**CHAPTER 1**

**1.1 INTRODUCTION ABOUT THE PROJECT:**

Single-image super-resolution (SR) aims at restoring a high resolution (HR) image from a single low-resolution (LR) input image. Since multiple HR images can be recovered from a single LR observation, single image SR is heavily ill-posed. To address this problem, additional prior information is required to constrain the solution space. Internal statistics of a single natural image provides useful priors [1], and has shown strong power to solve the SR problem .

Patches in a natural image tend to redundantly recur many times inside the image, both within the same scale, as well as across different scales [2]. Based on this internal statistical prior, Glasner et al. [2] presented a unified SR framework that combines classical multi-image and example-based SR methods. Yang et al. [3] proposed a SR method that exploits self-similarities and group structural information of image patches using only a single input image. Freedman and Fattal [4] followed a local self-similarity assumption on natural images and extracted nearest neighbours from extremely localized regions in the input image, accordingly reducing considerably the nearest-patch search time. Similar to [2], Zhang et al. [5] exploited similarity redundancy across different scales in a given LR image itself to achieve example-based SR using the neighbour embedding algorithm [9], and applied the nonlocal means method [10] to learn the similarity within the same scale. Bevilacqua et al. [6] made use of a ”double pyramid” of images, built starting from the input image itself, to extract the self-examples, and then employed a regression based method to directly map each LR patch into its HR version. To expand the internal patch space and achieve more stable nearest-patch search, Choi et al. [7] utilized context dependent multi-shaped sub patches and Huang et al. [8] used a factored patch transformation representation for simultaneously accounting for both planar perspective distortion and affine shape deformation of image patches.

Apart from the above internal example-based approaches, another kind of example-based method uses an external database of natural images to extract a large number of training LR and HR image patch pairs. These methods usually use machine learning techniques to learn the relationship between LR and HR image patches. Time often et al. proposed an anchored neighbourhood regression method [11] and its refined variant called A+ [12], where learned dictionary elements are used as the anchor points and multiple linear regressors are computed to map LR to HR feature subspace. Recently, deep learning was introduced to address the single image SR problem.

Dong et al. [13] proposed a Super-Resolution Convolutional Neural Network (SRCNN) model that directly learns an end-to-end mapping between LR and HR images. Motivated by the A+ method, in this paper, we propose an anchored neighbourhood regression based self-leaning SR model that restores a visually pleasing HR image from a single input image without using any extra training images. In the proposed method, we directly use sampled self-examples as the anchor points without training the sparse dictionary pair and learn multiple linear mapping functions based on anchored neighbourhood regression to transform LR space into HR space. Moreover, we utilize the flipped and rotated versions of the self-examples to expand the internal patch space for obtaining more compact subspaces spanned by the anchor points.

Experimental results show that the proposed model outperforms the compared state-of-the-art methods. The remainder of the paper is organized as follows: we describe existing method and our proposed method in section 2 and 3. We show experimental results in section 4, and section 5 concludes this paper. Images with high-resolution (HR) can offer more details which may be critical in various applications such as remote sensing [1], medical diagnostic [2] and intelligent surveillance [3]. However, limited to hardware and imaging conditions, we can only obtain down-sampled low-resolution (LR)images in practice. Super-resolution (SR) aims to induce an estimated HR image from an observed LR image, which is one of the enduring active research topics in the image processing community, while it is also a challenging task because it is a typical ill-posed inverse problem [4] Single image SR research spans decades. The traditional interpolation methods such as nearest neighbour bilinear or bicubic are still in broad use [5]–[8] because of low cost and computational complexity.

However, these interpolation based SR methods are prone to blur high-frequency details and therefore often lead to blurring edges and unclear textures in the super-resolved HR image. The example based SR direction is the one currently most active. According to the source of example samples, the example based SR methods can be divided into two subclasses: the internal example SR methods and the external example SR methods. The former solely extract and organize example patches online from the input LR image, which leads to high computational complexities [9]–[11]; the latter use external data [12]–[19]and move off-line most of the computation required to extract and build useful priors and models [20]–[25], which reduce the computational complexity in the SR processing greatly and often have better SR performance. To make these external example SR methods successful, different assumptions and priors are utilized. Neighbor embedding (NE) [12]–[14] approaches assume that LR and HR image patches lie on low dimensional nonlinear manifolds with locally similar geometry. Using the Locally Linear Embedding (LLE) [26] technique in manifold learning, Chang et al. propose a NE+LLE SR method [14],which assumes that the patch space is populated densely enough but it is often incorrect in practice. Assuming sparsity and performing sparse coding over learned dictionaries of LR and HR patches, Yang et al. [18] propose a SR method which just needs considerably smaller dictionary than the NE+LLE method. However, the time complexity increases greatly because the computation of sparse coding is required, and then Ze yd e et al. [19] speed up the overall frame work by using K-SVD and orthogonal matching pursuit to enforce sparsity.

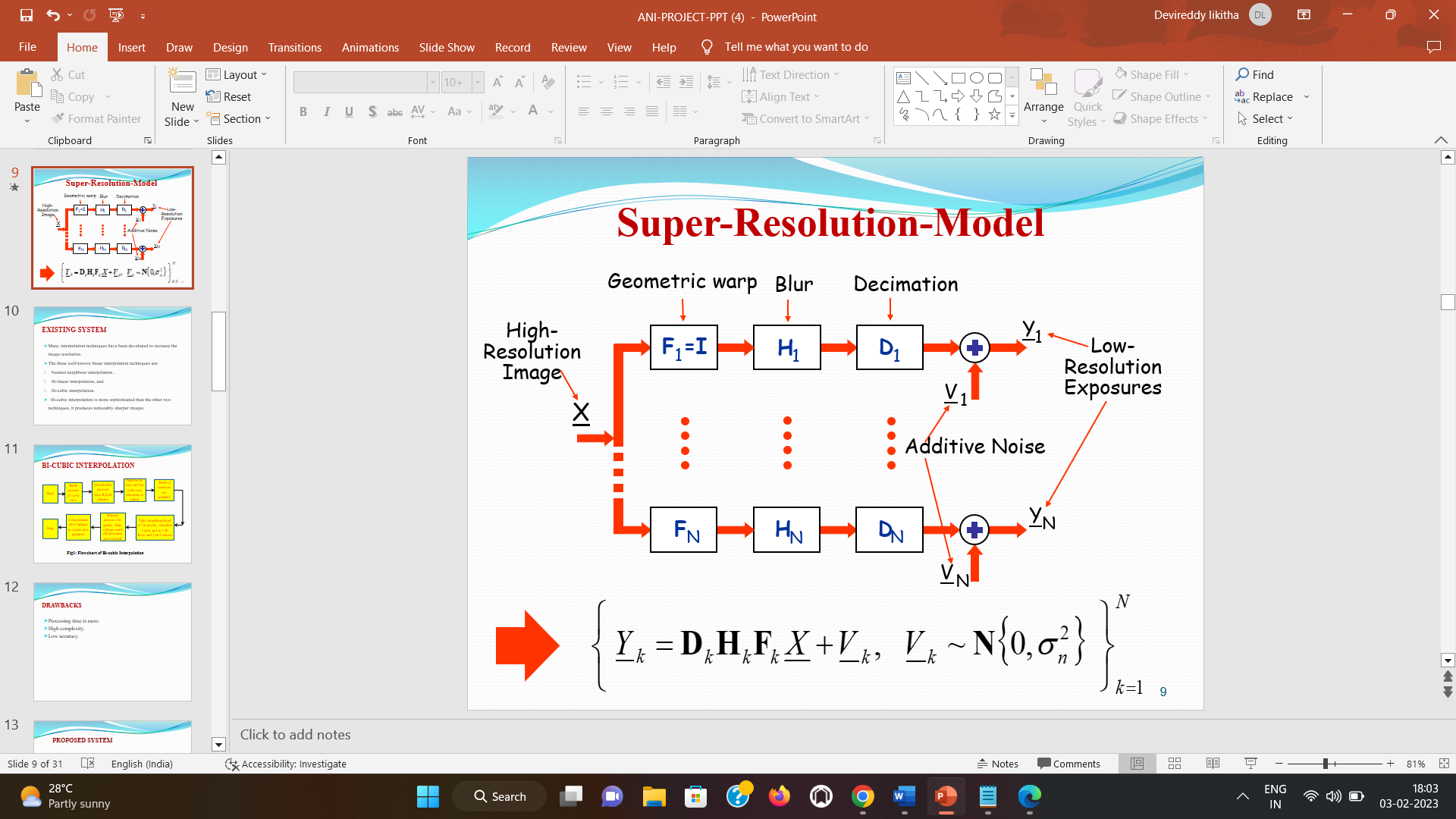
Timoften et al. [20] go further with Anchored Neighborhood Regression (ANR) method, which transforms the SR task into a linear search followed by a regression of the LR input patch. To improve the performance of ANR method, Jiang et al. [21], [22] propose Locally regularized Anchored Neighborhood Regression (LANR) which assigns different weight for each neighbor dictionary atom according to its correlation to the input LR patch. The follow-up AdjustedANR (A+) method [23] computes the regressors from training LR and HR sample patches instead of the dictionary, which uses the prior data better for performance improvement and becomes one of the current state-of-the-art methods for single image SR.A core assumption of the A+ method is that a given LR image patch can be linearly represented by K nearest LR sample neighbours of the nearest LR atom, and the estimated HR image patch can be linearly represented by the corresponding K nearest HR sample neighbours with the same representation coefficients. It should be noted that these K nearest HR sample neighbours are just chosen automatically according to the corresponding similar LR sample neighbours, but not to the similarities of the HR samples themselves. However, SR is an ill-posed problem due to the notorious ambiguity of patch correspondence: a LR image patch can be the down-sampled version of different non-similar

HR image patches. More importantly, the representation coefficients used for HR regression only depend on the similarities between the LR image patches, which implies that non-similar HR sample patches may be wrongly assigned large representation coefficients just because the corresponding LR sample patches are similar. To remedy this, we propose to introduce high-resolution similarity to adjust the HR representation coefficients, namely similar HR sample

neighbors have large representation coefficients and nonsimilar HR sample neighbours only have small representation coefficients. By doing so, the adjusted representation coefficients can reflect similarities between HR image patches more accurately which will be beneficial for SR performance improvement. It should be noted that our method is similar but different with the LANR method [21]. On the one hand, we compute the regressors from training LR and HR

sample patches instead of the dictionary which is used in LANR method; on the other hand, and more importantly, the adjusting weights in our method are computed according to the HR similarities but the weights in LANR method are computed according to LR correlations. The rest of this paper is organized as follows: in Section II, we introduce HR similarity directed adjusted anchored neighbourhood regression for single image SR in detail.

**1.2 SUPER-RESOLUTION-METHOD:-**



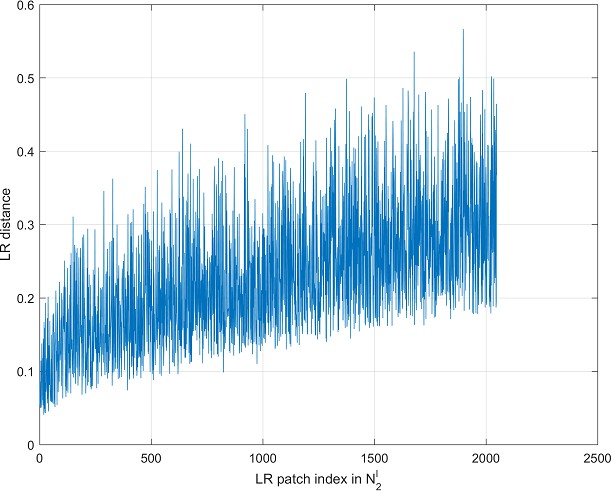
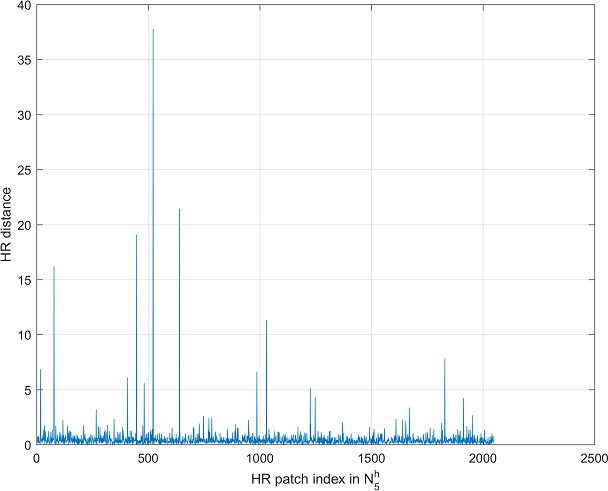
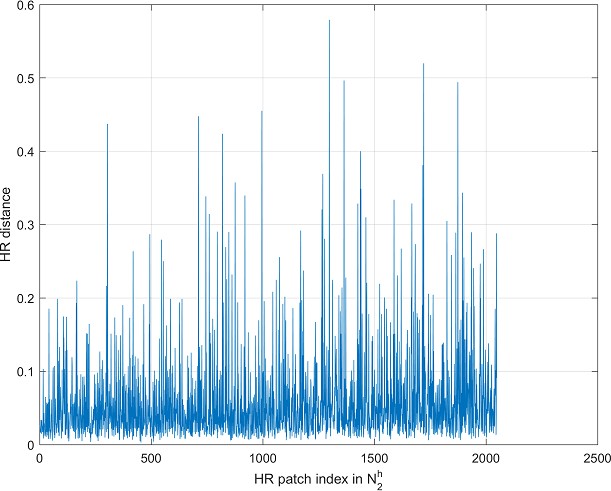
Super-resolution is based on the idea that a combination of low resolution (noisy) sequence of images of a scene can be used to generate a high resolution image or image sequence. Thus it attempts to reconstruct the original scene image with high resolution given a set of observed images at lower resolution. Super Resolution creates a single image with two times the linear resolution. That means the enhanced image will have twice the width and twice the height of the original image, or four times the total pixel count. The primary benefit of super-resolution is enhanced resolving power, with several super-resolution options available that allow the imaging of structures below 200 nm.

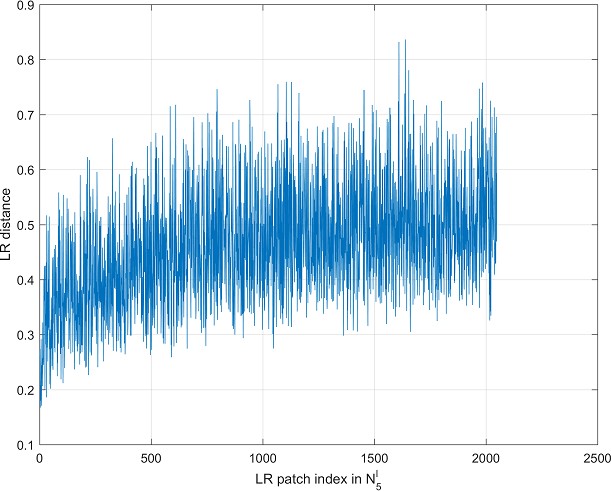
**1.3 HIGH RESOLUTION SIMILARITY DIRECTED ADJUSTED ANCHORED NEIGHBORHOOD REGRESSION A (REVIEW AND ANALYSIS OF THE A+ METHOD:-**

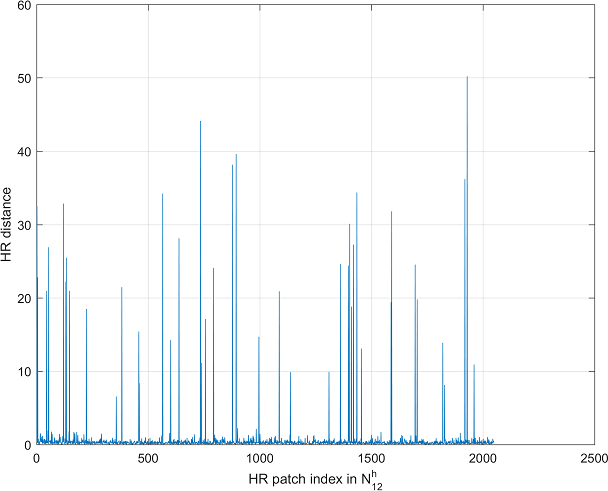
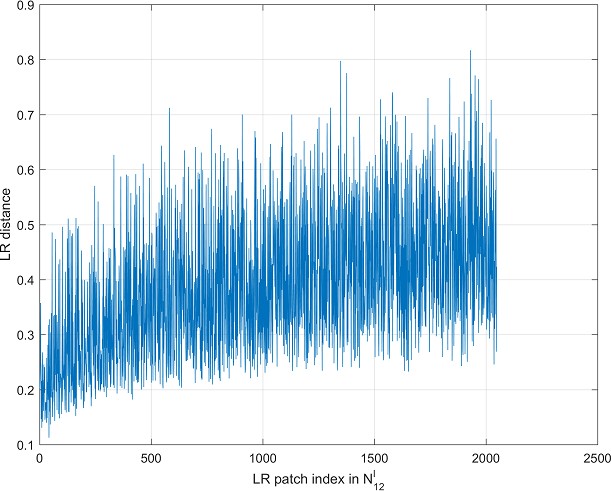
The A+ method has received considerable attention in recent years due to its top performance and low time complexity both at training and testing. The training data consists of extracted LR sample patches X = [p l 1 , p l 2 , · · · , p l M ] ∈ R dl×M and the corresponding HR sample patches Y = [p h 1 , p h 2 , · · · , p h M ] ∈ R dh×M with M the number of training samples. The A+ method contains the following key steps: (i) Dictionary learning: obtain the LR sparse dictionary D l = [d l 1 , d l 2 , · · · , d l N ] ∈ R dl×N and the corresponding HR dictionary D h = [d h 1 , d h 2 , · · · , d h N ] ∈ R dh×N of the training data by using the sparse method proposed by Zeyde et al. [19], where N M is the number of dictionary atoms. In the ANR method and the A+ method, the LR sparse dictionary atoms are called anchor points.

**(ii) Projection matrix computation**: for each anchor point d l j , find its K nearest LR sample neighbors N l j = [p l j,1 , p l j,2 , · · · , p l j,K ] in X and obtain the corresponding HR sample neighbors N h j = [p h j,1 , p h j,2 , · · · , p h j,K ] in Y automatically, and then compute the projection matrix corresponding to the anchor point d l j as follows Pj = N h j [(N l j ) T N l j + λI] −1 (N l j ) T , (1) where λ > 0 is a constant parameter.

(iii) SR processing: for a given LR image patch p l , find its nearest anchor point d l j in D l and the corresponding projection matrix Pj , and then obtain the estimated superresolved HR image patch as pˆ h = Pjp l . All anchor points and projection matrixes have been obtained and stored in the training process, and therefore the computational cost in the SR processing is very low. In the above steps, the step (ii) is the core of the A+ method, which is based on two assumptions: (ii-a) Any LR image patch p l can be linear represented approximately by the K nearest sample neighbours N l j of its nearest anchor point d l j , where the optimal representation coefficients (α ∗ j,1 , · · · , α∗ j,K ) T are computed according to the following ridge regression problem α ∗ j = (α ∗ j,1 , · · · , α∗ j,K ) T = arg min αj,1,··· ,αj,K ||p l − X K m=1 αj,mp l j,m ||2 2 + λ X K m=1 α 2 j,m = arg min αj∈RK ||p l − N l jαj ||2 2 + λ||αj ||2 2 , (2) which has explicit solution as follows α ∗ j = [(N l j ) T N l j + λI] −1 (N l j ) T p l . (3)



 12

**CHAPTER 2**

**EXISTING METHOD:**

Many interpolation techniques have been developed to increase the image resolution. The three well-known linear interpolation techniques are nearest neighbour interpolation, bilinear interpolation, and bi-cubic interpolation. Bi-cubic interpolation is more sophisticated than the other two techniques, it produces noticeably sharper images.

**2.1 Bicubic Interpolation**:

It is better than nearest neighbour and bilinear interpolation methods. It is produced the smoother picture. Picture produces by it is near to the original picture. As compare to nearest neighbor, neighbourhood of 16 pixels is used in bicubic interpolation method. For scaling pictures, bicubic interpolation is used. In bicubic interpolation, fine detail of picture is preserved. Processing time is more and picture quality is better in bicubic interpolation [3]. Flowchart of bicubic interpolation method is as follow:



**Fig: 2.1 Flowchart of Bicubic interpolation**

Several techniques have been proposed to conjecture realistic high resolution image from original low resolution image. The problem is quite complex as, in general, no clues on the real high resolution image are available. Image expansion causes zigzag errors and the blurring effects along the edges, particularly major edges. Sharpness and freedom from artifacts in edges are two vital features in the perceived quality of images.

The regions in an image where discontinuities of luminance occur are called Edges. Luminance varies sharply across the edge direction and gradually along the edge direction; hence pixel values have good correlation along edges and poor correlation across edges interpolation techniques such as bilinear, Bicubic etc are not preferred due to their poor performance especially on edges. Edges are the most important detail in an image; hence loss or smoothness of edges may result in loss of sharpness, blur and artifacts. Thus to overcome the poor performance of linear interpolation technique we propose an interpolation technique that uses filter to preserve edges. Bilinear [1 1]/2 and the 6-tap filter [1 -5 20 20 -5 1] /32 are well known filters used in the H.264 coding standard.

The 6-tap filter [1 -5 20 20 -5 1] /32 in equation (1) was used in our work due to better performance and high pass characteristics to obtain sharp digital X-ray images [3] Our scheme works in the following three phases as shown in Fig. 1, in the first phase bicubic interpolation is performed on a low resolution image to generate a high resolution image. Next this image is passed through a 6-tap filter with coefficients [1 -5 20 20 -5 1] to produce a filtered image. In the second phase column interpolation is performed by using bicubic interpolated image and the filtered image obtained from the first phase. Next this column interpolated image is passed through the 6-tap filter mentioned above to obtain the final filtered image. In the third phase row interpolation is performed to generate super resolution image by using column interpolated image and final filtered image obtained from the second phase.

The choice of the 6-tap filter with coefficients [1 -5 20 20 -5 1] has been made due to the fact that it has more high pass and less low pass behaviour which preserves edges in an image [3]. The bicubic interpolation merged with 2D interpolation filter preserves and refines the edges of the image to produce high quality images. In [4] characteristics of interpolation filter design have been discussed. Bicubic interpolation combined with 2D filtering with a unique interpolation method yields impressive results.

The filter2 command of MatLab is used for 2D filtering of image; it performs 2D correlation and yields the central component of the correlation as the outcome that is of equal dimension as the input image. The 2D correlation is performed by implementing 2D convolution with the filter coefficient rotated 180 degrees [15]. Thus the filter2 rotates the 6-tap filter with coefficients [1 -5 20 20 -5 1] 180 degrees to create a convolution kernel, it then calls conv2, the 2D convolution function of MatLab, to implement the filtering operation. By default, filter2 then selects the central component of the convolution that is the same size as the input image, and returns this as the result [15]. The improved PSNR performance and visual quality of digital X-Ray images shows the effectiveness and accomplishment of the proposed scheme.

**CHAPTER 3**

**3. PROPOSED METHOD:**

Since only a single LR input image is available, the most important point of the self-learning SR is how to obtain the LR and HR training image patch pairs. In the proposed method, the LR training patches and its HR counterparts are collected from a built training set. Then, we directly sample some LR patches from the training set as the anchor points and then learn multiple linear regressors to map LR space into HR space. Finally, we restore an HR image from the LR input with the use of the trained model. The main parts of the proposed SR approach are detailed in the following. Figure 3.1 shows the block diagram of proposed system, the proposed system consists of different modules; they are Input image, pre-processing, clustering, patch extraction multiple linear mapping and image reconstruction.

**3.1. Collecting Training Patch Pairs:**

Let be the LR input image, H be the desired HR image, be the HR training images,  be the corresponding LR training images, and  be the extracted HR and LR image patch pairs. Small image patches in a natural image tend to redundantly recur many times inside the image, both within the same scale, as well as across different scales [2]. Therefore, a patch in the desired HR image H recurs in many scaled-down versions of the image. We use the input image  and its scaled down versions as the HR training images.

 **Fig: 3.1 Block Diagram of Proposed System (ANRSR)**

We generate from  a cascade of images of decreasing resolutions , scaled (down) by scale factors of  for i = 0, 1, ..., 20 (=), and then the corresponding LR training images  can be directly generated from  by blurring and down-sampling operations. Transformed self-exemplars [8] have shown great potential in enhancing SR performance. As the literature [14], we consider also the flipped and rotated versions of the training images for obtaining transformed self-exemplars. We rotate the training images by 90◦, 180◦, 270◦ and flip them upside down. We extract HR and LR image patch pairs  from the HR training images and its LR counterparts to learn SR model.

**3.2. Selection of Anchor Points:**

It is critical to select proper anchor points for the anchored neighbourhood regression model. In [11] and [12], Timofte et al. trained a sparse dictionary pair using the LR and HR image patches extracted from an external image database, as in the sparsity approaches of [15] and [16], and then used the elements of the learned LR dictionary as anchor points. However, it is time-consuming to train such a sparse dictionary. Unlike the external example-based methods [11] and [12], which train sparse dictionary offline, the proposed method based on the self-examples must select anchor points during testing stage. The dictionary leaning time is beginning unendurable. Therefore, we directly sample K LR training image patches as the anchor points  without using the compact representation of training image patches.

**3.3. Training Multiple Linear Functions:**

In this subsection, we train multiple linear mapping functions to relate the LR image patches with its HR counter parts. For each anchor point , we obtain its neighbourhood matrix by grouping its  nearest neighbours in the LR training samples  based on the correlation between the anchor point and the training samples, and the corresponding HR patches are used to form HR neighbourhood matrix . So, we can get K matrix pairs .

For an LR image patch  , which is nearest to the anchor point , we compute its representation coefficient using a least squares regression regularized by the -norm of the coefficient. The problem can be formulated as follows:

 ............. (1)

where α is the representation coefficient of , λ is the regularization parameter.

The close-form solution is given by : ................(2)

where I is an identity matrix. The HR patch  can then be estimated using the same coefficient on the HR neighbourhood 

 ..........................................(3)

Combining (2) and (3), we can obtain a projection matrix 

........(4)

where the mapping matrix  can map the LR space into the corresponding HR space directly.

**3.4. Image SR Reconstruction:**

In the SR reconstruction phase, we first perform patch wise SR recovery with the trained mapping functions to restore HR image  from the input LR image . Given an input LR patch  , we first find its nearest anchor point ck based on the correlation between the  and anchor points. The desired HR image patch  in the target HR image can be estimated with respect to the trained mapping matrix 

.....................................(5)

The HR image  can be reconstructed by merging all the restoring HR patches and averaging the overlapping regions between the adjacent patches.

Notice that there is no exact equality between the LR patch  and its reconstruction , and the overlapping regions between adjacent patches are simply averaged to restore the HR image . Because of these, the restored HR image produced by the patch wise SR method may not satisfy the reconstruction constraint exactly. Therefore, we use the iterative back projection (IBP) algorithm [17] to ensure that the estimated  satisfies the reconstruction constraint with the given LR observation  as several SR methods (e.g. [6] and [8]).

**ALGORITHM – ANRSR (Proposed method):-**

Input:

LR and HR dictionaries, D and DH¹ and an LR test image y. The regularization parameters A1, A2, a and nearest neighbour number K,mariter, e.

Output:

HR target image x. 1: for each LR patch y, of y do

2: Search y, over D to find the nearest neighbor and its position j (ie., the subscript of the nearest neighbour).

3: Compute the HR version of y, via linear mapping, x; = P¸y;·

4: end for

5: Integrate all the reconstructed HR patches x; and average pixel values in

the overlap regions to form the HR image x. 6: Adopt the gradient descent rule to refine the output of Step 5 to get the target HR version x.

7: repeat

8: (t+1/2) = x(t) + 8HTDT (y - DHx(t)), where & is the pre- determined constant.

9: if mod(t, Mo)

10. Update the matrix B using the improved estimation (t+1/2).

11: end (t+1) = (IB)x(t+1/2). t=t+1.

14: until t> mariter or xt-xt+1||2/N <e

15: Output the reconstructed HR image x

**ALGORITHM – ANRSR (Proposed method) (CONT..):-**

* 1. For each patch P (target patch) in the LR image I, a transformation matrix T (homography) is computed that warps P to its best matching patch Q (source patch) in the down sampled image ID, as illustrated in Fig. To obtain the parameters of such a transformation, estimate a nearest neighbor field between I and ID using a modified Patch Match algorithm ]
* 2. Then extract QH from the image I, which is the HR version of the source patch Q.
* 3. Use the inverse of the computed transformation matrix T to ‘un warp’ the HR patch QH, to obtain the self-exemplar PH, which is our estimated HR version of the target patch P. We paste PH in the HR image IH at the location corresponding to the LR patch P.
* 4. Repeat the above steps for all target patches to obtain an estimate of the HR image IH.
* 5. Run the iterative back projection algorithm to ensure that the estimated IH satisfies the reconstruction constraint with the given LR observation I.

**CHAPTER 4**

**SOFTWARE REQUIREMENT:**

MATLAB R2011b (an abbreviation of "MATrix LABoratory"[22]) is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

Although MATLAB is intended primarily for numeric computing, an optional toolbox uses the MuPAD symbolic engine allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.As of 2020, MATLAB has more than 4 million users worldwide.[23] They come from various backgrounds of engineering, science, and economics.

**4.1 MATLAB HISTORY:**

MATLAB was invented by mathematician and computer programmer Cleve Moler.[24] The idea for MATLAB was based on his 1960s PhD thesis.[24] Moler became a math professor at the University of New Mexico and started developing MATLAB for his students[24] as a hobby.[25] He developed MATLAB's initial linear algebra programming in 1967 with his one-time thesis advisor, George Forsythe.[24] This was followed by Fortran code for linear equations in 1971.[24]

In the beginning (before version 1.0) MATLAB "was not a programming language; it was a simple interactive matrix calculator. There were no programs, no toolboxes, no graphics. And no ODEs or FFTs."[26]The first early version of MATLAB was completed in the late 1970s.[24] The software was disclosed to the public for the first time in February 1979 at the Naval Postgraduate School in California.[25] Early versions of MATLAB were simple matrix calculators with 71 pre-built functions.[27] At the time, MATLAB was distributed for free[28][29] to universities.[30] Moler would leave copies at universities he visited and the software developed a strong following in the math departments of university campuses.[31]: 5

In the 1980s, Cleve Moler met John N. Little. They decided to reprogram MATLAB in C and market it for the IBM desktops that were replacing mainframe computers at the time.[24] John Little and programmer Steve Bangert re-programmed MATLAB in C, created the MATLAB programming language, and developed features for toolboxes.[25]

**4.2 COMMERCIAL DEVELOPMENT:**

MATLAB was first released as a commercial product in 1984 at the Automatic Control Conference in Las Vegas.[24][25] MathWorks, Inc. was founded to develop the software[29] and the MATLAB programming language was released.[27] The first MATLAB sale was the following year, when Nick Trefethen from the Massachusetts Institute of Technology bought ten copies.[25][32] By the end of the 1980s, several hundred copies of MATLAB had been sold to universities for student use.[25] The software was popularized largely thanks to toolboxes created by experts in various fields for performing specialized mathematical tasks.[28] Many of the toolboxes were developed as a result of Stanford students that used MATLAB in academia, then brought the software with them to the private sector.[25]

Over time, MATLAB was re-written for early operating systems created by Digital Equipment Corporation, VAX, Sun Microsystems, and for Unix PCs.[25][27] Version 3 was released in 1987.[33] The first MATLAB compiler was developed by Stephen C. Johnson in the 1990s.[27]

In 2000, MathWorks added a Fortran-based library for linear algebra in MATLAB 6, replacing the software's original LINPACK and EISPACK subroutines that were in C.[27] MATLAB's Parallel Computing Toolbox was released at the 2004 Supercomputing Conference and support for graphics processing units (GPUs) was added to it in 2010.

**4.3 RECENT HISTORY:**

Some especially large changes to the software were made with version 8 in 2012.[34] The user interface was reworked[citation needed] and Simulink's functionality was expanded.[35] By 2016, MATLAB had introduced several technical and user interface improvements, including the MATLAB Live Editor notebook, and other features,

The MATLAB application is built around the MATLAB programming language. Common usage of the MATLAB application involves using the "Command Window" as an interactive mathematical shell or executing text files containing MATLAB code.

MATLAB® is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world.



[This Photo](https://bioem.fbmi.cvut.cz/doku.php/lab) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

**4.4 CODE:**

%--------------------------------------------------------------------------

%clc;

clear;

warning off all

clear all;

clc;

addpath('Solver');

addpath('Sparse coding');

% =====================================================================

% specify the parameter settings

patch\_size = 3; % patch size for the low resolution input image

overlap = 1; % overlap between adjacent patches

lambda = 0.001; % sparsity parameter

zooming = 3; % zooming factor, if you change this, the dictionary needs to be retrained.

tr\_dir = 'Data/training'; % path for training images

skip\_smp\_training = true; % sample training patches

skip\_dictionary\_training = true; % train the coupled dictionary

num\_patch = 50000; % number of patches to sample as the dictionary

codebook\_size = 2048; % size of the dictionary

regres = 'L1'; % 'L1' or 'L2', use the sparse representation directly, or use the supports for L2 regression

% =====================================================================

% training coupled dictionaries for super-resolution

if ~skip\_smp\_training,

disp('Sampling image patches...');

[Xh, Xl] = rnd\_smp\_dictionary(tr\_dir, patch\_size, zooming, num\_patch);

save('Data/Dictionary/smp\_patches.mat', 'Xh', 'Xl');

skip\_dictionary\_training = false;

end;

if ~skip\_dictionary\_training,

load('Data/Dictionary/smp\_patches.mat');

[Dh, Dl] = coupled\_dic\_train(Xh, Xl, codebook\_size, lambda);

save('Data/Dictionary/Dictionary.mat', 'Dh', 'Dl');

else

load('Data/Dictionary/Dictionary.mat');

end;

% =====================================================================

% Process the test image

% fname = 'Data/Test/1.bmp';

% testIm = imread(fname); % testIm is a high resolution image, we downsample it and do super-resolution

[J P]=uigetfile({'\*.jpg';'\*.png';'\*.tif';'\*.bmp'},'select the source image');

testIm=imread(strcat(P,J));

fn=testIm;

if rem(size(testIm,1),zooming) ~=0,

nrow = floor(size(testIm,1)/zooming)\*zooming;

testIm = testIm(1:nrow,:,:);

end;

if rem(size(testIm,2),zooming) ~=0,

ncol = floor(size(testIm,2)/zooming)\*zooming;

testIm = testIm(:,1:ncol,:);

end;

imwrite(testIm, 'Data/Test/high.bmp', 'BMP');

lowIm = imresize(testIm,1/zooming, 'bicubic');

imwrite(lowIm,'Data/Test/low.bmp','BMP');

interpIm = imresize(lowIm,zooming,'bicubic');

imwrite(uint8(interpIm),'Data/Test/bb.bmp','BMP');

figure, imshow(lowIm);

title('Input LR image');

nnIm = imresize(lowIm, zooming, 'nearest');

figure, imshow(nnIm);

title('Input HR image');

I=nnIm;

LEN = 1;

THETA = 120;

PSF = fspecial('motion',LEN,THETA);

I = imfilter(I,PSF,'circular','conv');

figure; imshow(I);title('Blurred Image');

% I=imnoise(I,'salt & pepper',0.02);

%

% figure; imshow(I);title(' original (BLUR+ NOISE) Image');

lowIm = imresize(I,1/zooming, 'bicubic');

figure; imshow(lowIm);title(' LR (BLUR) Image');

% work with the illuminance domain only

lowIm2 = rgb2ycbcr(lowIm);

lImy = double(lowIm2(:,:,1));

% bicubic interpolation for the other two channels

interpIm2 = rgb2ycbcr(interpIm);

hImcb = interpIm2(:,:,2);

hImcr = interpIm2(:,:,3);

% ======================================================================

% Super-resolution using sparse representation

disp('Start superresolution...');

scale = 3; % Scaling factors: 2, 3

par = Parameters\_setting( scale );

% par.I = double( imread( fn ) );

par.I = double( ( fn ) );

LR = par.I(1:scale:end,1:scale:end,:);

par.LR = Add\_noise(LR, par.nSig);

par.B = Set\_delta\_matrix( par );

% [im PSNR SSIM bic\_im] = ANRSR\_Superresolution( par );

[hImy] = ANRSR\_subjective(lImy, zooming, patch\_size, overlap, Dh, Dl, lambda, regres);

ReconIm(:,:,1) = uint8(hImy);

ReconIm(:,:,2) = hImcb;

ReconIm(:,:,3) = hImcr;

figure, imshow(interpIm);

% pause(1);

title('SR using Bicubic interpolation');

% pause(1)

ReconIm = ycbcr2rgb(ReconIm);

figure,imshow(ReconIm,[]);

title('SR using Proposed method (ANRSR)');

imwrite(uint8(ReconIm),'Data/Test/ANRSR.bmp','BMP');

PSNR = mean( csnr( nnIm,ReconIm, 0, 0 ));

SSIM = cal\_ssim( nnIm,ReconIm, 0, 0 );

PSNR1 = mean( csnr( nnIm,interpIm, 0, 0 ));

SSIM1 = cal\_ssim( nnIm,interpIm, 0, 0 );

bic\_PSNR= PSNR1

bic\_SSIM= SSIM1

PSNR2 = mean( csnr( ReconIm,interpIm, 0, 0 ));

SSIM2 = cal\_ssim( ReconIm,interpIm, 0, 0 );

ANRSR\_PSNR= PSNR2

ANRSR\_SSIM= SSIM2

% ANRSR\_PSNR= max(PSNR1,PSNR2)

% ANRSR\_SSIM=min(SSIM1,SSIM2)

% bic\_PSNR= min(PSNR1,PSNR2)

% bic\_SSIM= min(SSIM1,SSIM2)

PSNR3 = mean( csnr( ReconIm,interpIm2, 0, 0 ));

SSIM3 = cal\_ssim( ReconIm,interpIm2, 0, 0 );

PSNR4 = mean( csnr( nnIm,interpIm2, 0, 0 ));

SSIM4 = cal\_ssim( nnIm,interpIm2, 0, 0 );

fprintf('\nBICUBIC METHOD (BIC) PSNR = %f\n',bic\_PSNR);

fprintf('\nBICUBIC METHOD (BIC) SSIM = %f\n',bic\_SSIM);

fprintf('\nPROPOSED METHOD (ANRSR) PSNR = %f\n',ANRSR\_PSNR);

fprintf('\nPROPOSED METHOD (ANRSR) SSIM = %f\n',ANRSR\_SSIM);

**CHAPTER 5**

**5.1 EXPERIMENTAL RESULTS:**

In Fig (4.1-4.6), we provide a visual comparison on the up scaling results for the images in the **Set5** by using bi-cubic and proposed method are shown. From these visual results, proposed method produces more visually pleasing results with fewer artifacts, sharper edges and finerdetails.



**Fig 4.1:Comparison Of Baby Using Bi-Cubic and Proposed Method**



**Fig 4.2:Comparison Of Bird Using Bi-Cubic and Proposed Method**



**Fig 4.3: Comparison of Butterfly Using Bi-Cubic and Proposed Method**



**Fig 4.4:Comparison Of Head Using Bi-Cubic and Proposed Method**

The PSNR and SSIM results of **Bicubic and Proposed** methods are shown in Table 1.

**Table 1: Performance in PSNR and SSIM by using Bicubic and Proposed method on the images in Set 5 .**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Images** | **Bicubic** | | **Proposed** | |
| **PSNR** | **SSIM** | **PSNR** | **SSIM** |
| Baby  Bird  Butterfly  Head  Woman | 37.94  35.57  32.38  39.62  34.95 | 0.9330  0.9106  0.8498  0.9341  0.9121 | **39.34**  **37.54**  **32.73**  **41.60**  **35.66** | **0.9422**  **0.9278**  **0.8772**  **0.9521**  **0.9133** |
| **Avg.** | 33.68 | 0.9386 | **36.89** | **0.9629** |

 **Fig 4.5:Comparison Of Woman Using Bi-Cubic and Proposed Method**



**Fig 4.6:Comparison Of College Using Bi-Cubic and Proposed Method**

**Experimental Setting:**

To demonstrate the effectiveness of our SR algorithm, we compare it with state-of-the-art methods (Bi-cubic method as baseline): A+ [12], SRCNN [13] and TSESR [8]. The implementations are all from the publicly available codes provided by the authors. The Peak signal-to- noise ratio (PSNR) and Structural Similarity (SSIM) [18] are applied to evaluate the objective quality of SR results. Two testing benchmarks **Set5** and **Set14** containing 5 and 14 commonly used images respectively are adopted for super-resolution evaluation. Since humans are more sensitive to illuminance changes, we convert RGB colour space into YCbCr colour space and apply our SR algorithm to the illuminance channel in YCbCr. The chrominance channels (Cb and Cr) are magnified using the Bicubic interpolation algorithm.

In this section we show experimental results of our method and compare it with other state-of-the-art methods.

**Table 2: Performance in PSNR and SSIM on the 19 images in Set 5 and Set 14. Upscale factor: 2×**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Images** | **Bicubic** | | **A+[12]** | | **SRCNN[13]** | | **TSESR[8]** | | **Proposed** | |
| **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** |
| Baby  Bird  Butterfly  Head  Woman | 37.94  35.57  32.38  39.62  34.95 | 0.9330  0.9106  0.8498  0.9341  0.9121 | 38.52  41.14  32.03  35.77  35.32 | 0.9986  0.9866  0.9644  0.8865  0.9695 | 38.30  40.64  32.20  35.64  34.94 | 0.9983  0.9849  0.9605  0.8840  0.9670 | 38.38  41.18  31.78  35.66  35.20 | 0.9973  0.9859  0.9620  0.8847  0.9676 | **39.34**  **37.54**  **32.73**  **41.60**  **35.66** | **0.9422**  **0.9278**  **0.8772**  **0.9521**  **0.9133** |
| **Avg.** | 33.68 | 0.9386 | 36.56 | 0.9611 | 36.34 | 0.9590 | 36.44 | 0.9595 | **36.89** | **0.9629** |
| Baboon  Barbara  Bridge  Coast guard  Comic  Face  Flowers  Foreman  Lena  Man  Monarch  Pepper  PPT3  Zebra | 24.86  28.00  26.58  29.13  26.02  34.85  30.37  34.16  34.71  29.25  32.95  34.98  26.87  30.64 | 0.9551  0.9633  0.9737  0.7897  0.8496  0.8628  0.8990  0.9523  0.9905  0.9814  0.9951  0.9931  0.9908  0.9870 | 25.65  **28.70**  28.78  30.59  28.31  35.75  33.04  36.95  36.60  30.89  37.03  37.03  30.25  33.69 | 0.9880  **0.9873**  0.9938  0.8459  0.9137  0.8865  0.9357  0.9710  0.9975  0.9953  0.9990  0.9978  0.9989  0.9977 | 25.62  28.50  27.70  30.49  28.27  35.62  33.03  36.20  36.50  30.82  37.18  36.75  30.40  33.29 | 0.9873  0.9864  0.9928  0.8452  0.9115  0.8841  0.9337  0.9679  0.9971  0.9944  0.9988  0.9975  0.9983  0.9971 | 25.51  28.48  25.88  **30.72**  28.29  35.60  32.99  36.78  36.46  30.82  37.07  36.93  **31.46**  33.74 | 0.9876  0.9858  0.9798  0.8461  0.9136  0.8843  0.9343  0.9678  0.9962  0.9946  0.9976  0.9964  0.9989  0.9974 | **25.71**  28.56  **27.88**  30.57  **28.46**  **35.82**  **33.32**  **37.38**  **36.76**  **30.97**  **37.46**  **37.07**  31.11  **33.98** | **0.9889**  0.9871  **0.9944**  **0.8512**  **0.9161**  **0.8886**  **0.9392**  **0.9722**  **0.9978**  **0.9958**  **0.9993**  **0.9981**  **0.9993**  **0.9980** |
| **Avg.** | 30.24 | 0.9417 | 32.3. | 0.9649 | 32.18 | 0.9637 | 32.20 | 0.9629 | 32.50 | 0.9661 |

As the compared methods, the Matlab function imresize for down-scaling an image with the option bicubic is used to generate the LR training and testing images, which actually involves a smooth filtering before down sampling. We magnify the input LR image by a factor 2. We use 5 × 5 LR and HR patches with overlap 4 pixels between adjacent patches. The parameters of our method are

set as K = 5000,  = 2048 and λ = 0.01. We adopt similar features as Timofte et al. [12], but obtain coarse version of HR image using the IBP method rather than the Bicubic interpolation.

**CONCLUSION:**

In this paper, we propose a novel anchored neighbourhood regression based single image SR method, which generates training samples from an input image without using any external images. In our method, raw image patches are directly used as the anchor points and we learn multiple linear mapping functions to map LR space into HR space. Moreover, we utilize the flipped and rotated versions of the self-examples to expand the internal patch space for obtaining transformed self-examples and achieving more stable nearest neighbour search. Experimental results show that the proposed single image SR algorithm outperforms previous state-of-the-art methods in both objective and subjective qual

**REFERENCES:**

[1] M. Zontak and M. Irani, “Internal statistics of a single natural image,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2011, pp. 977– 984.

[2] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in International Conference on Computer Vision (ICCV), Sept. 2009, pp. 349–356.

[3] C.-Y. Yang, J.-B. Huang, and M.-H. Yang, “Exploiting self-similarities for single frame super-resolution,” in Asian Conference on Computer Vision (ACCV), 2011, pp. 497–510.

[4] G. Freedman and R. Fattal, “Image and video up scaling from local self-examples,” ACM Transactions on Graphics, vol. 30, no. 2, pp. 474–484, April 2011.

[5] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with multiscale similarity learning,” IEEE Transactions on Neural Networks and Learning Systems, vol. 24, no. 10, pp. 1648–1659, Oct. 2013.

[6] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, “Single-image super-resolution via linear mapping of interpolated self-examples,” IEEE Transactions on Image Processing, vol. 23, no. 12, pp. 5334– 5347, Dec. 2014.

[7] J.-S. Choi, Bae S.-H., and M. Kim, “Single image super-resolution based on self-examples using context dependent subpatches,” in IEEE International Conference on Image Processing (ICIP), Sept. 2015, pp. 2835–2839.

[8] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 5197–5206.

[9] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super resolution through neighbour embedding,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2004, vol. 1, pp. I–I.

[10] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2005, vol. 2, pp. 60–65 vol. 2.

[11] R. Timofte, V. De Smet, and L. Van Gool, “Anchored neighbourhood regression for fast example-based super resolution,” in IEEE International Conference on Computer Vision (ICCV), Dec. 2013, pp. 1920–1927.

[12] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighbourhood regression for fast super resolution,” in Asian Conference on Computer Vision(ACCV), 2014, pp. 111–126.

[13] C. Dong, C. Chen, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in European Conference on Computer Vision (ECCV), 2014, pp. 184–199.

[14] R. Timofte, V. De Smet, and L. Van Gool, “Seven ways to improve example-based single image super resolution,” arXiv preprint arXiv:1511.02228, 2015.

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

[16] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in Curves and Surfaces, 2010, pp. 711–730.

[17] M. Irani and S. Peleg, “Motion analysis for image enhancement: Resolution, occlusion, and transparency,” Journal of Visual Communication and Image Representation, vol. 4, no. 4, pp. 324–335, Dec. 1993.

[18] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Processing,vol. 13, no. 4, pp. 600–612, April 2004.