

**CNN-BASED SINGLE IMAGE SUPER-RESOLUTION NETWORK  
AND BIOMEDICAL IMAGE APPLICATIONS**

by

Samet Bayram

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Winter 2018

© 2018 Samet Bayram  
All Rights Reserved

**CNN-BASED SINGLE IMAGE SUPER-RESOLUTION NETWORK  
AND BIOMEDICAL IMAGE APPLICATIONS**

by

Samet Bayram

Approved: \_\_\_\_\_  
Mark Mirotznik, Ph.D.  
Professor in charge of thesis on behalf of the Advisory Committee

Approved: \_\_\_\_\_  
Mark Mirotznik, Ph.D.  
Acting Chair of the Department of Electrical and Computer Engineering

Approved: \_\_\_\_\_  
Babatunde A. Ogunnaike, Ph.D.  
Dean of the College of Engineering

Approved: \_\_\_\_\_  
Ann L. Ardis, Ph.D.  
Senior Vice Provost for Graduate and Professional Education

## ACKNOWLEDGMENTS

It is a big pleasure to thank everyone who helped me with my research and writing this thesis.

I would like to thank my advisor, Dr. Kenneth E. Barner who is also chair of Electrical and Computer Engineering Department at the University of Delaware. His patience, guidance and encouragement have led me to achieve the success of my study.

To Dr. Mark Mirotznik, the acting chair of Electrical and Computer Engineering Department, for his support during my thesis approval, thank you.

I am grateful to my friends, my colleagues in my research group and all school faculty for their kindness and support.

I would like to give my special thanks to Turkish Ministry of National Education for their financial support and guidance from the first day of my master's education to this thesis study.

Lastly, I would like to dedicate this thesis to my dear mother Hava and my all family. They have always been supporting me and believing me with love.

## TABLE OF CONTENTS

<b>LIST OF TABLES</b> . . . . .	<b>vi</b>
<b>LIST OF FIGURES</b> . . . . .	<b>vii</b>
<b>LIST OF SYMBOLS</b> . . . . .	<b>ix</b>
<b>ABSTRACT</b> . . . . .	<b>x</b>
 <b>Chapter</b>	
<b>1 INTRODUCTION</b> . . . . .	<b>1</b>
1.1 Thesis Structure . . . . .	2
<b>2 BACKGROUND AND RELATED WORK</b> . . . . .	<b>3</b>
2.1 Super Resolution Basics . . . . .	3
2.2 Interpolation Methods . . . . .	4
2.2.1 Nearest-Neighbor Interpolation . . . . .	4
2.2.2 Bilinear Interpolation . . . . .	6
2.2.3 Bicubic Interpolation . . . . .	7
2.3 Image Quality Evaluation . . . . .	12
<b>3 LITERATURE REVIEW</b> . . . . .	<b>14</b>
3.1 Single-Image Super-Resolution (SISR) Methods . . . . .	14
3.1.1 Sparse Coding . . . . .	15
3.1.1.1 Dictionary Learning . . . . .	15

3.1.1.2	Sparse Coding for Image Super Resolution . . . . .	16
3.1.2	Deep Network for Image SR . . . . .	16
3.1.2.1	Convolutional Neaural Network (CNN) Based Image SR . . . . .	17
3.2	Multi-Frame Image SR Methods . . . . .	20
3.2.1	Interpolation methods for Multi-frame image SR . . . . .	20
3.2.2	Frequency Domain Methods . . . . .	20
3.2.3	Regularization Methods . . . . .	21
<b>4</b>	<b>THE PROPOSED WORK . . . . .</b>	<b>22</b>
4.1	Contribution . . . . .	22
4.2	Motivation . . . . .	22
4.3	Framework of Proposed Method . . . . .	24
4.4	Summary . . . . .	27
<b>5</b>	<b>EXPERIMENTS AND RESULTS . . . . .</b>	<b>28</b>
5.1	Experiment-1: Simulation for Existing Datasets . . . . .	28
5.2	Expriment-2: Retinal Image Super Resolution . . . . .	37
5.3	Expriment-3: Brain MRI Image Super Resolution . . . . .	39
5.4	Summary . . . . .	39
<b>6</b>	<b>CONCLUSION . . . . .</b>	<b>41</b>
	<b>BIBLIOGRAPHY . . . . .</b>	<b>42</b>

## LIST OF TABLES

4.1	PSNR(dB) results of SC, SRCNN, SCN and interpolation methods.	23
5.1	PSNR(dB)/SSIM results of HR images, scaling factor=2, set5. . . .	29
5.2	PSNR(dB)/SSIM results of HR images, scaling factor=3,set5. . . .	29
5.3	PSNR(dB)/SSIM results of HR images, scaling factor=4, set5. . . .	29
5.4	PSNR(dB)/SSIM results of HR images, scaling factor=2, set14. . .	33
5.5	PSNR(dB)/SSIM results of HR images, scaling factor=3, set14. . .	33
5.6	PSNR(dB)/SSIM results of HR images, scaling factor=4, set14. . .	33
5.7	Image quality matrix for HR images, scaling factor=3 . . . . .	37
5.8	Image quality matrix for HR images, scaling factor=2 . . . . .	39

## LIST OF FIGURES

2.1	Taxonomy of super-resolution techniques [15]. . . . .	4
2.2	Nearest neighbor interolation . . . . .	5
2.3	Original image and nearest neighbor interpolated image. . . . .	6
2.4	Bilinear interolation . . . . .	7
2.5	Original image and bilinear interpolated image. . . . .	8
2.6	Original image and bicubic interpolated image. . . . .	10
2.7	Original image and the comparison of interpolation methods. . . . .	11
3.1	(a) Uses LR pixels from multiple LR images to create a HR image. (b) Extracts LR patches from a single LR image and uses them to reconstruct a HR image of the same scene with LR image. [30] . . .	15
3.2	SRCNN layers [2] . . . . .	17
3.3	VDSR Network [31] . . . . .	18
3.4	MSCN Network [33] . . . . .	19
3.5	An example for multi-frame SR [34] . . . . .	20
4.1	Visual comaparison of State-of-the-Art models and interpolation methods . . . . .	23
4.2	Framework of proposed method . . . . .	24
4.3	Example feature maps learned by CNN [2] . . . . .	25
4.4	LISTA network generates sparse coefficients $\alpha$ as an output from LR image patch $y$ . . . . .	26

5.1	Visual comaparison between proposd method and State-of-the-Art models and interpolation methods under scaling factor=2, set5. . .	30
5.2	Visual comaparison between proposd method and State-of-the-Art models and interpolation methods under scaling factor=3, set5. . .	31
5.3	Visual comaparison between proposd method and State-of-the-Art models and interpolation methods under scaling factor=4, set5. . .	32
5.4	Visual comaparison between the proposd method and State-of-the-Art models and interpolation methods under scaling factor=2, set14. . .	34
5.5	Visual comaparison between the proposd method and State-of-the-Art models and interpolation methods under scaling factor=3, set14. . .	35
5.6	Visual comaparison the between proposd method and State-of-the-Art models and interpolation methods under scaling factor=4, set14. . .	36
5.7	Super Resolution results for a retinal image. <i>a</i> is an original image from the retinal image database. <i>b</i> is down sampled version of LR image. <i>c</i> , <i>d</i> , and <i>e</i> are implementation results of Bicubic, Sparse Coding [1] and SRCNN [2] respectively. <i>e</i> is the result of our proposed method. . . . .	38
5.8	Visual comaparison of MRI images between the proposed method and State-of-the-Art models and interpolation methods under scaling factor=2. . . . .	40

## LIST OF SYMBOLS

$Y$	Low resolution image
$X$	High resolution image
$y$	Low resolution image patch
$x$	High resolution image patch
$\alpha$	Sparse coefficient
$D_y$	Low resolution image patch dictionary
$D_x$	High resolution image patch dictionary
$PSNR$	Peak signal-to-noise ratio
$SSIM$	Structural similarity
$RMSE$	Root mean square error
$\sigma$	Standard deviation
$\mu$	Intensity
$\lambda$	Regularization coefficient
$\theta$	Shrinkage threshold
$sign$	Signum function
$\nu$	Step size for gradient descent

## ABSTRACT

In this thesis, we propose a convolutional neural network (CNN) based single image super-resolution network model with sparse representation by combining three existing state-of-the-art methods SC [1], SRCNN [2] and SCN [3] models with a modified pre-processing step. Firstly, standard gaussian box filter is applied to test image, which is a low-resolution image (LR), to remove low-frequency noises. After that, the given low-resolution image is up-scaled by bicubic interpolation method to the same size with desired output high-resolution image (HR). Secondly, a convolutional neural network based dictionary learning method is employed to train input low-resolution image to obtain LR image patches. Also, a rectified linear unit (ReLU) thresholds the output of the CNN to get a better LR image dictionary. Thirdly, to get optimal sparse parameters, we adopted Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) [4] [5] network to train LR image patches. LISTA is a sparse-based network that generates sparse coefficients from each LR image patches. Finally, in the reconstruction step, corresponding high-resolution image patches are obtained by multiplying low-resolution image patches with optimal sparse coefficients. Then corresponding high-resolution image patches are combined to get final HR image. The experimental results show that our proposed method demonstrates outstanding performance compare to other state-of-the-art. The proposed method generates clear and better-detailed output high-resolution images since it is important in real life applications. The advantage of the proposed method is to combine convolutional neural network based dictionary learning and sparse-based network training with better pre-processing to create efficient and flexible single-image-super-resolution network.

## Chapter 1

### INTRODUCTION

We can classify images as high-resolution (HR) and low-resolution (LR) images basically. High-resolution images have more pixel density than low-resolution images. With this feature, high-resolution images are desired for much real-life application because HR images provide more detail and information about the scene. These advantages lead much research to achieve higher resolution of images to get better performance in their imaging systems. For instance, a medical doctor in neurology area can achieve better diagnosis by using higher-resolution Magnetic Resonance Imaging (MRI) images [6]. The output images of technical areas such as remote sensing, magnetic resonance imaging, etc. might be unsatisfying regarding resolution by several factors. These factors can be technical or environmental. For example, low quality of cameras and their limited resolutions, and non-stability of the observed object or scene. Therefore, acquired images might be noisy, indistinct and insufficient in spatial and temporal resolution [7]. Super-resolution techniques could be a remedy for these ill-posed problems [8].

Super-resolution (SR) estimates high resolution (HR) image from one or multiple low resolution (LR) images of the same scene [9]. Single Image Super Resolution (SISR) problem aims to gain an HR image from an LR image by deriving all the missing high-frequency components [3]. As long as super-resolution problem is concerned significantly, besides other methods such as sparse coding [1], deep learning [3] and machine learning [2] techniques have been started applicable for this ill-posed problem.

In the medical imaging area, the quality and resolution of images are vital to get better information for diagnosis or examination. SISR has been using for numerous medical imaging applications such as X-Ray [10], MRI [11], PET scans [12] or cardiac

image super-resolution [13]. Image resolution is a fundamental feature of retinal image inspection. An optical camera monitors through the pupil of the eye to the rear surface of the eyeball. The parts of the human eye such as optic nerve, fovea, retinal layers, and vessels are shown in images taken by the camera. Ophthalmologists, the medical doctors of the human eye, use those images to understand the function, the structure and the illness of the human eye. They consider visible differences or anomalies in the eyes during a clinical test. Afterward, the ophthalmologists evaluate these examination results to diagnose the subject [14].

## 1.1 Thesis Structure

In this thesis, we present an efficient and precise outstanding model for single image super-resolution problem. We attempt to solve the problem by training low resolution and high-resolution image patches for a deep learning network. We obtain sparse codes as an output of the network to reconstruct our final high-resolution image. Then we compare the results with some other available state-of-art methods.

After a brief introduction in the first chapter, we give some background information in chapter II. Then we provide a literature review including some state-of-the-art models related to the proposed method in chapter III. We explain details of the proposed method in Section IV and show how it differs from traditional sparse coding method [1]. In Section V, we present three different experiments to test the proposed method and describe the implementation details with the simulation results. These results include a comparison of other state-of-art methods. Then we conclude proposed method in the Sections IV.

## Chapter 2

### BACKGROUND AND RELATED WORK

Before taking a look at super-resolution literature and the proposed method, it is helpful to give brief information of the background required for better understanding of this study.

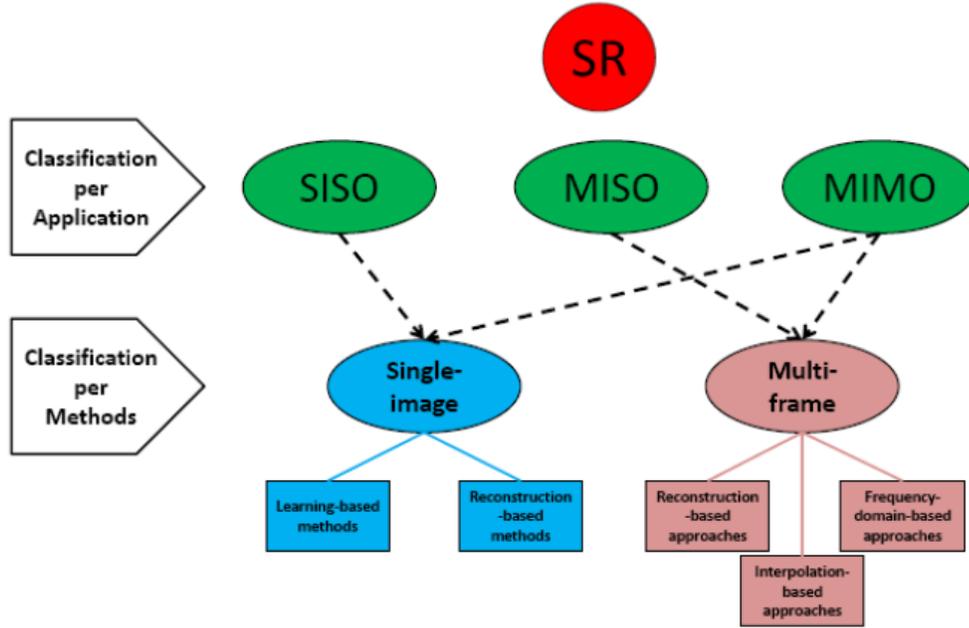
#### 2.1 Super Resolution Basics

Development of hardware has a significant impact on image resolution. However, replacing the equipment in an imaging system might cost a lot. Digital image processing techniques are a remedy to avoid big hardware costs. SR assigns to obtain a high-resolution image as output from single or multiple low-resolution images by using digital image processing techniques. There are three primary type of SR models regarding the number of inputs and outputs [15].

1. Single input single output (SISO)
2. Multiple input multiple output (MISO)
3. Multiple input multiple output (MIMO)

Figure 2.1 shows the taxonomy of super-resolution techniques and their relations regarding the number of input and output.

We can conclude two main SR methods from [15]: Single image super-resolution (SISR) and multi-frame super-resolution. SISR uses single LR image as an input, and reconstruct HR by finding missing high-frequency components. There are several methods to estimate those missing components and reconstruction. We can categorize SISR algorithms into two types [8]:



**Figure 2.1.** Taxonomy of super-resolution techniques [15].

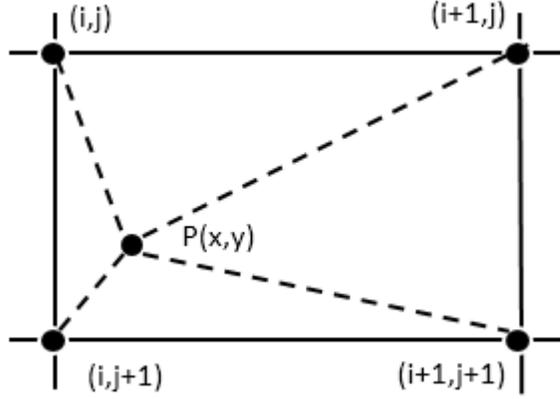
1. Learning Methods: HR components are estimated locally by using ML techniques. These may be pixel based [16] [17] or patch-based [18]. we can put Neighbor embedding [19] and sparse coding [1] algorithms in this category.
2. Reconstruction methods: Most of them are example based methods which use prior information from LR and HR. Neighbor embedding regressin [20] and deep convolutional neural network (CNN) [2] represent this type.

## 2.2 Interpolation Methods

There are three most common interpolation methods: nearest-neighbor, bilinear and bicubic interpolations.

### 2.2.1 Nearest-Neighbor Interpolation

Nearest neighbor interpolation is the most simple method to interpolate the pixels to the input low-resolution image. According to this method, each interpolated output pixel is referred from nearest sample pixel in the input image. In the figure 4.1a, known pixel values,  $(i,j)$ ,  $(i+1,j)$ ,  $(i,j+1)$ ,  $(i+1,j+1)$  and unknown pixel P to be interpolated were shown.



**Figure 2.2.** Nearest neighbor interolation

We first determine the distance from unknown point  $(x,y)$  to each known pixel point. Then we select nearest pixel values; in other word, minimum distance to unknown point as shown below:

$$D_{min} = \min\{D[(x, y), (i, j)], D[(x, y), (i, j + 1)], \quad (2.1)$$

$$D[(x, y), (i + 1, j)], D[(x, y), (i + 1, j + 1)]\} \quad (2.2)$$

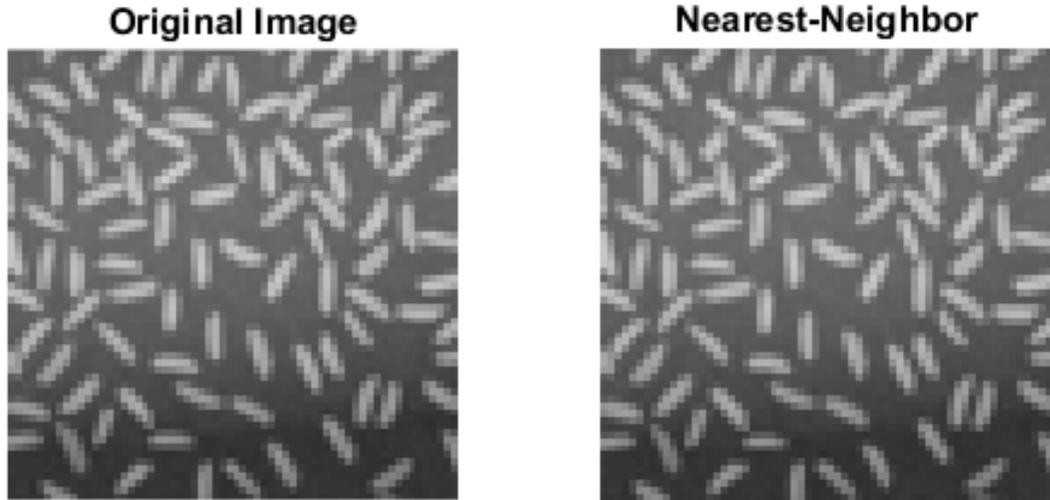
where  $D$  represents the distance between points. We can see in the figure 2.2 obviously that  $D_{min} = D[(x, y), (i, j + 1)]$ . Therefore, the value of the unknown point  $P$  will be  $(i,j+1)$  [6]. This interpolation technique uses a limited spatial kernel to estimate neighbor pixel values. The kernel for the nearest-neighbor interpolation is defined as:

$$y(x) = \begin{cases} 0 & |x| > 0 \\ 1 & |x| < 0 \end{cases} \quad (2.3)$$

The frequency response of the interpolation is

$$y(\omega) = \text{sinc}(\omega/2) \quad (2.4)$$

Although this interpolation method is quite simple, the quality of the output image is very poor [21]. Blurring and aliasing usually occur after applying the kernel [22] As we see in the figure 2.3, there is a minor improvement in the image quality. Thus, nearest-neighbor interpolation is less efficient method compare to other interpolation techniques.



**Figure 2.3.** Original image and nearest neighbor interpolated image.

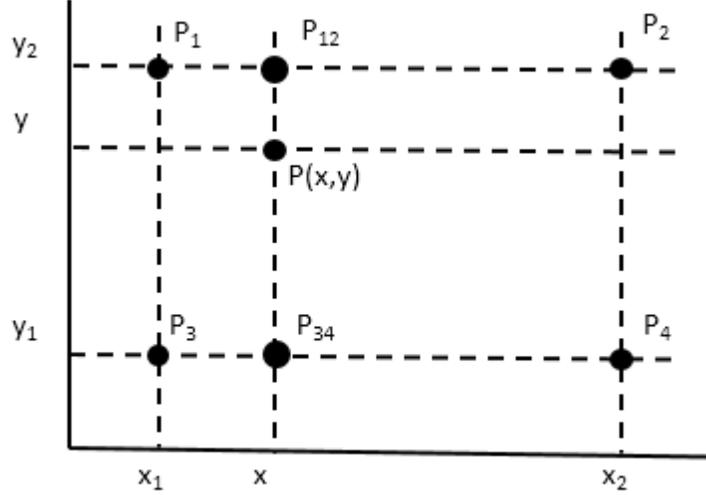
### 2.2.2 Bilinear Interpolation

Bilinear interpolation provides better image performance compare to the nearest neighbor interpolation. This interpolation method takes weighted average of the four closest pixels  $P_1, P_2, P_3, P_4$  to the certain point  $P(x, y)$  in the input image as shown in Figure 2.4. Then, refers that average value to the output image. The two linear interpolations are applied in one direction, and the next linear interpolation is applied in the perpendicular direction.

We assume an unknown function  $f$  takes values of those four known pixels as  $f(P_1), f(P_2), f(P_3), f(P_4)$  to determine unknown pixel value  $P(x,y)$ . First, we interpolate in x direction as:

$$f(P_{12}) = \frac{x_2 - x}{x_2 - x_1} f(P_1) + \frac{x - x_1}{x_2 - x_1} f(P_2) \quad (2.5)$$

$$f(P_{34}) = \frac{x_2 - x}{x_2 - x_1} f(P_3) + \frac{x - x_1}{x_2 - x_1} f(P_4) \quad (2.6)$$



**Figure 2.4.** Bilinear interpolation

Then we interpolate in  $y$  direction between  $P_{12}$  and  $P_{34}$  to find  $f(P)$  as in 2.7.

$$f(P) = \frac{1}{x_2 - x_1} f(P_1)(x_2 - x)(y - y_1) + f(P_2)(x - x_1)(y - y_1) + f(P_3)(x_2 - x)(y_2 - y) + f(P_4)(x - x_1)(y_2 - y) \quad (2.7)$$

The bilinear interpolation kernel is defined as

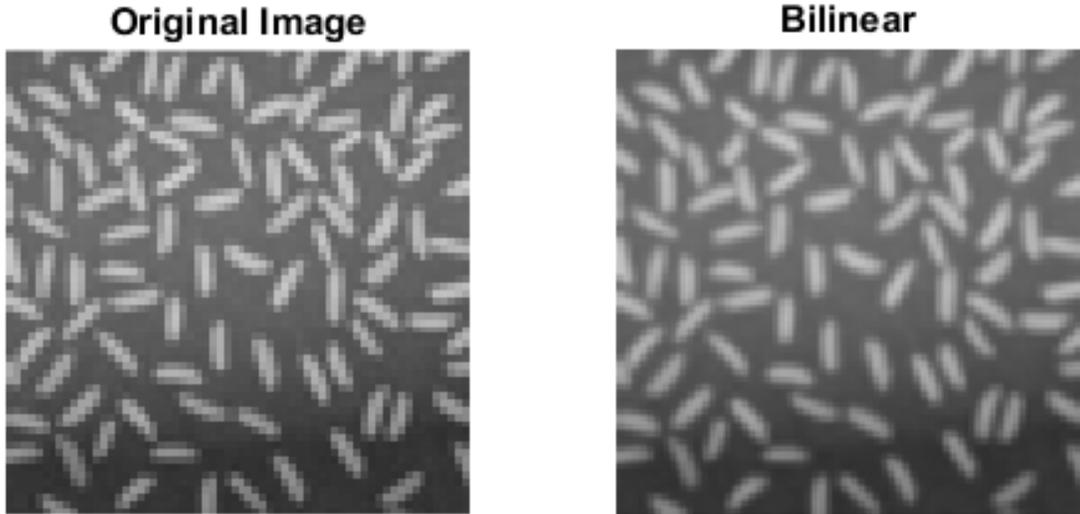
$$f(x) = \begin{cases} 0 & |x| > 1 \\ 1 - |x| & |x| < 1 \end{cases} \quad (2.8)$$

where  $x$  is distance between two points in the input image to interpolate [21].

We can see the resolution improvement in the figure 2.5 obviously.

### 2.2.3 Bicubic Interpolation

In two dimensional space, the bicubic interpolation provides a better quality image as an output than above techniques, bilinear, and nearest-neighborhood interpolations. The interpolated image has better and smoother surface with bicubic interpolation. This method assigns polynomials or cubic convolution algorithm. The Cubic Convolution Interpolation obtains the grey level value from the weighted average



**Figure 2.5.** Original image and bilinear interpolated image.

of the 16 closest pixels to certain input image points and employs the grey level values to the output image. Bicubic Interpolation evaluates 16 grid points with interpolation function. We Assume that w know the function  $f$  and its derivatives  $f_x, f_y$  and  $f_{xy}$  at the unit square  $(0,0),(1,0),(0,1),(1,1)$  on the input image. Then interpolated surface can be defined as:

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 \alpha_{ij} x^i y^j \quad (2.9)$$

To interpolate the certain unit square, we need to find 16 coefficients  $\alpha_{ij}$ . Four function values correspond with matching  $p(x, y)$  points:

$$f(0, 0) = p(0, 0) = \alpha_{00} \quad (2.10)$$

$$f(1, 0) = p(1, 0) = \alpha_{10} \quad (2.11)$$

$$f(0, 1) = p(0, 1) = \alpha_{01} \quad (2.12)$$

$$f(1, 1) = p(1, 1) = \alpha_{11} \quad (2.13)$$

The derivatives of eight functions in the  $x$  and  $y$  directions:

$$f_y(0, 0) = p_y(0, 0) = \alpha_{01} \quad (2.14)$$

$$f_y(1, 0) = p_y(1, 0) = \alpha_{01} + \alpha_{11} + \alpha_{21} + \alpha_{31} \quad (2.15)$$

$$f_y(0, 1) = p_y(0, 1) = \alpha_{01} + 2\alpha_{02} + 3\alpha_{03} \quad (2.16)$$

$$f_y(1, 1) = p_y(1, 1) = \sum_{i=1}^3 \sum_{j=0}^3 \alpha_{ij} i \quad (2.17)$$

$$f_x(0, 0) = p_x(0, 0) = \alpha_{10} \quad (2.18)$$

$$f_x(1, 0) = p_x(1, 0) = \alpha_{10} + 2\alpha_{20} + 3\alpha_{30} \quad (2.19)$$

$$f_x(0, 1) = p_x(0, 1) = \alpha_{10} + \alpha_{11} + \alpha_{12} + \alpha_{13} \quad (2.20)$$

$$f_x(1, 1) = p_x(1, 1) = \sum_{i=0}^3 \sum_{j=1}^3 \alpha_{ij} j \quad (2.21)$$

Lastly, four equations from partial derivatives for  $xy$  :

$$f_{xy}(0, 0) = p_{xy}(0, 0) = \alpha_{11} \quad (2.22)$$

$$f_{xy}(1, 0) = p_{xy}(1, 0) = \alpha_{11} + 2\alpha_{21} + 3\alpha_{31} \quad (2.23)$$

$$f_{xy}(0, 1) = p_{xy}(0, 1) = \alpha_{01} + 2\alpha_{12} + 3\alpha_{13} \quad (2.24)$$

$$f_{xy}(1, 1) = p_{xy}(1, 1) = \sum_{i=1}^3 \sum_{j=1}^3 \alpha_{ij} ij \quad (2.25)$$

Above unknown parameters  $\alpha_{ij}$  and functions  $f_{ij}$  can be shown in vector form as:

$$\alpha = [\alpha_{00}, \alpha_{10}, \alpha_{20}, \alpha_{30}, \alpha_{01}, \alpha_{11}, \alpha_{21}, \alpha_{31}, \alpha_{03}, \alpha_{13}, \alpha_{23}, \alpha_{33}]^T \text{ and}$$

$$x = [f(0, 0), f(1, 0), f(0, 1), f(1, 1), f_x(0, 0), f_x(1, 0), f_x(0, 1), f_x(1, 1), \dots$$

$$f_y(0, 0), f_y(1, 0), f_y(0, 1), f_y(1, 1), f_{xy}(0, 0), f_{xy}(1, 0), f_{xy}(0, 1), f_{xy}(1, 1)]^T$$

Then we can define above definitions with a linear equation:

$$A\alpha = x. \quad (2.26)$$

And unknown coefficients can be found when we invert the equation

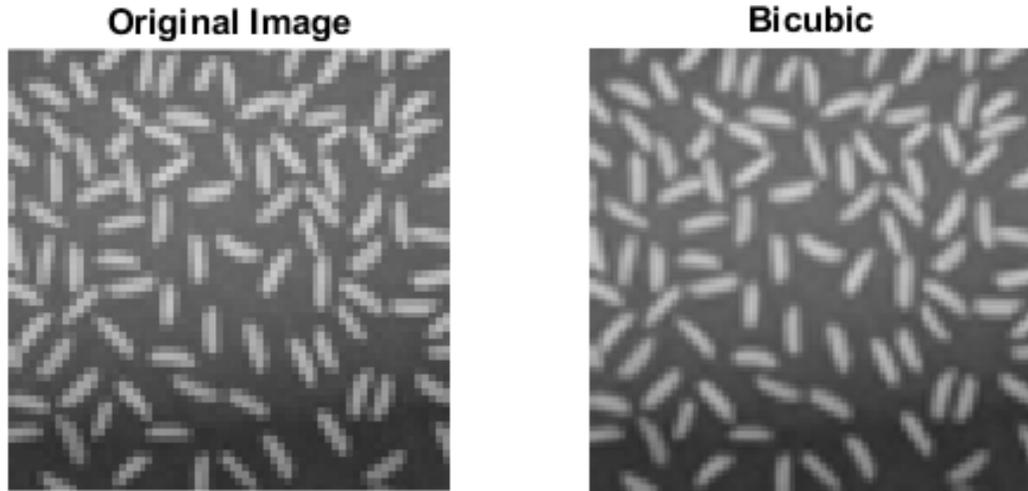
$$A^{-1}x = \alpha \quad (2.27)$$

Those coefficients are used in equation.2.9 to find bicubic interpolated points  $P(x, y)$  [23].

The bicubic convolution interpolation kernel is presented as:

$$h(x) = \begin{cases} (\alpha + 2)|x|^3 - (\alpha + 3)|x|^2 + 1 & |x| \leq 1 \\ \alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha & 1 < |x| < 2 \\ 0 & otherwise \end{cases} \quad (2.28)$$

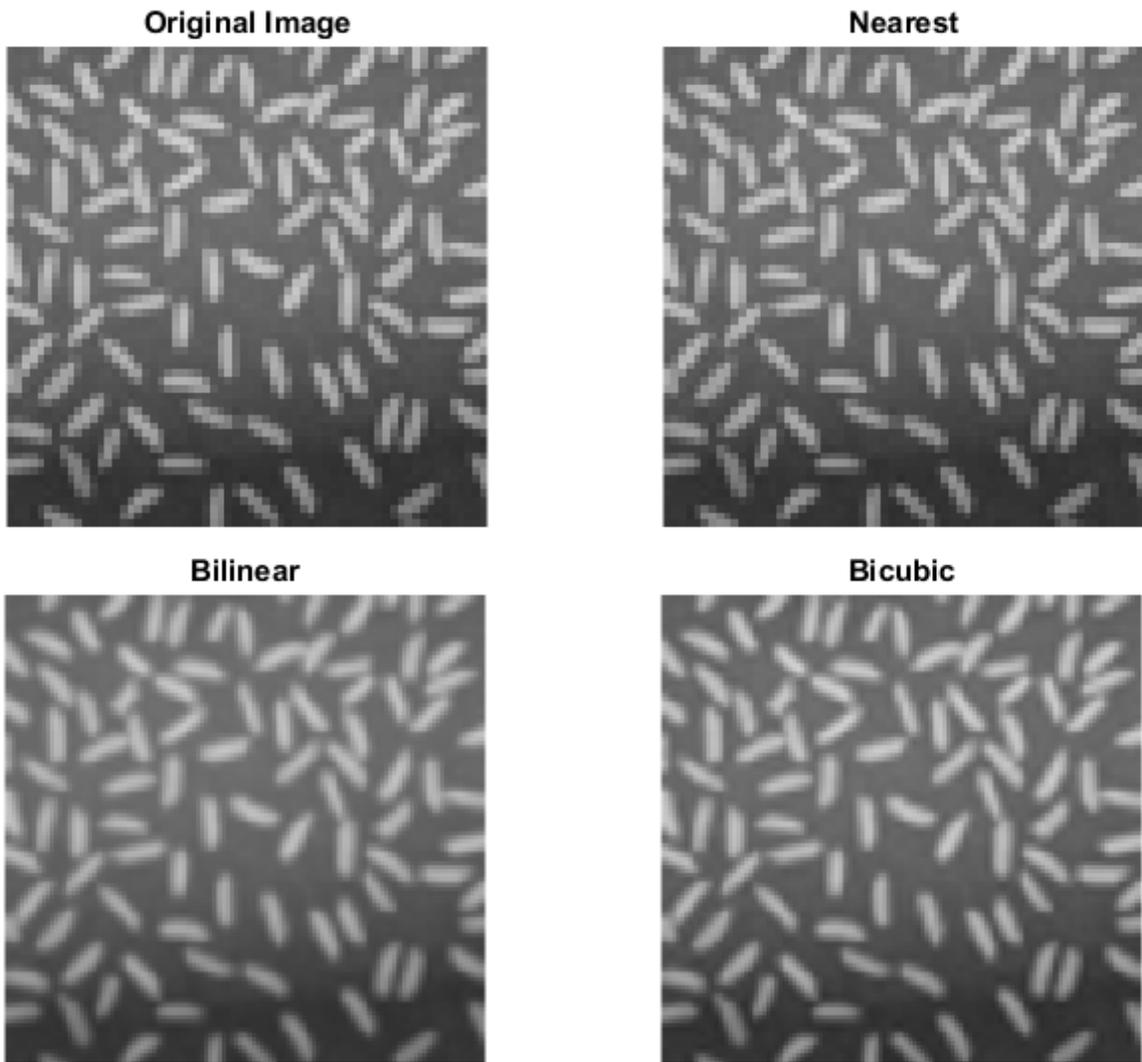
where  $\alpha$  is between -0.5 and -0.75 in most cases [21].



**Figure 2.6.** Original image and bicubic interpolated image.

As we see in the figure 2.6, there is a significant improvement on the bicubic interpolated image compare to the original image.

In the figure 2.7, we visually compared interpolation methods. We can see the differences regarding image qualities obviously. As we monitor the visual results, bicubic and bilinear interpolated images outperformed over nearest-neighbor interpolated image regarding image quality.



**Figure 2.7.** Original image and the comparison of interpolation methods.

### 2.3 Image Quality Evaluation

Image quality evaluation is a very significant task that will show the ability and strength of the methods in image processing area. There is various image quality methods are presented in the literature. The most common techniques in single image super-resolution are Structural Similarity(SSIM), Peak Signal-to-Noise Ratio (PSNR) and Root Mean Square Error (RMSE) because they are simple and efficient. In our experimental results, these three methods have been chosen to evaluate the reconstructed high-resolution images from our proposed algorithm. Then we use these image quality evaluations to compare our proposed way with other state-of-the-art techniques.

The most famous evaluation method in signal processing, especially in image processing, is PSNR. To calculate PSNR, we first find the mean squared error (MSE), which is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_g(i, j) - I_d(i, j)]^2 \quad (2.29)$$

where  $I_g$  is the ground truth, and  $I_d$  is the output image with size  $m \times n$ . We obtain the smiliarity between two image  $I_d$  and  $I_d$  by using equation 2.29. Then we define PSNR via MSE as:

$$PSNR = 10 \cdot \log_{10} \left[ \frac{(MAX_I - MIN_I)^2}{MSE} \right] \quad (2.30)$$

Where  $MAX_I$  stands for the most significant pixel value and  $MIN_I$  is the smallest pixel values of the image. Typically, this definition satisfies one channel images such as gray images. However, If we want to evaluate RGB images, then we need to calculate PSNR for each of channels then compute the average of them. The smaller MSE means, the higher PSNR. Thus, we can say that higher PSNR indicates better image quality.

Structural Similarity(SSIM) [24] is another image quality evaluation method that is used commonly. This technique focuses on the structural features similarity between two images. These features are luminance, contrast and their structures. The SSIM uses a combination of these features to get a better comparison between two

images. It has been shown that SISIM has more visual reliability compare to PSNR [6]. The definition of the SISIM of two images  $x$  and  $y$  is shown as:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (2.31)$$

where  $l, c, s$  stands for luminance, contrast, structure functions of  $x$  and  $y$  respectively, and  $\alpha, \beta, \gamma$  are their parameters. When we set the parameters  $\alpha, \beta, \gamma$  as 1 then we can express SSIM as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.32)$$

where  $\mu_x$  and  $\mu_y$  are represents the intensity of  $x$  and  $y$  which are shown as:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i, \quad \mu_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.33)$$

where  $\sigma_x$  and  $\sigma_y$  stand for the standard deviation of  $x$  and  $y$  which are dfined as:

$$\sigma_x = \left[ \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right]^{1/2}, \quad \sigma_y = \left[ \frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2 \right]^{1/2} \quad (2.34)$$

also  $C_1$  and  $C_2$  are constants which are shown as:

$$C_1 = (0.01L)^2, \quad C_2 = (0.03L)^2 \quad (2.35)$$

where  $L$  defines the dynamic range of the pixel values [6].

The last image quality evaluation method that we use in this thesis is Root Mean squared error (RMSE). The computation of RMSE of two image  $x$  and  $y$  is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}} \quad (2.36)$$

The RMSE is also defined as simple as:

$$RMSE = \sqrt{1 - r^2} \cdot \sigma_y \quad (2.37)$$

where  $\sigma_y$  stands for standard deviation of  $y$  and  $r$  is between -1 and 1. This equation is more simple and efficient than previous one.

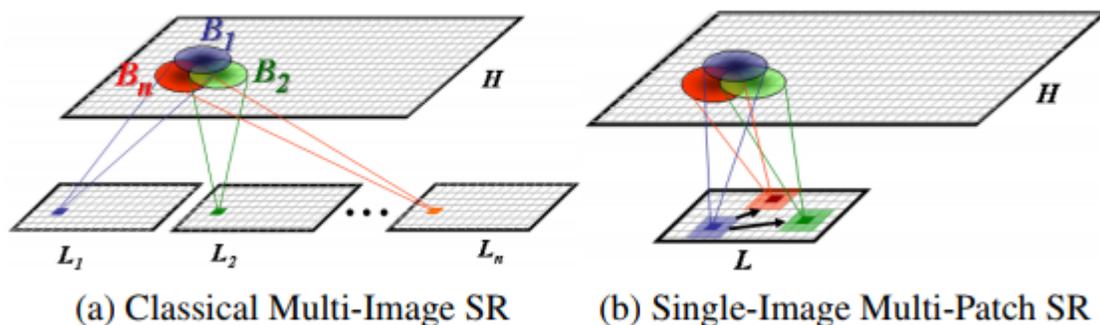
## Chapter 3

### LITERATURE REVIEW

There are several methods to solve super-resolution problem in the literature besides interpolation techniques. Super-resolution through neighbor embedding (NE+) [19] uses the nearest neighbor embedding method to find nearest neighbors among LR patches in the dictionary and search their corresponding HR patches to reconstruct final SR output. In [1], image super-resolution via sparse representation (SrSC) represents low-resolution image patches with sparsity coefficients and then finds corresponding HR patches sparsely by using their coefficients to reconstruct SR image. A sparse coding-based super-resolution image algorithm [25] proposes a method to improve sparse coding method [26] regarding efficient dictionary learning by combining K-SVD and Orthogonal Matching Pursuit (OMP). Anchored Neighborhood Regression (ANR) [27] and the further study of [27], advanced ANR (A+) [28], use ridge regression in addition to K-SVD and SC to get better performance. The centerpiece work [2] and [29] of deep learning along SISR is Super-Resolution Convolutional Neural Network (SRCNN). Since this work was published, numerous works have been using the method as a benchmark [8]. In this chapter, we will try to describe a different type of SR models in the literature and give some examples of state-of-the-art methods in single image super-resolution (SISR) area.

#### 3.1 Single-Image Super-Resolution (SISR) Methods

In this section, we will explain some main SISR methods and their detailed backgrounds for better understanding the proposed method since SISR is the focused method in this thesis. In the SISR techniques, the most remarkable part is that there is only one input LR image  $\mathbf{Y}$  and its recovered version HR image  $\mathbf{X}$  as output.



**Figure 3.1.** (a) Uses LR pixels from multiple LR images to create a HR image. (b) Extracts LR patches from a single LR image and uses them to reconstruct a HR image of the same scene with LR image. [30]

We can see a fundamental comparison between multi-frame image super resolution and SISR in figure 3.1.

### 3.1.1 Sparse Coding

#### 3.1.1.1 Dictionary Learning

The frame work of sparse representation for single image SR [1] focuses on these two constrains and finds their sparse representation to reconstruct final HR image. Reconstruction constrain is defined as:

$$Y = SHX. \quad (3.1)$$

where  $S$  represents downsampling operator and  $H$  blurring filter.  $\mathbf{X}$  is upsampled and deblurred version of  $\mathbf{Y}$ .

First, LR image patches  $\mathbf{y} \in R^{m_y}$  are extracted from input image. From upper left corner of the input image to the bottom right end, each LR patch is normalized by its mean and variance before creating the LR patch dictionary  $\mathbf{D}_y$ . The same mean and variance are used with standard stochastic gradient descent algorithm in equation 3.3 to obtain HR patch  $\mathbf{x} \in R^{m_x}$ , and to create HR patch dictionary  $\mathbf{D}_x$  [1]. Those dictionary pair  $(\mathbf{D}_y, \mathbf{D}_x)$  is related to their sparse codes.

### 3.1.1.2 Sparse Coding for Image Super Resolution

Super resolution based on sparse representation [1] takes each LR image patches  $\mathbf{y}$  from the input LR to obtain corresponding HR image patch  $\mathbf{x}$ . Each LR and HR image patch are saved in the dictionaries  $(\mathbf{D}_y, \mathbf{D}_x)$  which are not necessarily the same size. We can represent the relation between  $\mathbf{x}$  and  $\mathbf{y}$  with some sparse linear coefficients  $\alpha_x, \alpha_y$ . Since  $\mathbf{y}$  and  $\mathbf{x}$  are almost linearly related we can assume that  $\alpha_x = \alpha_y = \alpha$  [3]. Thus, the optimal sparse representation  $\alpha$  between each input LR patch  $\mathbf{y}$  and HR patch  $\mathbf{x}$  is defined in (3.2) as:

$$x = D_x \alpha \quad \text{s.t.} \quad \underset{\alpha}{\operatorname{argmin}} \|y - D_y z\|_2^2 + \lambda \|z\|_1, \quad (3.2)$$

where  $\lambda$  represents regularization coefficients,  $\mathbf{z}$  represents sparsity coefficient matrix  $z = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n] \in R^{m \times n}$  [3], and  $l_1$  norm is to balance sparsity [1]. Finding appropriate sparse coefficients are crucial for final reconstruction. Therefore, training data parameters and LR-HR patch sizes should be chosen carefully. After the optimization problem is solved as in equation 3.2 and high resolution image patches  $\mathbf{x}$  are generated we use gradient descent algorithm to obtain closest image to  $X_0$  which is as close as possible to the ground truth image and keep updating the gradient descent to obtain optimal output HR image  $\mathbf{X}$  as below.

$$X^* = \underset{\mathbf{X}}{\operatorname{argmin}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + c\|\mathbf{X} - \mathbf{X}_0\|_2^2, \quad (3.3)$$

where  $X^*$  represent SR image as output. And the update of the gradient descent algorithm is:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \nu[H^T S^T(\mathbf{Y} - SH\mathbf{X}_t) + (X - X_0)], \quad (3.4)$$

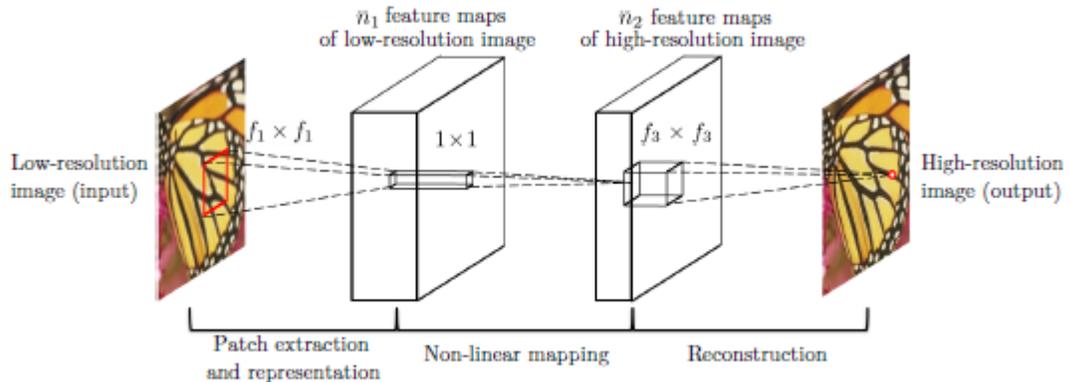
where  $\mathbf{X}_t$  represents the estimation of HR image after  $t$ th iteration,  $\nu$  is the step size for the gradient descent algorithm [1].

### 3.1.2 Deep Network for Image SR

There is a various method that uses the deep network and the idea of SRCNN [2] [29] as a framework. We try to describe SRCNN and some other leading CNN-based SR models in this section.

### 3.1.2.1 Convolutional Neural Network (CNN) Based Image SR

The cornerstone of the convolutional neural network for single image Super Resolution is called "Learning a Deep Convolutional Network for Image Super-Resolution" (SRCNN) [2] published by C. Dong et al. The method employs convolutional neural networks (CNN) for end-to-end mapping between the low-resolution image LR and the high-resolution image HR. Instead of taking each feature components one by one in the dictionaries like traditional sparse coding methods [1], SRCNN optimizes all layers at one time. As a result of this process, fast and better image quality is obtained with the SRCNN method. The SRCNN consists of three main steps as it is shown in the figure below.



**Figure 3.2.** SRCNN layers [2]

1. Patch extraction: The first layer is defined as a function set  $F_1$  as its shown in equation 3.5 . These functions are used to extract image patches by convolving the image.

$$F_1(\mathbf{Y}) = \max(0, W_1 * \mathbf{Y} + B_1), \quad (3.5)$$

where  $\mathbf{Y}$  represents the input LR image,  $W_1$  is filters and  $B_1$  is biases. The size of  $W_1$  is  $c \times f_1 \times f_1 \times n_1$  where  $c$  represents the number of image channels,  $f_1$  is the size of filter, and  $n_1$  is the number of convolution filters. Also, the Rectified Linear Unit (ReLU) is applied on the filter output.

2. End-to-end non-linear mapping: In this process, one high-dimensional is mapped onto another vector. Each non-linearly mapped vector represents a HR image patch. The equation 3.6 defines the second layer as:

$$F_2(\mathbf{Y}) = \max(0, W_2 * F_1 \mathbf{Y} + B_2), \quad (3.6)$$

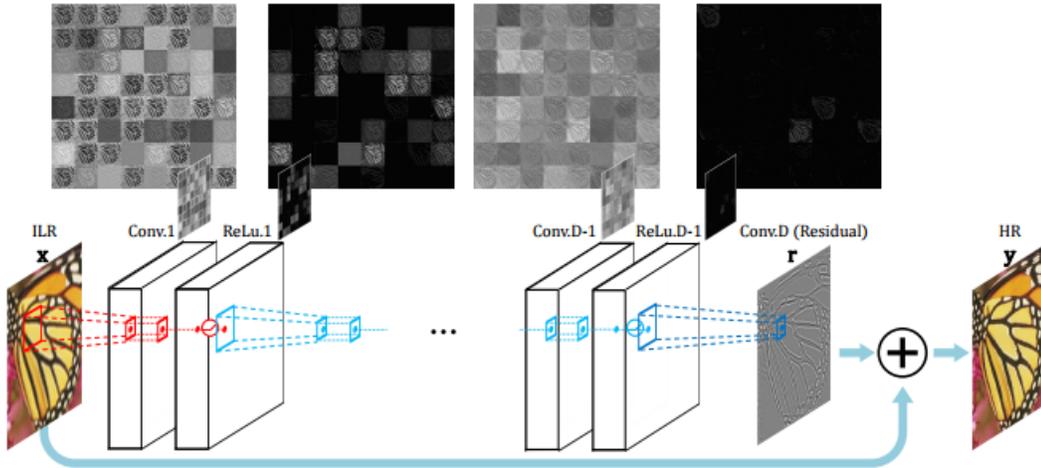
where  $B_2$  represents  $n_2$ -dimensional vector, and  $W_2$  is filters with  $n_1 \times 1 \times 1 \times n_2$  size. Each  $n_2$  sized vector represents a HR image patch. Then these vectors are used to reconstruct final HR image.

3. Reconstruction: In this layer, the generated overlapping HR patches are averaged to create the final HR image. The construction step is defined with a convolution layer which is presented as:

$$F(\mathbf{Y}) = W_3 * F_2 \mathbf{Y} + B_3, \quad (3.7)$$

where  $B_3$  is a vector with  $c$ -dimensional, and the size of  $W_3$  is  $n_2 \times f_3 \times f_3 \times c$ .

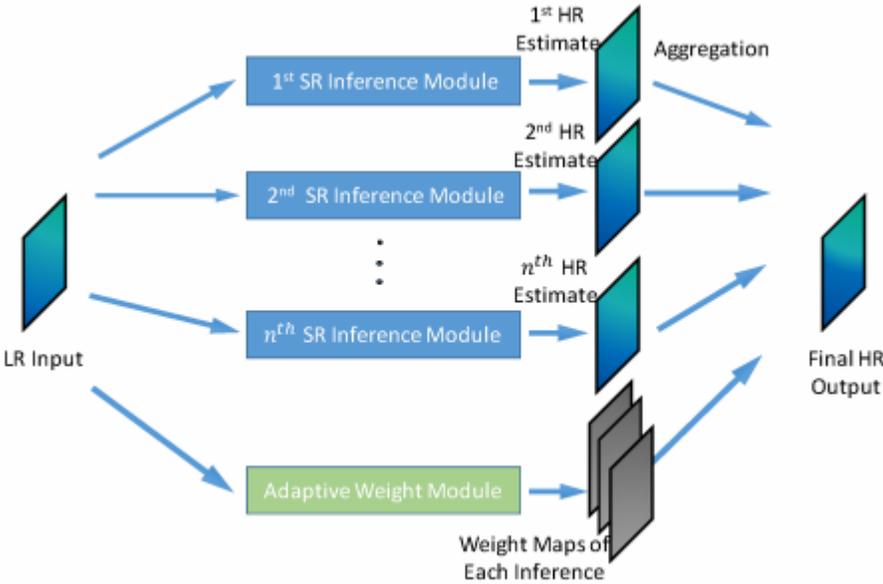
Very Deep Super-Resolution (VDSR) [31] creates many convolutional layers (approximately 20 layers) inspired by VGG-net which was used for ImageNet classification in previous works [32].



**Figure 3.3.** VDSR Network [31]

In the VDSR network shown in figure 3.3, convolutional and nonlinear layers are repeatedly cascaded to get better performance for output HR image from input

single LR image. VDSR claims superior results over SRCNN by improving limitations regarding training and network structure [31]. Although the popularity of algorithms based on sparse coding have been decreased after the domination of deep learning and CNN in the SISR area, sparse coding-based network (SCN) [3] and its advanced work "Learning a Mixture of Deep Network for Single Image Super Resolution" (MSCN) [33] show that sparse coding can be much more efficient if combined with appropriate deep learning methods. SCN and MSCN not only provide efficiency and better training but also reduces model size which means that better performance with fewer parameters compare to other sparse coding methods [8]. The MSCN network consists of SR in-



**Figure 3.4.** MSCN Network [33]

ference modules and one adaptive weight module which are applied to LR image to obtain one HR image. Then all predicted HR images are combined in aggregation layer by using adaptive weight module [33].

### 3.2 Multi-Frame Image SR Methods

The second type of SR image method is multi-frame Image SR which uses multiple LR images as the input of SR model to get one or multiple HR images as output. An example of general multi-frame SR is shown in the figure 3.5.

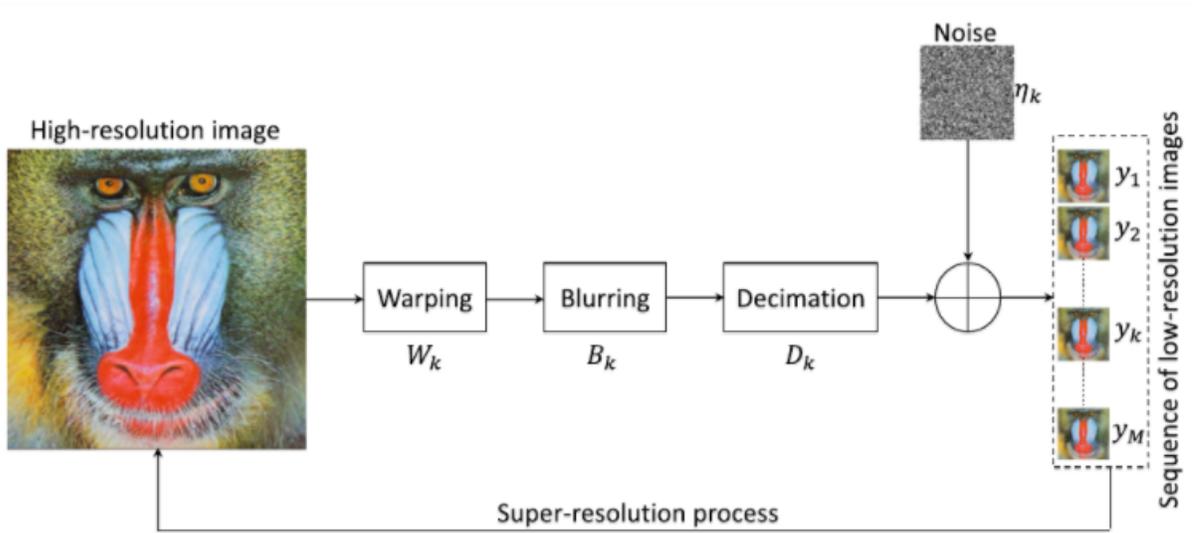


Figure 3.5. An example for multi-frame SR [34]

#### 3.2.1 Interpolation methods for Multi-frame image SR

Interpolation methods for multi-frame image SR have three steps mostly: registration, interpolation, and deblurring [35]

#### 3.2.2 Frequency Domain Methods

The studies such as "Superresolution and noise filtering using moving least squares" [36] take advantage of using Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) or Discrete Wavelet Transform(DWT) of LR images to gather absent high-frequency components of HR images.

### 3.2.3 Regularization Methods

Regularization methods for image SR employ either standard stochastic or deterministic regularization techniques when there is a limited number of input LR images. This strategy incorporates prior limited information about unknown HR image [37].

## Chapter 4

### THE PROPOSED WORK

In Chapter 3, we have shown CNN [2] and SCN [3] SR algorithms. However, these two methods can be improved further by concatenating them. Therefore, in this chapter, we propose a CNN dictionary learning based sparsity deep network method for single image super-resolution framework. Also, we use Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) network [4] [5] to get better sparse coding performance for the high-resolution image reconstruction.

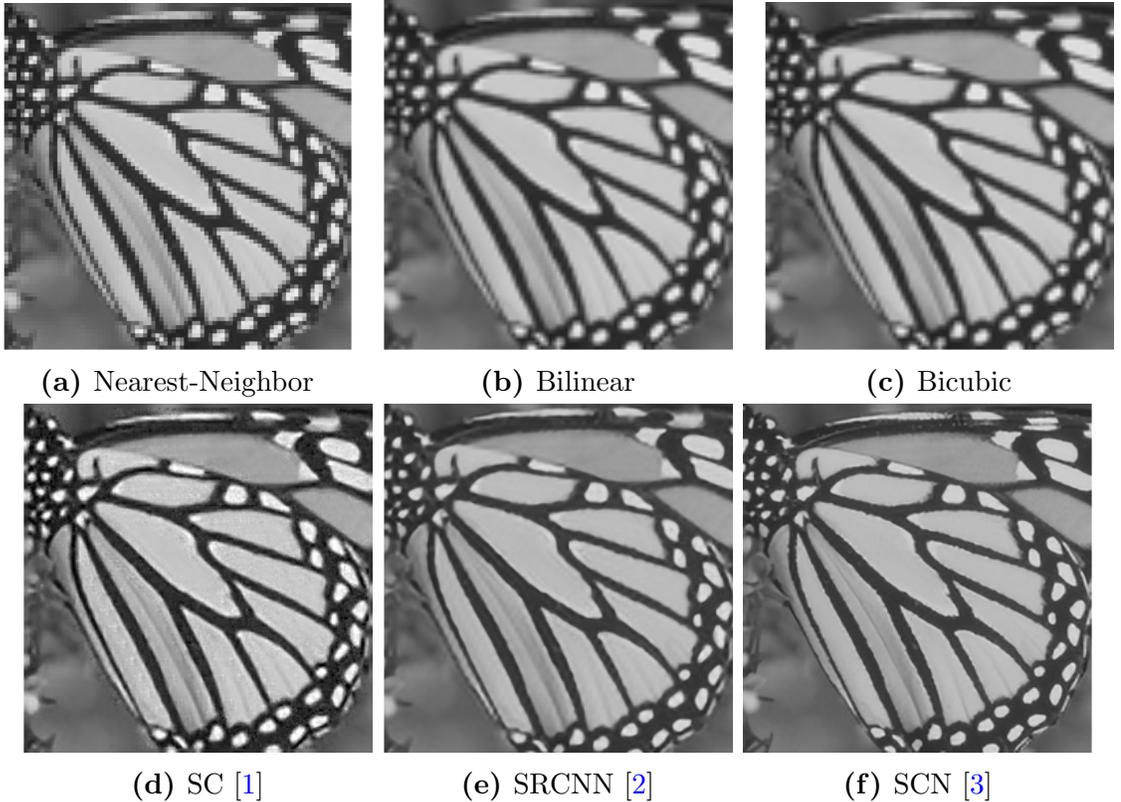
#### 4.1 Contribution

The proposed method is a combination of two existing state-of-the-art models. We noticed that the accuracy of the CNN based dictionary learning at Dong's work has a good impact on feature extraction process. So we use these features to find optimal sparse coefficients to represent each low-resolution LR image patches by using LISTA non-linear mapping network [4] [5] as Wang et al. suggested in their work [3].

Furthermore, we apply a Gaussian filter and bicubic interpolation on input LR image before the next step, patch extraction to get better image quality performance. Filtering and then upsampling the input images provides better performance as it is shown in the experimental results section.

#### 4.2 Motivation

In Chapter 3, we reviewed deep network, CNN based and Sparse Coding (SC) based methods for image super-resolution. There are pros and cons to each other regarding preprocessing, dictionary training and reconstruction steps. In this section, we have shown experimental results of SRCNN, SC, SCN and interpolation methods.



**Figure 4.1.** Visual comparison of State-of-the-Art models and interpolation methods

Figure 4.1 shows results of SC, SRCNN, SCN and interpolation methods visually. For the simulations, the scale factor was chosen as 3. The PSNR results of simulations are presented in Table 4.1. As we see on the table, SC method has better PSNR

**Table 4.1.** PSNR(dB) results of SC, SRCNN, SCN and interpolation methods.

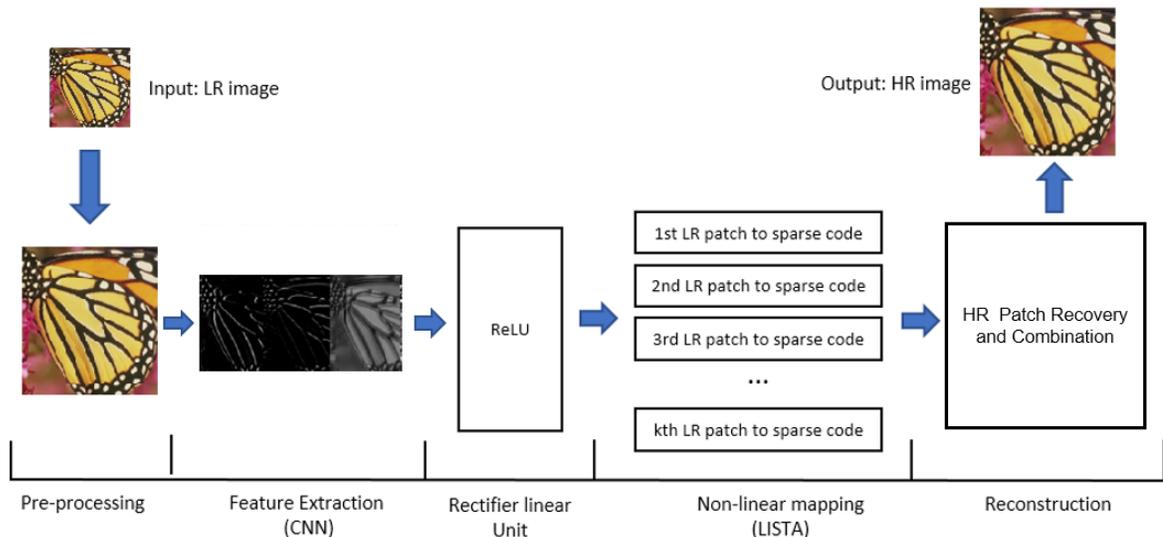
Method	PSNR
Nearest Neighbor	21.76
Bilinear	23.21
Bicubic	24.05
SC	25.16
SRCNN	27.95
SCN	32.51

performance compare to others. As we mentioned before, SCN uses a deep network method (LISTA) to get optimal sparse coefficients from LR image patches. Due to this information, we can say that this technique gives much better result than Yang’s

sparse representation method. Also, SRCNN has a better PSNR performance than SC, so that we can predict SRCNN has better dictionary learning method than SC.

From this conclusions, we can propose a method that concatenates above methods with improved preprocessing step. Moreover, we can take advantage of SRCNN and SCN for better dictionary learning and better sparse coding performance.

### 4.3 Framework of Proposed Method



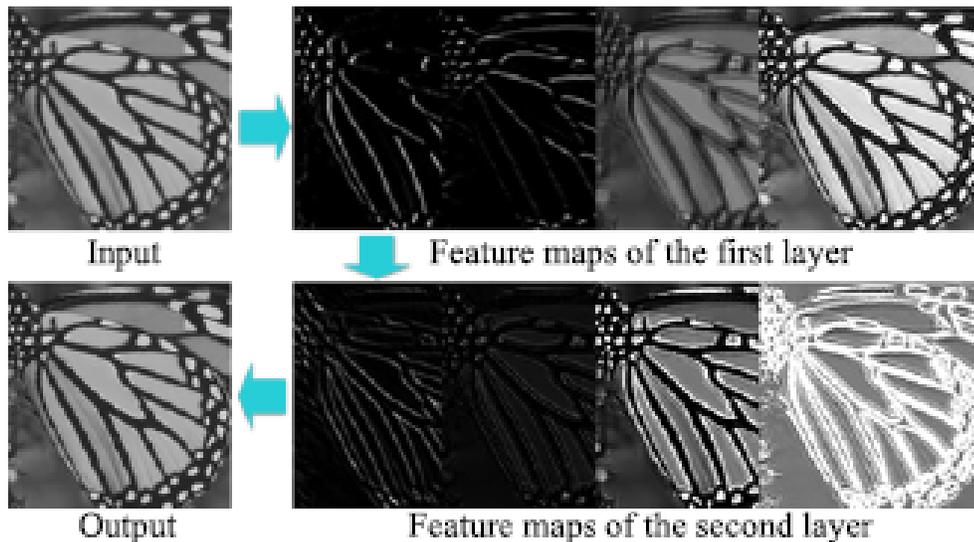
**Figure 4.2.** Framework of proposed method

The framework of proposed model has shown in Fig 4.2. We firstly apply a pre-processing step on a given LR image  $Y$ . The pre-processing step consists of Bicubic interpolation to upscale the LR image and Gaussian filtering to smooth it from minor noises. Then the next step is feature extraction by using Convolutional Neural Networks (CNN). We inspired from Dong’s model [2] to extract LR image patches from the input image and create an LR image patch dictionary  $D_y$  by using CNN. After each convolutional operation to obtain the LR patch, the rectified linear unit (ReLU) is applied to the output of the convolutional operation. The ReLU computes the function

below:

$$f(p) = \max(0, p) \quad (4.1)$$

where  $p$  is a feature in our method. As it is seen in the Equation 4.1, the activation threshold is held at zero. We use pre-trained feature weights in our model. An example



**Figure 4.3.** Example feature maps learned by CNN [2]

of feature maps has been shown in Figure 4.3.

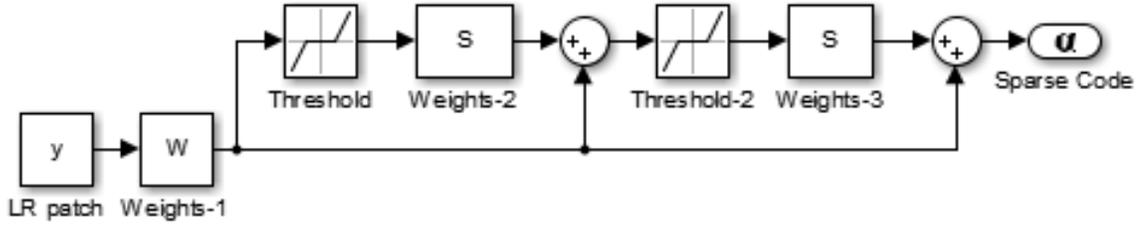
The Network for finding sparse coefficients is inspired by Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) [4], [5]. LISTA uses the relationship between sparse coding and neural network. A LISTA network generates sparse coefficients  $\alpha$  from each LR patch  $\mathbf{y}$ . The network for sparse coding has shown in figure 4.4:

The LISTA uses all network parameters such as linear weights  $W \in \mathbb{R}^{n \times m_y}$ ,  $S \in \mathbb{R}^{n \times n}$  and the shrinkage thresholds  $\theta \in \mathbb{R}^n$  and sparse coding parameters  $(D_y, \alpha)$  from training data [3]. The update of the network for sparse coding is shown as:

$$z_{k+1} = h_\theta(W_y + Sz_k), \text{ and} \quad (4.2)$$

$$[h_\theta(\alpha)]_i = \text{sign}(\alpha_i)(|\alpha_i| - \theta_i)_+, \quad (4.3)$$

where  $h_\theta$  is an element-wise shrinkage function and  $\theta$  is positive threshold. The activation thresholds  $\theta$  are updated during training process, and  $\mathbf{z}$  represents sparsity



**Figure 4.4.** LISTA network generates sparse coefficients  $\alpha$  as an output from LR image patch  $y$ .

coefficient matrix  $z = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n] \in R^{m \times n}$ . The threshold neuron consists of three parts: two linear scaling layers and one unit-threshold neuron. Therefore, those parts provide  $\theta$  adjustable threshold decomposition. We concatenate Wang’s model [3] to reconstruct the HR image. The reconstruction part consists of two steps: HR image patch recovery step  $H$  where HR patch dictionary  $D_x$  generated and a patch combination step  $G$ . In the HR patch recovery step, sparse coefficients  $\alpha$  are used to obtain corresponding HR image patches  $x$  of size  $m_x = s_x \times s_x$  for each LR image patch  $y$  by multiplying  $D_x \in \mathbb{R}^{m_x \times n}$  with sparse coefficient  $\alpha$  as below:

$$x = D_x \alpha. \quad (4.4)$$

In the next layer  $G$ , HR image patches are combined according to their corresponding locations to reconstruct final HR image  $X$ . This steps includes a convolutional filter with size  $s_g \times s_g$  and  $m_x$  channels. The number of  $s_g$  defines the number of overlapping pixels in each spatial direction. The appropriate filter weights assign each HR patches in corresponding spatial locations in the final HR image. The standard gradient descent algorithm is employed to train reconstruction layer. Standard gradient descent is defined as:

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + c\|\mathbf{X} - \mathbf{X}_0\|_2^2, \quad (4.5)$$

where  $X^*$  presents SR image. Update of the gradient descent algorithm is :

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \nu[H^T S^T(\mathbf{Y} - SH\mathbf{X}_t) + (\mathbf{X} - \mathbf{X}_0)], \quad (4.6)$$

where  $\mathbf{X}_t$  presents the estimation of HR image after  $t$ th iteration,  $\nu$  is the step size for the gradient descent algorithm [1]. Also, to train the network from LR to HR, mean square error (MSE) is used as a cost function. This optimization process is shown as:

$$\min_{\theta} \sum_i \|F(\mathbf{Y}^{(i)}; \theta) - \mathbf{X}^{(i)}\|_2^2, \quad (4.7)$$

where  $\theta$  denotes parameter used in training network,  $\mathbf{Y}^{(i)}$  and  $\mathbf{X}^{(i)}$  are  $i$ -th pair of LR and HR training images,  $F(\mathbf{Y}^{(i)}; \theta)$  defines obtained HR image  $\mathbf{X}$  by using LR image  $\mathbf{Y}$  and the set of parameters  $\theta$ .

#### 4.4 Summary

In this chapter, we proposed a CNN based single image super-resolution deep network model with sparse representation by concatenating SC [1], SRCNN [2] and SCN [3] models with modified preprocessing step. Firstly, we applied gaussian box filter to remove low-frequency noises, and then we used bicubic interpolation method to upscale the given low-resolution image. Secondly, we employed CNN based dictionary learning method to train LR image to obtain LR image patches. Thirdly, to get optimal sparse parameters, we adopted LISTA network to train LR image patches. Finally, in the reconstruction step, we obtained corresponding HR patches then combined them to get final HR image.

## Chapter 5

### EXPERIMENTS AND RESULTS

In this chapter, we present three main experiments to test the performance of our proposed method. The first experiment examines two of most familiar testing data sets. The second and third experiments are focused on two different biomedical imaging are brain MRI images and retinal images. Also, image quality evaluations are presented in each test. We use Structural Similarity(SSIM) and Peak Signal-to-Noise Ratio (PSNR) to evaluate obtained output HR images quantitatively. Also, we present simulation results visually.

#### 5.1 Experiment-1: Simulation for Existing Datasets

We use most commonly used data sets set5 [38] and set14 [39] which contain 5 and 14 colored images for testing super-resolution image models. The experiments have performed under scale factor 2,3 and 4. Also, 91 images are used to train the network to learn filters as it has done in [3] and [2]. We used those filters to convolve our input image to create LR image patch dictionary. At the beginning of the simulations, the input image LR is down-sized by bicubic interpolation method to obtain LR-HR image couples for evaluation and training. The original image (HR image) is used for evaluation, and the downsized LR image is used for training as an input to our network.

The Tables 5.1, 5.2 and 5.3 present the PSNR and SSIM of different methods compare to our proposed method. This experiment has been done under scaling factor=2,3 and 4. Set5 image data set is used as an input. We can see that PSNR result in Table 5.1 is not satisfied compare to other methods, although it has got the best SSIM performance among all methods that we compare. However, under scale factor 3 and 4 we can see that the proposed method achieved better PSNR/SSIM performance. Also,

Figure 5.1, 5.2 and 5.3 present the visual comparison between methods and proposed method under scaling factor 2,3 and 4.

**Table 5.1.** PSNR(dB)/SSIM results of HR images, scaling factor=2, set5.

Method	PSNR	SSIM
Nearest neighbor	27.13	0.9011
Bicubic	33.64	0.9292
SC	35.78	0.9485
SRCNN	36.28	0.9509
SCN	<b>37.14</b>	0.9667
Proposed Method	34.96	<b>0.9833</b>

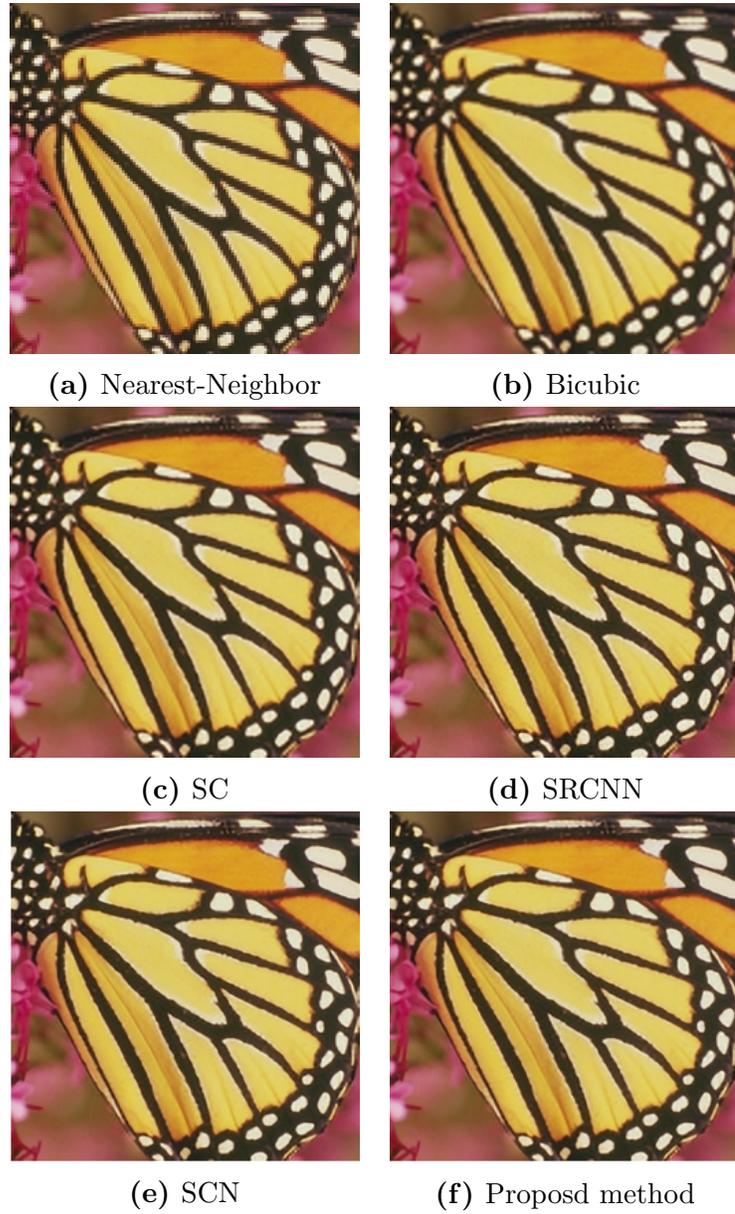
**Table 5.2.** PSNR(dB)/SSIM results of HR images, scaling factor=3,set5.

Method	PSNR	SSIM
Nearest neighbor	27.13	0.8112
Bicubic	30.39	0.8678
SC	31.34	0.8869
SRCNN	32.37	0.9025
SCN	33.26	0.9167
Proposed Method	<b>35.10</b>	<b>0.9399</b>

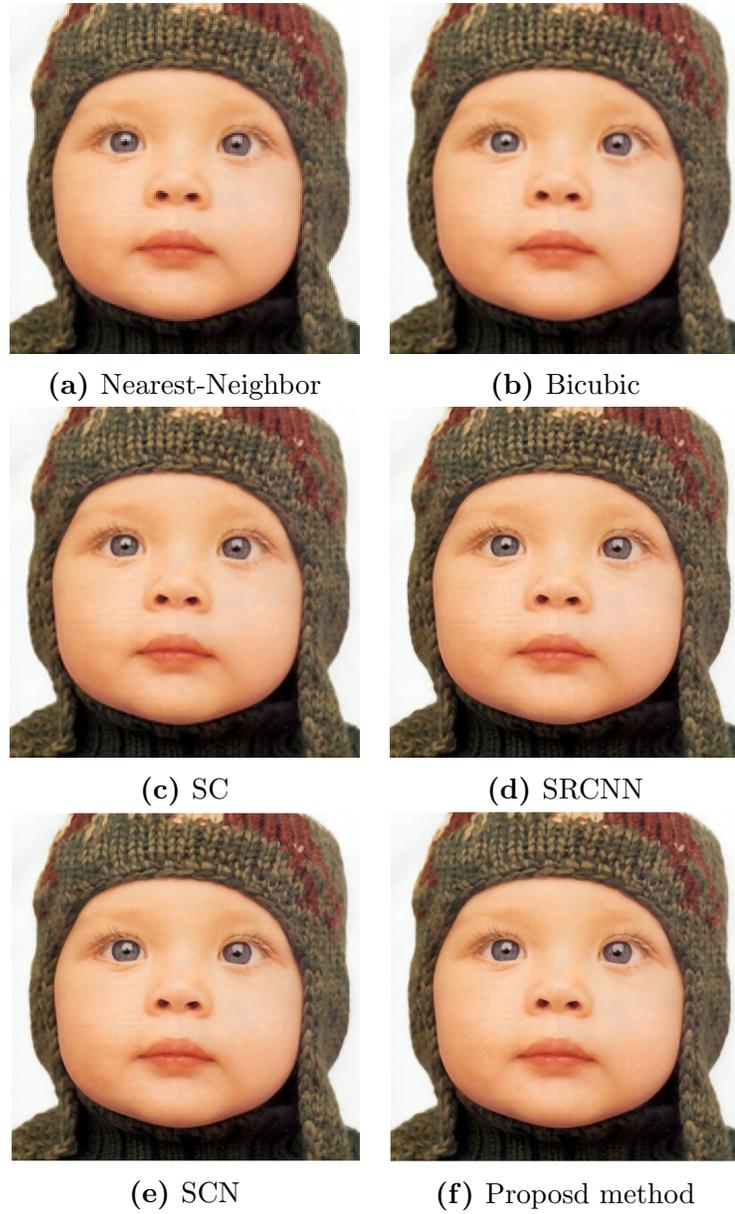
**Table 5.3.** PSNR(dB)/SSIM results of HR images, scaling factor=4, set5.

Method	PSNR	SSIM
Nearest neighbor	28.05	0.8012
Bicubic	28.42	0.8101
SC	29.07	0.8263
SRCNN	30.08	0.8525
SCN	31.04	0.8775
Proposed Method	<b>32.06</b>	<b>0.9678</b>

Tables 5.4, 5.5 and 5.6 show the PSNR/SSIM of proposed method and other methods under scaling factor=2, 3 and 4. Set14 image data set is used to test the network. We notice that the proposed method obtained better PSNR/SSIM performance. Moreover, Figure 5.4, Figure 5.5 and Figure 5.6 present the visual comparison between methods and proposed method under scaling factor 2, 3 and 4.



**Figure 5.1.** Visual comparison between proposed method and State-of-the-Art models and interpolation methods under scaling factor=2, set5.



**Figure 5.2.** Visual comparison between proposed method and State-of-the-Art models and interpolation methods under scaling factor=3, set5.



(a) Nearest-Neighbor



(b) Bicubic



(c) SC



(d) SRCNN



(e) SCN



(f) Proposed method

**Figure 5.3.** Visual comparison between proposed method and State-of-the-Art models and interpolation methods under scaling factor=4, set5.

**Table 5.4.** PSNR(dB)/SSIM results of HR images, scaling factor=2, set14.

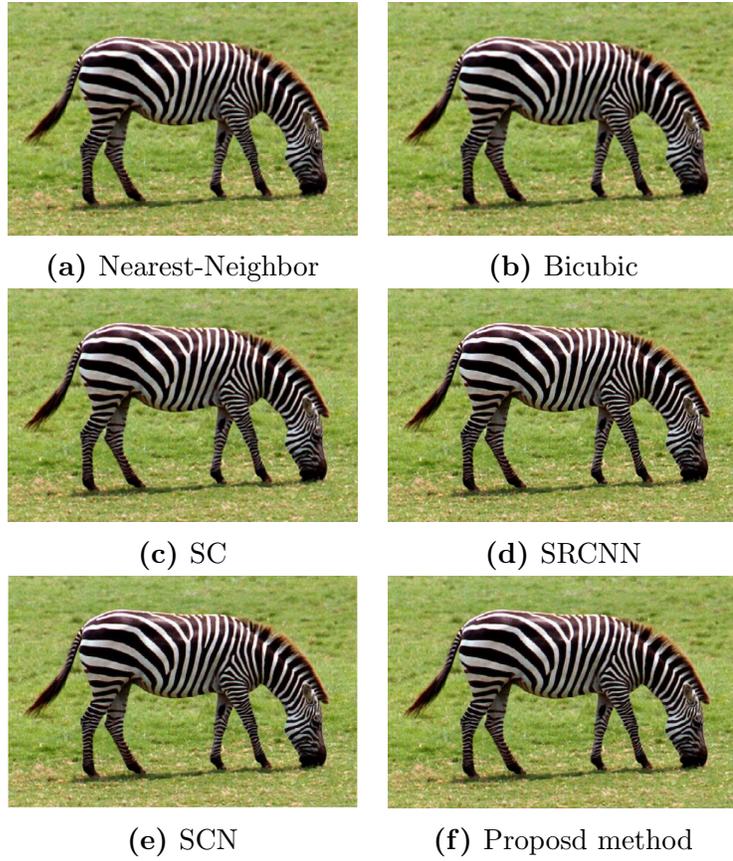
Method	PSNR	SSIM
Nearest neighbor	27.34	0.8121
Bicubic	30.22	0.8683
SC	31.64	0.8940
SRCNN	32.00	0.9012
SCN	32.71	0.9095
The Proposed Method	<b>33.37</b>	<b>0.9432</b>

**Table 5.5.** PSNR(dB)/SSIM results of HR images, scaling factor=3, set14.

Method	PSNR	SSIM
Nearest neighbor	25.12	0.7101
Bicubic	27.53	0.7737
SC	28.19	0.7977
SRCNN	28.90	0.8124
SCN	29.55	0.8271
The Proposed Method	<b>30.26</b>	<b>0.8308</b>

**Table 5.6.** PSNR(dB)/SSIM results of HR images, scaling factor=4, set14.

Method	PSNR	SSIM
Nearest neighbor	24.64	0.6863
Bicubic	25.99	0.7023
SC	26.40	0.7218
SRCNN	27.13	0.7395
SCN	27.76	0.7620
The Proposed Method	<b>30.26</b>	<b>0.8308</b>



**Figure 5.4.** Visual comparison between the proposed method and State-of-the-Art models and interpolation methods under scaling factor=2, set14.



(a) Nearest-Neighbor

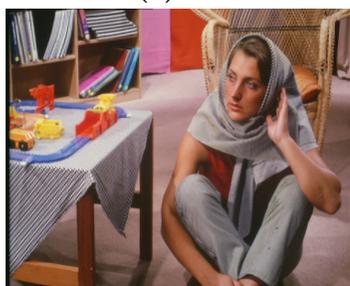
(b) Bicubic



(c) SC



(d) SRCNN

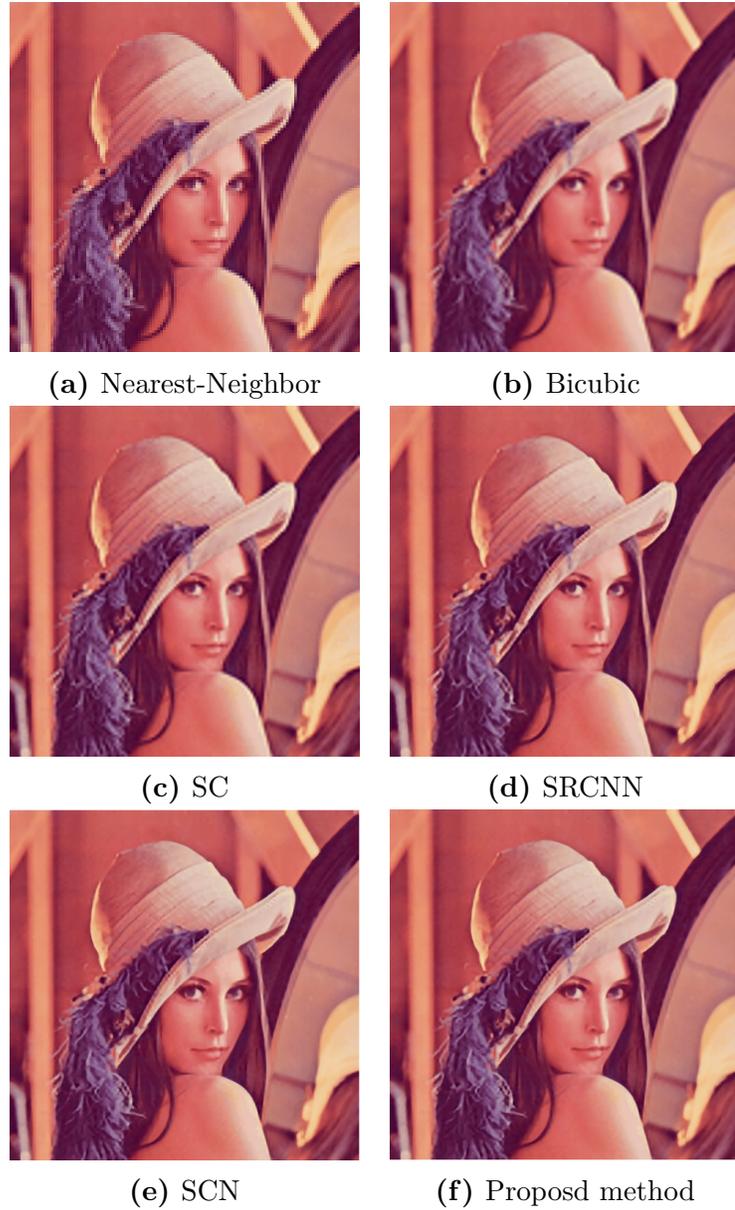


(e) SCN



(f) Proposd method

**Figure 5.5.** Visual comaparison between the proposd method and State-of-the-Art models and interpolation methods under scaling factor=3, set14.



**Figure 5.6.** Visual comparison between the proposed method and State-of-the-Art models and interpolation methods under scaling factor=4, set14.

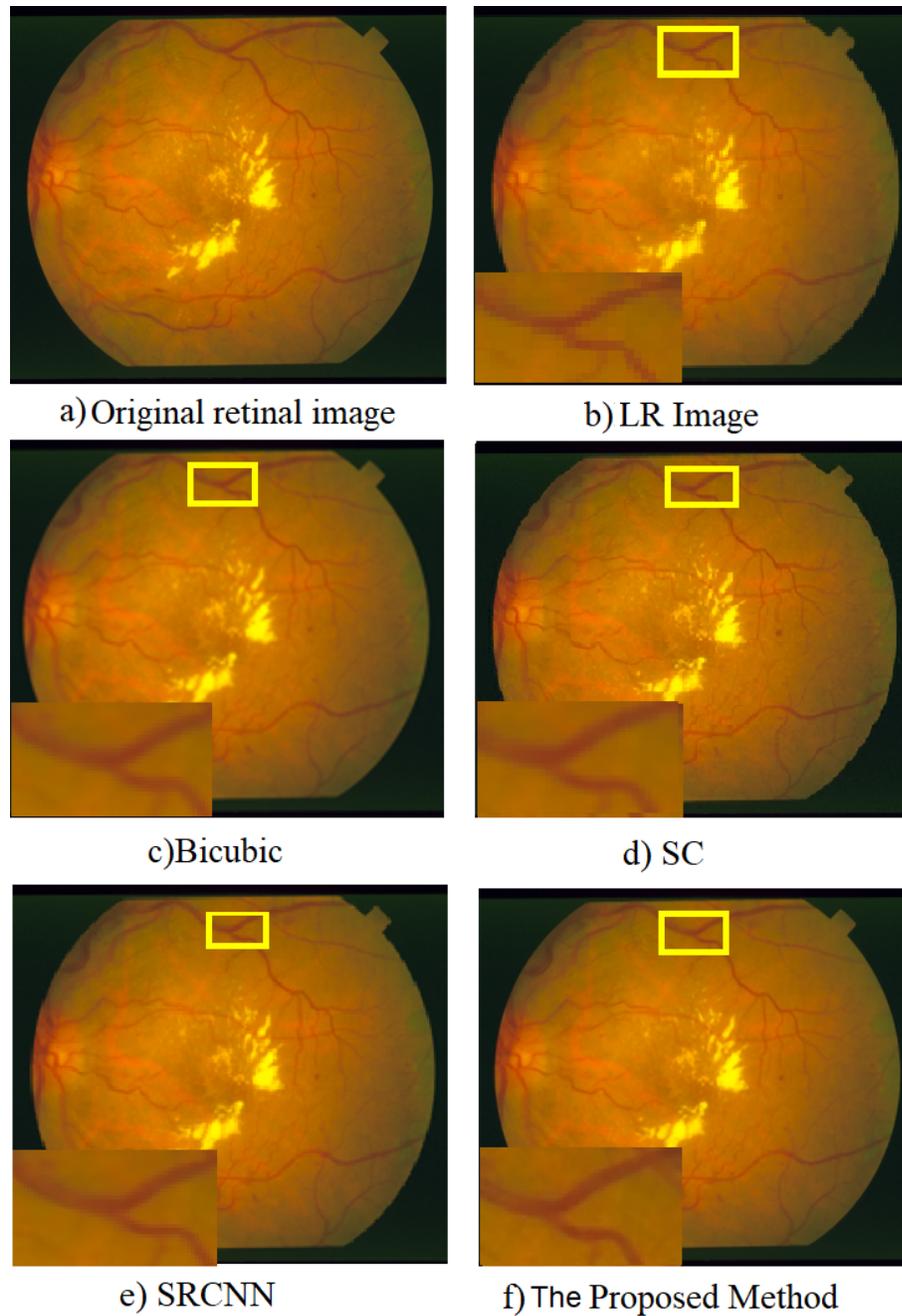
## 5.2 Experiment-2: Retinal Image Super Resolution

In the second experiment, we focus on retinal images. Retinal imaging is an essential branch of medical imaging. There is a various study in the literature and ongoing regarding retinal image super-resolution. Therefore, we applied our method to a single retinal image which is taken from Structured Analysis of the Retina project database (STARE) [14]. Obtained results have shown that our proposed network has a remarkable improvement in retinal images visually and quantitatively. As it is seen in the results, fusing sparse coding with the deep network can be beneficial for retinal image SR. Results have shown that proposed method has gained significant improvement

**Table 5.7.** Image quality matrix for HR images, scaling factor=3

Method	PSNR	SSIM
Bicubic	43.51	0.9765
SC	44.05	0.9896
SRCNN	46.23	0.9825
SCN	47.02	0.9899
The Proposed Method	<b>47.50</b>	<b>0.9964</b>

in SISR for the retinal image. To illustrate the results, we show output SR images in the Figure 5.7



**Figure 5.7.** Super Resolution results for a retinal image. *a* is an original image from the retinal image database. *b* is down sampled version of LR image. *c*, *d*, and *e* are implementation results of Bicubic, Sparse Coding [1] and SRCNN [2] respectively. *e* is the result of our proposed method.

### 5.3 Experiment-3: Brain MRI Image Super Resolution

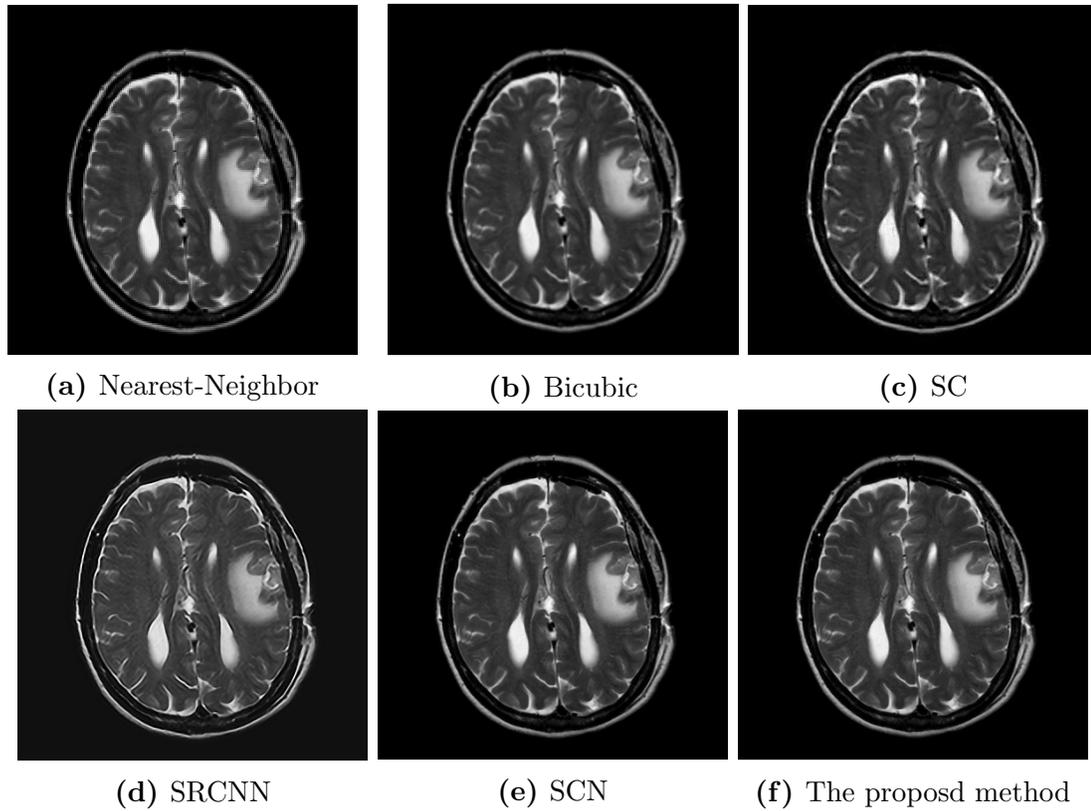
Brain Mri images are very significant to examine brain surface to get a better diagnosis. Also, capturing an MRI image is very expensive. Therefore, single image super-resolution fits very well to enhance their resolutions. In this section, we apply our super-resolution image method on a brain MRI to obtain more clear texture and clear image. Later on, the results are compared to other methods to measure the performance. Testing images are taken from DICOM medical image library [40]. The visual results of the simulation is shown in the Figure 5.8.

**Table 5.8.** Image quality matrix for HR images, scaling factor=2

Method	PSNR	SSIM
Nearest	31.15	0.8611
Bicubic	35.00	0.8989
SC	36.01	0.9056
SRCNN	37.12	0.9101
SCN	38.57	<b>0.9275</b>
The Proposed Method	<b>38.66</b>	0.9273

### 5.4 Summary

In this chapter, we presented simulation results of the proposed method. Then we compared these results with other methods and interpolation techniques. We provided visual comparisons and image quality evaluation. Also, we defined what kind of parameters were used during the experiments. The results have shown that the proposed method can reach outstanding performance regarding both SSIM/PSNR and visual comparison.



**Figure 5.8.** Visual comparison of MRI images between the proposed method and State-of-the-Art models and interpolation methods under scaling factor=2.

## Chapter 6

### CONCLUSION

Image super-resolution has been one of the most studied topics in image processing area. Specially single-image-super-resolution is the most focused branch of super image resolution. Obtaining a high-resolution image by using only one single input image is an outstanding and efficient idea for many reasons such as disability of taking multiple images or affordability. For example, MRI is an expensive imaging technique [41]. Taking it multiple times costs much. Therefore, the concept of single image super-resolution is a very efficient way to an overcome this issues.

In this thesis, we propose a single image super-resolution network based on three existing state-of-the-art methods SC [1], SRCNN [2] and SCN [3]. In the model, we also applied the different pre-processing procedure to get a better PSNR/SSIM performance. The experimental results show that the proposed method shows outstanding performance regarding image quantitative and visual comparison. Thus, the proposed method generates clear and better-detailed output HR images.

## BIBLIOGRAPHY

- [1] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [2] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *European Conference on Computer Vision*, pp. 184–199, Springer, 2014.
- [3] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 370–378, 2015.
- [4] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 399–406, 2010.
- [5] K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “Fast inference in sparse coding algorithms with applications to object recognition,” *arXiv preprint arXiv:1010.3467*, 2010.
- [6] J. Hu, *A Joint Dictionary-Based Single-Image Super-Resolution Model*. PhD thesis, Université d’Ottawa/University of Ottawa, 2016.
- [7] G. Cristóbal, E. Gil, F. Šroubek, J. Flusser, C. Miravet, and F. d. B. Rodríguez, “Superresolution imaging: a survey of current techniques,” 2008.
- [8] K. Hayat, “Super-resolution via deep learning,” *arXiv preprint arXiv:1706.09077*, 2017.
- [9] K. Nasrollahi and T. B. Moeslund, “Super-resolution: a comprehensive survey,” *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014.
- [10] Q. Dai, E. Pouyet, O. Cossairt, M. Walton, and A. K. Katsaggelos, “Spatial-spectral representation for x-ray fluorescence image super-resolution,” *IEEE Transactions on Computational Imaging*, 2017.
- [11] J. P. Haldar, D. Hernando, and Z.-P. Liang, “Super-resolution reconstruction of mr image sequences with contrast modeling,” in *Biomedical Imaging: From Nano to Macro, 2009. ISBI’09. IEEE International Symposium on*, pp. 266–269, IEEE, 2009.

- [12] I. J. Ahn, J. H. Kim, Y. Chang, W. H. Nam, and J. B. Ra, “Super-resolution reconstruction of 3d pet images using two respiratory-phase low-dose ct images,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 1, no. 1, pp. 46–55, 2017.
- [13] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, R. Guerrero, S. Cook, A. de Marvao, D. O’Regan, *et al.*, “Anatomically constrained neural networks (acnn): Application to cardiac image enhancement and segmentation,” *arXiv preprint arXiv:1705.08302*, 2017.
- [14] STARE, “Structured analysis of the retina project,” 2013, <http://www.cecas.clemson.edu/~ahoover/stare>.
- [15] M. Bevilacqua, *Algorithms for super-resolution of images and videos based on learning methods*. PhD thesis, Université Rennes 1, 2014.
- [16] H. He and W.-C. Siu, “Single image super-resolution using gaussian process regression,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 449–456, IEEE, 2011.
- [17] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with non-local means and steering kernel regression,” *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4544–4556, 2012.
- [18] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
- [19] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–I, IEEE, 2004.
- [20] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–I, IEEE, 2004.
- [21] S. Fadnavis, “Image interpolation techniques in digital image processing: An overview,” *International Journal of Engineering Research and Applications*, vol. 4, no. 10, pp. 70–73, 2014.
- [22] T. M. Lehmann, C. Gonner, and K. Spitzer, “Survey: Interpolation methods in medical image processing,” *IEEE transactions on medical imaging*, vol. 18, no. 11, pp. 1049–1075, 1999.

- [23] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.
- [24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [25] M. Aharon, M. Elad, and A. Bruckstein, “*rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [26] F. Farhadifard, *Single image super resolution based on sparse representation via structurally directional dictionaries*. PhD thesis, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ), 2013.
- [27] R. Timofte, V. De Smet, and L. Van Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1920–1927, 2013.
- [28] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *Asian Conference on Computer Vision*, pp. 111–126, Springer, 2014.
- [29] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.
- [30] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 349–356, IEEE, 2009.
- [31] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1646–1654, 2016.
- [32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [33] D. Liu, Z. Wang, N. Nasrabadi, and T. Huang, “Learning a mixture of deep networks for single image super-resolution,” in *Asian Conference on Computer Vision*, pp. 145–156, Springer, 2016.
- [34] H. Gao, B. Maiseli, and E. A. Ogada, “A multi-frame super-resolution method based on the variable-exponent nonlinear diffusion regularizer,” 2015.

- [35] N. K. Bose and N. A. Ahuja, "Superresolution and noise filtering using moving least squares," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2239–2248, 2006.
- [36] N. K. Bose and N. A. Ahuja, "Superresolution and noise filtering using moving least squares," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2239–2248, 2006.
- [37] J. Tian and K.-K. Ma, "Stochastic super-resolution image reconstruction," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 232–244, 2010.
- [38] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.
- [39] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, pp. 711–730, Springer, 2010.
- [40] DICOM, "Dicom medical image library," 2018, <https://www.osirix-viewer.com/resources/dicom-image-library>.
- [41] L. Zhou, Z. Jiang, N. Geng, and X. Wang, "Capacity allocation for balancing cost and fairness in a imaging facility setting," in *Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on*, pp. 721–725, IEEE, 2015.