Doctoral Thesis

Example-based Face Image Enhancement by Taking Correspondence of Facial Parts into Consideration

September 2019

Doctoral Program in Advanced Electrical, Electronic and Computer Systems
Graduate School of Science and Engineering
Ritsumeikan University

MUHAMMAD SUHAIL BIN MUHAMAD HAMDAN

Doctoral Thesis
Reviewed by Ritsumeikan University

Example-based Face Image Enhancement by Taking Correspondence of Facial Parts into Consideration

（顔の部位の対応を考慮したサンプルベースの顔画像の高精細化手法）

September 2019
2019 年 9 月

Doctoral Program in Advanced Electrical, Electronic and Computer Systems
Graduate School of Science and Engineering
Ritsumeikan University

立命館大学大学院理工学研究科
電子システム専攻博士課程後期課程

MUHAMMAD SUHAIL BIN MUHAMAD HAMDAN
ムハマド　スハイル　ビン　ムハマド　ハムダン

Supervisor: Professor IZUMI Tomonori
研究指導教員 ： 泉 知論 教授

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The thesis presents and investigates an approach for face image enhancement based on a machine learning technique utilizing facial image examples.

Faces are one of the main concerns in the image applications, especially in the security system such as a surveillance camera. There are cases that a face image of interest has unsatisfactory quality by some factors as the distance to the camera, the luminance environment, the motion speed of the target, noises, data compression, and so on. Thus, the enhancement of the image quality from a low-resolution degraded image is one of the key issues in the field of image processing.

A conventional approach in common to generate a higher-resolution image is interpolation using Lanczos (sinc). However, these analytic approaches suffer from a blurred appearance. Super-resolution (SR) techniques have emerged to resolve these problems by inferring detailed information on the missing texture. One of the major approaches in super-resolution is example-based method, which utilizes example images as a database to estimate the missing texture. Freeman et al. have proposed a novel example-based super-resolution method where example images are used to generate pairs of high-frequency (HF) and low-frequency (LF) components of the images and they are divided into patches of the image to compose a database of the relationship between low-resolution (LR) appearances and high-resolution (HR) textures. Given an LR image, a Markov Random Field (MRF) network is applied to find the best-fit HF patches for the LF component of the given image. Then the given image and the HF patches are combined to reconstruct an HR image.

The thesis introduces an idea specific for human faces to the Freeman's method, restoring each part of the image with patches corresponding to each facial part. The database composed patches with information of their original positions, and the distance between the targeted input patch and training patch original positions is incorporated to the compatibility function for patches matching process. An experiment on a set of facial images demonstrated that the proposed method achieved the best quality of 30.39 [dB] in terms of the peak signal-to-noise ratio (PSNR) compared to the previous' quality of 29.65 [dB]. Further analysis is conducted to determine appropriate parameters. An experiment on patch-databases

with different image scaling factors and different patch sizes revealed that the best patch size is interestingly adaptable to scaling factor, n and determined by $(2n+1)^2$.

The presented approach is also applied to noise reduction, especially for the block noise in highly compressed image data. The database is reconstructed by replacing LR-HR patch pairs with pairs of degraded (compressed) face images and the corresponding original images, and two variations of the image enhancement methods are presented. One method results in a 3.2 [dB] improvement in terms of PSNR on average for a very low quality rate of 1% (around 120:1 compression rate), while the conventional Gaussian-filtering method results in a 2.5 [dB] improvement. Another method yields more natural images and better PSNRs for images when the quality rate is around 5%.

An improved patch-searching algorithm is also presented which restricts the searching area nearby the targeted facial parts but searches patches in more detailed to select the patch candidates. An experimental result shows the improved version achieves better or comparable results while reducing by 90% computational time compared to the previous.

ACKNOWLEDGEMENTS

To my beloved parents, many thanks!

First and foremost, I would like to acknowledge that my PhD study in Ritsumeikan University, Japan was sponsored by the *Majlis Amanah Rakyat (MARA)*, under supervision of *Universiti Kuala Lumpur (UniKL)*, Malaysia.

I would like to thank my dedicated supervisors Professor Hironori Yamauchi, Professor Tomonori Izumi and Professor Yohei Fukumizu: without their valuable guidance, excellent cooperation and dedicated involvement in every step throughout the process, this thesis would have never been accomplished. I am very grateful for their great deal of encouragement and patience, tremendous support and assistance in overcoming numerous obstacles I have been facing through my research.

My sincere thanks to the examiners of my oral presentation (PhD Defense), Professor Takakuni Douseki and Professor Takeshi Kumaki for raising a number of interesting points of discussion and gave insightful comments that helped in addressing some of the shortcomings in the thesis.

I would also like to express my appreciation to Ritsumeikan University staffs especially from *Graduate School of Science and Engineering* office and *International Center*: with their assistance and support, I had smoothly furthered my study in Japan and had great experience of Campus life.

Special thanks to *MADANi* – a Malaysian student association in Japan, and *Doctorate Support Group (DSG)* Facebook Group: their sharing about successes and challenges in PhD journey, useful information and advice, have given me strength to keep striving and never give up.

Last but not the least, I would like to thank my family and friends for giving me the strength and moral support throughout writing this thesis and my life in general.

Thanks for all your encouragement!

CHAPTER 1

INTRODUCTION

1.1 Background

Installations of surveillance cameras' or CCTVs (closed-circuit television) have been widely used to fight crimes. Such installations are often used for policing in areas that may require monitoring like commercial areas, such as banks, airports, and convenience stores. Even nowadays it is common to notice that surveillance cameras installed in some residential areas, as a way of providing an additional protection and deterrent to potential criminals from entering. It appears that surveillance cameras act as a powerful investigative tool for many types of crime (Ashby, 2017). Information from the recordings of crime footage, especially those of criminals' faces and figures, can be of help to the police to identify suspected criminals or obtain clues for investigations, which may increase chances of crimes being solved.

Nevertheless, unfortunately most surveillance cameras involve long-term recording, resulting in low-resolution and highly compressed images with low frame decimation. Even though advance types of surveillance cameras that able to record footage in high definition are available recently, i.e. IP security camera, most places still adopting the standard ones, i.e. analog security camera systems, since they are less expensive and easier to maintain.

The quality of a standard security camera's footage is often too poor to be used in investigations since less information could be obtained. It is worthless if the obtained images of criminal suspect's face are unidentifiable due to poor quality. In fact, according to a publication of the National CCTV Strategy by The Joint Home Office ACPO in UK, "*anecdotal evidence suggests that over 80% of the CCTV footage supplied to the police is far from ideal, especially if it is being used for primary identification or identities are unknown and identification is being sought, for instance, by media release*" (Gerrard et al., 2007, p 12).

Thus, demands for image enhancement from surveillance fields are remarkably high. In crime investigation especially, police officers or detectives are often facing difficulty to identify a face image of a person of interest. The recording degraded

footages are usually sent to technicians or specialists for enhancement process, hoping that a desired quality of image could be restored. Until now, researchers especially in digital image processing majors have been motivated to explore various techniques of image enhancements and been developing numerous approaches to satisfy such demands.

Figure 1.1 Demands from surveillance fields

As digital image enhancement technology has rapidly advanced, recent years have witnessed remarkable progress of image super-resolution techniques. In the old days, since super-resolution involves complex algorithms, in-depth analysis of data, and utilization of huge number of images, for instance, multi-frame super-resolution and dictionary learning strategy, researchers had been struggling with the lack performance of computer available. But nowadays, along with rapid progress on the development of computer performance especially in terms of speed and memory, super-resolution has become the most common task that widely used in many fields, including medical imaging (Greenspan, 2009), face identification (Gunturk et al., 2003), remote sensing (Shen et al., 2009), license plate recognition (Seibel et al., 2017) and microscopy (Quan et al., 2010).

There have been extensive researches on vary approaches in super-resolution, which lead to an increase of number of proposed methods. Each method has different perspective and objective, with different framework of algorithms and adopted means. Therefore in general, methods of super-resolution have been classified into two types of processes: single-image and multi-frame. For the single-

image super-resolution, available techniques are basically categorized into: 1) learning-based and 2) reconstruction-based approaches, which will be elaborated later in Section 1.3.

Our study will be specifically focusing on face image enhancement for better identification, which is the most crucial theme in surveillance fields besides vehicle license number plate recognition. Although there are several researches concerning facial image, yet there is little research on face image enhancement that utilizes information of facial parts. In this thesis, a study on how a facial image is enhanced by taking facial parts into account is presented, and performance of our proposed methodologies is discussed.

1.2 Research Problems

Some of the main causes that lead to a degradation of footage start from the cameras themselves. Camera's quality, distance to the subject, angle and position are important factors that affect the quality of footage. Those footages are recorded in digital video recorder (DVR) for future reference, however due to storage constraint the footages have to be compressed. Because of those limitations involved in the surveillance systems, stored face images are usually small in size, worsened by block artifacts due to image compression, blurry, low illumination and so on. Even if we manually zoom in for a closer look, the face image is only going to get blurrier and grainier, which make it difficult to be identified.

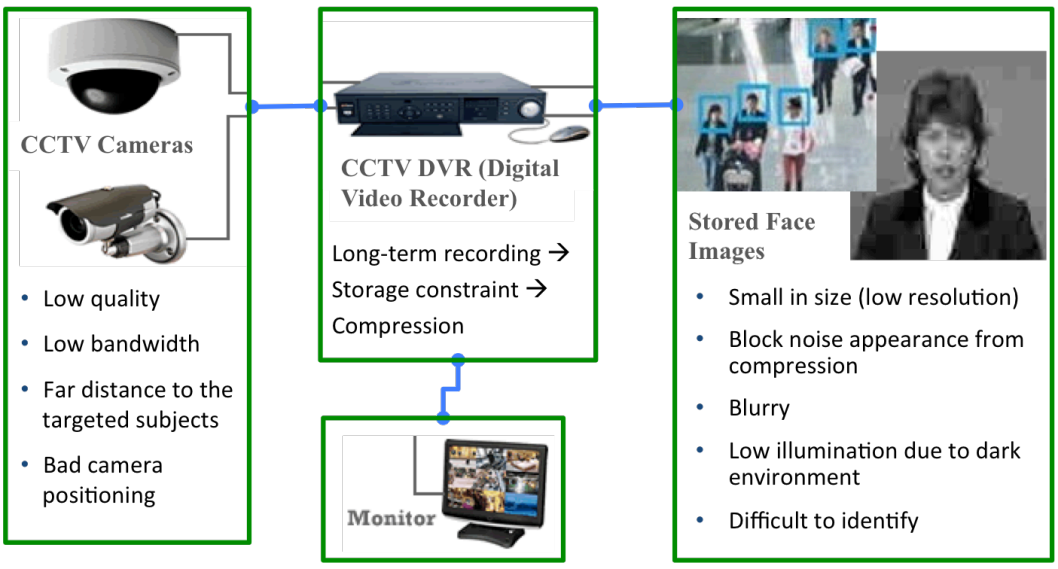


Figure 1.2 Typical surveillance systems

It is essential to process a degraded image from surveillance systems and enhance their quality for the sake of investigations. Enhancement of an image could be achieved by image transformations, such as making the image larger i.e. high-resolution, and clearer i.e. lesser noise and sharper texture, to obtain legible information from the shapes, features, or even small details of the subject of interest, such as face image or vehicle license number plate.

In image processing, there are already a number of conventional methods for image enlargement such as Bilinear, Bicubic, or Lanczos interpolation, which is commonly used to infer a more legible high-resolution image from a low-resolution image. Generally, almost every geometric transformation like translating, rotating, scaling, and warping requires interpolation to be performed on an image. However, these analytic approaches typically suffer from a blurred appearance due to perceived loss of detail in textured regions, causing unsatisfying quality in enlarged images. The perceived quality of the interpolated images is affected by several issues such as sharpness of edges, freedom from artifacts and reconstruction of high frequency details etc. (Ruikar and Doye, 2012).

On the other hand, to effectively reduce noise or block artifacts from a degraded image, point-spread-function (PSF) or smoothing filters are commonly used (Lai et al., 2005). The stronger the filter is, the lesser the noise. Footages from surveillance camera are often distorted by excessive block noise because of highly compression process in the systems. So, we need to apply a stronger filter for it to be effective. However, consequently the texture component in image such as sharp edges is inevitably lost together with noise component due to the filtering process, which yields a corrupted blurred image.

We can conclude that both conventional methods for image enlargement and noise reduction have a same common problem, which is the unavoidably missing of texture from the image after such process. Therefore, the missing texture in the enlarged or smoothed (de-noised) image has to be retained somehow, or estimated accurately. Here, super-resolution (SR) techniques can be employed as significant approaches to resolve these problems, where they not only enhance image resolution or reducing noise, but also able to predict detailed information on the missing texture to define a reconstructed image.

For face images in particular, enhancing the apparent sharpness of face features, i.e. eyes, nose and lips, is necessary for a better identification. Even though there are

several approaches that have been proposed specially for face image enhancement (Baker and Kanade, 2002), enhancing a face image by facial parts is an interesting technique that is not extensively explored, yet. Our research will adopt this idea in the super-resolution algorithms.

1.3 Research Scope and Approach

Numerous algorithms that have been recently employed for image super-resolution can be basically categorized into two kinds of processes, which as shown in Figure 1.3: single-image and multi-frame. Super-resolution for single-image has two different approaches, one is a learning-based and the other one is reconstruction-based. The reconstruction-based methods are highly dependent on two properties: (1) the registration of different input images, and (2) the point spread function (PSF). We have to reasonably make a prior assumption on both imaging conditions to obtain better gradient details, which is very constraining and unstable, inevitably bring to a poor performance as zoom factors increase. The resulting image either lacks details or produces overly smooth results.

Learning-based methods or also known as example-based methods, on the other hand, utilize external information from a set of training low-resolution (LR) and high-resolution (HR) image pairs, to generate a desired HR image. Patches within the input LR image are generally extracted and searched within the training set to estimate suitable or matched HR patches, by adopting a learning model. The example-based methods are best for specific applications, such as face super-resolution, since it typically uses specific kinds of images as its dictionary.

From a bigger point of view, there is another advanced approach that has been widely adopted in image processing recently, called deep learning. Compared to the example-based method, the deep learning approach provides better enhancement of image, nevertheless it involves much more complex algorithms and process that requires bigger size of data to perform well. This would cost considerably longer execution time, need heavy memory and require high-end machines for the process. Therefore, for a standard face image quality enhancement to assist identification process in crime investigation, the example-based technique is sufficient.

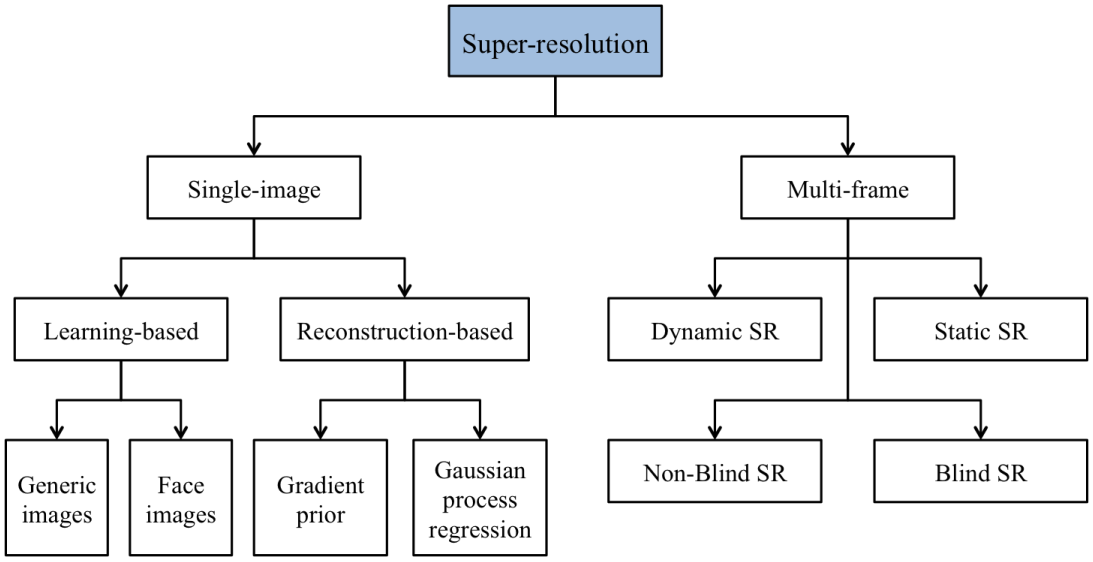


Figure 1.3 Classification of the super-resolution algorithms

Our research will be focusing on single-face-image enhancement, particularly for image enlargement and noise reduction process. It is appropriate to employ example-based approach instead of reconstruction-based approach, since the subject used in this research is specific kind of images, i.e. human faces. There are a lots of methods have been proposed using example-based algorithms so far. Generally, the main different between those proposed methods are, the different combination of different feature extraction method and types of learning model (this will be discussed in Section 3.2). Recent researches regarding example-based are competing on which combination would provides better performance. However, our works do not concern over those matter.

In present research, we implemented a novel strategy specifically for face image enhancement, where we take facial parts into account in super-resolution process. We studied the Freeman et al.'s proposal of example-based methods, then we improvised and extended the algorithms especially in terms of learning-model and database construction, that correspond to facial parts in face image. The original idea was to typically gather and classify training image patches into multiple databases based on facial parts, i.e. eye database, nose database and lips database, so that we can enhance a given face image part by part. For instance, when the face region which that we are enhancing is around the part of eye, patches from eye database will be utilized to estimate output patches. However, instead of constructing multiple databases, we proposed a distance function to determine which facial part is the patch from, and incorporate it with Freeman et al.'s learning-model functions.

1.4 Research Motivation and Goals

Have you ever seen a scene from an episode of CSI (a TV show), where two detectives are sitting together analyzing surveillance footage, and suddenly one of them says: “Zoom in! Let us see if we can get a reflection of her eye.”, and then the other detective simply make a magnification by 100 times, “Enhance it!”, and then suddenly, miraculously the small and grainy eye becomes bigger and crystal clear, that even reflects a precise object of what the suspect is seeing? Well obviously, that is a total fiction. In reality, that small image of eye from an analog security camera is only going to get blurrier and grainier when you zoom in for a closer look.

The Murder of Nurin Jazlin, is one of the most unforgettable yet horrific and heart-breaking case in Malaysian history. Nurin was the 8-year-old girl who was abducted and murdered over ten years ago, which shocked the nation. Despite a decent length of recording visual of the unidentified suspect, images taken from surveillance camera were unable to provide sufficient information for face identification. The footage was too blurry to make out a valid license plate number on the motorcycle. The lack of evidence is the reason for this case remains unsolved.

Ten years later, another shocking event took place in Kuala Lumpur International Airport, which was the Assassination of Kim Jong-Nam, who is the eldest son of deceased North Korean leader. The involved suspects were successfully identified thanks to the recent development of image enhancement technology. Investigation on this case had smoothly progressed since the murder happens at an open public or commercial place that has advanced surveillance systems installed, unlike the first case.

Following the two specific events, I have been motivated to get involve in the major of study. There are many things that had been a fiction before, have became reality today. We believe the extensive research on this theme, is the least we can do to contribute indirectly in advancing image enhancement technology. There are still a lot of room for improvement and demands for the benefit of the people, which have been a challenge and motivation to researchers in related fields.

Research Main Goal

Our goal is to enhance degraded face image particularly in image enlargement and noise reduction. We have previously explained the common disadvantages of conventional methods, which lead to a very same problem – missing of texture

details from the after-processed image. Our aim is to restore texture of facial image by taking into consideration their corresponded facial parts, which could not be done by conventional methods alone. Although prior research (Freeman et al., 2002) has significantly restored the missing texture by its proposed learning model, enhancing such face image by taking facial parts into account is a new approach. Most related studies that have been proposed onwards only consider the use of different ways of texture extraction or different types of learning model as improvisation. For now, learning-based approach using a learning model that takes correspondence of facial parts into consideration has not been proposed yet.

In this thesis, a proposal of a new methodology based on example-based approach, which incorporate a facial parts compatibility function into the learning model is presented. We executed the proposed learning-model onto two different cases under two different main objectives, which generally are:

Research Objective 1

To enhance apparent of sharpness of texture in a single enlarged face-image by restoring the missing texture, which cannot be done by conventional methods alone.

An enlarged image by a conventional enlargement method such as Lanczos interpolation usually suffered from blurred appearance due to the loss of texture details or high-frequency component. In order to reduce the blurry effect, we need to estimate missing texture of the enlarged image. Training patches database is constructed with a collection of enlarged face images in pair with their original HR images. We aims to enhance an enlarged face image's texture by using a learning model that observed their texture patch by patch, and facial-parts by facial-parts, in example-based super-resolution algorithms.

Research Objective 2

To reduce block noise in a compressed degraded face-image by estimating new texture details and restore distorted texture especially around face features.

When an image is compressed, texture in the image is distorted. Even if a point-spread-function is applied on such degraded noised image, the process only smooth or reduce the excessive noise, but not restoring the crucial texture. We implement the same idea, i.e. taking facial parts into account, but instead of using enlarged images as dictionary, we utilized a collection of compressed degraded image in

replace. The methodology is constructed to regenerate distorted texture in the degraded image especially on the crucial facial parts like eyes part.

In addition, we aims to analyze the proposed algorithms and improve them in terms of processing time and parameter setting for better performance. The experiments involve quantitative assessment. Most of the assessments are based on peak signal-to-noise ratio (PSNR), an error metric that is commonly used to measure image quality. Resulting face images from our proposed method is assessed and compare with Freeman et al.'s method and conventional methods, i.e. Lanczos interpolation method in image enlargement experiment and Gaussian filtering method in noise reduction experiment.

Hopefully this study will open a new dimension in learning-based super-resolution techniques and fill the gap of the study, where we not only use the same kind of images in database, i.e. a collection of face images, but also further considered facial part in patch matching process. Since noise and blur are major problems in image processing field, we highly expect this research would provide significant contributions especially in face image enhancement and identification.

1.5 Thesis Structure

The thesis is composed into seven chapters. Chapter 1 briefly elaborated the background of our study, field of study with the research objectives and contributions. Chapter 2 described the conventional methods of image enlargement and noise reduction, along with their common problems and disadvantages. Chapter 3 explains the theory of learning-based or also well known as example-based methods, which is one of two majors of single image super-resolution techniques. We then introduce our proposed methods: their algorithms, experiments, evaluations and discussions, in the next three chapters, where the proposed methods for enlarged image enhancement, compressed degraded image enhancement, and improved algorithms, are presented in details in Chapter 4, 5 and 6, respectively. Research Objective 1 is assessed in Chapter 4, while Objective 2 is assessed in the following Chapter 5. In the last chapter, the contributions and limitations of the study are reviewed, and the recommendations for how the framework should be further developed are provided.

CHAPTER 2

IMAGE DEGRADATION AND ENHANCEMENT

Degraded images may lead to failure of further analyses in many fields, including security surveillance, medical applications, object recognition, and video conversion. For example, recorded face images from closed-circuit televisions (CCTVs) or surveillance cameras are usually small and noisy due to the low quality of the cameras themselves and are also influenced by compression process due to memory constraint, making it very difficult to obtain any crucial information from the images. Thus, it is necessary to enhance the image in advance by enlarging them into larger and more legible images, as well as reducing noise in the image.

In digital image processing, image enhancement has been highly demanded especially for image analysis purpose including image enlargement, noise reduction, sharpening and transformation. In this chapter, conventional enlargement methods and noise reduction technique, which are directly related to the objectives of our study, are briefly described.

2.1 Image Resampling

Image resampling refers to the resizing of a digital image, involving transformation of image resolution. Image resizing is necessary when we need to up-sampling or down-sampling, i.e. increase or decrease the total number of pixels in image. In digital image processing, low-resolution image commonly refers to the image that has a small amount of pixels or small in size, while high-resolution image refers to the image that has a large amount of pixels. Therefore, up-sampling process for example, is an enlargement process that transforms a low-resolution image into a high-resolution image, where the number of pixels is increased more than the original ones. Here, comes the problem on how to estimate the unknown value for new pixels in the high-resolution image.

For instance, when we want to up-sample a low-resolution image's height and width by two times respectively, we need the new values information for 4 times of pixels number than the original image to generate an output high-resolution image. In digital imaging, image interpolation techniques are commonly used in order to

determine those unknown values of new pixels. In this section, we will explain the algorithm for two of the frequently used methods of resampling, Bilinear and Lanczos interpolation.

2.1.1 Bilinear Interpolation

Bilinear interpolation is one of the most basic resampling operators other than nearest neighbor and cubic convolution, for generating high-resolution image from a low-resolution image. The term bilinear refers to the performing of interpolations in two dimensions (horizontal and vertical), which is an extension of linear interpolation.

When a low-resolution image needs to be enlarged, each pixel of the image has to be moved in a certain direction based on the scaling constant. However, when scaling up an image by a non-integral scale factor, there are new pixels i.e. holes that are not assigned appropriate pixel values. In this case, those holes should be assigned appropriate RGB (or color) or grayscale values so that the output image does not have non-valued pixels.

The method assigns a new value to the unknown pixel by taking the distance-weighted average of the four nearest known pixel values. The key difference in bilinear interpolation is that it uses values of only the four nearest pixels, located in diagonal directions from a given pixel, to generate an appropriate output value of that pixel.

As shown in Figure 2.1, the bilinear operator is applied to the surrounding four pixel nodes (gray dots), Q(.) to infer the new value of the point (red dot) at which we want to interpolate, P(.). Let (i,j) and (x,y) are pixel coordinate or position in low-resolution (LR) image, Q(.), and enlarged image, P(.), respectively, the bilinear function is represented as:

$$\begin{aligned} P(x,y) = (1-a)(1-b)Q(i,j) + a(1-b)Q(i+1,j) \\ + b(1-a)Q(i,j+1) + abQ(i+1,j+1) \end{aligned} \qquad (2.1)$$

where a = x – i and b = y – j.

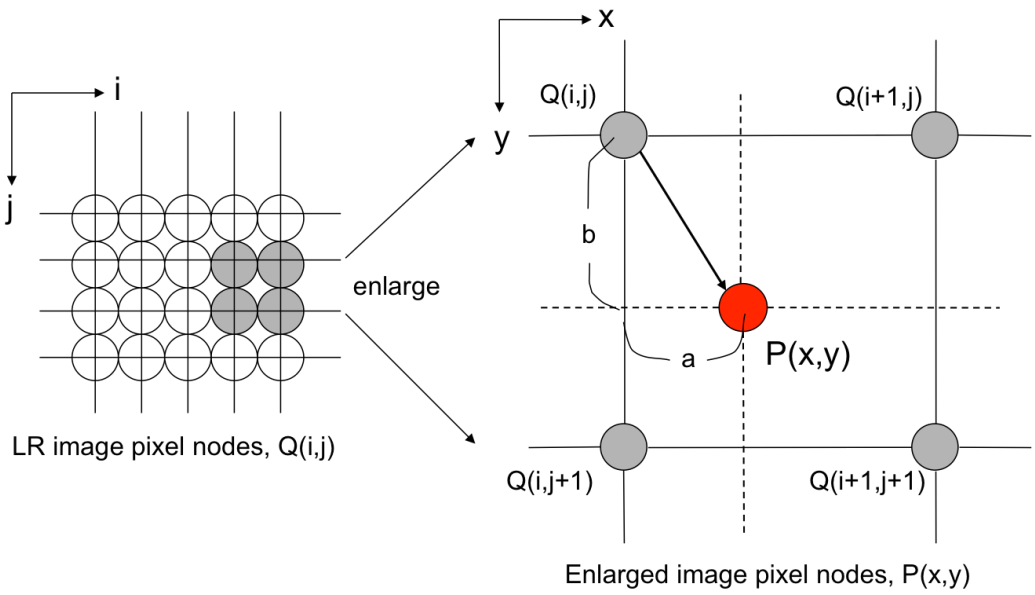
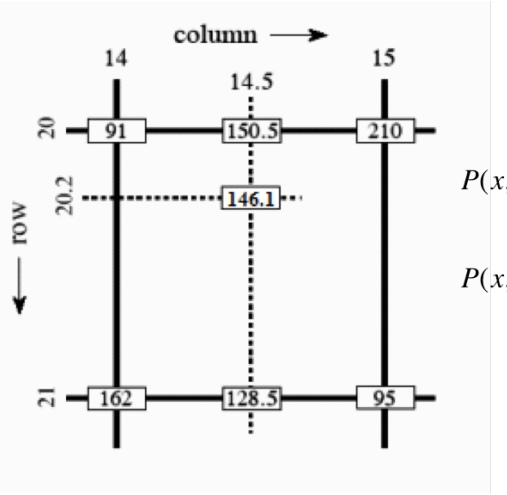


Figure 2.1 The relation of interpolated pixel in bilinear interpolation

Figure 2.2 is an example of bilinear interpolation in grayscale values. Each new pixel value is computed using Equation (2.1), and all of them will then generate an up-sampled image or high-resolution image. The example of an enlarged image using bilinear interpolation is given in Section 2.1.3.



$$P(x,y) = (1-a)(1-b)Q(i,j) + a(1-b)Q(i+1,j) + b(1-a)Q(i,j+1) + abQ(i+1,j+1)$$

$$P(x,y) = 0.5\times0.8\times91 + 0.5\times0.8\times210 + 0.2\times0.5\times162 + 0.5\times0.2\times95 = 146.1$$

Figure 2.2 Example of computation of bilinear interpolation

2.1.2 Lanczos Resampling

Another common interpolation technique is Lanczos resampling, which is named after its inventor, Cornelius Lanczos, who was a Hungarian mathematician and physicist. Lanczos resampling is widely used in video up-sampling for digital zoom applications and image scaling. Compared to linear interpolation, the Lanczos

method minimizes loss of details and can generate a smooth and natural image even after enlargement. The difference is that there are many pixels to be referred and that each referred pixel is weighted. It exploits 16 surrounding pixels (4x4 nearest pixels) or 36 surrounding pixels (6x6 nearest pixels) in interpolation function. The method weights the known nearest 16 pixels to compute a desired pixel value using a windowed sinc function as a practical up-sampling filter. The sinc function and distribution window is illustrated in Figure 2.3. The normalized sinc function is commonly used in digital signal processing and information theory.

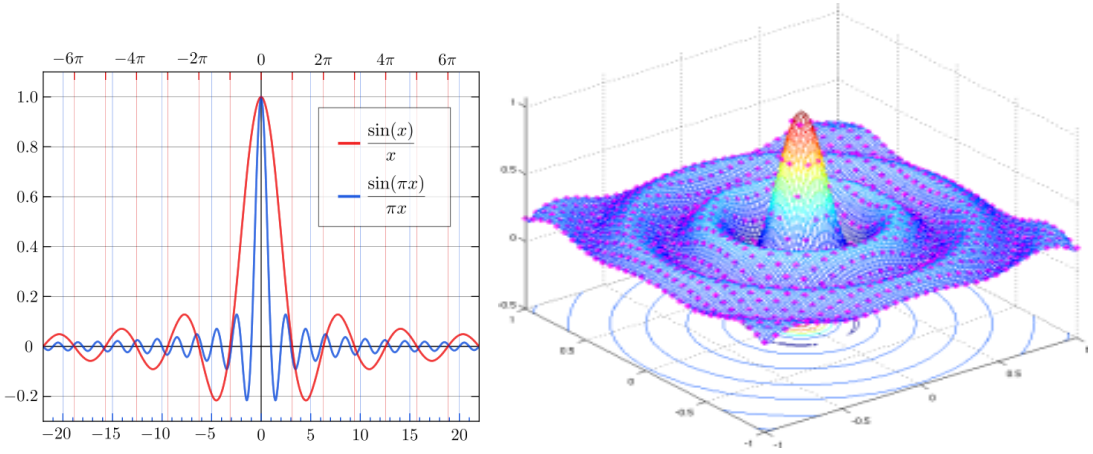


Figure 2.3 The normalized sinc (blue) and unnormalized sinc function (red) and distribution window

Example of an interpolated pixel in Lanczos interpolation is given in Figure 2.4, where P(x,y) is the unknown pixel value which we want to compute. D(.) are the known pixel values from low-resolution image, 16 points of D(.) near the desired pixel, P(x,y) are weighted according to sinc function, and a new value is obtained from these density values. The value for P(x,y) is expressed by the following equation.

$$P(x,y)=\begin{pmatrix} h(x1) & h(x2) & h(x3) & h(x4) \end{pmatrix}\begin{pmatrix} D(i,j+3) & D(i,j+2) & D(i,j+1) & D(i,j) \\ D(i+1,j+3) & D(i+1,j+2) & D(i+1,j+1) & D(i+1,j) \\ D(i+2,j+3) & D(i+2,j+2) & D(i+2,j+1) & D(i+2,j) \\ D(i+3,j+3) & D(i+3,j+2) & D(i+3,j+1) & D(i+3,j) \end{pmatrix}\begin{pmatrix} h(y1) \\ h(y2) \\ h(y3) \\ h(y4) \end{pmatrix} \quad (2.2)$$

Here, interpolation operators, *h(t)* is computed using sinc function as:

$$h(t) = \begin{cases} \sin c(t) \cdot \sin c(t/n) & (|t| \leq n) \\ 0 & (n < |t|) \end{cases} \quad . \qquad (2.3)$$

The parameter *n* is a positive integer, typically 2 or 3, which determines the order of the size of filtering window. If the order is chosen to be 2, 16 surrounding points or pixels are considered, whereas if the order is 3, 36 surrounding pixels are considered to estimate the P(x,y). However, the higher order of Lanczos interpolation requires high computational time, which makes them not suitable for many commercial software (Singh et al., 2013).

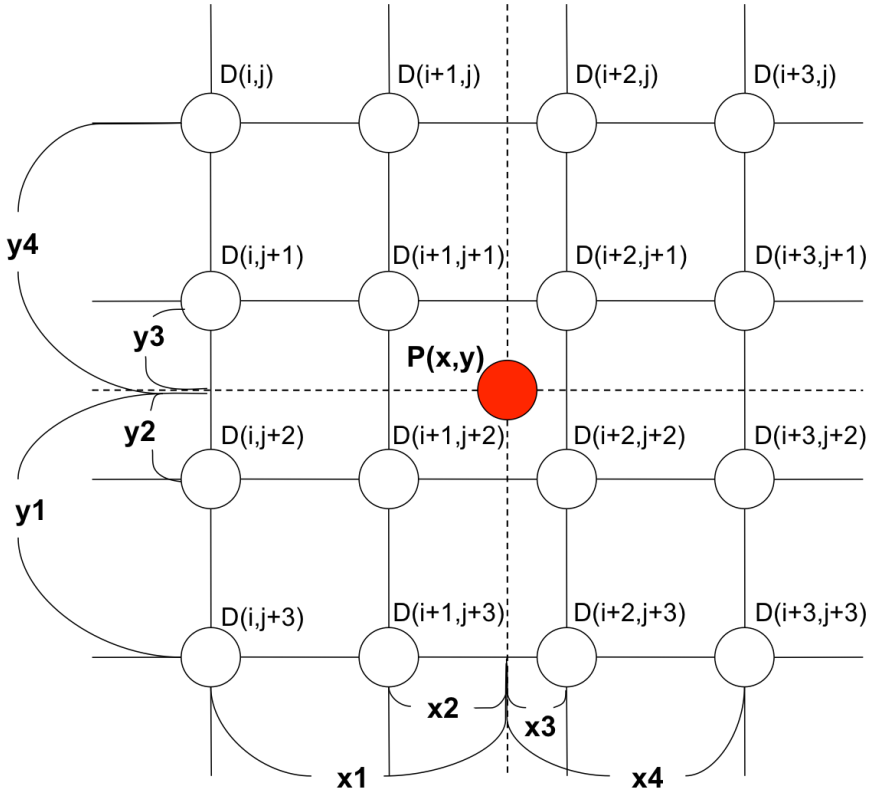


Figure 2.4 The illustration of interpolated pixel in Lanczos interpolation

2.1.3 Discussion

An example of enlarged image using bilinear and Lanczos interpolation is shown in Figure 2.5. The LR image was enlarged by four times the total number of pixels. The circle region is the zoom up version of edge part (texture) in the image for better observation. In comparison, the HR image from Lanczos interpolation seems better or more natural, where their texture is smooth unlike the HR image from bilinear

interpolation, which has block artifacts or grainy along the edges. However, both resulting images are inevitably less sharp compare to the ideal image.

Blur effect in the HR image becomes more obvious as the scaling factor is larger, as we can observe in Figure 2.6. We downsizing the original image by different scaling factor, i.e. quarter, 1/8, and 1/16 of the total pixels, then we upscale back to the original size, i.e. by four times, eight times and 16 times, respectively. The HR image obtained from higher times (scaling factor) of enlargement has an excessively smoothed texture, causing an unclear image.

When any of these interpolation operations are applied, the image edges are smoothed due to the inherent property of these methods of averaging (or weighted averaging) existing pixel values. This results in a loss of high-frequency components in the HR image, which has to be retained somehow, or estimated accurately. Otherwise, we can hardly get any information from the desired image, which that defeat the purpose of image enlargement in the first place. Here, super-resolution (SR) techniques have currently been widely adopted to resolve these problems. Super-resolution is a significant method that not only enhancing image resolution, but also predicting detailed information on the missing high-frequency details to define the enlarged image.

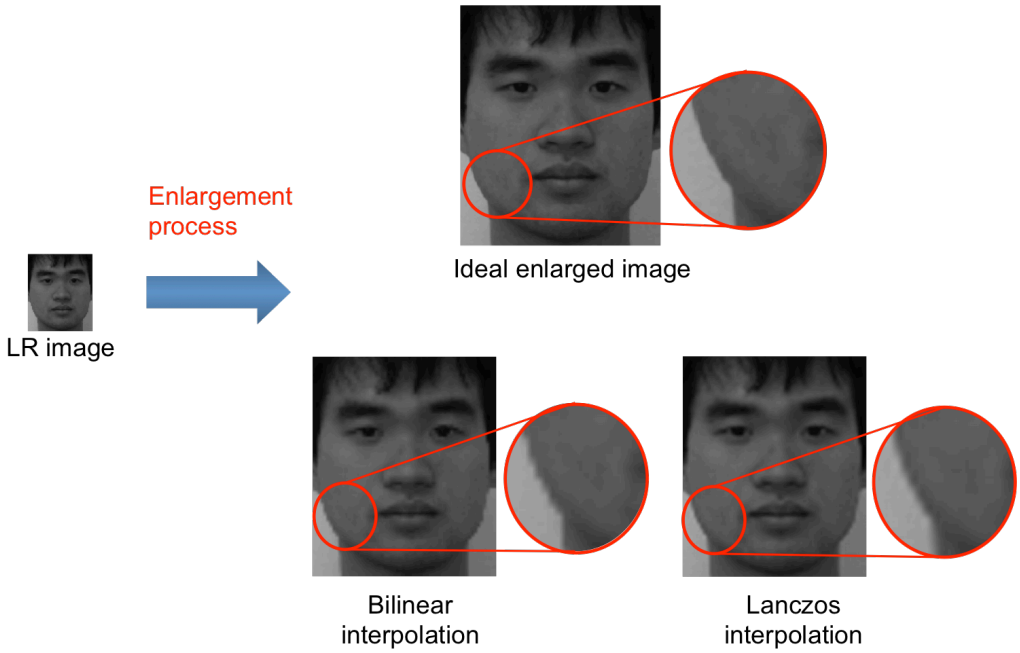


Figure 2.5 Enlarged image using bilinear and Lanczos interpolation

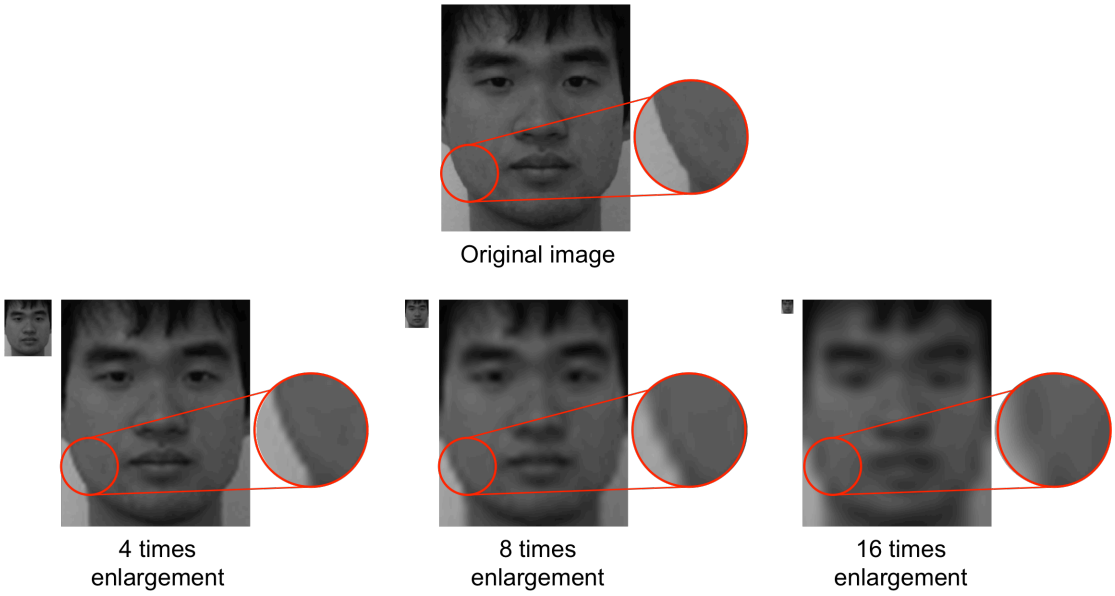


Figure 2.6 Example of enlarged image under different scaling factor using Lanczos

2.2 Image Compression and Noise Filtering

Image compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission. Without compression, the volumes of data produced by digitizing surveillance image streams would swamp the available storage. Since surveillance camera records long-term footage, the process of compression is necessarily applied to the recorded footages in order to reduce the amount of information that needs to be transmitted and stored.

Such compression may reduce the size of image data significantly and save a lot of memory storage, but at the same time the quality of image also inevitably decrease. Texture and details in a compressed image are distorted due to block noise appearances, causing less information could be obtained from the image. To overcome this, noise-filtering process is required to reduce the block noise for a better image observation.

2.2.1 Data versus Quality

Surveillance camera record non-stop footages and store them directly to Digital Video Recorders (DVRs). There is an inevitable trade-off between video quality and cost, since high-resolution, high-frame-rate and uncompressed video requires more bandwidth and memory space. For instance, a day's worth of surveillance can easily reach 1 Terabyte of storage! So, scene footages or images from surveillance camera

are usually compressed in order to reduce size of data due to memory constraint. Such compression yields block noise in images, which leads to degradation of image quality causing a loss of information. In other words, the biggest challenge in surveillance systems is, either we want to compromise data storage, or quality of footage.

Nowadays, there are high-definition surveillance cameras available that record high quality footage, indeed. However, the deployment cost is too expensive and unaffordable for many places or owners, so most of them still bear with the use of standard surveillance cameras. Owners are attempting to cut storage and transmission costs to save money running their surveillance systems and as a result compromising video quality over cost (Cohen, 2004 and Gerrard et al., 2007). Thus, such compression of image is still practically necessary, so the demand and dependency on image enhancement technology to resolve degraded footages problems is remarkably high and significant. Therefore, a lot of studies have been on going with various developments achieved recently in surveillance image restoration fields.

The compression formats used in surveillance cameras vary by manufacturer and product. But the four most commonly used compression formats are: Motion JPEG (also written as M-JPEG or M-JPG), H261, Wavelet and MPEG (also written as mpg). Image compression is measured as a ratio of the amount of data entering the compression system. A compression ratio of 1:1 indicates no compression; 10:1 indicates that there is 10 times less data after compression, and so on. Several compressed images with different compression ratio are given in Figure 2.7, where the compression ratio decrease from left to right. The lower the compression ratio is, the bigger the size of image data to be stored.

As we can see in the highly compressed image (the first image from the left), block noise has obviously appeared. Texture component, especially around important face feature, i.e. eye, nose and lips, are completely distorted compared to the original image. Such distortion would lead to a failure of analysis of a face image, either in face identification or recognition process, or face matching algorithms. In order to resolve the problems, several approaches have been introduced to remove block artifacts and enhance the degraded compressed image.

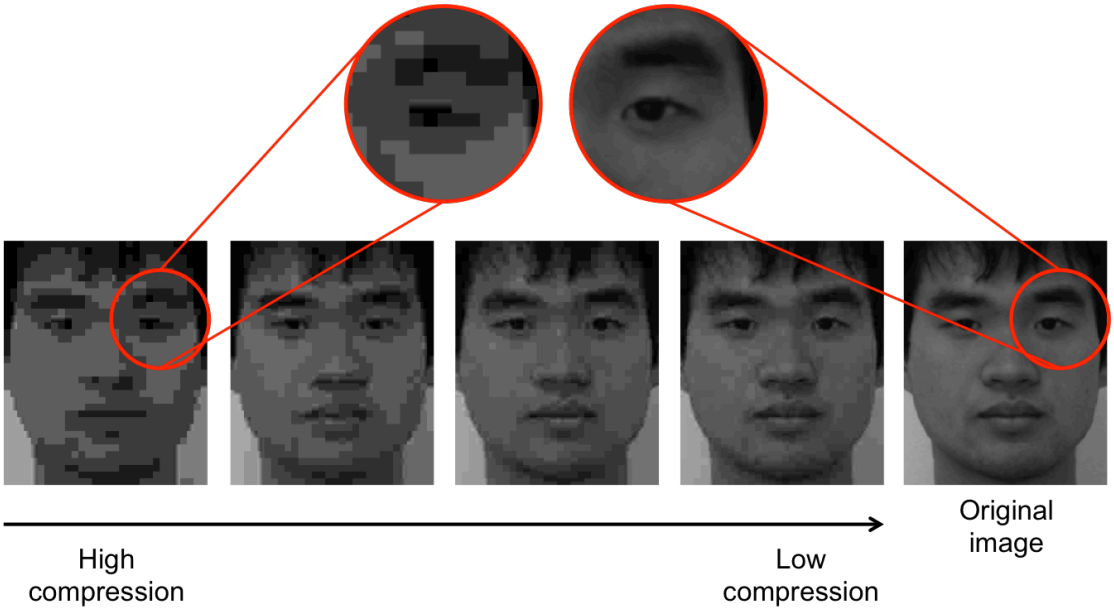


Figure 2.7 Degraded face images with different level of compression rate

2.2.2 Gaussian Filter

In image processing, point-spread-function (PSF) is commonly used to reduce noise of image. One of the PSF is Gaussian filtering method. It is a linear filter that usually used to reduce noise or to blur the image. The Gaussian distribution function is represented as:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \qquad (2.4)$$

Figure 2.8 shows a Gaussian distribution when σ is set by 5. The nearer the observed position to the focus or center point is, the higher the concentration. In digital image processing, Gaussian distribution is not only used to smooth an image, but have also being used to apply noise in image.

The higher the σ is, the stronger the filter. The stronger the filter is, the lesser the noise. Figure 2.9 shows resulting images after Gaussian filtering was applied with different σ value. For a highly compressed image, stronger filter (larger σ) is needed for a significant volume of noise reduction. Vice versa, for a lower compressed image, small σ value, i.e. typically 2 or 3, is sufficient. Larger σ value would only increase processing time and risk of texture component being distorted by over smoothing process.

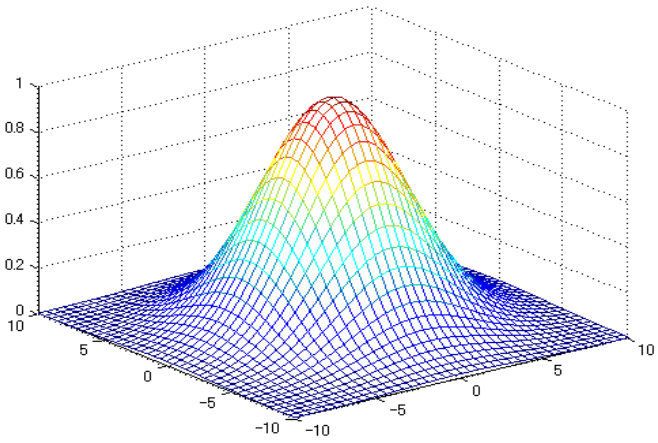


Figure 2.8 Gaussian distribution (σ = 5)

From the filtered images, we can observe that block noise have been greatly reduced, as the filter was stronger. However, the image become blur due to excessive spreading or overly smoothing process, where texture component have been missed out together with noise component. This is because both texture and noise are high-frequency component in a digital image. Here, it is essential to predict and restore back the missing texture details to generate a sharper texture and increase identifiability of face image.

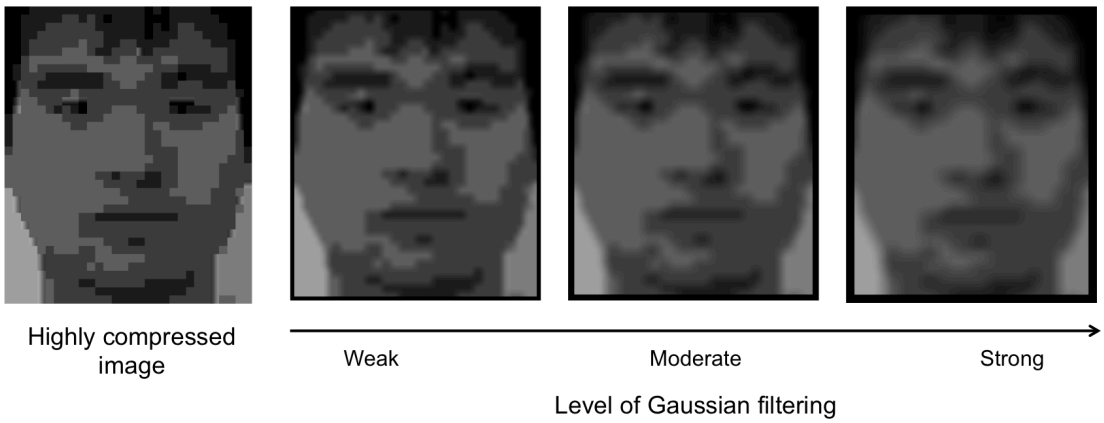


Figure 2.9 Block noise reduction using Gaussian filtering with different strength (the higher the σ is, the stronger the filter)

2.3 Chapter Summary

This chapter present conventional methods for image resampling and their common weaknesses or after effect. We have clarified that when interpolation function is applied, crucial information such as sharp edges and detailed texture is

inevitably smoothed. When scaling factor is increase, the blurry effect becomes obvious, encouraging the need to restore the distorted texture components somehow.

We have also described about block noise occurred in degraded image causing from a JPEG compression. For noise reduction in image, conventional methods using point-spread function (PSF) such as Gaussian filter are significant for less amount of degradation. But, for a high amount of degradation (lower quality of image), the PSF have to be applied stronger to be effective. However, new problems arise. When stronger PSF is applied, texture details or sharp edges are also also affected, leading to a blurred resulting image.

Similar problem with conventional method of image enlargement and noise filtering is the missing of texture details in a desired image after the process. Hence, it is really useful if there are methods that can be utilized to reconstruct texture and reduce the blur effects in the after-processed image.

CHAPTER 3

EXAMPLE-BASED METHOD

Several conventional methods of image enlargement, such as Bilinear, Bicubic and Lanczos interpolation, are commonly used to infer a more legible high-resolution image from a low-resolution image. In previous chapter, we have briefly elaborated on how these analytic approaches typically suffer from a blurred appearance due to perceived loss of detail in textured regions, causing unsatisfying quality in an enlarged image. The image quality appears to be more blurry or fuzzy, especially along the texture or edge when the magnification scale is larger. In order to resolve these problems, super-resolution techniques have currently been widely adopted. This is because super-resolution techniques have not only enhanced image resolution, but also predicted detailed information on the missing texture component to define and reconstruct the enlarged image's texture.

Super-resolution aims to produce a high-resolution image using one or several observed low-resolution images. The major difference between interpolation and super-resolution methods is that interpolation only involves up-sampling the low-resolution image, while super-resolution usually involves three major processes, which are up-sampling (interpolation), de-blurring and de-noising (Siu and Hung, 2012). Example-based method or also well known as learning-based method is one of super-resolution techniques, which is most suitable for specific applications such as face image super-resolution, since it typically uses specific kinds of images as its dictionary. This chapter will describe in details about example-based method that has been proposed by Freeman et al. (see reference [26]), which is closely related to our proposed algorithms later.

3.1 Example-based Algorithms

Example-based method utilizes external information from a set of a large volume of training image pairs of low-resolution (LR) images and high-resolution (HR) images created using pyramidal image system. Compare to conventional interpolation methods that only use information from input images, example-based

method enables new information to be predicted for the missing high-frequency components by exploiting external information available in database.

Example-based processes basically consists of two phases, which are, 1) a training patches database construction phase that extracts patch pairs from both LR and HR images, and then stores them as training patches in the database, and 2) a super-resolution phase that executes a learning model to reconstruct HR images by searching patches in the database that are best matched to the input image patches. Freeman et al. has proposed example-based approach for SR using a bandpass filter to extract training images' texture in database to be used as external information, and Markov Random Field (MRF) network is employed as learning model to estimate plausible texture for output HR image.

3.1.1 Training Database Construction

We start from a collection of high-resolution images to construct a database. We restrict use of the same kinds of images to obtain better probability of similarity among patches. For instance, to process a face-image, multiple different face images are used in database. The HR training images are down-sampled typically by one-quarter the number of original pixels in each dimension (1/16 of the total number of pixels) to create LR training images. After that, we initially up-scale the LR images back into their original size using interpolation methods such as Lanczos.

Then, we preprocess the training images to extract high-frequency information so that only textures are being observed. We apply a Gaussian filter to the images to extract their middle frequency component and stored them into database, Po, as outlined in Figure 3.1. Corresponded image texture details are extracted by subtracting initial up-scaled images from original HR images, then storing them as patches in database, Pc. These patches are typically 5x5 or 7x7 pixels.

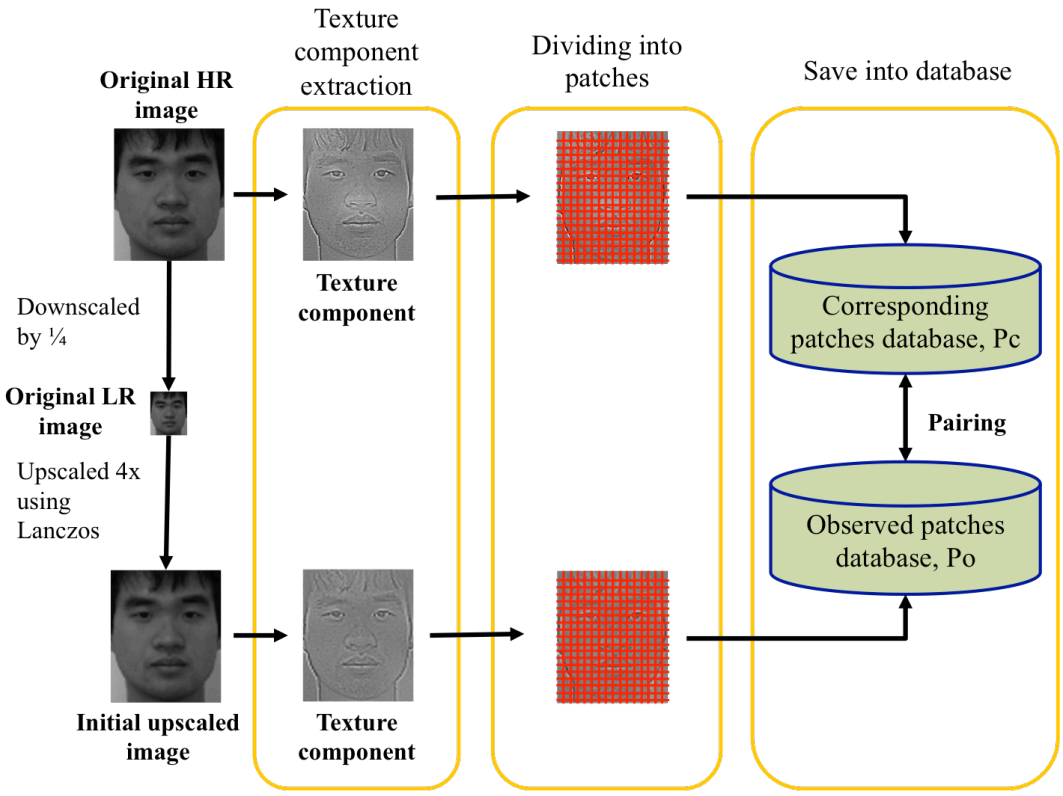


Figure 3.1 Database construction process

3.1.2 Super-resolution Phase

Example-based method implements a Markov Random Field (MRF) network as its learning model. The MRF network probabilistically models the relationships between input image patches and estimated training images patches, and between neighboring high-resolution patches by using an iterative algorithm. The method models the relationship between patches by using the MRF network in high-frequency domain (texture component) of an image.

Figure 3.2 illustrates the structure of the MRF network where each circle represents a network node and each line indicates the spatial relationship and statistical dependencies between nodes. The nodes Y(i,j) represent the LR patches at the position (i,j) in the input image. The nodes X(i,j) represents the LR patches in the database whose corresponding HR patches are used to estimate the texture components at the position (i,j) in the restored (hidden) image.

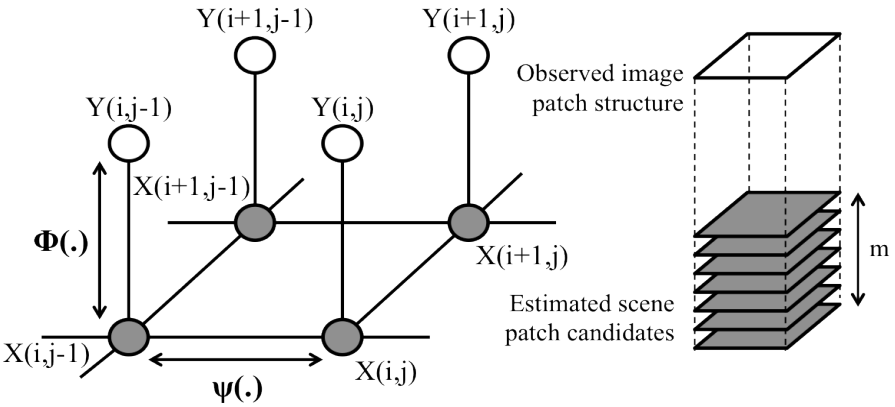


Figure 3.2 MRF network model

The dependency between nodes are represented as two compatibility functions ϕ(.) and ψ(.). For a position (i,j) in the MRF network, the function ϕ[X(i,j),Y(i,j)] known as the patch similarity represents the similarity of texture between the observed patch Y(i,j) and the estimated patch X(i,j). For a position (i,j) and its adjacent position (u,v), the function ψ[X(i,j),X(u,v)] called as the border compatibility represents the compatibility of the common border between the estimated patches, X(i,j) and adjacent patches, X(u,v). The number of patch candidates is given as a constant parameter m.

The joint probability of X(.) under the condition of Y(.) is defined as:

$$P(X \mid Y) = \prod_{ij} \phi\left[X(i,j), Y(i,j)\right] \times \prod_{ij,(u,v)\in NB(i,j)} \psi\left[X(i,j), X(u,v)\right] \quad (3.1)$$

where NB(i,j) denotes the neighbors of structure X(i,j). The right, left, top, and bottom patches of the observed patch are considered as its neighbors.

To specify the MRF network's ϕ(.) function, a similar quadratic penalty is imposed on differences between the observed input-image-patch, Y(i,j), and the scene-patches from the training set, X(i,j).

$$\phi\left[X(i,j), Y(i,j)\right] = \exp\left\{-\frac{d\left[X(i,j), Y(i,j)\right]}{2\sigma_1}\right\} \quad (3.2)$$

where d(.) is the distance of the two matrices (or vectors) and σ_1 is a constant parameter. We use the root sum of the squared differences of the pixel values between the two patches.

The input image's patches are divided in a manner where they overlap each others by one or more lines of pixels, to specify the Markov network's ψ(.) later. Figure 3.3 illustrates the overlap region between two patches. Let *p* be the pixels of patch X(i,j) in the overlap region between patches X(i,j) and X(u,v), and let *q* be the corresponding pixels of patch X(u,v), next to patch X(i,j). In this case, the ψ(.) function can be represented as:

$$\psi\left[X(i,j),X(u,v)\right]=\exp\left[-\frac{d\left(p,q\right)}{2\sigma_2}\right] \qquad (3.3)$$

where p (q) is the vector of pixels of the overlap region in patch X(i,j) (X(u,v), respectively) and σ_2 is a constant parameter.

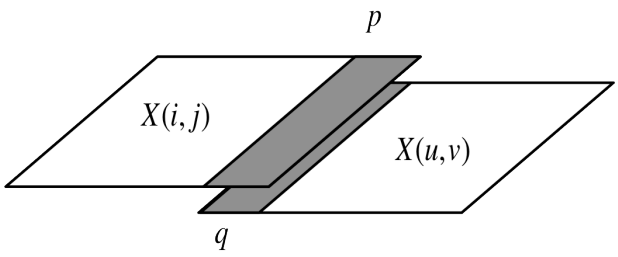


Figure 3.3 Region of overlap

3.1.2.1 Patch Candidates Selection

We previously chose a number of patch candidates that had the most similar pattern of texture with the input patch for each node according to its ϕ(.) value from Equation (3.2). The number of patch candidates should be reasonable; not too few which possibly causes lack of good choice of patches, and not too many since processing time would increase unnecessarily or undesirable patches would be selected as candidates inevitably.

Thus far, the number of patch candidates was typically fixed at 16 or 30 closest examples. However, more training images means more patches have to be considered. If we fix the number of candidates in the case of 100 training images same as in the case of 10000 training images, we may unintentionally neglect some good candidates from 10000 training images. They probably do not have the best

ϕ(.) value, but they may have better ψ(.) value. As was previously explained, we must consider both compatibilities justly to obtain the best P(X|Y).

Therefore, in present research, we set the number of patch-candidates variable based on the number of training patches in database. We set one for one, i.e., one patch candidate for one training image. For example, if we used 100 training images as database, 100 closest examples among the total patches would be chosen as candidates. For research experiments later, we have conducted a preliminary experiment to find in advance the optimum number of training images needed in database. Figure 3.4 shows that the more number of training images used in database the better the resulting image (higher PSNR means better image quality). PSNR value is optimum when 70–110 images are used in database. For the sake of better result, we used 110 images in database for our proposed method later.

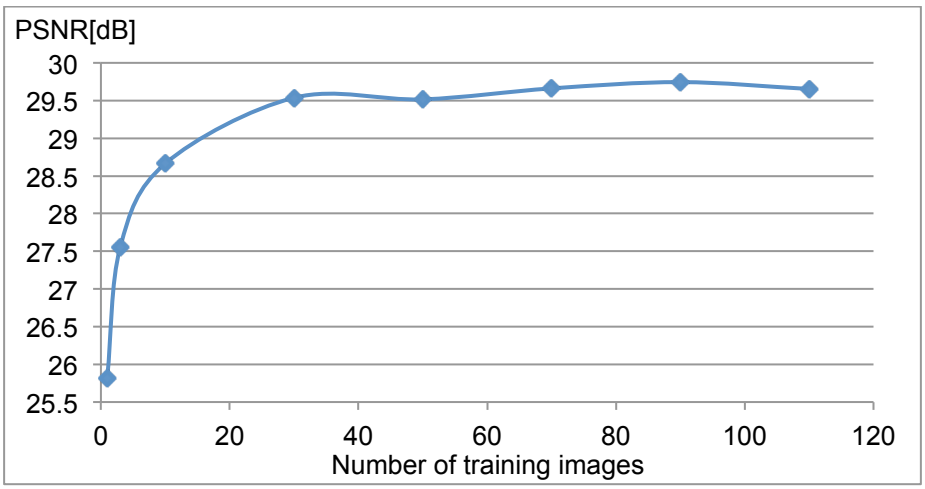


Figure 3.4 Number of training images in database against PSNR value for resulting image

3.1.2.2 Super-resolution Process

Figure 3.5 outlines the inferring process for image super-resolution, i.e., deriving an output HR image from a single input LR image. We initially up-scaled the input LR image into the targeted size of the HR image, and then extracted its texture component (middle frequency) using a Gaussian filter. We divided it into patches to find the most similar patches in database Po.

Those corresponding patches from database Pc were placed together to build a base image for the MRF learning model. We processed the image by several iterations learned from the MRF relationship to derive its output high-frequency

image. Finally, the expected high-frequency component was combined with the previous up-scaled image to obtain the final output HR image.

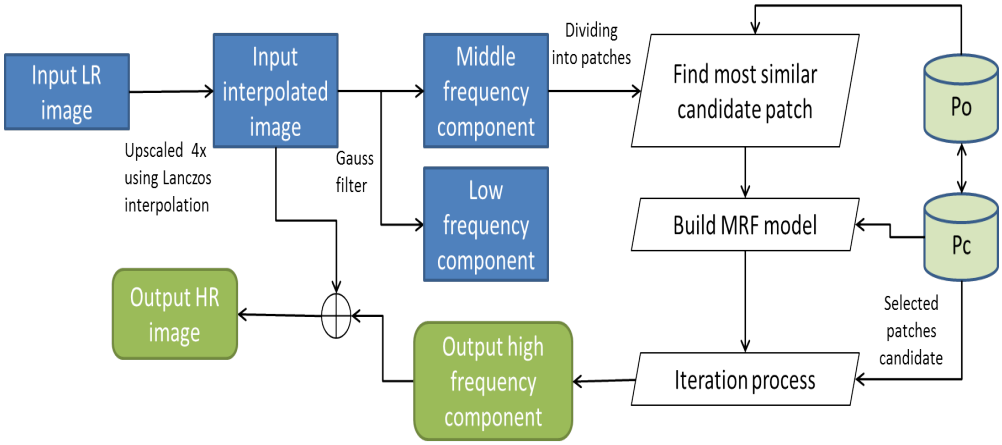


Figure 3.5 Super-resolution process

3.2 Previous Related Studies

Various image processes benefit from example-based methods such as in panoramic imaging, image compression, pattern recognition, and 3D modeling. Several methods have been proposed through different approaches in terms of feature extraction and type of learning model. Freeman et al. has proposed example-based algorithms, using a bandpass filter to extract image textures and a Markov random field (MRF) as a learning model to infer the suitable HR patches. Since then, many algorithms have been developed using different combination of techniques as improvisation. Thus far, Lui et al. suggested another MRF-based SR using wavelet feature maps. Yang et al. implemented a dictionary learning strategy to SR using sparse representation. Jiji et al. applied a best matching model on wavelet and contourlet feature maps. Several years later, Wu et al.'s proposed method adopts a contourlet transform feature maps with MRF as its learning model. Some other examples of proposed methods are listed in Table 3-1 (the list given was derived from reference [35]).

Table 3-1 Feature extraction and learning models for example-based super-resolution

Authors	Feature extraction	Learning models
Freeman et al.	Bandpass filtered feature map	Markov random field
Sun et al.	Gaussian derivative feature map	Markov random field and backprojection
Baker et al.	Gradient derivative feature map	Best matching
Suetake et al.	Low-and high-frequency components	Best matching
Chan et al.	First-order gradient and norm luminance map	Neighbor embedding
Li et al.	Laplacian pyramid feature map	Content-based classification and class-specific predictors
Su et al.	Steerable pyramid feature map	Best matching
Jiji et al.	Wavelet feature map	Best matching
Lui et al.	Wavelet feature map	Markov random field
Jiji et al.	Contourlet feature map	Best matching
Ni et al.	DCT coefficients	Support vector regression
Chang et al.	Bandpass filtered feature map	LLE Construction
Wang et al.	Luminance map	PCA Construction
Yang et al.	Bandpass filtered feature map	Sparse representation
Hua et al.	PCA coefficients	Canonical correlation analysis
Ma et al.	Luminance map	Position-patch construction
Shi et al.	Luminance map	Learning based interpolation method and deconvolution
Dong et al.	Luminance map	Sparse representation and nonlocal similarity
Glasner et al.	Luminance map	Reconstruction-based and learning based super-resolution
Wu et al.	Contourlet feature map	Markov random field

CHAPTER 4

FACE IMAGE SUPER-RESOLUTION

Surveillance cameras have been recording digital footage of crime scenes that are useful as clues in investigation. Face image especially, from surveillance camera often provides valuable information that enables us to identify the identity of unidentified person of interest in investigation. Unfortunately, most standard surveillance cameras record low-resolution and highly compressed frames due to storage constraint, causing poor quality of footage. Thus, most recorded face images are frequently too poor to be helpful cues for investigation since less information could be obtained.

One of the main problems is, face images are small in size or low in terms of resolution since surveillance camera records from a certain distance, and need to be enlarged for a better look and recognition. Several conventional methods of image enlargement, such as Lanczos interpolation, are used to infer a larger image from a small image, but typically suffer from a blurred appearance due to perceived loss of detail in texture regions. While texture of facial features are crucial in face identification. Therefore, it is necessary to enlarge a face image and at the same time preserve or restored somehow their texture details.

In this chapter, we present our proposed method specifically for face image super-resolution, which takes facial parts into account. Compatibility function for learning model in our method is an improvised version of MRF function in Freeman et al.'s method before, where a function to estimate facial parts is employed.

4.1 Normalized Face Image Super-resolution

Example-based algorithms are most suitable for face image super-resolution since they typically use specific kinds of images as their dictionary. So for face image enhancement, only face images are used as training images in database. By restricting usage of the same kinds of images, the probability of similarity among features is better. With the exact same reason, we come up with a significant idea to reconstruct face image based on their facial parts. This will certainly increase the probability of similarity since patches are chosen from same facial parts.

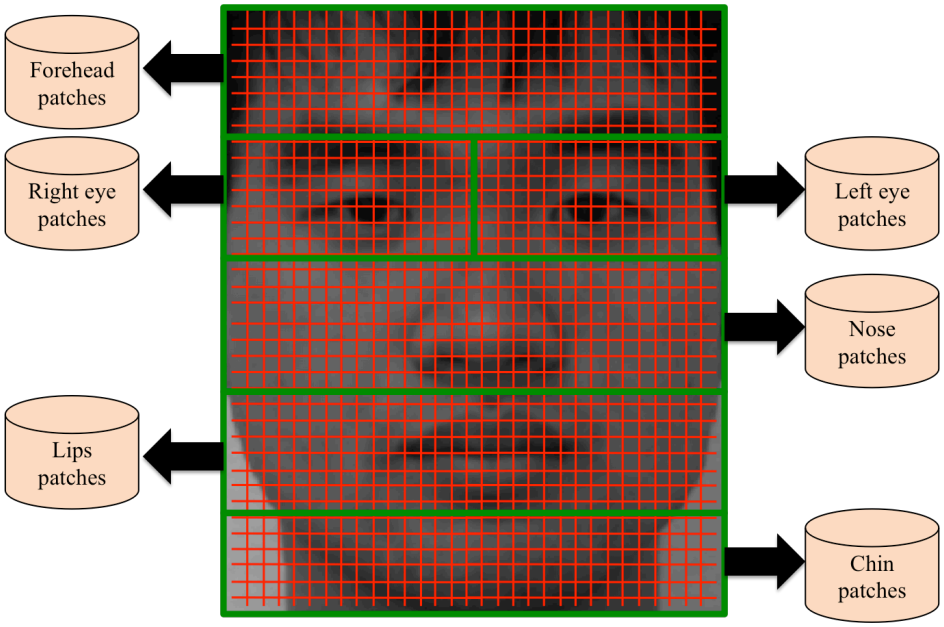


Figure 4.1 Multiple databases based on facial parts

The original idea was to categorize training patches into multiple databases based on their facial parts, for example, a database of eye patches, a database of nose patches, and so on as shown in Figure 4.1. However, we propose another way of constructing the training database in order to simplify the algorithm, which is by utilizing a set of normalized high-resolution face images. These normalized face images should all have same size with same ratio of facial features, where face feature points (e.g., eyes, noses, mouths, chins, and face boundary lines) in each image are approximately at the closer or comparable position.

By using normalized images, we can simply take patches' original locations on its image into account to determine which facial parts they belong to, rather than typically divide multiple facial images into parts manually or categorize training patches into several databases beforehand. Based on patch-position information, the nearer the distance between training patches' position to a targeted input patch's position, the higher the compatibility of facial parts. Figure 4.2 illustrates facial parts estimation according to patch position. (i,j) refers to a targeted input patch coordinate, (k,l) refers to training images' patch coordinate, while ℓ is the distance between both patches. If the targeted patch comes from the eye part, the algorithms would prefer patches of eye parts, i.e. patches with shorter ℓ, from the training set to be its corresponding patch candidates.

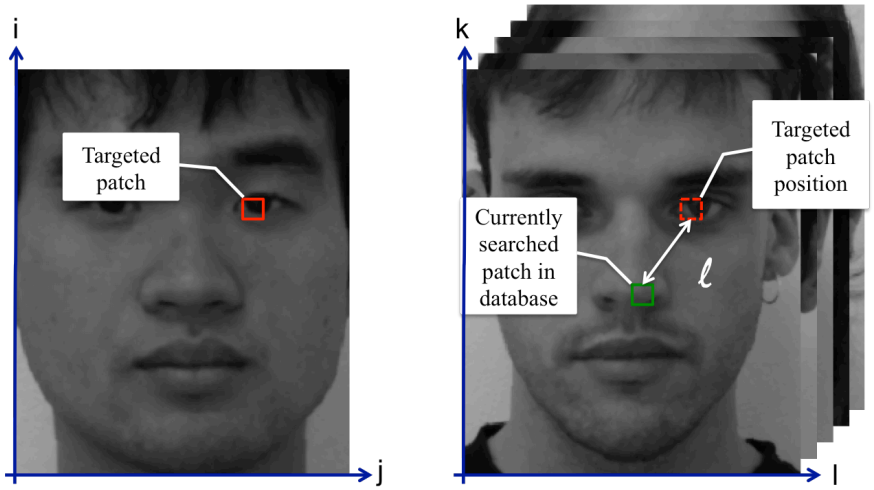


Figure 4.2 Facial parts estimation based on patches' distance

4.1.1 Database Construction

A collection of normalized high-resolution images of human faces was utilized as training images. Same as Freeman et al. method, only texture component in image should be observed during super-resolution process. Each image was down-sampled into a low-resolution image by typically one-quarter the number of original pixels in each dimension (1/16 the total number of pixels), and resampled back into its original size to create an initial up-scaled or enlarged image. Lanczos resampling method was employed in this process.

After that, low-frequency component on both original high-resolution and initial up-scaled images was filtered out to retain only their texture, which are high-frequency component for the former and middle-frequency component for the latter, as outlined in Figure 4.5. Both components are then divided into patches in a manner where both patches correspond to each other, and overlap their adjacent patches by a line of pixels to later specify the Markov network.

The main feature of our proposed database is each patch's position coordinate was observed as in Figure 4.3 to be used later in learning-model during super-resolution process. Therefore, patches were divided like a grid map, where if patch size was 5x5 and overlap region is a line of pixels, patches had to be divided by interval of four pixels for each dimension. Given a set of face images with size of 288x240 in pixels, with interval of four pixels between each patch, a set of 72x60 patches (4320 patches) can be created exactly from each image.

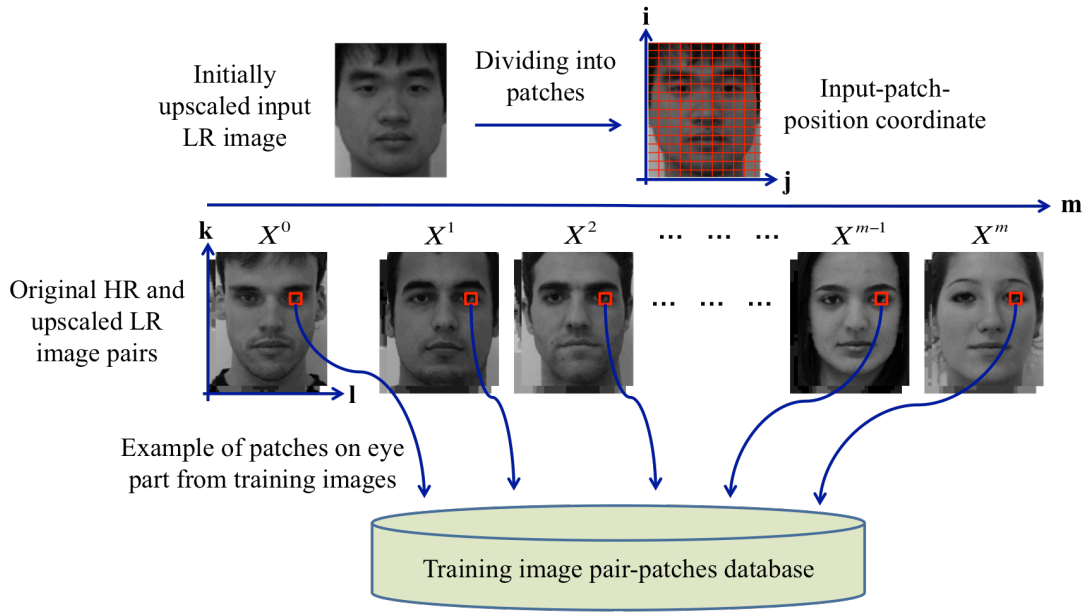


Figure 4.3 Patch coordinate-based database construction

4.1.2 Learning-model with facial parts estimation

The underlying idea is to select patches in the database according to their facial parts, i.e. using eye patches to reconstruct eyes and nose patches to reconstruct nose part. We propose an innovative function to estimate the facial parts compatibility, represented as λ(.), which is defined as:

$$\lambda\left[X(i,j)\right] = \exp\left[-\frac{\ell}{2\sigma_3}\right] \qquad (4.1)$$

where ℓ is a Euclidean distance from the original position of training patch X(i,j) in the training face image to the targeted node position (i,j) in the restored image, and σ_3 is a constant parameter. Hence, the joint probability of X(.) under the condition of Y(.) is an extension of Equation (3.1), which can be expressed as:

$$\begin{aligned} P(X \mid Y) = & \prod_{ij} \phi\left[X(i,j), Y(i,j)\right] \\ & \times \prod_{ij,(u,v)\in NB(i,j)} \psi\left[X(i,j), X(u,v)\right], \\ & \times \prod_{ij} \lambda\left[X(i,j)\right] \end{aligned} \qquad (4.2)$$

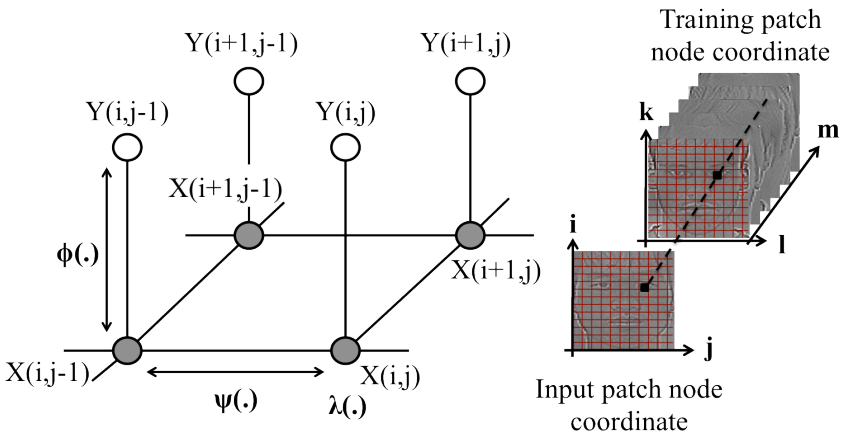


Figure 4.4 Proposed learning model with facial-parts compatibility function

Figure 4.4 shows our proposed network model. m represents the number of training images available in database, while (k,l) is coordinate for original position of training patch X(i,j) in their image. The main benefit of using normalized facial images for the training set is that we do not need to typically divide the images into parts manually or categorize them beforehand into eye, nose and mouth databases. Since face images have been normalized, we can simply consider patches' original position on its image to determine which facial parts they belong to.

Hence, the nearer the distance between a targeted input patch, Y(i,j) and a training patches, X(k,l) are, the higher facial parts compatibility becomes. In other words, if the input patch came from the eye part, the learning model would prioritize training patches from eye region in the training set to be its corresponding patch candidates.

4.1.3 Patch Candidates Selection

The major difference between Freeman et al.'s method and our proposed method is the algorithms for patch-candidates selection. Our method selects patch candidates for each nodes by not only based on patch's texture similarity, ϕ(.), but also take facial parts compatibility, λ(.) into consideration. Training patches with higher pixel values similarity and nearer position to the targeted input patch have a higher possibility to be selected.

Since this time there is a trade-off between two aspects in patch-candidates selection process, i.e. ϕ(.) and λ(.), we applied a weighting factor as:

$$P_C(X \mid Y) = \prod_{ij} \phi\left[X(i,j), Y(i,j)\right]^{\alpha} \times \prod_{ij} \lambda\left[X(i,j)\right]^{(1-\alpha)} \quad . \qquad (4.3)$$

$P_C(X|Y)$ refers to probability of patches candidates, where if we want to select 100 best patches, C is 1 to 100. α is between zero and one.

The α can be treated as determining factor to examine the effectiveness of our proposed facial parts compatibility function. If α is one, the function would not take the facial parts' compatibility, λ(.), into consideration at all, which means this is similar to the Freeman et al. method. Vice versa, if α is zero, it will only take the λ(.) into consideration and neglect the patch similarity, ϕ(.). The lower the α value, the higher its preference for the facial parts' compatibility. Regarding the number of patch candidates for each node, it is set same as number of training images used in database. In experiment later, 110 training face images were used as dictionary so the number of patch candidates is 110 patches.

4.1.3.1 Sorting the candidates

In another words, α can also be defined as percentage of preference. For instance, if α is 0.8, percentage of preference is 80% for ϕ(.) over λ(.). 80% of probability value for ϕ(.) and 20% of λ(.) are totaled together to get $P_C(X|Y)$. In this case, probability is weighted towards texture similarity, where patches with better ϕ(.) have higher chance or priority to be selected as candidates, even if their λ(.) is not so good (far from targeted facial parts). According to α, patches were sorted amongst hundreds thousand patches available in database. Different α makes different sequence of patch candidates. Let the number of patch candidates for each nodes is 110 patches, the first best 110 patches from those patches will be the shortlisted patch candidates.

As example, five patches' sequence under four different cases of α are sorted as in Table 4-1. Values written under Patch 1 to Patch 5 row represent the sum of squared differences of patch pixel values, ϕ(.) and distance of patch position, λ(.) to a targeted patch. The lower the value, the more similar the patch to, or the closer the patch's position to the targeted patch. It is like a demerit points system where the one with lowest points will be the winner.

Table 4-1 Example of patches sorting based on demerit points

	Percentage of preference		Patch 1		Patch 2		Patch 3		Patch 4		Patch 5		Patches sequence
			ϕ(.)	λ(.)	ϕ(.)	λ(.)	ϕ(.)	λ(.)	ϕ(.)	λ(.)	ϕ(.)	λ(.)	
α	ϕ(.)	λ(.)	5	5	3	7	7	3	20	1	1	20	
0.5	50%	50%	5		5		5		10.5		10.5		1,2,3,4,5
0.2	20%	80%	5		6.2		3.8		4.8		16.2		3,4,1,2,5
0.9	90%	10%	5		3.4		6.6		18.1		2.9		5,2,1,3,4
0.6	60%	40%	5		4.6		5.4		12.4		8.6		2,1,3,5,4

According to ϕ(.) and λ(.), those five patches' traits can be summarized as follows:

- Patch 1 : Patch similarity and position is fair
- Patch 2 : High patch similarity but far from targeted patch
- Patch 3 : Low patch similarity but near to targeted patch
- Patch 4 : Considerably low patch similarity but very near to targeted patch
- Patch 5 : Considerably high patch similarity but very far from targeted patch

Different α will results in different priorities in patches selection process. Table 4-1 shows different sequences of patches when different α was applied.

Case 1 : When α is 0.5, the function considers ϕ(.) and λ(.) equitably. So, Patch 2 and Patch 3 have same demerit points as Patch 1 despite that they have different traits. This is because although Patch 2 has a little bit higher λ(.), it has much better ϕ(.) compare to Patch 1. Vice versa for Patch 3. Patch 4 and 5 have extremely higher points for ϕ(.) and λ(.), respectively, so Patch 1, Patch 2 and Patch 3 are prioritized.

Case 2 : When α is 0.2, 80% of λ(.) points are accounted, which means patches with better facial-parts compatibility are prioritized. Thus, Patch 3 and 4 which have lower demerit points of λ(.) are preferred over the others, despite the fact that Patch 4 have worse points in terms of ϕ(.).

Case 3 : Conversely, when α is higher, for instance 0.9, patches with better ϕ(.) are prioritized. As a result, Patch 5 which has the best ϕ(.) is preferred over Patch 2 even though it has worse λ(.), followed by Patch 1, Patch 3 and Patch 4.

Case 4 : When α is slightly weighted towards ϕ(.), for instance, 60% of ϕ(.) and 40% of λ(.), Patch 2 is prioritized over the others instead of Patch 5 which has best ϕ(.). We can conclude that when a moderate value of α is applied, the function would recommend patches which have moderate points, and demote those which has extremely high demerit points either of ϕ(.) or λ(.), i.e. Patch 4 and Patch 5.

Above assessment shows how five different patches are sorted during patch-candidates selection process. In reality, there are hundreds thousand training patches available in database that are needed to be sorted. The huge number of patches comprising patches with minor different of traits, some may have same ϕ(.) but slightly different λ(.), or slightly different ϕ(.) but same λ(.) and so on. Hence, best α for each different face sample super-resolution is varied.

4.1.4 Iteration Process

Figure 4.5 illustrates overall algorithms for the proposed face image super-resolution method, starting from how a training patches database was constructed, until how a resulting super-resolved face image finally generated.

As shown in the algorithms, before iteration process, an initial image model was created by merging the best matched HR patches (patches with the best ϕ(.) – at this time λ(.) was not considered yet) of all nodes. This image would look grainy or has block artifacts since the chosen patches are not fit with their adjacent patches yet, which proves that local patches alone are not sufficient to estimate plausible looking high-resolution images (Freeman et al. 2002). Here, the iteration process was applied to remove those artifacts.

Iteration process is performed to find a set of output patches that are best matched with each other in overlap border. The aim is to find a set of output patches, X(.), where P(X|Y) is maximum. By using only the already selected patch candidates before (no need to observe all patches in the database), P(X|Y) would typically be at an optimum value, and additionally the processing time can also be reduced.

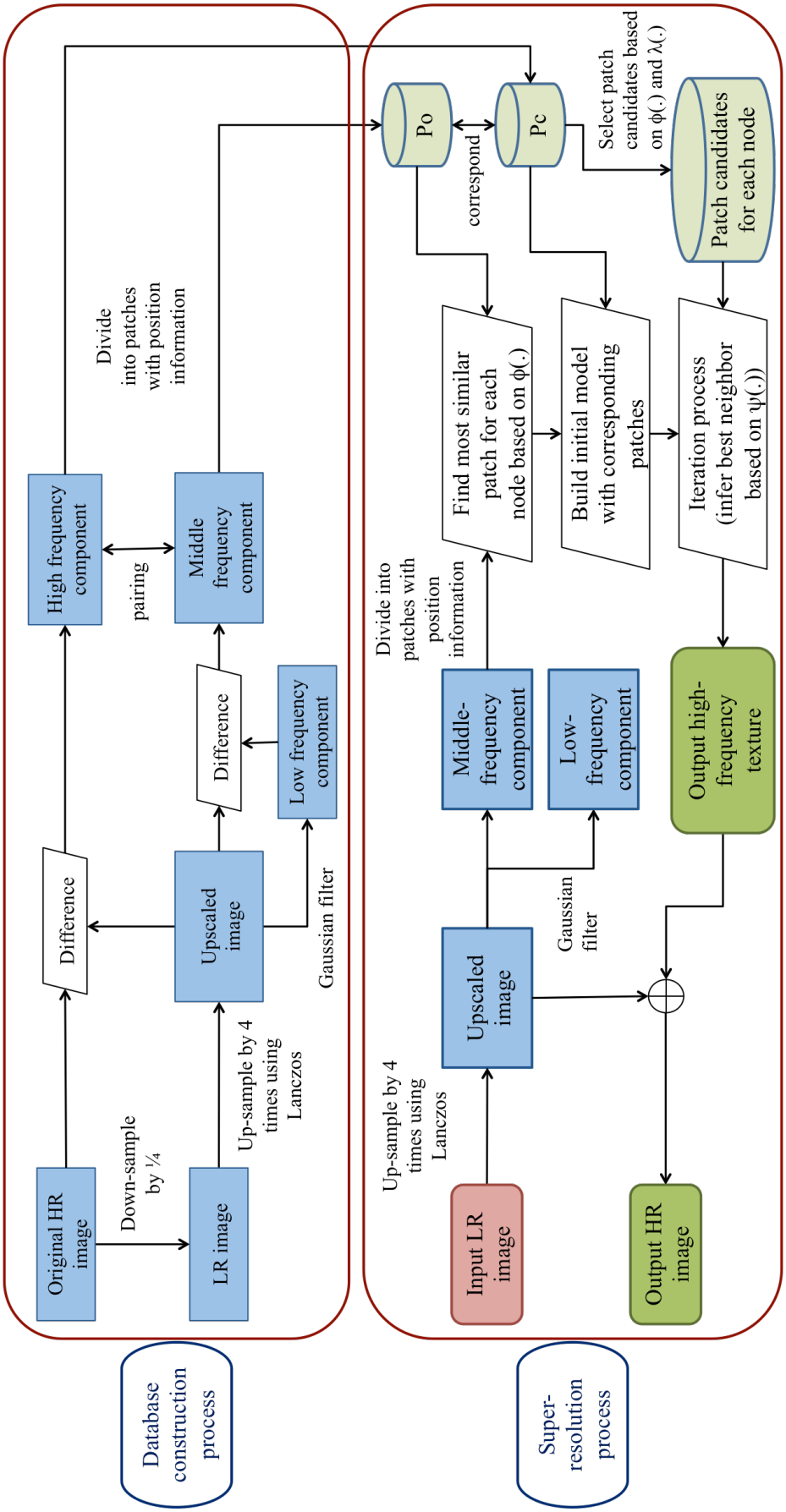


Figure 4.5 Master algorithms for face image super-resolution

Iteration process concerns the border compatibility, ψ(.) value. X^C in Equation (4.4) is replaced with the selected patch candidates in sequence for each node until P(X|Y) shows no improvement or reach optimum value. $P_C(X|Y)$ is the known probability value from Equation (4.3), which has different value for each patch candidates.

$$P(X \mid Y) = \prod_{(u,v)\in NB(i,j)} \psi\left[X^C(i,j), X(u,v)\right] \times P_C(X \mid Y) \quad (4.4)$$

We replace the initial chosen patches with the best patches that are compatible together with the neighboring patches (patches with the best ψ(.) values) amongst the patch candidates on all nodes. First iteration is done when all nodes have been processed. The image resulting from the first iteration is smoother than that from the initial image model because patches were connected to each other better than before iteration process. We carry out the same procedure iteratively until the P(X|Y) value becomes optimum.

After iteration process, the reconstructed texture component will then be synthesized with the initially enlarged image before (from Lanczos resampling) to generate the final resulting HR image.

4.1.5 Resulting Image Evaluation Method

Performance of our proposed method is evaluated based on how closer the resulting HR image to the ideal output image. Ideal output image is the original HR image of a test sample before it was down-sampled or degraded. In image processing practice, there are several error metrics were introduced to measure the degree of dissimilarity or error between the two images, in terms of image's pixel value. One of the most commonly used error metrics is peak signal-to-noise ratio (PSNR).

PSNR is most easily defined by mean squared error (MSE). Given an original image, O(i,j) and a processed image of which we want to evaluate, P(i,j), with MxN is the image size in pixels, MSE is defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [O(i,j) - P(i,j)]^2 \quad (4.5)$$

The MSE represents the cumulative squared error between the processed and the original image, whereas PSNR represents a measure of the peak error. The PSNR (in dB) is defined as:

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right). \qquad (4.6)$$

Here, MAX is the maximum valid value for a pixel. For an 8 bits image, the maximum value for the data is 255. The higher the PSNR, the better the quality of the reconstructed image.

Quality of a resulting image can also be assessed by eye alone, by looking and comparing the sharpness of texture or edges with those in ideal image, but in most cases, it is difficult to distinguish between them especially when we want to estimate quality of resulting images from several methods. In our case, we will compare the performance of our method with previous Freeman's method and conventional Lanczos resampling method, based on PSNR evaluation.

4.2 Experimental Results and Discussions

4.2.1 Experiments Setup

As the proposed method's algorithms involve a lot of variable parameters and options to be considered, which would leads to vary possible outcomes, it is recommended to find the most preferred values of those parameters before we carry out the main experiments. The important parameters that have significant role in the method include size of patch, number of training images in database, times of iteration, value of σ for Gaussian filtering and value of weighting factor, α in facial-part's compatibility function. Since this experiment's objective is to assess the effectiveness of our proposed facial-part's compatibility function, we let value of α be variable from 0.0 to 1.0 with 0.1 increments (see Section 4.1.3).

Previously, we have assessed the optimum number of training images needed in database (see Figure 3.4), i.e. 70–110 face images. Regarding the size of patch, we typically use 5x5 in pixels for now. The detailed assessment on the best setting of patch size will be presented and discussed later in Section 4.4. Here, we conducted preliminary experiments to find the best number of iterations and best value of σ for Gaussian filtering using Freeman et al.'s method (when α is 1).

(1) Times of iteration process

Iteration process refers to repetition of inferring process, where patch on each node was replaced with patch from a set of selected patch-candidates consecutively to find the best matched patch with adjacent or neighboring patches (see Section 4.1.4). The number of selected patch candidates for each node is 110 patches. We raster-scan starting from bottom to top and left to right. For instance, at a current inferred node's position, we search and replace within 110 patch-candidates one by one to find the best compatible patch in terms of patch similarity, $\phi(.)$, facial parts compatibility, $\lambda(.)$ and overlap border compatibility, $\psi(.)$. First iteration is complete when all nodes were inferred. Resulting face image after first iteration will visibly better than before iteration because patch with patches in adjacent nodes had been connected as if they are stitched to each other thanks to the operation of $\psi(.)$.

But how many times of iteration is necessary? Generally, one time of iteration is sufficient but we insist to find out how much different would it make if the times of iteration is increase. So we ran the method on a sample of face image, with 1, 3, 5, 10, and more times of iteration to observe the improvement of resulting PSNR. The assessment results are given in Table 4-2, and their graph is plotted in Figure 4.6, with comparison to resulting PSNR for Lanczos resampling result.

Table 4-2 PSNRs of resulting image for different times of iteration

Example-based		Lanczos resampling
Times of iteration	PSNR [dB]	PSNR [dB]
0	26.334	27.025
1	29.261	
3	29.615	
5	29.639	
10	29.654	
15	29.654	
20	29.654	

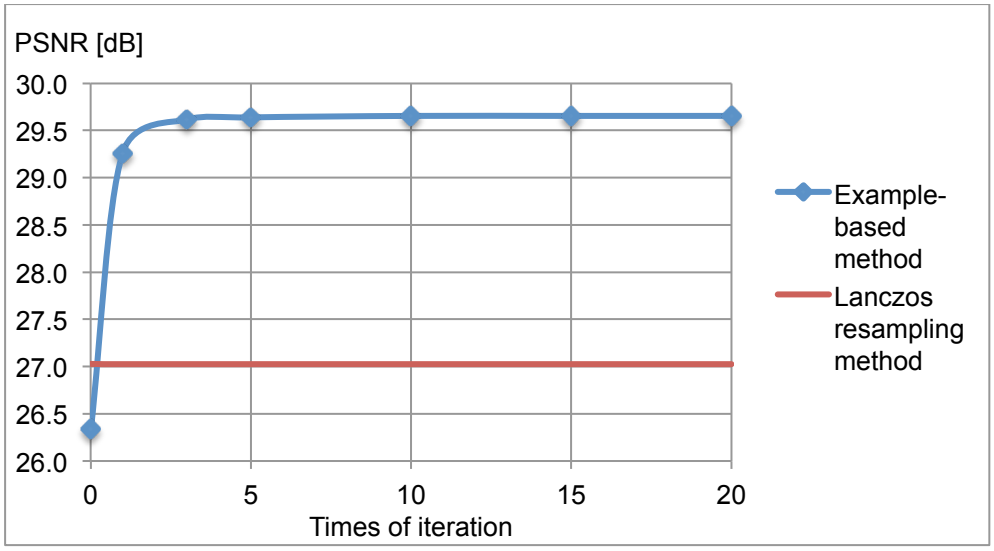


Figure 4.6 Resulting PSNRs for different times of iteration process

From the PSNRs assessment, one time of iteration is sufficient to achieve a better resulting image than Lanczos resampling method, indeed. Whereas when no iteration process was applied, the resulting PSNR is worse than the resulting PSNR from Lanczos resampling method, proving that iteration process is absolutely necessary. This is because before iteration process, patches were selected by not considering border compatibility yet, so they are not well connected to each other causing block artifacts and grainy appearance, as can be observed in images of texture (high-frequency maps) given in Figure 4.7. With meticulously selected patches, such noisy elements can be greatly reduced as we can observe in texture image below.

Before iteration process | After one time of iteration | After 10 times of iteration

Figure 4.7 Texture image (high-frequency maps) of a face sample before and after iteration process

As the number of iteration reaches 10 times, no improvement is observed onwards. Henceforth, we will run 10 times of iterations for execution of our proposed method in all experiments. Even though one time of iteration is sufficient, the different between one and 10 times of iteration is merely one second of processing time, so it is still reasonable to do iterations as much as it provides improvement of PSNR.

(2) σ value for Gaussian filter

In the proposed method's algorithm (see Figure 4.5), Gaussian filter is used to extract texture (middle-frequency component) of an initial up-scaled image, during both process of database construction and super-resolution. The difference (or subtraction) of a filtered image and its original image will generate texture component. Patches of this texture will be observed during the search and match routine in the method.

In Gaussian filter, the σ parameter plays an important role (this is discussed in Chapter 2, Section 2.2.2) to determine strength of filter applied on an image. Higher σ value makes a stronger filter, generating a lower frequency image. If we subtract the filtered image from the original ones, we will get a high-frequency image representing the image texture. Different σ would generate different amount or degree of texture. For example, in Figure 4.8, texture component appears thicker when the σ value is larger.

$\sigma = 1$ $\sigma = 2$ $\sigma = 3$ $\sigma = 4$

Figure 4.8 Gaussian filtered face image with different σ

Different sharpness of extracted texture yields different results of super-resolution process. It is better to find in advance the best σ before we conduct the main experiments. So we ran the SR method with same conditions: size of patch is 5x5, α is 1, number of training images in database is 110 face images, on the same face sample before, with different σ value in the filtering process. We evaluated their

PSNR improvement to find the best σ value for Gaussian filtering. According to resulting PSNRs in Table 4-3, the best σ value to be used in main experiments later is 2.

Table 4-3 PSNR values for a resulting image when different σ value was applied

σ value for Gaussian filtering	1	2	3	4	5
PSNR [dB] of resulting image	29.596	**29.654**	29.372	29.330	28.976

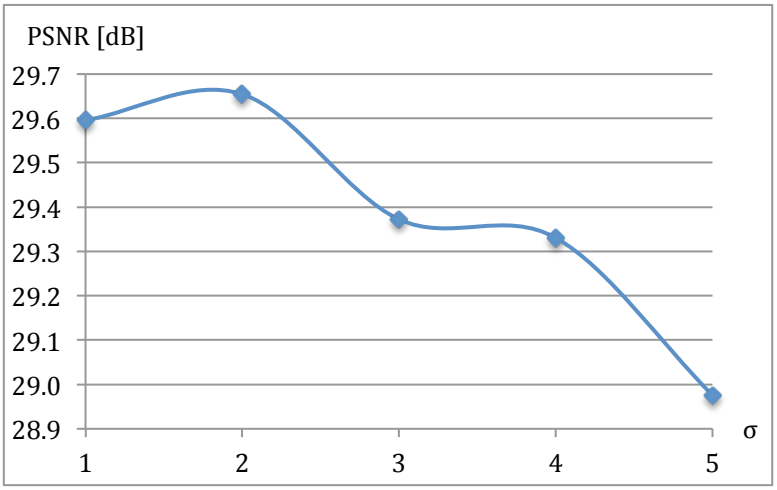


Figure 4.9 PSNR evaluation graph for resulting images of different σ

4.2.2 PSNR Assessments and Resulting Images

A set of face images taken from 110 subjects (see Appendix A) was utilized for the database, among whom 74 were males and 36 were females. All the original HR face images used in the experiment were in grayscale, and their size is 288x240 in pixels. These HR face images were down-sampled to generate their corresponding LR images using Lanczos resampling, by quarter-size of the number of pixels on each dimension, i.e. 72×60 pixels. We then up-sampled back these LR images to their original size to get initial enlarged images. The HR and LR (initially up-scaled) images were divided into patches of 5x5, and stored in pairs with information of patch position.

Experiments with the proposed method were conducted on six samples of face images shown in Figure 4.10. These six samples are not included in database. We execute the proposed method with different values of weighting factor, α to evaluate

its performance and effectiveness. Peak signal-to-noise ratios (PSNRs) are employed to estimate quality of super-resolved images to assess the performance of the proposed method. The higher PSNR values indicate better results.

Figure 4.10 Input LR face images

The PSNR evaluations' graph against weighting factor, α is shown in Figure 4.11. If α is zero, the algorithm will only take into considerations the facial part function, or else, if α is one, it will not take into account the function, which is same as that in Freeman et al.'s method. PSNR evaluation results indicate that as the α value was between 0.5–0.9, the PSNR value achieved better results than Freeman et al.'s results (when α is 1), which simultaneously proves that our proposed function is effective.

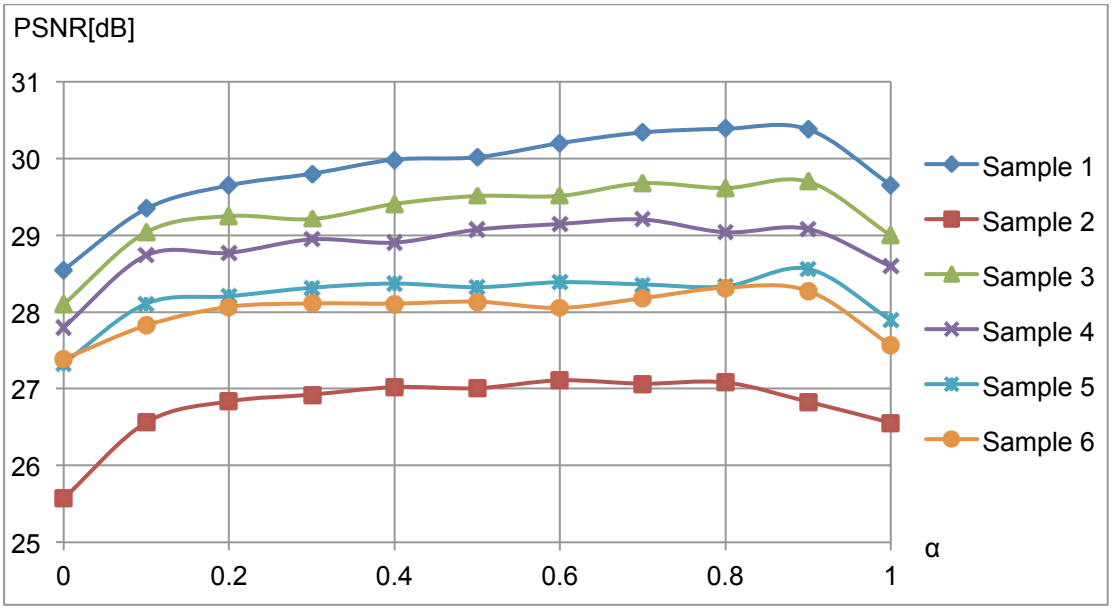


Figure 4.11 PSNR for resulting images of different samples against α parameter

The PSNR assessment results are listed in Table 4-4, with resulting PSNR for Lanczos resampling and Freeman et al.'s method for comparison. Table 4-4 indicates that the proposed method (when the best α value was applied) achieved the highest PSNR for all six samples.

Table 4-4 PSNR assessment for six samples

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
Lanczos method	27.03	24.88	26.28	25.40	25.29	25.62
Freeman et al.'s method	29.65	26.56	29.00	28.59	27.89	27.57
Proposed method	**30.39** (α = 0.8)	**27.11** (α = 0.6)	**29.70** (α = 0.9)	**29.20** (α = 0.7)	**28.56** (α = 0.9)	**28.32** (α = 0.8)

Figure 4.12 shows resulting images for Sample 1 to 6 with a scaling factor of four. Figure 4.12(a) has input LR images with 72×60 in size, and Figure 4.12(e) has original HR images with 288×240 in size for each sample. The resulting images of the Lanczos method, Freeman et al.'s method, and the proposed method are provided in Figure 4.12(b), Figure 4.12(c) and Figure 4.12(d), respectively. Boxed image on the right side of each resulting images of the first sample is the zoomed up version of the marked regions, where the details of the results of all methods can be observed. The Lanczos interpolation method blurred most of the details of facial images in Figure 4.12(b), especially along the face lines. The super-resolved results with Freeman et al.'s method and the proposed method in Figure 4.12(c) and Figure 4.12(d) are significantly sharper than those from interpolation approaches. This is due to the reason that patches were used as process units instead of pixels and the correlation of adjacent patches was taken into consideration by the learning model.

We have also executed the proposed method on more face samples (30 samples including the six samples) for assessment. Appendix B gives the images of 30 faces in original HR size (288x240 in pixels), while their PSNR results under different α value are compiled in Appendix C, which will be referred later for experiments in Section 4.3.

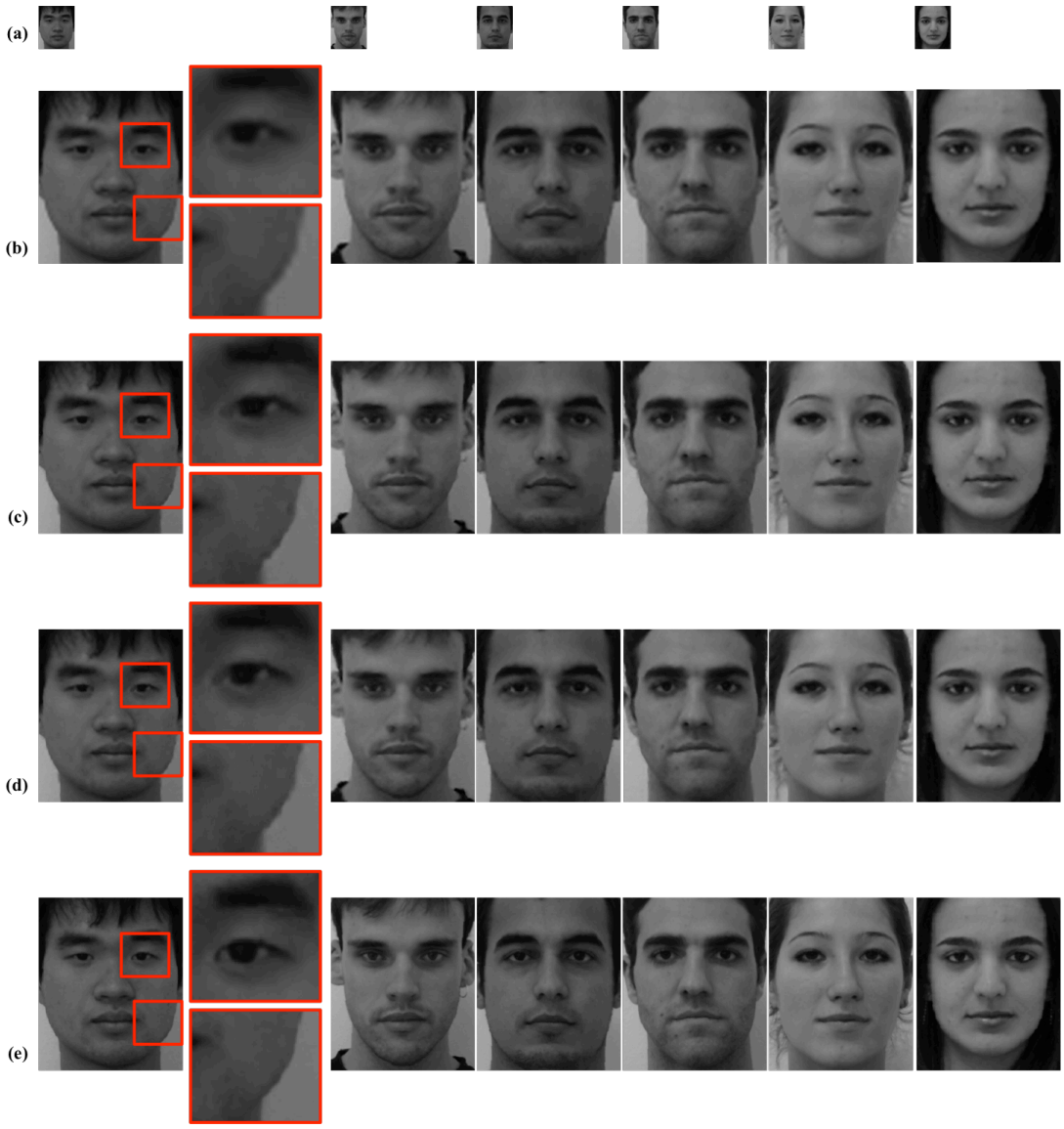


Figure 4.12 Resulting images for enlargement by four times of scaling factor: (a) input LR facial images; (b) Lanczos resampling method; (c) Freeman et al.'s method; (d) the proposed method; (e) original HR facial images.

4.2.3 Comparing Quality of Texture

We observed the resulting images from the proposed and the Lanczos methods. As we can see from Figure 4.13, specifically in the boxed region, the proposed method generates sharper and clearer edges than those with Lanczos resampling. The main reason for this is that SR methods manage to predict the missing high-frequency information using external images, preserving sharp edges and image details.

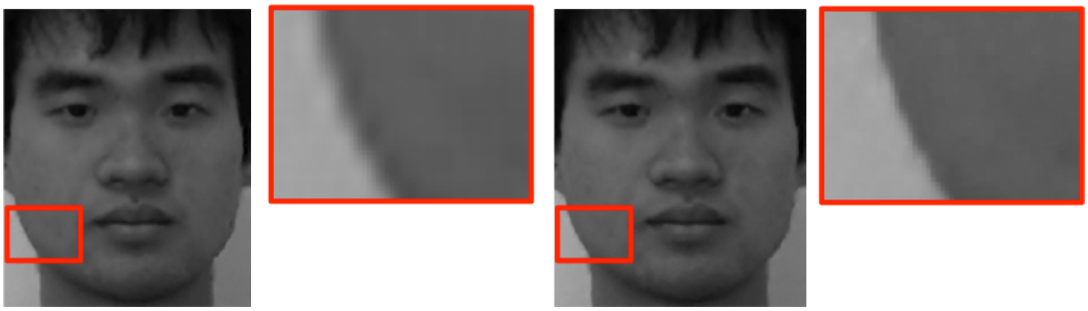


Figure 4.13 Comparison between Lanczos resampling and proposed method's resulting images

We also analyzed the resulting image with Freeman et al.'s method. Freeman et al.'s and the proposed methods yielded comparable resulting images. However, Freeman et al.'s method could mistakenly predict output patches in some places. Figure 4.14 shows that unnatural artifacts were observed in circled area in the resulting image with Freeman et al.'s method, while the resulting image from the proposed method is free from such artifacts or noise, and looked more natural. This is because our proposed method not only took into consideration the similarity between patches' pixel values, but also took into account which part of the face those patches came from.

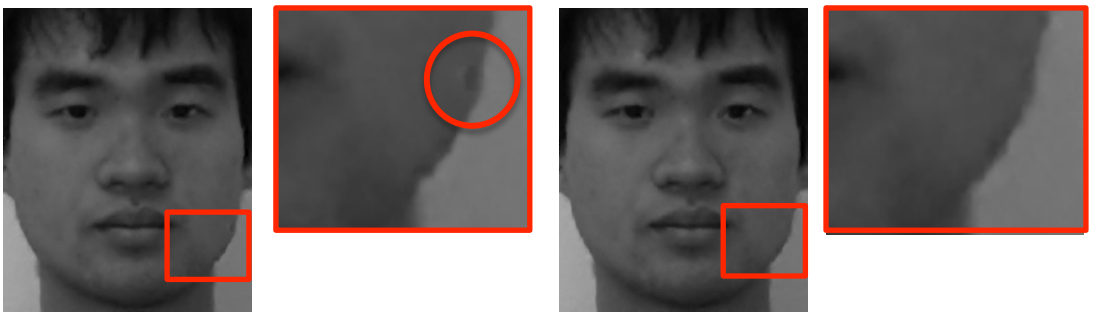


Figure 4.14 Comparison between Freeman et al. and proposed method's resulting images

4.3 Experiments on Various Conditions of Face Images

Until now, we have performed the proposed method onto normalized face images as the test images. In this section, we want to take a challenge by conducting further experiments on various conditions of face images. We want to observe the effectiveness of our method in different circumstances. How is the method's performance if we use a random facial image that is not normalized as sample, while the same database of normalized images is utilized in the process? For the sake of

experiment, we used a variety set of face samples such as shifted face, magnified face and different angle of face as test images.

Before we conduct the experiment, it is necessary indeed to fix the α parameter in advance, so that we would not have to observe every α value for each sample. Therefore, we performed proposed method on 30 samples of normalized face samples to find the average PSNRs for each α value.

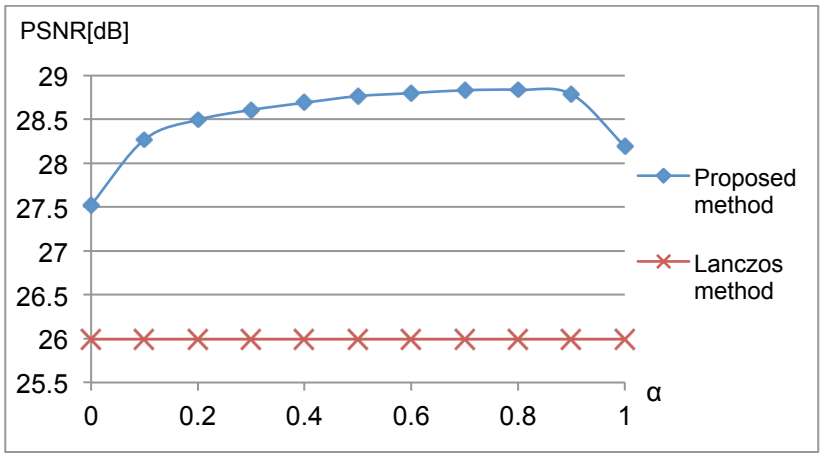


Figure 4.15 Average PSNR value from 30 samples against α

Figure 4.15 presents the average PSNR from 30 output images. The best output image for each sample has different α value. We take the average of best α value from each sample to be the fixed value. The average of the best α value for 30 samples is 0.71 (see Appendix C).

We performed the algorithm with α value is fixed as 0.71, to compare with Freeman method i.e. when α = 1, on another different 30 samples. Each output image from proposed method for those 30 samples give better PSNR than Freeman et al. results. Table 4-5 indicates that even with the fixed α value, our proposed method shows improvement with better PSNR result in average than Freeman et al.'s result. Improvement value in average is 0.61 [dB].

Table 4-5 Average PSNR improvement for 30 samples between Freeman et al. and our proposed method

	Freeman et al.	Proposed	Improvement
Average, μ	28.271 [dB]	28.881 [dB]	+0.610
Standard deviation, σ	1.292	1.294	0.203

4.3.1 Shifted Face Image

Figure 4.16 shows samples of shifted face images. We shifted the face to the left and right sides by 5%, 10%, 15% and 20% pixels respectively. We down-sampled each samples into quarter-size of original size by Lanczos resampling to create input images. By using the same set of normalized database, we performed the Lanczos, Freeman et al. and proposed method to get HR resulting image for each samples.

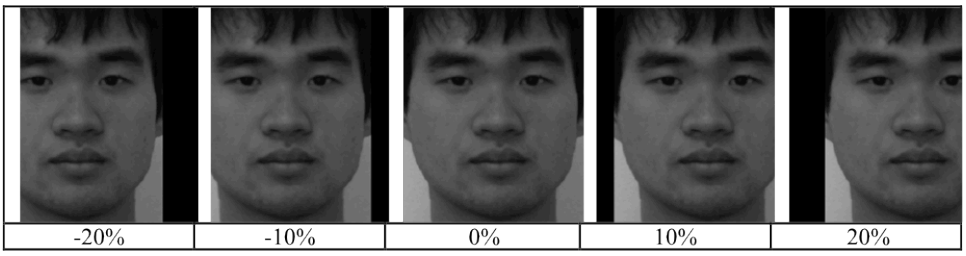


Figure 4.16 Shifted image example

Figure 4.17 outlines the PSNR graph for resulting images from those shifted face samples. The graph indicates that with our proposed method, when α is fixed as 0.71, gives better results in spite of shifting occurred. But, it become worse as the image is shifted, since the position of face features have changed. Freeman et al. and Lanczos method gives nearly same PSNR value even the position of face is shifted, since both methods do not take patch position into consideration. While α is 0, PSNR value is lower than PSNR from Freeman et al. method and becoming worse when the face position is shifted. Therefore, it is necessary to consider the similarity of patch, ϕ(.) as well as we consider the category or position of patch, λ(.).

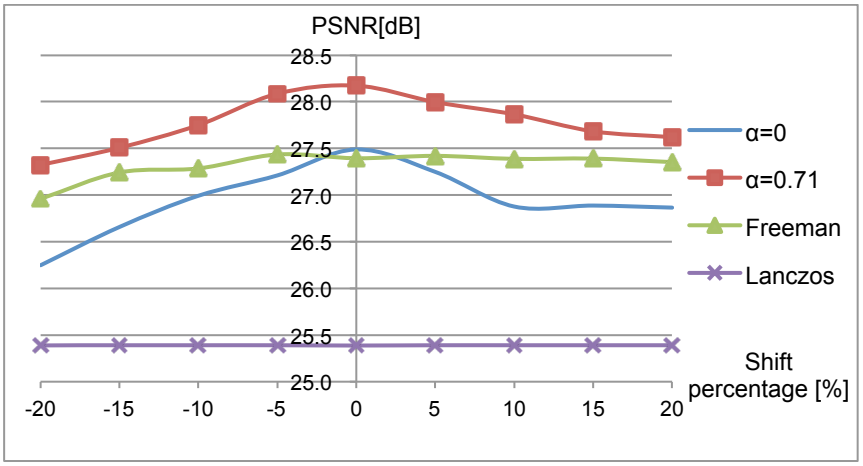


Figure 4.17 PSNR graph against shift percentage for Freeman et al., Lanczos and proposed method (when α = 0 and 0.71)

4.3.2 Magnified and Shrink Face Image

Figure 4.18 shows magnified and shrink face image samples with different percentage. The facial parts in the magnified face, such as eyes appear bigger than those in the original face image (0% of magnification). We execute our proposed method on those images and measured the PSNR for their resulting images.

For a fair evaluation, we only evaluate the part of face in boxed area, since some images' facial parts were cropped due to magnification. For example, face image of 30% magnification loses some parts of texture of chin and hair, compare to the original size of face. The existence and absence of such texture would affect the PSNR assessment, differently. Thus, it is important to standardize the part of image, which we want to evaluate.

The PSNR results are shown in Figure 4.19. PSNR graph shows that even for magnified or shrink face, our proposed method achieves better results than Freeman et al. and Lanczos resampling method. We assume that when the face is magnified, minor details of texture in patches will decrease or become less complex since the size of patch is same. Therefore, the magnification did not give any unwanted effects to the results, but conversely has better results than the original size. On the other hand, when the face is shrink, details of texture in a patch increase or become more complex, causing the declination of PSNR.

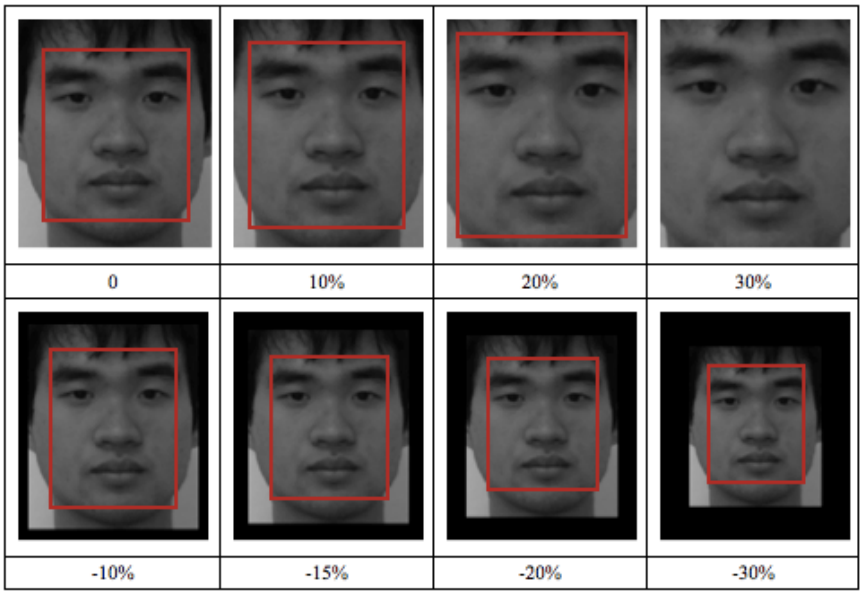


Figure 4.18 Magnified and shrink face samples

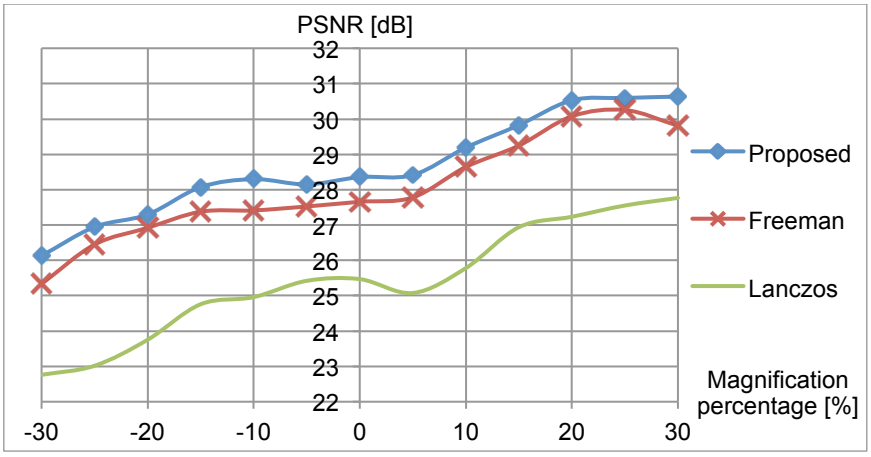


Figure 4.19 PSNR results for magnified and shrink face samples using Lanczos resampling, Freeman et al. and proposed method

4.3.3 Different Direction of Face

Figure 4.20 shows samples with different facing direction. We execute those three methods on samples below to compare the results.

Figure 4.20 Different direction of face samples

Figure 4.21 shows PSNR comparison between those three methods. Our proposed method gives better results for every sample when α is 0.71. When α is set as 0, where only patch position is considered, the resulting PSNRs for every sample are lower than PSNR result from Freeman et al. method. Nevertheless, our proposed method gives better PSNR than the PSNR results from Lanczos method.

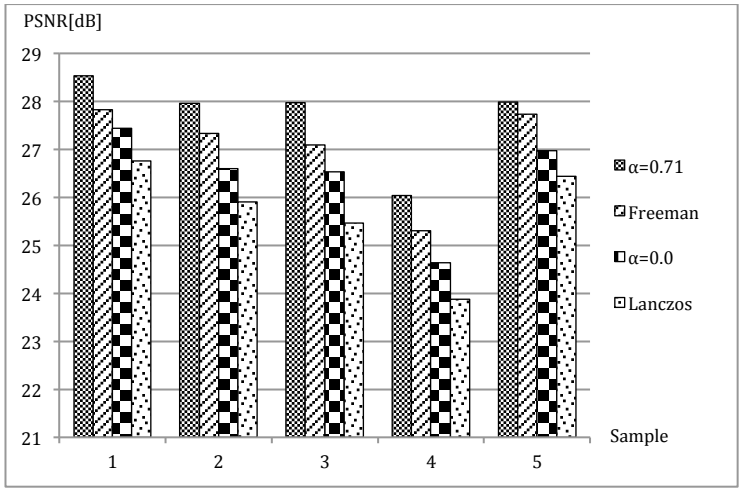


Figure 4.21 PSNR results for five different directions of face samples using Freeman et al., Lanczos and proposed method

4.4 Adaptive Patch Size to Scaling Factor

There are many variable parameters in this method, which may lead to varied possibility of resulting images. Some of the main parameters are image patch size, weighting factor in compatibility functions, number of training images in database and scaling factor of magnification.

Till now, the size of patch used in example-based methods is typically 5x5 or 7x7 in pixels, and the scaling factor is set typically as 4 times enlargement (16 times the total number of pixels of LR image). However, there is no thorough analysis yet regarding those two parameters, i.e. patch size and scaling factor. Is it really the best patch size for the SR process? How about other cases of magnification, where their scaling factor is different? What is the most suitable setting of patch size when we want to enlarge an LR image by different scaling factor, e.g. 3 times or 5 times?

It is important to find the most suitable parameter's value in order to make the application more convenience, stable or less sensitive, besides user-friendly. Therefore, we attempt to perform an in-depth analysis of the super-resolution algorithms in terms of patch size and scaling factor. Our aim is to find the most appropriate patch size to be set in the process, which may provides better results, under different cases of magnification.

4.4.1 Parametric Analysis

We conducted an experiment by demonstrating the SR algorithms onto several sets of input LR images that have different scaling factor, i.e. 2, 3, 4, 5, and 6 times respectively. We used different size of patch during SR process, i.e. 5x5, 7x7, 9x9, and 11x11, and applied them onto each set of images to reconstruct HR images. The overlap region between patches is one pixel. Since the image size must be fit or compatible to the scaling factor and patch size to avoid any unnecessary remain part of image during SR process, we beforehand processed all the original HR face images by adjusting them to a desired size. The adjusted sizes of HR images are respectively shown in Table 4-6.

Table 4-6 Adjusted size of HR images [(width) x (height)] to be fit with scaling factor and patch size setting.

Patch Size	Scaling factor				
	2	3	4	5	6
5x5	240x288	240x288	240x288	240x280	240x288
7x7	240x288	240x288	240x288	240x300	240x288
9x9	240x288	240x288	240x288	240x280	240x288
11x11	240x280	240x300	240x280	240x280	240x300
13x13	240x288	240x288	240x288	240x300	240x288

Given an image's original size is 240x288 and scaling factor is 2, the size of downscaled image (LR image) is 120x144. We generated training face images and input samples (HR and LR) accordingly to Table 4-6, producing 25 sets of training images database and input samples. Each set of training images database consists of 110 face images, while the set of input LR images (different face images from those in database) consists of 30 samples.

We ran the proposed method onto those sets of input LR images to reconstruct sets of resulting HR images, respectively according to image size setting and scaling factor in Table 4-6. Each set is processed using different patch size as illustrated in Figure 4.22. As a result, 25 sets of resulting HR images were generated. We employed peak signal-to-noise ratios (PSNRs) to evaluate the super-resolved images for assessment. For overview analysis, we observed the average of resulting

PSNRs of each set of 30 output HR images. The resulting average PSNRs are shown in Table 4-7.

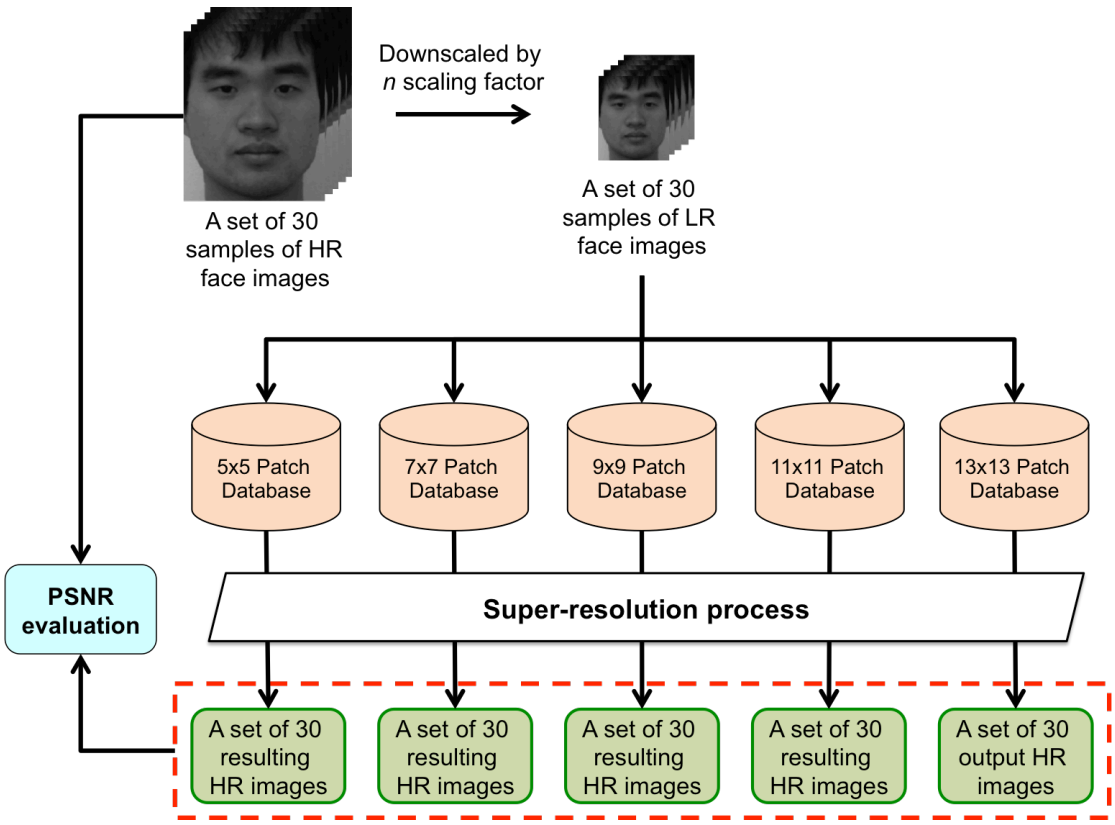


Figure 4.22 PSNR assessments process (set by set)

Table 4-7 Average PSNR[dB] for sets of 30 output HR images

Patch Size	Scaling factor				
	2	3	4	5	6
5x5	**35.973**	31.112	28.913	25.572	24.103
7x7	35.852	**32.027**	28.487	26.123	25.624
9x9	35.600	31.498	**29.262**	26.523	24.904
11x11	35.280	30.924	28.506	**27.070**	25.163
13x13	35.221	31.442	28.898	26.578	**25.718**

The bolded values note the best patch size for each case of magnification. We can observe that in Table 4-7, there is a consistency of relation between the best patch size and scaling factor. Surprisingly, the best patch size is adaptable to the scaling factor of magnification. Let the scaling factor is n, we can conclude that the best setting of patch size for each scaling factor can be expressed as $(2n+1)^2$. With

this formula, we can now easily determine the size of patch to be used in the proposed SR method when we want to enlarge an LR image under a certain scaling factor.

4.4.2 Improved Algorithms

We employed the formula into the SR method. The proposed algorithms can be simply illustrated as in Figure 4.23. The bolded lines indicate the preferred flow of SR. For example, if we want to magnify an LR image by 3 times, the algorithms will select the most suitable database, i.e. 7x7 patch database, to be used in SR process, yielding the best possible output HR image.

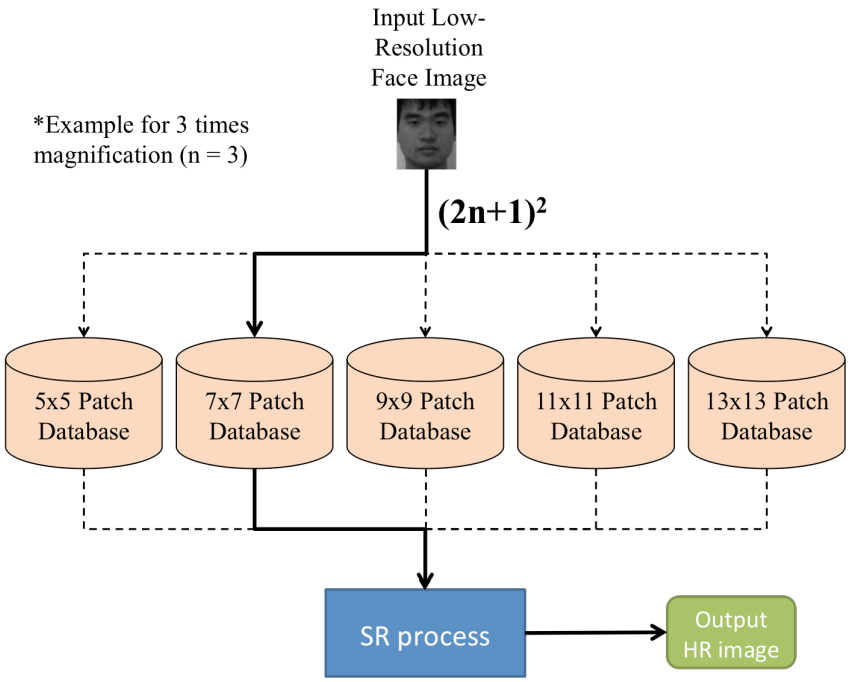


Figure 4.23 Algorithms for the using of the best patch size in SR process

4.5 Chapter Summary

In this longest chapter of the thesis, a novel method specifically for face image super-resolution is presented. We have proposed an innovative example-based method to predict missing texture in face image by taking their facial parts into consideration during patch selection process. First, training database was constructed using a set of normalized facial images, which then were divided into patches and stored with information on their original position on their image. Second, Markov random field was used to model patch structures for a super-resolved image,

by incorporating the proposed facial parts compatibility function in the learning model.

Compare to Freeman et al.'s method and Lanczos resampling method, the proposed method performed better in terms of PSNRs. Texture along the edge of face in resulting image from proposed method are much sharper and clearer especially when we compare to Lanczos resampling image, emphasizing that our proposed method manage to well restore the missing texture.

We have also found the best α, in average, for the proposed method. Even with fixed α, i.e. 0.71, the proposed method achieves better resulting PSNRs. Furthermore, experiments on random face images which are not normalized has clarified that when α is 0.71, the proposed method still provides better or comparable results as normalized face sample.

In addition, an in-depth analysis of face image super-resolution method was conducted, concerning the patch-size setting and scaling factor. We conducted an experiment using different setting of patch size and scaling factor, to find the best patch size to be used when performing the SR method. We also have analyzed their relationship with scaling factor of enlargement. According to resulting PSNRs assessment, interestingly we found that the best patch size is adaptable to the scaling factor. The best setting of patch size for can be determined by $(2n+1)^2$, where n refers to scaling factor. Since the proposed method involves many variable parameters, which may lead to varied possibility of results, this finding would contributes to make the application of the method more stable and convenience. By using the algorithms to determine a patch size for SR process, we have narrowed down the ways of finding the best possible resulting HR image for the method.

CHAPTER 5

COMPRESSED FACE IMAGE RESTORATION

Since surveillance cameras records long-term footage, practically the footage is being compressed due to memory constraint. Consequently scenes from footage usually suffer with block noise from such compression. In order to obtain better information from the footage, it is necessary to restore the degraded compressed image by reducing the block noise and restore image texture. Gaussian filter is one of the Point Spread Function (PSF) that has been commonly adopted in image enhancement to simply filter out noise component in an image (this is discussed in more detail in Chapter 2, Section 2.2).

The problem of filtering method is texture component will be indiscriminately filtered out together with noise component, causing a blur image. It gets blurrier if the filter is applied stronger, especially in case of restoring a highly compressed image, where a stronger filter is required to reduce the heavy block noise. In this chapter, an image restoration method using example-based approach to enhance quality of a compressed face image is proposed.

5.1 Block Noise Reduction for Face Image Restoration

In example-based tradition, the database should only include images that are similar type with input image for better probability of similarity among textures. The key idea of our method is to further categorize patches based on facial parts, i.e., as a database of eye patches, a database of nose patches and so on. Since learning-model already includes λ(.), multiple databases of facial parts are unnecessary.

Previously, we have proposed a method for example-based super-resolution by taking facial parts compatibility, λ(.) into account, to restore texture in a high-resolution image generated from a targeted low-resolution image. Here, we want to apply the same approach, but this time our target is a compressed (block noised) image. We aims to reconstruct back distorted face texture in a compressed image, which cannot be done by Gaussian filter.

We incorporate the idea from Chapter 4 into image restoration for block noise reduction, but this time we used original HR face images and their degraded images as training set. We have proposed two methods of restoration, one is named as Direct method and the other one is Smooth method.

5.1.1 Proposed Restoration Methods

We used a collection of normalized high-resolution face images to construct a database. The normalized training images have the same size and ratio of facial features, where facial-feature points (e.g., eye, nose, mouth, chin, and face boundary lines) in each image are at approximately closer positions. Unlike the method in Chapter 4, this time we utilized degraded images instead of LR images, as shown in Figure 5.1. So we applied JPEG compression to HR images to create their corresponding degraded images. Then, each patch's original position in the training images, where (k,l) is training-patch-position coordinate, was observed later in the proposed learning-model function during restoration process.

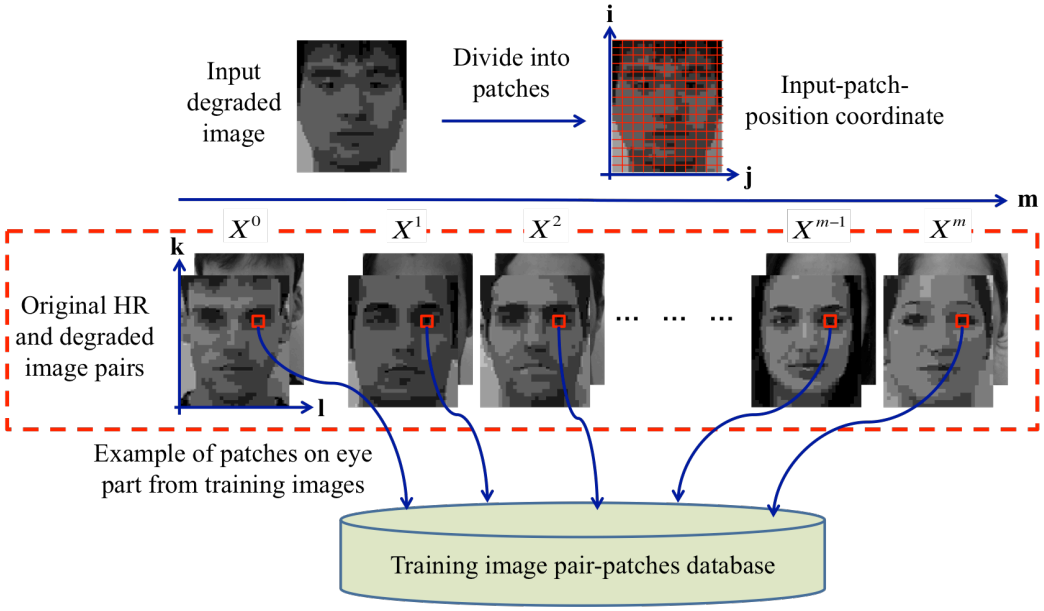


Figure 5.1 Training database comprising multiple compressed face images

5.1.1.1 Direct Method

The training images in database need to be pre-processed to extract high-frequency information so that only texture is observed. There are several methods of texture extraction in image processing that have led to the proposal of several example-based super-resolution methods, as mentioned in Chapter 3.

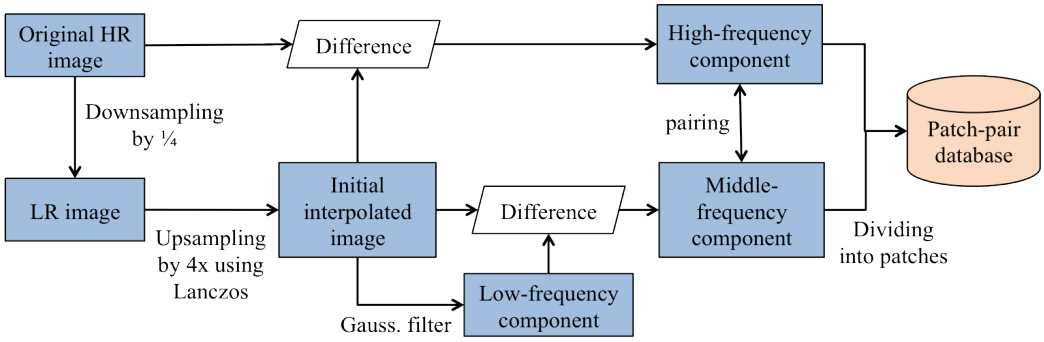


Figure 5.2 Previous super-resolution method's database construction

Figure 5.2 shows how a training patches database was constructed before in our proposed super-resolution method. We applied a Gaussian filter onto initial enlarged LR image to extract texture (middle-frequency component), while texture in HR image (high-frequency component) is its difference with the enlarged image. Both texture components were then divided into patch pairs.

In present chapter, our targets were different from those of previous chapter. The objective now is to restore texture in a compressed (block noised) image. We adopted the same process as super-resolution method before, but we used original HR and degraded face images as training set where a degraded face sample is the target.

One of the main concerns in example-based method is, what is the appropriate method to extract high-frequency information to reconstruct output texture? In particular, this time a noisy image is the target. Here, we have proposed two methods of restoration, which have different way of HR image's texture extraction: Direct method and Smooth method.

Direct method uses a straight-forward approach, as outlined in Figure 5.3, which directly takes the difference between the original HR image and its corresponding degraded image to be the high-frequency components. While for degraded-image's texture extraction, we applied a Gaussian filter on the degraded image then subtracted the filtered image from the degraded image to obtain the middle-frequency components, which is similar to the previous SR method. Both high-frequency component and middle-frequency component are stored in pairs in the database.

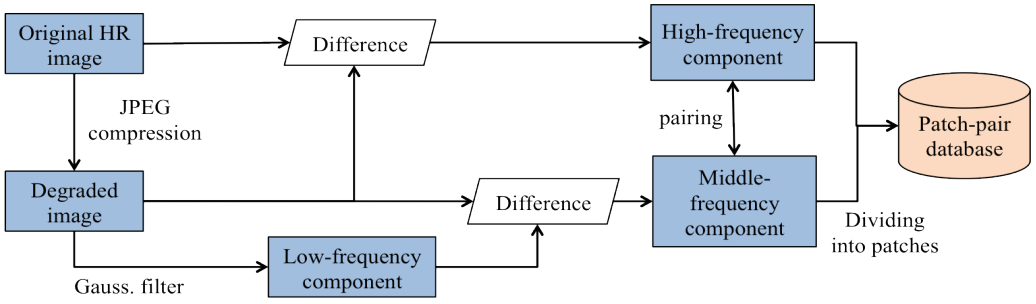


Figure 5.3 Direct method's database construction

Figure 5.4 outlines the overall image-restoration algorithms for the Direct method. The texture components from the degraded input image were extracted through the same process as in the database, which is using a Gaussian filter. We then found a set of patch candidates for each input patch node for the iteration process.

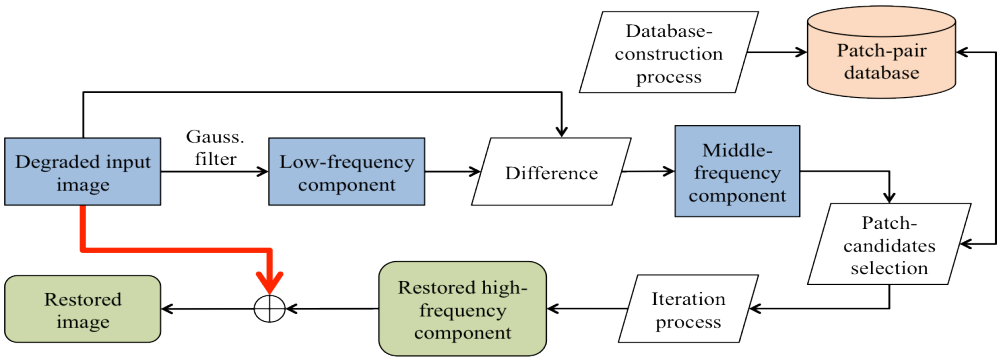


Figure 5.4 Restoration process for Direct method

The MRF network in Figure 4.4 probabilistically models the relation between the input patches and the training patch-pairs in terms of ϕ(.) and λ(.), and between the neighboring estimated HR patches in terms of ψ(.). A set of patch candidates for each input node according to ϕ(.) and λ(.) were selected in advance with weighting factor, α as in Equation (4.3) previously. Then, the only element left to be considered is which patch amongst the candidates is the best neighbor.

Iteration process involves a stitching algorithm that iteratively infers a set of best neighboring patches that have the most compatible ψ(.) values on the overlap region between nodes, as illustrated in Figure 3.3. Before iteration, we created an initial image as the base image by merging the most similar patches (patches with the lowest ϕ(.)) of all nodes. This image would looks uneven and unnatural since the chosen patches would not yet be compatible with their neighboring patches. Hence, the iteration process is executed to reduce the incompatible effects in the image.

Theoretically, the aim with the iteration process is to find a set of X(.), that will improve the joint probability, P(X|Y) in Equation (4.2) as best as possible. By only using a limited number of selected candidates to determine the P(X|Y), the processing time can be reduced. We found the best P(X|Y) to get the best neighboring patches by replacing X(.) with the selected patch-candidates alternately, as has been discussed in Chapter 4, Equation (4.4). The $P_C(X|Y)$ in Equation (4.4) is the known probability value from Equation (4.3), which is different for each patch candidate.

We replaced the initial chosen patches on every node with the best patches among patch candidates that have highest compatibility with their neighboring nodes. The first iteration was done when all nodes have been processed. We carried out the same procedure iteratively until 10 times of iteration as recommended in previous chapter.

Finally, we combined the estimated restored high-frequency component image directly with the degraded input image to obtain a final restored image.

5.1.1.2 Smooth Method

Smooth method uses a different texture-extraction process for original HR image in database construction, as outlined in Figure 5.5. It independently extracts high-frequency component from original HR image by applying a Gaussian filter. Gaussian filter smoothed out the original HR image to obtain low-frequency component. The difference between the original HR image and low-frequency component represents their textures (high-frequency component). As for degraded image, we used the same process of extraction as for the Direct method. Both texture components were then divided into patches in which both patches correspond to each other and store them in the database.

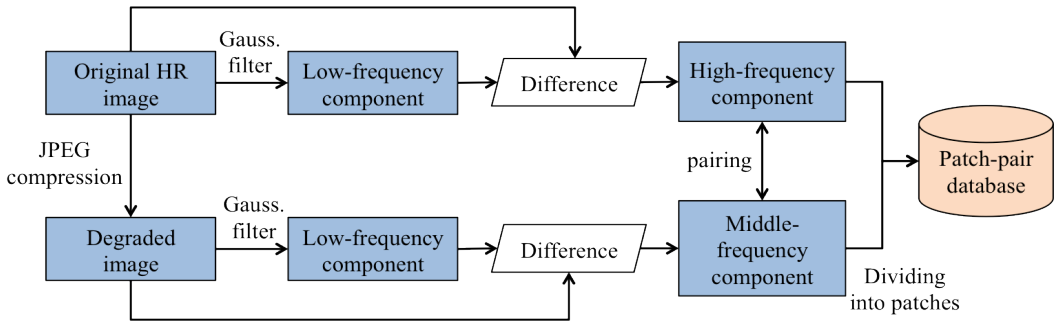


Figure 5.5 Smooth method's database construction

Since the extraction process during database construction is different for both methods, the combined components that reconstruct restored image are also different. For the Smooth method, we combined the restored high-frequency component (restored texture) with the low-frequency component from the degraded input image to obtain a final restored image, as shown in Figure 5.6.

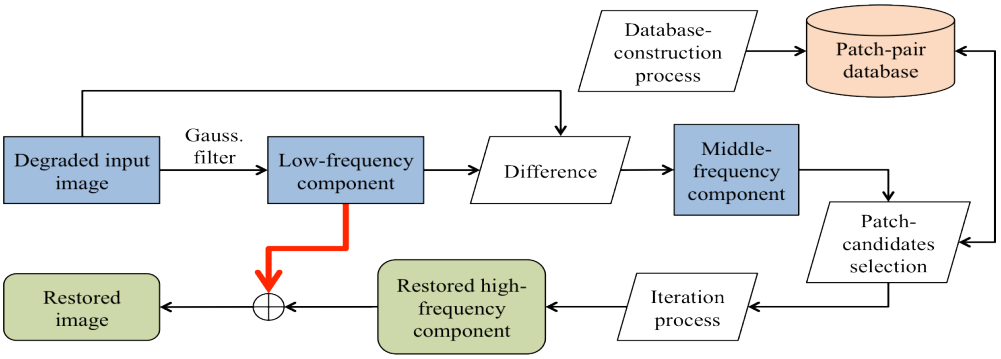
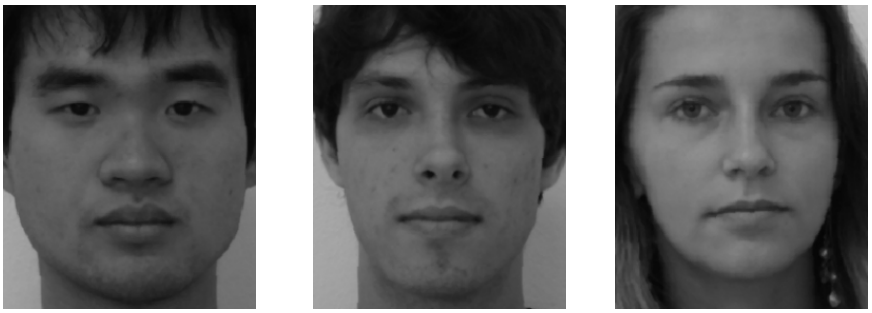


Figure 5.6 Restoration process for Smooth method

5.2 Results and Discussions

For evaluation, we gathered a database comprising normalized face images taken from 100 subjects. All the original HR face images used in the experiment were 288x240 size in pixels. We applied block noise to the original HR images by using a JPEG compression tool with different quality rates between 1 to 13%, resulting in compression ratios from 134:1 to 111:1, to generate six sets of degraded training images. JPEG-block sizes in 8x8 pixels yielded similar noise to a simple eight times magnification of 1/8 low-resolution image, but further degraded because of mosquito and quantization noise. Our main target was the lowest quality, i.e., 1% quality-rated degraded samples. A set of highly compressed face images used in database (100 subjects) and as test samples (30 subjects) are given in Appendix D and Appendix E, respectively. Figure 5.7 presents three examples of the lowest quality rate (1%) of JPEG-degraded face samples.

We evaluated our methods on a set of 30 lowest-quality face images to determine their effectiveness. We ran the iteration process 10 times under different α (0.0, 0.1, …, and 1.0), and Gaussian parameter's σ values was changed from 4 to 12 to find the best resulting images. We used PSNRs to evaluate the resulting images and compared them to the original ones (ideal result) to assess the performance of our methods.



Sample 1 Sample 2 Sample 3

Original face samples

Sample 1 Sample 2 Sample 3

Degraded input face images

Figure 5.7 Original and degraded test images

5.2.1 Fixing σ and α Value

It is necessary to fix the σ and α for assessment. To find the best σ for the Direct and Smooth methods, we ran the restoration process on 30 input images of lowest quality samples under different σ to find their average PSNRs.

Figure 5.8 shows the average PSNRs for 30 samples: input images, Gaussian-filtered images, and best output images from our methods (when best α was applied) for the Direct and Smooth methods for different σ. Gaussian-filtered blurred images and our methods' output images effectively exhibited higher average PSNRs than the input images. However, the higher the σ, the lower the PSNRs value of the Gaussian-filtered images. We can observe that PSNRs fell drastically when σ was more than 5. Conversely, those for the output images from both of our methods increased gradually.

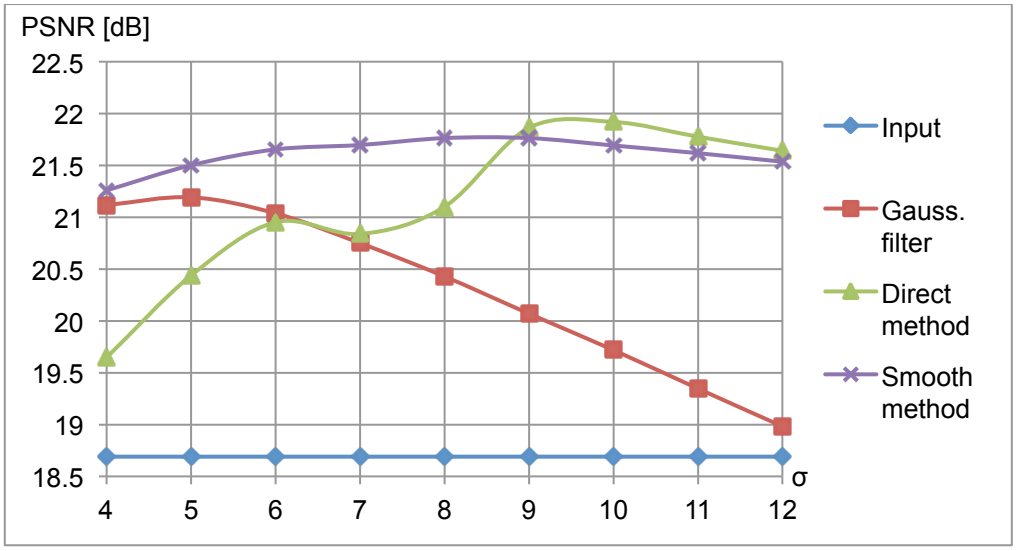


Figure 5.8 Average PSNRs of 30 samples: input image, Gaussian-filtered image, and Direct and Smooth methods' output images

Even the Gaussian filter has significantly reduced noise, unfortunately the filtered image losses their important high-frequency texture component. The image becomes too blurry when σ was 12. Conversely, the output images with our methods' exhibited higher PSNRs since they also included predicted high-frequency components.

Table 5-1 lists those average PSNRs for each σ. According to the list, the best σ for the Gaussian-filtering method is 5, for the Direct method is 10, and for the Smooth method is 8. So, we fixed the σ for each method onwards.

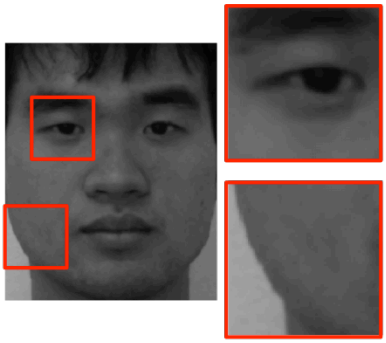
We then calculated the average α of a set of best α values when σ was 10 for the Direct method and 8 for the Smooth method, from 30 output samples to have a fixed α. From the set of the 30 best output images from assessment in Figure 5.8, where α is variable from 0.0 to 1.0, the best α in average was 0.4 for both the Direct and Smooth methods.

According to Equation (4.3), when α was set to 0.4, the learning model is weighted towards λ(.) rather than ϕ(.), i.e. 60% of λ(.) over 40% of ϕ(.), during patch-candidate selection. In other words, the method prefers patches with higher facial parts compatibility value rather than patch similarity. Hence, we can conclude that the proposed λ(.) function in learning model is very significant and effective for a compressed image restoration.

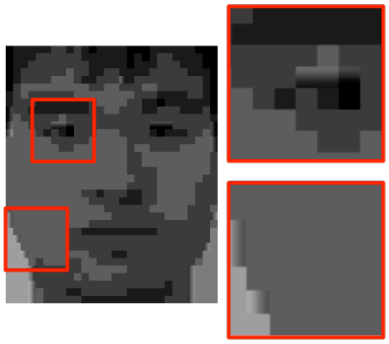
Table 5-1 Average PSNRs of 30 Face Images: Input Image, Gaussian-filtered Image, and Proposed Methods' Output Images

Gaussian parameter, σ	PSNR [dB]			
	Input	Gaussian-filtered	Direct method	Smooth method
4	18.690	21.121	19.646	21.260
5	18.690	**21.193**	20.439	21.503
6	18.690	21.039	20.954	21.656
7	18.690	20.752	20.840	21.698
8	18.690	20.430	21.102	**21.765**
9	18.690	20.071	21.864	21.765
10	18.690	19.719	**21.921**	21.694
11	18.690	19.348	21.779	21.620
12	18.690	18.983	21.642	21.537

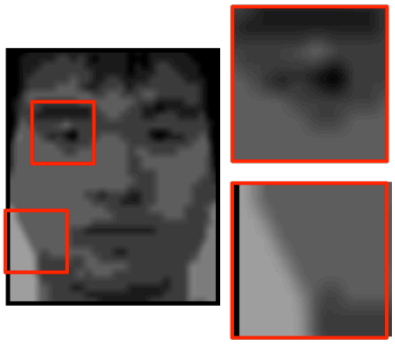
Figure 5.9 presents resulting images for the lowest quality of 1% for Sample 1. Figure 5.9(a) is the original HR image and Figure 5.9(b) is its degraded input image. The results obtained from the Gaussian-filtering, Direct, and Smooth methods are shown in Figure 5.9(c), Figure 5.9(d), and Figure 5.9(e), respectively. The boxed images on the right side of each image are the zoomed versions of the marked regions in the images, where the details of the results can be observed. The Gaussian-filtering method effectively reduced block noise in the image; however, most of the face details are blurred, as shown in Figure 5.9(c), especially along the edges. The restored images with our methods in Figure 5.9(d) and Figure 5.9(e) are significantly sharper than the Gaussian-filtered output image. The proposed methods effectively restored image texture including eye parts.



a) Original image

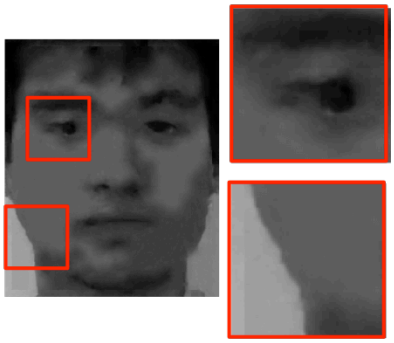


b) Degraded image



c) Gaussian filtered image

d) Direct method's resulting image



e) Smooth method's resulting image

Figure 5.9 Resulting images for Sample 1 with fix σ and α: (a) original face image; (b) degraded input image; (c) Gaussian-filtered image (σ=5); (d) Direct method's output image (σ=10, α=0.4); (e) Smooth method's output image (σ=8, α=0.4).

5.2.2 Comparison between Direct and Smooth Methods

We compared the resulting PSNRs amongst the output images from our Direct and Smooth methods for the 30 samples. Among them, we found that outputs for Sample 2 and 3 had the largest difference in PSNRs when the Direct method

outperformed the Smooth method and vice versa. Table 5-2 shows the different PSNRs for the two samples.

Table 5-2 PSNRs for resulting image of Sample 2 and 3 from Direct and Smooth methods

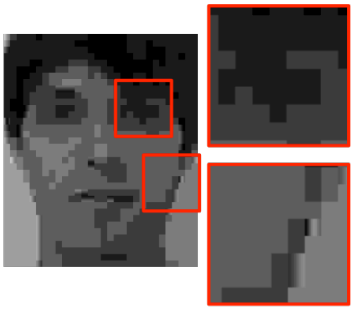
Method	Samples	
	2	3
Direct	22.850	22.561
Smooth	22.153	22.844

For Sample 2, the Direct method outperformed the Smooth method with a difference of 0.697 dB. For Sample 3, the Smooth method outperformed the Direct method with a difference of 0.283 dB. Figure 5.10 and Figure 5.11 show the details of the processed images for Samples 2 and 3, respectively, showing the entire face and parts of the (a) original, (b) degraded input, (c) output images with the Direct method, and (d) output images with the Smooth method.

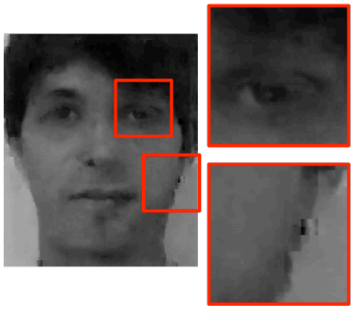
Because the patches and input images were normalized from the positions of the eyes, the algorithms of our methods, which use the positions of facial parts, tend to use patches of eyes for the target eyes. Therefore, the eye parts were accurately restored. Regarding the PSNR evaluation, the Direct method outperformed the Smooth method due to the fact that the Direct method numerically reduced the block noise directly since patches for the Direct method are generated from the image including JPEG noise, while patches for the Smooth method are generated by eliminating the effect of JPEG noise.

However, the images generated with the Smooth method seemed to be more natural, and the quality improvement was stable for different σ. Hence, the Smooth method can be used when natural appearing images are required. Nevertheless, both methods manage to reconstruct the distorted texture especially the eyes part, which was hardly visible before in the degraded image.

Original image



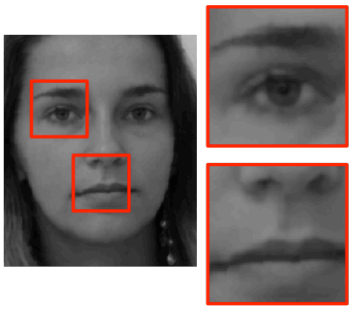
Degraded image



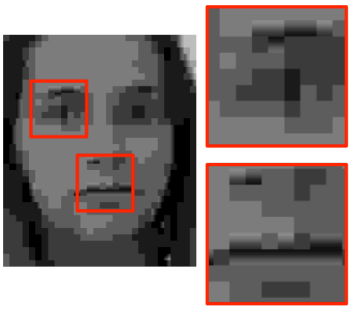
Direct method's resulting image

Smooth method's resulting image

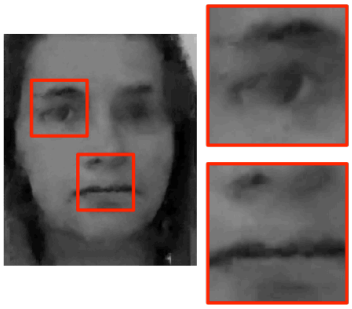
Figure 5.10 Resulting images of Sample 2.



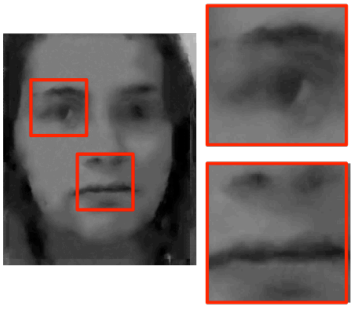
Original image



Degraded image



Direct method's resulting image



Smooth method's resulting image

Figure 5.11 Resulting images of Sample 3.

5.2.3 Evaluation for Different Quality-rate Samples

It is necessary to evaluate our method's performance on different quality of images, too. In previous experiment, we have used the lowest quality rated face images as test images, i.e. 1%, which compression ratio is approximately 134:1, and observed their effectiveness. In this subsection, we want to test our methods on different case of image quality for comparison. A set of 30 face samples were compressed by different quality rate, i.e. 1, 3, 5, 8, 10, and 13%, generating six sets of degraded face samples. Figure 5.12 shows three samples under different quality rate. As quality rate is higher, compression ratio decreases and block noise in images also decreases. Texture in lowest quality rated image is completely distorted especially eye and nose parts, compared to the highest quality ones.

We demonstrated Gaussian-filtering and our two proposed methods on six sets of 30 samples. For the Gaussian-filtering method, we set the σ value from 1 to 6 orderly to find the highest average value of output PSNR for each set. Table 5-3 shows average PSNRs of Gaussian-filtering method on different σ for the six sets of 30 samples. The bolded values are the best resulting PSNRs obtained after block noise was filter out.

We fixed σ at 10 and 8 for the Direct and Smooth methods, respectively, regardless of the sample quality rate to standardize the results for the lowest-quality-rate samples. Table 5-4 shows the average PSNRs for input images, best Gaussian-filtering output images (from Table 5-3), and our methods' output images for different quality rates. Table 5-5 shows the improvement of PSNR, while Figure 5.13 illustrates their graph.

Quality rate [%]	Sample 1	Sample 2	Sample 3
1			
3			
5			
8			
10			
13			

Figure 5.12 Degraded images of JPEG under different quality rate.

Table 5-3 Average PSNRs [dB] of 30 samples for Gaussian-filtered output images under different σ

Quality rate for set of 30 samples [%]	Gaussian filtering, σ					
	1	2	3	4	5	6
1	19.64	20.38	20.87	21.12	**21.19**	21.04
3	19.89	20.62	21.08	21.32	**21.38**	21.20
5	23.23	24.03	**24.31**	24.14	23.72	23.12
8	26.36	**26.95**	26.34	26.03	25.08	24.12
10	27.74	**28.15**	27.60	26.52	25.38	24.29
13	29.27	**29.39**	28.35	26.99	25.65	24.46

Table 5-4 Average PSNRs [dB] of 30 Samples for Input, Best Gaussian-filtered, and Proposed Methods' Output Images for Different Quality Rates

Sample	JPEG quality rate [%]					
	1	3	5	8	10	13
Input	18.690	18.961	22.157	25.130	26.516	28.044
Gaussian-filtering method	21.193	21.376	24.308	26.948	28.151	29.392
Direct method	21.921	22.255	24.618	26.715	27.607	28.676
Smooth method	21.765	22.059	24.930	27.098	27.987	28.985

Table 5-5 Average PSNRs' Improvement from Gaussian-filter and Proposed Methods for Different Quality of Samples

PSNR improvement [dB]	JPEG quality rate [%]					
	1	3	5	8	10	13
Gaussian-filtering method	2.503	2.415	2.151	1.818	1.635	1.347
Direct method	3.231	3.294	2.461	1.585	1.091	0.632
Smooth method	3.075	3.098	2.773	1.968	1.471	0.941

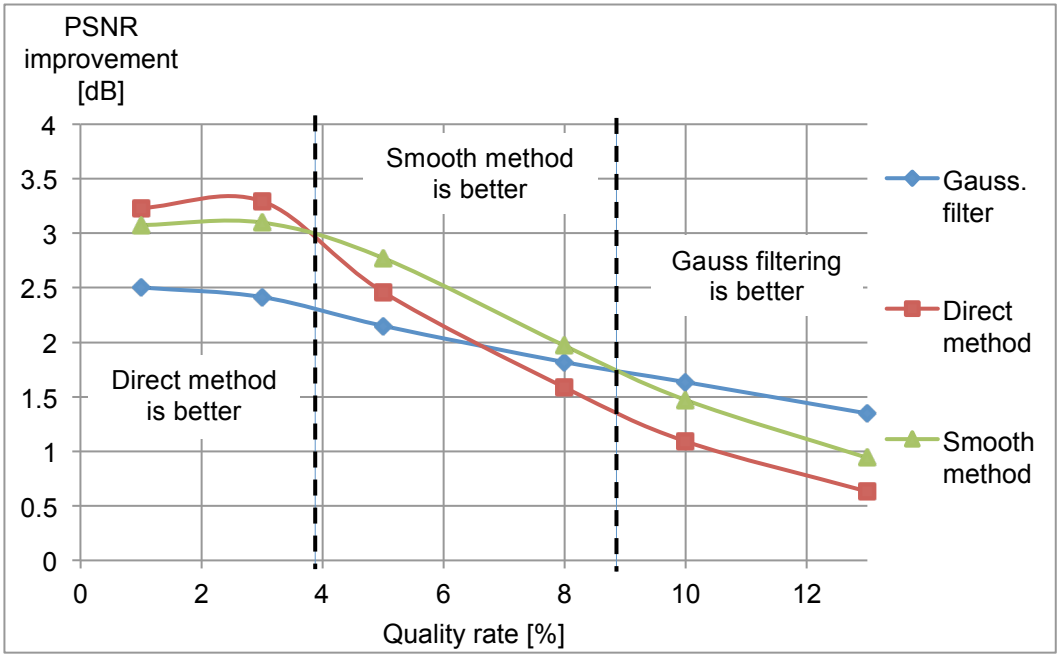


Figure 5.13 Average PSNR improvement of Gaussian-filtered and proposed methods' output images against quality rate of input images

Based on results in Figure 5.13, our methods provided better improvement for images of lower quality rates. Direct method outperformed Smooth method and Gaussian-filtering method at a quality rate of 3% and below, while Smooth method gave best improvement at a quality rate between 5 to 8%. However, as the quality rate increased, i.e., higher than 9%, the conventional Gaussian-filtering method performed better for noise reduction. In other words, our proposed methods perform well for low-quality images.

5.3 Chapter Summary

In present chapter, we have proposed an image-restoration process that involves the example-based method by constructing a patch database using a set of image pairs between original and degraded images. We used the same learning model proposed in Chapter 4, which includes the facial-parts compatibility function. Our proposed restoration methods involve the Direct and Smooth methods that are appropriate to restore a block-noised image. The differences between both methods are, 1) way of texture extraction for HR image in database construction, and 2) final restored image reconstruction process. The experimental results obtained from face images demonstrated excellent performance of our methods in terms of PSNR in comparison with the conventional Gaussian-filtering method. Both proposed methods managed to reconstruct crucial facial parts that were hardly visible or completely distorted before. However, when quality rate is over 9%, which means lesser noise, Gaussian filter alone is sufficient.

CHAPTER 6

IMPROVED PATCH SEARCHING ALGORITHM

In Chapter 4, we have presented the framework of our proposed method for face images super-resolution purpose that takes the correspondence of facial parts into consideration. The method utilizes a set of normalized human face images as training database, where facial parts can be estimated according to patch's original position in image. The nearer the position of training-patch to a currently observed patch's position, the higher the facial-parts compatibility. We have proposed a learning model that select patch candidates not only based on texture similarity, but also based on facial parts' compatibility, i.e. using eye patches for eyes part and nose patches for nose part. The proposed method gives better resulting high-resolution image in terms of peak signal-to-noise ratio (PSNR) compare to the Freeman et al.'s and Lanczos interpolation methods, indirectly indicates that super-resolution process using facial parts compatibility is a significant approach.

However, since the method depends on patches' position or coordinate, patches from each image in database must also be divided in a manner of a grid map, i.e. if patch size is 5x5, patches should be cut with interval of five pixels vertically and horizontally. This way of patch-division consequently reduces diversity of patches' texture patterns from an image since some texture patterns may had been overlooked compare to pixel-by-pixel cutting style. Figure 6.1 shows an example of comparison between both types of patch-division method, where colored box is an example of patch within a face image. Figure 6.1(a) presents the patch-by-patch dividing method with patch's coordinate (a,b) with one pixel overlap border, while Figure 6.1(b) presents a fine way of patch division (pixel-by-pixel). We can observe that one-pixel-interval cutting method gives more fine patches compare to patch-by-patch cutting. The number of patches from (b) is about 15 times more than the number of patches extracted from (a).

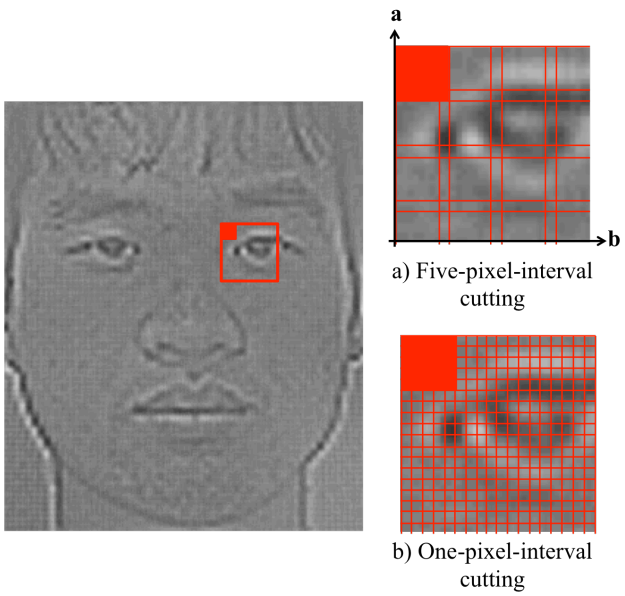


Figure 6.1 Patches generation method.

In this chapter, we proposed a method that utilizes an improved patch-searching algorithm, which not adopt patch-position information but compatible with the one-pixel-interval patch cutting method. The proposed method restricts a specific patch-searching area around the targeted facial part to find patch candidates, in order to ensure that patches were selected by taking facial parts into consideration and cope with the massively increased size of the patch database. We employ a currently observed patch's position information as pinpointed position and specified a certain range of searching area nearby. In other words, we have liberalized the way of patch-division to get much better choice of patch's position or texture, but on the other hand restrict patch-searching area to ensure that patch candidates were selected by taking facial parts into consideration. Our aim is that the proposed method obtains better resulting image since more detailed patches were observed.

6.1 Face Image Super-resolution with Restricted Patch-searching Area

The previous face image SR method relies on patch-position's coordinate during patch candidates selection process, so patches from face images in database must be divided in a manner of a two-dimensional grid map, where if patch size was 5x5, patches were divided by interval of five pixels horizontally and vertically. This cutting method caused some fine texture patterns unavoidably being neglected, compare to typical one-pixel-interval cutting. Therefore, we proposed a face image SR method

that also considers facial parts during patch estimation, and on the other hand compatible with one-pixel-interval cutting approach.

Even a set of normalized face images was used, yet each face has slightly different size or shape of face feature, i.e. small or big eye, full or thin lip, pointed or rounded nose tip and so on. By using one-pixel-interval fine cutting method, we can increase accuracy of texture similarity and choice since more detailed patches were observed.

We proposed new patch-searching algorithms that independent from relying on patch-position information to estimate their facial parts. The algorithms use only information of test image's patch position to indicate the targeted position, and then restricted a certain range of searching area nearby the currently targeted position or facial part.

Figure 6.2 illustrates the searching method to select patch candidates for each node. Let (a,b) is test image's current patch position, we pinpoint the exact same position onto training images in database. Then, we set a certain range of searching area, $(2s+1)^2$ around the pinpointed position. We raster-scanning patches available within the specified search area in each training images through the pixels from left to right, top to bottom (scan line by scan line) to select a set of 100 most similar patches as patch candidates for each nodes. If *s* is 10, searching area will be within 21x21 pixels of perimeter around the pinpointed position of node, giving 441 patches to be considered from each training face image.

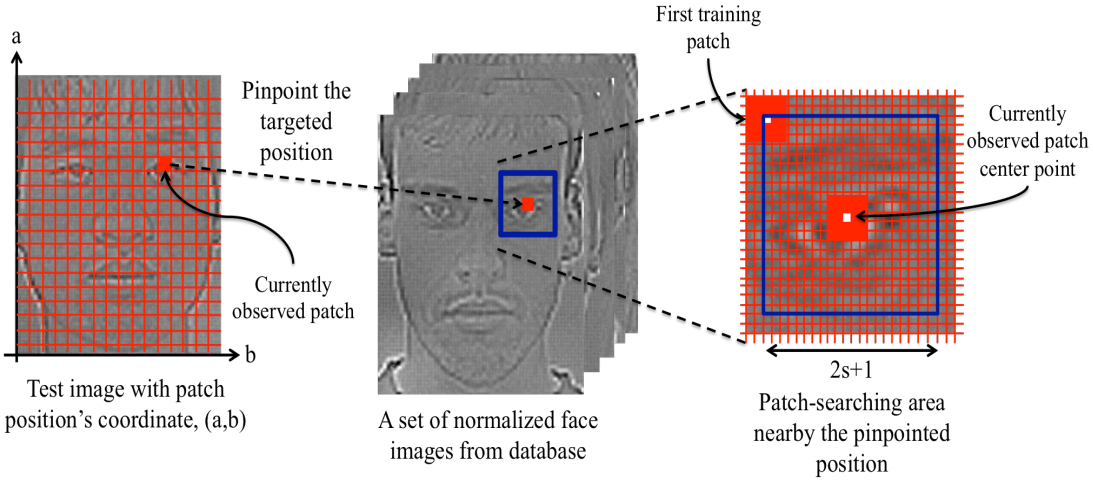


Figure 6.2 Proposed patch-searching algorithms.

Since facial parts compatibility function, λ(.) is not relevant for this method, we only observed patch similarity, ϕ(.) as in Freeman et al.'s method (see Equation (3.1)) to find patch candidates. Then, we iteratively replace patch for each nodes amongst selected patch candidates to find the best possible set of neighboring patches according to border compatibility, ψ(.).

6.2 Results and Discussions

We demonstrated the proposed improved searching algorithm in SR method to enlarge a set of 30 LR face images. Size of searching area, s was set differently from 3, 5, 10, 15, 20, 30, and 40 to observe their performance. The size of patch was 5x5. We demonstrated our proposed method on a set of 30 test LR face images using different size of search area, $(2s+1)^2$, where *s* is set as 3, 5, 10, 15, 20, 30 and 40. The number of face images available in database (excluded test images) is 100.

6.2.1 PSNR Assessment for Different Range of Search Area

We observed average PSNRs for 30 resulting HR face images as shown in Table 6-1, while Figure 6.3 presents the graph.

Table 6-1 Average PSNR for 30 resulting HR images with different size of search area

Search area radius, *s*	Average PSNR [dB]
3	28.635
5	28.922
10	29.080
15	29.124
20	29.140
30	29.016
40	28.950

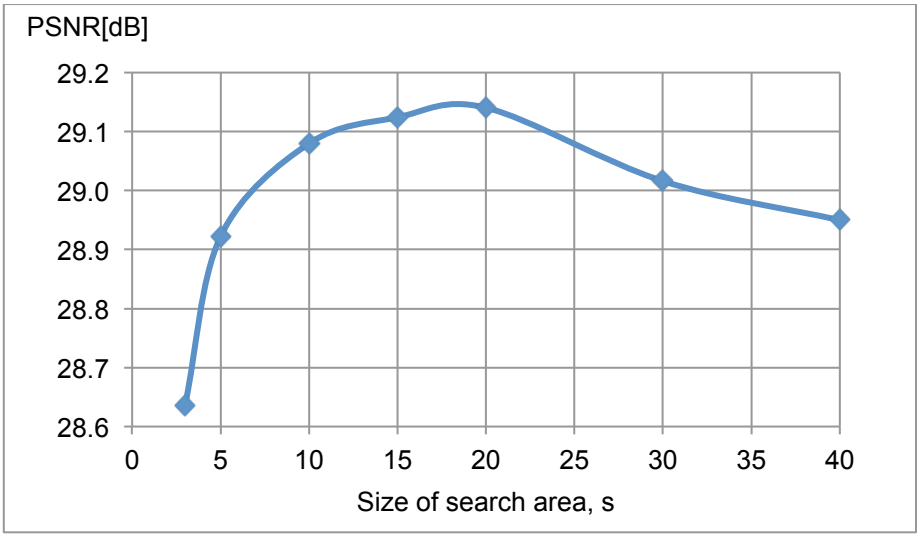


Figure 6.3 Average PSNR for 30 resulting HR images with different size of search area.

Hypothetically, the wider the search area the better the results, since more number of training patches were observed for selecting patch candidates. However, Figure 6.3 indicates that the PSNR value decreased as *s* was larger than 20. We found that the results were better when a certain range of search area was set, i.e. when *s* is between 10 and 20. We conclude that even if we may found much higher similar LR texture (middle-frequency component) from patches available outside the perimeter or far from the targeted position, they do not necessarily give suitable corresponding HR texture (high-frequency component) since they came from different facial parts. Therefore, by restricting search area nearby the targeted facial part, the proposed method manages to restrict then find similar LR patches from database that give plausible corresponding HR patches to reconstruct the HR image.

6.2.2 Computational Cost

Since our method restricted patch-searching area, the number of searched patches during patch-candidates selection was greatly reduced and considerably fewer than the number of searched patches in the previous method, as shown in Table 6-2. From a set of 100 training face images available in database, the number of searched patches for each node for the improved method is about one-third (when *s* is 20) of the number of searched patches in previous method. In addition, the improved method achieves better resulting PSNR than the previous ones thanks to fine cutting of patch (pixel by pixel).

Table 6-2 Number of searched patches for previous and improved method

	Previous method (using facial parts compatibility, λ(.))	Improved method (using restricted search area, s = 20)
Number of searched patches for each node (from 100 training images)	432,000	168,100
Average PSNR [dB]	28.91	29.14

The smaller the size of search area, the smaller the number of searched patches, the faster the processing time. From Figure 6.3, according to the resulting average PSNRs, we can observe that we only need to search within 12,100 patches (s=5) in order to get a resulting HR image that has better or similar quality as the resulting HR image from previous method, i.e. 28.91[dB]. That was approximately 1/35 times the number of searched patches in previous method. In other words, the proposed method has not only improved searching algorithms and given better results yet on the other hand, greatly reduced computational cost.

We compared the improved method's processing time with the previous one's. Table 6-3 presents experiment environment for the process. We demonstrated the previous method, where α is set as 0.71 as recommended in Section 4.3, to reconstruct an HR image. While for the proposed method, we set s as 20. We calculated the processing time for three stages of operation, which are database construction, patch-candidates selection and iteration process, for the sake of comparison.

Table 6-3 Experiment environment

OS	Windows 7 Home Premium SP1
CPU	Intel Core i5 3.2GHz
Memory	4GB
Language	C++
Software	Visual Studio 2012

Table 6-4 Comparison of computational time between previous and proposed improved method

Process	Computational time [s]	
	Previous method	Improved method
Database construction	71.47	49.91
Patch candidates selection	**607.52**	**52.83**
Iteration	2.44	2.66
Whole process	918.62	141.44

Table 6-4 shows the record of computational time for each stage for previous and proposed improved method. In comparison, computational time for patch-candidates selection process shows huge different between both methods. We successfully reduced the computational time by approximately 90% by using the proposed patch-searching algorithms for patch-candidates selection. The computational times for other stages show insignificant different between both methods since those stages are unassociated with the proposed patch-searching algorithms.

6.3 Chapter Summary

This chapter proposed improved patch-searching algorithms for previously proposed face image super-resolution. Prior method in Chapter 4 employed patch's position information to estimate facial parts' compatibility. Since the method relies on patch's coordinate to estimate the facial parts, patches from images in database were divided in a manner of five or more pixels intervals according to patch size. This way of patch-division caused some detailed texture patterns may have been overlooked, compare to pixel-by-pixel patch-division. Hence, we have improved patch-searching method to be compatible with one-pixel-interval patch division so that more detailed texture or position can be observed. We restricted patch-searching area so that patch-candidates are selected from amongst patches around targeted facial parts, and at the same time to cope with the massively increased size of the patch database due to fine division of patches. In other words, we have simplify the way of patch-division to obtain much better details of texture, but on the other hand restricted patch-searching area to take facial parts into consideration, in replace of facial parts compatibility before.

Furthermore, typically the wider the size of searching area the better the results, since more number of patches were observed, but our experiment results show differently. The resulting PSNRs show that when searching area, s was larger than 20, the PSNRs value decreased. We found that even if a selected patch has higher similarity of LR texture (middle-frequency component), it does not necessarily give plausible HR texture (high-frequency component) for an observed LR patch. Therefore, by restricting patch-searching area to ensure that patches were selected from a targeted facial part, the method increases the possibility of selecting plausible correspondence HR patch. Although less number of training patches was observed due to limited search area, the improved method achieves better resulting PSNRs compare to the previous method, thanks to the fine way of patch-division. On top of that, since the number of searched patches was reduced, execution time has greatly speed up by 90%. In summary, the proposed patch-searching algorithms achieve better resulting PSNR of HR image than the previous method's with faster processing times.

CHAPTER 7

CONCLUSIONS

This final chapter recalls the objectives for the research carried out in this thesis. A summary of the methodological contributions is also given, followed by a critical assessment of these contributions. This thesis concludes by presenting an agenda for future research to improve the performance of face image enhancement methodology.

In the beginning of the thesis, the demands of face image enhancement in surveillance fields were clarified, and some issues with conventional image enlargement and noise reduction methods in particular were raised. Recent development of super-resolution technique was briefly discussed, and approach which is related to our research, i.e. example-based methods were explained in details in terms of theory and algorithms.

A framework for face image enhancement by taking facial parts into consideration was proposed in the middle of the thesis. The present research methodology can be simplified into three parts:

1. Chapter 4 : Method of face image super-resolution by taking facial parts into account
2. Chapter 5 : Method of face image restoration for block noise reduction, which includes two proposed approaches: Direct and Smooth Method
3. Chapter 6 : Improved patch-searching algorithms for fast and effective execution of the proposed method

7.1 Research Goals Revisited

Our main goal is to enhance a single degraded face image by taking facial parts into account. To achieve that, a facial parts compatibility function was adopted in the learning-model. We had examined performance of the proposed methods based on PSNR assessment. According to experimental results, the proposed methods provide excellent performance and manage to restore the missing texture of a face

image, which clarified that the idea of considering facial parts in machine learning technique provides an effective measure to enhance a face image.

Research Objective 1

To enhance apparent of sharpness of texture in a single enlarged face-image by restoring the missing texture, which cannot be done by conventional methods alone.

Presented experimental results show that our proposed super-resolution method has effectively restored the missing texture of facial parts, and yields sharper and clearer image of face feature than those from conventional interpolation method of Lanczos. According to PSNRs assessment given in Table 4-4 and Table 4-5, proposed method achieves improvement of PSNR by approximately 3 [dB] and 0.6 [dB], in average, compared to the Lanczos resampling method and Freeman et al. method, respectively.

Since the proposed learning-model utilized patch position to determine facial parts, we have to use standardized face images where face feature points are at nearly same position on each image, as dictionary. It is recommended to examine performance of the proposed learning-model on non-standard face sample, for instance, different angle and size of face. From experiments in Section 4.3, resulting PSNRs indicate that our proposed method still achieves better performance than the other two methods. This is because the method also takes patch pattern similarity into account with adjustment of the α.

Till now, amongst learning-based methods that have been proposed, size of patch that have been used is typically 5x5 or 7x7 in pixels, for a typical four times magnification. In Section 4.4, we have further conducted an in-depth analysis of parameter setting to increase our method's performance. Surprisingly, PSNR assessments have revealed that the best setting of patch size is adaptable to the magnification times (or scaling factor). Best patch size in the algorithms can be determined by a formula: $(2 \times (\text{scaling factor}) + 1)^2$.

In conclusion, our research has fulfilled the above-mentioned objective. On top of that, we have also improved the algorithms involved in the proposed method in terms of parameter setting.

Research Objective 2

To reduce block noise in a compressed degraded face-image by estimating new texture details and restore distorted texture especially around face features.

Chapter 5 has introduced two variations of compressed face image restoration methods: Direct Method and Smooth Method. A set of highly compressed (lowest quality rates) face samples was used in experiments. Both proposed methods are executed with different σ (for Gaussian filtering) to get the best resulting images. Based on experimental results, proposed method achieves better PSNR than Gaussian filtered image (conventionally de-noised image), proving that the proposed learning model can also works for degraded compressed image restoration.

As we can see on the given highly degraded face samples, texture of facial features is completely distorted. Facial features like eyes do not seem like eyes anymore. Unlike the prior experiment (in Objective 1), this time the challenge is to predict new information for the ruined part of face. In response to that, that is why the best α, in average, for compressed image restoration is lower, i.e. α =0.4 (the assessment is given in Subsection 5.2.1), than the best α for enlargement process, i.e. α = 0.71, before. Lower α means patches with higher facial parts compatibility are highly preferred or prioritized. In other words, our proposed facial parts function has played a significant role to achieve this second objective.

We have also extended our experiments using samples of different compression rate. From the assessment, we have concluded that, Direct Method is best for a highly compressed image, and Smooth Method is best for medium level of compressed image. For an image of which noise are barely visible, a Gaussian filtering is sufficient.

7.2 Contributions

Methodological Contributions

Numerous approaches in learning-based super-resolution field that have been proposed do not involve the strategy of taking facial parts into account for face image enhancement, yet. The proposed modified learning model that incorporate facial-parts function may open a new topic of interest in related fields.

Moreover, the special characteristic of learning-based approach is its dictionary where a collection of same kind of images is exploited. In our works, we have not only utilized images from the same kind, i.e. training face images for face image enhancement, but further classifying them by facial parts, i.e. eye patches for eye parts enhancement. This would greatly increase the likelihood of a chosen patch to be the best-fit patch in the algorithms.

In terms of image quality improvement, the proposed method for enlarged face image enhancement achieved 3[dB] and 0.6[dB] of PSNR improvement, in average, compared to the Lanczos resampling and Freeman et al.'s method, respectively. For face identification in crime investigation, even a 1[dB] of improvement is significant. We have also upgraded the proposed method by modifying the patch-searching algorithm, which contributes to speed up the processing time by ten times for better application.

Research publications and proceedings

[1] S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : "Example-based Face Image Super-resolution Taking into Consideration Correspondence of Facial Parts", IEEJ Transactions on Electronics, Information and Systems, vol. 12, no. 6, pp. 917-924 (July 2017).

[2] S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : "Face Image Super-Resolution with Adaptive Patch Size to Scaling Factor", Journal of Image and Graphics, vol. 6, No. 2, pp. 167-173 (December 2018). A paper presented at the 10th International Conference on Digital Image Processing (ICDIP 2018), Best Paper Award, Shanghai, China.

[3] S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : "Improved face image super-resolution with restricted patch-searching area", in Proceedings of the 3rd International Conference on Cryptography ICCSP '19, Security and Privacy, pp. 184-190 (January 2019).

[4] S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : "Example-based Face-image Restoration for Block-noise Reduction", to appear in Journal of Image and Graphics. A paper presented at the 3rd International Conference on Frontiers of Image Processing (ICFIP 2019), Best Presentation Award, Florence, Italy.

Research presentations in Japanese

[1] スハイル・ハムダン, 福水洋平, 泉知論, 山内寛紀 : "顔部品の位置を考慮した学習型顔画像超解像", 電気学会研究会資料, PI-17-080/IIS-17-085, pp.21-26, 2017 年 9 月

* S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : “Example-based Face Image Super-resolution by Taking Facial Parts Position into Account”, Seminar Notes in Electrical Institute of Electrical Engineers of Japan (IEEJ) Conference, September 2017

[2] スハイル・ハムダン, 福水洋平, 泉知論, 山内寛紀：“ブロックノイズにより劣化した顔画像の学習型復元手法”, 電気学会研究会資料, ST-17-077, pp.17-22, 2017 年 12 月

* S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : “Example-based Restoration Method for Degraded Block Noised Face Image”, Seminar Notes in Electrical Institute of Electrical Engineers of Japan (IEEJ) Conference, December 2017

7.3 Limitations

In present framework, since the proposed learning-model estimates facial parts according to patch position in image, we have to use a set of standardized face image which facial feature points are at nearly same position. This specific condition limits the usage of the proposed method since in reality, face images taken from surveillance systems in some cases are random, for instance, has different angle or rotation. We must preprocess the targeted face image to somehow fit with face images used in database. However, set of a standard face images for database can easily be acquired, where we can use face photos of identity cards.

In addition, images database must be preprocessed according to scaling factor, or compression rate of a targeted sample. How do we enhance a degraded face sample which degradation level is unknown? Using training images that have different compression rate with the targeted image as database, is an interesting experiment to be conducted in future tasks.

7.4 Directions for Future Research

Further experiments can be carried out following the task-oriented approach proposed in this section.

7.4.1 Utilization of feature maps

Various related studies in example-based super-resolution fields basically proposed different combination of types of feature maps and learning-model (see Table 3-1). In present research, we use a Gaussian filter to extract texture (middle-and-high frequency component) from images and employed Markov Random Field as learning model. We have proposed an improvised learning model involving facial parts compatibility function. For future experiments, we may apply several different

methods of texture extraction in the proposed method to examine their performance, as shown in Figure 7.1. By this assessment, we may find which feature maps would provides best performance for the proposed method.

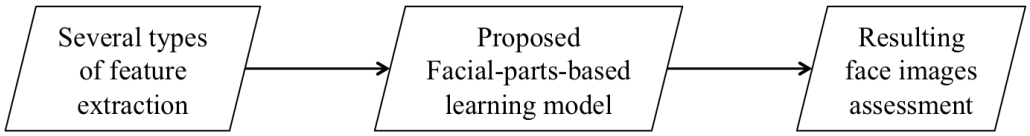


Figure 7.1 Further experiments using different feature maps

7.4.2 Restoration for different types of noise

Other than block noise of JPEG compression, there are several common types of noise in digital imaging that has yet to be examined in the present work, comprising:

1. Gaussian noise
2. Salt-and-pepper noise (or also well known as impulse noise)
3. Quantization noise

Experiments on the performance of proposed method on degraded image from different type of noise can be conducted. Assessments of which types of noise has effectively reduced by the proposed method would be useful for application.

BIBLIOGRAPHY

1) M. P. J. Ashby : “The Value of CCTV Surveillance Cameras as an Investigative Tool: An Empirical Analysis”, European Journal on Criminal Policy and Research, vol. 23, no. 3, pp. 441-459 (2017)

2) G. Gerrard, G. Parkins, I. Cunningham, W. Jones, S. Hill and S. Douglas : “National CCTV strategy”, Home Office and ACPO publication (2007)

3) H. Greenspan : “Super-Resolution in Medical Imaging”, The Computer Journal, vol. 52, no. 1, pp. 43–63 (2009)

4) B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes and R. M. Mersereau : "Eigenface-domain super-resolution for face recognition", IEEE Transactions on Image Processing, vol. 12, no. 5, pp. 597-606 (2003)

5) H. Shen, M. K. Ng, P. Li, L. Zhang : “Super-Resolution Reconstruction Algorithm To MODIS Remote Sensing Images”, The Computer Journal, vol. 52, no. 1, pp. 90–100 (2009)

6) H. Seibel, S. Goldenstein, A. Rocha : “Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos”, IEEE Access, vol. 5, pp. 20020-20035 (2017)

7) T. Quan, P. Li, F. Long, S. Zeng, Q. Luo, P. N. Hedde, G. U. Nienhaus, and Z. -Li. Huang : “Ultra-fast, high-precision image analysis for localization-based super resolution microscopy”, Optics Express, vol. 18, no. 11, pp. 11867-11876 (2010)

8) R. Keys : “Cubic Convolution Interpolation for Digital Image Processing”, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 6, pp. 1153-1160 (1981)

9) C. E. Duchon : “Lanczos Filtering in One and Two Dimensions”, Journal of Applied Meteorology, vol. 18, pp. 1016–1022 (1979)

10) C. Singh, S. Singh, R. Saini and A. K. Saini : “A Comparative Analysis of Image Scaling Algorithms”, International Journal of Image, Graphics and Signal Processing, pp. 55-62 (2013)

11) S. D. Ruikar and D. D. Doye : “Image Denoising using Tri Nonlinear and Nearest Neighbour Interpolation with Wavelet Transform”, International Journal of Information Technology and Computer Science, vol. 4, pp. 36-44 (2012)

12) N. Muelle and T. K. Nguyen : “Image interpolation using classification and stitching”, in Proceedings of the 15th IEEE International Conference on Image Processing, pp. 901-904 (2008)

13) V. Patel and K. Mistree : “A Review on Different Image Interpolation Techniques for Image Enhancement”, International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 12, pp. 129-133 (2013)

14) S. C. Park, M. K. P. and M. G. Kang : “Super-Resolution Image Reconstruction: A Technical Overview”, IEEE Signal Processing Magazine, vol.20, pp.21-36 (2003)

15) D. Capel and A. Zisserman : “Computer vision applied to super resolution”, IEEE Signal Processing Magazine, vol.20, pp.75-86 (2003)

16) H. Ozdemir and B. Sankur : “Subjective Evaluation of Single-Frame Super-resolution Algorithms”, in Proceedings of the 17th European Signal Processing Conference, EUSIPCO, pp.1102-1106 (2009)

17) S. Farsiu, M. Robinson, M. Elad, and P. Milanfar : “Fast and robust multiframe super resolution”, IEEE Transactions On Image Processing, vol. 13, no.10, pp. 1327–1344 (2004)

18) J. Hu and Y. Luo : “Single-image superresolution based on local regression and nonlocal self-similarity”, Journal of Electronic Imaging, vol. 23, no.3 (2014)

19) W.-C. Siu and K.-W. Hung : “Review of image interpolation and super-resolution”, in Proceedings of Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, pp. 1–10 (2012)

20) H. He and W.-C. Siu : "Single image super-resolution using Gaussian process regression", in Proceedings of IEEE International Conference Computer Vision and Pattern Recognition (CVPR2011), pp.449–456 (2011)

21) S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos : “SoftCuts: A soft edge smoothness prior for color image super-resolution”, IEEE Transactions on Image Processing, vol.18, no.5, pp.969–981 (2009)

22) J. Sun, J. Sun, Z. Xu, and H. Y. Shum : "Gradient Profile Prior and Its Applications in Image Super-Resolution and Enhancement", IEEE Transactions on Image Processing, vol.20, no.6, pp. 1529–1542 (2011)

23) S. Baker and T. Kanade : “Limits on super-resolution and how to break them”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 9, pp. 1167–1183 (2002)

24) Z. Lin and H.-Y. Shum : “Fundamental Limits of Reconstruction-Based Super-resolution Algorithms under Local Translation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 1 (2004)

25) Z. Lin, J. He, X. Tang and C.-K. Tang : “Limits of Learning-Based Superresolution Algorithms”, in Proceedings of the 11th IEEE International Conference on Computer Vision (2007)

26) W. T. Freeman, T. R. Jones, and E. C. Pasztor : “Example-based super-resolution”, IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56–65 (2002)

27) W.T. Freeman, E.C. Pasztor, and O.T. Carmichael : “Learning Low-Level Vision”, International Journal of Computer Vision, vol. 40, no. 1, pp. 25–47 (2000)

28) W.T. Freeman and E.C. Pasztor : “Markov Networks for Super-resolution”, in Proceedings of the 34th Annual Conference Information Sciences and Systems (CISS 2000), Department of Electrical Engineering, Princeton Univ. (2000)

29) S. Shuji, S. Takashi, and I. Akihiko : “Example-based super-resolution to achieve fine magnification of low-resolution images”, NEC Technical Journal, vol. 7, no. 2, pp. 81–85 (2012)

30) S. F. Lui, J. Y. Wu, and H.-S. Mao : “Learning-based super-resolution system using single facial image and multi-resolution wavelet synthesis”, Asian Conference on Computer Vision (ACCV), pp. 96–105 (2007)

31) J. Yang, J. Wright, T. S. Huang, and Y. Ma. : “Image super-resolution via sparse representation”, IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861–2873 (2010)

32) J. Yang, Z. Wang, Z. Lin, and S. Cohen : “Coupled Dictionary Training for Image Super-Resolution”, IEEE Transactions on Image Processing, vol. 21, no. 8, pp. 3467–3478 (2012)

33) C. V. Jiji and S. Chaudhuri : “Single frame super-resolution using learned wavelet coefficients”, International Journal of Imaging Systems and Technology, vol. 14, no. 3, pp. 105–112 (2004)

34) C. V. Jiji and S. Chaudhuri : “Single-frame images super-resolution through contourlet learning”, EURASIP Journal on Advances in Signal Processing, pp. 1–11 (2006)

35) W. Wu, Z. Liu, W. Gueaieb, and X. He : “Single-image super-resolution based on Markov Random Field and contourlet transform”, SPIE Journal of Electronic Imaging, vol. 20, no. 2, 023005 (2011)

36) X. Lai, Z. Lin, E. S. Ward and R. J. Ober : “Noise suppression of point spread functions and its influence on deconvolution of three-dimensional fluorescence microscopy image sets", Journal of Microscopy, vol. 217, pp. 93-108 (2005)

37) S. Al-ameri, N. Kalyankar, and S. Khamitkar : “Deblured Gaussian blurred images”, Journal of Computing, vol. 2, no. 4, pp. 33–35 (2010)

38) G. M. P. van Kempen and L. J. van Vliet : “Improving the restoration of textured objects with prefiltering”, in Proceedings of the 3rd Annual Conference of the Advanced School for Computing and Imaging (ASCI’97), pp. 174–179 (1997)

39) G. Dougherty and Z. Kawaf : “The point spread function revisited: Image restoration using 2-D deconvolution”, Radiography, vol. 7, no. 4, pp. 255–262 (2001)

40) T. Nir and N. Karpel : "Example based learning of image stitching for an omni-directional camera using a variational optical flow methodology", in Proceedings of SPIE 7000, Optical and Digital Image Processing (2008)

41) J.-Y. Cui et al. : "Example-Based Image Compression", presented at the 17th IEEE International Conference on Image Processing (ICIP) (2010)

42) K. Grauman, G. Shakhnarovich and T. Darrell : "Virtual Visual Hulls: Example-Based 3D Shape Inference from Silhouettes", in Proceedings of the 2nd Workshop on Statistical Methods in Video Processing (SMVP), Prague, Czech Republic (2004)

43) B.-Y. Koo, E.-J. Park, D.-K. Choi et al. : "Example-based statistical framework for parametric modeling of human body shapes", Computers in Industry, vol. 73, pp. 23–38 (2015)

44) Y. Fukumizu, Y. Akamatsu, T. Izumi and H. Yamauchi : “Images up-scaling algorithm based on the total variation method and morphological emphasizing”, The Journal of the Institute of Image Electronics Engineers of Japan, vol. 41, no. 5 (2012)

45) S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi, “Example-based Face Image Super-resolution Taking into Consideration Correspondence of Facial Parts”, IEEJ Transactions on Electronics, Information and Systems, vol. 12, no. 6, pp. 917-924 (July 2017)

46) S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : “Face Image Super-Resolution with Adaptive Patch Size to Scaling Factor”, Journal of Image and Graphics, vol. 6, No. 2, pp. 167-173 (December 2018)

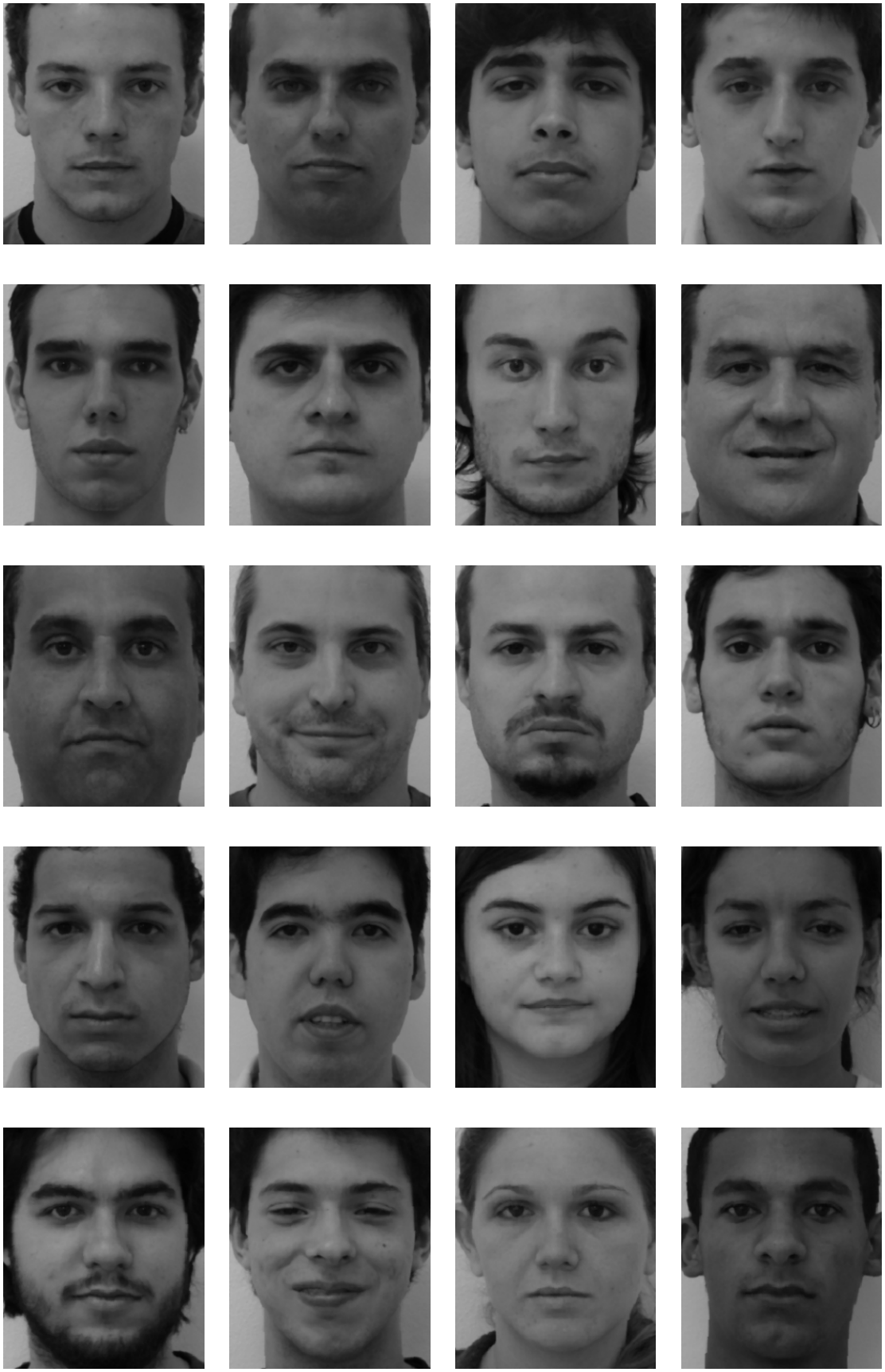
47) S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi, “Improved face image super-resolution with restricted patch-searching area”, in Proceedings of the 3rd International Conference on Cryptography ICCSP '19, Security and Privacy, pp. 184-190 (January 2019)

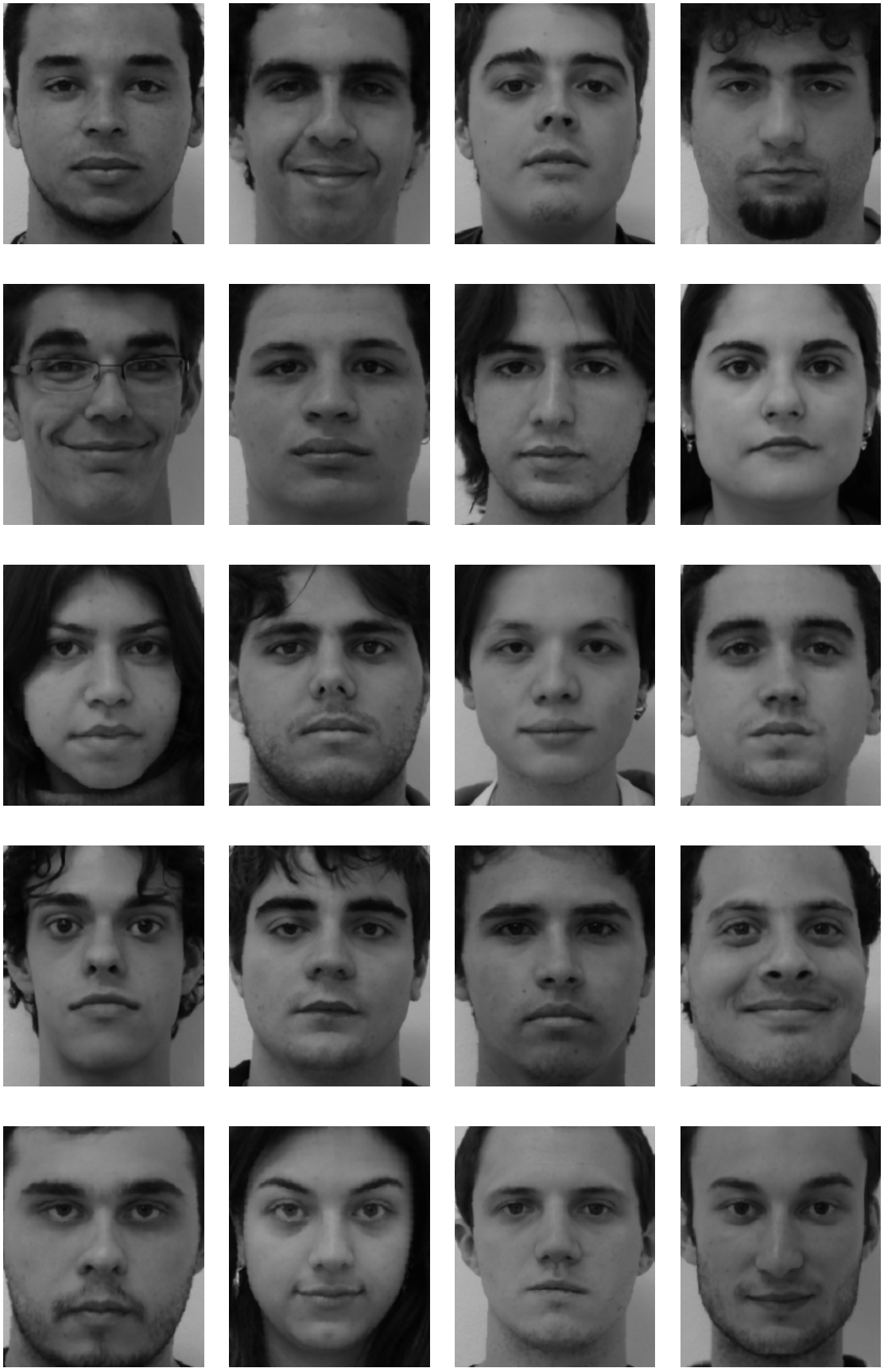
48) S. Hamdan, Y. Fukumizu, T. Izumi and H. Yamauchi : “Example-based Face-image Restoration for Block-noise Reduction”, to appear in Journal of Image and Graphics.

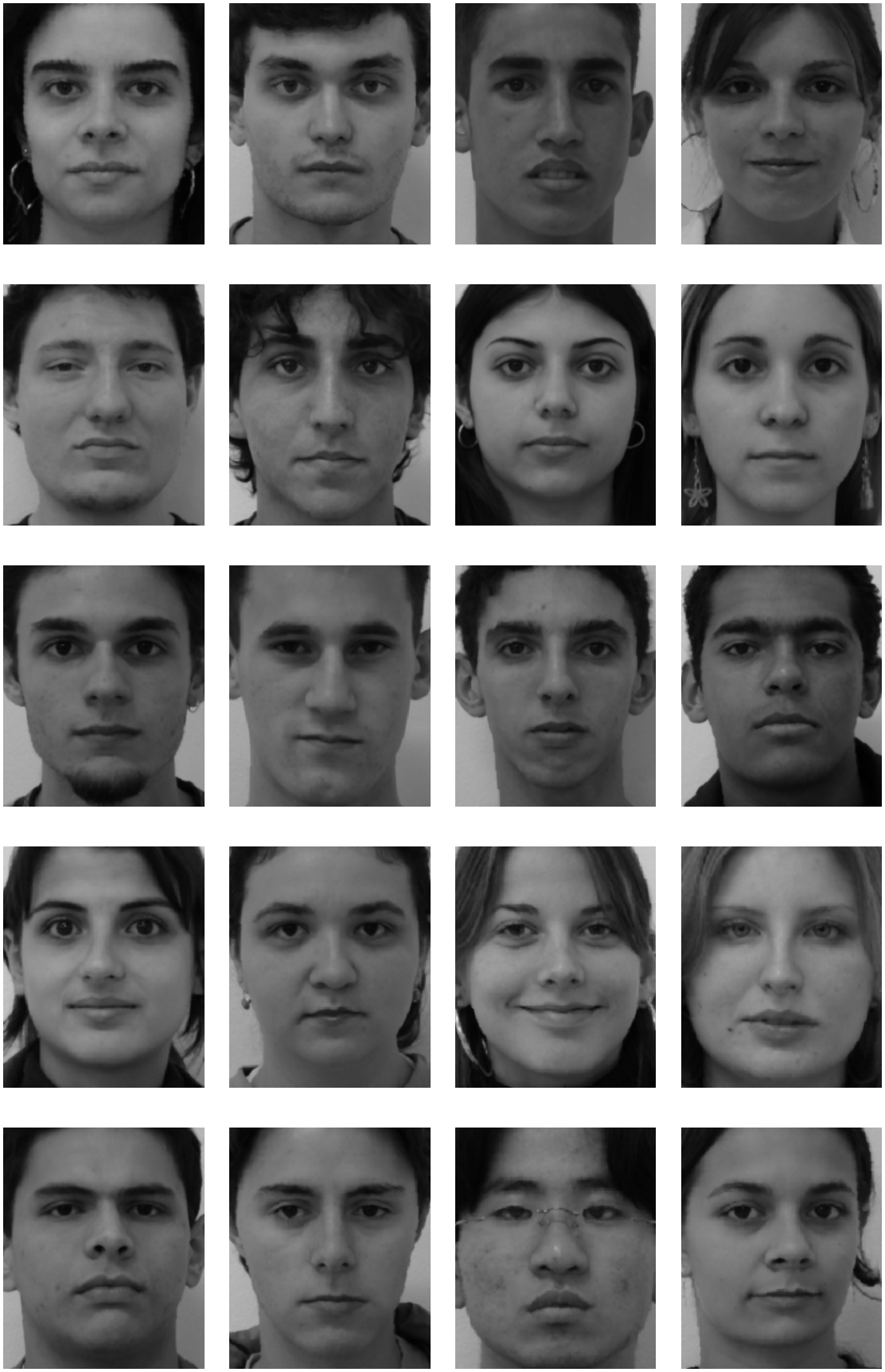
APPENDIXES

Appendix A

Face Images for Training Patches Database [288 x 240 pixels]

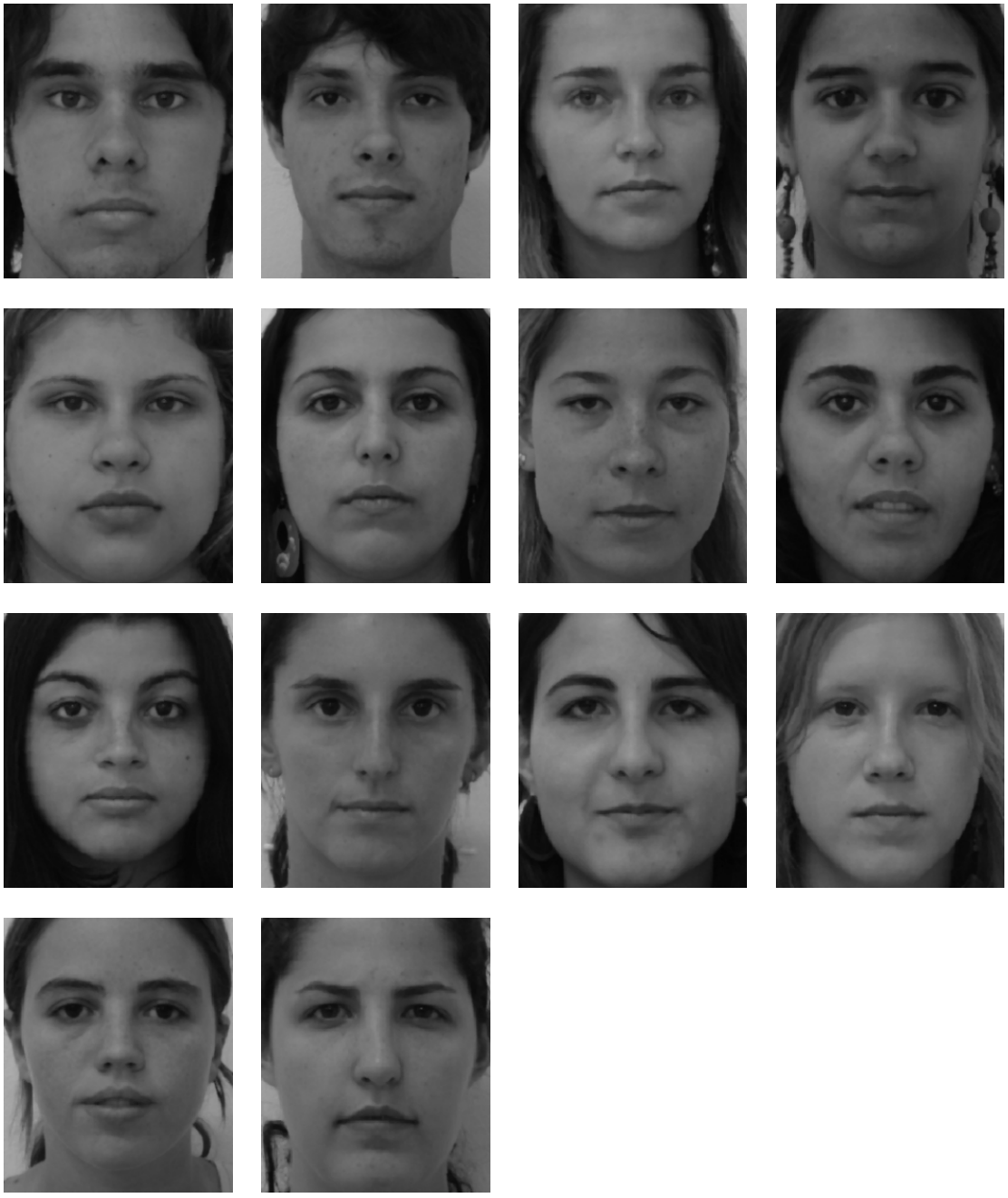






Appendix B

30 Samples of Face Images for Experiment Purpose [288 x 240 pixels]



Appendix C

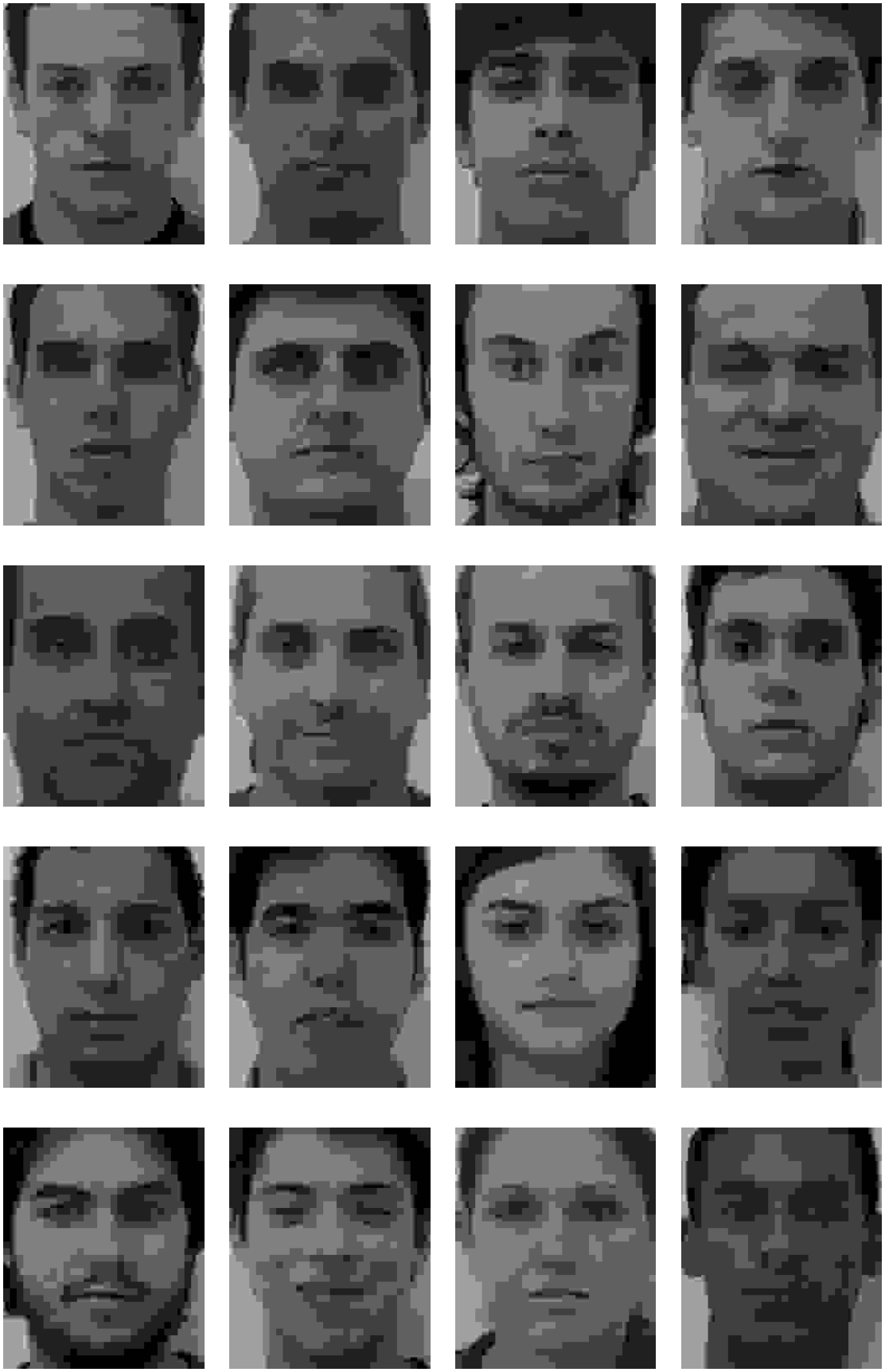
Face Image Super-resolution Method : Compilation of PSNR [in dB] Results for 30 Face Samples with Different α

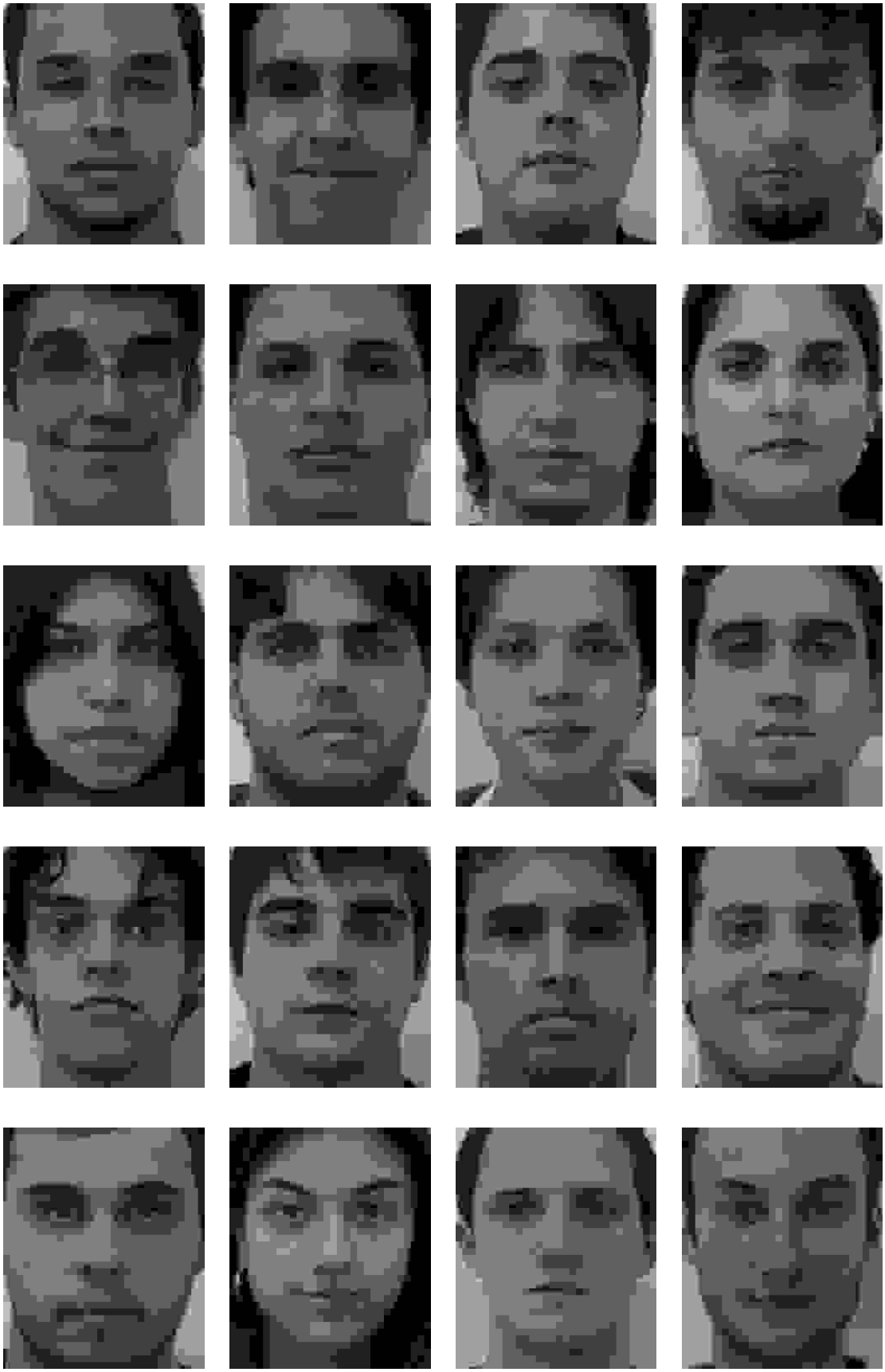
Face sample	Proposed method with different α											Lanczos resampling	Best result	Best α
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1			
Sample 1	28.550	29.345	29.649	29.800	29.984	30.015	30.198	30.338	**30.385**	30.379	29.654	27.025	30.385	0.8
Sample 2	25.571	26.562	26.835	26.923	27.021	27.008	**27.109**	27.066	27.081	26.831	26.557	24.881	27.109	0.6
Sample 3	28.100	29.038	29.249	29.210	29.407	29.510	29.511	29.676	29.613	**29.696**	28.997	26.282	29.696	0.9
Sample 4	27.799	28.737	28.771	28.946	28.904	29.072	29.145	**29.204**	29.038	29.081	28.593	25.399	29.204	0.7
Sample 5	27.330	28.108	28.204	28.316	28.370	28.323	28.388	28.359	28.333	**28.560**	27.895	25.285	28.560	0.9
Sample 6	27.385	27.825	28.067	28.111	28.108	28.133	28.053	28.179	**28.317**	28.275	27.567	25.621	28.317	0.8
Sample 7	24.663	25.387	25.613	25.805	25.821	**25.922**	25.900	25.728	25.834	25.767	25.192	24.205	25.922	0.5
Sample 8	29.401	29.521	29.707	29.827	29.974	30.041	30.152	30.151	30.280	**30.282**	29.615	27.450	30.282	0.9
Sample 9	28.946	29.570	29.819	29.945	30.108	30.135	30.209	30.367	30.394	**30.396**	29.667	27.480	30.396	0.9
Sample 10	28.650	29.657	29.865	30.058	30.225	**30.391**	30.209	30.185	30.375	30.112	29.266	26.811	30.391	0.5
Sample 11	27.458	28.286	28.454	28.437	28.505	28.549	28.633	**28.780**	28.687	28.675	28.023	25.488	28.780	0.7
Sample 12	28.824	29.460	29.792	29.967	29.832	29.874	29.929	**29.999**	29.881	29.905	29.594	27.723	29.999	0.7
Sample 13	29.026	29.435	29.699	29.594	29.991	**30.123**	29.750	29.921	29.877	29.881	28.982	27.077	30.123	0.5
Sample 14	26.830	28.093	28.296	28.524	28.480	**28.753**	28.667	28.680	28.598	28.693	28.432	24.867	28.753	0.5
Sample 15	27.997	28.826	29.085	29.088	29.224	29.329	**29.433**	29.375	29.296	29.336	28.535	26.898	29.433	0.6

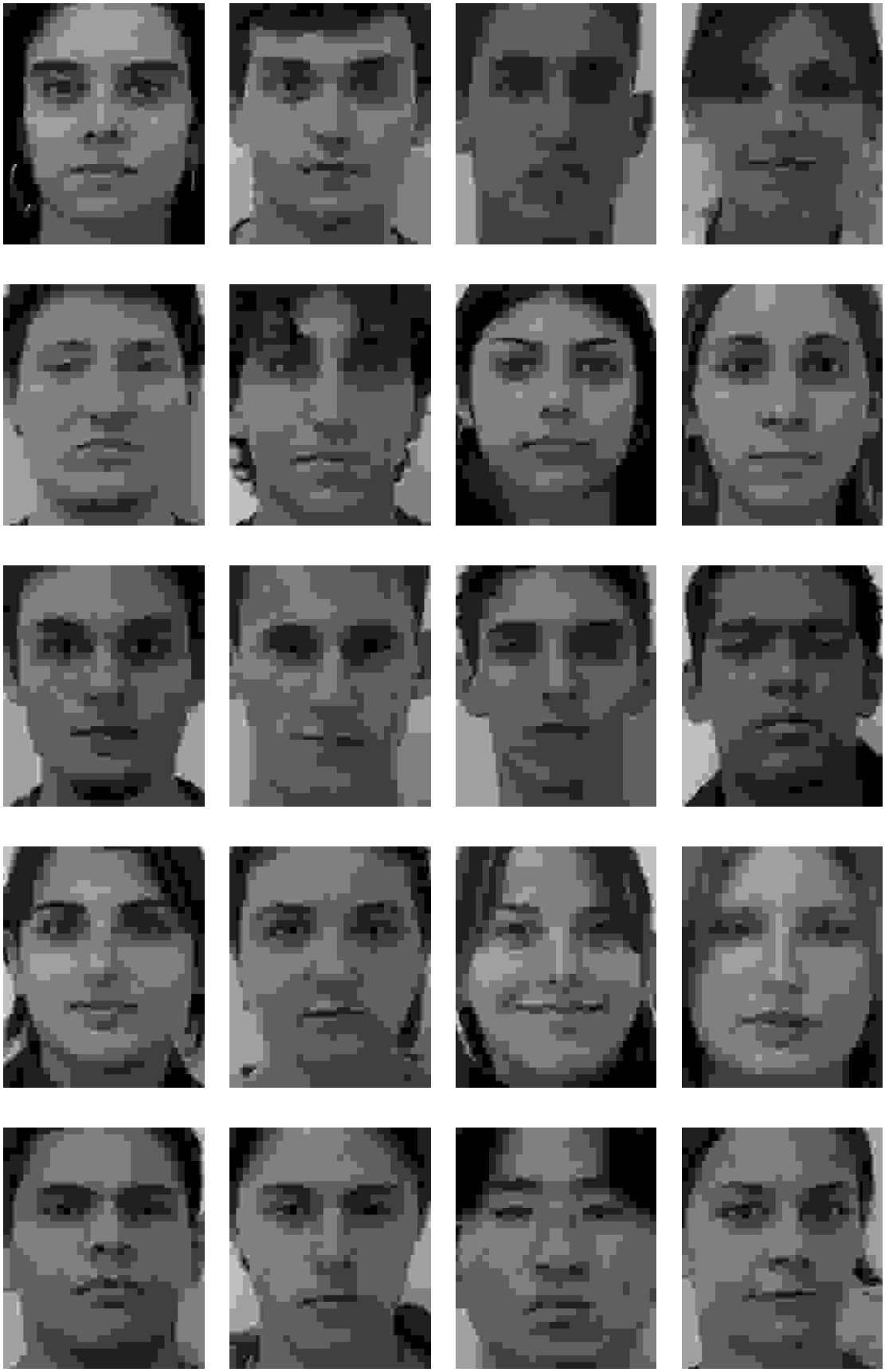
Face sample	Proposed method with different α											Lanczos resampling	Best result	Best α
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1			
Sample 16	27.281	28.176	28.515	28.608	28.585	**28.670**	28.609	28.617	28.550	28.509	28.083	23.544	28.670	0.5
Sample 17	25.620	26.687	26.983	27.183	27.108	27.183	27.519	**27.550**	27.500	27.443	26.916	24.968	27.550	0.7
Sample 18	28.978	29.606	29.983	30.034	30.095	30.068	30.134	**30.204**	30.161	30.186	29.717	27.626	30.204	0.7
Sample 19	28.200	28.608	28.744	29.010	29.038	29.088	29.194	29.187	**29.458**	29.219	28.462	27.238	29.458	0.8
Sample 20	24.778	25.662	25.871	25.961	26.081	26.122	**26.226**	26.189	26.007	25.835	25.251	24.103	26.226	0.6
Sample 21	25.915	26.593	26.835	27.023	27.121	27.190	27.216	27.107	27.172	**27.260**	26.638	25.130	27.260	0.9
Sample 22	29.242	30.050	30.300	30.308	30.090	30.157	30.326	**30.384**	30.383	30.156	29.666	26.321	30.384	0.7
Sample 23	27.185	28.032	28.245	28.145	28.372	28.383	28.387	28.340	28.456	**28.466**	27.698	26.027	28.466	0.9
Sample 24	26.685	27.383	27.595	27.745	27.825	27.999	27.966	27.924	**28.035**	28.021	27.363	25.644	28.035	0.8
Sample 25	29.402	29.803	29.980	30.257	30.346	30.443	30.522	**30.646**	30.530	30.445	29.743	27.635	30.646	0.7
Sample 26	27.985	28.837	29.038	29.269	29.182	29.367	29.230	29.425	**29.562**	29.517	28.983	27.016	29.562	0.8
Sample 27	26.879	27.520	27.626	27.718	28.098	28.142	28.234	**28.288**	28.175	27.890	27.524	25.228	28.288	0.7
Sample 28	29.001	29.642	29.883	29.969	30.097	30.250	30.260	30.384	30.365	**30.388**	29.842	27.925	30.388	0.9
Sample 29	25.704	26.622	26.978	27.058	27.175	**27.277**	27.274	27.227	27.275	26.876	26.269	24.074	27.277	0.5
Sample 30	26.348	27.046	27.186	27.419	27.549	27.463	**27.622**	27.506	27.487	27.565	27.101	24.830	27.622	0.6
Average	27.524	28.271	28.496	28.609	28.690	28.766	28.800	28.833	28.837	28.788	28.194	25.993	28.913	**0.71**

Appendix D

JPEG Compressed Face Images of the Lowest Quality Rate for Training Patches Database [288 x 240 pixels]

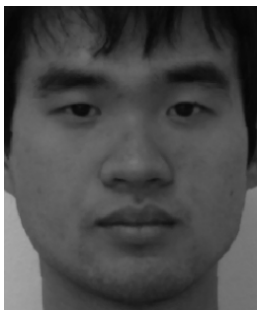
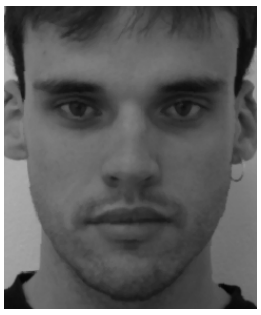
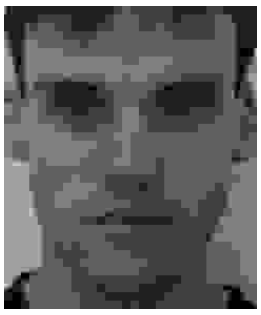
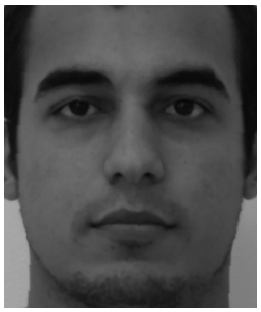
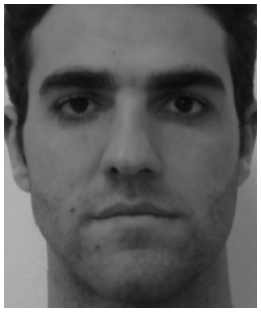
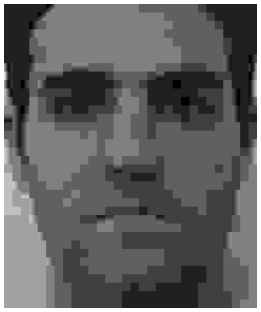




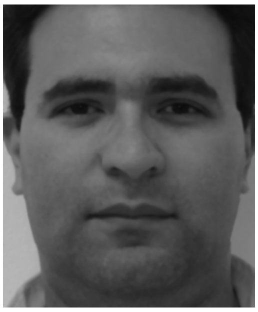
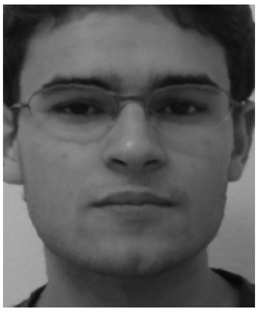
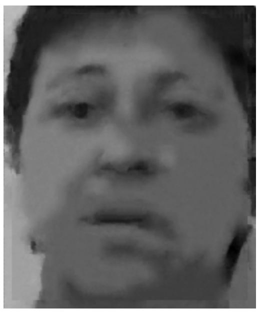


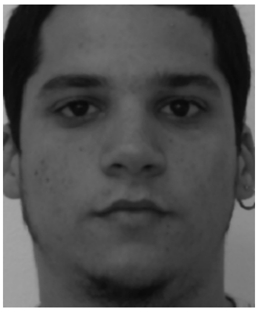
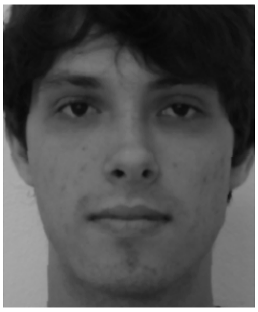
Appendix E

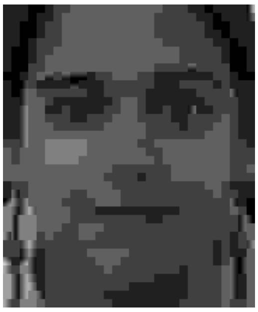
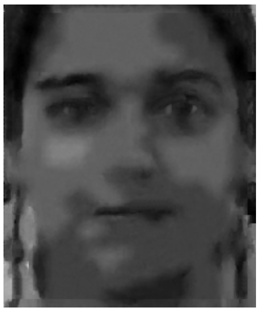
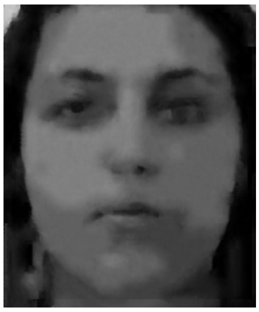
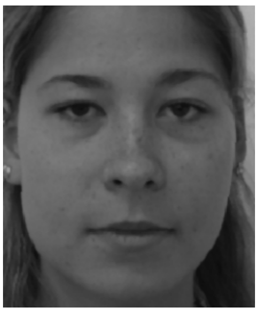
30 Samples of Degraded Face Images for Experiment Purpose and Their Resulting Images from Direct Method and Smooth Method

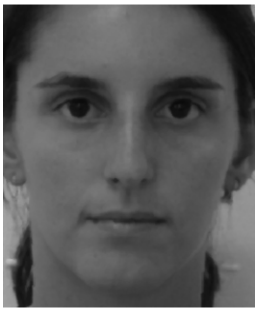
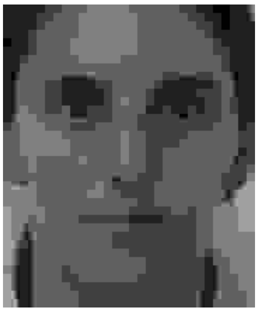
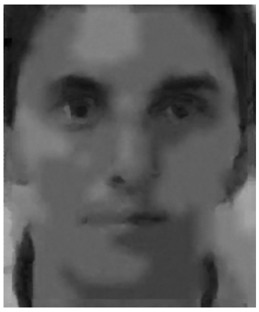
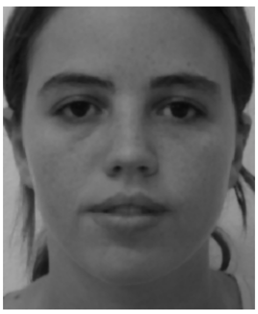
Original image	Degraded image	Direct method	Smooth method
			
			
			
			

Original image	Degraded image	Direct method	Smooth method
			
			
			
			
			

Original image	Degraded image	Direct method	Smooth method
			
			
			
			
			

Original image	Degraded image	Direct method	Smooth method
			
			
			
			
			

Original image	Degraded image	Direct method	Smooth method
			
			
			
			
			

Original image	Degraded image	Direct method	Smooth method
			
			
			
			
			

Original image	Degraded image	Direct method	Smooth method