TAMPERE UNIVERSITY OF TECHNOLOGY

VIDA FAKOUR SEVOM

LEARNING-BASED SINGLE IMAGE SUPER RESOLUTION

Master's thesis

Examiners: Prof. Karen Eguiazarian
Examiners and topic approved by the Faculty Council of the Faculty of Natural Sciences
on 14 January 2015.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Science and Bioengineering
FAKOUR SEVOM, VIDA: LEARNING-BASED SINGLE IMAGE SUPER RESOLUTION
Master of Science Thesis, 58 pages
January 2015
Major: Bioimaging
Examiners: Prof. Karen Eguiazarian
Keywords: Super resolution, Single-image super resolution

Recent advancements in signal processing techniques have led to obtain more high resolution images. A high resolution image refers to an image with high density of pixels. The importance and desire of high resolution images are obvious in the field of electronic and digital imaging applications.

The quality of an image can be improved either by hardware or software approaches. Hardware approaches are straightforward solutions to enhance the quality of a given image, but some constraints, such as chip size increment, making them expensive to some extend. Therefore, most of the researchers are focused on software methods.

Super resolution is one of the software image processing approaches where a high resolution image can be recovered from low resolution one(s). The main goal of super resolution is the resolution enhancement. This topic has been widely brought into attention in image processing society due to the current and future application demands especially in the field of medical applications.

Super resolving a high resolution image can be performed from either a single low resolution or many low resolution images. This thesis is completely concentrated on Single Image Super Resolution (SISR) where a single low resolution image is the candidate to be exploited as the input image. There are several classes of methods to obtain SISR where three important ones, i.e., the Example-based, Regression-based and Self-similarity-based are investigated within this thesis.

This thesis evaluates the performance of the above-mentioned methods. Based on achieved results, the Regression method shows better performance compared to other approaches. Furthermore, we utilize parameters, such as patch size, to improve the numerical and virtual results in term of PSNR and resolution, respectively. These modifications are applied to the Regression-based and Self-similarity-based methods. The modified algorithms in both methods lead to improve results and obtain the best ones.

PREFACE

This thesis is based on research in super resolution field for developing a method to reconstruct a high resolution image. The research was completed in Signal Processing Department of Tampere University Of Technology (TUT). I am sincerely grateful to God for all I have been blessed with.

I would like to express my heartfelt gratitude to Prof. Kare Eguiazarian; he is a great man not only as a supervisor, but also as a friend. He never let me get disappointed through all ups and downs. Working with him for the past year has been an unutterable learning experience. This work would not be completed without him and his invaluable guidance.

My thanks are extended to my friends Orod Raeesi, Saeed Afrasiabi, Mona Aghababaee, Nader Daneshfar and Kamiar Rad Nosrati for all their sincere kindness. They are valuable friends that made my experience at TUT unforgettable. My special thanks must go to Orod and Mona for reviewing the manuscript of this thesis and also to Saeed for his immeasurable helps. I would like to express my appreciation to Farid Shamani for his endless patience. I can not finish my appreciation to him in a sentence. The least I can do is to dedicate my work to him. I will forever be thankful for everything he has done for me.

I also want to thank my father Mohammad Hassan Fakour and my mother Zahra Varasteh. I would never be in this stage of my life without their supports. Last- but by no means least- I must thank my lovely sister Homa and my brother Farhad. I will never forget the memories I have had with them for 23 years we lived together.

Tampere, 21 January 2015
VIDA FAKOUR SEVOM

CONTENTS

LIST OF FIGURES

TERMS AND DEFINITIONS OR LIST OF SYMBOLS AND ABBREVIATIONS

SR Super Resolution

RE Resolution Enhancement

HR High Resolution

LR Low Resolution

HF High Frequency

LF Low Frequency

CCD Charge-Couple Device

CMOS Complementary Metal-Oxide-Semiconductor

PSF Point Spread Function

SISR Single Image Super Resolution

MRI Magnetic Resonance Imaging

CT Computed Tomography

NN Nearest Neighbor

MRF Markov Random Field

ScSR Sparse Coding Super Resolution

NE Neighbor Embedding

LLE Locally Linear Embedding

HVS Human Visual System

PSNR Peak Signal-to-Noise Ratio

RMSE Root Mean Squared Error

SNR Signal to Noise Ratio

1. INTRODUCTION

1.1 Super resolution of images and videos

The problem of spatial resolution enhancement has been an active research area since early 1980s when the first paper regarding Super Resolution (SR) or Resolution Enhancement (RE) was proposed by Tsai and Huang [1].

The concept of the *resolution* in image processing is related to the amount of information embedded in an image. Resolution can be described in various ways such as *Pixel Resolution*, *Spectral Resolution*, *Spatial Resolution*, *Temporal Resolution* and *Radiometric Resolution*. In this thesis, spatial resolution will be of the main interest [3]. Henceforth, the term **"resolution"** will refer to a spatial resolution.

The term **Pixel** was invented from **"Picture Element"** word. Pixels are smallest addressable points used to create an image. Each image is a matrix consisting of thousands of pixels. Since these square-like pixels are so small, they are not distinguishable when looking at an image [35]. Spatial resolution is related to the number of pixels contributing to construct an image. Increasing the number of pixels in an image causes to achieve higher spatial resolution as well as images with improved sharpnesses [3]. In order to see pixels of an image, zooming is a an appropriate tool. In Figure 1.1, pixels of a small region have been visualized by extreme zooming.

The importance and desire of High Resolution (HR) images are obvious in the field of electronic or digital imaging applications. High resolution image refers to an image with high density in terms of pixels. In principle, an HR image contains more information, in comparison with a low resolution image, which is a key point in most of the applications. One of the crucial needs for HR images is in medical applications where the accuracy of the diagnosing is proportional to the quality of the image. It can be seen by the fact that having more resolution means having more information which leads to more accuracy from medical application point of view [2].

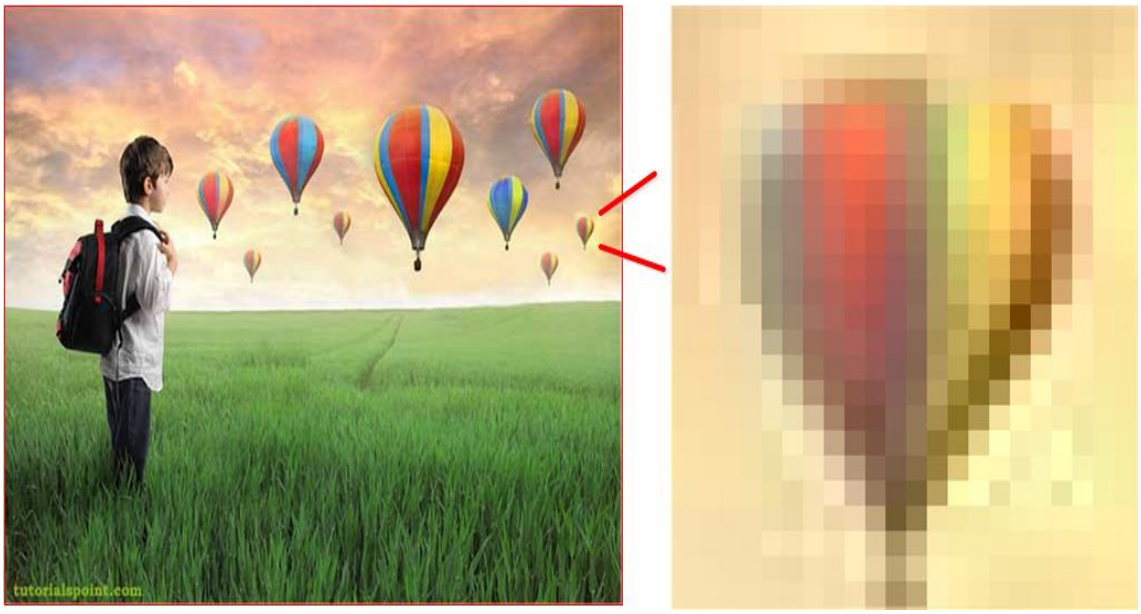


Figure 1.1: Image and its zoomed version for a certain region

Figure 1.2 shows the difference between high and low resolution images. According to the picture, it is obvious that Figure 1.2(a) has more sharpness compared to Figure 1.2(b) which is blurred due to the lack of adequate resolution. It is worthwhile to mention that sharpnesses and edges are shown by High Frequency (HF) components while LF components create a blurry image where the main structure of the image can just be distinguished. Moreover, HF components present the pixels whose values are changing rapidly while LF components values are changing slowly [4].

(a) (b)

Figure 1.2: a) High Resolution Image, b) Low Resolution Image

1.1.1 Hardware perspective of image resolution enhancement

Considering image resolution, the first limitation is appeared in the hardware part, i.e., image sensors. The two common image sensors are Charge-Couple Device (CCD) and Complementary Metal-Oxide-Semiconductor (CMOS) active-pixel. These sensors are used to take digital images and they are appropriate for most imaging fields. The significant point is that they are not desirable for a current need of resolution due to some hardware limitations [3].

As it was mentioned earlier, spatial resolution is proportional to the number of pixels. As a result, the straightforward solution in order to increase the spatial resolution is increasing the number of pixels (increasing the number of sensor elements). The size of a pixel is reciprocal to the amount of the shot noise. In other words, as long as the pixel size is decreasing, the amount of the shot noise is increasing. The shot noise is created due to the decrement in available light along with fluctuations seen in the number of detected photons [5]. Moreover, the hardware cost is increased by utilizing more sensors. Therefore, there is always a trade-off between the pixel size and the amount of noise. It is worth mentioning that the lower limited pixel size is approximately 40 μm^2 for a $0.35\mu m$ CMOS process [2]. Increasing the chip size in order to improve the resolution is another procedure. This approach causes increasing the total capacitance. Since large capacitance leads to have low charge transfer rate, this approach also can not be successful in resolution enhancement [2]. Hence, it is demanding to find alternative approaches in order to increase the level of resolution.

1.1.2 Software perspective of image resolution enhancement

According to the problems caused by the hardware approach, a wise idea is to emphasize more on software methods rather than the hardware ones. SR is a signal processing technique in which an HR image is produced from one or multiple LR images. In other words, SR techniques intend to improve the resolution and details of an image by estimating the edges and textures which are the fundamental components of an image. Recently, the SR has become more popular in different fields of research which are mostly focused on image resolution enhancement. There are several methods and algorithms to achieve SR which can be found in particular, in [6]-[12].

There are several advantages in exploiting SR. One of the decisive advantages is the cost of the development. These techniques are more efficient than the hardware ones. Another prominent feature enabled by SR in signal processing field is that the

low resolution imaging systems can still be used [2].

Super resolution algorithms are divided into two main categories: **Multi-frame** super resolution [13],[14] and **Single-Image** Super Resolution (SISR) [15]-[18]. In this thesis, some of the algorithms targeting SISR are studied and compared. Figure 1.3 illustrates some categories of super resolution methods described in following sections.

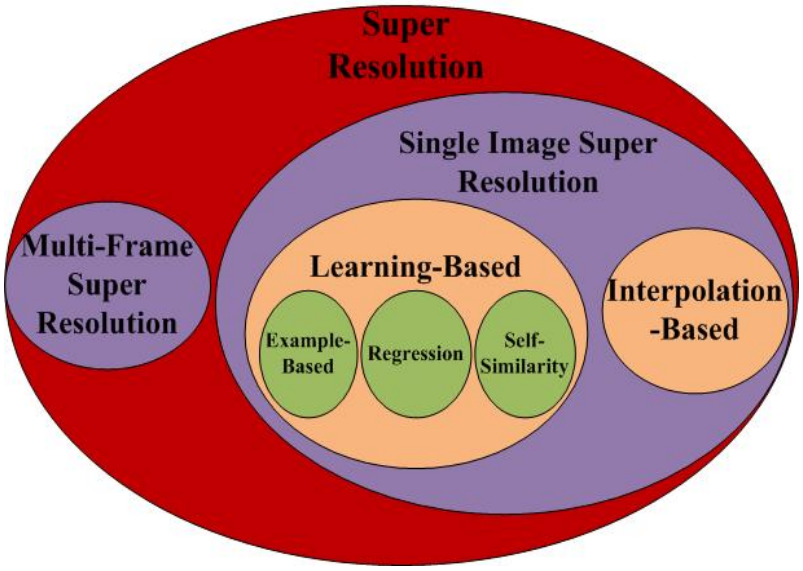


Figure 1.3: Scheme of the Super resolution techniques in details

1.2 Multi-frame super resolution

The traditional methods in SR are reconstruction-based, where multiple LR input images are used to super resolve an HR image. They utilize information existing between multiple LR images which are obtained either by capturing several pictures from the same scene, or using multiple cameras placed in several locations. The images have the same content, however a non-redundant information exists between images introduced as *subpixel shift*. Subpixel displacements occur when there is a movement in the objects, shifting in scenes or shifting between imaging systems. The extra (non-redundant) information helps in recovering the HR image [3]. In contrast, there is no extra information when a shifting occurs not within the subpixel displacements. In this situation, each image has the same information. Figure 1.4 depicts the basic idea of reconstruction-based methods in super resolution. As it can be seen, the green circles represent the integer shifts, the red squares and blue triangles represent the subpixel shifts used in SR reconstruction. It also shows two different LR image acquisition mechanisms, i.e., either only with one camera or multiple cameras [2].

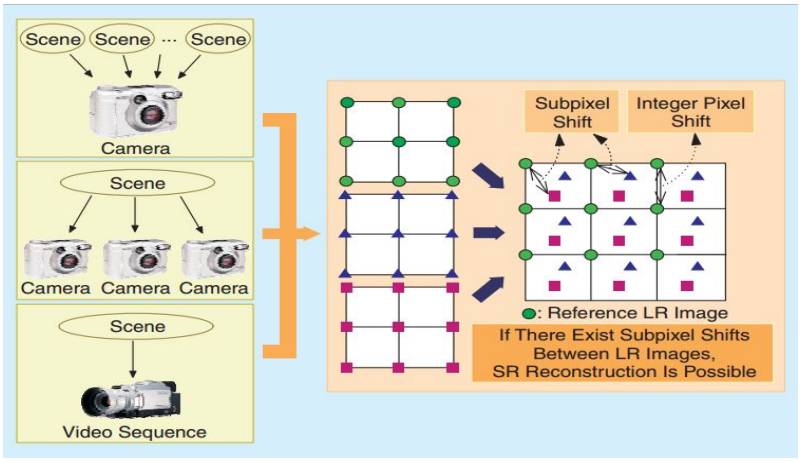


Figure 1.4: Schematic representation of a HR image reconstruction from multiple LR images [3]

In the process of reconstructing an HR image, some natural noise and artifacts are introduced to the image. Optical distortions, motion blur, sensor noise, etc are introduced to the image as degradation factors, as well. These types of impairments cause the image to loose the spatial resolution. Therefore, SR methods in addition to reconstructing an HR image from LR ones, should cover image restoration technique simultaneously. Image restoration technique assists to mitigate the mentioned impairments as well as producing a high quality image without noise or blurring effects. Image restoration does not change the image size [3].

The first step in SR reconstruction methods is to form a relationship between HR and LR images called the observation model. There are several observation models proposed in the literatur [3]. Figure 1.5 represents an example of an observation model between HR and LR images.

Two main parameters used in reconstruction-based SR techniques are magnification factor (size of the desired enlarged image) and a number of observed LR images. The performance of reconstruction-based methods are degraded if the magnification factor is chosen to be large [23]. Hence, reconstruction-based methods are not applicable for any enlargement factor. In some cases, the number of LR images is not sufficient enough to achieve efficient results for super resolving problem. In addition, information related to the sensors, Point Spread Function (PSF), LR images alignment, etc. are needed in multi-frame super resolution methods. Therefore, these parameters can cause some constraints in some cases. In the next section, we will consider the SR applications when there is only one input image.

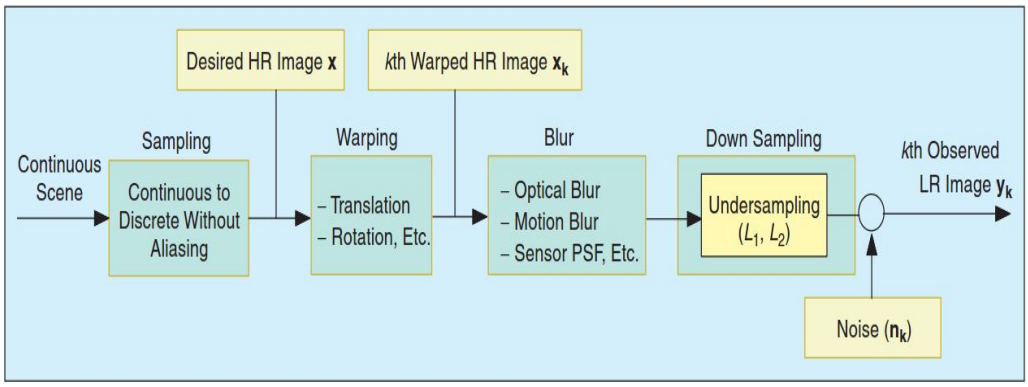


Figure 1.5: Observation model, real imaging system [3]. First desired HR image is achieved by sampling continues signal beyond the Nyquist rate. After translation and rotation due to some optical or motion events, they got blurred. Observed LR images are achieved by downsampling (sampling so that the size of the image is reduced) the blurred images from last step[2]

1.3 Single-Image Super Resolution

In SISR, the super resolving algorithm is applied to only one input image. Since in most cases there is no underlying ground truth, the significant issue is to create an acceptable image. Hence, possessing a reasonable prior information is more interesting rather than estimating the characteristics of the sensors [19]. SISR approaches can be classified in two main categories: *interpolation-based [20],[21]*, and *learning-based [6],[12]*.

1.3.1 Interpolation methods

According to [28], the goal of interpolation-based methods is to achieve a high resolution image by employing a combination of methods such as upsampling, denoising, etc. Interpolation-based methods treat SR as a non-uniform interpolation problem [25]. The assumption in interpolation algorithm is that the input LR image is the downsampled version of an HR image. Based on this assumption, it tries to recover the HR image by upsampling LR image while considering the impact of dealiasing [28]. In [28], the interpolation based methods can be divided in two groups each of which splits into couple of categories. Interpolation algorithms are listed as following: *polynomial-based interpolation*, and *edge-directed interpolation*.

Single image interpolation can predict new pixels by e.g. bilinear or bicubic interpolation. However, results of these interpolation methods are not satisfactory since they are not able to overcome the strong discontinuity contents of the images. Discontinuities are nominated to natural specification of each typical image. Having various type of edges in a typical image is a good example of such discontinuities.

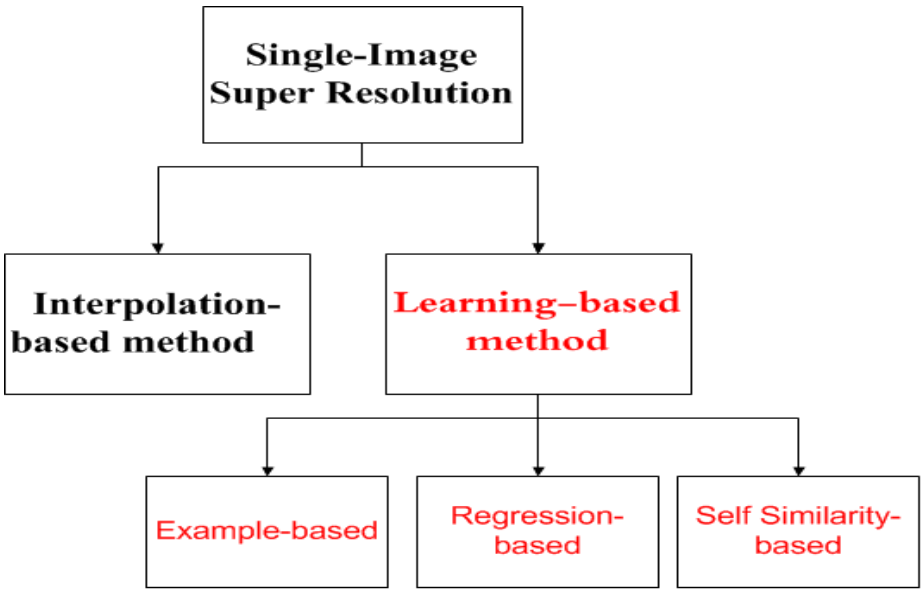


Figure 1.6: Scheme of the Thesis

Although the interpolation based methods are simple, they produce variety of impairments. The most prominent one is blurring the higher frequency components resulting in smoothing of the entire image [18].

1.3.2 Learning-based method

Learning-based methods are used to recover the HR image by referring to missing HF components of the LR ones. In addition, they arrange pertinent information for the target image in advance. In this Thesis, learning-based methods are organized in three categories: *example-based [6],[12]*, *self-similarity-based [26], [22]*, and *regression-based [15],[8]* .

The main focus of this Thesis is based on learning-based methods of single image super resolution. In following chapters, they are discussed in more details. The advantages and disadvantages of each method are investigated as well.

As it can be seen from Figure 1.6, this work is going through three different SISR classes. In this figure, the red blocks show the methods which are going to be discussed through this work.

1.4 Applications of Super Resolution

There are many applications where SR is applied such as satellite imaging, medical imaging, video usage, high definition television, high performance color LCD, remote sensing, etc. In all these applications, image reconstruction has a vital role [2]. The

image quality in some medical imaging systems such as Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) is not quite high. However, they still have the potential to be employed in the field of SR due to the ability of acquiring multiple images.

Another example is a surveillance application which is available for both market and home in order to monitor the environment. In such systems, the motion events are recorded as long as there is no abnormality. Once the abnormality happens, an alert is sent to the system controller. The video is also possible to be shown to an authorized person in any location. In this situation, video streaming occupies most of the bandwidth. Therefore, the band limitation issue is emerged. In order to resolve this problem, data should be sent at the lowest possible rate. Hence, both temporal and spatial SR would be required to achieve a reasonable video from the received LR one [17].

1.5 Outline

The rest of the Thesis is organized as follows.

Chapter one presents the example-based methods. This chapter covers the basics of example-based SISR. An overview to the Freeman's method and a Neighbor Embedding method will be given, as well.

Chapter two discusses the regression-based method. In this chapter, we are going through the paper [15] and implementation of the proposed method. Afterwards, we will present the modified algorithm.

Chapter three presents the self-similarity method which has been widely used recently. In this chapter, the main idea of the paper [26], will be investigated. The results will be presented after applying few modifications to the original code. Finally, a comparison between modified code and the original one will be given.

Chapter four concludes the entire thesis focusing on different SR methods. This chapter also discusses the results and the performance of each of the implemented methods in various situations and environments.

2. EXAMPLE-BASED METHODS

This chapter is mainly focused on *Example-based* methods from the group of learning-based algorithms in super-resolution. We present the basics of learning-based approach by implementing some example-based methods and show visual and numerical results of simulations.

2.1 Introduction

In early super resolution algorithms, e.g. reconstruction-based, the main idea was to aggregate multiple LR images which have the complementary information in the spatial domain. This cannot always be practical due to some constraints appearing during the measurements. In some cases, a problem appears, for instance, when just one low resolution image is available. Learning-Based methods utilize external examples to conquer the limitations of measurements in super resolving an image.

Example-based SR methods are based on sampling other images in training set [3]. In other words, the super resolved image is achieved by splitting training samples in two different sets of patches (small blocks or windows of (3×3) pixels). One set is related to LR components and the other one to HR components. The training set consists of huge number of LR and HR patches. The modeled relationship between HR and LR patches helps each new given LR patch to find its corresponding HR one according to the learning process in the training stage.

One of the earliest paper regarding the **Example-based** image super resolution, published by Freeman et al. [6], has initialed an intensive research in the learning-based field of single-image super resolution. Although Freeman's method is bounded by some constraints, e.g. over-fitting, dependency on external data in some cases and etc, it can be considered as an acceptable SR method. The main idea behind it is to recover the missing HF components which can not be achieved by simple sharpening.

2.2 Overview of example-based methods

Some of example-based algorithms (mostly previous ones), e.g. [6], [29], [13], [30] are categorized in Nearest Neighbor (NN)-based estimation approach [19]. This

classification is due to the similarity of the algorithms based on NN estimation and example-based methods. Here, LR patches and their corresponding HR ones are collected. The nearest patch to the input LR patch among the training set is found (checking neighbor compatibility) [19]. Super resolving is performed by finding the corresponding HR patch to the candidate LR patch.

In [30], the SR method has been performed according to analogies which can be found between images. Baker et al. in [13] have used Laplacian pyramid to represent the images and estimated the HR image by NN-based estimation. In [29], initial interpolation of the given image is performed for any desired scaling factor. Afterwards, NN-based estimation of missing HF components is applied in order to super resolve the LR image. In this implementation, *Markov Random Chain (MRC)* has been utilized to resolve the compatibility of the output. Chang et al. in [12] have made their efforts on eliminating some constraints which can be caused by previous NN-based methods. In [24], Sparse coding Super Resolution (ScSR) has been introduced as a method which can select the best patches adaptively.

In this section some of the mentioned example-based methods such as Freeman's example-based SR algorithm [6], Neighbor Embedding (NE) method [12] and the ScSR method [24] will be discussed.

2.2.1 Freeman's Method

The main goal of the Freeman's algorithm is to estimate the high-resolution components which are missing in the original image. The entire algorithm can be categorized in two main sections. First section is related to the training stage where the example patches are obtained from an external comprehensive database. The database learns correlations between low and high resolution image patches which form pairs. Example-based method is working in such a way that two sets of data, which are employed as training patches to construct the new HR image, are produced. Assume set $\{x_i\}_{i=1}^n$ is sampled from HR images and the second one, $\{y_i\}_{i=1}^n$, is sampled from the LR images. The observation model, which connects the HR and LR image patch pairs, (x_i, y_i), is $y_i = DFHx_i + V$. In this model, D is a downsampling operator, F is a motion information, H is a blurring model and V is a noise [3]. In [16], the HR patch is predicted by applying the above mentioned observation model to the test image. In the second section, the super resolving algorithm is completed by utilizing a graphical framework (MRC or One-Pass algorithm) which helps to preserve the neighbor compatibility.

(a) (b)

(c)

Figure 2.1: a) Low-resolution downsampled image b) Initial Interpolated image c) Original image

Training procedure

In the training stage, fine details for each region in LR image should be learned first. Then, the learned algorithm is used to predict fine details for any given image.

Obtaining LR images in order to create the training set must be done before starting training procedure. This can be achieved when the HR images are subsampled (subsamplings used to reduce the image resolution). Afterwards, an initial interpolation algorithm, e.g. the cubic spline, is applied to the obtained subsampled images. The interpolation algorithm tends to produce an image with the same size as the original HR one. The only difference is that the generated image, which is called smoothed version of the original image, suffers from lack of details. Figure 2.1 depicts the mentioned images as well as the corresponding low resolution and interpolated versions. In this figure, picture (c) is the original image whereas pictures (a) and (b) are the downsampled (the image achieved by sampling and reducing the number of pixels) and smoothed versions of the original image, respectively.

It is expected that each HR image patch and its corresponding LR patch are stored in two different sets of patches to perform the training procedure. Storing this amount of information causes some problems in terms of memory capability or algorithm processing speed. In order to avoid such problems, a preprocessing step must be executed beforehand. Preprocessing stage removes unnecessary variability within the image and makes the training set more appropriate. The preprocessing application includes filtering of redundant data which are not essential through the whole super resolution algorithm. An example of redundant data can be introduced as the low frequency components of the blurred (interpolated) image. The LF components of the interpolated image are not needed for predicting the missing HF components [6].

Super resolution procedure

In this method, the input LR image is upsampled (predicting new pixels in order to increase the size of an image) using bicubic interpolation. Thus, the upsampled image, assume I_l^h, is enlarged with the desired scaling factor. Certainly, this image is blurred compared to the original version of itself. Afterwards, I_l^h is passed through a band-pass filter which filters the LF components while preserving the HF components in the image. Therefore, the number of training examples is reduced due to the elimination of low frequency components as well as normalization of contrast.

The super resolution algorithm is not a simple procedure. One of the major issues in this method is that the LR image, i.e. I_l^h, is divided into several small patches. This means that, the observation model which is made between LR and HR patches is local (i.e. the model is between patches not images). Hence, it is possible that the chosen HR patch would not be compatible with its spatial neighbors. Consequently, the algorithm should not only search through the LR database to find the closest patch to the input image patch. Indeed, in order to have an accurate prediction the surrounding patches in HR database are needed to be taken into consideration. For example, the training set, in [6], includes e.g. 100,000 patches. For each given patch, 16 closest patches are selected. Although these patches are quite similar to the input ones, their corresponding HR patches are completely different from each other (Figure 2.2). Hence, finding the closest LR patch (to the input LR patch) along with its corresponding HR ones are not the only candidates to resolve an HR patch.

Markov Random Chain is one of the solutions for the output compatibility problem. As Figure 2.3 shows, there are some blocks connected to each other where the connections model the similarities between LR patches and their corresponding HR

Figure 2.2: Representation of input patch. Second row shows 16 closest patches in LR training dataset. The last row depicts the corresponding HR patches of 16 closest patches [6]

ones. The compatibility between spatially neighboring HR patches are shown as well. Figure 2.3 presents the MRC model, where the circles represent network nodes and each line represents the dependency of the corresponding node. In Markov network, the probability of selecting the appropriate HR patch is given by:

$$P(x|y) = \frac{1}{Z} \prod_{(ij)} \psi_{ij}(x_i, x_j) \prod_{(i)} \phi_i(x_i, x_j) \tag{2.1}$$

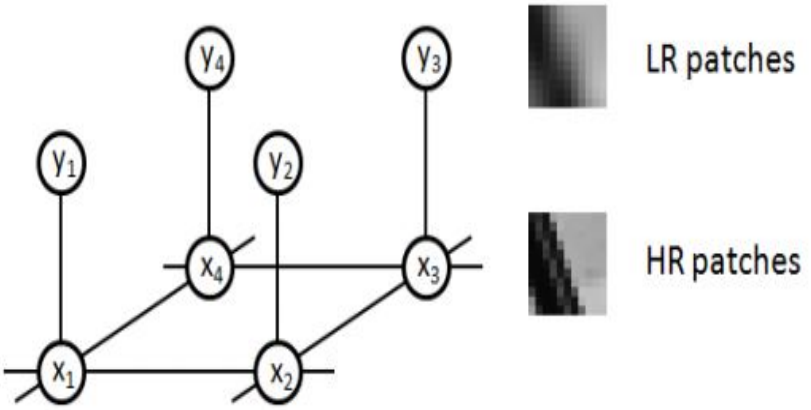
where ψ is the compatibility matrix which expresses the possible states of each pair, ϕ is a vector which makes the relation between each observation and underlying hidden states, Z is the normalization constant, y_i is the representation of the observed LR input patch and x_i is the found candidate LR patch in the training set.

If i and j present two nodes, the compatibility matrix is:

$$\psi_{ij}(x_i, x_j) = exp(-\frac{d_{ij}(x_i, x_j)}{2\sigma^2}) \tag{2.2}$$

where σ is a parameter of the noise. In order to estimate ϕ, the following equation should be applied:

$$\phi_{ij}(x_i, x_j) = exp(\frac{|x_i - x_j|^2}{2\sigma^2}) \tag{2.3}$$



(a)

Figure 2.3: MRF framework structure [3]

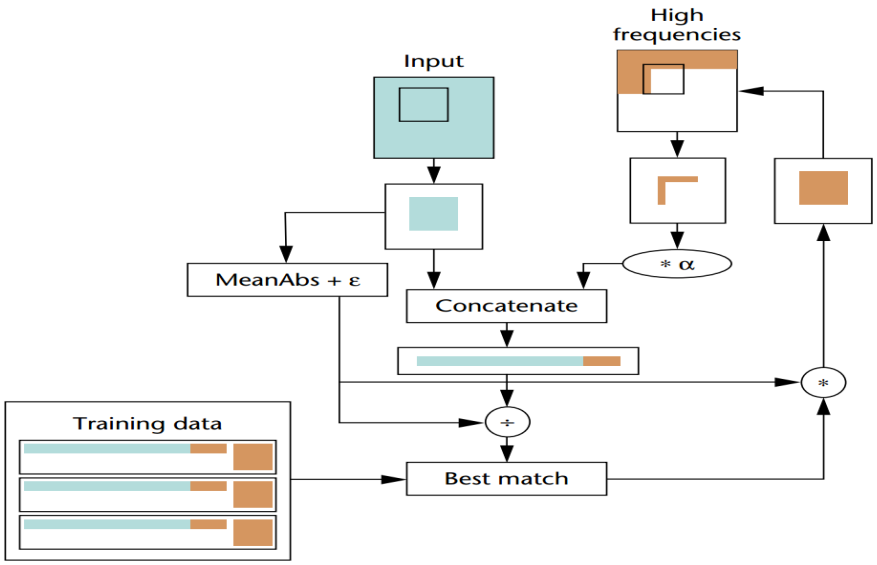
In Freeman's method instead of applying MRC algorithm, **One-Pass** has been utilized. Although the one-pass algorithm is similar to Markov network, it is faster in implementation because of considering only HR patch compatibilities. Figure 2.4 shows the one-pass algorithm.

The example-based algorithm is expressed briefly as following:

Data: Input LR image I_l, training set (including e.g. 100,000 LR and HR patches), interpolated image I_l^h
Result: High resolution super resolved image I_h
for *each patch of* I_l **do**
  searching in LR patches and find e.g. the 16 closest patches to it;
  Finding their corresponding HR patches;
  Applying Markov network to consider the neighboring information and find the best LR patch, i.e $x^{'}$;
  Select its corresponding HR patch, $y^{'}$, add it with the patch in the interpolated image and insert it to the right place in the output image
end

Algorithm 1: Example-based super resolution algorithm

As it was previously mentioned, the Freeman's method is one of the first approaches toward the example-based super resolution algorithms. Later works with the concept of example-based methods have made their efforts on eliminating some constraints introduced by Freeman's algorithm. The most significant issue in example-based method is the requirement of a huge training data (high dimensional data) set to cover all possible patterns.



(a)

Figure 2.4: One-pass algorithm [6]

2.2.2 Neighbor Embedding

The idea of NN-based methods has been used widely in super resolution algorithms. It is worth mentioning that NN-based methods have impressively attracted the most attention in super resolution field. However, there is still a need to achieve more improvements in the algorithms. For instance, over-fitting problem (which appears due to using high dimensional data) should be resolved [19]. The over-fitting usually happen during the learning process of NN-based methods. Consequently, having a robust method, such as **Neighbor Embedding (NE)**, to resolve the above-mentioned constraint in NN-based methods is crucial.

NE is an algorithm proposed by Chang et al. in [12]. The term **"neighbor"** refers to the nearest LR patches to the input LR patch. The algorithm was focused on making a linear combination between training patches. Applying linear combinations are only applicable when similar geometries for both LR and HR images are available.

The super resolving algorithm starts by searching for a set of nearest LR patches to the LR input patch. In this method, the input patch is reconstructed as a linear combination of the recovered patches. This linear combination estimates the output patch as a linear combination of the stored HR patches using the coefficients calculated from **Locally Linear Embedding (LLE)** algorithm. This method causes enhancements compared to the NN-based. However, it can be useful as long

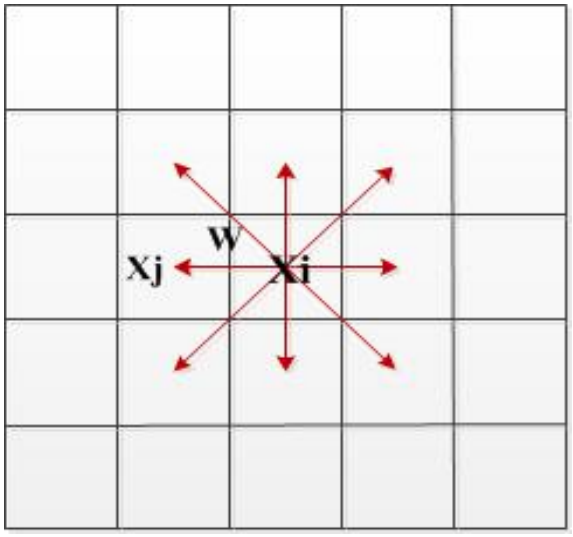
as the local geometry between LR and HR images is the same.

Locally Linear Embedding

LLE is an unsupervised non-linear learning technique where the high-dimensional data is considered to reduce the data dimension while local geometry is preserved. In LLE each original point, X_i, is reconstructed according to its neighbors, i.e. X_j. The reconstruction is based on a linear combination obtained from the weight matrix W_{ij}. The reconstruction error is simply computed by:

$$E(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2, \tag{2.4}$$

where $E(W)$ is the computed error, W_{ij} represents the amount of contribution of point X_j in reconstructing point X_i. Consequently, if X_j is not the adjacent neighbor of X_i, the weight factor W_{ij} would be zero. Figure 2.5 depicts a simple representation of reconstructing the original point. The Neighbor Embedding method is more clarified using the following algorithm.



(a)

Figure 2.5: Weights of reconstruction

Data: Input LR image I_l, training set (including e.g. 100,000 LR and HR patches), Interpolated Image I_l^h
Result: High resolution super resolved image I_h
for *each patch of* I_l **do**
searching in raster order through LR patches and find the closest ones;
Selecting K nearest LR patches, $N_x = \{x_1, x_2, ..., x_k\}$;
Finding the best weights W=$\{w_1, w_2, ..., w_k\}$ which can be utilized in presenting input patch x as a linear combination of $\{x_1, x_2, ..., x_k\}$;
Constructing y as a linear combination of $N_y = \{y_1, y_2, ..., y_k\}$. N_y is the HR correspondence to the N_x. They also use the same weights, i.e. y=N_y×W;
Finally, when y is achieved, it should be added to the correct position in I_h
end

Algorithm 2: Neighbor Embedding super resolution algorithm

2.2.3 Sparse representation

In sparse representation, just a few desired coefficients present the information. Signals, unlike the sparse representation, consist of enormous amount of data which make finding the relevant data as a difficult procedure. In order to make a sparse representation, signal should be decomposed into basic elements in a category called *dictionary* [36].

ScSR method in [24] introduces a sparse linear combinations of over-complete dictionary's elements used to represent the image patches. Patch-based method introduced by Yang et al. [24] derived from a theory which confirms that the relationship between HR patches can be recovered from their low dimensional projections. Two dictionaries of HR and LR patches, i.e. $D_h = [x_1, x_2, ..., x_n]$ and $D_l = [y_1, y_2, ..., y_n]$, construct the training sets. The LR images are achieved by downsampling an HR image. In these methods, a sparse representation is searched for each patch of the LR image. The achieved coefficients from this representation are used to create the HR image. Sparse linear combination of patches x of HR image X in dictionary D_h (HR patches) can be written as $x \approx D_h\alpha$ for some $\alpha \in R^K with \parallel \alpha \parallel \ll K$.

The problem of recovering the sparse representation of a given test LR image patch y_k^t (t shows the test patch from training set) can be solved by minimizing the l_1-norm:

$$\hat{\alpha} = min \parallel \alpha \parallel_1$$
$$s.t. \parallel y_t - D_l\alpha \parallel^2 < \sigma^2 \tag{2.5}$$

where D_l is the LR patch dictionary, y_t is a patch from LR dictionary and α is a desired coefficient.

The corresponding recovered HR patch is $x_k^t = D_h\hat{\alpha}$. It can be seen that, unlike the NE estimation algorithm where K fixed neighbors are selected in order to reconstructing the HR patch, few essential coefficients are chosen. This issue is also helpful in overcoming the over-fitting problem.

2.3 Results and discussion

In this section, implementation results (visual and numerical) of the Freeman's example-based, Neighbor Embedding and also ScSR methods are presented. Before going through the discussion, two major assessment tools should be addressed first. These are **Subjective** and **Objective** methods. In the subjective method, a human with the Human Visual System (HVS) evaluates the output image. In this case, there is no reference image to compare the result with. In the objective method, the numerical results present the performance of the proposed algorithm. Consequently, in order to compute the amount of distance between the input image and its reconstructed HR version, both images must have the same size. Hence, to compute the numerical results, the downsampled version of the original image should always be used through the computations. Afterwards, the super resolved HR image has the same size as the original image and they can be compared with each other.

Peak Signal-to-Noise Ratio (PSNR) and **Root Mean Squared Error** (RMSE) are the two common numerical tools in assessing the performance of an algorithm. PSNR measures the signal power ratio to the noise power and RMSE shows the error of reconstruction algorithm. They can depict the amount of similarity between the reconstructed result and the original data. The PSNR and RMSE are defined by:

$$RMSE = \sqrt{(1/mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2)}$$
$$PSNR = 20 \times \log_{10} \frac{MAX_I}{RMSE} \tag{2.6}$$

where, m and n depict the size of the images, I and K are the original reference and reconstructed images, respectively.

Table 2.1 shows the numerical results of Freeman's method for various scaling factors. As it can be seen the PSNR values are decreased when scaling factor is increased. The amount of decrement in PSNR values from ratio 2 to 3 would be higher than the one from ration 1.5 to 2. When an image is enlarged e.g. 3 times it means that for each pixel 3 pixel are needed to be predicted. This is due to blurring in the reconstructed image and losing more HF components. As a result, higher scaling factor would cause lower PSNR value.

In order to make a fair comparison, the implementations of different methods must be under the same situations and conditions. In this experiment, both methods have one similar training image as well as the same scaling factor of 4. It should be mentioned that all the experiments have been accomplished in luminance channel. Figure 2.6 and Table 2.2 represent the result of two different types of pictures.

Table 2.1: Freeman method, PSNR values of Lena and Textured images, different scaling factors (R).

	PSNR values		
Images	R=1.5	R=2	R=3
Lena	26.3059	25.6644	24.9691
Textured Image	21.7661	21.6270	21.0465

Table 2.2: PSNR values of Lena and Textured images, Scaling factor is 4

	PSNR values	
Images	Example-based Freeman	Neighbor Embedding
Lena	24.4292	27.7688
Textured Image	20.3653	24.7296

As it can be seen from Figure 2.6 picture (b), some noise is introduced due to the heavy dependency of the algorithm to the training data. Numerical and visual results in Table 2.2 and Figure 2.6 show that the NE method has better performance compared to the Freeman's method. The values of PSNR for both *Lena* and *Textured* images in NE estimation are considerable higher than the ones in Freeman's method.

Figure 2.7 and Table 2.3 depict another results through various images. In this figure and table, ScSR has been compared with other methods. The results are gathered from [24] to make wide experiments. Pictures (a) and (b) in Figure 2.7

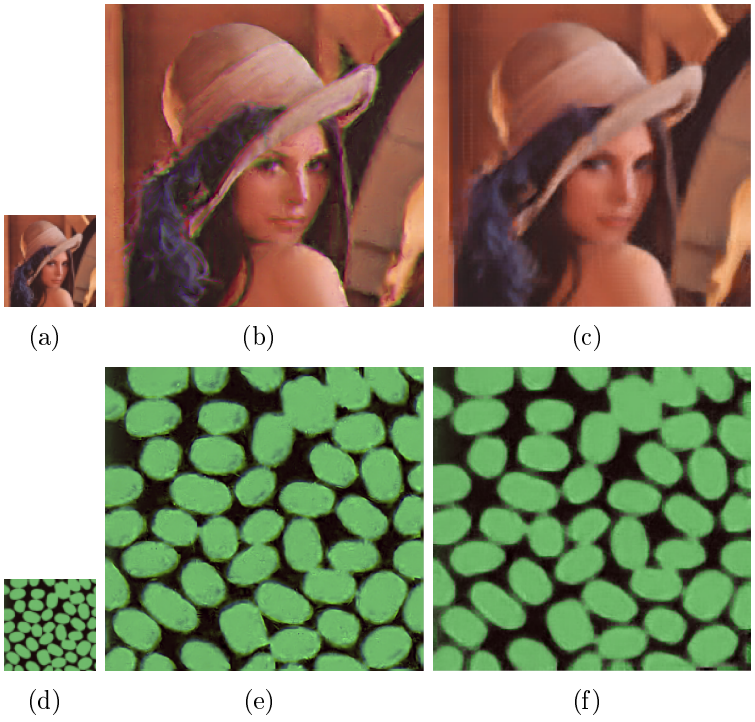


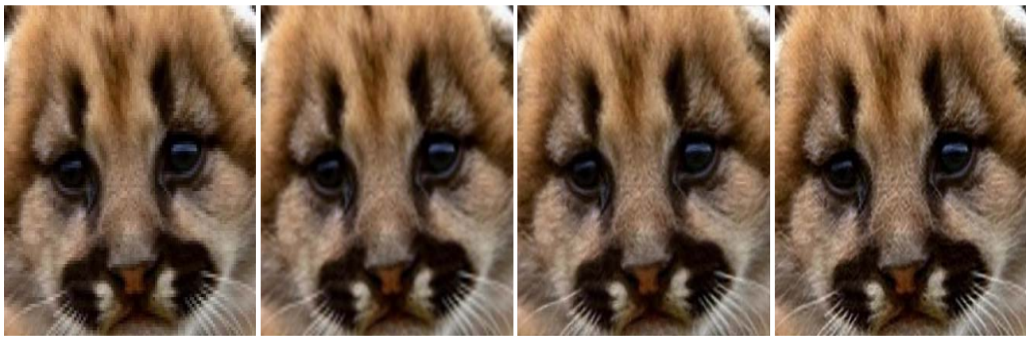
Figure 2.6: a)LR Lena Image b) Super-Resolved image achieved by Freeman algorithm c) Super-Resolved image achieved by NE algorithm d) Textured Image e)Super-Resolved image achieved by Freeman algorithm f) Super-Resolved image achieved by NE algorithm. Scaling factor is 4.

Table 2.3: The RMS values of different images achieved using different methods [24].

	RMS values for scaling factor 3		
Images	Bicubic	Neighbor Embedding	ScSR
Flower	3.5052	4.1972	3.2276
Girl	5.9033	6.6588	5.6175
Parthenon	12.7431	13.5562	12.2491
Raccoon	9.7399	9.8490	9.1874

depict input, bicubic interpolated, NE estimated, ScSR estimated and the original images from left to right respectively. In this comparison, Neighbor Embedding is able to create sharp edges compared to bicubic results, however it causes some blurring. Sparse method shows more details in the results compared to NE. Table 2.3 illustrates ScSR method has better performance in reconstructing an HR image compared to other methods (lowest RMS values). According to this table, NE method was not successful in comparison with bicubic interpolation in terms of RMS value.

(a)



(b)

Figure 2.7: a)Left to right: Input, Bicubuc, NE [12], Sparse Representation [24] b) Left to right: Input, Bicubuc, NE, Sparse Representation. scaling factor is 3 [24].

2.4 Conclusion

The learning process in example-based algorithm is introduced as an inaccurate and slow algorithm. The proposed example-based method by Freeman et al. recovers the high resolution output image. Although the Freeman's algorithm sharpens the edges, it consists of magnificent amount of noise as well as irregularities. This impairments mainly happen due to the lack of relevant examples or patches. Freeman's algorithm is strongly dependent on a huge data set used in the training stage. NE method is a good candidate to reduce artifacts and keep the general shapes safe. It works similarly to Freeman's algorithm in terms of reconstructing an HR image. However, the NE method cannot be successful for all images due to the lack of adaptive selection of training elements. The number of elements in the dictionary as well as the amount of neighboring region (which is considered in each patch) should be selected adaptively. Consequently, this method can not be successful for all images since it is dependent on the selection of the training images. This issue could be almost covered by sparse coding methods. ScSR method by addaptively choosing the desired data could overcome over-fitting problem. However, there is no certain number of raw samples available for constructing dictionaries.

Example-based super resolution methods are effective in environments with less availability of the observations. However, a number of improvements are needed to find out which patch size would be appropriate for a given image based on the image content. Moreover, it would be better to make the training data adaptive for each input image in order to increase the efficiency and decrease the computational cost.

3. REGRESSION-BASED

This chapter presents another technique of learning-based category known as *Regression-based*. We will clarify this method by implementing a regression-based approach. Extracting some significant parameters may help to know algorithm design in different situations. Finally, this parameter representation helps to obtain improvements through the performance of the algorithm both visually and numerically.

3.1 Introduction

One of the major demands in learning-based super resolution methods is to discover a relationship between LR patches with the corresponding HR ones. In other words, it is challenging to find out an appropriate mapping function, f, in order to map patch $\mathbf{X}$ in the low resolution space into $\mathbf{Y}$ in the high resolution space [31]. As it was mentioned in the previous chapter, Chang et al. in [12] presented a novel method to overcome the constraints, such as overfitting, in NN-base method. Another method introduced in this chapter is making a regularization on regressor directly. This is a direct approach to overcome the NN-based method's problems [19]. For instance, **Support vector regression** is a technique to cope with regression problems in frequency domain proposed in [32]. In [15], matrix-value regression has been used to overcome super resolution constraints. We also exploit this method in our experiments while making some modification to the original code. Implementing the modified algorithm alongside considering prominent parameters such as patch size, we could achieve better results compared to the original code.

3.2 Regression algorithm

Matrix-value regression technique helps to improve the efficiency of the learning process by diminishing the impact of the large training data in single-image super-resolution algorithms. The aim of the matrix-value regression is to promote learning efficiency from image pairs [15]. Tang et al. in [15] mostly concentrated on training data to have an efficient data set. In this method, the regression operator is considered as a matrix learned irrespective to the number of training set images. SISR has been presented as a multi-task learning model in this regression method. Multi-task learning facilitates performing several related task, simultaneously. It

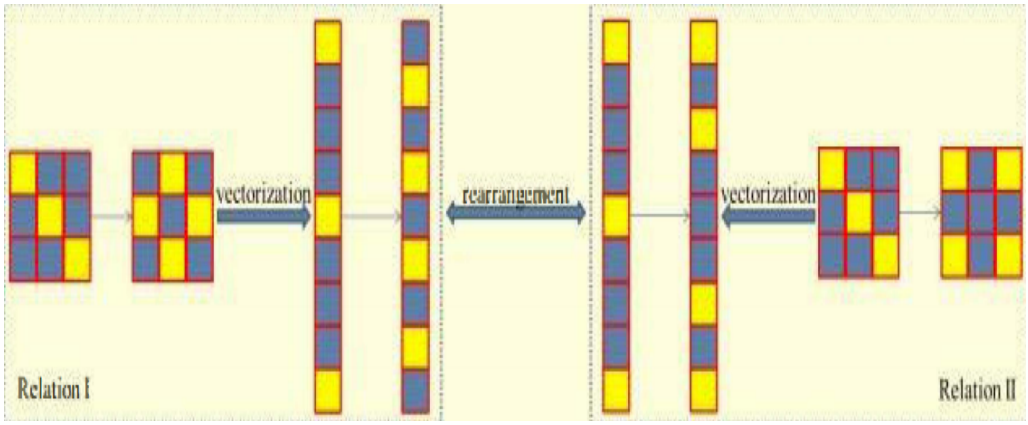


Figure 3.1: Vector to vector VS Matrix to Matrix [15]

has also claimed that multi-task learning can not be adequate to present the relationship between patches when the vector-to-vector model is employed. Therefore, they have proposed a *matrix-to-matrix* multi-task learning. Figure 3.1 presents the vector-to-vector versus matrix-to-matrix relations.

The matrix operator makes the relationship between low and high resolution patches. The prior information in training images are used to make the operation matrix. A super resolved image patch is estimated using the matrix operator. In the next section, we will present the training stage which is the most important part of this algorithm.

Training procedure

At this stage, all training images are read. These images are considered as high resolution images. The low resolution ones are obtained by executing down-sampling and, consecutively, up-sampling operations on the HR images. Afterwards, images are segmented based on the defined patch size, e.g. 3. All the patches in LR and HR training images are employed to create the operation matrix A. The training procedure is terminated as soon as the matrix A is estimated.

The Matrix operation algorithm can be achieved as following. Let's assume $A : X \longrightarrow Y$ to be the operator which maps the low resolution patches to high resolution ones. Thus, the training set which contains n pairs of the training samples is formed as:

$$S = (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n) \tag{3.1}$$

where X_i represent LR patches and Y_i their corresponding HR ones. The least square regression technique is used to learn the optimal matrix operator as following:

$$A = argmin \sum_{i=1}^{n} \|Y_i - AX_i\|_F^2 \tag{3.2}$$

where A is the matrix operator and $||.||_F$ is the Frobenius norm (an Euclidean norm). In order to reach the optimal matrix operator, F(A) should be:

$$F(A) = \sum_{i=1}^{n} \|Y_i - AX_i\|_F^2 \tag{3.3}$$

By taking into account the essential condition of the *minimum*, where the derivative is equal to zero, the matrix A is learned as:

$$\begin{aligned}
&\frac{\partial F(A)}{\partial A} = 0 \\
&\frac{\partial}{\partial A} \sum_{i=1}^{n} (< Y_i, Y_i >_F - 2 < Y_i, AX_i >_F + < AX_i, AX_i >_F) = 0 \\
\Leftrightarrow &\frac{\partial}{\partial A} \sum_{i=1}^{n} (< Y_i, Y_i >_F - 2 < Y_i X_i^T, A >_F + < AX_i X_i^T, A >_F) = 0 \\
&\Leftrightarrow \sum_{i}^{n} AX_i X_i^T = \sum_{i}^{n} Y_i X_i^T
\end{aligned} \tag{3.4}$$

The optimal matrix operator obtained based on equation (3.4) is:

$$A = \sum_{i}^{n} Y_i X_i^T (\sum_{i=1}^{n} X_i X_i^T)^{\dagger} \tag{3.5}$$

where X_i^T is the transposed version of **matrix** X_i and $()^{\dagger}$ is the generalized inverse.

After the training process, the final procedure is performed as soon as the matrix A is available. This part consists of executing the super resolution algorithm on the input test image. Algorithm starts by enlarging the test image with a desired scaling factor. Each patch of the enlarged image is multiplied by the matrix A. Eventually, the result of this operation will give the recovered HR image. Figure 3.2 illustrates the entire process of the algorithm. In this figure, the first row shows the test image which is interpolated by any interpolation algorithm, e.g. bicubic, and then its HF components are recovered by estimated matrix A (estimated from learning stage). In the second row, the LR and HR version of the training image are achieved in

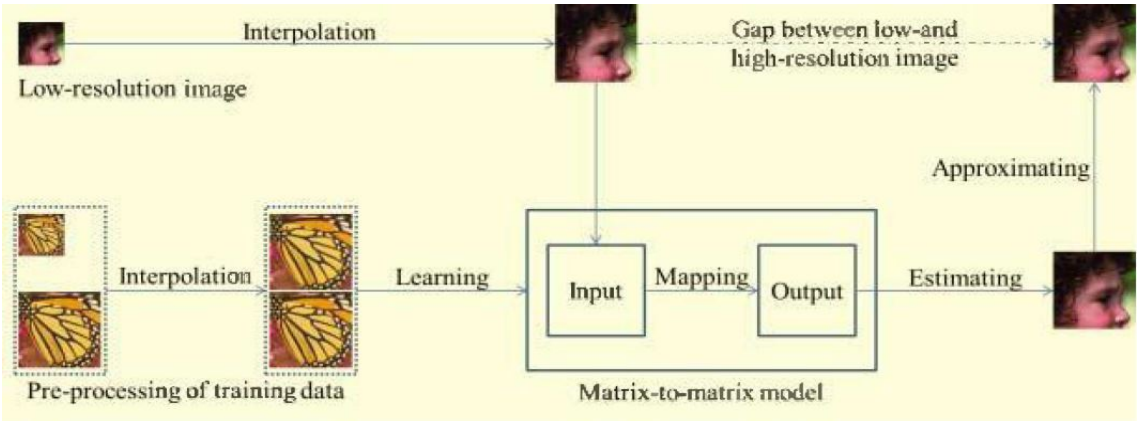


Figure 3.2: The algorithm model for super resolution [15]

order to start learning procedure and find mapping function.

The entire algorithm is presented as following:

Data: Input image I_0, training images I_t, Smoothed version of the input image L_1

Result: High resolution super resolved image I_T

Set I_0= Original Image, Input image to the algorithm is the smoothed version of the original image, L_1=U(D(I_0)), where U and D are the bicubic upsampling and downsampling operators respectively;

Training:

Reading all training images and consider them as HR training images;

First down-sample, then, up-sample the HR training images to achieve the blurred version considered as the LR training images; **for** *each patch (e.g. 3 ×3) x in LR images and y in HR images* **do**

| $A = \sum y_i x_i^T (\sum x_i x_i^T)^{\dagger}$

end

Super Resolving Algorithm:

for *each patch z in smoothed image* **do**

| Super Resolved-patch=$A \times z$;

| Inserting the achieved patch to the correct place in the output image I_T;

end

Algorithm 3: Reconstruction of high resolution images by Matrix Regression method

We have exploited Freedman's method in our work, however there are some modifications applied in the main code. In the next section, we present a modified algorithm and discuss more about the modifications which may lead to improvements in results.

3.3 Modifications of the progress

In the previous section we have presented the entire algorithm and the implementation levels. In this section, the modifications and parameters which have been applied to improve the results will be investigated. As it can be seen from previous section, the algorithm was based on patches. In other words, patches from two sets of images (HR and LR) are collected to achieve the operation matrix A. In contrast, in our experiments the transformed version of the patches, i.e. its corresponding vectors, are used. For instance, patch size equal to 3 should be converted to a vector of nine elements (9×1). Afterwards, all the operations are applied on the achieved vectors. The transformation to a vector can be performed in any direction. In other words, the results of a patch transform to either a column or a row are the same. This technique is applied for any desired patch size. The corresponding results of the modified code have been improved in comparison with the original one in both visual and numerical points of view. The conversion from patches to vectors causes changing in size of operation matrix A. It is the squared size of the corresponding patch (i.e. for patch size 3×3, matrix A is 9×9). Operator matrix, B, is achieved as following:

$$B = \sum_{i}^{n} \mathbf{y_i} \mathbf{x_i^T} (\sum_{i=1}^{n} \mathbf{x_i} \mathbf{x_i^T})^{\dagger} \tag{3.6}$$

where $\mathbf{x_i^T}$ is the transformed version of LR **vector** $\mathbf{x_i}$, $\mathbf{y_i}$ is the HR vector, and $()^{\dagger}$ is the generalized inverse. According to the experiments, transforming a patch to other possible forms of matrix is not improving results. Consequently, all experiments have been performed with vectors form of patches.

The effect of patch size is another major concern which should be considered in SR algorithms. In order to assess both performance of the algorithm and sensitivity to the selected patch size, the algorithm has been repeated for various patch sizes in our experiments. It was observed that increasing the patch size can make some improvements in visual and numerical results. Since these global improvements have been achieved through an image recovered from various patch size, the efforts were made in reconstructing a switching operator. This operator made some local improvements.

3.3.1 Switching Operator

The achieved results from the modified version showed that image quality can be dependent on the patch size. Hence, in this thesis instead of using one fixed patch size, a mixture of different patch sizes are employed by applying a proper switching operator. When the four resultant images have been estimated from fixed

patch sizes (3, 5, 7, 9), the switching operator compares each 9×9 patches (in order to have a correct comparison, size of this patch should be equal to the maximum size of patches which have been used.) of these images with the original one. The comparison is performed by calculating the mean squared error between each patch and the original one. As a result, four numbers will indicate the distance of these patches from the original patch. The patch corresponding to the minimum number is the closest candidate to the original one. The central pixel of the closest patch is selected to be the corresponding pixel in reconstructed patch. This procedure will be continued for all 9×9 patches within the images. Finally, the image is created with a mixture of different patch sizes. For each 9×9 comparison, the selection of switching operator is stored in a matrix. The corresponding index of the closest patch size is stored in a matrix each reference. The values of the created matrix, i.e. gray levels, are changed from 1 to 4 according to the contents of the input image and selected patch. The small patch sizes are good candidates in smooth area. Since more changes exist in the areas with more contents, the larger patch size covers more discontinuities and edges. Figure 3.3 depicts the image reconstructed by indexes to each patch size.

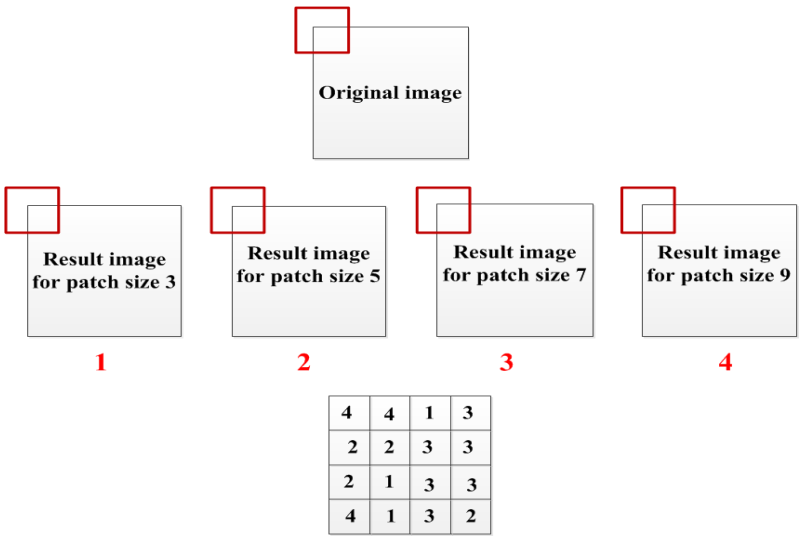


Figure 3.3: Top row: original image, middle row: reconstructed images from fixed patch sizes, bottom row: image reconstructed by patch size selection

As it can be seen, for the first pixel, each 9×9 red patch is compared with original one. The central pixel of the nearest patch to the original one is stored in a new reconstructed image. The number of corresponding image is stored in a matrix (see bottom image). In this example, for the first comparison the nearest patch is for image number 4 (i.e. image reconstructed with patch size 9). This is performed for all 9×9 patches.

3.3.2 Initial interpolation

In all experiments, the bicubic interpolated version of the input image is used to recover its missing HF components. We repeated the algorithm again in order to experiment whether it operates fine for any other interpolation methods. This time, instead of using bicubic algorithm as an initial interpolation, we have used the recovered image from patch size equal to 3. This image is sent through the algorithm to recover the image with the patch size (i.e. 5). This procedure is performed sequentially until the last patch size. In these experiments, the improvements can be observed in results. Figure 3.4 shows this sequential procedure. It should be mentioned that every interpolation algorithm used as the initial interpolation method should be also used in the training procedure.

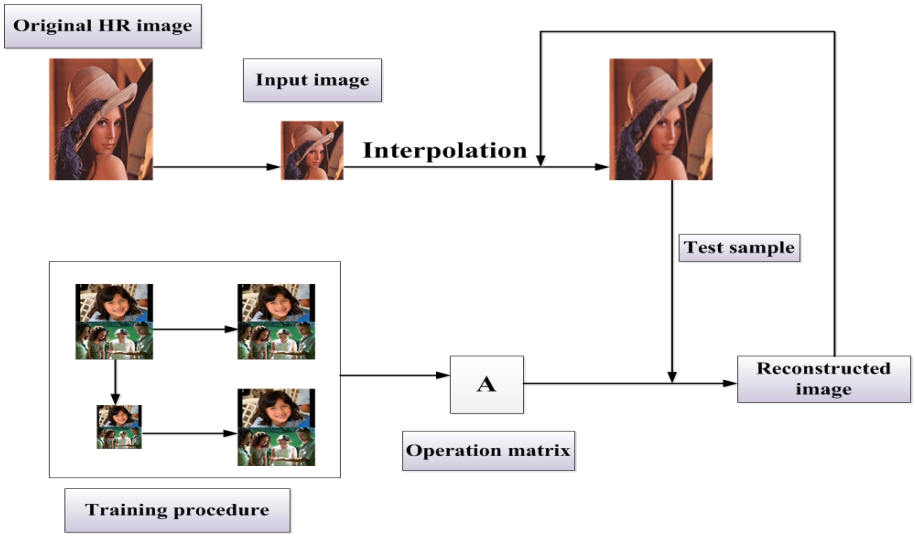


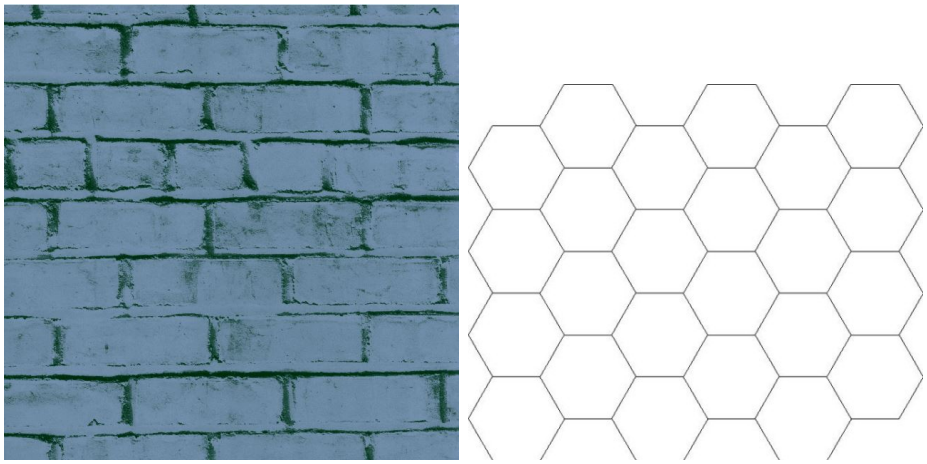
Figure 3.4: sequential algorithm

3.3.3 Training images

Figure 3.5 shows the input training image for both main and modified algorithms. In the last experiment, we changed the input training image in order to investigate the effect of different types of training images. Figure 3.6 shows two different training images where picture (a) is related to the geometric images and picture (b) refers to the textured ones.

Images which are used in training process should be chosen properly to achieve satisfactory results. Experiments showed that the performance of the algorithm has some dependencies to the images used in the training procedure. In other words, if the training image is far from the test image (in terms of content), the results will not produce significant improvements.

Figure 3.5: Training Image



(a) (b)

Figure 3.6: a) Textured, b) Geometric. Both are used as training Images

It is worth mentioning that all low and high resolution color images are presented in YCbCr channels. Since the human visual system is more sensitive to luminance channel, the algorithm and computation steps were performed in Y channel. Moreover, for evaluating the performance of different algorithms, the results must be compared under the same conditions. For instance preliminary interpolation method, zooming factor, image domain, etc. must be the same for different implementations.

3.4 Results

The results shown in this section are achieved from implementing the main algorithm proposed in [15] as well as the modified version of it. In order to obtain more experiments, this algorithm is examined using different images in terms of content. Tables 3.1 and Figure 3.7 present the numerical and visual results.

Figure 3.7 depicts three images with their interpolated versions in three different methods. From left to right original image, bicubic interpolated image, image

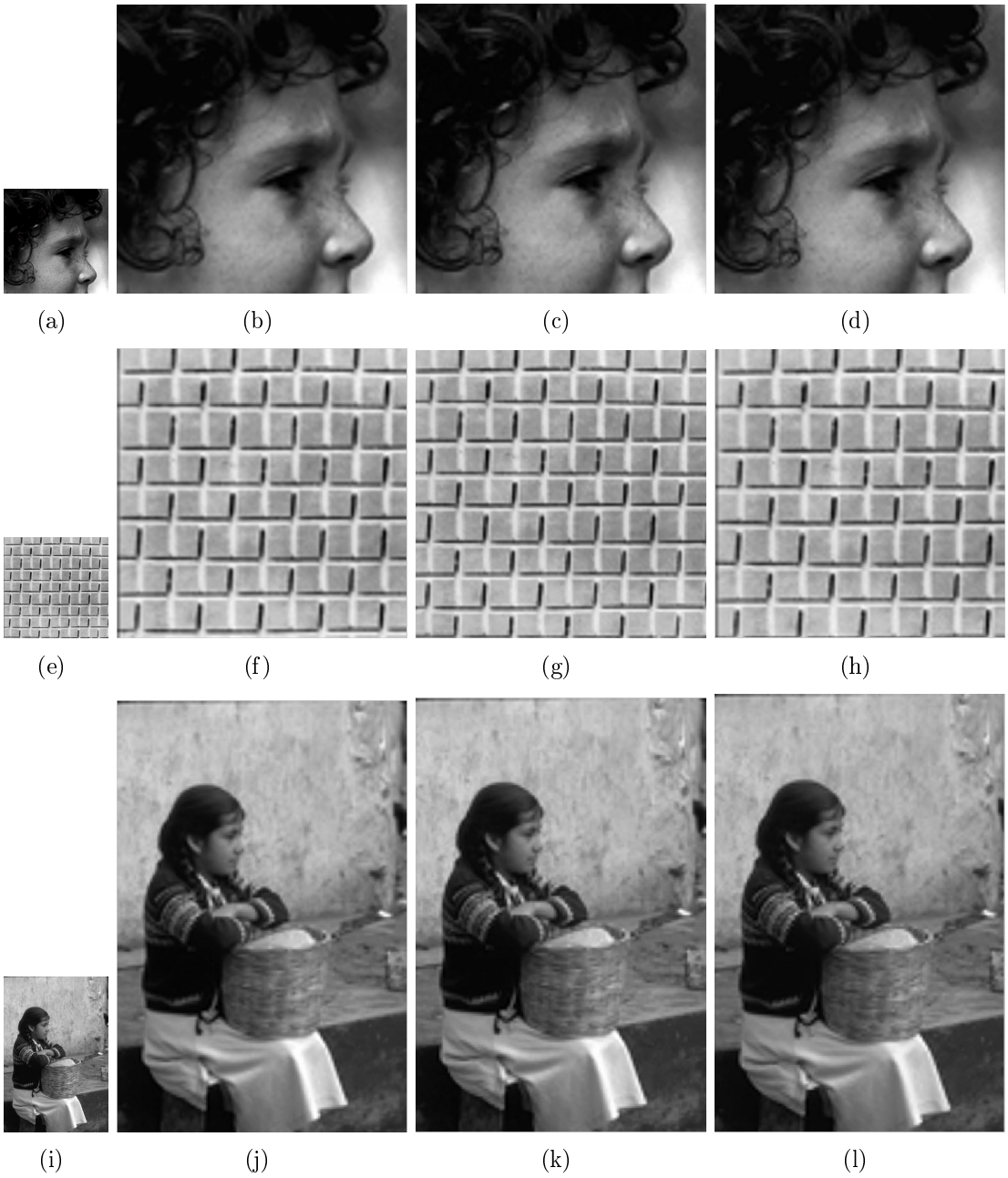


Figure 3.7: a)e)i) Original images b)f)j) Bicubic interpolated images c)g)k) images obtained by main regression-based method d)h)l) images obtained after modifications, Patch size=3

Table 3.1: PSNR values, Ratio=3 and Patch size=3

	PSNR values		
Images	Bicubic	Main Regression	Modified Regression
Child Face	32.90	33.00	33.13
Textured image	27.42	27.51	27.72
Girl image	29.03	29.09	29.1

achieved from implemented main regression algorithm and finally image obtained from our modified code are depicted from left to right respectively. As it can be seen, the impact of the algorithm for both numerical and visual results is not impressive. The difference between PSNR values in bicubic interpolation and regression method for patch size equal to 3 with enlargement factor equal to 3 is negligible.

In Table 3.2 the numerical results of two methods, for scaling factor 3 with different patch sizes, are collected. In our experiments, by increasing the patch size, PSNR values in the modified code are increased, while the corresponding values in the main code decreased. These results are obtained from a textured image with total improvement of approximately 1 dB (between the PSNR of last patch size and the bicubic one). Another image which differs from the previous image (in terms of content) is used. Table 3.3 and Figure 3.8 present the results for this facial image. In this table, the same as Table 3.2, improvements in PSNR values is observed when the patch size increases. In Figure 3.8, the differences between results of two codes are clear. The images on the left are obtained from our experiments with modified code, the images on the right are achieved from the main regression code. Our results show more details in comparison with the original code's results. Moreover, moving from up to down in this figure shows super resolved images with patch sizes 3, 5 and 7 respectively.

Table 3.2: Textured image, PSNR values, Ratio=3 and Different patch sizes

	PSNR values		
Patch size	P=3	P=5	P=7
PSNR, modified code	27.72	28.01	**28.39**
PSNR, original code	**27.51**	27.07	26.95

Table 3.3: Child face, PSNR values, Ratio=3 and Different patch sizes

	PSNR values		
Patch size	P=3	P=5	P=7
PSNR, modified code	33.13	33.30	**33.46**
PSNR, original code	**32.9950**	32.6084	32.6824

In order to consider the differences in images achieved from two codes (Figure 3.8), results for patch size equal to 5 are zoomed and shown in Figure 3.9. It can be seen that more details are detectable in Figure 3.9 (b).

Table 3.4 depicts numerical results of different images with various patch sizes. From this table, it can be observed that the best PSNR values are achieved when

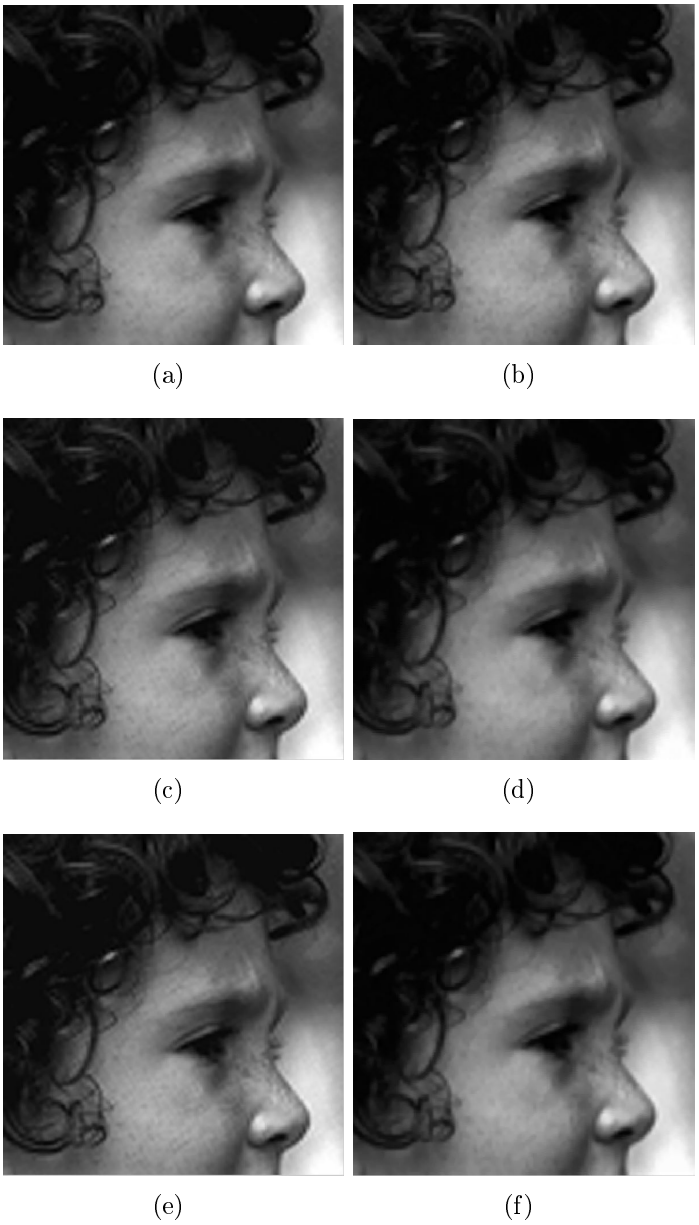


Figure 3.8: a) Patch size is 3, modified code b) Patch size is 3, main code c) Patch size is 5, modified code d) Patch size is 5, main code e)Patch size is 7, modified code f) Patch size is 7, main code

the patch size is not fixed (last column). Changing the patch size not only globally but also locally leads to have improved numerical results. The improvements are not always considerable. Comparing result of last patch (i.e. 9) with the mixed one depict that the improvement for the textured image is about 2 dB, however for others are almost about 0.1 to 0.2 dB.

Figure 3.10 shows the visual results of the super resolved image from the mixture of patch sizes. Moreover, in Figure 3.10, another image is created according to the

(a) (b)

Figure 3.9: a) Main Code b) Modified Code, Both images are zoomed in and the patch size is 5.

Table 3.4: Modified regression, PSNR values, Different patch size and the mixed

	PSNR values				
Images	P=3	P=5	P=7	P=9	Mixed patches
Child face	33.13	33.30	33.46	33.49	**33.50**
Textured image	27.72	28.01	28.39	28.5	**30.19**
Camera man	26.21	26.39	26.65	26.68	**27.78**
Girl	29.09	29.25	29.40	29.42	**29.82**

switching operator which selects the proper patch size for each HR patch reconstruction. It is expected to have larger window in edges. For instance, in Figure 3.10 (a) for edges contents (lines in the figure), white regions are appeared in the Figure 3.10 (b). The same issue has happened for the girl and cameraman images when the background is interrupted by object.

As it has been mentioned before, the initial bicubic interpolation can be replaced by other methods. For instance, after super resolving an image with patch size equal to 3, we have utilized the result image as the input to obtain next image with patch size equal to 5. The result image is again used (as an input) to achieve the next recovered image with patch size equal to 7. Table 3.5 shows the PSNR values of this sequential experiments. As it can be seen, increasing the patch size improves the PSNR values.

Table 3.5: Child face, PSNR values, Different patch size and the mixed

	PSNR values				
Images	P=3	P=5	P=7	P=9	Mixed patches
Child face	33.19	33.39	33.52	33.56	**33.60**
Textured image	27.58	27.92	28.39	28.55	**30.04**

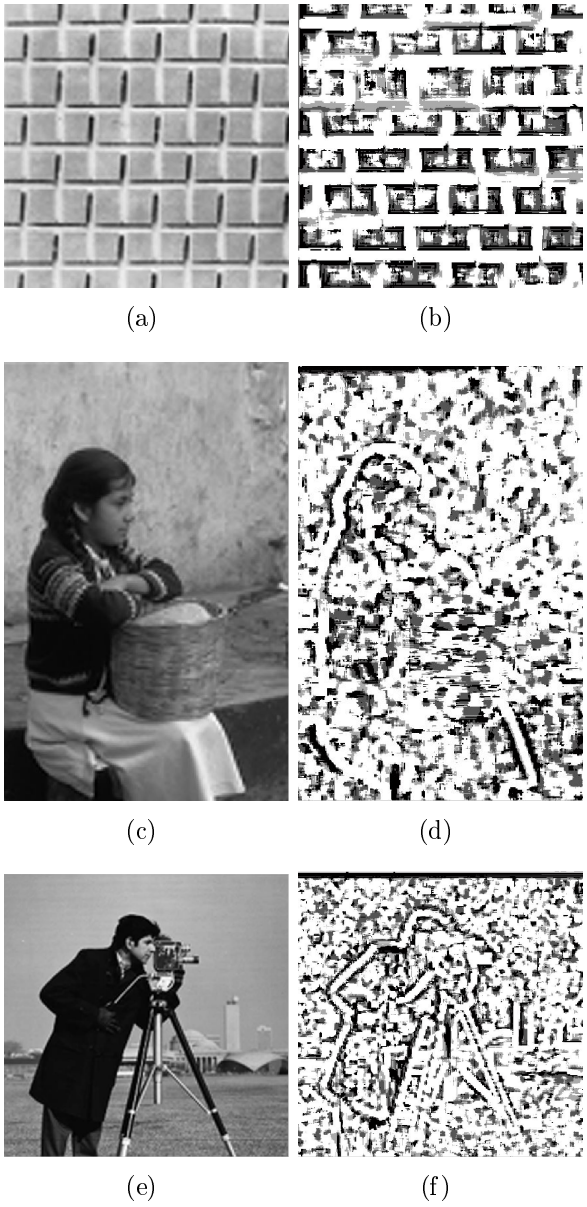


Figure 3.10: a)c)e) Super-resolved images with mixed patch sizes b)d)f) Produced image from switching operator

All the previous experiments have been performed with the same training image (Figure 3.5). In the next experiment, we will show how the training image may impact the results. Table 3.6 illustrates the results of two different images, when the training image is a geometric one. Table 3.7 also shows the results of two different test images, when the training image is the textured image. As it is clear from both tables, textured test image can be recovered better from the textured training image and, consequently, the geometric test image can be recovered better from the geometric training image. This is due to the similarity between the training and the test images, in terms of content. In case of having different training and test images (one is textured and other one is geometric), PSNR value is less than

the case when both training and test images are from the same category.

Table 3.6: PSNR values, Geometric training Image, Two different test images

Patch size	P=3	P=5	P=7	Mixed patches	Bicubic
PSNR, Textured test image	28.9293	28.9431	29.6088	**29.9078**	28.7271
PSNR, Geometric test image	26.2804	26.6898	27.0612	**27.3795**	25.8854

Table 3.7: PSNR values, Textured training Image, Two different test images

Patch size	P=3	P=5	P=7	Mixed patches	Bicubic
PSNR, Geometric test image	25.9808	25.9857	26.2547	**26.3666**	25.8854
PSNR, Textured test image	28.9074	29.2648	29.8021	**30.1341**	28.7271

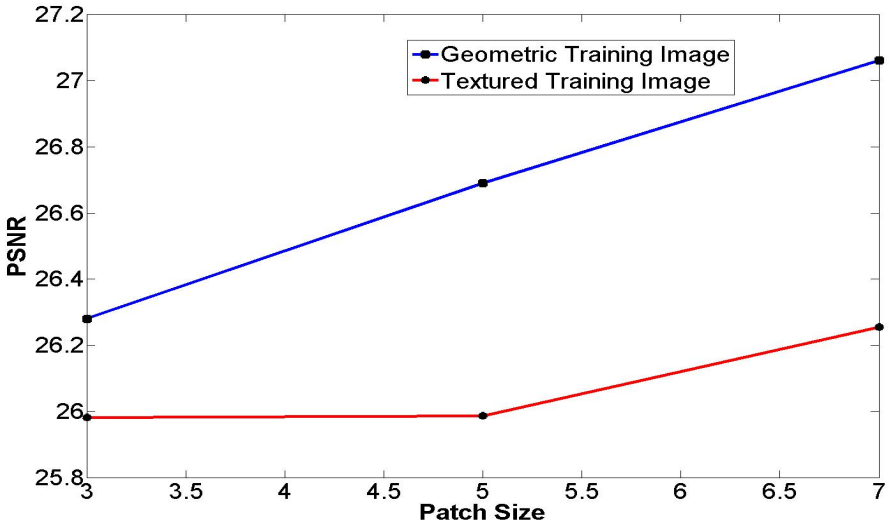


Figure 3.11: PSNR values of super resolving a geometric test image for different patch sizes. Red curve is for Textured Training image and the blue one is for geometric image

Figure 3.11 depicts changes of PSNR values for the geometric image which has been super resolved by two different training image sets. As it is shown, the red curve and the blue one are related to the textured and geometric training image sets, respectively. In this experiment, since the test image is the geometric one, more relevant data exist between geometric training image and the test image. Hence, it is expected to obtain larger PSNR values when the training and test images have more correlation. As the Figure 3.11 shows, the curve with geometric training image stands higher than the curve with textured training image in terms of PSNR values.

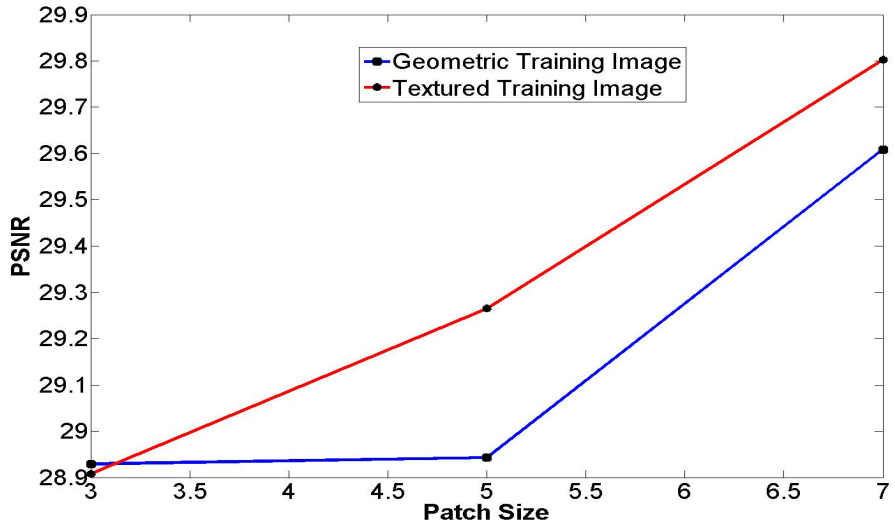


Figure 3.12: PSNR values of super resolving a textured test image for different patch sizes. Red curve is for Textured Training image and the blue one is for geometric image

The same scenario occurs when the test image is textured. Figure 3.12 shows how the position of the curves are altered when the test image has more correlation to the textured training image.

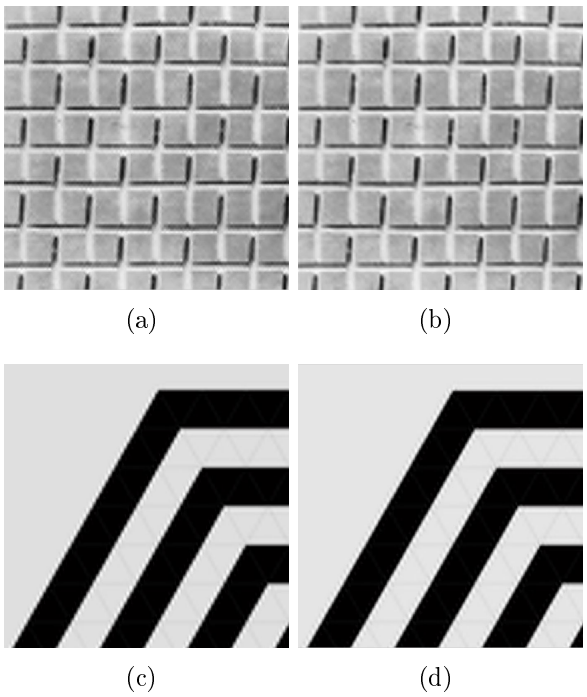
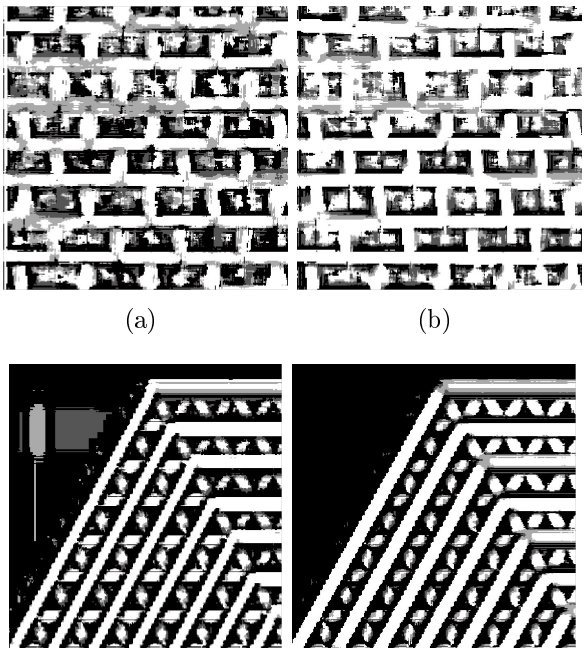


Figure 3.13: a)Test textured image recovered from geometric training image b) Test textured image recovered from textured training image c)Test geometric image recovered from textured training image d) Test geometric image recovered from geometric training image



(a) (b)

(c) (d)

Figure 3.14: a) Test textured image recovered from geometric training image b) Test textured image recovered from textured training image c)Test geometric image recovered from textured training image d) Test geometric image recovered from geometric training image

Figure 3.13 shows the visual results of these experiments. Figure 3.13 pictures (a) and (b) show recovered textured image from geometric and textured images respectively. Figure 3.13 (c) and (d) depict recovered geometric image from textured and geometric images, respectively. In images (a) and (c), some irrelevant information which degrade the image quality exist. This is due to recovering the test image from a different training image (in terms of content) .

Figure 3.14 illustrates images created by the selection of proper patch size (the result of using various patch size). The training image can have impact on switching along with the selection of proper patch. Figure 3.14 (a) and (c), include highlighted parts where the switching has not operated perfectly.

3.5 Conclusion

This chapter illustrates a single image super resolution method which could overcome some constraints, e.g. dependency to huge training data set, originated from previous work (example-based). All the experiments have been completed through the regression-based method. Regression method aims to find a mapping function between low and high resolution patches. The main idea of this chapter was based on one of the previous work in regression methods. Improved results were achieved by

exploiting some parameters and features such as patch size, vector-based multi-task learning, etc.

One of the advantages of the regression-based method is the cost of the implementation in SR. The complexity of the computations in an NN-based algorithm (which uses the nearest neighbour method) is compromised due to the local characteristics. However, the cost of the computation for a huge training set is potentially high. Hence, the regression-based method can obtain satisfactory results regardless to the amount of training set.

Parameters investigated through this work cause significant effects on the results. Output image created by different patch sizes showed the best numerical values as well as visual presentations. Implementing a switching operator could help to achieve the best result in PSNR within various images. Moreover, converting the patch size through the training process to a vector led to obtain more acceptable results. One of the major concern which should be taken into consideration is about training image. The correlation between training and test images would make results more accurate.

4. SELF-SIMILARITY-BASED

This chapter describes self-similarity as the final learning-based method investigated in this thesis. The implementation procedure along with the corresponding numerical and visual results are presented. Furthermore, few modifications applied to the main code as well as the idea behind that are discussed in detail.

4.1 Introduction

Self-similarity is a method which exploits similarities alongside the structural information of patches which can be employed to reconstruct the HR image. Chih-Yuan Yang et al. in [18] introduced the problem of super resolution from Example-based perspective. They also claimed that it is not necessary to use an external imaging set to perform mapping learning between low-resolution and high-resolution patch pairs. In such techniques, instead of extracting patches from external image data base, further efforts are made to study the relationship between patches in different scales of the input image.

In [27], Sae-Jin Park et al. categorized self-similarity into two different classes. The classification is performed upon the used domain. In the first class, the concentration is mainly on the spatial components meaning that the searching application is performed through spatial version of pyramid scales. This kind of self-similarity approach can be seen in [23] where the resolution enhancement procedure is based on patch redundancy. In the second category, the self-similarity technique is based on the frequency components of the input image. In frequency-based self-similarity algorithms, detachment of low frequency and high frequency components is occurred. This decomposition in the image components performed to find the similar patches through the pyramid scales of the original image. Since this technique is built based on LF and HF domain, it can be expanded in Learning-based SR methods as well. However, in self-similarity SR methods, there is no need to have a prior dataset. The pyramid scales of a given image is illustrated in Figure 4.1.

There are several papers focused on developing self-similarity methods. Noriaki Suetake in [22] tried to design a codebook to show the relationship between low and high frequency components according to the local self-similarity. In this mapping,

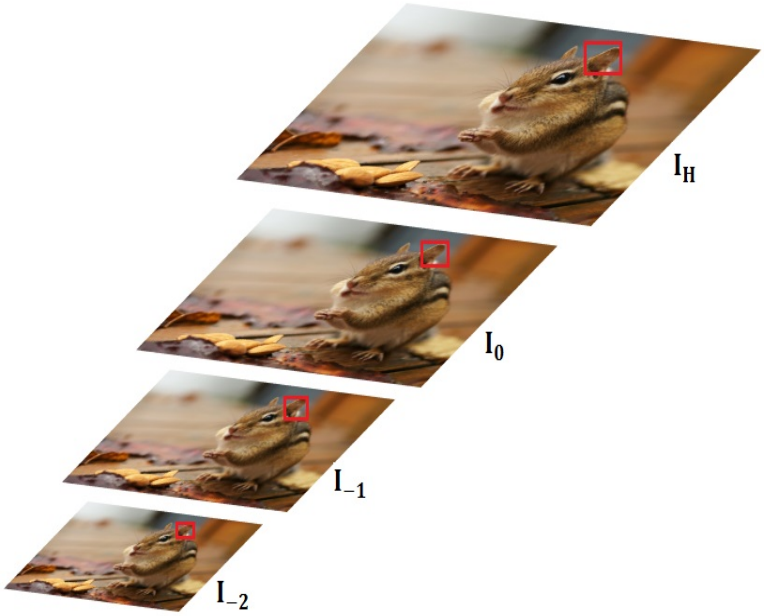


Figure 4.1: Pyramid Scales. I_0 is the LR input image. I_{-1}, I_{-2} and etc. are the downscaled version of input. I_H is the HR output image.

an image whose high frequency components are estimated locally is reconstructed. Daniel Glasner et al in [23] have proposed a self-similarity approach which is a combination of two previously mentioned methods: traditional *Multi-frame SR* and *Example-based*. The idea behind this work is that each patch in a natural image tends to be repeated several times over either the same or different scales. The repetition over the same scale is originated from the classical multi-frame SR techniques and the repetition over different scales is originated from the example-based SR methods, respectively.

In following sections, the self-similarity method proposed by Freedman et al. in [26] is discussed. It will be shown how self-similarity can preserve details and HF components.

4.2 Local self-similarity method

As it was mentioned before, there is no prior example database in self-similarity method. The proposed SR technique in [26] was performed using local self-similarity method. The algorithm is based on a characteristic of natural images attributed as the similarity of small patches when the scaling factor is small. This algorithm is appropriate for small scaling factors. Moreover, large scaling factors are used by employing several upscaling procedures to obtain the required factor.

4.3 Implementation

The implementation procedure is split into two main sections. The first section covers explanations related to the upscaling method process. The second part emphasizes more on the matching algorithm procedure to super resolve an image with a reasonable quality along with the high frequency components. This procedure is performed by utilizing both LR and HR images.

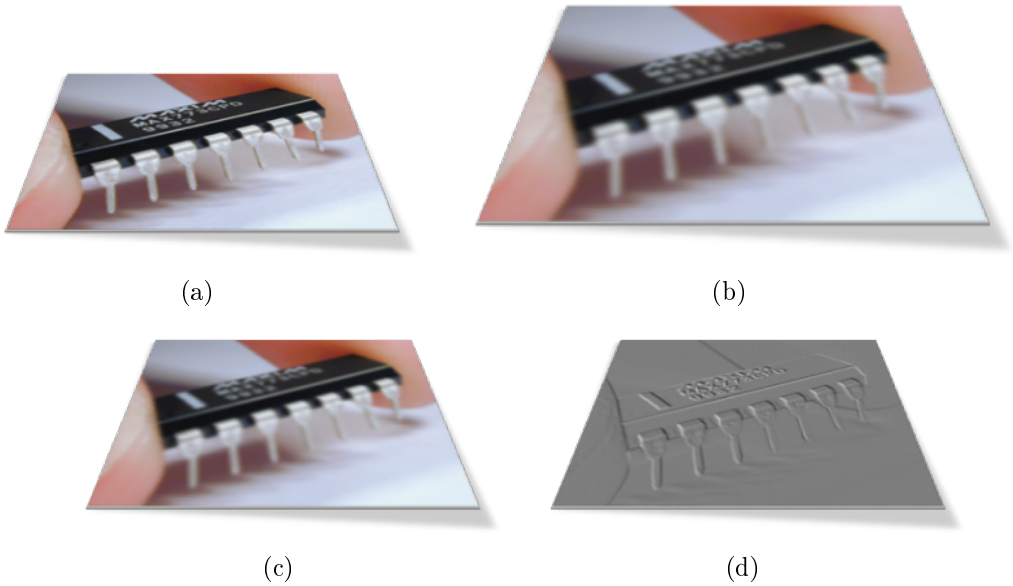
4.3.1 Upscaling process

In this section, simple interpolation algorithms are exploited in order to perform upsampling and downsampling procedures. The aforementioned operators perform the interpolation by *bicubic* algorithm. In order to go through the super-resolution procedure, the required images for the matching step must be prepared. This required pack of images consists of low resolution input image, up-sampled version of the input image and the high resolution components image. The explanation of these three images are given as following.

The first image, i.e. low resolution input image, is $L_1 = D(I_0)$ where D is the downsampling operator and I_0 is the original input image, respectively. It should be taken into account that instead of taking I_0 as the input image, L_1 is selected as the image during the computation process. This is due to achieving numerical results in addition to visual results. This issue has been widely discussed in the second chapter.

The second image, $L_0 = U(D(I_0))$, is the up-sampled version of the input image. According to the previous notation, the result of $D(I_0)$ shows the input image. Consequently, the $U(D(I_0))$ is upsampled version of the input image. U and D are the upsampling and downsampling operators, respectively.

The last image, which contains the high frequency components, is a representation of image details. In this image, details and edges are more visible rather than the main structure of the input image. Since details and edges of an image are embedded in high frequency components, these sort of images are called high frequency components images. High frequency components are easily obtained by subtracting the original image from the smoothed one. The original image usually consists of both low and high frequency information while the smoothed version of an image only represents the low frequency components. Therefore, the subtraction process results in preserving high-frequency components of the image. Figure 4.2 (d) is a good example of the mentioned scenario.



(a) (b)

(c) (d)

Figure 4.2: a) original image b)interpolated image c) low-Pass image d) high-pass image [26]

4.3.2 Matching

In this stage, the images obtained from upscaling procedure are used to construct the HR image. Henceforth, the interpolated image(L_0) is called the "target" image. In this step, missing high frequency components are recovered using different scales of the input image.

The matching process is a localized algorithm. It started by comparing each patch of the interpolated image, e.g. 3×3, with the corresponding ones in the LR image. The comparison is performed by computing the squared difference between two patches. Each patch of the target image is compared not only with the corresponding one in the LR image, but also with the adjacent neighbors of the LR patch. Thus, the local regions are examined within the LR image in order to find the closest patch to the target one. The differences between each patch in the target image and the ones in the LR image are stored in a matrix. Thereafter, for each patch in the target a matrix is created. This matrix represents the squared differences of a set of patches in the LR image with the target patch. The corresponding patch to the lowest value is chosen as the closest patch to the one in the target image. When the best patch in the LR image is determined, the corresponding one in the HF image is chosen and stored in a new matrix. This algorithm is iterated for all the patches in the interpolated image. Finally, the created HF matrix is added to the interpolated image and the HR image is reconstructed. Figure 4.2 illustrates the matching part as well as searching procedure. In this figure, red and green patches

represent HF and LF components, respectively. The dash-lined patch in the input image is the corresponding one in the upsampled image. However, it can be observed that the closest LR patch to the one in the interpolated image has been found in the adjacent neighbor. Consequently, the corresponding HR patch to the closest LR one is transfered as the recovered HF components of the patch in the interpolated image. Potentially, self-similarity method can be categorized as a fast algorithm. This is due to local searching of patches (instead of global searching) as well as not using an external data base. Following presents the above-mentioned algorithm.

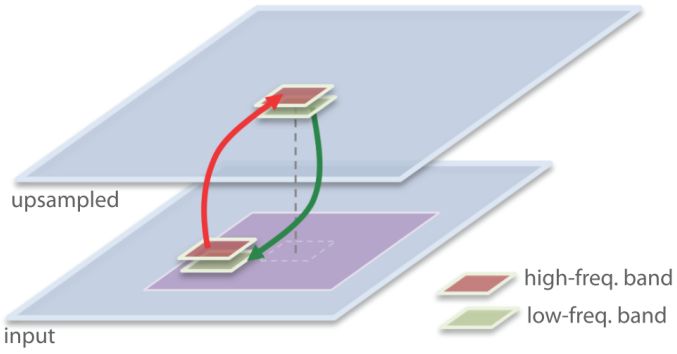


Figure 4.3: Matching Scheme [26]

Data: Input LR image $L_1(m, n)$, high frequency components image HF, Interpolated image $L_0(zm, zn)$, Zooming factor z, Size of patch w, Search Window 2w

Result: High resolution super resolved image I_T

Set I_0= Original Image, L_1=D(I_0) where D is the bicubic downsampling operator, L_0=U(D(I_0)) where U is the bicubic upsampling operator;

for *each pixel of* L_0 **do**

  for *every patch x in* L_0 *and corresponding one y in* L_1 **do**

    %Differences of patch x with the region in search window area.

    Diff=x-y(-2w:2w);

    S=Sum(Diff);

  end

  % Finding the closet patch and its location.

  [P1, P2]=min(s);

  %Finding corresponding HF patch of the closest LR patch

  HF_new=HF(P1,P2);

end

%Adding the HF patch to the interpolated patch

Reconstructed_Img=HF-new+L_0;

Algorithm 4: Reconstruction of high resolution images by self-similarity method

4.4 Modifications of the progress

There are several advantages by exploiting the proposed idea in [26], however in the wavelet interpolation part no reasonable results could be achieved. Hence, we have replaced the interpolation part with a simple interpolation method, e.g., bicubic. In addition, other changes performed in the whole implementation algorithm, as well. The proposed method in [26] was performed for a fixed size of patch (i.e. 5×5). In our experiments, we exploit the effect of patch size in reconstructing an image. Repeating the algorithm for different patch sizes showed that the patch size has a magnificent impact on the results. This issue comes more into attention by choosing the appropriate patch size. Therefore, it is more demanding to create a reconstructed HR image not only with a fixed patch size, but also with various patch sizes.

The most appropriate patch size is different for each given image. In other words, the magnitude of the patch size is changed based on the image content. Contents of the images not only change between different images, bus also vary within an image, as well. Thus, there is no fixed patch size candidate as the best one even for just a single image. Therefore, a switching operator can make the reconstructing HR images more accurate.

4.4.1 Switching operator

In this stage, a switching operator is applied to the algorithm in order to reconstruct the best recovered HR image. The operation starts as soon as resultant images reconstructed from four fixed patch sizes (i.e. 3, 5, 7 and 9) are achieved. Each patch of these reconstructed images is compared with the corresponding one in the original image (it is assumed that the ground truth image is available). Switching operator chooses the best patch size in each comparison by computing the mean of subtraction of each patch with the original one. The minimum value shows the closest patch to the original one. Hence, the corresponding HR patch to the closest one is selected to reconstruct the best image.

Numbers 1, 2, 3, 4 are assigned to patch sizes 3, 5, 7, 9, respectively. The corresponding number of the closest patch size is stored in a matrix at each reference. Eventually, these numbers create an image whose values are changed from 1 to 4 according to the contents of the input image alongside the selected patch. In this image, the main structure of the image along with the corresponding edges are

visible.

The selection of patch size in each comparison leads to find out the relation between the patch size and the content of the current patch. According to our experiments, switching operator selects the largest possible patch size in smoother area whereas the smaller patch size is chosen for regions with edges and discontinuities. The smaller patch sizes are able to recover more details in such regions (the useful information within the image might be lost if large patch size is selected). In the flattened areas it is not necessary to select the small patch. These areas can be properly recovered by even the large patch size.

4.5 Results

In this section, numerical and visual results of the algorithm are presented. Table 4.1 compares the PSNR values of the resultant images achieved by self-similarity algorithm and the bicubic one. The numerical results imply an improvement about 2dB in the PSNR value of self-similarity method (compared to the bicubic method).

Table 4.1: Lena and texture images PSNR values, Ratio=1.5 and Patch size=3

	PSNR values	
Images	Bicubic PSNR	Proposed method
Lena	35.74	37.23
Textured Image	35.03	37.43

Figures 4.4 illustrates the visual results of the images in Table 4.1. As it can be seen, images (b) and (e) are blurred while image (c) and (f) show more sharp edges and details. The self-similarity can be considerably successful compared to bicubic specially in small scaling factor (in this experiment scaling factor was 1.5).

In the next experiment, as it has been mentioned before, the effect of different patch sizes is investigated. Table 4.2 illustrates the fact that each patch size produces different result. For instance, patch size equal to 5 in both images has the best result among the other patch sizes. The changes in the PSNR values will give an overview about how various patch sizes might have different effects on the final results. In this table, the last column is related to our switching operator where the best PSNR can be achieved. Visual results of the Table 4.2 are given in Figure 4.5.

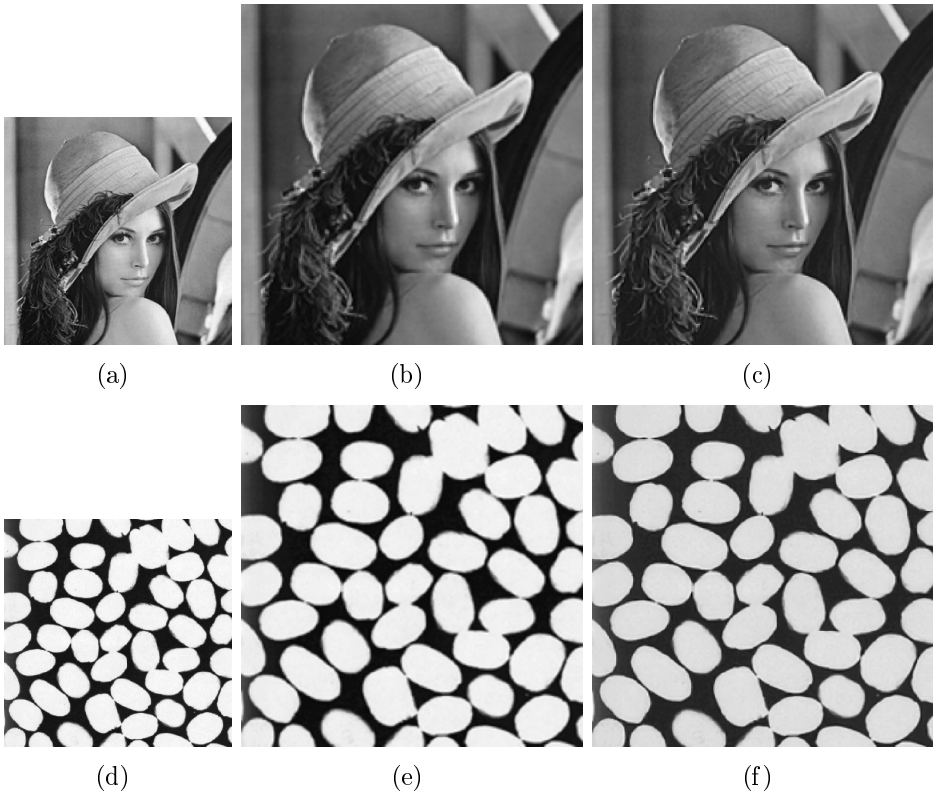


Figure 4.4: a)Lena: Original image b)Lena: Bicubic interpolated image c)Lena: Self-Similarity d)Textured: Original image e)Textured: Bicubic interpolated image f)Textured: Self-Similarity. Patch size=3, Ratio=1.5

Table 4.2: Self-similarity method. PSNR values of Lena and textured images, Ratio=1.5 and Patch size=3,5,7,9 and also the mixed one

	PSNR values				
Matching method	P=3	P=5	P=7	P=9	various patch size
Lena	37.23	37.64	37.39	37.17	37.88
Textured Image	37.43	38.92	37.74	37.35	38.29

As it can be seen from Figure 4.5, since the reconstructed images from various patch sizes are close to each other (in terms of quality) visual assessment would not be accurate. In this case, quantitative tools such as PSNR, helps to make a correct decision.

So far, all the experiments have been performed with a small scaling factor (e.g. 1.5). The input images were interpolated directly with the desired scaling factor. In order to explore the impact of this method in larger scaling factors, the experiments repeated. In Table 4.3, the results of reconstructing the image with scaling factor 3 are presented. In the first experiment, the scaling factor is applied directly and the image is interpolated 3 times. In the second experiment, small factors of 3

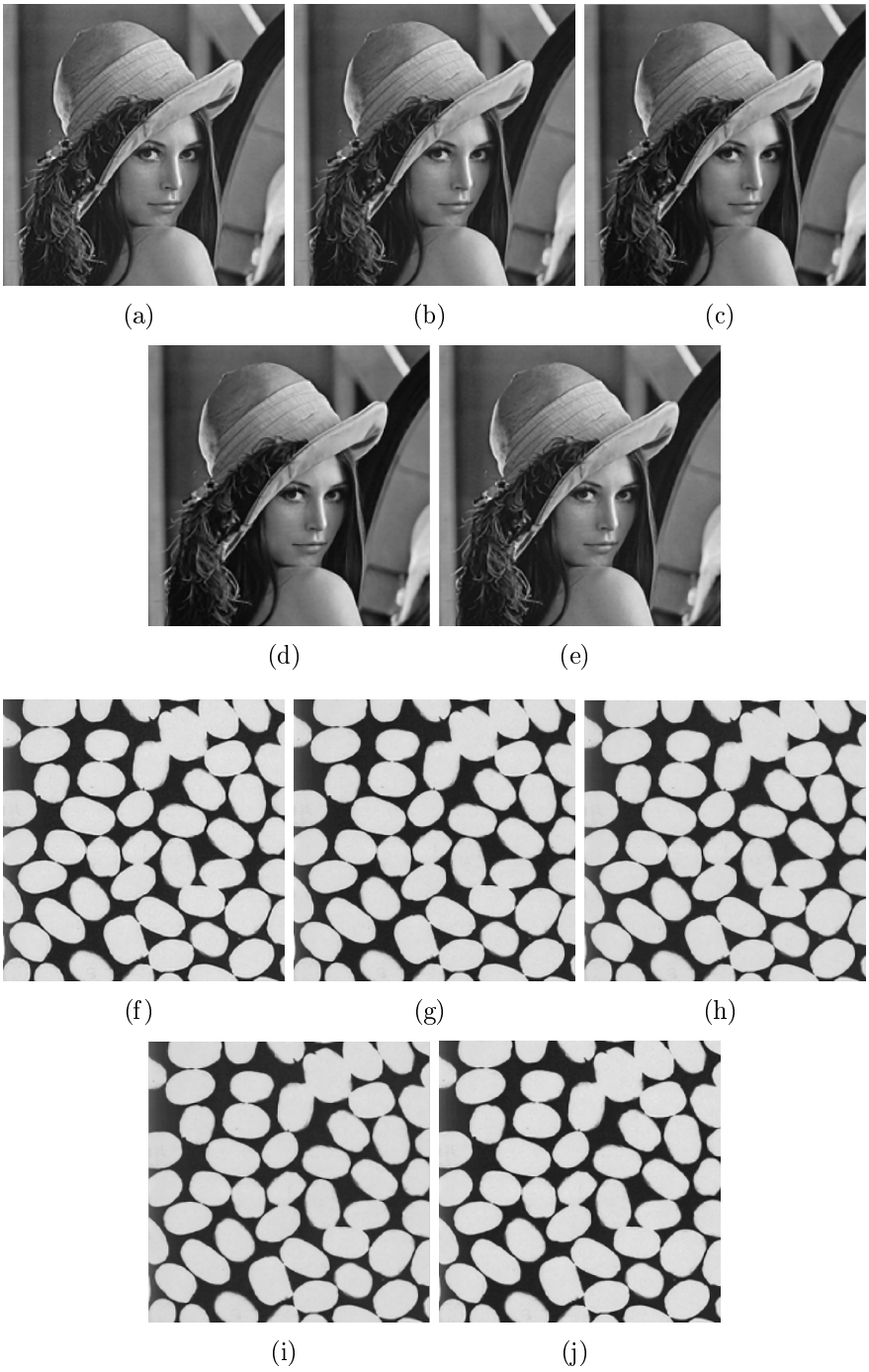


Figure 4.5: Self-similarity method a) Lena, patch=3 b)Lena, patch=5 c) Lena, patch=7 d) Lena, patch=9 e) Lena, various patch size f) Textured image, patch=3 g) Textured image, patch=5 h) Textured image, patch=7 i) Textured image, patch=9 j) Textured image, various patch size

are applied to the input image. In other words, the image is interpolated by small ratios, i.e., 4:3, 3:2 and 3:2, respectively. It can be seen that the proposed switching operator in both experiments has achieved the best results (last column). The same experiments have been performed for the textured image (see Table 4.4).

Table 4.3: Lena image. PSNR values, Ratio=3 and Patch size=3,5,7,9 and also the mixed one

	PSNR values				
Scaling types	P=3	P=5	P=7	P=9	Mixed Patch
Direct scaling	28.86	30.08	30.52	30.67	30.83
Small scaling	29.88	30.57	30.72	30.78	31.22

Table 4.4: Textured image. PSNR values, Ratio=3 and Patch size=3,5,7,9 and also the mixed one

	PSNR values				
Scaling types	P=3	P=5	P=7	P=9	Mixed Patch
Direct scaling	26.03	28.48	29.2	28.88	29.59
Small scaling	27.30	28.60	28.91	29.02	29.82

According to Tables 4.3 and 4.4, it can be observed that using small coefficients of a large scaling factor can lead to obtain better numerical results. This is due to enlarging an image in couple of phases instead of applying the scaling factor directly. This enlargement causes to preserve more details in interpolation procedure.

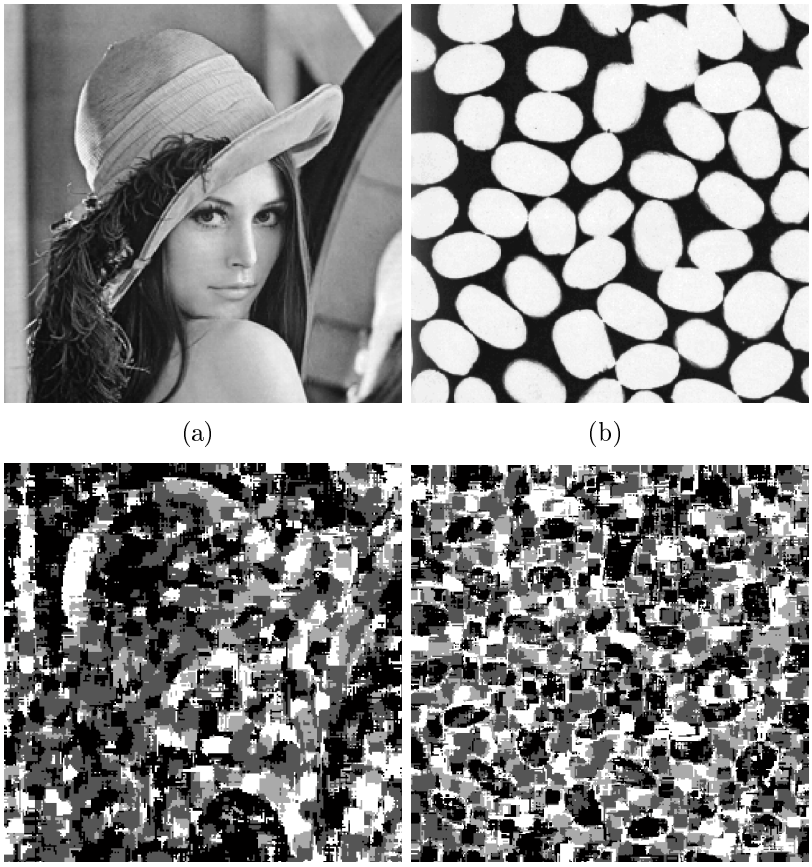
The visual results of Lena image for scaling factor equal to 3 are depicted in Figure 4.6. It can be seen that the image is suffering from some noises during the implementation phase. This noise is added to the image due to a use of large scaling factors. However, it is seen that Figure 4.6 (b) has been reconstructed better than (a) and (c).

(a) (b) (c)

Figure 4.6: Lena image a)bicubic interpolation b) self-similarity, various patch size, small coefficients of ratio 3 c) self-similarity, direct ratio 3, various patch

Figure 4.7 shows the images which are created according to the patch size selection of switching operator. These images have been achieved from the first experiment

for a zooming factor ratio equal to 1.5.



(a) (b)

(c) (d)

Figure 4.7: a)Original Lena Image b) Original Textured Image c) Image of selected patch size, Lena Image d)Image of selected patch size, Textured Image

A short comparison presented in Table 4.5 to observe the performance of self-similarity with other techniques presented in previous chapters. Different input images have been used in this experiment. Regression and self-similarity methods (as it can be found from previous chapters) are performed for different patch sizes. Consequently, the best achieved PSNR values (i.e. the one which is reconstructed from mixed patch sizes) for these methods are presented in the table.

Table 4.5: Lena image. PSNR values, Ratio=3 and Patch size=3,5,7,9 and also the mixed one

	PSNR values			
	Image 1		Image 2	
Methods	R=1.5	R=4	R=1.5	R=4
Freeman	22.53	22.06	24.24	23.12
Regression-based	39.2	28.33	33.95	27.20
Self-similarity	38.50	27.70	33.84	26.88

It can be seen that although the PSNR values of regression method for all images is higher than other methods, self-similarity can be considered as an appropriate SR method. In this method, despite the fact that there is no extra data base missing HF components have been recovered satisfactorily. The Freeman's method has the lowest performance between two others approaches.

4.6 Conclusion

Self-similarity as a single-image super resolution method has shown enough maturity to be employed for reconstructing an HR image. This method exploits the similarity among different patches within an image or through different scales of the image. Self-similarity such as other methods, has its own advantages and disadvantages. This method is applied to given images to recover the missing HF components. Self-similarity is dependent on the type of the input image, prior information, initial interpolation, etc.

The significant advantage of the self-similarity method is the fast computation time. In addition, the complexity of the algorithm is reduced compared to example-based method which has a huge dataset. The benefit of this local selection method is to reduce the amount of time spent to find the nearest patch. The small enlargement factors work better compared to the large factors in local self-similarity. This issue, in self-similarity method might be more sensible due to the nature of the algorithm. In this method, recovering an HR image is depending to the various scales of the

input image. Using large scaling factors lead to miss the resolution in the scales. As a result, in this situation this method cannot be useful.

It is worth mentioning that this method is more useful for the images with initial good quality. Self-similarity used the scaled version of the input image to estimate the missing HF components of the interpolated image. In order to recover the HF components, input image must contain more sharpness and high quality components. In other words, a low resolution input image does not contain HF components. Consequently, the scaled versions of a low resolution input do not include HF components, as well. Furthermore, this method might not work for a wide range of images since most of the available images are not high quality ones. Choosing a method for super resolving application must be performed by taking several conditions in to account such as input content, quality, desired scaling factor, etc. Moreover, one major issue which should be taken into consideration is that, the numerical results are not always a good measurement tool in assessing the performance of algorithms. On the other hand, visual results have an essential role in making a decision with regard to the performance.

5. DISCUSSION AND CONCLUSION

The ultimate goal of this thesis was to encourage studying super resolution as a demanding technique utilized in a variety of applications. Super resolution techniques involve image enhancement in terms of resolution. The SR attracted attentions of many researchers because of its critical role in medical applications, surveillance applications, etc. SR techniques are performed not only by signal processing approaches, but also by several hardware applications such as reducing the chip area, improving internal camera sensors, etc. Due to the constraints introduced in hardware side (such as limited data transferring rate, chip size, pixel size, etc.), they are not considered as the main candidates for SR methods. Moreover, generating an HR image with a desirable resolution using hardware improvements would not be efficient. Hence, signal processing techniques are considered as the main possible solution for SR applications.

This thesis is comprised of several SR methods where the input is just one image (SISR). One of the SISR approaches is exampled-based methods where an external database assists the algorithm to find the missing HF components. These methods are strongly dependent to the external dataset. However, they are partially successful in SR approaches specially when there are no more observations obtained from a scene. The first and the most significant disadvantage of example-based methods is the high probability existence of irregularities and noise introduced within the resultant image. This problem is happening due to using huge training dataset which might be included to the input due to the plenty of irrelevant images.

The regression-based methods are another technique in SISR approaches. These methods are also demanding due to the capability of finding an appropriate relationship function between LR and HR patches. Many researches have focused on regression-based methods which were successful in recovering the missing HF components. In these methods, the impact of large training dataset is a dominant shortcoming. The main concerns in regression-based methods are generating an efficient mapping function, extracting parameters impacts when changing, finding the appropriate parameters for each input, etc.

Eventually, self-similarity were investigated as the third group of methods in SISR within this thesis. These methods introduced an impressive novel technique in super resolution by omitting dependencies to the training dataset (not applicable in all methods). Self-similarity methods, can resolve the HR image by exploiting similarities between different patches irrespective to the external dataset. These techniques create a variety scales of the input image. Afterwards, they recover missing HF components of interpolated version of the input image either from different scales or from various regions in one scale. These techniques work efficiently as long as the input image contains adequate resolution. This is due to the nature of the techniques which employ HF components obtained from different scales of the image. Hence, if the input image suffers from low quality, these methods can not be successful further.

As the final words, presenting a method as the best candidate in SR is not possible. There is always a trade-off between different SR approaches, each of which introduces a new constraint to the corresponding method. Choosing an appropriate technique to have satisfactory results depends on several factors. Initial quality of the image, type of the image content, desired zooming factor, the application of HR image (i.e. in which field is going to be used), cost, time, etc. are factors which should be taken into consideration while choosing the appropriate super resolution technique.

REFERENCES

[1] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," Adv. Comput. Vis. Image Process., vol. 1, pp.317-339, 1984.

[2] Sung Cheol Park, Min Kyu Park, Moon Gi Kang, "Super-resolution image reconstruction: a technical overview", Signal Processing Magazine, IEEE , vol.20, no.3, pp.21,36, May 2003

[3] Jianchao Yang, Thomas Huang, "Image super-resolution: Historical overview and future challenges", In "Super-resolution imaging", CRC Press, 2010.

[4] "How to extract high freqency and low frequency component using Bilateral filter?", [WWW], Signal Processing beta, [Accessed on 09.11.2014], Available at http://dsp.stackexchange.com/questions/6452/how-to-extract-high-freqency-and-low-frequency-component-using-bilateral-filter

[5] "Shot Noise",[WWW], RP Photonics Encyclopedia, [Accessed on 09.11.2014], Available at http://www.rp-photonics.com/shot_noise.html

[6] Freeman, W.T.; Jones, T.R.; Pasztor, E.C., "Example-based super-resolution," Computer Graphics and Applications, IEEE , vol.22, no.2, pp.56,65, Mar/Apr 2002

[7] Changhyun Kim; Kyuha Choi; Jong Beom Ra, "Example-Based Super-Resolution via Structure Analysis of Patches," Signal Processing Letters, IEEE , vol.20, no.4, pp.407,410, April 2013

[8] Jianchao Yang; Zhe Lin; Cohen, S., "Fast Image Super-Resolution Based on In-Place Example Regression," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on , vol., no., pp.1059,1066, 23-28 June 2013

[9] Taniguchi, K.; Ohashi, M.; Xian-Hua Han; Iwamoto, Y.; Sasatani, S.; Yen-Wei Chen, "Example-Based Super-Resolution using Locally Linear Embedding," Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on , vol., no., pp.861,865, Nov. 29 2011-Dec. 1 2011

[10] Jianchao Yang; Wright, J.; Huang, T.S.; Yi Ma, "Image Super-Resolution Via Sparse Representation," Image Processing, IEEE Transactions on , vol.19, no.11, pp.2861,2873, Nov. 2010

[11] Jian Sun; Jian Sun; Zongben Xu; Heung-Yeung Shum, "Gradient Profile Prior and Its Applications in Image Super-Resolution and Enhancement," Image Processing, IEEE Transactions on , vol.20, no.6, pp.1529,1542, June 2011

[12] Hong Chang; Dit-Yan Yeung; Yimin Xiong, "Super-resolution through neighbor embedding," Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on , vol.1, no., pp.I,I, June 27 2004-July 2 2004

[13] Baker, S.; Kanade, T., "Limits on super-resolution and how to break them," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.9, pp.1167,1183, Sep 2002

[14] Kwok-Wai Hung; Wan-Chi Siu,, "New motion compensation model via frequency classification for fast video super-resolution," Image Processing (ICIP), 2009 16th IEEE International Conference on , vol., no., pp.1193,1196, 7-10 Nov. 2009

[15] Yi Tang; Hong Chen, "Matrix-value regression for single-image super-resolution," Wavelet Analysis and Pattern Recognition (ICWAPR), 2013 International Conference on , vol., no., pp.215,220, 14-17 July 2013

[16] Bill Freeman ; Ce Liu, "Markov random fields for super-resolution and texture synthesis", Advances in Markov Random Fields for Vision and Image Processing book, MIT Press, 2011

[17] Aggelos K. Katsaggelos; Rafael Molina ; Javier Mateos,"Super Resolution of Images and Video," Synthesis Lectures on Image, Video, and Multimedia Processing,Morgan and Claypool Publishers,2007

[18] Chih-Yuan Yang; Jia-Bin Huang; Ming-Husan Yang, "Exploiting Self-Similarities For Single Frame Super-Resolution", Proceedings of Asian conference on computer Vision (ACCV 2010),pp.1807-1818,Nov,2010

[19] K.I. Kim; Y.Kwon, "Exampled-based learning for single-image SR and JPEG artifact removal", Technical Report 173, Max Planck Institute, Aug, 2008

[20] Hou, H.S.; Andrews, H., "Cubic splines for image interpolation and digital filtering," Acoustics, Speech and Signal Processing, IEEE Transactions on , vol.26, no.6, pp.508,517, Dec 1978

[21] Xin Li; Orchard, M.T., "New edge-directed interpolation," Image Processing, IEEE Transactions on , vol.10, no.10, pp.1521,1527, Oct 2001

[22] Noriaki Suetake; Morihiko Sakano; Eiji Uchino, "Image super-resolution based on local self-similarity," Optical Review, vol.15, pp.26,30, 2008

[23] Glasner, D.; Bagon, S.; Irani, M., "Super-resolution from a single image," Computer Vision, 2009 IEEE 12th International Conference on , vol., no., pp.349,356, Sept. 29 2009-Oct. 2 2009

[24] Jianchao Yang; Wright, J.; Huang, T.; Yi Ma, "Image super-resolution as sparse representation of raw image patches," Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on , vol., no., pp.1,8, 23-28 June 2008

[25] Fei Zhou; Wenming Yang; Qingmin Liao, "Interpolation-Based Image Super-Resolution Using Multisurface Fitting," Image Processing, IEEE Transactions on , vol.21, no.7, pp.3312,3318, July 2012

[26] Gilad Freedman; Raanan Fattal, "Image and Video Upscaling from Local Self-Examples," ACM Transactions on Graphics, vol.30, pp.342,351, 2001

[27] Sae-Jin Park; Oh-Young Lee; Jong-Ok Kim, "Self-similarity based image super-resolution on frequency domain," Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific , vol., no., pp.1,4, Oct. 29 2013-Nov. 1 2013

[28] Wan-Chi Siu; Kwok-Wai Hung, "Review of image interpolation and super-resolution," Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific , vol., no., pp.1,10, 3-6 Dec. 2012

[29] William T. Freeman; Egon C. Pasztor; Owen T. Carmichael, "Learning Low-Level Vision", Int. J. Comput. Vision 40, 1, Oct, 2000

[30] A. Hertzmann; C. E. Jacobs; N. Oliver; Brian Curless; D. H. Salesin, "Image analogies", In Computer Graphics (Proc. Siggraph 2001), P. 327?340, New York, 2001

[31] Jing Hu; Yupin Luo, "Single image Super Resolution Based on Local Regression and Nonlocal Self-Similarity", J. Electron. Imaging, May,2014.

[32] K. I. Kim; D. H. Kim; J.-H. Kim, "Example-based learning for image super-resolution", In Proc. the third Tsinghua-KAIST Joint Workshop on Pattern Recognition, P. 140?148, 2004

[33] K. Ni; T. Q. Nguyen, "Image superresolution using support vector regression", IEEE Trans. Image Processing, P.1596?1610, 2007

[34] Shengyang Dai; Mei Han; Wei Xu; Ying Wu; Yihong Gong, "Soft Edge Smoothness Prior for Alpha Channel Super Resolution," Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on , vol., no., pp.1,8, 17-22 June 2007

[35] "Pixel",[WWW], TechTerms.com, [Accessed on 18.11.2014], Available at http://www.techterms.com/definition/pixel

[36] Stphane Mallat, "A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way", Academic Press, 2008