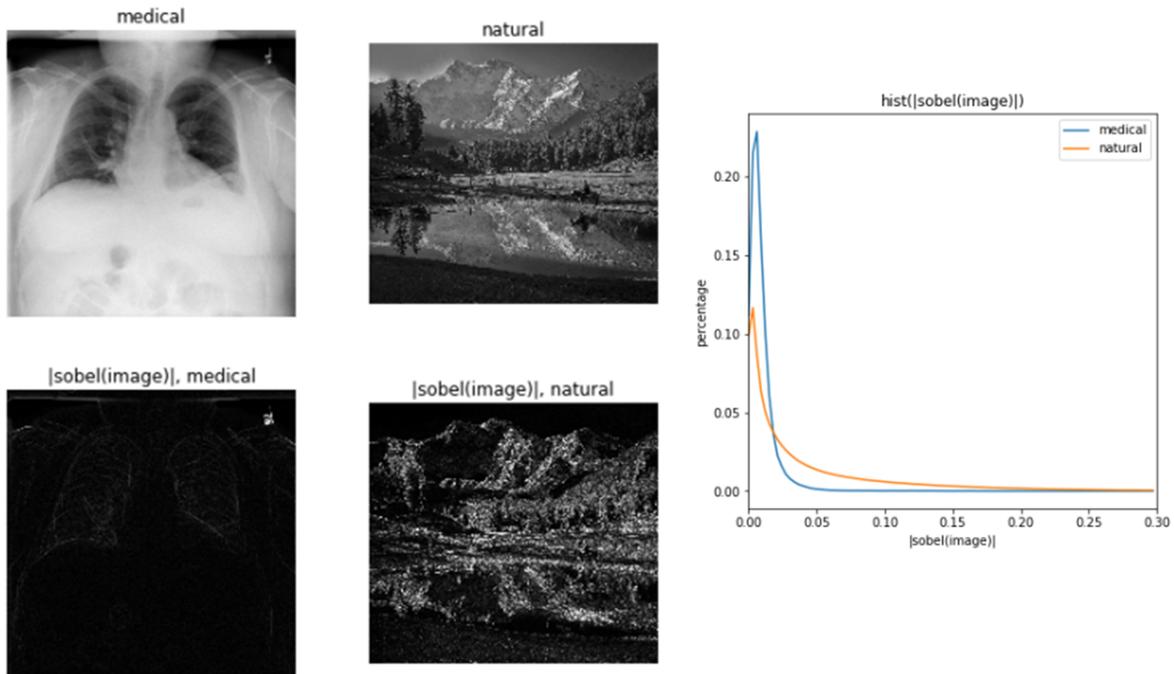


FIGURE 3.4: Compare high-frequency patterns in medical images and natural images. Solbel operator is a high-pass edge detection function. The absolute value of Solbel operator is used to measure the magnitude of high-frequency components. This picture shows natural images has more high-frequency patterns.

the resolution of training images. The proposed MARS is trained by a merged training data: 90% from the original training data and 10% from the augmented training data.

3.3.5.3 Use L1-norm Loss Function for Medical Image SISR

In image reconstruction applications, the L2-norm loss function usually generates blurry images to have small mean squared error (MSE). Comparing to natural images, there are more smooth pixels in medical images, so this blurry problem is more serious in medical images while using L2-norm loss function. In order to get sharp edges and more textures in images, researchers use the L1-norm loss function [42] or Charbonnier penalty [39], which is a robust version of L1-norm.

$$Loss_{Char}(\theta, x^{T-1}, y) = \rho(R_{gt}^T - R_{\theta}(x^{T-1}, y)) \quad (3.6)$$

where $\rho(x) = \sqrt{x^2 + \epsilon_{char}^2}$ is Charbonnier penalty function with a small constant hyperparameter ϵ_{char} .

3.4 Experimental Results

Three SISR experiments demonstrate the performance of the proposed MARS SISR method.

1. A preliminary experiment with human-crafted synthetic training data proves if having enough and good training data, MARS SISR can break the limitation of Nyquist frequency and reconstruct high-frequency patterns even from aliasing patterns. This experiment shows the upper-bound performance of MARS SISR.
2. A natural images SISR with public training/testing data. Because state-of-the-art SISR methods are focusing on natural image SISR, it is not fair to these natural image methods to compare SISR performance with the proposed MARS methods, in a new environment of medical images. For example, the hyperparameter fine-tuning is crucial for SISR performance, and these methods search hyperparameters for natural images, not medical images. In addition, three special problems for medical image SISR are concerned in MARS, not in these methods. It is a fair play game for both MARS and these methods within the same natural image environment.
3. A chest X-ray images SISR shows that MARS can dramatically enhance the resolution of medical images.

3.4.1 A preliminary Experiment

A preliminary experiment shows that if it has enough good quality training data, it is possible to use MARS dramatically boost SISR results. This experiment only uses one sun-shape phantom

X-ray image Figure 3.5(A), which is provided by GE Healthcare. The testing patch is one 64x64 center patch from the left side of the image. The testing patches are randomly cropped from the right side of the images. The raw training patches are resized to 64x64, randomly rotated within [180,270] degree. After the rotation, the training patches have many similar patterns with the testing patch. 1000 synthetic training patches are generated. The LR patches are generated by a 4x down-sample bicubic function. The initial solution is also from a 4x up-sample bicubic function.

The neural network is a simple version of MARS: the reconstruction is directly from 1x to 4x, so A and B are 4x down/up-sample bicubic interpolation; the number of layers in a CNN block is 2; the number of stages is 1; 100 epochs for training. The results are shown in Figure 3.5. The PSNR of the 4x bicubic interpolation is 16.56 dB and the PSNR of the MARS result is 29.92 dB. So MARS boosts 13.36 dB from the bicubic interpolation. Because the spatial resolution is too low to satisfy the Nyquist frequency, the aliasing artifacts can be seen in the bicubic interpolation. The MARS result breaks the limitation of the Nyquist frequency and reconstructs high-frequency information from low-frequency patterns, even aliasing patterns.

3.4.2 Natural Image SISR

Data Set: The training image data set is DIV2K data set [4], which is used in the SISR competitions NTIRE (CVPR 2017/2018) and PIRM (ECCV 2018). DIV2K includes 800 high-quality natural images. The testing data is a public benchmark dataset Set14 [64], which has 14 natural images. The high-resolution patch size is 128x128. The down-sample scale factor is 4x. In this experiment, the LR image y is 4x bicubic down-sample image from the ground truth image x_{gt} . The initial solution x^0 is 4x bicubic up-sample image from y . The intermediate images $z_{gt} = bicubic(x_{gt}, 0.5)$, and $z^0 = bicubic(y, 2)$.

Configuration of MARS. The default configuration of the proposed method is: the number of filters is 64; batch size is 16, and each epoch has 100 batches; the epoch number is 300; the optimizer is Adam with the initial learning rate is $\alpha = 0.001$, and after every 30 epochs

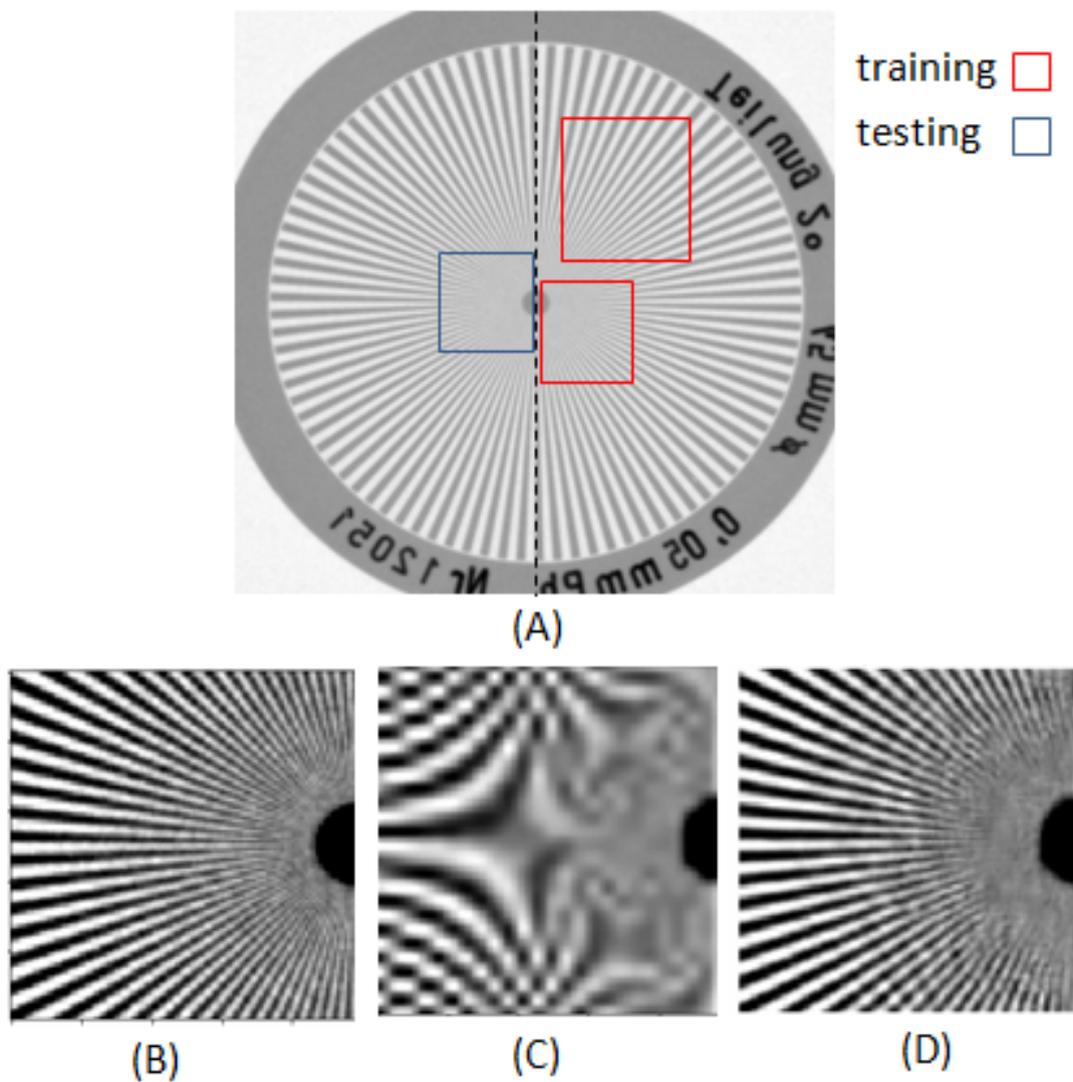


FIGURE 3.5: A preliminary Phantom X-ray image SISR

(A) training and testing data from a phantom X-ray image, (B) is ground truth (C) is a bicubic interpolation from 4x down-sample LR (PSNR=16.56dB) (D) is the proposed MARS result (PSNR=29.92dB). This experiment shows the upper-bound performance of MARS SISR. With good training data, MARS SISR can recover high-frequency patterns from low-resolution patterns with strong aliasing artifacts.

$\alpha = \alpha/1.2$; loss function is Charbonnier penalty function with $\epsilon_{char} = 0.01$. The PSNR and SSIM of testing data are performance metrics to measure the quality of testing results. 3 stages/iterations multi-channel MARS is used. The number of channels of all CNN layers is 64. MMARS uses the pyramid structure in Figure 3.3. The intermediate image z^t and HR images x^t are progressively reconstructed. The multi-resolution loss is used.

$$\begin{aligned} y &= \text{bicubic}(x_{gt}, 0.25) \\ x^0 &= \text{bicubic}(y, 4) \end{aligned} \tag{3.7}$$

TABLE 3.1: Configurations of natural image SISR methods. Recon. means reconstruction strategy. Direct recon. is one-step up-sampling from LR to HR; progressive recon. only up-sample 2x at each stage. RL is ‘‘Residual Learning’’, which means there are bypass connections between CNN layers, globally or locally.

Method	Input	Recon.	Depth	Parameters	RL	Loss
SRCNN [13]	Bicubic	Direct	3	57k	N	l_2
LapSRN [38]	LR	Progressive	24	812k	Y	Charbonnier
MS-LapSRN [39]	LR	Progressive	84	222	Y	Charbonnier
MMARS	LR + Bicubic	Progressive	78	788k	Y	Charbonnier

The MARS SISR codes are built on Tensorflow [1]. with a GPU Geforce 1080(8Gb memory). PSNR and SSIM [59] are used to evaluate SISR performance. Visual comparisons are shown in Figure 3.6

3.4.2.1 Comparisons with state-of-the-art SISR methods

The proposed MARS SISR is compared to the bicubic interpolation and state-of-the-art method MS-LapSRN [39].

Table 3.2 shows MMARS has higher PSNR but lower SSIM. This experiment proves that MMARS has comparable performance to state-of-the-art SISR methods in a natural image environment. The next step is to use MMARS SISR with a medical image data set.

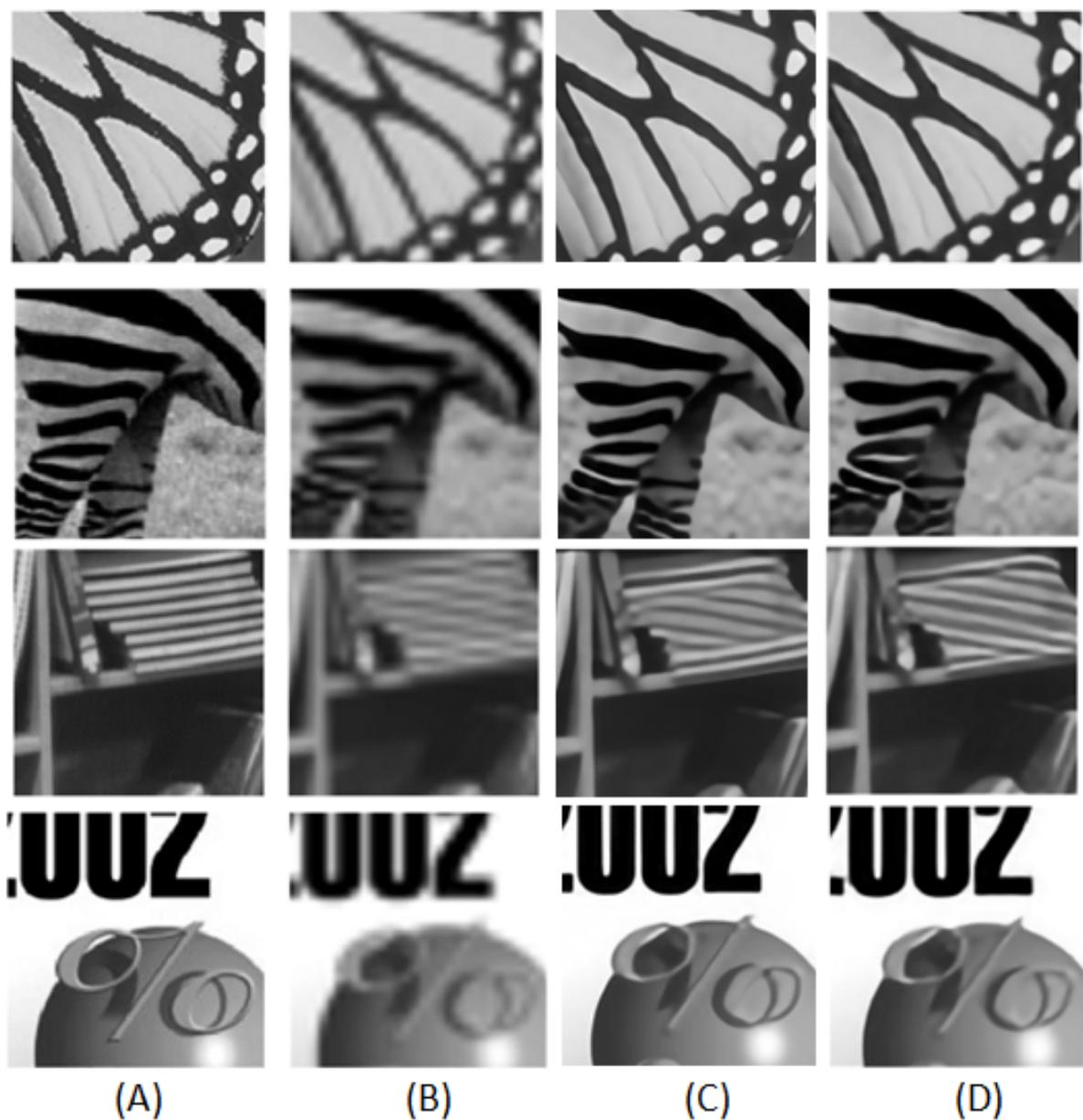


FIGURE 3.6: SISR results: (A) is ground truth, (B) is bicubic interpolation (C) is MS-LapSRN [39] result (D) is the proposed MMARS result.

TABLE 3.2: Average PSNR and SSIM for natural image SISR methods. Other methods' PSNRs and SSIMs is from [39]. The red numbers are highest PSNR or SSIM.

Method	PSNR	SSIM
Bicubic	25.85	0.660
SRCNN [13]	27.52	0.753
LapSRN [38]	28.19	0.772
MS-LapSRN [39]	28.26	0.774
MARS	27.73	0.744
MMARS	28.49	0.758

TABLE 3.3: Training/ testing time of LapSRN and MARS for SISR.

For a SISR problem, from 128x128 to 512x512, the testing time of one image is $0.0069 \times 16 = 0.1104$ sec/image.

Method	HR Patch size	Depth	# of parameters	Train (sec/patch)	Test (sec/patch)
LapSRN [38]	128x128	24	812k	N/A	0.0078
MARS	128x128	78	788k	0.0145	0.0069

3.4.3 Chest X-ray image

3.4.3.1 Data Set and Platform

The data set is a Chest X-ray image data set [58]. There are 10000 images that are randomly split to 8000 training images and 2000 testing images. These images are already normalized to 0-255. The original image size is 1024x1024. The used training/testing images are 2x down-sample of the original images. The HR patch size is 128x128. The down-sample scale factor is 4x. In this experiment, the LR image y is 4x bicubic down-sample image from the ground truth image x_{gt} . The initial solution x^0 is 4x bicubic up-sample image from y . Uses the same configuration of MARS in the last section.

3.4.4 Results and Analysis

The analysis of results in Table 3.4:

TABLE 3.4: Average PSNR/ SSIM of X-ray image SISR

The red numbers are highest PSNR or SSIM.

Model	PSNR	SSIM
Bicubic	38.68	0.884
MMARS 5 stages	41.54	0.925
MMARS 3 stages	41.46	0.925
MMARS 3 stages + no prior	41.37	0.924
MMARS 3 stages + no likelihood	41.03	0.921
MMARS 3 stages 4x	40.61	0.915
MMARS 3 stages + no mask	40.13	0.913

TABLE 3.5: PSNR/ SSIM of Figure 3.9

Bicubic	MMARS without likelihood	MMARS
24.94/ 0.7716	25.72/ 0.8737	27.53/ 0.9161
21.39/ 0.7796	23.18/ 0.8988	24.62/ 0.9194
25.00/ 0.9093	24.43/ 0.9319	25.57/ 0.9493

1. Model augmented structure improves the performance of the proposed neural network. The differences are PSNR $41.46 - 41.03 = 0.43dB$ and SSIM $0.925 - 0.921 = 0.004$.
2. Prior block also improves results. The differences are PSNR $41.46 - 41.37 = 0.09dB$ and SSIM $0.925 - 0.924 = 0.001$.
3. More stages,better performance. The 5 stage result is slightly better than 3 stage result. The differences are PSNR $41.54 - 41.46 = 0.08dB$ and SSIM $0.925 - 0.925 = 0.000$.
4. Progressive reconstruction is better than direct reconstruction. The differences are PSNR $41.46 - 40.61 = 0.85dB$ and SSIM $0.925 - 0.915 = 0.010$.
5. Balancing mask improves results. The differences are PSNR $41.46 - 40.13 = 1.33dB$ and SSIM $0.925 - 0.913 = 0.012$.

An experiment shows the performance of the balancing mask. The balancing mask is showed in Figure 3.7. The results are the upper row in Figure ???. The lower picture is profiles from the result. This experiment demonstrates that the balancing mask makes the result sharper.

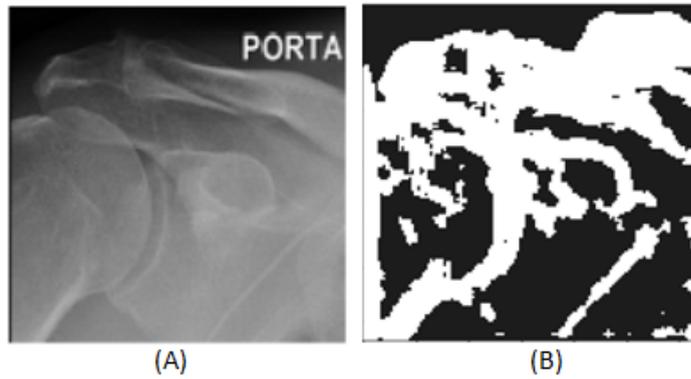


FIGURE 3.7: Balancing mask in MARS SISR: (A) is ground truth, (B) is the balancing mask used in loss function. The mask is a binary image, black is 0.01 and white is 1.

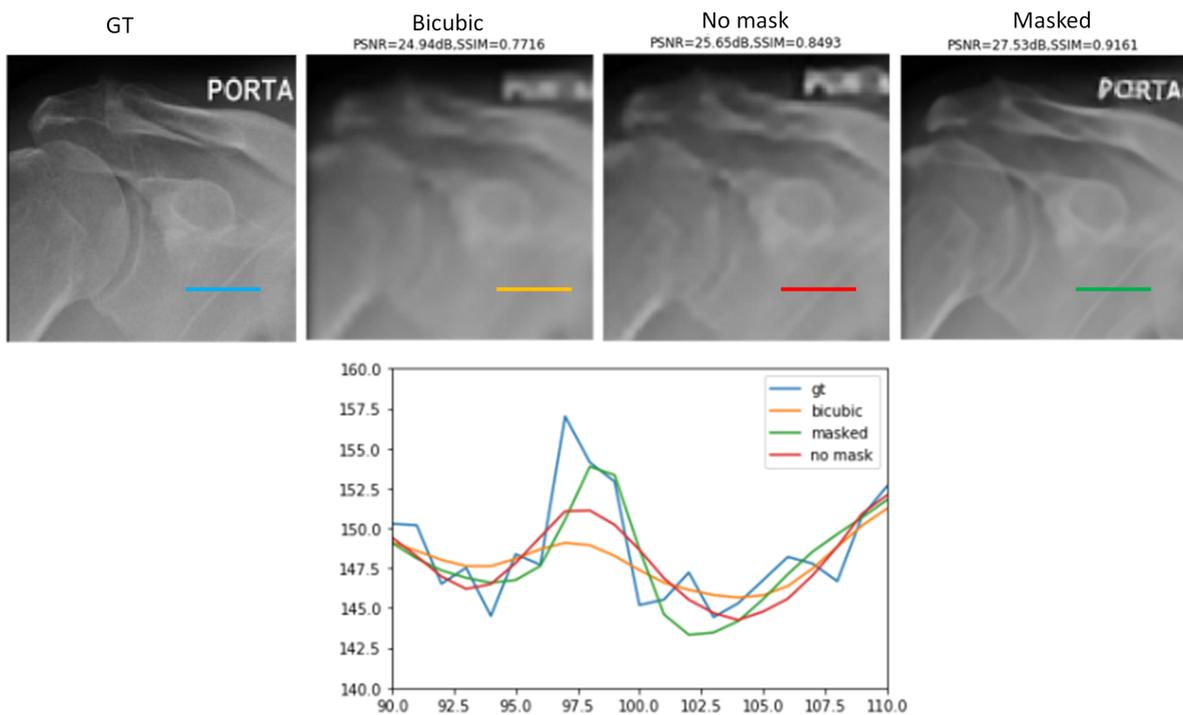


FIGURE 3.8: Without a balancing mask vs with a balancing mask. The first column is ground truth, the second column is bicubic interpolation, the third column is from MMARS without the balancing mask, the fourth is from MMARS with the balancing mask. The result with the balancing mask is much closer to the ground truth.

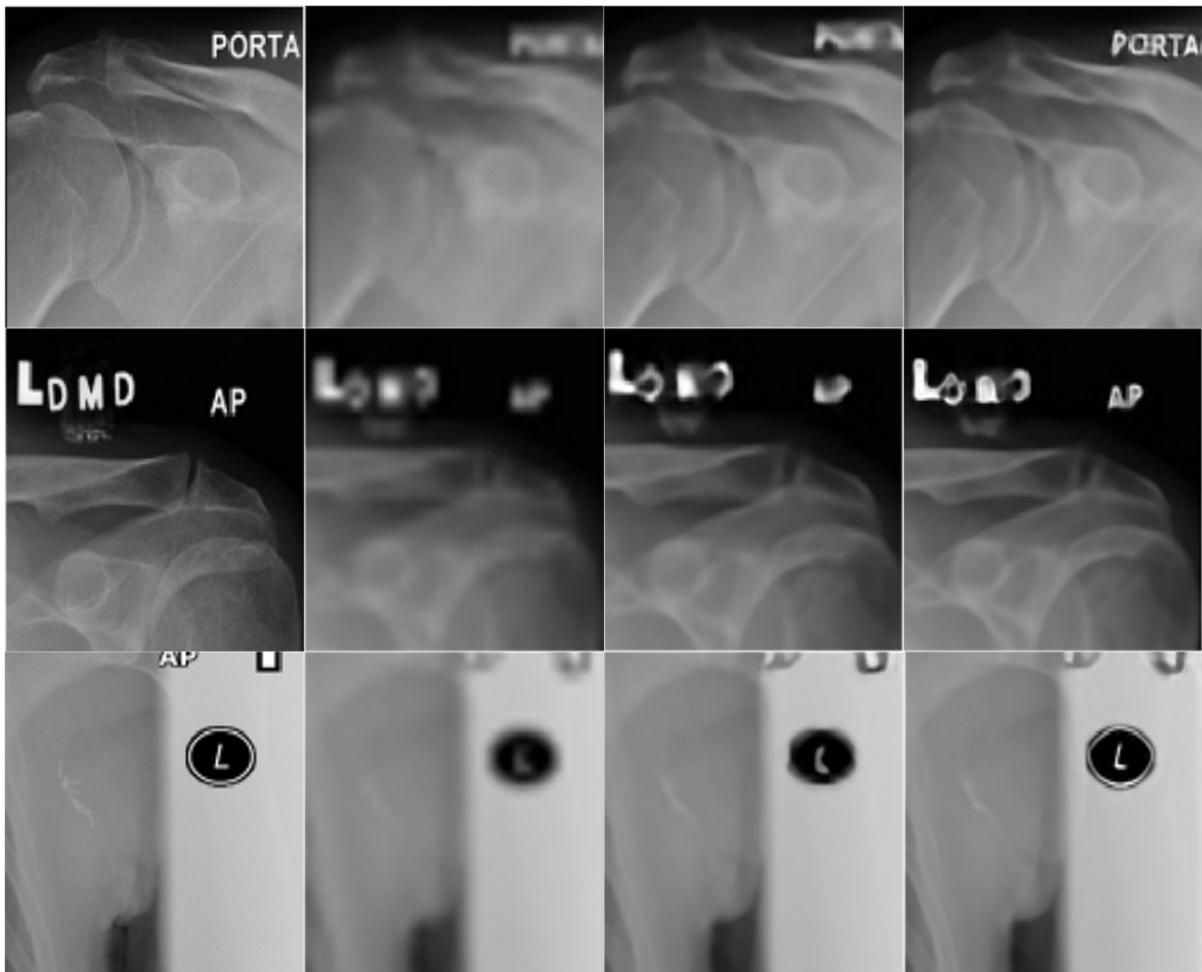


FIGURE 3.9: Chest X-ray SISR results. The first column is ground truth, the second column is a bicubic interpolation, the third column is from MMARS without the likelihood block, the fourth is from MMARS with the likelihood block. This figure shows that the likelihood block improves the results of the proposed MARS SISR, which have more shape edges and fine patterns.

3.4.4.1 Limitations

The Figure 3.11 shows the 2x/4x/8x up-sample results. The proposed MARS SISR can handle a large up-sample scale 8x for some simple cases, e.g. bone edges. But it cannot reconstruct words in the image with higher frequencies. For 4x up-sampling, MARS can generate the words in the image.

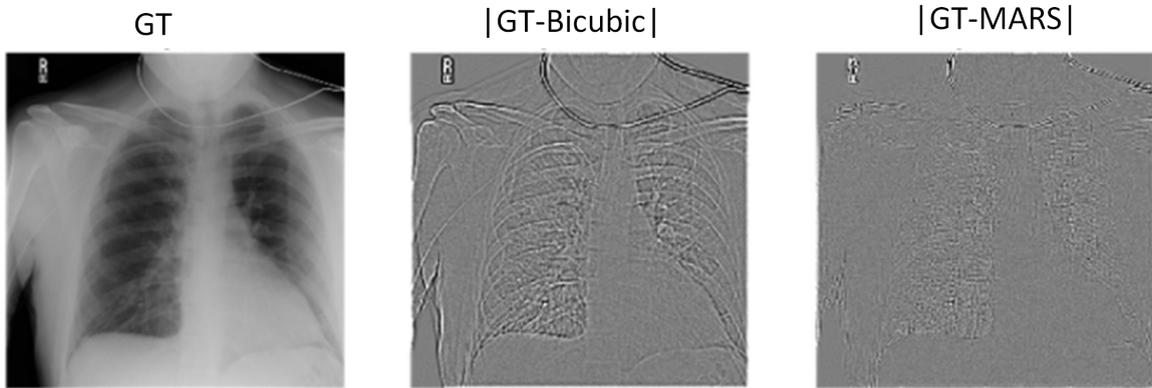


FIGURE 3.10: Chest X-ray SISR results: absolute errors of reconstructions. The first column is ground truth x_{gt} , the second column is the absolute error between ground truth and the bicubic interpolation $|x_{gt} - bicubic(y, 4)|$, the third column is the absolute error between ground truth and the MMARS result $|x_{gt} - x^T|$. The absolute error of the MMARS result is smaller than that of the bicubic interpolation.

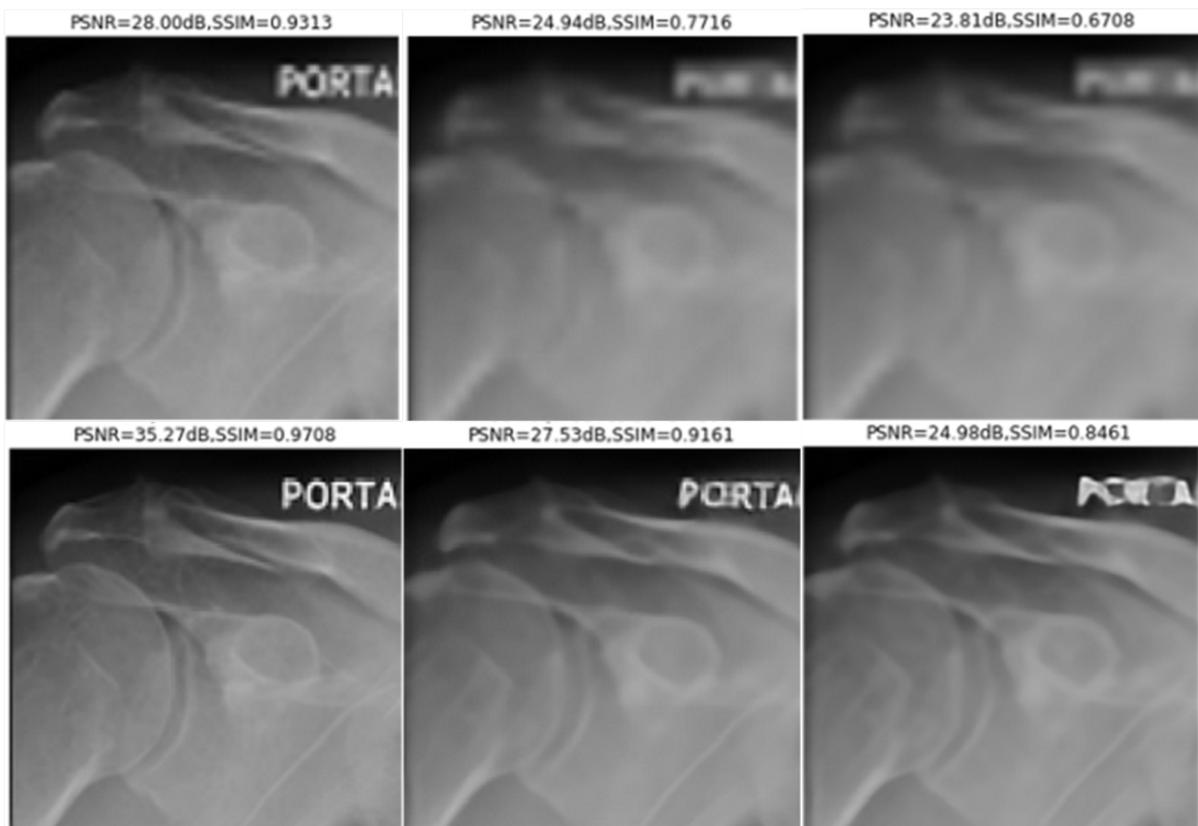


FIGURE 3.11: Chest X-ray 2x/4x/8x SISR results. The first row is the bicubic interpolation, the second row is MMARS results. The first/second/third column is 2x/4x/8x SISR.

3.4.5 conclusion

This chapter describes the proposed MARS method for natural and medical image SISR. Three experiments show MARS SISR can compete with state-of-the-art SISR methods. It also can handle medical image SISR after addressing the classes-imbalance problem and the data augmentation problem.

Chapter 4

CT Reconstruction

4.1 Introduction

Computed tomography (CT) generates patient cross-section images and is one of the most important medical imaging modalities in use today. In CT imaging, X-ray energy goes through the body of patient in a number of directions and the resulting projections are collected and used in an inverse process to reconstruct the cross-section images [26] [9]. A simple 2D parallel CT scan data is called sinogram, which is an 2D Radon transform of objects, see Figure 4.1. The CT reconstruction is an inverse process to estimate a cross-section image of objects from a sinogram, see Figure 4.2. There are three categories of CT reconstruction methods: the conventional one-step inversion methods (e.g. filtered back projection (FBP)), iterative methods and machine-learning-based methods. A comparison of three categories of CT reconstruction methods is showed in Figure 1.2 and reviewed in this introduction.

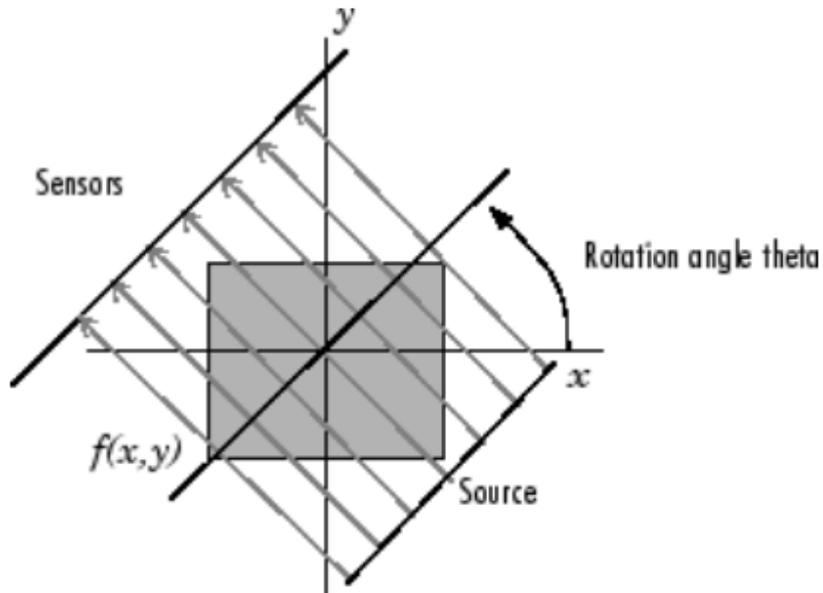


FIGURE 4.1: A simple parallel CT scan that is an 2D Radon transform of objects. The picture is from Matlab's web site.

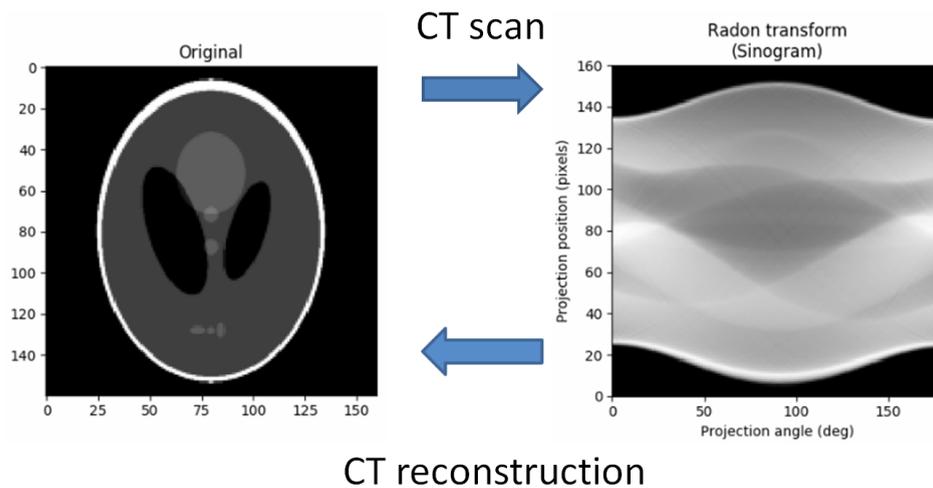


FIGURE 4.2: The right hand side is a sinogram of CT scan data and the left hand side is an cross-section image. The CT reconstruction estimates the cross-section image from the sinogram.

4.2 Related Works

4.2.1 Filtered Back-Projection

Mathematically, the projections and cross-section images in 2D CT are related through a Radon transform and this transform can be approximately inverted by the filtered back-projection (FBP) algorithm to produce the cross-section images [26][9]. The Radon transform is showed in Figure 4.2. The FBP is computationally efficient because it uses the fast fourier transform (FFT), and thus has become a dominant reconstruction technology in commercial CT products.

The FBP algorithm, however, also has some problems. Specifically, the X-ray projection process is inherently stochastic, with the received X-ray energy fluctuating at the detectors (receivers), but FBP treats it essentially as deterministic. As a result, “outliers” or more extreme projection values, even small in number, can still produce significant artifacts. Furthermore, the FBP algorithm is spatially global in that it does not allow spatially local feature specific treatment, even though such treatment could be quite useful in artifact reduction. Finally, 3D CT techniques such as cone beam CT usually have more complex geometries and the FBP cannot adapt easily, leading to some geometrical artifacts [53]. To solve these problems, a stochastic approach known as statistical iterative reconstruction has attracted more attention (e.g., see [53]).

4.2.2 Iterative Methods

The statistical iterative reconstruction is also known as stochastic iterative reconstruction or model-based iterative reconstruction). In an iterative reconstruction approach, the cross-section images to be reconstructed and the imaging (projection) process are modeled as stochastic processes and reconstruction is achieved by iteratively maximizing the posterior probability distribution of the cross-section images. The stochastic models allow one to account and compensate for outliers and incorporate spatially local models and these have led to artifact reduction and image quality improvement; this approach is especially useful when the dose of the X-ray is

reduced (a trend in CT), where the imaging process becomes more stochastic. Iterative reconstruction has achieved some commercial successes as well, and it has been incorporated into some companies' CT products. Despite these advantages and successes, iterative reconstruction also has a weakness, i.e., it is computationally expensive and, even with powerful hardware, it still could take a long time to generate a reconstructed 3D image. Consequently, there is great interest in faster iterative reconstruction. For iterative reconstruction, the problem of being computationally expensive stems from the fact that it is optimizing a function with a large number of intertwined variables (pixels) by iteration and the convergence is slow. Previously proposed acceleration techniques, such as coordinate descent, provide improvements but a significant part of the problem still remains.

4.2.2.1 Statistical Model

Follow the notations in Chapter 2, let $x = \{x_j\}$ be the image to be reconstructed, where j indicates a pixel location. Similarly, let $y = \{y_i\}$ be the received projection X-ray energy that went through a patient's body, where i indicates a particular projection angle and a particular detector location in that angle. The CT image reconstruction problem is then: find x from y . In iterative reconstruction, both x and y are modeled as stochastic, with probability distributions $P(x)$ and $P(y|x)$ and the problem is to find the best x , denoted as \hat{x} , by maximizing the posterior distribution $P(x|y)$. For the sake of convenience, equation (2.1) is written here again,

$$\begin{aligned}
\hat{x} &= \arg \min_x \text{Loss}_{MAP}(x, y) \\
&= \arg \min_x (-\log P(y|x) - \log P(x)) \\
&= \arg \min_x (F_1(x, y) + F_2(x)) \\
&= \arg \min_x F(x, y)
\end{aligned} \tag{4.1}$$

where $\text{Loss}_{MAP}(x, y) = -\log P(y|x) - \log P(x)$ is loss function of the MAP estimation. The first and second terms in equation (4.1) are likelihood and prior functions, denoted by $F_1(x, y)$ and $F_2(x)$.

Typically, $P(y|x)$ is a conditionally independent Poisson model with mean Ax , where A is a linear projection that captures the imaging geometry. In practice, the Poisson model is often approximated by a Gaussian with spatially varying variances [53], with

$$\begin{aligned}
F_1(x) &= -\log P(y|x) \\
&= -\log e^{-\frac{1}{2}(y-Ax)^T D(y-Ax)} \\
&= \frac{1}{2}(y-Ax)^T D(y-Ax) + cont_1
\end{aligned} \tag{4.2}$$

where D is the covariance matrix of noise, $cont_1$ is a constant. In CT reconstruction, D is a diagonal matrix, with the i th element d_i proportional to received photon count of the i th detector, and d_i is the inverse of the variance of the received photon count at the i th detector.

$$\lambda_i = I_0 e^{-y_i} \tag{4.3}$$

where I_0 is received photon count of the i th detector when scanning air (not dependent on i)

4.2.2.2 Prior

$P(x)$ is a prior model that incorporates various constraints about image x , such as “it is smooth except at region boundaries.” $P(x)$ often is an energy function with a sparse probability distribution and comes in the form of a Markov random field (MRF) [53]. The probability distribution of an MRF usually has the form

$$\begin{aligned}
F_2(x) &= -\log P(x) \\
&= \beta \sum_j V_j(x) + cont_2
\end{aligned} \tag{4.4}$$

where $\beta > 0$ and $cont_2$ are constants and $V_j(x)$ is local function of x_j and its neighbors (also known as a clique function). For example, a simple “edge-preserving smoothness” local function with

$$V_j(x) = \frac{1}{2} \sum_{k \in N_j} |x_j - x_k| \quad (4.5)$$

where N_j set of four immediate neighbors of location j ; other types of $V_j(x)$ can also be used.

4.2.2.3 Optimization

equation (4.1) is designed as a convex optimization problem. Iterative reconstruction generally solves the reconstruction problem by (iterative) gradient descent

$$x^{t+1} = x^t - \alpha \nabla_x F(x^t, y) \quad (4.6)$$

where x^t is the reconstructed image at t th iteration. In actual implementation, coordinate descent (CD) is often used to make the algorithm faster. In this technique, iteration is done pixel by pixel, rather than altogether to improve the speed of convergence.

4.3 Contributions and Proposed Methods

4.3.1 Contributions

The contributions of this chapter are listed below:

1. Used the proposed model augmented recursive neural networks (MARS) in CT image reconstruction and metal artifact reduction (MAR)

2. Built a guideline of adjusting the distribution of trainable parameters in the proposed MARS neural networks: when the observed data contains strong noise, the likelihood block should have more trainable parameters. An experiment in this chapter demonstrated that when the initial solution, e.g. FBP, contained strong artifacts, the model-based likelihood block provided more improvements than the model-free prior block.
3. Used deep learning optimization algorithms, e.g. Momentum and Adam, for CT image iterative reconstructions.

4.3.2 MARS for CT Reconstructions

The proposed model augmented recursive neural networks (MARS) in Algorithm 2 is used for CT reconstruction. The structure of MARS for CT reconstruction is similar to MARS for SISR.

4.3.2.1 Bypass Branch

One bypass branch passes a current estimate of a CT image x^t to a next iteration/stage estimate of a CT image x^{t+1} . The target of MARS is to minimize the difference between the ground truth image x_{gt} and the final estimate of a CT image x^T . This new target is called residual in boosting algorithms that convert a group of weak learners to a strong learner in order to reduce bias and variance of estimation. The bypassed signal x^t keeps low-frequency information and the residual R^t has high-frequency information which is the learning target of our neural network. The high-frequency information provides more detail information to help doctors or machines to know inside of patient's bodies.

4.3.2.2 Likelihood Feature Block

Fixed projection and back-projection. For CT reconstruction, a fixed transpose matrix A^T is used as the back-projection B . The reason was mentioned in Chapter 2: for CT reconstruction, the projection A and the back-projection B are both global operators. In neural network structures, a global operator is a fully-connected layer with sparse non-zero parameters. The

space complexity of a fully-connected layer is $\mathcal{O}(nm)$, where n is the size of a CT image and m is the size of a CT sinogram. For example, A CT system has 1024×1024 images x and 1000×1000 sinograms y (1000 views and 1000 detectors). Then the B is $1000^2 \times 1024^2$ sparse matrix. It needs 4TB memory to store B in a full matrix format. Tensorflow does not support sparse trainable variables. So it is impossible to train B in Tensorflow. Meanwhile, the fixed matrix A^T only needs 2Gb to be stored in a sparse matrix format.

4.3.2.3 Prior Feature Block

The traditional prior models are usually smooth functions of x , such as Tikhonov [31], total variation (TV) [65] and Generalized Gaussian Markov random Field (GGMRF) [53]. These functions rely on some general sparse models of the image gradient. For example, the probability distribution of image gradients is Gaussian or Laplacian distribution, then the corresponding regularization/ prior model is L2-norm or L1-norm of image gradients. However these sparse prior models emphasize the generalization for all of the natural images, then they provide weak constraints to reconstruction. Data-driven models, e.g. deep learning, can learn more complex mapping functions in a large amount of training data [55]. In proposed MARS, a CNN block is used to learn a prior model based on a medical image data set.

4.3.3 Experimental Results of MARS for CT Reconstructions

A CT reconstruction simulation is designed to show the performance of MARS neural network. The original data is the DeepLesion CT images [61] from NIH Clinical Center. Randomly choose 1000 images and split them into 800 training images and 200 testing images. Finally, crop the center 64×64 patch. A 50 views projection matrix A is generated by Siddon algorithm [48]. The sinogram is a 95×50 matrix, which has 50 views and 95 detectors.

Controlled Noise Level. In this CT reconstruction experiment, the Poisson noise level can be controlled by a hyper-parameter. In addition, one or two 2×2 metal blocks are added into ground truth 64×64 patches at random location. These metal blocks generate strong metal

TABLE 4.1: Average PSNR/ SSIM of CT reconstruction results

Noise	Method	PSNR	SSIM
weak	FBP	24.08	0.516
weak	MARS w/LP	32.75	0.847
weak	MARS w/P	32.66	0.843
weak	MARS w/L	32.78	0.845
strong	FBP	2.54	0.543
strong	MARS w/LP	25.78	0.979
strong	MARS w/P	24.08	0.975
strong	MARS w/L	25.76	0.979

TABLE 4.2: Training/ testing time of MARS for CT reconstruction. For a 512x512 CT reconstruction problem, the testing time of one image is $0.00248 \times 64 = 0.15872$ sec/image. The iterative method with Adam has 10000 iterations. MARS only has 3 stages/iterations. MARS is $152.602/0.00248=61533$ times faster than an iterative method with Adam.

Method	HR Patch size	Train (sec/patch)	Test (sec/patch)
Iterative method with GD	64x64	NA	302.445
Iterative method with Momentum	64x64	NA	450.635
Iterative method with Adam	64x64	NA	152.602
MARS	64x64	0.00537	0.00248

artifacts. The purpose of adding noise and metal artifacts is to make the reconstruction task more difficult and give more space to show the improvement from MARS. Otherwise, for an easy reconstruction task, FBP can provide good results. Based on the experiment results shown in Figure 4.6 and Table 4.1, when the noise is weak, the FBP results are good, and the likelihood block cannot improve the results. When noise is strong, the FBP results are unacceptable, and the likelihood block improves the results.

4.3.4 Deep Learning Optimization Algorithms for Iterative Methods

Iterative methods CT reconstruction by gradient descent or coordinate descent algorithms still have two problems, which limit its clinic applications: first, because of the slow convergence rate of these algorithms, it takes a long time to get a reconstruction image, (could take hours;) second, iterative methods do not use external high-quality CT data to boost reconstruction

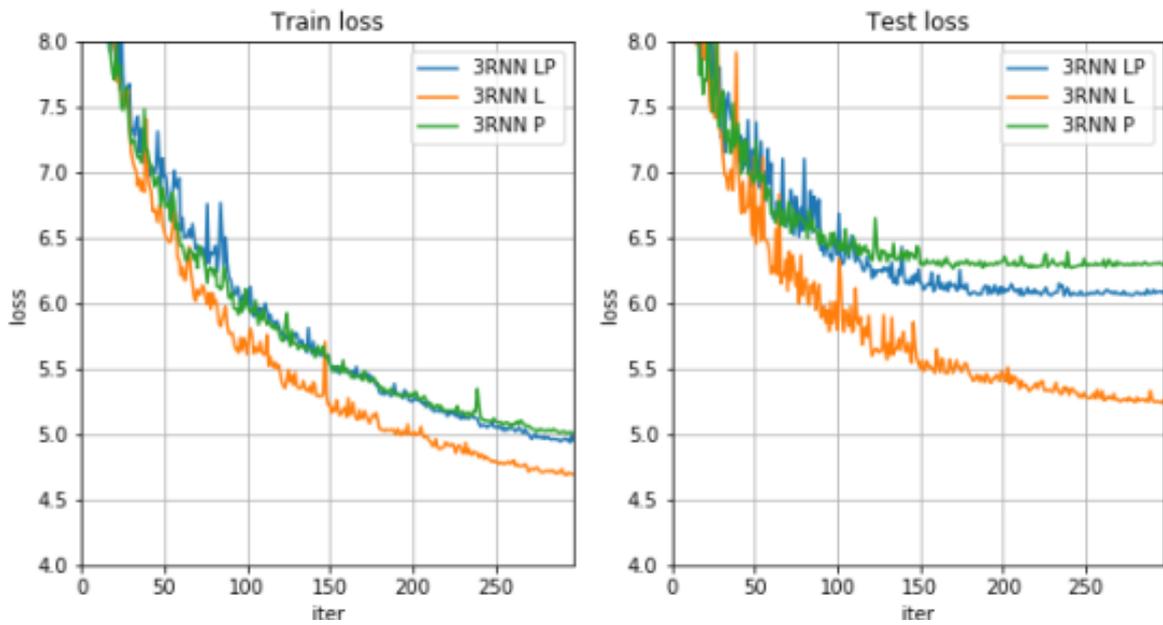


FIGURE 4.3: CT reconstruction training and testing loss curves with 3 configurations. Left is training loss curves; right is testing loss curves. 3RNN means a 3-stage MARS with shared parameters. LP means with both the likelihood and prior blocks, L means with the likelihood block and without the prior block, P means without the likelihood block and with the prior block. The loss function is Charbonnier loss function.

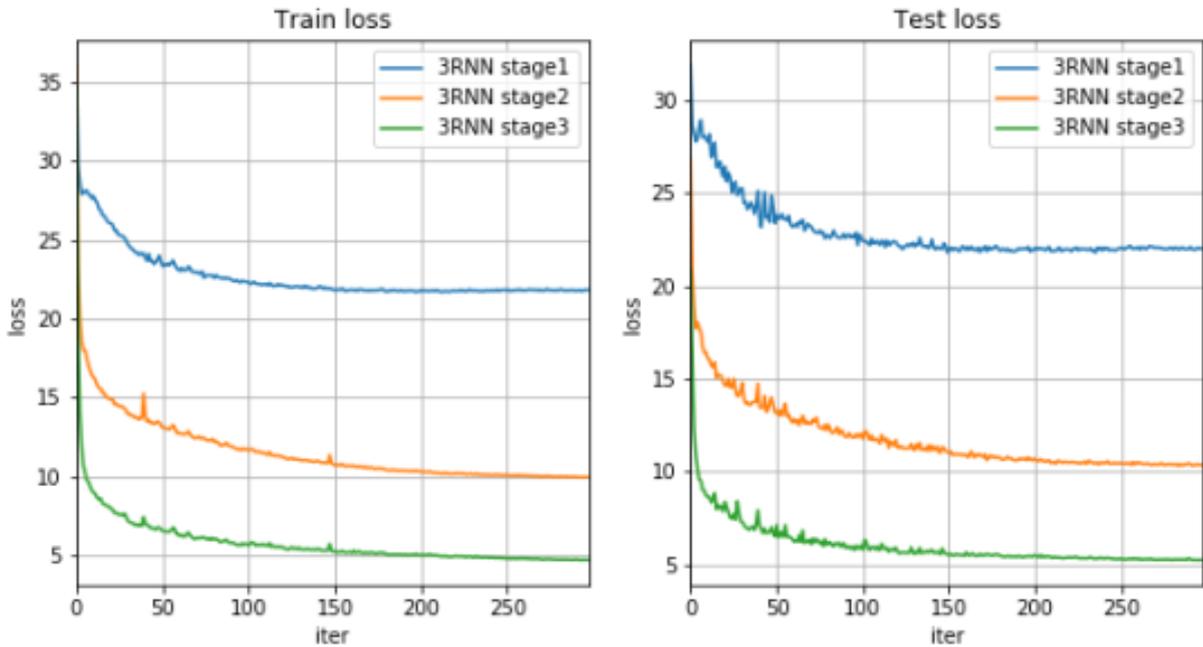


FIGURE 4.4: CT reconstruction training and testing loss curves of each stages. Left is training loss curves; right is testing loss curves. These 3 curves are for stage 1, 2, 3. The loss function is Charbonnier loss function.

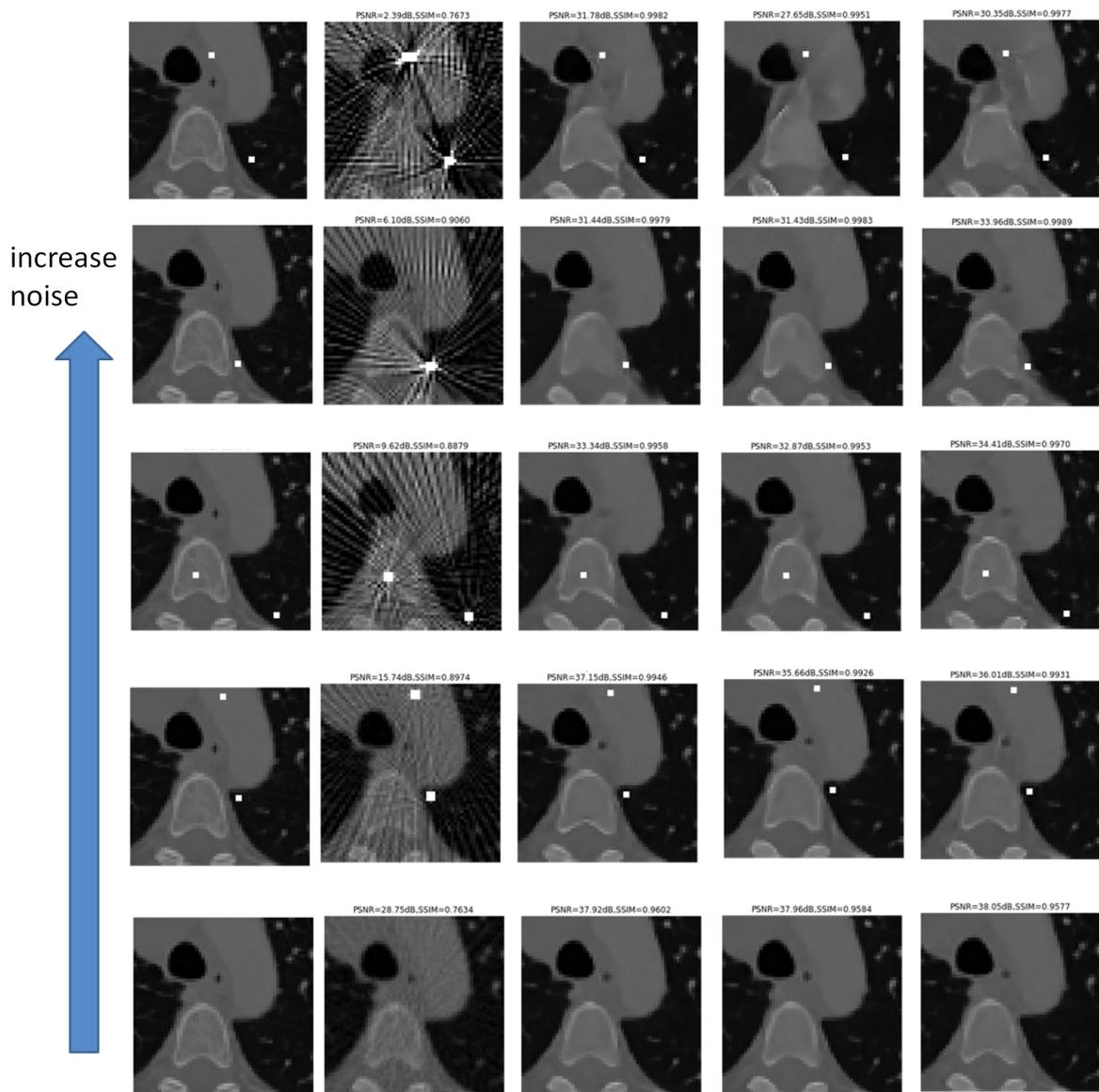


FIGURE 4.5: CT reconstruction result while increasing Poisson noise. The first column is ground truth, the second column is FBP, the third column is from MARS with the likelihood block and prior block, the fourth is from MARS without the likelihood block and with the prior block, the fifth column is from MARS with the likelihood block and without prior block.

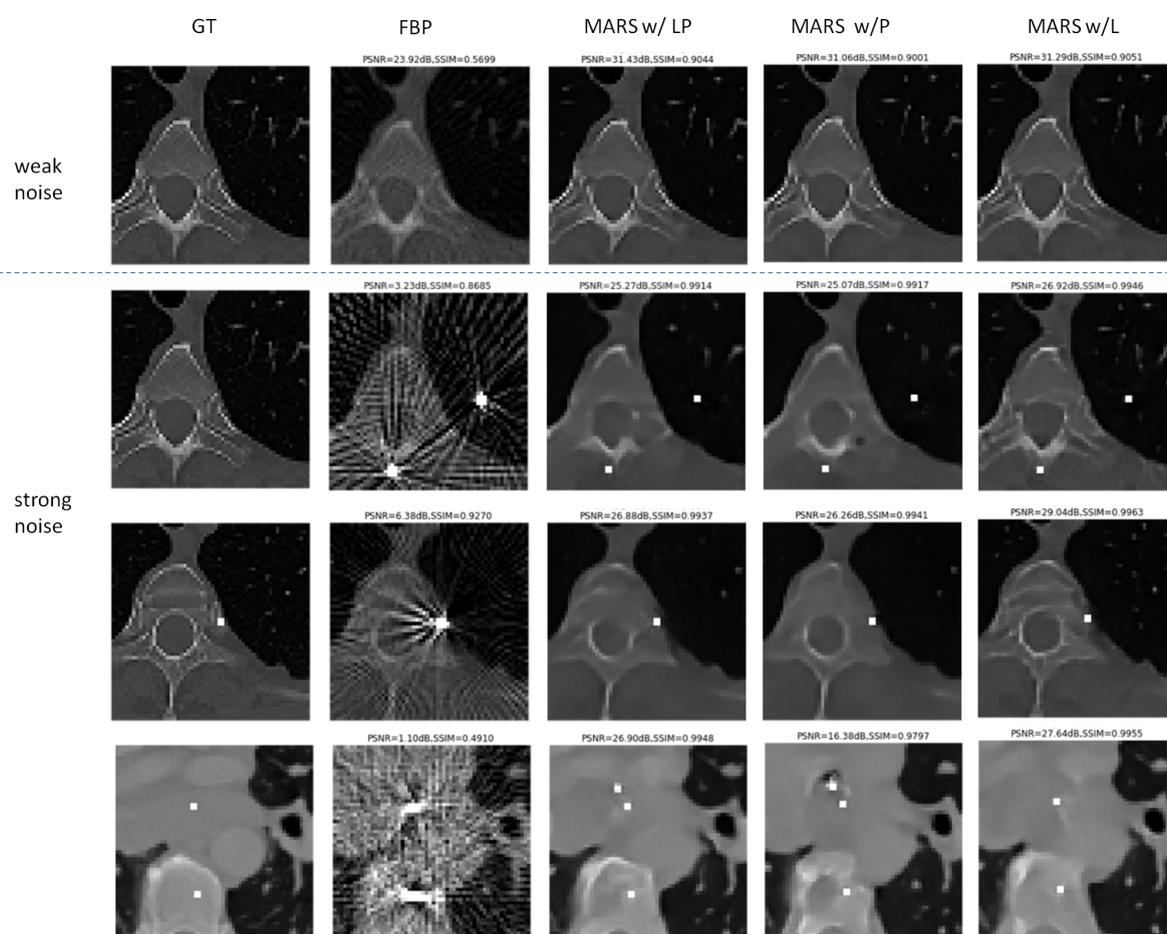


FIGURE 4.6: CT reconstruction results at two noise levels. The first to fifth columns are images: ground truth; FBP; MARS with the likelihood and prior blocks; MARS only with the prior block; MARS only with the likelihood block. The first row is images with a weak Poisson noise. It is an easy case to remove artifacts. The second to fourth rows are images with strong Poisson noise and metal artifacts. This picture shows that when the Poisson noise is weak, all these three models can have good results. However, when the Poisson noise is strong, the likelihood block provides more gains to reduce artifacts.

performance. This dissertation has addressed these two problems: 1) using deep learning optimization algorithms to speed up the convergence of iterative methods; 2) training a deep MARS neural network to improve reconstruction, even with a strong Poisson noise.

The past few years have seen a burst of advances in machine learning technologies, especially in “deep learning” [40]. “Deep” neural networks of various kinds have achieved near-human performances in speech and image recognition experiments. A “deep” neural network is essentially a highly nested function with many layers, that provide the network the capacity to

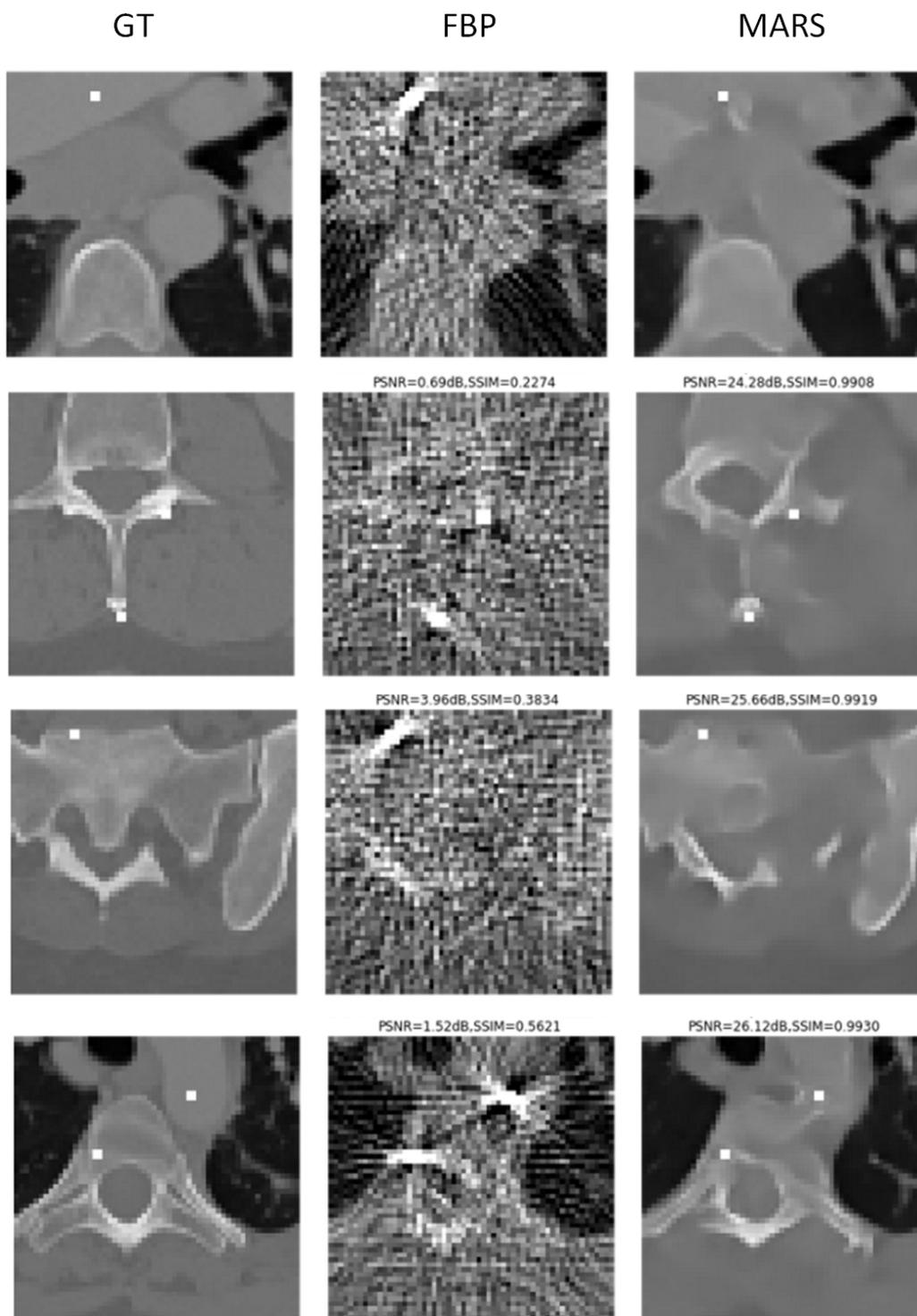


FIGURE 4.7: CT reconstruction results with strong noise and metal artifact. The first to third columns are images: ground truth; FBP; MARS.

represent/implement extremely complex functions and input-output relationships. Each layer of a deep neural network comes with a number of parameters that have to be learned by using training data, which is usually a large number of example input-output pairs. The learning process is an iterative optimization process, where the neural network is directed to adjust its parameters to match the output for the input in the training data. Since the number of parameters in a deep neural network can be extremely large, the iterative optimization process is very challenging. Indeed, although deep neural networks were already studied in the 1980s, their effective training and applications have only become possible and successful recently as a result of significant advances made in the iterative optimization algorithms over the past few years [40][20].

In the study of neural networks, a basic gradient descent algorithm similar to equation (4.6) has also been used for a long time to learn or estimate the network parameters from training data; the algorithm is known as backpropagation [41]. For deep networks, this algorithm does not lead to good convergence and for many years, deep neural networks had very limited success. This situation has changed dramatically over the past few years with the advent of a new set of better iterative optimization algorithms. These algorithms incorporate and combine a number of new ideas/features into the original gradient descent algorithm, such as momentum (acceleration), adaptive learning rate (faster in regions where the gradient is small and slower in regions where the gradient is large), and better initialization. This has led to much better training algorithms that finally made deep neural networks successful. Among these are “Momentum” [51] and “Adam” [36]; the former is simple but very effective and has some relation to the conjugate gradient method while the latter incorporates momentum and adaptive learning rates as well as other ideas and is widely recognized as one of the best iterative optimization algorithms today for deep neural network training.

Both Momentum and Adam are used as replacements to the basic gradient descent algorithm for iterative reconstruction. Specifically, the Momentum algorithm is given by

$$\begin{aligned} m^t &= \gamma m^{t-1} + \nabla_x F(x^t, y) \\ x^{t+1} &= x^t - \alpha m^t \end{aligned} \tag{4.7}$$

where m^t is a momentum vector at the t th iteration, $\gamma > 0$ is a momentum hyper-parameter, and $F()$ is the loss function. The Adam algorithm expands the moment algorithm with

$$x^{t+1} = x^t - \alpha^t m^t \oslash (\sqrt{s^t} + \eta) \quad (4.8)$$

Here, m^t is a momentum vector, s^t is a vector that makes the learning rate adaptive $\alpha > 0$ is a step size, $\eta > 0$ is a small constant vector, \oslash is element-wise division; both m^t and s^t are functions of the gradient $\nabla_x F(x^t, y)$ and their details can be found in [36]. Compared to the basic gradient descent, Momentum and Adam are more powerful algorithms that provide better and faster convergence.

4.3.5 Experimental Results of Deep Learning Optimization Algorithms for Iterative Methods

One simulation experiment is designed to apply Momentum and Adam to interactive reconstruction. A typical 64x64 phantom image used in CT image reconstruction simulations was reconstructed from 50 views of Poisson distributed projections using the FBP, basic gradient descent, coordinate descent, Momentum, and Adam algorithms. The projection matrix A is generated by Siddon algorithm [48]. A is a 4750x4096 sparse matrix. The reconstructions by Momentum and Adam are implemented using Tensorflow. In Figure 4.8, a typical phantom image used in CT image reconstruction simulations was reconstructed from 50 views of Poisson distributed projections using the FBP, basic gradient descent, coordinate descent, Momentum, and Adam algorithms, with the last two implemented using Tensorflow; also shown in Fig. 1 are the MSE (mean square error) of each reconstruction. Here, we kept the number of projections to be relatively small because in many applications, such as CT imaging for air travel security, having a smaller number of projections without sacrificing reconstruction quality too much will help to reduce cost; furthermore, this situation is also more challenging to a reconstruction algorithm. As shown in Figure 4.8, the FBP produced a very noisy reconstruction while the basic gradient descent did not provide much improvement due to its relatively slow

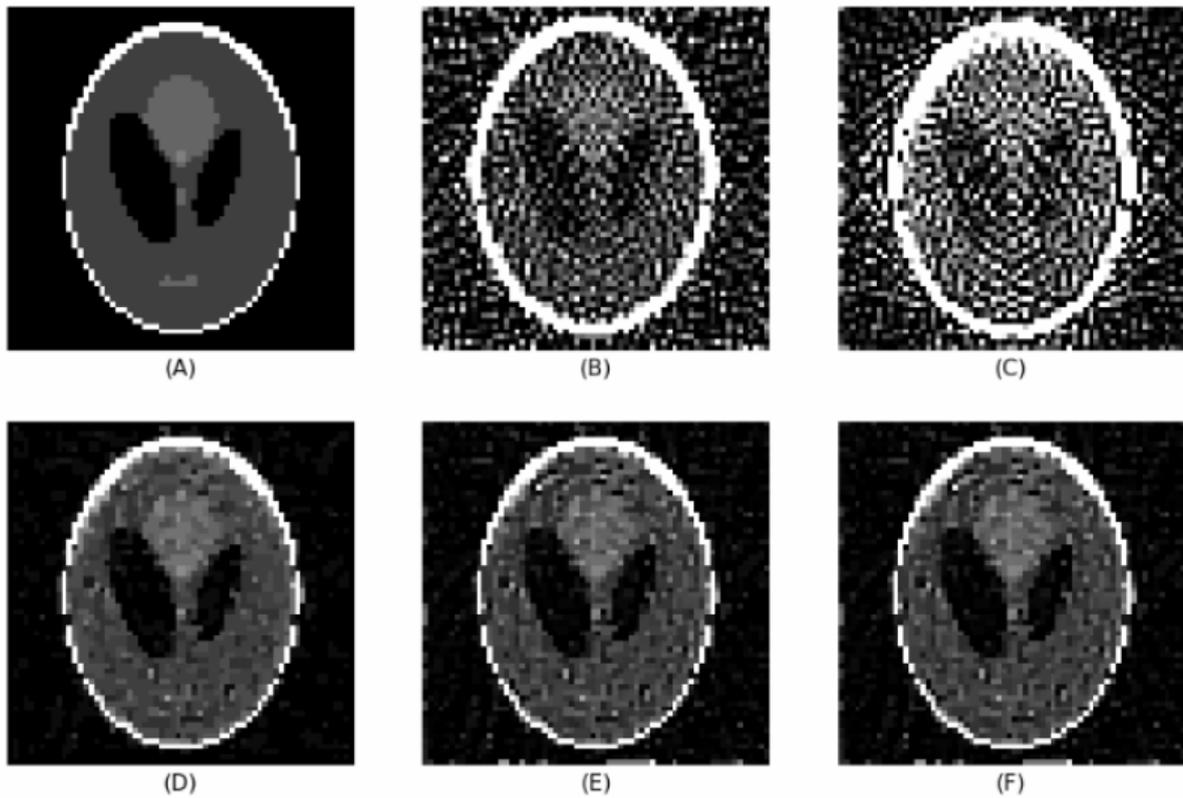


FIGURE 4.8: CT reconstruction results with iterative methods. (A) ground truth, (B) FBP, mse=1254.14, (C) basic gradient descent, mse=505.16, (D) coordinate descent, mse=6.74, (E) Momentum, mse=6.36, (F) Adam, mse=6.09

convergence. Both Momentum and Adam algorithms, on the other hand, produced significantly better results, as did coordinated descent. As a complement to Figure 4.8, the convergence behavior of the algorithms in Figure 4.8, in terms of both the loss function and mean square error (MSE) with respect to the ground truth, are shown in Figure 4.9, where Momentum and Adam algorithm converged much faster than the basic gradient descent. Momentum and Adam also converged faster than coordinate descent, especially in MSE. Furthermore, unlike coordinate descent, which updates pixels in the reconstructed image sequentially, Momentum and the Adam update all the pixels simultaneously hence can be implemented as parallel computations, which can potentially be a lot faster.

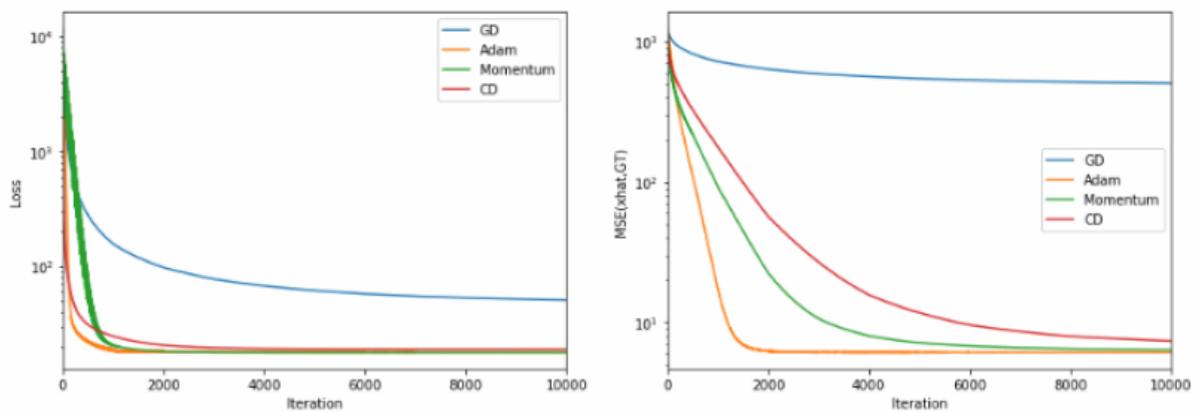


FIGURE 4.9: Convergence results with iterative methods. Left: curves of loss function, right: curves of MSE between reconstructions and the ground truth.

4.4 Conclusions

This chapter reviews FBP, interactive methods and deep learning methods in CT reconstruction. One experiment shows that the deep learning optimization algorithms Momentum and Adam can accurately converge and improve results. The second experiment shows the proposed MARS network can improve CT reconstruction results. Especially when the sinogram is noised and the reconstruction has metal artifacts, MARS can reduce metal artifact and other artifacts.

Chapter 5

Summary, Limitations and Future Works

The proposed model augmented recursive neural networks is a general deep neural network structure to solve ill-posed inverse problems, e.g. single image super-resolution and CT reconstruction. This network structure combines the projection model A and back-projection model B into the structure of the neural network, which is called the likelihood block in MARS. The likelihood block extracts reconstruction error and back-projection them to image space X to compensate current estimation x^t . MARS shows competitive performance with state-of-the-art SISR method for natural images SISR and satisfactory results for medical image SISR. For CT reconstruction when sinograms contain a strong Poisson noise, MARS can dramatically improve reconstruction results and reduce metal artifacts, without any extra metal artifact reduction step.

The limitations of MARS are: 1) a MARS neural network needs a large amount of training data and powerful computers with GPUs. If an inverse problem is well-posed, the conventional methods can handle this problem. 2) The likelihood block in MARS can improve the result if the observed data contains strong noise. Otherwise, it is hard to see improvement from the likelihood block.

My future works are studying how to build Momentum, Adam, and other optimization algorithms into neural network structure, in order to accelerate training convergences and improve results.

Appendix

Two examples of One-step Inversion: WLS and FBP

WLS (weighted least squares) Singular value decomposition (SVD) of A is

$$A = USV^T$$

WLS solution is

$$\begin{aligned}\hat{x}_{WLS} &= \arg \min_x (Ax - y)^T D (Ax - y) \\ &= (A^T D A)^{-1} A^T D y\end{aligned}$$

where D is a weights matrix, which is usually the inverse matrix of the covariance matrix of y . $y = y_j$ is a random vector. Assume y_j are independent, then D is a diagonal matrix.

$$\begin{aligned}\hat{x}_{WLS} &= (A^T D A)^{-1} A^T D y \\ &= (V S U^T D U S V^T)^{-1} V S U^T D y \\ &= V (S D S)^{-1} S D U^T y \\ &= V S^{-1} U^T y\end{aligned}$$

FBP (filtered back-projection) The i th view of the sinogram is $y_i = A_{i*}x$. According to the central slice theorem and the geometry of CT [26], y_i is a low-pass version of the real projection of a cross-section image. High-Pass filtering is needed to compensate for the high-frequency components. Filtered sinogram $Q\mathcal{F}A_{i*}x$, where Q is a diagonal matrix, the diagonal elements q_i s are DFT coefficients of the ramp filter. $\mathcal{F}^*Q\mathcal{F}$ is a high-pass operator.

$$\hat{x}_{FBP} = \sum_i A_{i*}^T \mathcal{F}^* Q \mathcal{F} y_i$$

where \mathcal{F}^* is inverse DFT matrix.

Compare WLS and FBP One-step inversions have a similar procedure: first, from the observation space Y , projects y to a latent space; second, compensates weak components with small singular values or high frequencies; third, back-project the compensated coefficients to the target space X .

References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.
- [2] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.
- [3] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.
- [4] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [5] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.
- [6] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4. Austin, TX, 2010.

- [7] John D Boice Jr, Dale Preston, Faith G Davis, and Richard R Monson. Frequent chest x-ray fluoroscopy and breast cancer incidence among tuberculosis patients in massachusetts. *Radiation research*, 125(2):214–222, 1991.
- [8] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [9] Thorsten M Buzug. Computed tomography. In *Springer Handbook of Medical Technology*, pages 311–342. Springer, 2011.
- [10] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.
- [11] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10. ACM, 2016.
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.
- [14] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.
- [15] David L Donoho et al. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.
- [16] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

- [17] E Faramarzi, D Rajan, and M. P Christensen. Unified blind method for multi-image super-resolution and single/multi-image blur deconvolution. *IEEE Transactions on Image Processing*, 22(6):2101–2114, 2013.
- [18] S Farsiu, M.D Robinson, M Elad, and P Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, 2004.
- [19] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.
- [20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [21] Antonio Gulli and Sujit Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.
- [22] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.
- [23] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1397–1409, 2013.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [25] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- [26] Jiang Hsieh et al. Computed tomography: principles, design, artifacts, and recent advances. SPIE Bellingham, WA, 2009.
- [27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

- [28] J. B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015.
- [29] Daniel Glasner Shai Bagon Michal Irani. Super-resolution from a single image. In *Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan*, pages 349–356, 2009.
- [30] Jithin Saji Isaac and Ramesh Kulkarni. Super resolution techniques for medical image processing. In *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, pages 1–6. IEEE, 2015.
- [31] Kazufumi Ito and Bangti Jin. *Inverse problems: Tikhonov theory and algorithms*. World Scientific, 2015.
- [32] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.
- [33] Nikhil Ketkar. Introduction to pytorch. In *Deep learning with python*, pages 195–208. Springer, 2017.
- [34] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.
- [35] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.
- [36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q.

- Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [38] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.
- [39] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [41] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [42] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.
- [43] Brian MacMahon. Prenatal x-ray exposure and childhood cancer. *Journal of the National Cancer Institute*, 28(5):1173–1191, 1962.
- [44] Antonio Marquina and Stanley J Osher. Image super-resolution by tv-regularization and bregman iteration. *Journal of Scientific Computing*, 37(3):367–382, 2008.
- [45] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.
- [46] Fatemeh Pak, Hamidreza Rashidy Kanan, and Afsaneh Alikhassi. Breast cancer detection and classification in digital mammography based on non-subsampled contourlet transform (nsct) and super resolution. *Computer methods and programs in biomedicine*, 122(2):89–107, 2015.

- [47] Sébastien Racanière, Théophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in neural information processing systems*, pages 5690–5701, 2017.
- [48] Robert L Siddon. Fast calculation of the exact radiological path for a three-dimensional ct array. *Medical physics*, 12(2):252–255, 1985.
- [49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [50] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *Image Processing, IEEE Transactions on*, 20(6):1529–1542, 2011.
- [51] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
- [52] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.
- [53] Jean-Baptiste Thibault, Ken D Sauer, Charles A Bouman, and Jiang Hsieh. A three-dimensional statistical approach to improved image quality for multislice helical ct. *Medical physics*, 34(11):4526–4544, 2007.
- [54] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016.
- [55] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

- [56] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, pages 550–558, 2016.
- [57] Lara G Villanueva, Gustavo M Callicó, Félix Tobajas, Sebastián López, Valentín De Armas, José F López, and Roberto Sarmiento. Medical diagnosis improvement through image quality enhancement based on super-resolution. In *2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, pages 259–262. IEEE, 2010.
- [58] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3462–3471, 2017.
- [59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [60] Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomenech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017.
- [61] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M Summers. Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *Journal of Medical Imaging*, 5(3):036501, 2018.
- [62] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Thomas Huang. Coupled dictionary training for image super-resolution. *IEEE transactions on image processing*, 21(8):3467–3478, 2012.
- [63] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 2019.

- [64] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.
- [65] Jun Zhang, Jun Wang, Hongquan Zuo, Guangwu Xu, and Jean-Baptiste Thibault. Compressed sensing algorithms for fan-beam computed tomography image reconstruction. *Optical Engineering*, 51(7):071402, 2012.

Curriculum Vitae

Hongquan Zuo

Place of birth: Yibin, Sichuan, China

Education:

- B.A., UESTC, May 2004
Major: Computer Networking
- M.S., UESTC, May 2007
Major: Signal and Information Processing
- Ph.D., UW-Milwaukee, Aug 2019
Major: Electrical Engineering

Dissertation Title: MODEL AUGMENTED DEEP NEURAL NETWORKS FOR MEDICAL IMAGING RECONSTRUCTION PROBLEMS

Publications:

- Jun Zhang, Jun Wang, Hongquan Zuo, Guangwu Xu, Jean-Baptiste Thibault, "Compressed sensing algorithms for fan-beam computed tomography image reconstruction," Opt. Eng. 51(7) 071402 (15 May 2012)
- Hongquan Zuo, Jun Zhang, "Resolution enhancement for x-ray images," Proc. SPIE 10133, Medical Imaging 2017: Image Processing, 101331L (24 February 2017)
- H. Zuo, Y. Lu, D. Xiang, J. Zhang, J. Peplinski, M. Adamak, "Detector output prediction for CT detector array manufacturing," Proc. SPIE 10573, Medical Imaging 2018: Physics of Medical Imaging, 105735Z (9 March 2018)
- Jun Zhang, Hongquan Zuo, "Iterative CT image reconstruction using neural network optimization algorithms," Proc. SPIE 10948, Medical Imaging 2019: Physics of Medical Imaging, 1094863 (1 March 2019)