Study on Algorithms for Image Super-resolution based on Filtering and Learning Methods

March 2017

Muhammad Haris

Study on Algorithms for Image Super-resolution based on Filtering and Learning Methods

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2017

Muhammad Haris

”Bismillahi r-rahmani r-rahim”
In the name of Allah , the Entirely Merciful, the Especially Merciful.

”Laa ilaha illa-llah, muhammadur-rasulu-llah”
There is no God but Allah, and Muhammad is his messenger

“...Not even the weight of an atom that is within the Heavens nor in the Earth is hidden from Him...” (Surah Saba, Verse 3)

Acknowledgements

Spending 5 years living and studying in Tsukuba University is undeniably one of life episodes that I will cherish the most. During these years I met truly impactful people that broaden my perspective, and without their support (in any respect), this study would never been completed. Especially, I must express my deepest gratitude to my academic advisor, Professor Hajime Nobuhara for his endless support and kind assistance during this past 6 years. From him I learn how to see the problem, and derive the optimal solution.

I am very much indebted to Prof. Tsutomu Maruyama and Prof. Itaru Kitahara for tremendous support and suggestion during research process. I also thank my thesis committee members: Prof. Yasunori Endo and Prof. Shibuya Sensei for review and comments. I would also like to extend my heartfelt gratitude to CMU Lab Members for the warmth and supportive environment during my study, especially Hashikami Hidenobu who introduce me to image super-resolution.

This sincere gratitude also goes to my favorite brothers, Abdul Karim and Mahmoud Ben Othman thank you for becoming more than good friends and may Allah pour you with his blessings.

I gratefully acknowledge the scholarship I received during doctoral's degree as this journey would not have been accomplished without financial support from Indonesia Endowment fund (LPDP).

Most important, I want to thank the core supporter of my study. My parents, sisters and brother, thank you for showering me with abundant cheering and unending support. For my wife, Dita, who has been through thick and thin for these past three years, thank you for the continuous encouragement. For Shofiyyah Kyoko, our first baby daughter, thank you for becoming your parents' source of happiness and strength. This thesis is dedicated for both of you.

Abstract

This thesis focuses on developing theory and algorithms for the single-image super-resolution problem based on filtering and learning methods. Our proposed methods are divided into three categories.

First part, First-order Derivatives- based Super-resolution is filtering based method. A single fast super-resolution method based on first-order derivatives from neighbor pixels is proposed. The basic idea of the proposed method is to exploit a first-order derivatives component of six edge directions around a missing pixel; followed by back projection to reduce noise estimated by the difference between simulated and observed images. Using first-order derivatives as a feature, the proposed method is expected to have low computational complexity, and it can theoretically reduce blur, blocking, and ringing artifacts in edge areas compared to previous methods. Experiments were conducted using 900 natural grayscale images from the USC-SIPI Database. We evaluated the proposed and previous methods using peak signal-to-noise ratio, structural similarity, feature similarity, and computation time. Experimental results indicate that the proposed method clearly outperforms other state-of-the-art algorithms such as fast curvature based interpolation.

Second part, Super-Resolution via Adaptive Multiple Sparse Representation is learning based method. We propose a super-resolution algorithm based on adaptive sparse representation via multiple dictionaries for images taken by Unmanned Aerial Vehicles (UAVs). The super-resolution attainable through the proposed algorithm can increase the precision of 3D reconstruction from UAV images, enabling the production of high-resolution images for constructing high-frequency time series and for high-precision digital mapping in agriculture. The basic idea of the proposed method is to use a field server or ground-based camera to take training images and then construct multiple pairs of dictionaries based on selective sparse representations to reduce instability during the sparse coding process. The dictionaries are classified on the basis of the edge orientation into five clusters: 0, 45, 90, 135, and non-direction.

The proposed method is expected to reduce blurring, blocking, and ringing artifacts especially in edge areas. We evaluated the proposed and previous methods using peak signal-to-noise ratio, structural similarity, feature similarity, and computation time. Our experimental results indicate that the proposed method clearly outperforms other state-of-the-art algorithms based on qualitative and quantitative analysis. In the end, we demonstrate the effectiveness of our proposed method to increase the precision of 3D reconstruction from UAV images.

Last part, Deep Residual Learning Super-resolution is learning based method. The light and efficient residual network for super-resolution is proposed. We adopt inception module from GoogLeNet to exploit the features from the low-resolution images and residual learning to have fast training steps. The proposed network called Deep Residual Learning Super-resolution (DRLSR). The network is proven to have fast convergence and low computational time. It is divided into three parts: feature extraction, mapping, and reconstruction. In the feature extraction, we apply inception module followed by dimensional reduction. Then, we map the features using simple convolutional layer. Finally, we reconstruct the HR component using inception module and 1×1 convolutional layer. The experimental results show our proposed method can reduce more than half of computational time from the-state-of-the-art methods, while still having clean and sharp images.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Computer vision is a multidisciplinary field that deals with how computers can be use to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do [33]. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images. It deals with the extraction of high-dimensional data from the real world in order to represent it as numerical or symbolic information. There are many computer vision algorithms such as object recognition and object tracking. To get accurate result, the input images must be in acceptable quality and resolution. However, numerous object was taken in low-resolution (LR) due to several reason such as small charge-coupled device (CCD) sensors or image compression. Therefore, the ability of super-resolution (SR) to create high-resolution (HR) image and enhance the quality to get more accurate result is necessary as shown in Fig. 1.1.

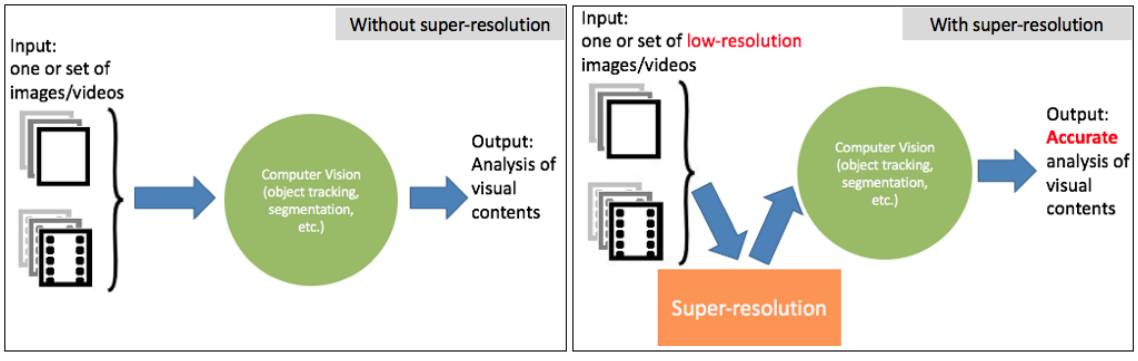


Figure 1.1: The use of super-resolution in computer vision task

SR algorithms were motivated to solve the problems caused by digital imaging devices [43]. The invention of digital scanners facilitate the conversion from paper-based docu-

ments into digital images. However, the image quality was poor, in low-resolution, and present some noise from COD sensors. With the goal of acquiring sharper and higher resolution image, SR algorithms were developed to combine multiple input LR from repeatedly scanning the same document with shifts and rotations.

Digital image data are unfortunately often at a lower quality than the desired one, because of several possible causes: spatial and temporal down-sampling due to noise degradation, high compression, etc. When we consider still images, the new sources of image contents, like the Internet or mobile devices, have generally a lower quality than high-definition display standard. Moreover, if we consider the past production, there is an enormous amount of images collected in the years, that are valuable but have a poor quality. The need of increasing the resolution of an image can also be required by the particular application context. Many applications, e.g. video surveillance and remote sensing, in fact, require the display of images at a desired resolution, for specific computer vision tasks like object recognition, zoom-in operations, or 3D reconstruction. For example, in Fig. 1.2, we can clearly see that after preprocessing using SR algorithm, the accuracy of 3D model increased. From these reasons, the urgency to improve the image quality is very important issue.

Figure 1.2: The example of super-resolution algorithm in 3D reconstruction

With the improvement of computational capability and mobile imaging devices, single SR has gained more attention with proven success. The fundamental difference is the number of input LR images required for SR to produce HR image. Since there is merely one input image, the formulation becomes an under-determined problem rather than an over-determined one as posited in the classical SR research. Because the problem is ill-posed and the available image data are limited, priors are exploited in the process to determine the generated pixel intensities. Numerous methods have been proposed based on different image properties and they can be roughly categorized into two approaches: filtering-based (non-learning) and learning-based.

Filtering methods include, among others, analytic interpolation methods, e.g. traditional bilinear and bicubic interpolation, which compute the missing intermediate pixels in the enlarged HR grid by averaging the original pixel of the LR grid with fixed filters. Edge-direction-based algorithms have been used to improve the limitation of traditional methods by exploiting local features such as edges by adapting each interpolating surface locally and assuming local regularity in a curvature. Once the input image has been upscaled to HR via interpolation, image sharpening methods can be applied. Sharpening methods aim at amplifying existing image details, by changing the spatial frequency amplitude spectrum of the image: in this way, the existing high frequencies in the image are enhanced, thus producing a more pleasant and richer output image.

Starting nineties, many powerful algorithms have been developed to solve different problems in a variety of scientific areas. Among single-image SR methods, the other important category is represented by algorithms that make use of machine learning techniques or learning-based approach. Although covering different meanings, machine learning can be generally referred to as that branch of artificial intelligence that concerns the construction and study of algorithms that can learn from data. In SR, learning method aims at estimating missing high-resolution detail that is not present in the original image, by adding new plausible high frequencies from the training data.

Several fundamental questions are still remained for single SR. In this thesis, we aim to address some of these important issues. For example, what are the important structures that can exploit and ensure for high-quality results? How to learn generating high-resolution image patches from low-resolution with and without learning process? In summary, single SR involves exploiting rich information contained in a single image. The challenges of single SR include recognizing important visual artifacts, refilling the HR details, and rendering them as faithfully and aesthetically pleasing as possible to be able to increase more accurate result on doing computer vision task. Addressing these challenges effectively and efficiently is the main motivation behind the research in this thesis.

1.2 Organization

Interested in the SR approach to the task of increasing the resolution of an image, and intrigued by the effectiveness of filtering- and learning-based techniques, during this doctorate we mostly investigated the SR problem and the application to it. On filtering based SR, we focus on reducing computational complexity by using only first-order derivative which involve only subtraction operator. In the other hand, learning-based SR procedures are patch-based procedures: the input image is partitioned into patches and from a single

LR input patch a single HR output patch is estimated via learning methods by learning the correspondences stored in the learned system. Finally, the whole set of estimated HR patches is then reconstruct to finally build the super-resolved image.

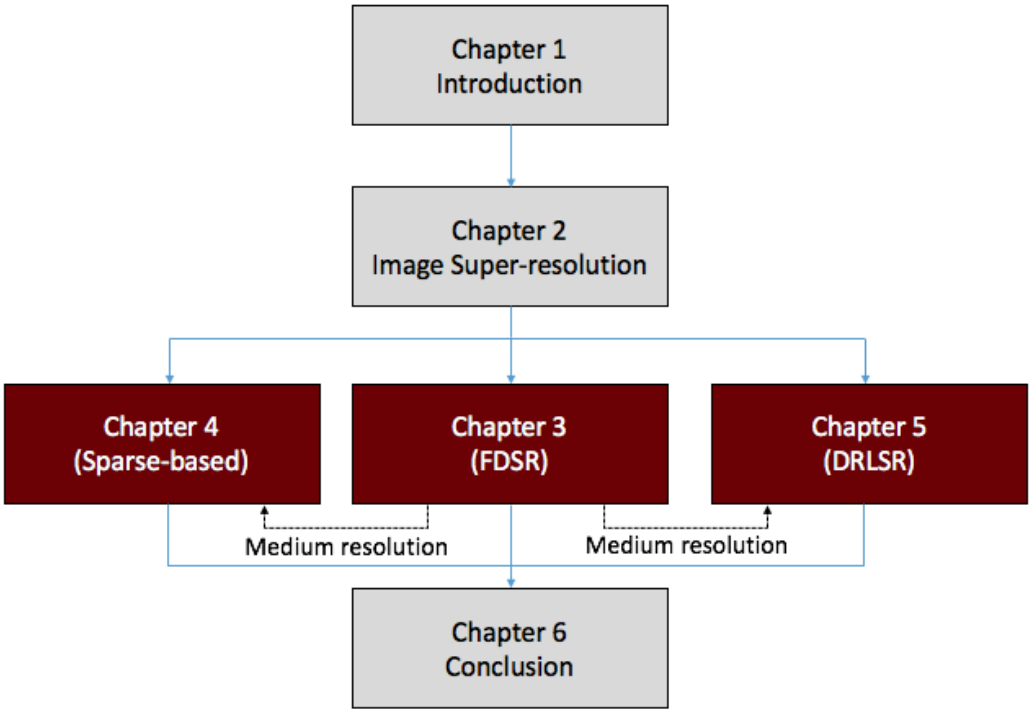


Figure 1.3: Research flowchart

The rest of this manuscript is structured as illustrated in Fig. 1.3. We start with Chapter 1 by explaining the introduction and motivation of our thesis. In Chapter 2, we give a general overview of SR and going deeper into the classification. The novel filtering based methods presented in Chapter 3 are a single fast SR method based on first-order derivatives from neighbor pixels which exploit a first-order derivatives component of six edge directions around a missing pixel; followed by back projection to reduce noise estimated by the difference between simulated and observed images. In Chapter 4, we presented an SR algorithm based on adaptive sparse representation via multiple dictionaries for images taken by Unmanned Aerial Vehicles (UAVs) which construct multiple pairs of dictionaries based on selective sparse representations to reduce instability during the sparse coding process. Then, to deal with very high non-linear relation between high- and low resolution images, we exploit the deep learning capability to propose efficient and fast architecture of convolutional neural networks based SR in Chapter 5. Finally, in Chapter 6 we end the thesis by summarizing our accomplishments, drawing conclusions from them and discussing about future directions.

Chapter 2

Image Super-resolution

2.1 Introduction

Super-resolution (SR) is the process of obtaining high-resolution (HR) image from one or more input low-resolution (LR). Numerous SR algorithms have been proposed and attracted many researchers to investigate the theory and application of SR [29]. It is found that SR can be applied in many practical applications such as image and video enhancement, medical images analysis, text analysis, satellite imaging, facial recognition. They are mainly divided based on the input and output image assumptions which can be categorized into two different types: spatial or temporal. In the spatial domain, SR aims to create an image with higher resolution and sharper image. While in the temporal domain, SR aims to insert extra frames in the video. Spatial SR or image SR has many applications and is the focus of this thesis. In the following, the term SR refers to algorithms in the spatial domain unless mentioned otherwise.

Depending on the input image, SR is mainly divided into two types: single- and multi-image SRs. Multi-image SR requires multiple images to acquire intrinsic characteristics. It then combines the information to construct a higher resolution image. Multi-image SR is highly suitable for video enlargement. It can exploit intrinsic characteristics that may differ from one sequence to another as illustrated in Fig. 2.1. For example, Liu et al. [25] proposed a Bayesian approach to adaptive video SR that involved simultaneous estimation of underlying motion, blur kernel, and noise level to reconstruct original HR frames; however, this approach has high computational complexity. Furthermore, the accuracy of multi-image SR is highly dependent to the variation of input LR images which is unnatural to obtain multiple images using common camera with different and complex motion, and known parameters.

The other method, single-image SR, requires only a single image to construct a higher resolution image. Single-image SR typically exploits the characteristics of the input image

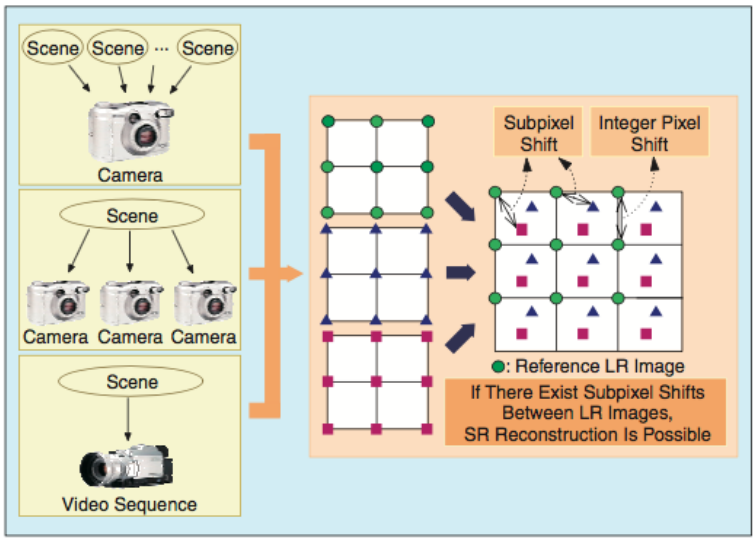


Figure 2.1: Basic premise for multi-image super-resolution [31]

and uses prior knowledge to learn the relationship between the LR and HR images. Single-image SR filled the missing pixels by observed the input LR or training data as illustrated in Fig. 2.2. Therefore, in this thesis, we focus on single-image SR which is highly applicable to the real world.

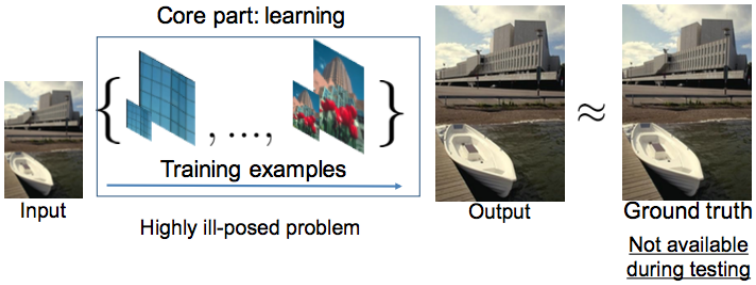


Figure 2.2: Basic premise for single-image super-resolution

Based on the approaches [29], single-image SR can be divided into three approaches: filtering-based, learning-based (non-direct examples), and reconstruction-based (direct examples). Each approach has published many research papers and designed for both specific and general purpose. However, the reconstruction-based method is eliminated from this dissertation because it requires high computational load for searching adequate instances in the exemplar set. If the exemplar set is large, the load for searching adequate exemplars will be high. Moreover, reconstruction approaches did not require training phase which make direct learning to the examples and produce more noise and instability during enlargement process.

The filtering methods were proven to have short computational time. However, it is hard to achieve the optimal result. Furthermore, the learning-based methods was able to accurately estimate the HR information by using training data but require long computational time. The more detail description of these two approaches are explained in the following sections.

2.2 Filtering-Based Approaches

Filter-based approaches focus on obtaining reasonably good result with short computational time. The focus of this approach is to be able to minimize the use of computational resource and mainly works on spatial domain. The first conventional methods utilize low complexity and easy implementation. The classic nearest neighbor, bilinear, and bicubic interpolation methods have been widely applied for real-time processing in image viewers and image-processing tools [30]. However, these methods produce unnatural images due to excessive blurring and jagged artifacts [2]. Such conventional methods do not use a prior model between HR and LR images, which plays a strong role in algorithm performance relative to quality improvement.

Edge-direction-based algorithms, often called edge-adaptive algorithm, have been used to overcome that limitation by exploiting local features such as edges [14, 24, 17, 18, 39]. For example, new edge directed interpolation (NEDI) [24] provides good results by adapting each interpolating surface locally and assuming local regularity in a curvature. Fast curvature based interpolation (FCBI) [14], inspired by NEDI, obtains interpolated pixels by averaging two pixels determined by second-order directional derivatives of image intensity. An improved version of the FCBI algorithm, i.e., iterative curvature based interpolation (ICBI), which optimizes interpolated pixels using iterative correction has been introduced [14]. Haris et al. [18] proposed the improvement of FCBI algorithm by introducing single-image SR that extends from two to six directions and accommodates a wide range of the interpolating directions of the missing pixels, then improve the result by back-projection algorithm.

Many researcher explore on this approach because it is highly suitable for real-time application due to its low computation and insensitivity to training data. Moreover, this type of SR is very easy to implement. However, the result cannot produce sharper and clearer HR image compare to learning-based approach.

2.3 Learning-Based Approaches

Learning-based SR were first introduced in 1985 by [28] which used neural-network to improve the resolution of fingerprint images. This approach can be divided into two types of input domain: spatial- and frequency-based. In the spatial-based approaches, the SR algorithm directly extracts the features or high frequency components from the pixel values. However, in frequency-based approach, the input image first transforms to the frequency domain, such as wavelet transform and fourier transform, then transforms back to spatial domain.

Takeda et al. [36] generalized the use of spatially adaptive (steering) kernel regression, which produces results that preserve and restore details with minimal assumptions about local signal and noise models. An improvement of previous algorithm also proposed using adaptive enhancement and spatiotemporal up-scaling of videos without explicit motion estimation [37]. However, this method is not robust and is sensitive to parameters such as smoothing.

Danielyan et al. [6] proposed spatially adaptive filtering in the image domain and projection in a wavelet domain. Mallat et al. [27] introduced a class of inverse problem estimators computed by adaptively mixing a family of linear estimators corresponding to different priors computed over a wavelet frame. Demirel et al. [7] investigated discrete wavelet transform to decompose the input image into different sub-bands. Celik et al. [4] exploit a forward and inverse dual-tree complex wavelet transform to construct an HR image from the given LR image. However, these methods are computationally very complex.

SR using sparse representation has become popular because of its ability to naturally encode the semantic information of images [10]. By collecting representative samples in order to create an over-completed dictionary, it is possible to discover the correct basis for correctly encoding an input image. The studies by Yang et al. [44] and Zeyde et al. [45] focused on using a single pair of dictionaries; intuitively, however, using a single pair of dictionaries can produce many redundancies, which may cause instability during the image reconstruction process.

The latest convolutional neural networks (CNNs) is used in many image processing algorithm with large improvement in accuracy. On SR algorithm, Cao dong et al.[8] has demonstrated a CNNs' ability mapping LR to HR patches called Super-resolution Convolutional Neural Networks (SRCNN). The method is constructed by a very simple and a lightweight structure CNNs using two hidden layers and 3×3 filter size. Jiwon Kim et al. [22] introduces Very Deep Convolutional Networks (VDSR), a very deep CNN with residual learning, which proven have accurate result but have critical issues on convergence

speed. VDSR includes 20 layer of CNN using 3×3 filter size. The recent improvement has been published. FSRCNN [9] demonstrated superior performance than previous SRCNN. They focused on improving the current SRCNN and proposed faster and more accurate algorithm. FSRCNN redesign the network using three main principal: deconvolution, dimension shrinking, and smaller filter.

2.4 Our Contributions

The SR algorithm is the core algorithm to support computer vision tasks, such as pattern recognition and 3D reconstruction. It has the ability to transform the input image/video to acceptable resolution for improving the accuracy of computer vision tasks. However, in terms of the application, the requirements of each task are different and unique. For example, in video streaming application, the SR algorithm has to offer low computation algorithm without the use of training data to avoid the bottleneck during data transfer in the network. In the application for satellite images, the training data is limited, the proposed SR algorithm should be insensitive to training data. Moreover, in 3D reconstruction, the details and quality of input images are necessary, we should use many training data to improve the proposed SR algorithm.

The existing SR problems solved by varieties solutions offered by researchers. The same with our research, we aim to offer various solutions which suitable for many applications depend on the requirements. Nowadays, the researchers focus on dividing SR based on the theoretical approach as mentioned in the previous section. However, in the application problems, the author found three main problems existed during SR algorithm implementation: computational time, sensitivity to training data, and quality improvement. Therefore, in this dissertation, we deeply investigate the SR based on the application problems which divided into three categories: non training data, limited training data, and unlimited training data.

On non training data approaches which is low computational process, we proposed filtering based methods using first-order derivatives from neighbor pixels on six edge directions around a missing pixel, then followed by back projection to reduce noise estimated by the difference between simulated and observed images. The next proposed method is insensitive to training images. We develop an adaptive sparse representation via multiple dictionaries based on selective sparse representations to reduce instability during the sparse coding process using limited training data. Last, we propose a method where training data is unlimited. In this case, we propose to use convolutional networks which has been proven to construct the best image quality.

In details, we also show the importance of feature variation in developing SR algorithms. In our proposed methods, we focus to use multiple features, such as multiple edge direction and convolution filter, to extract the contextual information from the input images or videos. Moreover, we show that multiple feature extractions are not only able to increase the quality of SR result, but also deliver efficient and low computation algorithm if treated correctly based on the nature of the images.

In summary, we offer the solution for different problems based on the main implementation problem. We aim to develop SR algorithm as a service where the end user can easily choose the required SR algorithm for each application. With many application requirements, the end user can use our proposed methods easily and produce the expected result.

Chapter 3

First-order Derivatives- based Super-resolution

3.1 Introduction

The need for a fast super-resolution (SR) method has become increasingly necessary due to increased availability of SR hardware such as televisions and smartphones, which have low computational capacity. Mobile devices have limited ability to enlarge images and videos, which are still available in lower-resolution formats (such as older videos on the Internet). The primary problem of an enlarging process is to predict missing areas using existing pixels. Therefore, developing an algorithm to predict the most suitable pixel value in the missing area effectively is extremely challenging.

Depending on the input image, SR is primarily divided into two types, i.e., single- and multi-image SRs. Multi-image SR requires multiple images to acquire intrinsic characteristics. It then combines the information to construct a higher resolution image. However, in daily applications, it is unnatural to obtain multiple images using common camera with known parameters. Single-image SR requires only a single image to construct a higher resolution image. Single-image super-resolution typically exploits the characteristics of the input image and uses prior knowledge to learn the relationship between the low- (LR) and high-resolution (HR) image. Therefore, our proposed method uses single-image SR which is highly applicable to the real world.

Utilizing their low complexity and easy implementation, classic nearest neighbor, bilinear, and bicubic interpolation methods have been widely applied for real-time processing in image viewers and image-processing tools [30]. However, these methods produce unnatural images due to excessive blurring and jagged artifacts [2]. Such conventional methods do not use a prior model between HR and LR images, which plays a strong role in algorithm performance relative to quality improvement.

Multi-image SR is highly suitable for video enlargement. It can exploit intrinsic characteristics that may differ from one sequence to another. Liu et al. [25] proposed a Bayesian approach to adaptive video SR that involved simultaneous estimation of underlying motion, blur kernel, and noise level to reconstruct original HR frames; however, this approach has high computational complexity.

Takeda et al. [36] generalized the use of these techniques to spatially adaptive (steering) kernel regression, which produces results that preserve and restore details with minimal assumptions about local signal and noise models. An improvement that uses adaptive enhancement and spatiotemporal up-scaling of videos without explicit motion estimation has been proposed [37]. However, this method is not robust and is sensitive to parameters such as smoothing.

Danielyan et al. [6] proposed spatially adaptive filtering in the image domain and projection in a wavelet domain. Yang et al. [44] designed a pair of sparse to construct an HR image. Mallat et al. [27] introduced a class of inverse problem estimators computed by adaptively mixing a family of linear estimators corresponding to different priors computed over a wavelet frame. Demirel et al. [7] used discrete wavelet transform to decompose the input image into different sub-bands. Celik et al. [4] used a forward and inverse dual-tree complex wavelet transform to construct an HR image from the given LR image. However, these methods are computationally very complex.

Edge-direction-based algorithms, often called edge-adaptive algorithm, have been used to overcome that limitation by exploiting local features such as edges [14, 24, 17]. For example, new edge directed interpolation (NEDI) [24] provides good results by adapting each interpolating surface locally and assuming local regularity in a curvature. Fast curvature based interpolation (FCBI) [14], inspired by NEDI, obtains interpolated pixels by averaging two pixels determined by second-order directional derivatives of image intensity. An improved version of the FCBI algorithm, i.e., iterative curvature based interpolation (ICBI), which optimizes interpolated pixels using iterative correction has been introduced [14].

Learning from the FCBI algorithm, we propose single-image SR that extends from two to six directions and accommodates a wide range of the interpolating directions of the missing pixels. The use of first-order derivatives can reduce computational complexity because the main process uses only a subtraction operator. As mentioned before, previous interpolation methods have several drawbacks, including (1) blurring, blocking, and ringing artifacts in edge areas; (2) less smoothness along edges; (3) discontinuity along edges; and (4) high computational complexity. Therefore, a simple and fast mechanism to interpolate edges based on the largest first-order derivatives is proposed to solve these problems. Ta-

ble 3.1 shows a comparison of the proposed method and previous methods based on our experiment results.

Table 3.1: Comparison between proposed algorithm and previous methods (◯ = good, △ = normal, × = not good)

Method	Computation Time	Image Quality
Nearest neighbor	◯	×
Bilinear	◯	×
Bicubic	△	△
KR[37]	△	△
SpR[44]	×	◯
SME[27]	×	◯
FCBI[14]	◯	△
ICBI[14]	△	◯
NEDI [24]	△	△
DFDF[47]	△	△
Proposed	◯	◯

This remainder of this paper is organized as follows. Section 3.2 explains the FCBI algorithm. Section 3.3 introduces the proposed algorithm and a flowchart of the system. Section 3.4 demonstrates the results of experiments and analysis on the basis of four factors: peak signal-to-noise ratio (PSNR), structural similarity (SSIM), feature similarity (FSIM) index, and computational time.

3.2 Fast Curvature Based Interpolation

Here, we describe the FCBI algorithm [14]. First, we present the mechanism by which the FCBI algorithm copies original pixels to a new enlarged grid and then performs edge confirmation. Next, we describe the calculation of edge direction. After the edge direction is obtained, new interpolated pixels are calculated on the basis of the direction. Last, we discuss the limitation of FCBI and describe how the proposed method can outperform FCBI.

3.2.1 Copying original pixels (FCBI Step 2.2.1)

The original image is formulated with M as height and N as width, where total pixels is $M \times N$. Let $X = \{x_m : m = 0,1,2,...,M-1\}$ and $Y = \{y_n : n = 0,1,2,...,N-1\}$ be finite

sets that determine the number of pixels. The original image is defined as the function $f : X \times Y \rightarrow I$ where $I = \{0,1,2,...,255\}$ is the value of each pixel. Then, an enlarged grid $(M^{'} \times N^{'})$ applies the following condition, $M^{'} > M$ and $N^{'} > N$. Let $X^{'} = \{x^{'}_{m} : m = 0,1,2,...,M^{'}-1\}$ and $Y^{'} = \{y^{'}_{n} : n = 0,1,2,...,N^{'}-1\}$ be finite sets that determine the number of pixels in the enlarged image.

Technically, FCBI image $I_f(x_p, y_q)$ copies all pixels from the original image and then calculates missing pixels using first- and second-order derivatives filter. The FCBI image's size is powered by order two from the original image. The total number of pixels is $M^{'} \times N^{'}$, where $M^{'} = ((M \times 2^{zk}) - (2^{zk} - 1))$, $N^{'} = ((N \times 2^{zk}) - (2^{zk} - 1))$, and $zk \in \mathbf{Z}$ (zk is the interpolation zoom factor). To obtain the relation between $(m^{'}, n^{'})$ and (m,n), we calculate $m = \lfloor((m^{'} - 1)/2)\rfloor$ and $n = \lfloor((n^{'} - 1)/2)\rfloor$ if $m^{'}$ and $n^{'}$ are available. The floor function is defined as $\lfloor x \rfloor = max\{r \in Z | r \leq x\}$, where x is a real number and $\{Z, r\}$ are sets of integers, i.e., positive, negative, and zero.

3.2.2 Checking edge or texture (FCBI Step 2.2.2)

In signal processing, first-order derivatives can be represented as three types of signals: discontinuity (edge), texture, and smoothness. In this step, FCBI uses a parameter of constant value to determine the edge. Here, v_1, v_2, p_1, p_2, and TM are parameters used to interpolate $I_f(i,j)$ based on Algorithm 1, where v_1 and v_2 are first-order derivatives of intensity in the particular coordinates, p_1 and p_2 are the average of two neighbor pixels, I_1, I_2, I_3, I_4 are neighbor pixels, and TM is a constant value as shown in Fig. 3.1.

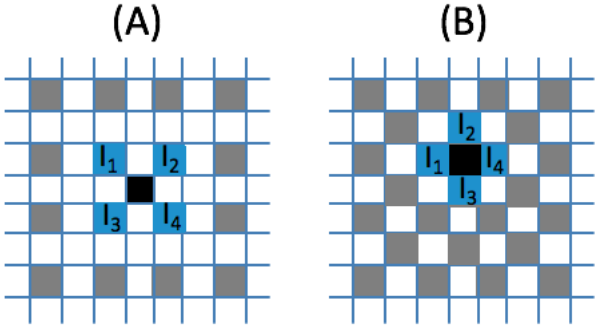


Figure 3.1: Checking edge or texture (FCBI Step 2.2.2): (A) Diagonal left-right; (B) horizontal-vertical (I_1, I_2, I_3, and I_4 are neighbor pixels)

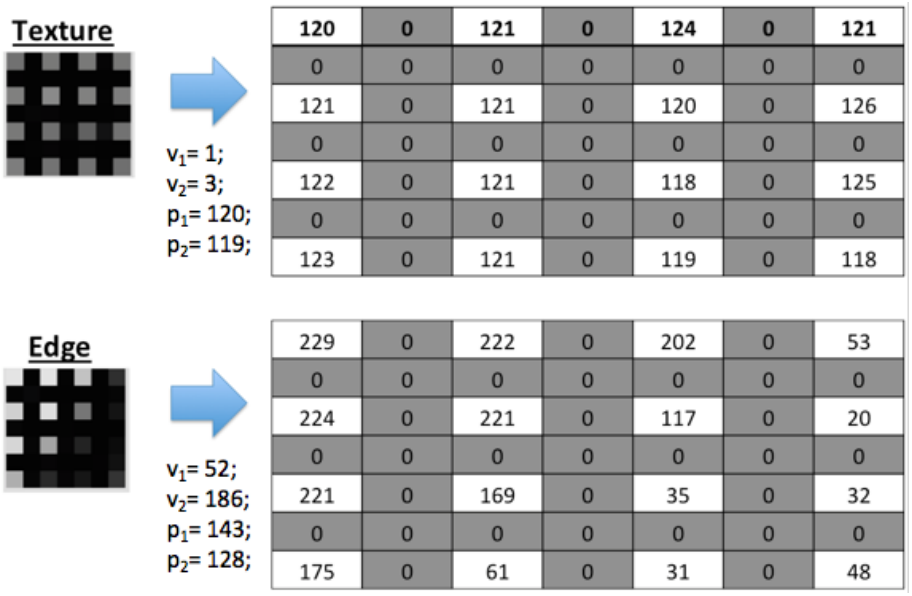
Figure 3.2 illustrates the edge determination step. First, we calculate the value of parameters v_1, v_2, p_1, and p_2 on the basis of Algorithm 1. In Fig. 2, the discontinuity from the corresponding pixels is clearly shown in the edge area. The texture area shows only gra-

dation of texture without clear discontinuity. Using the constant value TM as a threshold, FCBI can obtain the edge area using simple calculation.

Algorithm 1: FCBI Step 2.2.2: Interpolation for edge area [14]

Result: Missing pixel $I_f(i,j)$

initialization;
$v_1 = |I_1 - I_4|$;
$v_2 = |I_2 - I_3|$;
$p_1 = (I_1 - I_4)/2$;
$p_2 = (I_2 - I_3)/2$;
if $(v_1 < TM \wedge v_2 < TM \wedge |p1 - p2| < TM)$ **then**
    it is not edge, go to (Step 2.2.3)
else
    it is edge;
    if $v_1 < v_2$ **then**
        $I_f(i,j) = p_1$
    else
        $I_f(i,j) = p_2$



120	0	121	0	124	0	121
0	0	0	0	0	0	0
121	0	121	0	120	0	126
0	0	0	0	0	0	0
122	0	121	0	118	0	125
0	0	0	0	0	0	0
123	0	121	0	119	0	118

229	0	222	0	202	0	53
0	0	0	0	0	0	0
224	0	221	0	117	0	20
0	0	0	0	0	0	0
221	0	169	0	35	0	32
0	0	0	0	0	0	0
175	0	61	0	31	0	48

Figure 3.2: Sample of FCBI Step 2.2.2. For the texture image, $v_1 = 1, v_2 = 3, p_1 = 120$, and $p_2 = 119$. For the edge image, $v_1 = 52, v_2 = 186, p_1 = 143, p_2 = 128$.

3.2.3 Checking texture direction (FCBI Step 2.2.3)

After the edge area is determined, each identified texture area requires a further calculation to obtain the interpolation direction. This stage uses a second-order derivatives filter from

eight-neighbor pixels. The equation is expressed as follows [14].

$$\begin{aligned}\tilde{I}_1(x_{m'},y_{n'}) = {} & f(x_{m-1},y_{n+1}) + f(x_m,y_n) \\ & + f(x_{m+1},y_{n-1}) - 3f(x_m,y_{n+1}) \\ & - 3f(x_{m+1},y_n) + f(x_m,y_{n+2}) \\ & + f(x_{m+1},y_{n+1}) + f(x_{m+2},y_n),\end{aligned} \tag{3.1}$$

$$\begin{aligned}\tilde{I}_2(x_{m'},y_{n'}) = {} & f(x_m,y_{n-1}) + f(x_{m+1},y_n) \\ & + f(x_{m+2},y_{n+1}) - 3f(x_m,y_n) \\ & - 3f(x_{m+1},y_{n+1}) + f(x_{m-1},y_n) \\ & + f(x_m,y_{n+1}) + f(x_{m+1},y_{n+2}).\end{aligned} \tag{3.2}$$

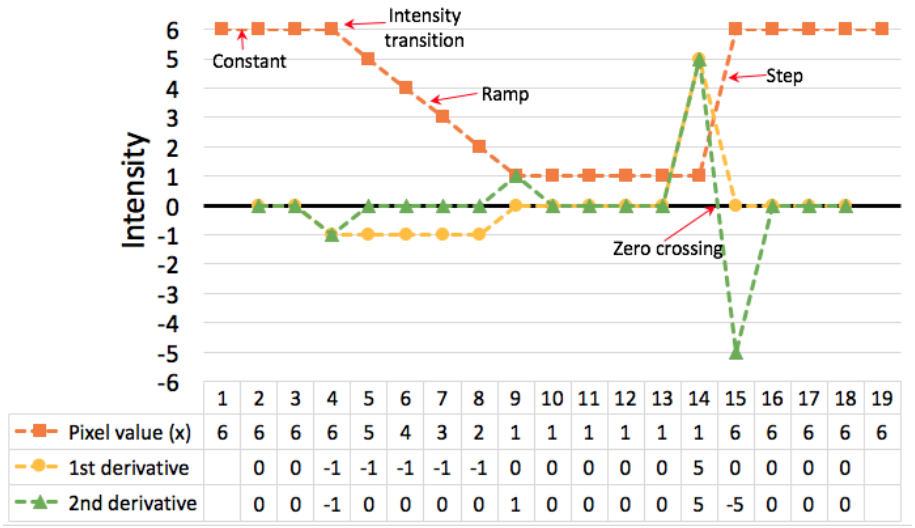
3.2.4 Interpolating missing pixel (FCBI Step 2.2.4)

After all direction calculations are completed, as shown in Eqs. (1) and (2), we interpolate $I_f(x_{m'},y_{n'})$ by calculating the average of two neighbors in the direction wherein the derivative is lower, which is expressed as follows [14].

$$\begin{aligned}I_f(x_{m'},y_{n'}) = {} & \begin{cases} \left\lfloor \dfrac{a_1+b_1}{2} \right\rfloor ; \text{ if } \tilde{I}_1(x_{m'},y_{n'}) < \tilde{I}_2(x_{m'},y_{n'}) \\ \left\lfloor \dfrac{a_2+b_2}{2} \right\rfloor ; \text{ otherwise.} \end{cases} \\ & \text{Here,} \\ & a_1 = f(x_m,y_n) \\ & a_2 = f(x_{m+1},y_n) \\ & b_1 = f(x_{m+1},y_{n+1}) \\ & b_2 = f(x_m,y_{n+1})\end{aligned} \tag{3.3}$$

3.2.5 Limitation of FCBI

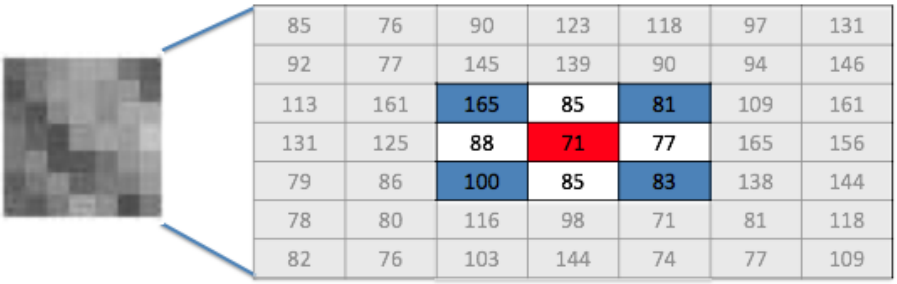
In Fig. 3.3, we illustrate how the first- and second-order derivatives work in a 1D signal. The first-order derivatives are useful for selecting the strongest edges by thresholding the gradient magnitude. The zero-crossings of the second-order derivatives are useful for localization of the edge. Both are used by FCBI to obtain the interpolation direction. Diagonal gradients of the surrounding blocks of missing pixels are used to ensure better detection of edge locations in natural images; then, the average of two neighbors from the directions are used to fill the missing pixel. However, the diagonal gradient is insufficient to accommodate all possible edge directions. This limitation can cause a blur effect and makes it



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Pixel value (x)	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	-1	-1	-1	-1	-1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	-1	0	0	0	0	1	0	0	0	0	5	-5	0	0	0	

Figure 3.3: First- and second-order derivatives

difficult to preserve the detail element during interpolation. Therefore, extending to six edge directions is proposed.



85	76	90	123	118	97	131
92	77	145	139	90	94	146
113	161	165	85	81	109	161
131	125	88	71	77	165	156
79	86	100	85	83	138	144
78	80	116	98	71	81	118
82	76	103	144	74	77	109

$p_1 = (100+81)/2 \approx 90$; $p_2 = (165+83)/2 = 124$

Figure 3.4: FCBI interpolation can cause some artifacts. Original pixels is 71, while interpolation from FCBI is $p_1 = 90$ or $p_2 = 124$

Figure 3.4 shows the differences between a real HR pixel value and FCBI results. Both directions ($p_1 = 90$ and $p_2 = 124$) have different values from the original pixel ($I(m,n) = 71$), which is shown in red. FCBI uses both first- and second-order derivatives filters to detect the interpolation direction, which still have limitations as shown by this case. Meanwhile, the proposed method only uses first-order derivatives, which can ensure lower computational complexity and better quality.

3.3 Proposed Method

In this section, the core algorithm of the proposed method is explained. Then, we describe backprojection, which is an algorithm to smoothen the high frequency of the proposed method's result. The proposed algorithm consists of three stages, i.e., initialization, interpolation, and smoothing. A complete flowchart of the proposed method is given in Fig. 3.5.

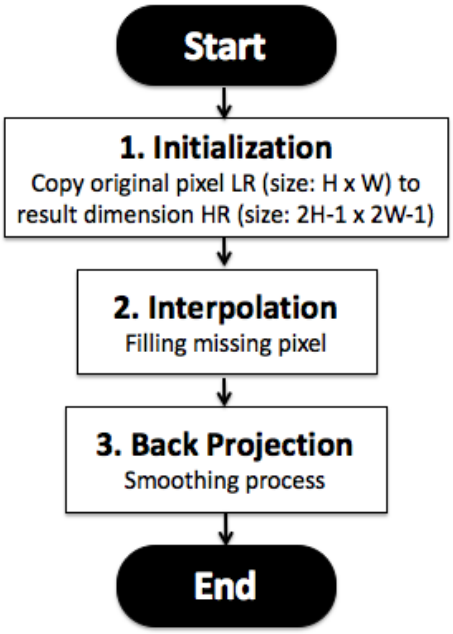


Figure 3.5: Flowchart of the proposed method

3.3.1 Initialization (Proposed Step 1)

The proposed method uses $2\times$ magnification. The total number of pixels is $M^{'} \times N^{'}$, where $M^{'} = ((M \times 2^{zk}) - (2^{zk} - 1))$, $N^{'} = ((N \times 2^{zk}) - (2^{zk} - 1))$, and $zk \in \mathbf{Z}$. Here, zk is the interpolation zoom factor and $M \times N$ is the width and height of an LR image. The initialization stage begins by copying all original pixels $I(m,n)$ to an enlarged grid $I_f(m^{'},n^{'})$ as shown in Fig. 3.6.

3.3.2 First-order derivatives interpolation (Proposed Step 2)

Many methods, such as the Sobel operator, have been developed to discover the direction of an edge. First-order derivatives, which have low computational complexity, are a common feature used to estimate edge direction. The core module of this algorithm is discussed in the following section.

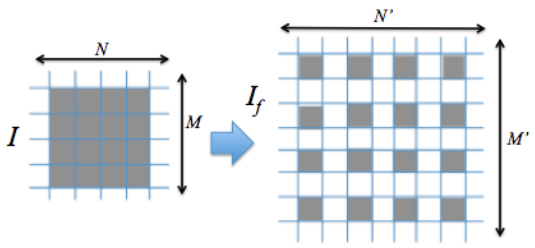


Figure 3.6: Proposed Step 1: Copy all original pixels $I(m,n)$ to an enlarged grid $I_f(m',n')$.

3.3.2.1 Edge direction and weight calculation

The interpolation step of the proposed method is divided into two stages: diagonal (Stage 1) and vertical-horizontal (Stage 2), as shown in Fig. 3.7.

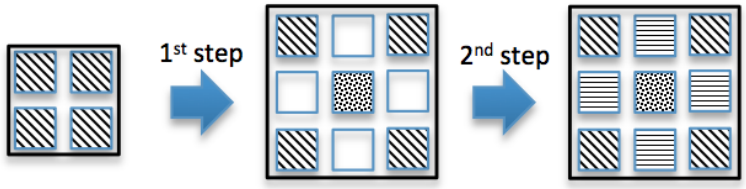


Figure 3.7: Interpolation in the proposed method. Stage 1 is diagonal; Stage 2 is vertical-horizontal.

The six directions proposed by our algorithm are shown in Figs. 3.8 and 3.9. In the figures, black corresponds to a missing pixel ($I_k(x_{m'}, y_{n'})$) and blue corresponds to neighbor pixels. In each direction, we group the pixels into two matrices a and b. This rule is also applied to the second stage of the interpolation.

Let K be the filter to calculate the weight from each element of neighbor pixels. Functions $diff1(a,b,t)$ and $diff2(a,b,t)$ are used to calculate the absolute difference between matrix a and b for each direction, where t is the interpolation direction. $diff1(a,b,t)$ calculates the direction where a missing pixel is interpolated from the side neighbor pixels (where t=$\{1,2,3,4\}$), and $diff2(a,b,t)$ calculates the direction where a missing pixel is interpolated from the crossing neighbor pixels (where t=$\{5,6\}$). These functions are expressed by Eqs. 4, 5, and 6.

$$
\begin{aligned}
&K = [1,-1,-1,1];\\
&dd_t = \begin{cases} K * diff1(a,b,t), & \text{if } t = \{1,2,3,4\} \\ K * diff2(a,b,t), & \text{else if } t = \{5,6\}; \end{cases} \\
&dh_t = \begin{cases} K * diff1(a,b,t), & \text{if } t = \{1,2,3,4\} \\ K * diff2(a,b,t), & \text{else if } t = \{5,6\}; \end{cases}
\end{aligned} \tag{3.4}
$$

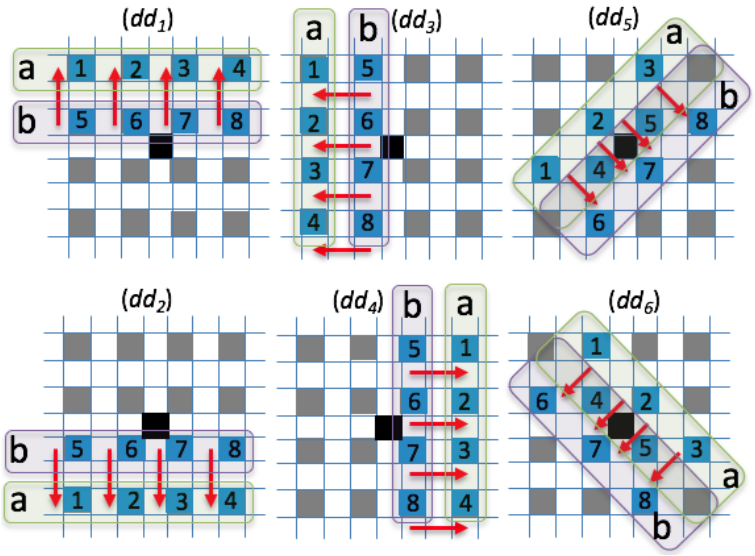


Figure 3.8: Stage 1 edge directions (dd_t)

$$diff1(a,b,t) = \begin{pmatrix} |a_1^t - b_5^t| \\ |a_2^t - b_6^t| \\ |a_3^t - b_7^t| \\ |a_4^t - b_8^t| \end{pmatrix} \tag{3.5}$$

$$diff2(a,b,t) = \begin{pmatrix} |a_1^t - b_6^t| \\ \left|\frac{a_2^t + a_4^t}{2} - \frac{b_4^t + b_7^t}{2}\right| \\ \left|\frac{a_2^t + a_5^t}{2} - \frac{b_5^t + b_7^t}{2}\right| \\ |a_3^t - b_8^t| \end{pmatrix} \tag{3.6}$$

3.3.2.2 Interpolation direction

The last step is to obtain the interpolation direction and calculate the missing pixel. In Fig. 3.10, black corresponds to the missing pixel $(I_k(x_{m'}, y_{n'}))$ and red corresponds to neighboring pixels.

This step begins after completing Proposed Step 2. We calculate the missing pixel by averaging the neighbor from the strongest direction. We determine the largest value of dd_t and dh_t and obtain the maximum index t_{max}. Then, we obtain the corresponding

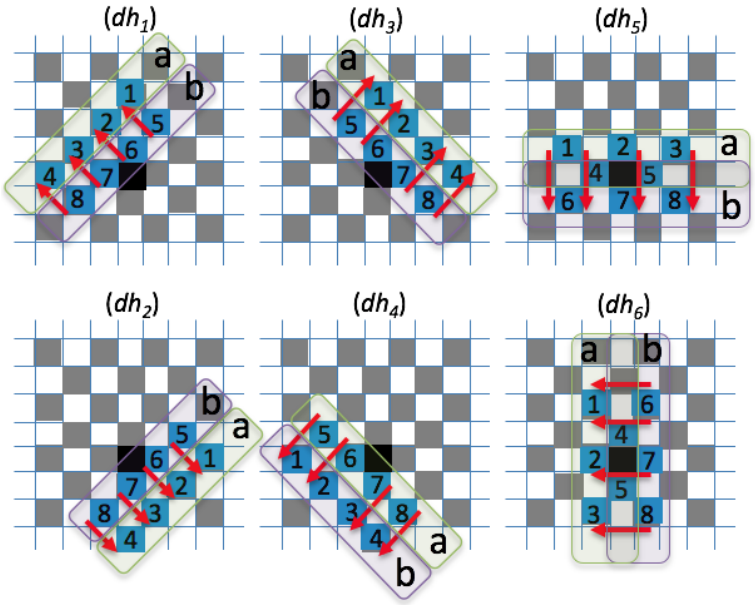


Figure 3.9: Stage 2 edge directions (dh_t)

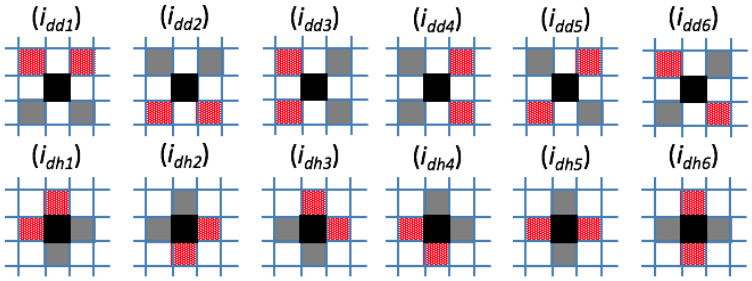


Figure 3.10: Interpolation pixels are calculated from the average of neighbor pixels (Stage 1: i_{ddt}) and (Stage 2: i_{dht})

interpolation value with the maximum index (Eq.3.7 for Stage 1 and Eq.3.8 for Stage 2.

$$\begin{aligned} dd_{max} &= max(dd_1, dd_2, dd_3, dd_4, dd_5, dd_6); \\ I_k(x_{m'}, y_{n'}) &= i_{dd_{max}} \end{aligned} \tag{3.7}$$

$$\begin{aligned} dh_{max} &= max(dh_1, dh_2, dh_3, dh_4, dh_5, dh_6); \\ I_k(x_{m'}, y_{n'}) &= i_{dh_{max}} \end{aligned} \tag{3.8}$$

3.3.3 Back Projection (Proposed Step 3)

Back projection is used to construct an image by taking each view and smearing the image. The HR image is estimated by back projecting the difference between the simulated and observed LR images. The process is iterated until some criterion is met, such as minimiza-

tion of the energy of the error, or the maximum number of allowed iterations is reached.

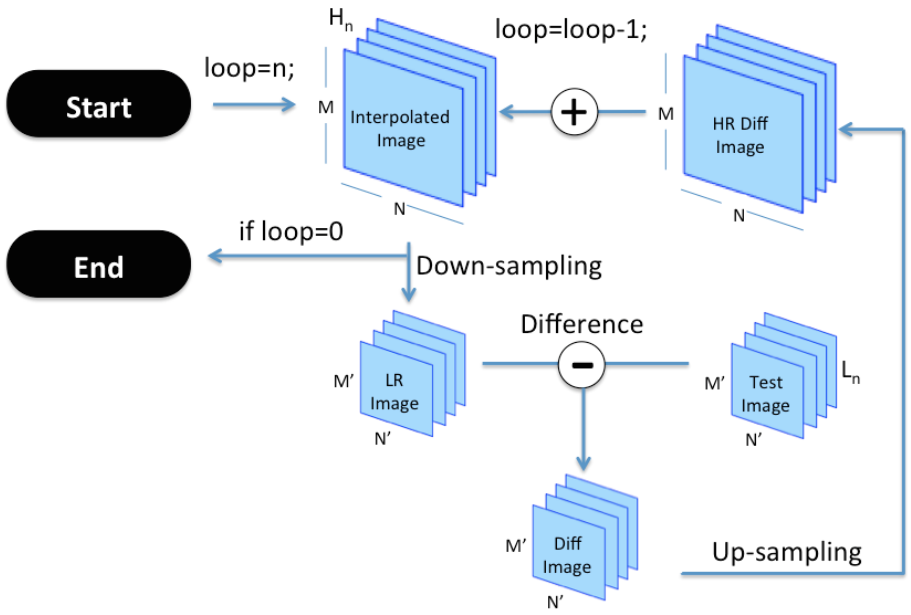


Figure 3.11: Back-projection flowchart

The process begins with the input LR and HR image. First, the initial HR image (interpolated image) is degraded and down-sampled to generate the observed LR image. The input LR image is subtracted from the observed LR image. Then, the difference is upscaled and added to the initial HR image. Generally, the HR image is estimated by a high-pass filter for edge projection and back-projecting the error (difference) from the simulated and the observed LR image. The back-projection step is illustrated in Fig. 3.11.

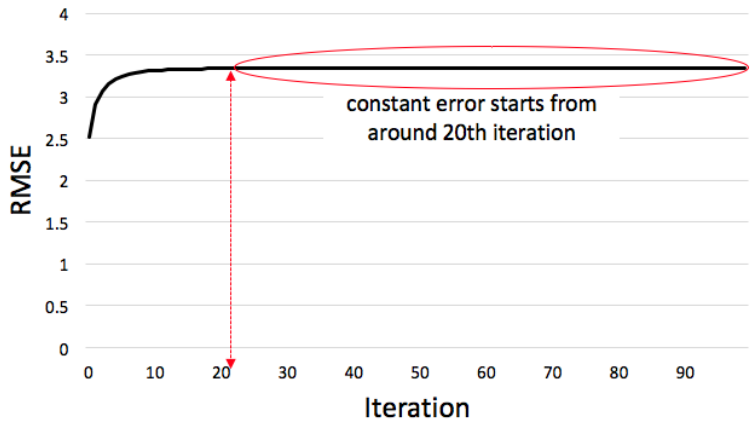


Figure 3.12: RMSE for each back projection iteration

In the proposed method, we use 20 iterations without other criteria. The number of iteration was selected to simplify the process and was chosen on the basis of the analysis of the experiment results. We found that 20 iterations were optimal to reduce image noise. In graph in Fig. 3.12 shows that the error is constant after 20 iterations.

3.4 Experimental conditions and results

We conducted experiments to confirm the efficiency of the proposed method and analyzed the result quantitatively and qualitatively. All experiments were conducted using Matlab R2009b on OS X Yosemite 10.10.3 (Intel Core i5@2.3GHz, 8GB RAM). We used an images dataset from the USC-SIPI Image Databases. The test images contained various patterns and natural objects. Example images from the dataset are shown in Fig.3.13. The image criteria for the experiments were grayscale images (intensity range, 8 bits), images (256 x 256 pixels), and using 900 images.

Figure 3.13: Example test images from the USC-SIPI Image Database

The experiments compared the observed images obtained by downsampling original images and enlarging the downsampled results using different methods and various scales. We compared 11 methods: nearest neighbor, bilinear, bicubic, kernel regression (KR) [37], sparse representation (SpR) [44], sparse mixing estimator (SME) [27], FCBI, ICBI [14], NEDI [24], directional filtering and data fusion (DFDF) [47], and the proposed method.

The algorithms used in the experiments have different characteristics. Therefore, to obtain objective comparisons, all parameters used in training and testing were similar to those recommended in the respective literature.

3.4.1 Quantitative analysis

PSNR [21], SSIM [41], FSIM [48], and computational time were used as quantitative measurements. The PSNR (dB) between the original image and the upscaled image was determined according to the literature [21]. The SSIM measures the quality of images on the basis of the structural content of the original image and the magnified image. FSIM is based on the fact that the HSV primarily understands an image according to its low-level features. Two features are considered in the FSIM computation: the primary feature (i.e., phase congruency), which is a dimensionless measure of a local structure's significance, and the secondary feature (i.e., image gradient magnitude). FSIM combines both features to characterize the local quality of the image. Higher PSNR, SSIM, FSIM indicate better quality. CPU time was computed using Matlab functions (tic and toc), which measure the elapsed time of a certain process.

Table 3.2: Average quantitative results (PSNR, SSIM, and FSIM for 2x magnification)

Methods	PSNR	SSIM	FSIM
Nearest neighbor	25.712 ± 3.55	0.776 ± 0.07	0.841 ± 0.04
Bilinear	26.118 ± 3.72	0.778 ± 0.07	0.830 ± 0.04
Bicubic	26.712 ± 3.75	0.803 ± 0.07	0.848 ± 0.04
FCBI	26.107 ± 3.70	0.782 ± 0.07	0.834 ± 0.04
ICBI	26.604 ± 3.73	0.802 ± 0.07	0.848 ± 0.04
DFDF	26.249 ± 3.72	0.787 ± 0.07	0.836 ± 0.04
NEDI	25.444 ± 3.61	0.758 ± 0.08	0.819 ± 0.04
SpR	25.884 ± 3.54	0.794 ± 0.07	0.846 ± 0.04
SME	26.521 ± 3.73	0.799 ± 0.07	0.845 ± 0.04
KR	26.085 ± 3.71	0.777 ± 0.07	0.818 ± 0.04
Proposed	$\mathbf{27.268} \pm 3.71$	$\mathbf{0.830} \pm 0.06$	$\mathbf{0.872} \pm 0.03$

Table 3.2 and 3.3 shows the qualitative results for images with 2x and 4x magnification. The average values from three measurements are provided. The results confirm that the proposed method outperforms the other methods. The proposed method achieves the best PSNR, SSIM, and FSIM values. As observed in Table 3.2 and 3.3, the proposed method gave higher PSNR, SSIM, and FSIM values for 2x magnification (approximately 4.5%, 6.1%, and 4.6%) and 4x magnification (approximately 4.4%, 7.3%, and 6.4%) respectively, compared to FCBI, which has relatively equal elapsed time.

Generally, the proposed method gave the highest values for PSNR, SSIM, and FSIM compared to the other methods, ranging from 2-7%, 3-9%, and 3-6% for 2x magnification and 1-12%, 2-15%, and 3-14% for 4x magnification.

Table 3.3: Average quantitative results (PSNR, SSIM, and FSIM for 4x magnification)

Methods	PSNR	SSIM	FSIM
Nearest neighbor	23.014 ± 3.32	0.615 ± 0.11	0.678 ± 0.06
Bilinear	23.459 ± 3.46	0.632 ± 0.11	0.731 ± 0.06
Bicubic	23.864 ± 3.51	0.653 ± 0.11	0.749 ± 0.06
FCBI	23.106 ± 3.43	0.622 ± 0.11	0.729 ± 0.06
ICBI	23.400 ± 3.46	0.638 ± 0.11	0.743 ± 0.06
DFDF	23.179 ± 3.45	0.626 ± 0.11	0.728 ± 0.06
NEDI	22.131 ± 3.27	0.582 ± 0.12	0.708 ± 0.06
SpR	21.530 ± 3.20	0.579 ± 0.12	0.735 ± 0.05
SME	23.361 ± 3.46	0.636 ± 0.11	0.741 ± 0.06
KR	23.095 ± 3.45	0.619 ± 0.12	0.722 ± 0.06
Proposed	$\mathbf{24.131} \pm 3.48$	$\mathbf{0.668} \pm 0.10$	$\mathbf{0.775} \pm 0.05$

The elapsed times of each method are shown in Table 5.3. Nearest neighbor, bilinear, and bicubic are excluded because we used Matlab's built-in function. As observed, the proposed method required the least amount of time among all methods. The proposed method and FCBI required nearly the same amount of time, followed by ICBI, which is an extension of FCBI with an iterative function. SpR required the greatest time (approximately 215 seconds for 2x enlargement), and SME required approximately one-half the elapsed time of SpR. However, it should be noted that our proposed method has been optimized by using built in function of bicubic interpolation during the back-projection step.

Table 3.4: Average elapsed time for 2x magnification (seconds)

Methods	Elapsed time
FCBI	0.756 ± 0.03
ICBI	1.201 ± 0.19
DFDF	5.589 ± 0.34
NEDI	7.129 ± 0.45
SpR	215.040 ± 16.55
SME	102.180 ± 4.96
KR	6.442 ± 0.33
Proposed	0.705 ± 0.06

3.4.2 Qualitative analysis

Here, we present a qualitative evaluation of the results obtained by the proposed and previous methods. 2x and 4x magnification was used to clarify the results without blurring,

ringing, blocking artifacts, etc. Note that in the following figures, red arrows indicate clearly identifiable noise.

In Fig. 3.14 (2x magnification), with the exception of the proposed method, all images suffer from many types of artifacts. However, in 4x magnification, it is very difficult to distinguish the best quality image. Note that most algorithms suffer from some noise. We also compared the proposed method to FCBI, which is most closely related to the proposed method. In Fig. 3.15, the proposed method clearly demonstrates better texture and smoother edges with less blur.

However, there are some anomalies in the qualitative and quantitative experiments. For example bicubic, which ranked second in our quantitative analysis, suffers from significantly more artifacts, particularly blurring, than SpR, which demonstrated lower PSNR, SSIM, and FSIM values. From a qualitative perspective, SpR demonstrates very good results that are close to the quality of the proposed method. Other methods, such as ICBI and SME, also demonstrate very good results, yet suffer from some artifacts.

This analysis verifies that the proposed method can reduce common artifacts such as ringing, blurring, and blocking. It is also proven that the proposed method can successfully reconstruct image details.

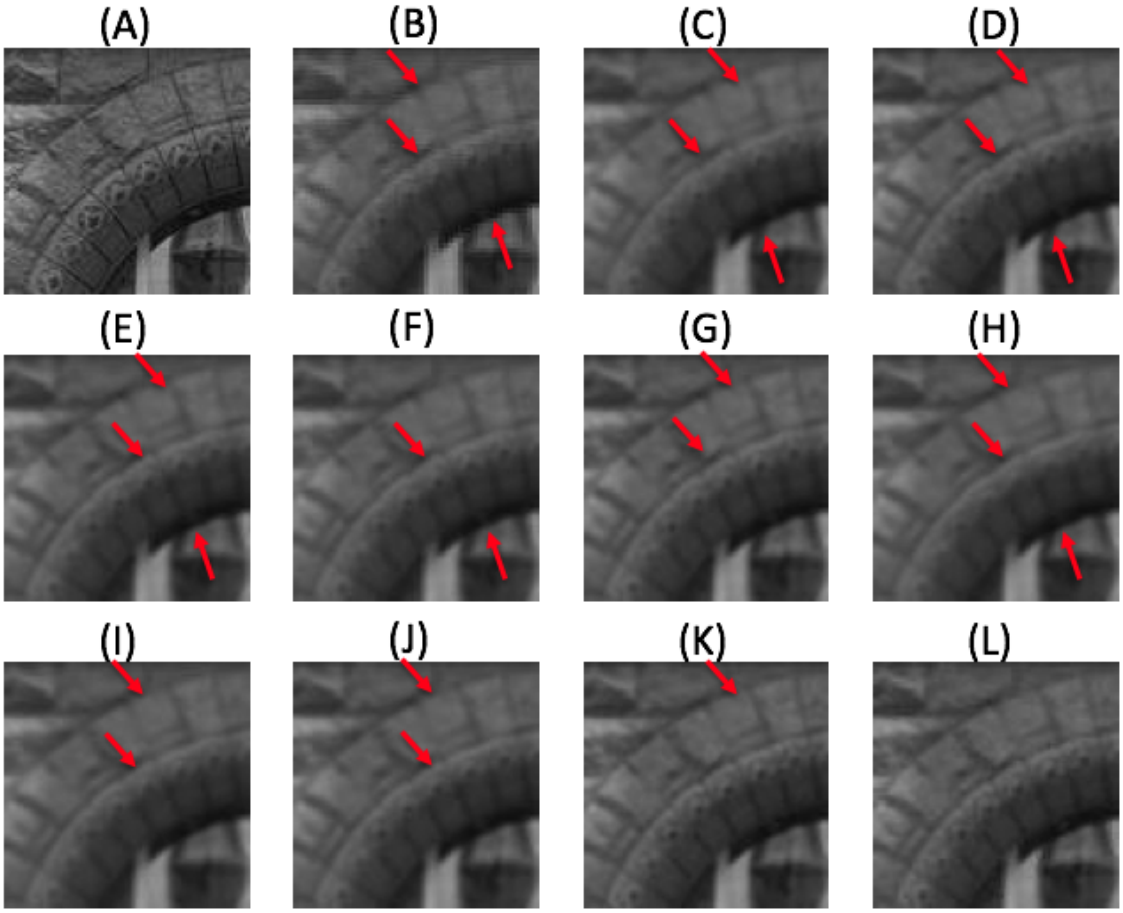


Figure 3.14: Experimental results for 2x magnification. (A) ground truth, (B) nearest neighbor, (C) bilinear, (D) bicubic, (E) DFDF, (F) FCBI, (G) ICBI, (H) KR, (I) NEDI, (J) SME, (K) SpR, and (L) the proposed method

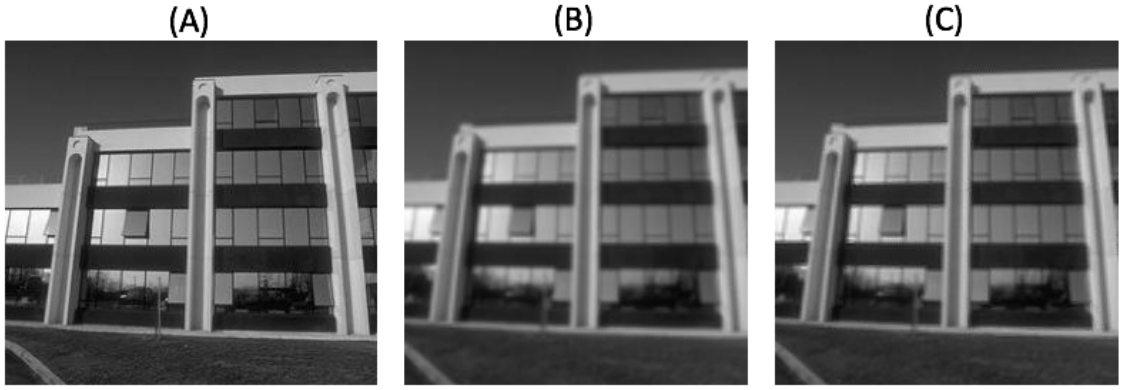


Figure 3.15: Comparison of FCBI and the proposed method for 2x magnification: (A) ground truth, (B) FCBI, and (C) the proposed method

Chapter 4

Super-Resolution via Adaptive Multiple Sparse Representation

4.1 Introduction

The use of unmanned aerial vehicles (UAVs) in agriculture has increased in recent years [32, 11, 15]. The use of UAVs offers alternatives to manual breeding methods in agriculture, which are laborious, time-consuming, unreliable, and often impossible to implement. For example, high-frequency time series data are almost impossible to obtain without the use of a UAV. Moreover, large-scale, hilly landscapes make it impractical to manually analyze each tree individually using hand-held or ground-based devices. The use of UAVs can overcome such limitations, and UAV imaging offers advantages in terms of high-resolution data and precise 3D imaging.

Table 4.1: Comparison of agricultural monitoring systems (◯ = superior, △ = average, × = poor).

Method	Hand-held device	Ground-based device	UAV	Aircraft	Satellite
Frequency	×	△	◯	△	×
Coverage	×	×	△	◯	◯
Cost	◯	△	△	×	×
User friendly	◯	△	◯	×	×
Resolution	◯	◯	△	△	×

Examples of some of the advantages offered by the use of UAVs over traditional field-based monitoring methods are listed in Table 4.1. UAV imaging can efficiently provide high-frequency time series data, whereas aircraft and satellite systems are very complicated and their use requires arrangements be made in advance. Hand-held and ground-based

devices have short preparation times but require long execution times. In terms of coverage, aircraft and satellites perform well because they can rapidly image several hectares in area, but they produce low-resolution images. By contrast, UAVs can provide better resolution as they have adjustable flight altitudes. Although hand-held and ground-based devices can provide the best resolution because they can observe parts of plants in detail, they cannot be used for large area and coverage or to produce high-frequency time series data. UAVs also require lower expenditures than aircraft or satellite as UAV sensors are much cheaper. As a UAV can be operated autonomously, control by the end user is much simpler. These advantages make UAV utilization in agricultural monitoring quite useful by offering a new perspective from which to monitor the ground with high precision [46].

The main problems in constructing 3D high-resolution maps using UAV images are flight-time limitations and image quality from the target object. Taking aerial images of a large field will consume a large amount of time, and to reduce time consumption, it is necessary to set an optimum height for UAV flight. However, maximizing the height, which increase the viewing perspective of the UAV and thus potentially reduces the flight time, reduces the optical detail of a target object. Therefore, it is necessary to use a super-resolution (SR) technique to obtain higher-resolution, high-precision images of target objects [5].

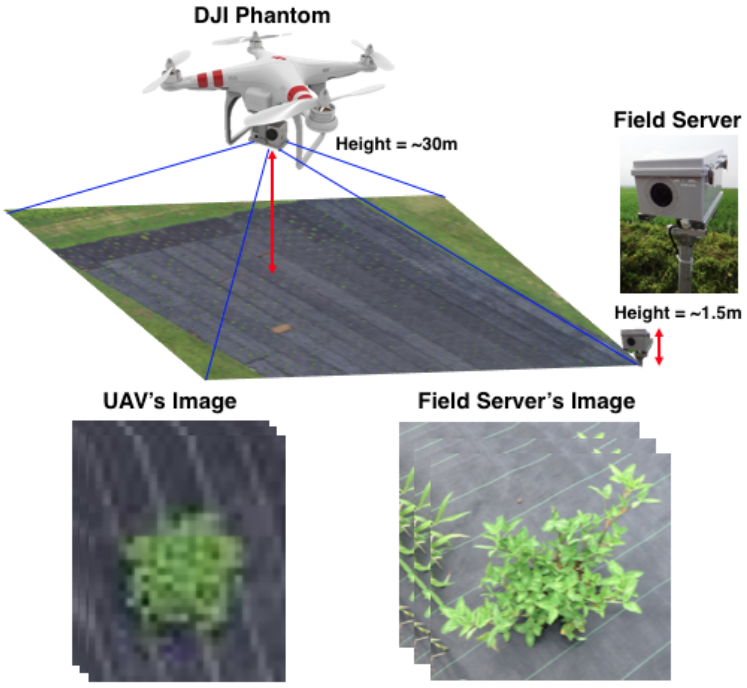


Figure 4.1: DJI Phantom and Field Server sample images.

Field Server (FS) systems [12, 16] can be used for ground-based monitoring via a series of small sensor nodes equipped with a Web server that can be accessed via the Internet

and communicate, unlike traditional sensor nodes, through a wireless LAN over a high-speed transmission network. An FS system can be easily installed for remotely monitoring field information anywhere. By including the functionality of a Web server in each module, an FS system can collectively manage each module over the Internet, producing high-resolution images that can be used as training images for an SR algorithm. A comparison of FS and UAV images is shown in Fig. 4.1.

Depending on the input image, SR imaging is primarily divided into two types, i.e., single- and multi-image SR imaging. Multi-image SR requires multiple images to acquire intrinsic characteristics; it combines the information from each image to construct a higher-resolution image. In day-to-day applications, however, it is unusual to obtain multiple images using a generic camera with known parameters. Single-image SR requires only a single image to construct a higher-resolution image - a much simpler task than multi-image SR. Single-image SR typically exploits the characteristics of the input image and uses prior knowledge to determine the relationship between a low- (LR) and high-resolution (HR) image. Our proposed method therefore uses single-image SR, which is highly suitable for the use real world applications. Furthermore, training based on SR can produce better prediction using a training model for enlarging images of phenotyping fields.

Owing to their low complexity and ease of implementation, classic nearest neighbor, bilinear, and bicubic interpolation methods have been widely applied in image processing [30]. However, such methods produce unnatural images due to excessive blurring and jagged artifacts [2].

Multi-image SR is highly suitable for video enlargement. It can exploit intrinsic characteristics that may differ from one sequence to another. Liu et al. [25] proposed a Bayesian approach to adaptive video SR that involved the simultaneous estimation of the underlying motion, blur kernel, and noise level to reconstruct original HR frames; however, this approach has high computational complexity.

Edge-direction-based algorithms, which are applied to single-image SR and often termed edge-adaptive algorithms, have been used to overcome computational complexity limitations by exploiting local features such as edges [14, 24, 17, 18]. For example, new edge directed interpolation (NEDI) [24] produces good imaging results by adapting each interpolating surface locally and assuming local regularity of curvature. Iterative curvature-based interpolation (ICBI), inspired by NEDI, produces interpolated pixels by averaging sets of two pixels using second-order directional derivatives of the image intensity [14].

SR using sparse representation has become popular because of its ability to naturally encode the semantic information of images [10]. By collecting representative samples in order to create an over-completed dictionary, it is possible to discover the correct basis for

correctly encoding an input image. The studies by Yang et al. [44] and Zeyde et al. [45] focused on using a single pair of dictionaries; intuitively, however, using a single pair of dictionaries can produce many redundancies, which may cause instability during the image reconstruction process.

In this paper, we propose adapting multiple pairs of dictionaries that classify by edge orientation in order to select the most suitable pair of dictionaries for a particular signal. These dictionaries are obtained by determining bases from HR images produced by FS. Following this, we discuss how input images from a UAV can be enlarged to obtain higher-resolution images. Finally, we demonstrate the effectiveness of the proposed method in reconstructing 3D images.

The paper is organized as follows. Section 4.2 presents an explanation of the current state-of-the-art of sparse-based SR. Section 4.3 explains the proposed edge orientation measurement-based algorithm, multiple dictionaries construction, and enlargement process. Section 4.4 discusses the results of our experiments and analysis. Finally, section 4.5 shows an application of the proposed method for 3D reconstruction.

4.2 Super-resolution Based On Sparse Representation

Sparse signal representation is widely used as a powerful tool for representing and compressing high-dimensional signals. The success of this method primarily depends on the ability to find a proper basis for naturally representing a signal as, for example, audio or images, and sparse representation can be used to naturally generate the semantic information of the input data. These advantages, however, make it challenging to effectively construct sparse systems, which differ from conventional systems for which it is usually assumed that sufficient and suitable properties have already been obtained.

Much research has confirmed the strength of sparsity as a powerful visual representation tool [45, 44]. Sparse representation naturally chooses the most relevant patch bases in a dictionary to best represent a patch for an LR input image. There are two constraints to solving ill-posed SR problems proposed in this system: (1) the reconstruction constraint requires forcing the recovered input X to be consistent with the input Y; (2) the prior sparsity constraint requires that every patch from an image can be represented as a sparse linear combination in the dictionary.

Let X be an HR image recovered from an input LR image Y. In eq.(4.1) below, the patch x of the HR image X is represented as a sparse linear combination in a dictionary D_h of high-resolution patches sampled from training images.

$$x \approx D_h\alpha \text{ for some } \alpha \in \mathbb{R}^K \text{ with } \|\alpha\| \ll K \tag{4.1}$$

The sparse representation α is recovered by representing the patches y of the input image $\boldsymbol{Y}$ with respect to an LR dictionary D_l trained with D_h.

Yang et al. [44] proposed an algorithm that attempts to infer the HR image patch for each input LR image patch. For this system, they developed two dictionaries, D_h and D_l, which are trained to have the same sparse representations. In order that each patch can be represented as a texture rather than an absolute intensity, a mean value is obtained from each patch. Finally, in the recovery process the mean value for each high-resolution patch is predicted from its LR patch.

For each input LR patch y, this algorithm obtains a sparse representation from D_l; and the corresponding HR patch bases D_h are then combined according to the coefficients in D_l to generate an output HR patch x. The problem of finding a sparse representation of y can be defined as in eq. (4.2) below:

$$min\|\alpha\|_0 \; s.t \; \|FD_l\alpha - Fy\|_2^2 \leq \varepsilon \tag{4.2}$$

where F is the feature extraction operator, which plays primary role as a perceptually meaningful constraint to presenting the relation between α and y.

4.2.1 Single Pair Dictionary Limitation

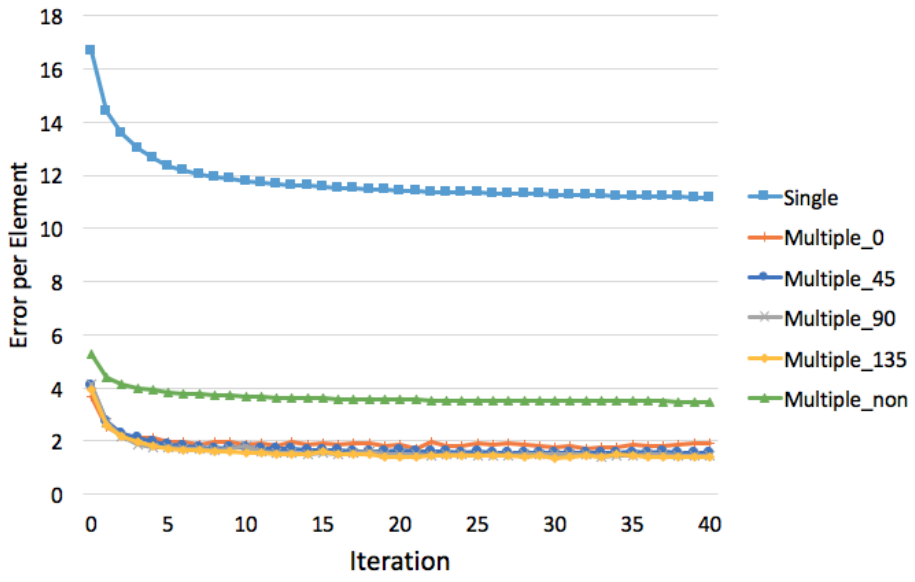


Figure 4.2: Error produced in K-SVD dictionary learning for single and multiple pair dictionaries with 1024 atoms. Single pair dictionary error, labeled as "Single", produces higher error than multiple pair dictionaries that classify into five classes based on edge orientation (0, 45, 90, 135, and non-directional).

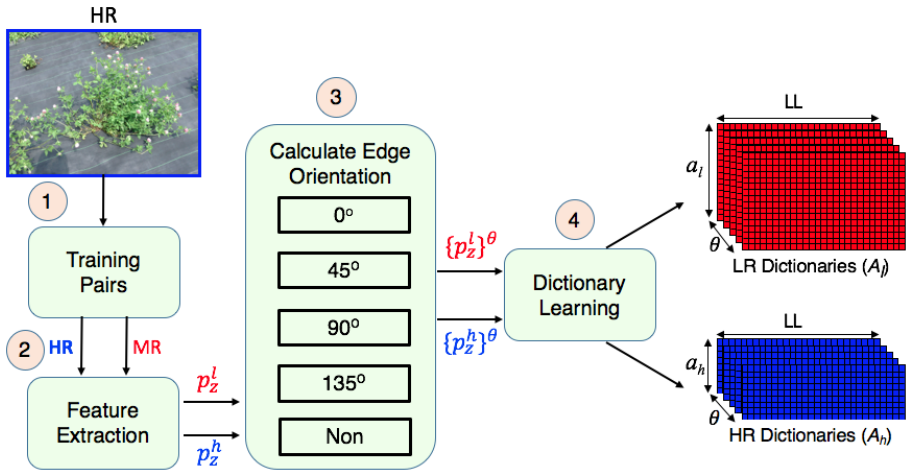


Figure 4.3: Process of dictionary construction.

The studies conducted by Yang et al. [44] and Zeyde et al. [45] focused on constructing single pairs in the sparse dictionary. However, as the training patch is not categorized into specific categories, it can produce many redundancies that lead to instability during the sparse coding process. We found that selectively choosing the training patches and then categorizing them into particular classes could reduce the error produced during the sparse coding process.

A comparison of single and multiple pair dictionaries from the K-SVD dictionary learning algorithm is shown in Fig. 4.2. The experiment used 40 iterations to calculate the error per element for each iteration. The figure indicates that the single pair dictionary produces a higher error rate per element than multiple pair dictionaries that have subtle errors for each class.

4.3 Proposed Method

In this section, the core algorithm is explained. The section is divided into two subsections: multiple dictionary construction (training step), and SR algorithm (testing step).

4.3.1 Multiple Dictionary Construction

In the training step, the proposed method constructs multiple pairs of dictionaries that categorize by edge orientation. This step will produce five pairs of dictionaries that will be used in the sparse coding step to form an HR image. The complete steps of the dictionary construction process are explained in Algorithm 2.

Algorithm 2: The proposed multiple pair dictionary construction.

Input: HR images set as training images.
Output: Multiple pairs of dictionaries A_h and A_l.

1 Create LR images by blurring and downsampling HR images
2 Upsample each LR image to create MR images
3 Apply feature extraction filters to each MR image and obtain high-frequency elements from HR images
4 Estimate the edge orientation from each HR image
5 Divide each HR and MR feature into patches then reshape each into one pair of vectors
6 Gather and cluster the vectors into 5 classes based on edge orientation
7 Combine the vectors into an array of multiple class MR patches ($\{p_z^l\}^0$, $\{p_z^l\}^{45}$, $\{p_z^l\}^{90}$, $\{p_z^l\}^{135}$, $\{p_z^l\}^{non}$) and HR patches ($\{p_z^h\}^0$, $\{p_z^h\}^{45}$, $\{p_z^h\}^{90}$, $\{p_z^h\}^{135}$, $\{p_z^h\}^{non}$)
8 For each cluster, learn a pair of coupled dictionaries
9 **return** $X^\star$

Dictionary construction is divided into four steps: training pair collection; feature extraction for each patch; categorizing the set of features into five clusters based on edge orientation; and finally, dictionary construction. For each cluster, we construct HR and LR dictionaries using the learning algorithm K-SVD. A brief outline of this process is illustrated in Fig. 4.3.

4.3.1.1 Training pairs collection (step 1)

The HR image is formulated with height M and width N, where the total number of pixels is $M \times N$. Let $X_H = \{x_m : m = 0,1,2,\cdots,M-1\}$ and $Y_H = \{y_n : n = 0,1,2,\cdots,N-1\}$ be finite sets that determine the number of pixels. The HR image is defined as the function $f_H : X_H \times Y_H \rightarrow I$ where $I = \{0,1,2,\cdots,255\}$ is the value of each pixel. Then, downsampled grid $(M^{'} \times N^{'})$ applies the following condition, $M^{'} < M$ and $N^{'} < N$. Let $X_L = \{x_m : m = 0,1,2,\cdots,M^{'}-1\}$ and $Y_L = \{y_n : n = 0,1,2,\cdots,N^{'}-1\}$ be finite sets that determine the number of pixels in the LR image. The LR image is defined as the function $f_L : X_L \times Y_L \rightarrow I$.

Using a set of HR images $I_h \in f_H$ from a field server or a hand-held camera as the input for the dictionary construction process, an LR image $I_l \in f_L$ is constructed by blurring and downsampling each HR image. Middle resolution (MR) image $I_m \in f_H$ is then obtained by up-sampling this LR image and have the same size with the HR images. This condition simplifies the process of extracting training pairs for the dictionary learning since both training images (HR and MR) have the same indexes. The algorithm starts by gathering pairs of HR and MR images, as illustrated in Fig. 4.4.

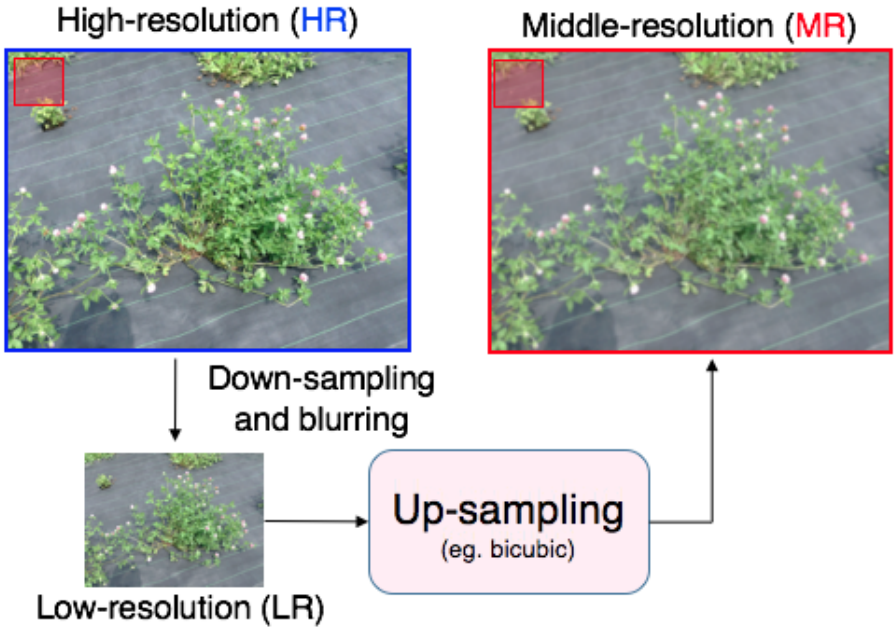


Figure 4.4: Training pairs (step 1).

4.3.1.2 Feature extraction (step 2)

The feature extraction process is intended to produce more informative and non-redundant data for large sets of resources. Features are extracted directly onto full images rather than small image patches in order to avoid boundary problems. Then, features based on the image patch indices are obtained from the patches.

In this step, features from the HR and MR images obtained from previous step are extracted. The HR features consist of high-frequency components which collected by subtracting the MR images from the HR images, and the LR features consist of first- and second-order derivative components collected by applying four types of filter and then convoluting to MR images.

The HR feature is obtained by computing the difference images $I_d(x_m, y_n) = I_h(x_m, y_n) - I_m(x_m, y_n)$ where $(x_m, y_n) \in X_H \times Y_H$ that is, $I_d \in f_H$. This step serves to establish the relation of HR and MR images to edges and textures. The LR features are obtained by convoluting to k-filter d_k where $d_k * I_m \in f_H$ for $k \in \{1,2,3,4\}$. The filter function is expressed in the eq. (4.3).

$$d_1 = [1,-1]; d_2 = d_1^T; d_3 = [1,-2,1]; d_4 = d_3^T; \tag{4.3}$$

After the step described above, the HR feature patches p^h are extracted from the HR features I_d and then the LR feature patches p^l are obtained from each corresponding filter

of the LR features. By using patch size $s \times s$ where $s = 9$, all features are merged into one vector with total length $p^l = 324$, and $p^h = 81$, so that $p^h \in \mathbb{R}^{81}$ and $p^l \in \mathbb{R}^{324}$.

The variable s can be changed depend on the zoom factor. In our experiment, we use $s = 9$ because we enlarge the input image by 3. Other reason is computational time. Higher s might produce better result but surely produce longer computational time. While, lower s (e.g. $s = 3$) cannot represent the information of HR well but have shorter computational time. Therefore, we need to optimize the value of s and based on our experiment, $s = 9$ is the optimum value.

Let $Z = \lfloor M/s \rfloor \times \lfloor N/s \rfloor$ be the total number of patches. The floor function is defined as $\lfloor x \rfloor = max\{r \in \mathbb{Z} | r \leq x\}$, where x is a real number and $\{\mathbb{Z}, r\}$ are sets of positive, negative, and zero integers. Let $q \in \{0, s \times 1, s \times 2, ..., \lfloor M/s \rfloor\}$ and $r \in \{0, s \times 1, s \times 2, ..., \lfloor N/s \rfloor\}$ be the coordinates to obtain the patch by size s. The HR patch $p_z^h \in \mathbb{R}^{81 \times Z}$ can be expressed in eq. (4.4).

$$\begin{aligned} p_z^h =& [p^h(0,0)\ p^h(0, s \times 1)\ \cdots\ p^h(\lfloor M/s \rfloor, \lfloor N/s \rfloor)] \\ & \text{where} \\ p^h(q,r) =& [I_d(q,r)\ I_d(q,r+1)\ \cdots\ I_d(q,r+s-1) \\ & \cdots I_d(q+1,r+s-1)\ \cdots\ I_d(q+s-1,r+s-1)]^T \end{aligned} \tag{4.4}$$

The MR patch $p_z^l \in \mathbb{R}^{324 \times Z}$ can be expressed in eq. (4.5) below where $\bigoplus$ is the operator used to concatenate each LR feature into one vector:

$$\begin{aligned} p_z^l =& [p^l(0,0)\ p^l(0, s \times 1)\ \cdots\ p^l(\lfloor M/s \rfloor, \lfloor N/s \rfloor)] \\ & \text{where} \\ p^l(q,r) =& [\bigoplus_{k=1}^{4} d_k * [I_m(q,r)\ I_m(q,r+1)\ \cdots\ I_m(q,r+s-1) \\ & \cdots I_m(q+1,r+s-1)\ \cdots\ I_m(q+s-1,r+s-1)]]^T \end{aligned} \tag{4.5}$$

4.3.1.3 Edge orientation measurement (step 3)

As shown in Fig. 4.5, five edge orientations are defined to classify the features of both patches. There are four directional edges and one non-directional edge, with the four directional edges including vertical, horizontal, 45-degree, and 135-degree diagonal edges. These directional edges are extracted from the 9 × 9 image-blocks; if an image-block contains an arbitrary edge without directionality, it is classified as a non-directional edge (shown as the black area at Fig. 4.6).

Fig. 4.6 details the steps used to obtain the edge orientation. First, the edge image is obtained from an HR image using canny edge detection [3]. The edge image is then used to

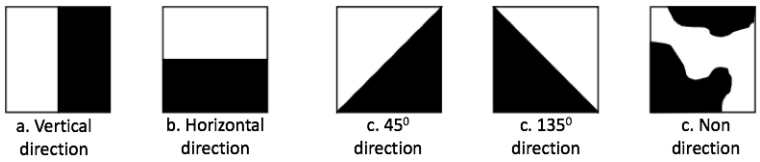


Figure 4.5: Five types of edge orientation

calculate the edge orientation as follows. For each patch of size 9×9, the gradient, which is a scalar that specifies the angle between the x-axis and the major axis of the ellipse that has the same second-moments as the region, is calculated. If θ is the angle of a particular edge and ranges in value from -90 to 90 degrees, including null value, then the image can be calibrated into 5 classes of pair patches (MR patches $\{p_z^l\}^0$, $\{p_z^l\}^{45}$, $\{p_z^l\}^{90}$, $\{p_z^l\}^{135}$, $\{p_z^l\}^{non}$ and HR patches $\{p_z^h\}^0$, $\{p_z^h\}^{45}$, $\{p_z^h\}^{90}$, $\{p_z^h\}^{135}$, $\{p_z^h\}^{non}$). The function $C(\theta)$ is used to classify the edge as follows:

$$C(\theta) = \begin{cases} 0 \text{ ; if}(-67.5 \leq \theta < 22.5) \\ 45 \text{ ; if}(22.5 \leq \theta < 67.5) \\ 90 \text{ ; if}(-22.5 < \theta \leq 67.5) \\ 135 \text{ ; if}(-67.5 < \theta \leq -22.5) \\ non \text{ ; if}(\theta \text{ is null}) \end{cases} \tag{4.6}$$

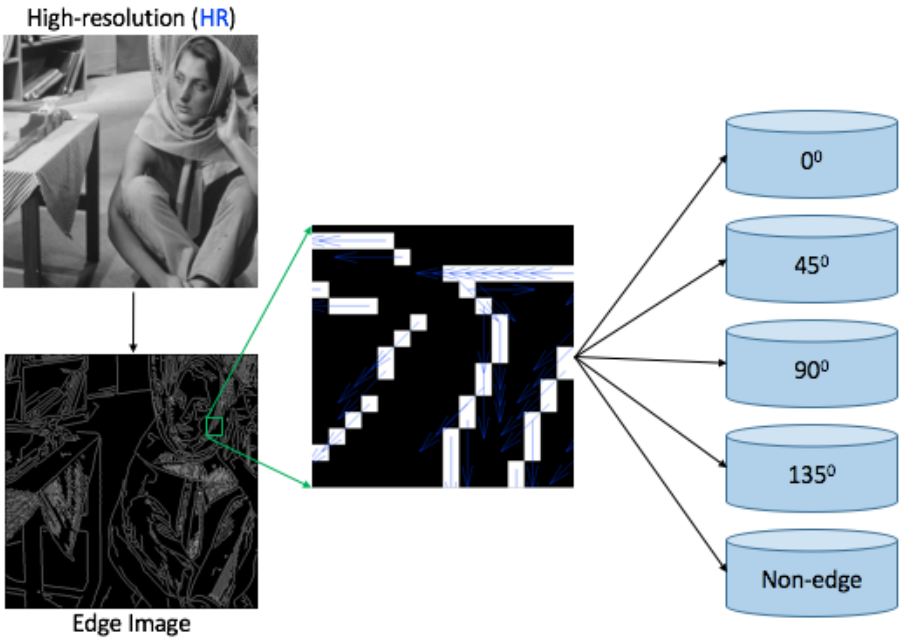


Figure 4.6: Process of edge orientation calculation. The blue arrow in the edge image shows the edge orientation of a particular patch (step 3).

Fig. 4.7 shows the average number of elements for each edge cluster for two set of images. Each set is consist of 10 agricultural images (image A) and 10 natural images (image B). It is seen from the figure that image A mostly contains diagonal edges, while the types of component in image B are distributed almost normally. Thus, natural images have different characteristic to the agriculture images. We can therefore classify each patch into a group, which can help reduce inconsistency during the sparse coding process.

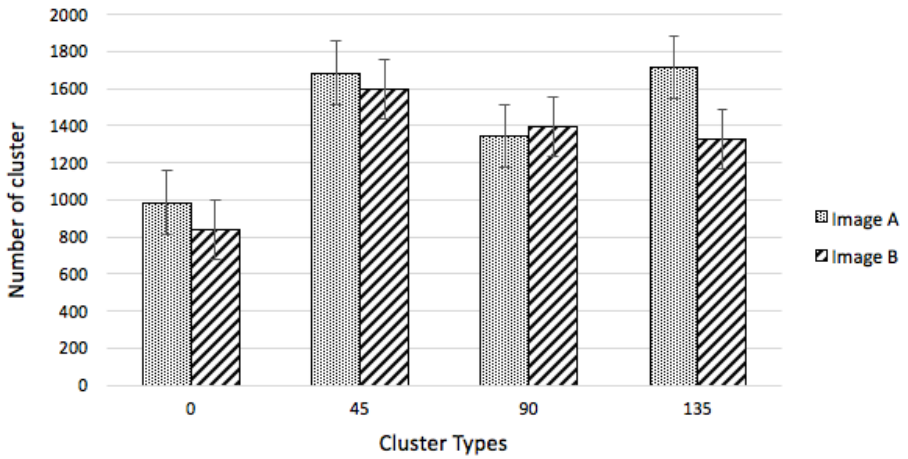


Figure 4.7: Edge distribution based on orientation. The y-axis gives the number of members, while the x-axis gives the cluster types. Image A is an agricultural image, and image B is a natural image.

4.3.1.4 Dictionary learning (step 4)

In this step, we construct HR and LR dictionaries (A_h and A_l) for each edge cluster where $a_h = |p^h|$, $a_l = |p^l|$, and LL is the dictionary number of atoms, so that $A_h \in \mathbb{R}^{a_h \times LL}$ and $A_l \in \mathbb{R}^{a_l \times LL}$. Starting by constructing A_l, we use LR features p_z^l and apply a dictionary learning procedure using OMP [38] and K-SVD [1]. In addition to A_l, this process also produces a sparse representation vector q_z that corresponds to p_z^l. This process is expressed in eq. (4.7) as follows:

$$Al, \{q_z\} = argmin_{Al,\{q_z\}} \sum_z \|p_z^l - A_l q_z\|_2 \tag{4.7}$$

After obtaining A_l, we proceed to the construction of the HR dictionary A_h. We multiply q_z, from the preceding equation, by A_h, as shown in eq. (4.8):

$$A_h = argmin_{A_h} \sum_z \|p_z^h - A_h q_z\|_2^2 \tag{4.8}$$

4.3.2 Super-resolution algorithm

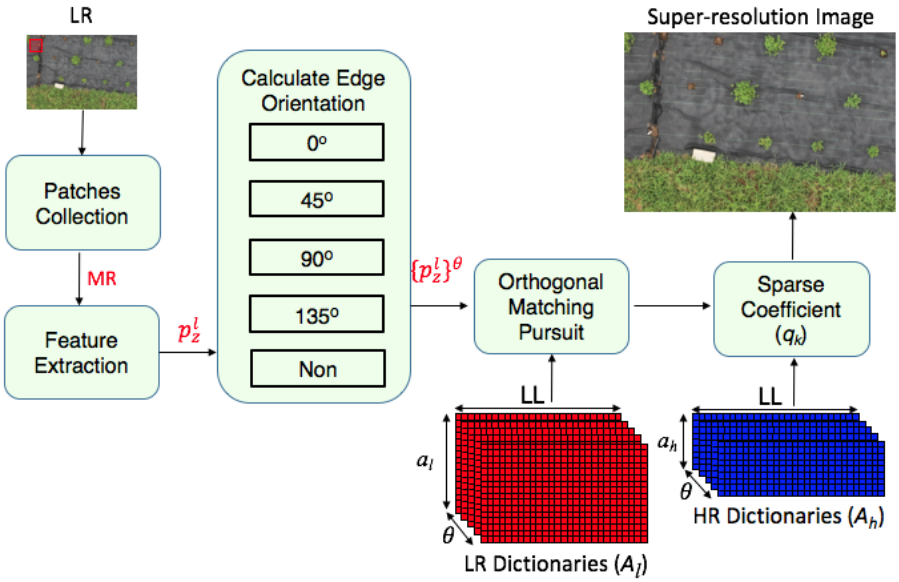


Figure 4.8: The proposed super-resolution algorithm.

The reconstruction process starts by upsampling an LR image into an MR image using a conventional interpolation e.g., bicubic. After the features are extracted, each patch of LR features p_z^l is reshaped into a one-row vector as is done in dictionary construction in the training step. After the edge orientation is calculated, the LR features p_z^l are classified based on their edge orientation $\{p_z^l\}^\theta$. Using the corresponding dictionaries obtained from the learning steps, the sparse coding coefficients q_k of each LR feature over the LR dictionary are calculated. Finally, an HR patch is obtained by multiplying the cluster HR dictionary by the sparse coding coefficients obtained from the previous step. The proposed algorithm is briefly outlined in Fig. 4.8.

4.4 Experimental Results

To confirm the efficiency of the proposed method, we conducted several experiments. The analysis of these experiments is divided into two subsections: quantitative and qualitative analyses. All experiments were conducted using Matlab R2012b on Win 8.1 64-bit (Intel Core i7@3.2GHz, 8GB). The images used in the experiment were taken at Kazusa DNA Research Institute, Chiba, Japan, red clover tree phenotyping field.

The image dataset consisted of two sub-datasets: training and testing. The training dataset was obtained using a hand-held camera (size: 2592×1936). The testing dataset was taken using DJI Phantom 2 Vision (resolution: 14 Megapixels; sensor size: 1/2.3”;

FOV: $120^o/110^o/85^o$) with an original size of 5472×3648 pixels. Noted that the image produced by DJI Phantom 2 Vision might have been enhanced before delivered to the users. To simplify the process, we divided the image into 256×256 pixel sub-images. In total, we used 5 training images and 300 testing images, as shown in Fig. 4.9 and Fig. 4.10.

In Fig. 4.11, we illustrate the transformation from original color image into YCbCr color channel. We only enhanced the brightness components (Y) while enhancing the other components using bicubic interpolation because human vision is more sensitive to brightness change. Then, each resulting image channel was combined to produce a final color image. This procedure will be very effective because it can speed up the computational process of the proposed method.

Figure 4.9: Samples of training images taken by hand-held digital camera.

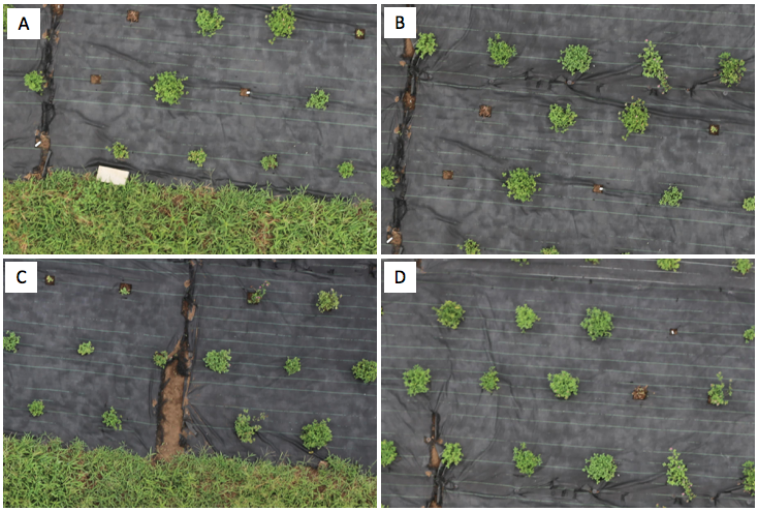


Figure 4.10: Images A-D show sample testing images taken by UAV (DJI Phantom 2 Vision).

In the experiments, we obtained images by downsampling and blurring the original images and then enlarging using different methods to $3\times$ magnification. We compared the effectiveness of seven methods: nearest neighbor, bilinear, bicubic, Yang et al. [44], Kim

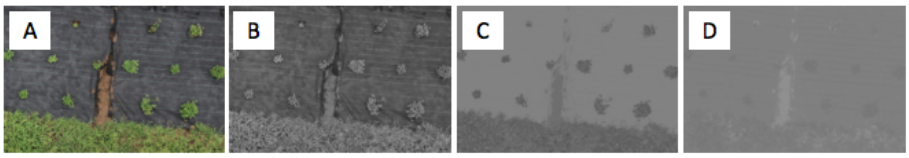


Figure 4.11: YCbCr color components. A) Original color image, B) Y component, C) Cb component, D) Cr component.

et al. [23], Zeyde et al. [45], and the proposed method. The algorithms associated with these vary in nature; therefore, in order to produce objective comparisons, all parameters used in the training and testing had to be similar. However, no specific parameter needed to be used for the conventional interpolation methods.

Our proposed method uses 3×3 patches with no overlapping pixels and five pairs of dictionaries. The algorithm of Yang et al. [44] uses 5×5 patches with a 4-pixels patch overlap and a single pair of dictionaries with 1024 atoms with back-projection. The algorithm of Zeyde et al. [45] uses 3×3 patches with 2-pixels patch overlap and a single pair of dictionaries with 1000 atoms. As mentioned above, these algorithms have different characteristics, and therefore obtaining objective comparisons required that all parameters used in training and testing were similar to those recommended in the respective literature. However, our proposed method is not sensitive to number of training images. In our experience, five training images with size 2592×1936 is enough to obtain the best result.

4.4.1 Quantitative Analysis

Methods for measuring the peak signal-to-noise ration (PSNR) [21], structural similarity (SSIM) [41], feature similarity (FSIM) [48], and elapsed time were used for quantitative measurement. The PSNR in decibels (dB) between the original image and the upscaled image is given by [21]. SSIM is a method that measures the quality of images based on the structural content of the original and magnified images. FSIM is based on the fact that the human visual system processes an image mainly in terms of its low-level features. Two features are considered in FSIM computation: the primary feature, i.e., phase congruency (PC), which is a dimensionless measure of a local structure's significance; and the secondary feature, i.e., the image gradient magnitude. FSIM combines both of these features to characterize the local quality of an image. Higher values of PSNR, SSIM, and FSIM indicate better quality. CPU time was computed using Matlab functions (tic and toc) to measure the elapsed time for a certain process. All measurements used only the luminance channel (Y) to simplify and objectively calculate the error.

Table 4.2: Comparison of the average quantitative results produced by PSNR, SSIM, and FSIM for 3× magnification (bold font indicates the best values).

Methods	PSNR	SSIM	FSIM	Time
Nearest neighbor	22.762 ± 3.85	0.637 ± 0.12	0.736 ± 0.06	-
Bilinear	23.243 ± 3.91	0.650 ± 0.12	0.767 ± 0.06	-
Bicubic	23.361 ± 3.93	0.663 ± 0.12	0.779 ± 0.06	-
Kim et al. [23]	23.205 ± 3.93	0.674 ± 0.11	0.789 ± 0.06	5.568 ± 1.83
Yang et al. [44]	23.213 ± 3.93	0.673 ± 0.11	0.795 ± 0.05	67.189 ± 4.78
Zeyde et al. [45]	23.328 ± 3.93	0.677 ± 0.11	0.794 ± 0.05	$\mathbf{0.669} \pm 0.04$
Proposed	$\mathbf{25.847} \pm 4.35$	$\mathbf{0.768} \pm 0.09$	$\mathbf{0.845} \pm 0.05$	6.290 ± 1.15

Table 4.2 lists the average values from four measurements, with the best values shown in bold. These result confirm that our proposed method clearly out-performs other methods in terms of PSNR, SSIM, and FSIM. Our method obtains a PSNR value of 25.847 dB, which is at least 11% higher than the other methods. Our proposed method also obtains an SSIM value higher by at least 14% than the other methods. In terms of FSIM, our methods outperforms the others by at least 6%. However, it should be noted that PSNR is not suitable for measuring the quality of bicubic and bilinear, as the quantitative and qualitative analysis for both methods produced some anomalies.

Although our proposed method does not provide the lowest computational time, it is still far better in this respect than Yang et al.'s algorithm [44]. Zeyde et al. [45] produced the lowest computational time in our experiments, while our method competes competitively with Kim et al. [23] with a less than 1 s differential. Moreover, in future applications and research, the use of a graphics processing unit (GPU) application should offer the opportunity to decrease the computational time of the proposed method.

Nearest neighbor, bilinear, and bicubic were all excluded from the time evaluation as these had salient differences in nature to the proposed and other methods; these conventional methods are simple interpolators that do not use prior information or any learning processes. Moreover, their implementations use Matlab built-in functions, making the comparison unfair as these implement the optimization process automatically.

In different scenes, such as breeding of broad-acre cereals, the edge orientation will be ultimately diverse. However, since we classify the features into 5 groups, our proposed method will have smaller error than other methods. It can reduce the redundancy that lead to instability during image reconstruction process. Yet, we also agree that it will need some modification towards uniform scenes. First, we need to have training images that contains uniform scenes. Second, add new image construction's constraint. Third, use features

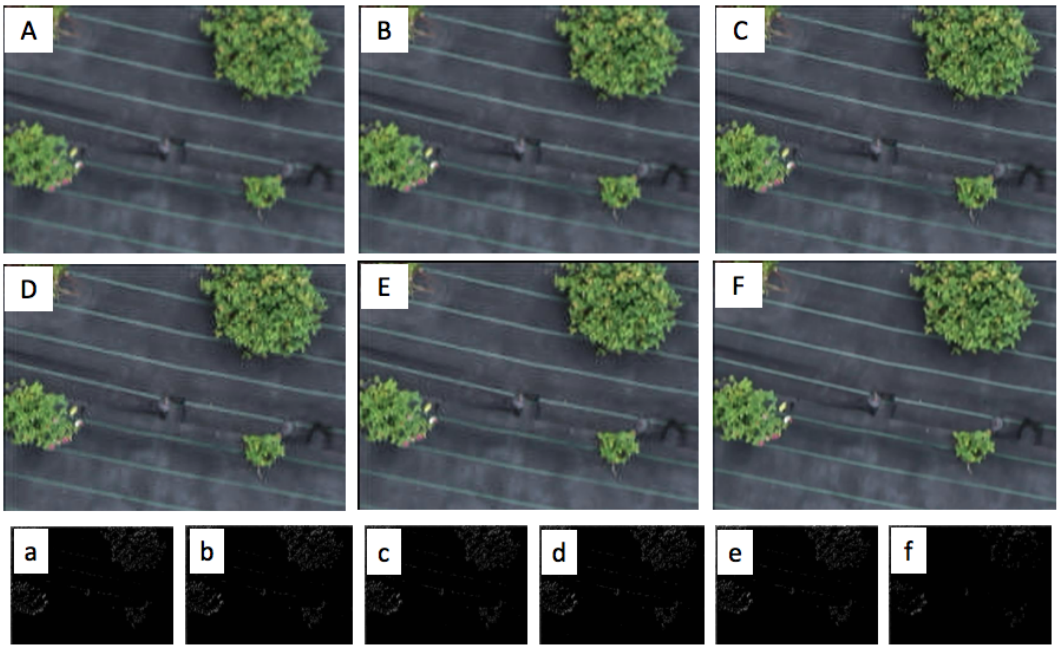


Figure 4.12: Results of experiment for 3× magnification (uppercase for color image, lowercase for difference image): A-a) Bilinear, B-b) Bicubic, C-c) Kim et al. [23], D-d) Yang et al. [44], E-e) Zeyde et al. [45], F-f) The proposed method.

that have better rotation properties, such as Zernike moments. More detail investigation is needed since the modification should be based on the characteristic of the image.

4.4.2 Qualitative Analysis

To evaluate the proposed method in terms of visual results, we conducted experiments using 3× magnified images to compare the proposed method to the other five methods: bilinear, bicubic, Kim et al. [23], Yang et al. [44], and Zeyde et al. [45].

Fig. 4.12 shows the differences between the original images and the results produces by the respective methods. Our method clearly produces sharper and smoother edges and is able to clearly construct the details of a scene. The other methods all produced images with some artifacts, especially in the line and tree areas, while bicubic and bilinear also produced blurring effects in the enlarged image. Although Yang's and Zeyde's methods generate sharp edge, they still suffer from some noise and produce undesired smoothing. By contrast, Kim's method produces too strong of an edge with unrealistic result. Moreover, it is seen that our proposed method has the least amount of difference from the original image, which means that the proposed method has produces the least amount of artifacts, as it can clearly reconstruct edges better than the other algorithms.

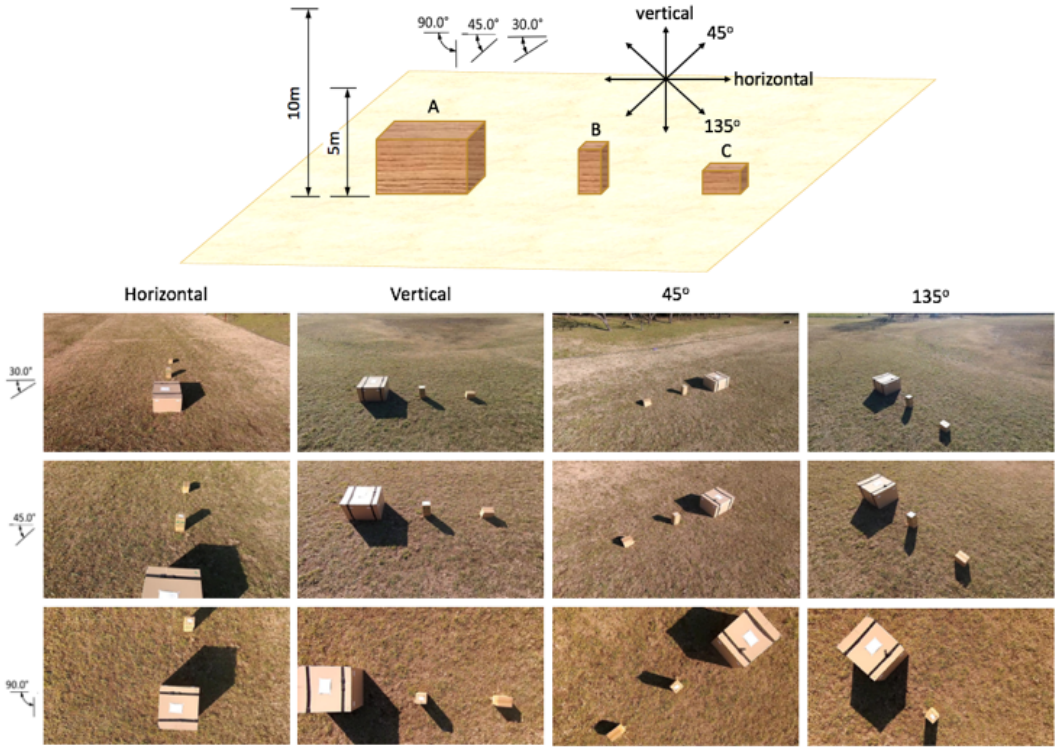


Figure 4.13: Flight experimental procedure and sample of image taken at 5m.

4.5 Application to 3D Reconstruction

High-resolution imaging is necessary in the construction of high-precision 3D images; correspondingly, the resolution of the input image affects the quality of the 3D reconstruction precision. In this section, we describe an application of our proposed method as a preprocessing step in 3D reconstruction and then compare it with the other methods.

A DJI Phantom 2 Vision was used to take aerial images of boxes in a field oriented at differing angles, directions, and heights with 25% end-lap for each direction. However, since the flight direction is divided into 5 directions (which are horizontal, 45, vertical, 135) towards the main object, it is very hard to calculate the side-lap percentage because each image will have different percentage. Before collecting the images, we created a flight plan that considered altitude, latitude, longitude, the distance of each turning point, and flight speed. The UAV periodically collected images from different angles in order to create 3D images. Only one operator was required to oversee the autonomous flight because the equipment was configured and the UAV can run in fully autonomous mode using defined parameters (e.g., sensors and flight plan). The flight procedure and sample of images are shown in Fig. 4.13.

Next, we implemented the Structure from Motion (SfM) algorithm developed by the

Table 4.3: Results of matching points between original images and particular methods (bold font indicates the best values).

	Key points			Matching points		
Methods	30^o	45^o	90^o	30^o	45^o	90^o
Bilinear	11866	20906	20039	488 (4.1%)	976 (4.6%)	727 (3.6%)
Bicubic	15984	22222	23259	598 (3.7%)	1213 (5.4%)	844 (3.6%)
Proposed method	**27935**	**32053**	**29409**	**1812 (6.5%)**	**2708 (8.4%)**	**2068 (7%)**

Table 4.4: 3D measurement results on 5m. The measurement is determined by picking six pairs of random points (XYZ) in each corner area of the boxes then calculating the average distances and error is the difference between real and observed measurement. (A* is used as reference and bold font indicate as the best value.)

		Height (cm)		Width (cm)		
Methods	$A*$	B	C	A	B	C
Real scale	61	47	25	101	24	32
Original	61	**45.25**	**22.76**	106.34	**22.94**	**32.27**
		(-1.75)	**(-2.24)**	(+5.34)	**(-1.06)**	**(+0.27)**
Proposed	61	42.83	20.45	**105.82**	20.34	30.26
		(-4.17)	(-4.55)	**(+4.82)**	(-3.66)	(-1.74)
Bicubic	61	41.03	16.94	107.54	20.25	25.98
		(-5.97)	(-8.06)	(+6.54)	(-3.75)	(-6.02)
Bilinear	61	38.93	17.32	–	–	–
		(-8.07)	(-7.68)			

authors of [13] on a PC (Windows 8.1, 64-bit; CPU: Intel® Core™ i7- 4790, RAM: 32 GB, GPU: GeForce TX780). SfM is the converse problem of estimating the locations of 3D points from multiple images given only a sparse set of correspondences between image features. This process often involves simultaneously estimating both 3D geometry (structure) and camera pose (motion) [35].

Table 4.3 lists the result produces by particular methods from matching points with original images. Using a SIFT algorithm [26], we extracted the feature points from each image and aligned the matching points. The results show that our proposed method produced the highest number of matching points of all of the methods.

We measured the height and width of each box and then calculated the error by comparing these to the real scale with the best result indicated by the lowest error value. First, we align our point cloud data with xy-axis by calculating the transformation matrix using Helmert Transformation [42]. We use 4 pairs of reference points from our mesh (source coordinates) and the destination position after the transformation (destination coordinates).

Table 4.5: 3D measurement results on 10m. The measurement is determined by picking six pairs of random points (XYZ) in each corner area of the boxes then calculating the average distances and error is the difference between real and observed measurement. (A* is used as reference and bold font indicate as the best value.)

		Height (cm)		Width (cm)		
Methods	$A*$	B	C	A	B	C
Real scale	61	47	25	101	24	32
Original	61	40.71 (-6.29)	20.68 (-4.32)	107.69 (+6.69)	16.73 (-7.27)	26.37 (-5.63)
Proposed	61	**45.78 (-1.22)**	**28.13 (+3.13)**	**100.37 (-0.63)**	**17.88 (-6.12)**	**28.84 (-3.16)**
Bicubic	61	40.95 (-6.05)	17.98 (-7.02)	89.73 (-11.27)	12.09 (-11.91)	15.01 (-16.99)
Bilinear	61	35.18 (-11.82)	–	115.95 (+14.95)	15.06 (-8.94)	–

This condition can occur since we know the real coordinates of our objects. We used an application in this link (`http://helmparms3d.sourceforge.net`) to apply Helmert Transformation. Second, we apply the transformation using CloudCompare. Then, we obtain the final point cloud data. Last, the height of box A was used as the scale reference to determine the height and width of other boxes. The observed measurements were calculated by randomly taking six pairs of points (XYZ) in each corner area of the boxes and determining the average distances of each pair using the Euclidian distance. The measurement results are listed in Table 4.4 and 4.5. In the case where the width of box A is 10m, it is seen that our proposed method can decrease the measurement error to a millimeter order of magnitude, while other methods have at least an approximate 11cm error. Some results for the bilinear method could not be calculated owing to bad reconstruction results.

In the case where the imaging was performed from a height of 5m, the original image has the highest precision, even better than the proposed method. However, the proposed method can still keep its measurement error lower than the other methods, and it has the least error in measuring the width of box A.

In the case where imaging occurred from 10m, we found that the proposed method produced an error even lower than that of the original image - a striking result. The greater height of the UAV meant that images with lower detail, or lower amount of pixels per centimeter (PPCM), were produced. In this case, an image taken from 10m has around 1 PPCM, while one taken from 5m has around 2 PPCM. Based on this, we know that the images from 10m suffered at least twice the noise of the 5m image, and the results prove

Table 4.6: Mean of C2C distance between original and particular methods (bold font indicates the best values).

	Height	
Methods	$5m$	$10m$
Bilinear	0.195012 ± 0.204566	0.092167 ± 0.047541
Bicubic	0.123733 ± 0.125171	0.092898 ± 0.050812
Proposed method	$\mathbf{0.077964} \pm 0.069467$	$\mathbf{0.053951} \pm 0.028842$

that our proposed method is able to recover test images, reinsert high-frequency details, and repair some of the inconsistency in edges owing to a lowered PPCM.

Finally, we measured cloud to cloud (C2C) distance between the original image and particular methods including the proposed method shown in Table 4.6. The measurement is implemented using Open Source Software - CloudCompare (3D point cloud and mesh processing). Our proposed method has the closest distance compare to other methods which means it is the most similar to the original image. Detail comparison can be seen in Fig. 4.14 and Fig. 4.15. Bilinear and bicubic suffer from bad contours. Meanwhile, our proposed method can produce excellent contours which mostly the same with original image.

Bigger, well-shaped objects are easy to reconstruct. In this experiment, we used boxes, not trees, to simplify the experiment. However, in the future we will attempt to conduct real field phenotyping. We note that the lowest error was achieved by our proposed method in calculating the width of box A, which did this with an accuracy within a millimeter order of magnitude. However, for smaller dimensions such as the height of C or the width of B, it will be harder to obtain accurate measurements.

The use of our proposed method is not restricted to 3D reconstruction. We are currently assessing the procedure to collaborate with other agronomy researchers as well. One of the challenges is to observe flowering timing on paddy rice. Flowering (spikelet anthesis) is one of the most important phenotypic characteristics of paddy rice, and researchers expend efforts to observe flowering timing. Observing flowering is very time-consuming and labor-intensive, because it is still visually performed by humans. An image-based method that automatically detects the flowering of paddy rice is highly desirable. However, varying illumination, diversity of appearance of the flowering parts of the panicles, shape deformation, partial occlusion, and complex background make the development of such a method challenging. In Fig. 4.16, it shows that higher resolution can boost the accuracy to detect the flowers of paddy rice [16].

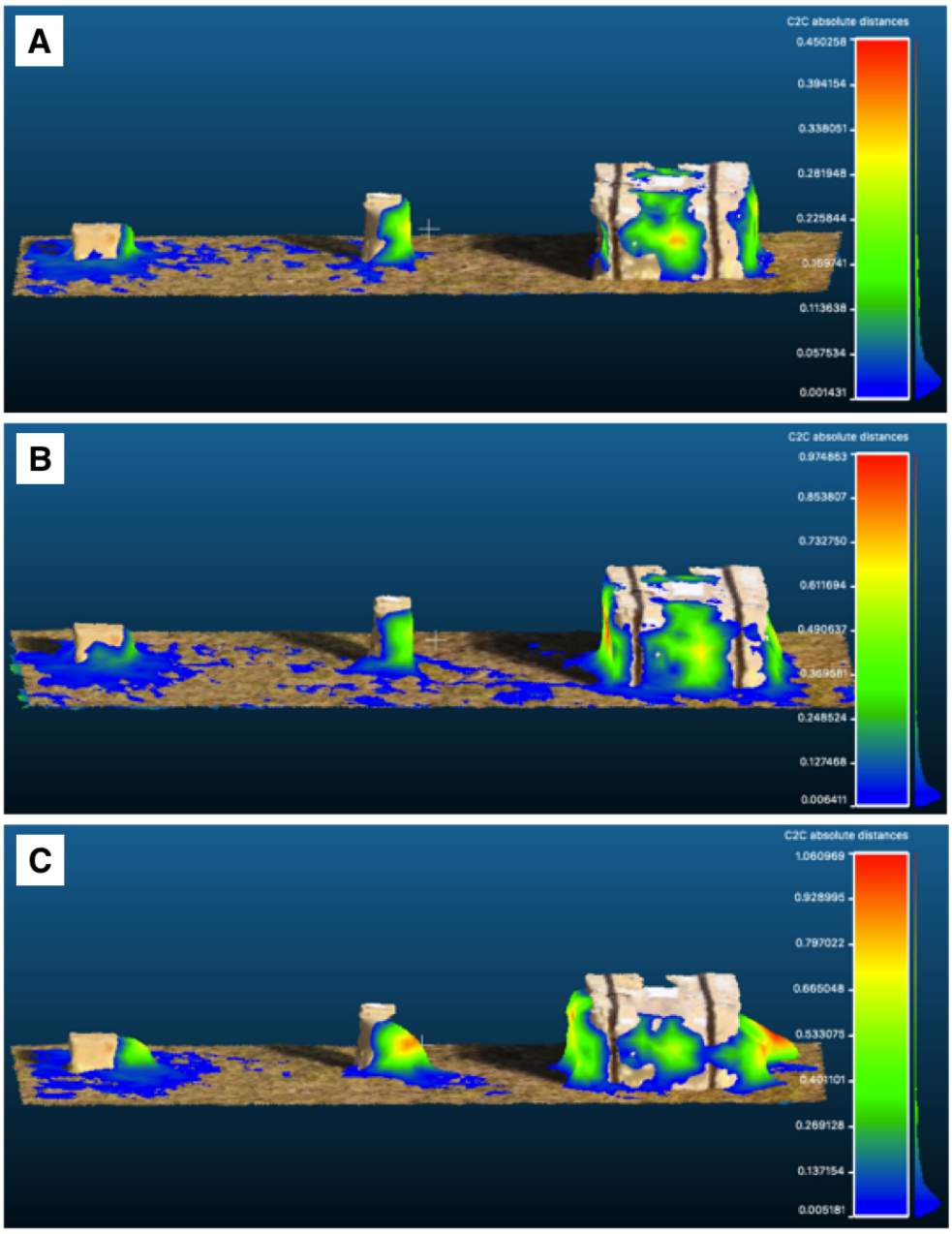


Figure 4.14: Alignment result from original image and particular methods on 5m's height images. (A) The proposed method (B) Bicubic, (C) Bilinear

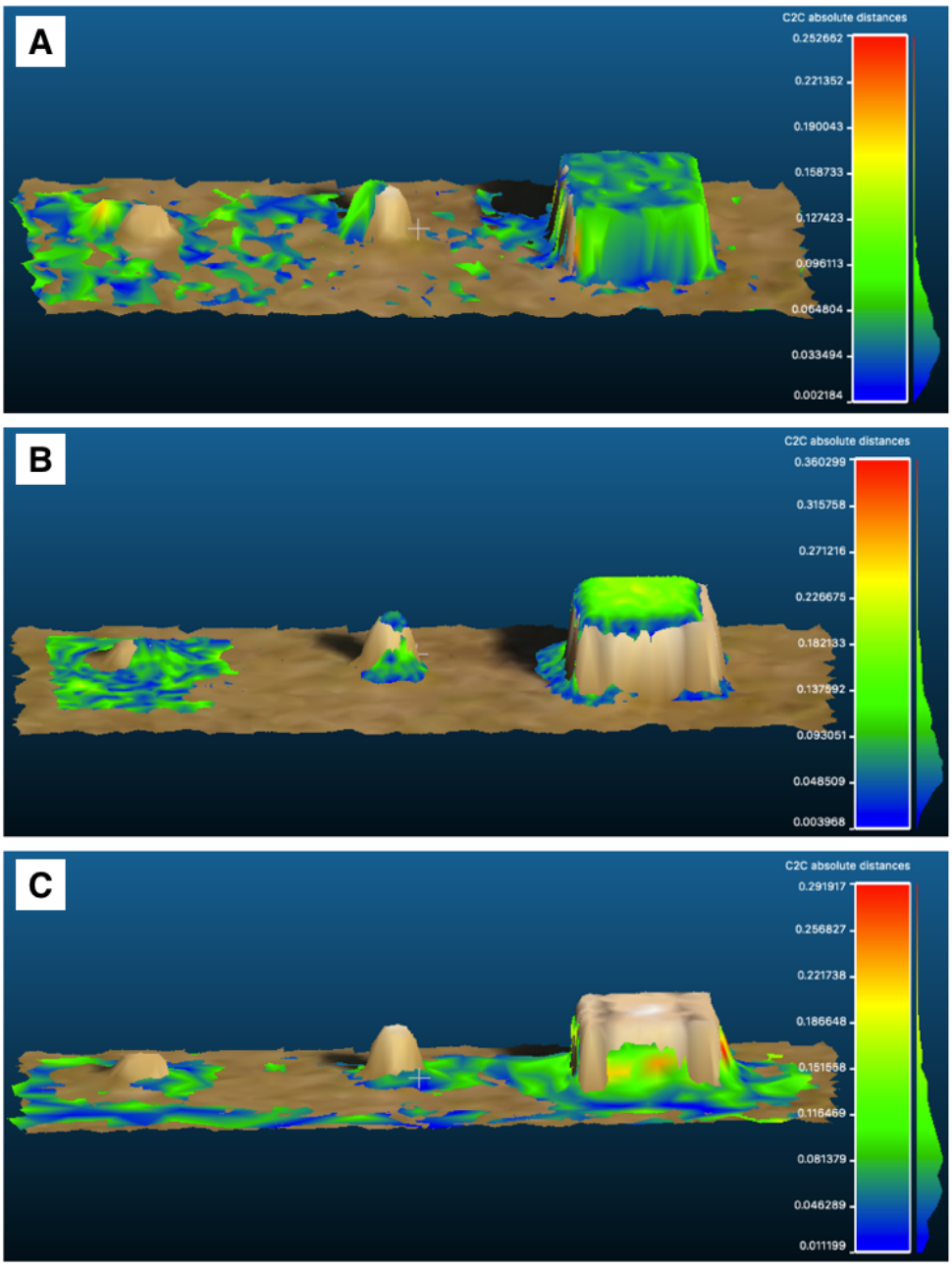


Figure 4.15: Alignment result from original image and particular methods on 10m's height images. (A) The proposed method (B) Bicubic, (C) Bilinear

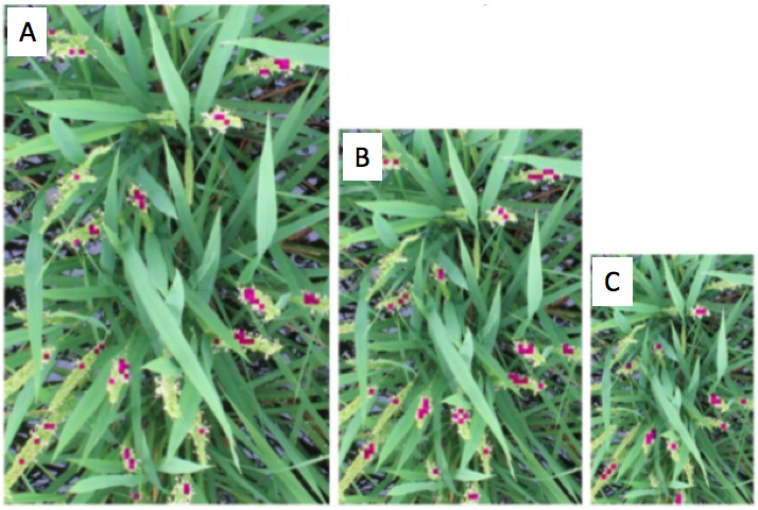


Figure 4.16: Example of flower detection in different resolution. Each violet block indicates a part of detected flower where higher resolution can provide better accuracy. (A) Size: 2001 x 1301 pixels, (B) Size: 1501 x 976, (C) Size: 1001 x 651 [16].

Chapter 5

Deep Residual Learning Super-resolution

5.1 Introduction

The availability of various types of images due to internet technologies provide big chance for learning algorithm to learn image characteristic deeply. This opportunity has been exploited by many researchers to develop robust super-resolution (SR) algorithms based on learning approaches. The main goal of SR is to recover high-frequency information from the input low-resolution (LR) image to be able to produce high-resolution (HR) one. Other goal of SR algorithm is to increase the accuracy of computer vision task. The SR algorithm is expected to reconstruct the LR input image in acceptable quality and resolution.

Currently, learning methods are widely used to map from LR to HR patches. Super-resolution using sparse representation shows its popularity because of the ability to naturally encode the semantic information of images [10]. By collecting representative samples in order to create an over-completed dictionary, it is possible to discover the correct basis for correctly encoding an input image. The studies by Yang et al. [44] and Zeyde et al. [45] focused on using a single pair of dictionaries; intuitively, however, using a single pair of dictionaries can produce many redundancies, which may cause instability during the image reconstruction process.

Lately, convolutional neural networks (CNN) is used in many image processing algorithm with large improvement in accuracy. On SR algorithm, Cao dong et al.[8] has demonstrated a CNNs' ability mapping LR to HR patches called Super-resolution Convolutional Neural Networks (SRCNN). The method is constructed by a very simple and a lightweight structure CNNs using two hidden layers and 3×3 filter size. Jiwon Kim et al. [22] introduces Very Deep Convolutional Networks (VDSR), a very deep CNN with residual learning, which proven have accurate result but have critical issues on convergence

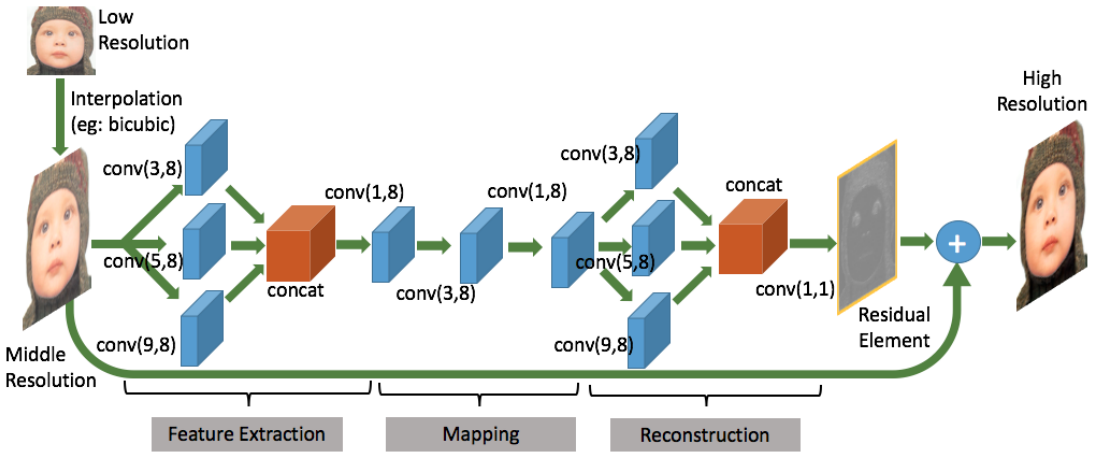


Figure 5.1: The proposed network.

speed. VDSR includes 20 layer of CNN using 3×3 filter size.

The recent improvement has been published. FSRCNN [9] demonstrated superior performance than previous SRCNN. They focused on improving the current SRCNN and proposed faster and more accurate algorithm. FSRCNN redesign the network using three main principal: deconvolution, dimension shrinking, and smaller filter.

In this paper, we propose fast convergence and low-computation convolutional network for image super-resolution as shown in Fig. 5.1. Our proposed network is inspired by inception module and residual learning. GoogleNet [34] introduces inception concept which use multiple type of filter size then combine it into one stream. This concept has been proven in the 2015 ILSVRC challenge. While, residual learning introduces by He et al. [19] to ease the training of networks and gain better accuracy.

The paper is organized as follows. Section 5.2 explains the proposed CNN's called Deep Residual Learning Super-resolution (DRLSR) and training strategy. Section 5.3 discusses the results of our experiments and analysis.

5.2 Proposed Method

This section is divided into two subsections: the proposed network and training strategy. In the proposed network, we explain how to achieve fast convergence and low-computation network. Moreover, the use of inception module in the image super-resolution. Next, the training strategy describe the initialization strategy, followed by tuning using multiple image dataset.

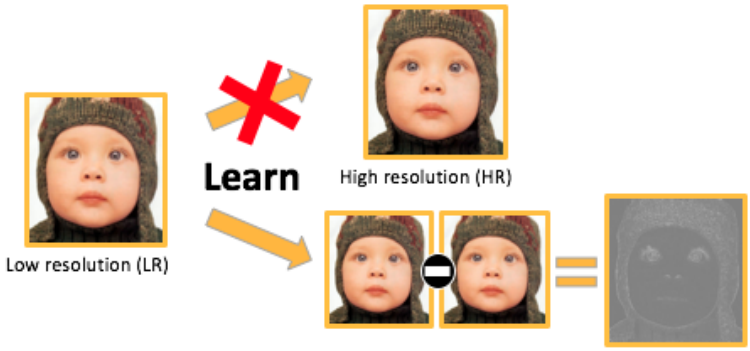


Figure 5.2: Residual Learning.

5.2.1 Proposed Network

In Fig. 5.1, we illustrate the proposed network. The network is only composed by convolutional layer. For better understanding, our network can be divided into 3 parts: feature extraction, mapping, and reconstruction. Let $conv(f_t, n_t)$ as convolutional layer where f_t represent the filter size and n_t represent the number of filters. In total, we use 10 convolutional layers.

We aim to have faster convergence during training. To achieve this objection, we construct the network using residual learning and gradient clipping. Residual learning [19] has the ability to ease the training of the networks. This assumption is motivated by the general SR problem where reconstructed HR images lose its HR component while doing enlargement process. This HR component can be substituted by residual component from the proposed network as illustrated in the Fig. 5.2.

In our proposed method, we use high learning rate to achieve fast convergence. However, high learning rate can effect the infinity loss during training process. Therefore, we use gradient clipping to avoid the infinity error. Gradient clipping is illustrated in Fig. 5.3. Gradient clipping is suitable for residual learning because it has the ability to limit the individual gradient to the predefined range. Using gradient clipping, we can avoid infinity error and ensure the fast convergence. The same concept also used by VDSR [22] which proven to have high PSNR value than other state-of-the-art methods.

The proposed network is inspired by Inception module from GoogLeNet [34] as shown in Fig. 5.4. The feature extraction and reconstruction parts exploit the ability of inception module. The inception module basically use because of the greedy assumption. It started from the confusion of using filter size. Instead of choosing the optimal filter size, it use multiple filter size, then combine results. We found that this ability is very suitable with image super-resolution concept where we can have various features from the image. Other

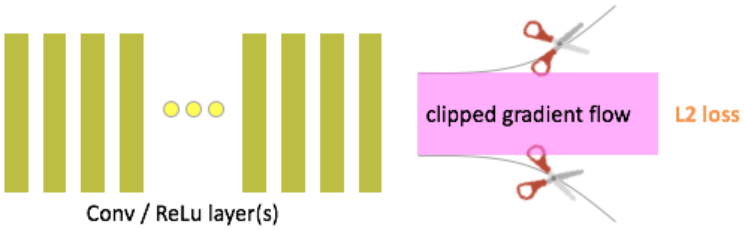


Figure 5.3: Gradient Clipping.

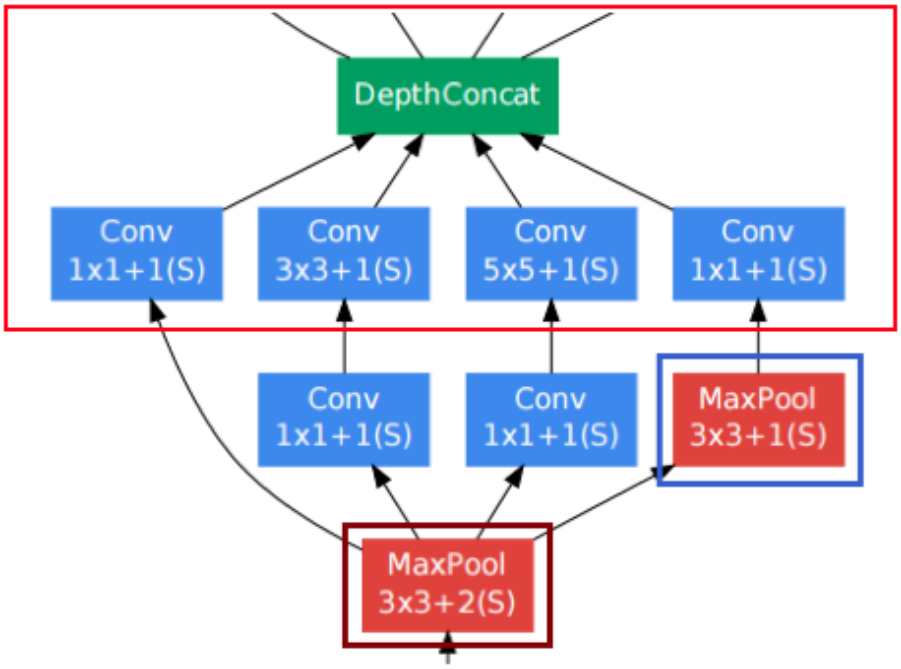


Figure 5.4: Inception module.

useful layer is 1×1 convolutional layer, which reduce dimension of the layer. We use 1×1 convolutional layer to select the best features in the feature extraction and reconstruction parts.

Low computation network is required due to the needs of real time application. Our proposed network optimized the use of number of filter. We analyze the optimum number of filter and filter size based on our experiment. The use of inception module can caused long computational time, however, we can still achieve fast computation by reducing the number of filter.

5.2.2 Training Strategy

The learning step is very crucial to construct optimal convolutional networks. Many researchers have been investigated this issue for last couple years. During initialization, the

Figure 5.5: Sample of 91-images dataset.

Figure 5.6: Sample of 100-general-images dataset.

std value for each layer should be determined. In our proposed network, we calculate the std value based on [20]. Let the std computed by $(\sqrt{2/n_l})$ where $n_l = k_l^2 d_l$. The k_l is the filter size, and d_l is the number of filter. For example, filter size $k_l = 3$ and $d_l = 8$, then the std is 0.111.

The use of various training image which have clear edge and texture also can give better performance to the proposed network. We adopt training images which used by Cao et al. [9]. First, we train our proposed network using 91 images as shown in Fig. 5.5. In the first training step, we use 0.1 learning rate for each convolutional layer. Then, after the network is saturated, we use mixed images, the combination of 91-images and 100-general-images for fine-tuning (see Fig. 5.6). In the tuning step, we increase the learning rate into 1 for each convolutional layer.

In the fine tuning step, the training images are generated using image augmentation process which proposed by Wang et al. [40]. We downscaled the image into several scales, and rotated each image with multiple degrees.

Figure 5.7: Set-5 dataset for experiment testing (From left to right: baby (size: 512×512), bird (size: 288×288), butterfly (*size* : 256×256), face (size: 280×280), woman (size: 228×344)).

5.3 Experimental Result

To confirm the efficiency of the proposed method, we conducted several experiments. The analysis of these experiments is divided into two subsections: quantitative and qualitative analyses. All experiments were conducted using Caffe in Windows 8 64bit, Intel Core i7@3.2GHz, RAM 32GB, NVIDIA GTX780. The images used in the experiment was taken from Yang et al. [44] and Dong et al. [9].

We only enhanced the brightness components (Y) while enhancing the other components using bicubic interpolation because human vision is more sensitive to brightness change. Then, each resulting image channel was combined to produce a final color image. This procedure will be very effective because it can speed up the computational process of the proposed method. Fig. 5.7 shows Set5 which used during testing step. First, we downscale the original image into one third from the original size to produce testing image. Then, testing image is used to be an input for various methods.

In our network, we used a set of 41×41 pixels sub-images taken from 91-images dataset with total size 267MB during the first training session. During this session, we use 0.1 learning rate for each convolutional layer. Then, we applied fine tuning of the network using a set of 41×41 pixels sub-images taken from 100-general-images dataset with total size 5.9GB. In the tuning session, we increased the learning rate into 1 for each convolutional layer.

Figure 5.8 shows that our network has very fast convergence. In the experiment, it shown that we only train the network for 300000 epochs compare to FSRCNN which use 12×10^8 epochs during training. In the fine tuning, we use another 300000 epochs, so in total, we construct the network only from 600000 epochs.

To evaluate the proposed method, we conducted experiments using $3\times$ magnified images to compare the proposed method to the other five methods: bicubic, sparse-based [45],

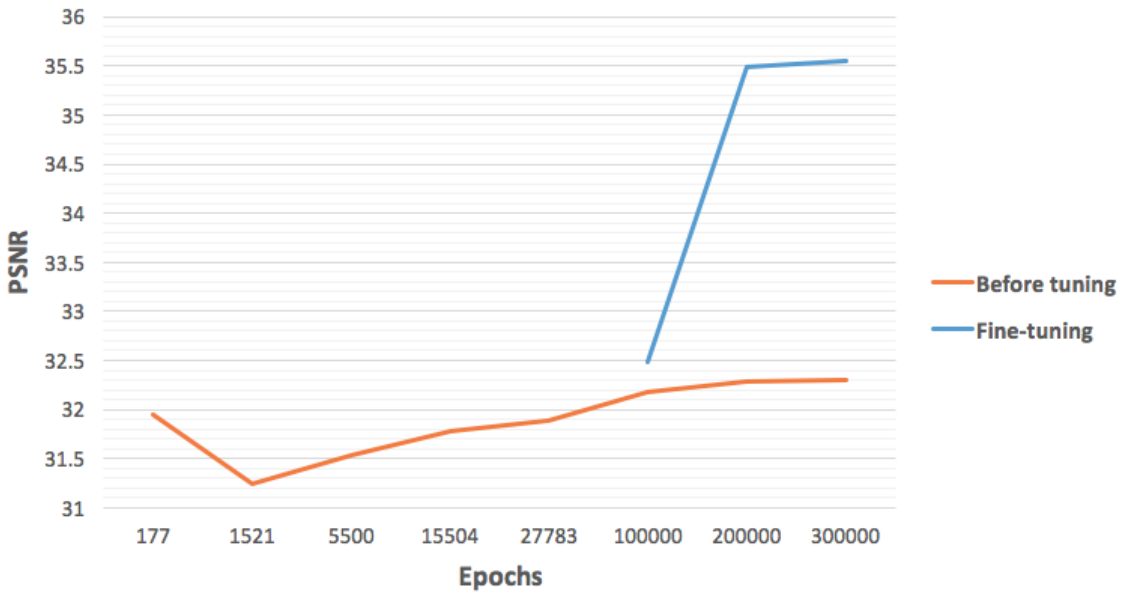


Figure 5.8: PSNR value during first training and fine-tuning.

SRCNN-ex[8], FSRCNN [9], and FSRCNN-s [9]. All parameters and training images are provided by the original authors using the optimized setting.

5.3.1 Quantitative Analysis

Methods for measuring the peak signal-to-noise ratio (PSNR) [21], structural similarity (SSIM) [41], feature similarity (FSIM) [48], and elapsed time were used for quantitative measurement. The PSNR, in decibels (dB), calculates the similarity between the original image and the upscaled image. SSIM is a method that measures the quality of images based on the structural content of the original and magnified images. FSIM is based on the fact that the human visual system processes an image mainly in terms of its low-level features. Two features are considered in FSIM computation: the primary feature, i.e., phase congruency (PC), which is a dimensionless measure of a local structure's significance; and the secondary feature, i.e., the image gradient magnitude. FSIM combines both of these features to characterize the local quality of an image. Higher values of PSNR, SSIM, and FSIM indicate better quality. CPU time was computed using Matlab functions (tic and toc) to measure the elapsed time for a certain process. All measurements used only the luminance channel (Y) to simplify and objectively calculate the error.

Table 5.1 and Table 5.2 list the average values from four measurements on Set5 and B100 datasets. The result confirm that our proposed method has the least computational time compare to other methods. However, it should be noted that there is not much differ-

Table 5.1: Comparison of the average quantitative results produced by PSNR, SSIM, FSIM, and computational time for 3× magnification on Set5.

Methods	PSNR	SSIM	FSIM	Time
Bicubic	30.392	0.868	0.897	0.001
Sparse-based [45]	31.906	0.897	0.924	1.035
SRCNN-ex [8]	32.749	0.909	0.941	3.471
FSRCNN [9]	33.155	0.914	0.946	2.588
FSRCNN-s [9]	32.604	0.906	0.938	1.274
Proposed	32.555	0.908	0.939	0.917

Table 5.2: Comparison of the average quantitative results produced by PSNR, SSIM, FSIM, and computational time for 3× magnification on B100 images.

Methods	PSNR	SSIM	FSIM	Time
Bicubic	27.207	0.738	0.827	0.001
Sparse-based [45]	27.875	0.773	0.856	1.565
SRCNN-ex [8]	28.412	0.786	0.876	5.321
FSRCNN [9]	28.518	0.789	0.878	3.417
FSRCNN-s [9]	28.284	0.783	0.873	1.690
Proposed	28.323	0.783	0.872	1.394

ence in PSNR, SSIM, and FSIM with other convolutional networks methods. For example, compared to FSRCNN-s who has the closest computational time, our method has higher PSNR value on B100 dataset. In Set5, our proposed method shows the higher value on SSIM and FSIM. The detail illustrations can be see in the qualitative analysis subsection.

Table 5.3 shows the detail measurement of computational time for all testing images. It is shown that our proposed network has the lowest computational time except Bicubic. This measurement did not use GPU processing, all measurement used CPU-based methods. Therefore, to achieve real-time application, the computational time can be reduced by implemented GPGPU-based method.

Table 5.3: Detail computational time for 3× magnification (in seconds).

Methods	Baby	Bird	Butterfly	Face	Woman
Bicubic	0.0029	0.0016	0.0009	0.0011	0.001
Sparse-based [45]	2.400	0.737	0.634	0.666	0.739
SRCNN-ex [8]	9.374	1.905	2.365	1.880	2.099
FSRCNN [9]	5.398	2.146	1.683	1.868	1.928
FSRCNN-s [9]	2.843	1.022	0.754	0.914	0.888
Proposed	2.625	0.481	0.529	0.485	0.510

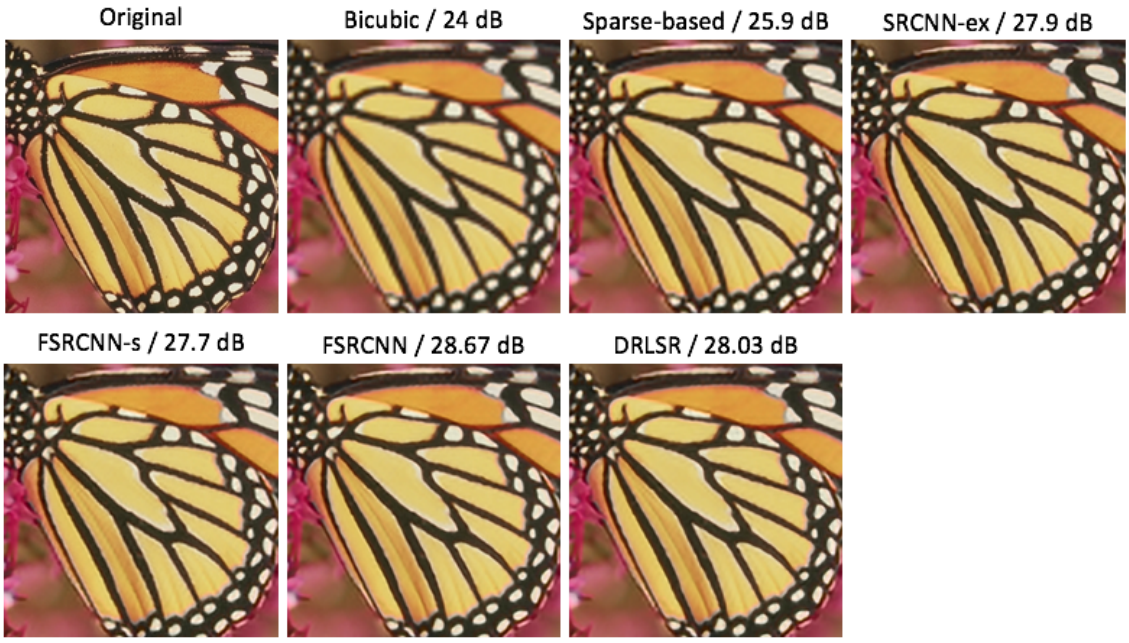


Figure 5.9: Results of experiment for 3× magnification on "Butterfly" image.

5.3.2 Qualitative Analysis

In the quantitative measurement, Table 4.2 shows our proposed network has the lowest PSNR value except for bicubic, and sparse-based. However, we can see by human visualizations that our method clearly produces sharper and smoother edges and is able to clearly construct the details of a scene especially better than FSRCNN-s. The FSRCNN-s still suffer from some artifact especially in the edge area. Bicubic and sparse-based also produced blurring effects in the enlarged image. The best result still produce by FSRCNN and SRCNN-ex which have the sharper edge and less artifact.

We specially compare our proposed network with FSRCNN-s because of the similarity in nature of low computation. On butterfly image (Fig. 5.9), our proposed network has higher PSNR value than FSRCNN-s. We can see the wings pattern produced by our proposed method has clear shape and pattern. While, FSRCNN-s produce halo artifact.

The same case also happen in the woman image (Fig. 5.10). In the chin area of the image, we can see our proposed network is able to produce sharper contour and shape, better than FSRCNN-s.

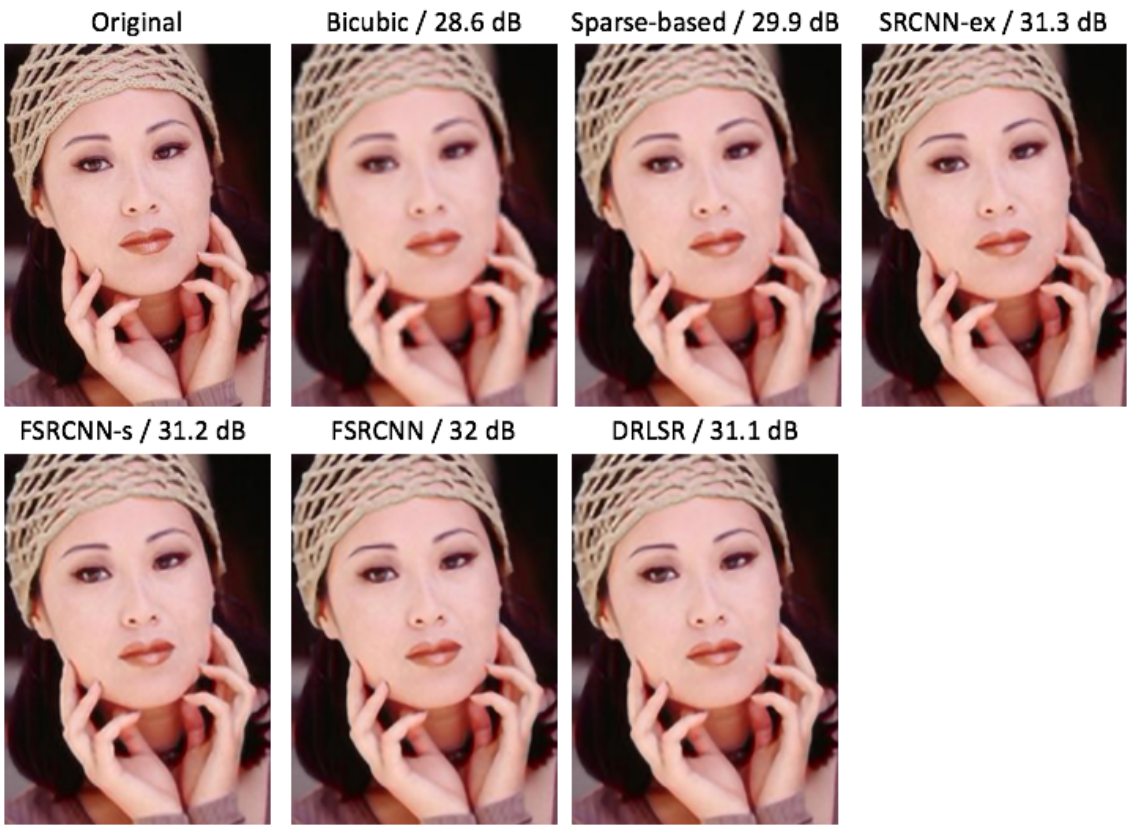


Figure 5.10: Results of experiment for 3× magnification on ”Woman” image.

Chapter 6

Conclusion and Future Works

6.1 Summary

The work of this thesis focused on the study of super-resolution (SR) as a technique to augment the spatial resolution of images, to a greater extent than conventional methods. In particular, we adopted the single-image SR approach based on filtering and learning methods. The filtering method predict the HR component based on curvature modeling using first-order derivatives. Then, the SR procedure based on machine learning paradigm, where the HR output image is predicted/estimated patch by patch: for each LR input patch we compute a model on the basis of local examples and we use this model to predict the related HR output patch.

In the first part, the main contribution is the extension of edge direction based on first-order derivatives for single-image SR. In the proposed method, we employ six edge directions and first-order derivatives as a feature to extract the interpolation direction. This is followed by a back-projection process to refine the image. The proposed method was implemented and evaluated. The results of our evaluations show that the our proposed method has the lowest computational complexity and demonstrates superior quality compared to other methods. The experiment results from both quantitative and qualitative analysis show that the proposed method outperforms previous method. Furthermore, the proposed method can preserve image details and reduce artifacts, such as blurring and ringing around edges.

In the second part, an SR based on adaptive multiple pairs of dictionaries for UAV images was proposed. The proposed method employs a classification based on edge orientation to obtain selective patches by creating five clusters, each of which obtains a pair of dictionaries A_l and A_h. The proposed method was implemented and out-performed other methods. The experimental results show the superiority of our proposed method for both quantitative and qualitative analysis by preserving detail and reducing artifacts such as

blurring and ringing around the edge. Our method was also proven effective for 3D reconstruction and produced an image superior to the original image from a 10m height. The use of a GPU application could further enhance our method by enabling opportunities to decrease its computational time.

In the third part, we proposed Deep Residual Learning Super-resolution (DRLSR). The network inspired by Inception module of GoogLeNet to produce multiple features during feature extraction and reconstruction process. Our strategies ensure the network having fast convergence and low computational time. The proposed network was assessed. The results show that our proposed network can cut half of computational time from the the-state-of-the-art network. Furthermore, our proposed network successfully exploit the Inception module and residual learning in the SR approach.

In summary, Fig. 6.1 shows the summary of our dissertation. We aim to solve the three main problems during SR implementation: computational time, sensitivity to training data, and quality improvement. In the beginning, we focus to create low computation SR algorithm which considered as filtering based method. Then, we investigate the ability of multiple sparse coding in the SR approach with insensitivity from training images. Finally, we develop efficient convolutional networks with superb quality compare to current-state-of-the-art methods. Moreover, the proposed method from Chapter 3 can be used as interpolated method to produce middle or medium resolution which is used in Chapter 4 and 5 to create training pairs.

6.2 Future works and perspectives

Apart from the results, we are aware that our work is far from finished. In the last section, we would highlight some questions as the future works.

First part is First-order Derivatives- based Super-resolution. The proposed method is very simple and light. However, the edge direction can be wrongly interpolated and cause some noise in the image. Currently, this noise can be polished by back-projection method. For the next step, we need to observe the edge and texture modeling using first-order derivatives. Furthermore, we can correctly interpolate the direction of the edge in the input image.

Second part is Super-Resolution via Adaptive Multiple Sparse Representation. Sparse-based method is notably one of the-state-of-the-art in the super-resolution methods. It has the ability to create basis to connect low-resolution image and high-resolution image. However, the sparse and dictionary initialization is crucial for this method. We can observe carefully the impact of the initialization. Moreover, the possibility of K-SVD giving the

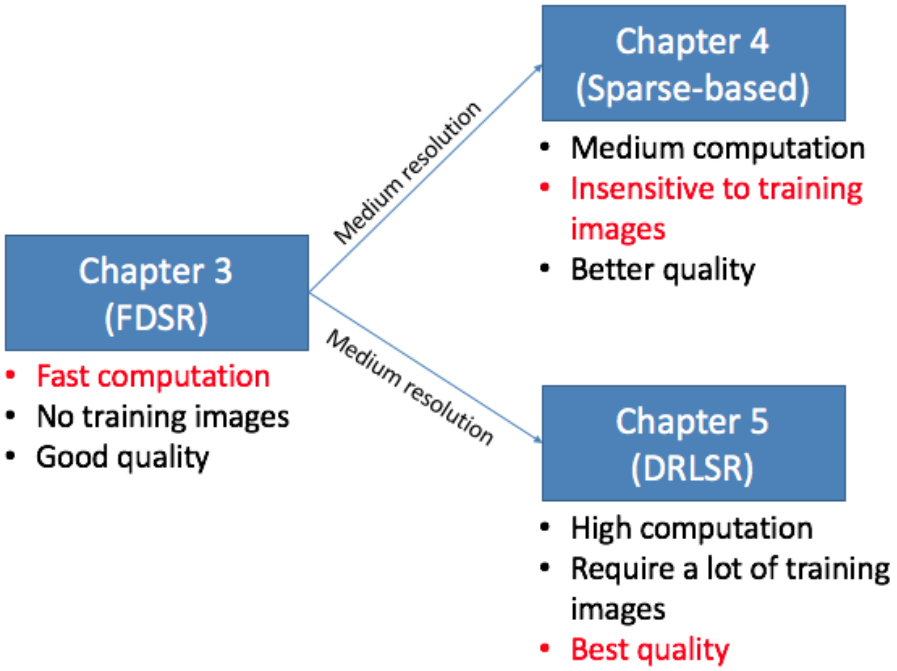


Figure 6.1: The summary of the proposed methods

local optimum solution is high especially using single dictionary. Therefore, the chance to improve the current methods is high.

The last part is Deep Residual Learning Super-resolution. We aim to have light network yet constructing clear and sharp HR image. In the experiment, we have not observed and analyzed deeply regarding the advantages of Inception modules and various settings. The current network have high possibility to be trapped in local optimum solution. In the future, more efficient network is need to be designed to produce better quality of obtained HR image.

Appendix A

Publication List

A. Peer-reviewed international journals:

(J1) M. Haris, K. Sawase, M. Rahmat Widyanto, and H. Nobuhara, "An efficient super resolution based on image dimensionality reduction using accumulative intensity gradient", Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 18, No. 4, pp. 518-528, July 20, (2014)

(J2) M. Haris, M. Rahmat Widyanto, and H. Nobuhara, "First-order derivative-based super-resolution", Signal, Image and Video Processing, Vol. 11, No. 1, pp. 1-8, January 16, (2017)

(J3) M. Haris, T. Watanabe, L. Fan, M. Rahmat Widyanto, H. Nobuhara, "Super-Resolution for UAV Images via Adaptive Multiple Sparse Representation and its Application to 3D Reconstruction", IEEE Trans. On Geoscience and Remote Sensing (minor revision)

B. Peer-reviewed international conferences:

(C1) M. Haris, K. Sawase, T. Sawada, K. Kamijima, M. R. Widyanto, H. Nobuhara, "Parameter Optimization of Fast Curvature Based Interpolation Using Genetic Algorithm," Fifth International Symposium on Computational Intelligence and Industrial Applications (ISCIIA2012), Sapporo, Japan, Aug. 24 – 26, (2012)

(C2) M. Haris, K. Sawase, T. Shimizu, T. Yoshioka, R. Widyanto, H. Nobuhara, "Example Based Super-resolution Using Neighbor Edge Similarity on Aerial Images," 2013 International Workshop on Smart Info-Media Systems in Asia (SISA2013), Aichi Industry and Labor Center, Nagoya, Japan, Sep. 30 – Oct. 2, (2013)

(C3) K. Maekawa, D. Harima, M. Haris, K. Sawase, H. Nobuhara, “Multi-resolution Dijkstra’s algorithm for multi-agent simulation and its application to disaster management,” The 3rd International Workshop on Soft Computing and Disaster Control (SocDic2013), Bali, Indonesia, Nov. 9 – 10, (2013)

(C4) M. Haris, T. Sugiura, L. Ziyang, K. Ishii, K. Sawase, and H. Nobuhara, “Optimized Example Based Super Resolution On UAV Imagery Using Probabilistic Tree and Neigbor Similarity,” The 3rd International Workshop on Soft Computing and Disaster Control (SocDic2013), Bali, Indonesia, Nov. 9 – 10, (2013)

(C5) M. Haris, K. Ishii, L. Ziyang, T. Sugiura, M. Qi, T. Watanabe, S. Sukisaki, T. Tanabata, S. Isobe, T. Shimizu, T. Yoshioka, H. Nobuhara, “High-resolution Digital Map Construction Aims to Support Citrus Breeding Using Autonomous Multicopter,” The Third International Symposium on Citrus Biotechnology (ISCB2014), Shizuoka, Japan, Nov. 11 – 14, (2014)

(C6) M. Haris, and H. Nobuhara, “Super-resolution Based on Edge-aware Sparse Representation via Multiple Dictionaries,” The 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP2016), Vol. 3, pp. 40-47, Rome, Italy, Feb. 27 – 29, (2016)

(C7) T. Sugiura, M. Haris and H. Nobuhara, “Tactile Feedback for Intuitive UAV Control System,” 2nd International Conference on Digital Fabrication (ICDF2016), Tokyo, Japan, Mar. 3 – 5, (2016)

(C8) M. Haris, S. Sukisaki, R. Shimomura, Z. Heming, L. Hongyang, and H. Nobuhara, “Development of High-Precision 3D Measurement On Agriculture Using Multiple UAVs”, The 42nd Asia Pacific Advanced Network (APAN42), Hong Kong, Jul. p. 47-55, 31 July – Aug. 5, (2016)

C. Domestic conferences / workshops:

(D1) 劉 子揚，杉浦 巧美，ムハマド ハリス，苗琦，石井　健登，澤勢　一史, 延原 肇, クアッドコプターによる空撮映像を用いたパノラマ画像の生成および植生観測への応用, 日本知能情報ファジィ学会 知的システム研究会, 筑波大学, 2013年7月6日

(D2) 苗 キ, ムハマド ハリス, 杉浦 巧美, 劉 子揚, 石井 健登, 澤勢 一史, 延原 肇, 育種支援のためのデジタルマップ・アーカイビングおよびその領域分割と特徴解析手法の提案, 日本知能情報ファジィ学会 第64回知的システム研究会, 法政大学（東京）, 2013年12月7日

(D3) 劉子揚, 杉浦巧美, ムハマド ハリス, 苗キ, 石井健登, 延原肇, 清水徳朗, 吉岡照高, マルチコプターを用いたカンキツ樹木のデジタルマップ・アーカイビングとその育種支援への応用, ロボティクス・メカトロニクス講演会2014, 富山国際会議場（富山県）, 2014年5月25日-29日

(D4) 杉浦巧美, 劉子揚, ムハマド ハリス, 苗キ, 石井健登, 延原肇, クアッドロータにおける可変アームの実現とその安定性解析シミュレーション, ロボティクス・メカトロニクス講演会2014, 富山国際会議場（富山県）, 2014年5月25日-29日

(D5) 鋤先星汰，石井健登，劉子揚，杉浦巧美，ムハマド ハリス，苗琦，渡邉拓也，延原肇, 振幅変調赤外光による自己位置発信とその3 次元位置推定, 第15回　計測自動制御学会システムインテグレーション部門講演会, 東京ビッグサイト（東京）, 2014年12月15日-17日

(D6) 劉子揚，石井健登，杉浦巧美，鋤先星汰，ムハマド ハリス，苗琦，渡邉拓也，延原肇, クアッドコプターに対するファジィ制御とPID制御の適用比較, 第15回　計測自動制御学会システムインテグレーション部門講演会, 東京ビッグサイト（東京）, 2014年12月15日-17日

(D7) 杉浦巧美，石井健登，苗琦，ムハマド ハリス，劉子揚，渡邉拓也，鋤先星汰，延原肇, 小型UAV のランダムウォークによる3D 環境モデリング, 第15回　計測自動制御学会システムインテグレーション部門講演会, 東京ビッグサイト（東京）, 2014年12月15日-17日

(D8) 苗琦, 渡邊 拓也, ムハマド ハリス, 杉浦 巧美, 鋤先 星汰, 延原 肇, 磯部祥子, 七夕高也, 大規模屋外圃場を対象とした情報視覚化とそれに基づく適応型ビューアーの構築, 第21回画像センシングシンポジウム (SSII15) パシフィコ横浜アネックスホール, 2015年6月10日-12日

(D9) 渡邊 拓也, ムハマド ハリス, 苗琦, 杉浦 巧美, 鋤先 星汰, 延原 肇, 磯部祥子, 七夕高也, 高周波成分および自己相似性を考慮した学習型超解像と農業支援用UAV空撮画像への応用, 第21回画像センシングシンポジウム (SSII15) パシフィコ横浜アネックスホール, 2015年6月10日-12日

(D10) 霜村 瞭, 鋤先 星汰, 張鶴鳴, 李宏陽, ムハマド ハリス, 延原肇, 複数ドローン編隊飛行に基づく同時画像撮影と高精細計測への応用, 第34回日本ロボット学会学術講演会（RSJ2016）, 山形大学, 2016年9月7日-9日

(D11) 鋤先 星汰, ムハマド ハリス, 霜村 瞭, 張 鶴鳴, 李 宏陽, 延原 肇, 振幅変調パルス光を用いたドローン位置推定システムにおけるセンサー群の構成とキャリブレーション方法, 農業×計測×情報通信ワークショップ, 名古屋大学 東山キャンパス, 2016年11月17日-18日

D. Magazines

(B1) 延原 肇, ムハマド ハリス, 渡邊 拓也，鋤先 星汰, “解像度不足のギザギザさようなら！ 高精細時代の基本技術 超解像処理アルゴリズム入門”, Interface, CQ出版, pp. 28-38, 2015年6月号

(B2) 延原 肇, ムハマド ハリス, 渡邊 拓也，鋤先 星汰, “画像アルゴリズムの実験にピッタリ！無償版で試せる！ MATLAB×ラズベリー・パイ！超解像初体験”, Interface, CQ出版, pp. 40-49, 2015年6月号

E. Awards

(I1) Scholarship for Master Program from Ministry of Communication and Informatics, Indonesia (Nov 2011 – March 2014)

(I2) Indonesia Endowment Fund for Education Scholarship from the Ministry of Finance, Indonesia (April 2014 – March 2017)

(I3) The 42nd Asia Pacific Advanced Network (APAN42) Best Student Paper Award (1st August 2016)

Bibliography

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[2] Nicola Asuni and Andrea Giachetti. Accuracy improvements and artifacts removal in edge based image interpolation. *VISAPP (1)'08*, pages 58–65, 2008.

[3] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[4] Turgay Celik and Tardi Tjahjadi. Image resolution enhancement using dual-tree complex wavelet transform. *Geoscience and Remote Sensing Letters, IEEE*, 7(3):554–557, 2010.

[5] Dengxin Dai, Yujian Wang, Yuhua Chen, and Luc Van Gool. Is image super-resolution helpful for other vision tasks? In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[6] Aram Danielyan, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image upsampling via spatially adaptive block-matching filtering. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE, 2008.

[7] Hasan Demirel and Gholamreza Anbarjafari. Discrete wavelet transform-based satellite image resolution enhancement. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(6):1997–2004, 2011.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[9] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016.

[10] Michael Elad. Sparse and redundant representation modeling - what next? *Signal Processing Letters, IEEE*, 19(12):922–928, 2012.

[11] J Everaerts et al. The use of unmanned aerial vehicles (uavs) for remote sensing and mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:1187–1192, 2008.

[12] Tokihiro Fukatsu and Masayuki Hirafuji. Field monitoring using sensor-nodes with a web server. *Journal of Robotics and Mechatronics*, 17(2):164–172, 2005.

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[14] A. Giachetti and N. Asuni. Real time artifact-free image upscaling. *Image Processing, IEEE Transactions on*, 20(10):2760–2768, October 2011.

[15] GJ Grenzdörffer, A Engel, and B Teichert. The photogrammetric potential of low-cost uavs in forestry and agriculture. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 31(B3):1207–1214, 2008.

[16] Wei Guo, Tokihiro Fukatsu, and Seishi Ninomiya. Automated characterization of flowering dynamics in rice using field-acquired time-series rgb images. *Plant Methods*, 11(1):1–15, 2015.

[17] Muhammad Haris, Kazuhito Sawase, Muhammad Rahmat Widyanto, and Hajime Nobuhara. An efficient super resolution based on image dimensionality reduction using accumulative intensity gradient. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 18(4):518–528, 2014.

[18] Muhammad Haris, M Rahmat Widyanto, and Hajime Nobuhara. First-order derivative-based super-resolution. *Signal, Image and Video Processing*, 11(1):1–8, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[21] Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993.

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral)*, June 2016.

[23] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1127–1133, 2010.

[24] Xin Li and Michael T. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10:1521–1527, 2001.

[25] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 209–216. IEEE, 2011.

[26] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[27] Stéphane Mallat and Guoshen Yu. Super-resolution with sparse mixing estimators. *Image Processing, IEEE Transactions on*, 19(11):2889–2900, 2010.

[28] Eric Mjolsness. *Fingerprint Hallucination*. PhD thesis, California Institute of Technology, 1985.

[29] Kamal Nasrollahi and Thomas B Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6):1423–1468, 2014.

[30] M.A. Nuno-Maganda and M.O. Arias-Estrada. Real-time fpga-based architecture for bicubic interpolation: an application for digital image scaling. In *Reconfigurable Computing and FPGAs, 2005. ReConFig 2005. International Conference on*, pages 8 pp.–1, Sept 2005.

[31] Sung Cheol Park, Min Kyu Park, and Kang Moon Gi. Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20:21–36, 2003.

[32] Santhosh K Seelan, Soizik Laguette, Grant M Casady, and George A Seielstad. Remote sensing applications for precision agriculture: A learning community approach. *Remote Sensing of Environment*, 88(1):157–169, 2003.

[33] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[35] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[36] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. Kernel regression for image processing and reconstruction. *Image Processing, IEEE Transactions on*, 16(2):349–366, 2007.

[37] Hiroyuki Takeda, Peyman Milanfar, Matan Protter, and Michael Elad. Super-resolution without explicit subpixel motion estimation. *Image Processing, IEEE Transactions on*, 18(9):1958–1975, 2009.

[38] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[39] Qing Wang and Rabab Kreidieh Ward. A new orientation-adaptive interpolation method. *IEEE Transactions on Image Processing*, 16(4):889–900, 2007.

[40] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deeply improved sparse coding for image super-resolution. *arXiv preprint arXiv:1507.08905*, 2015.

[41] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[42] GA Watson. Computing helmert transformations. *Journal of Computational and Applied Mathematics*, 197(2):387–394, 2006.

[43] Chih-Yuan Yang. *Example-Based Single-Image Super-Resolution*. PhD thesis, UNIVERSITY OF CALIFORNIA, MERCED, 2015.

[44] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010.

[45] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2012.

[46] Chunhua Zhang and John M Kovacs. The application of small unmanned aerial systems for precision agriculture: a review. *Precision agriculture*, 13(6):693–712, 2012.

[47] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *Image Processing, IEEE Transactions on*, 15(8):2226–2238, 2006.

[48] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: a feature similarity index for image quality assessment. *Image Processing, IEEE Transactions on*, 20(8):2378–2386, 2011.