SUPER-RESOLUTION USING DEEP LEARNING

A Thesis Submitted in Fulfillment of the Requirement for the Award of the Degree of

MASTER OF ENGINEERING

In

Electronics and Communication

Submitted By

Shailza Sharma

801661022

Under Supervision of

Dr. Vinay Kumar

Assistant Professor

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

JULY, 2018

DECLARATION

I, Shailza Sharma hereby declare that the work presented in this thesis entitled " **Super-Resolution using Deep Learning** " in fulfillment of the requirement for the award of degree of Master of Engineering (ECE) submitted at Department of Electronics and Communication, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of Dr. Vinay Kumar (Assistant Professor, Department of Electronics and Communication, Thapar Institute of Engineering & Technology (Deemed to be University) from July 2016 to July 2018. The matter presented in this has not been submitted either in part or full to any other university or institute for the award of any other degree.

Shailza

Date: 27 August, 2018

(Shailza Sharma)

(801661022)

Dr.Vinay Kumar

(Assistant Professor)

Electronics And Communication Engineering Department

Thapar Institute Of Engineering & Technology

(A Deemed To Be University), Patiala, Punjab

Date: 27 August, 2018

ACKNOWLEDGEMENT

I would like to convey my deep sense of gratitude to my project guide, **Dr. Vinay Kumar**, **Assistant Professo**r, **ECED** who is a constant source of motivation and firm support in carrying out this project. The support and supervision that he gave has helped me to progress in the project. His cooperation is highly appreciated and I highly oblige to him for his valuable comments and moral support during this research period. I value his concern and support at all times, good and bad. He has always emphasis on self-motivation during rough or bad periods and appreciated in good days. The words are not enough to thank him.

I am also thankful to Thapar University for the facilities and healthy environment for study. I also express my sincere thanks to my Head of the Department, **Dr. Alpana Agarwal** for providing me adequate environment in carrying the work.

A big thanks to my friends for their support in accomplishment of my course work. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

Above all I thank the Almighty God who is being with me and showers his blessings and his grace towards me in all walks of my life.

Shailza Sharma

ME-ECE

801661022

TABLE OF CONTENTS

ABSTRACT

Digital images have a broad range of applications such as remote sensing, image and data storage for transmission in business applications, medical imaging, acoustic imaging, forensic sciences and industrial automation. Images acquired by satellites are useful in tracking of earth resources, geographical mapping, and prediction of agricultural crops, urban population, weather forecasting, flood and fire control. Space imaging applications include recognition and analyzation of objects contained in images obtained from deep space-probe missions. There are also medical applications such as processing of X-Rays, Ultrasonic scanning, Electron micrographs, Magnetic Resonance Imaging, Nuclear Magnetic Resonance Imaging, etc. If these images are degraded, blurred or not captured accurately, then we are not able to find these images wide variety of applications. So, we need to enhance quality of these images.

The proposed method for enhancing these types of images in this report is convolutional neural networks. CNN is a mathematical model or computational model that tries to simulate the structure and functional aspects of biological neural networks. CNN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. CNN are adjusted or trained to a specific target output which is based on a comparison of the output and the target, until the network output matches the target.

LIST OF ABBREVIATIONS

DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
MLP	Multilayer Perceptron
DBM	Deep Belief Machine
AE	Auto Encoder
SAE	Sparse Auto Encoder
RBM	Restricted Boltzmann Machines
SR	Super-Resolution
LR	Low-Resolution
HR	High-Resolution
SISR	Single-Image Super-Resolution
MISR	Multi-Image Super-Resolution

List of Figures

List of Tables

CHAPTER-1
INTRODUCTION

1.1 MOTIVATION

To analyse images for different purposes such as in medical applications: processing an X-ray image, MRI etc., in military applications: for detecting, recognizing and tracking large number of targets, in remote sensing and in acoustic imaging; there is a need of images which are having very high-resolution. But if these images are blurred, degraded and not captured properly then it is not possible to find their applications in above mentioned fields. Also, for later image processing, images with high-resolution are desired. But in real life scenario, to capture an image with high-resolution is very tough**.** There are three possible ways of improving image resolution:

1. By increasing sensor density,
2. Increasing chip size of the capturing device, and
3. Super resolution.

Increasing the sensor density leads to decrease in the sensor size which gives rise to other problem i.e. the amount of light that incidents on every sensor will decrease and this will introduce noise in an image which is known as shot noise. In most of the real-life and practical applications, to construct image chips for imaging system and the optical components for capturing images with high resolution is very expensive. The high resolution for the surveillance camera is not only limited by the cost but also by the speed of camera and the hardware storage. So first two methods are not preferred since they lead to increased cost of device and higher computational complexity.

A better way of getting a high-resolution image from a low-resolution image is to capture a low-resolution image by a image acquisition device, do some processing on the captured low-resolution image and get a image with high-resolution. To estimate or to recover missing high-resolution (HR) details from the low-resolution (LR) image is known as Super Resolution of an image. And this method is preferred over the other two methods because of low cost and easy implementation.

1.2 OBJECTIVE OF THE THESIS

Convolutional neural networks (CNN) are achieving great success in image super resolution. Dong et al. [1] proposed a method to achieve SR using CNN which learns direct end-to-end mapping between low resolution image and high-resolution images. But the computational complexity of the approach is very high; since input image is up sampled to desired output image by bicubic interpolation before passing it to the CNN network. Dong et al. [2] proposed other network where a deconvolution layer is introduced at the end of the network to make the size of input and output images equal. By introducing this network computational complexity decreases and speed of the network increases as compare to the previous architecture. But the PSNR value for this architecture is very low and visual quality of the output images is very poor. Wang et al. [3] proposed other network for SR known as ESPCNN; i.e., efficient sub-pixel convolutional neural network, where sub-pixel layer is introduced at the end of the network to make the size of the input image equal to the desired output image. It reduces the computational complexity as compare to SRCNN. But this architecture has also some disadvantages like network is not very deep that means less number of convolution layers are used. High frequency components are missing. So, our objective is to develop a CNN architecture that overcomes all the drawbacks discussed above

1.3 CONTRIBUTION TO THE PROBLEM STATEMENT

Two CNN architectures are proposed to increase the efficiency and resolution of final image.

- First CNN architecture uses two sub-pixel layers followed by convolution layers instead of one sub-pixel layer at the end of the CNN network (compared to ESPCNN [3]). The change produces better results than ESPCNN. Because our network is deep, as the number of convolution layers are more in our network. So, the capability of our network to learn feature maps increases.

- Second CNN architecture introduces residual learning to our network to preserve the high-frequency details of an image.

- We have created a dataset from pascal VOC dataset. We pseudo-randomly selected 4000 images from the dataset (random selection from the subset consisting of high variation data). Out of these 4000 images, 3000 images are used for the training purpose and 1000 images are used for the testing purpose. These images are better than the images used in previous papers because these images are based on real life data. So, variation of frequency is high in these images as these images have data ranging from low-frequency to high-frequency.

1.4 OUTLINE OF THE THESIS

In chapter 2, we have discussed about the basics of Super-Resolution, Machine learning and its types, artificial neural networks and different categories falls under artificial neural networks. In chapter 3, literature review of super resolution techniques is done. Super-Resolution techniques are divided into three categories that are interpolation based, reconstruction based and example-based. And then we have discussed each of these three methods in details. In chapter 4, we have discussed how Convolution Neural Networks are used for purpose of Super-resolution. Also, brief overview is given about the loss functions and optimizers used in the reconstruction purpose. Introduction to Residual-Learning is also given. In chapter 5, we have discussed the proposed methodology in which we have introduced two CNN architectures to convert a low-resolution image to the high-resolution image. Implementation of these two architectures with the help of equations are discussed in detail. In chapter 6, we have evaluated our model performance on different datasets. Then we have compared our model performance with the other state-of-the-art methods. Chapter 7 gives the conclusion of the thesis and future scope for the work done.

CHAPTER -2

2.1 IMAGE ENHANCEMENT

The main goal of enhancing images is to process the image so that we can get the result which is more applicative than our original image. It means that we are processing an image so that we can improve the quality of image by doing manipulation on the pixels. It emphasis on the features of an image for example the edges, contrast or the boundaries so the graphic display becomes more significant for analysis and displaying the images.

Enhancing images does not mean that we are increasing the built-in content or the data in images, but it means that we are increasing dynamic range of the selected features so that we can be able to detect them easily. Enhancing images makes images favorable by:

1. Accentuating interesting features in images.
2. To Take away the noise from images.
3. To make images visibly more appealing.

Examples of image enhancement includes enhancement of edge and contrast, filtering of noise, to sharp an image or to magnify an image. Enhancement of images is useful in case we need to extract features from an image, analysis of an image, or to display an image.

2.1.1 Techniques for image enhancement

- **Spatial domain**

  In spatial domain there is direct manipulation on the pixels of an image. Transformation of the gray level, Histogram Processing, doing enhancement using arithmetic and logic operations. Computation is more efficient in case of spatial domain techniques for image enhancement and it also need less processing resources for the implementation. Filters which are used for enhancement of images in spatial domain are smoothing and sharpening filters.

- **Frequency domain**

  We need to compute the Fourier transform of an image which is to be enhanced, multiply this result with a filter and then take inverse transform to get the enhanced image. Filters used in frequency domain for enhancing images are Butterworth filters, Laplacian filters, Ideal filters, Gaussian filters and Homomorphic filters.

2.2 SUPER-RESOLUTION

In most of all the digital imaging applications, high resolution images and videos are mainly desired for further processing of the images and their analysis. The need of high image resolution has two main application fields:

- To Improve the picture information for human understanding.
- Helping representation for automatic machine perception.

Image resolution gives the details which are present in an image. Higher the resolution, more will be the details present in an image. The resolution of a digital image can be classified in different ways: pixel resolution, spatial resolution, spectral resolution, temporal resolution, and radiometric resolution. But the main area of interest is spatial resolution.

Spatial resolution: A digital image is composed of small picture elements which are called as pixels. Spatial resolution refers to the density of pixels in an image and it is measured in terms of pixels per unit area.
Imaging sensors and the acquisition devices mainly restrict the spatial resolution of the images. Most of the image sensors in the present day are made up of charge coupled devices (CCD) or by using complementary metal oxide semiconductors (CMOS) active pixel sensors [4]. The arrangement of these sensors is in the form of array which is two-dimensional to record image signals which are also two-dimensional. Captured image's spatial resolution is determined by the size of the sensor or equally the number of elements in the sensor per unit area. For imaging devices, the possibility for more spatial resolution depends upon the density which is present in the sensors i.e. if there is high density present in sensor then there will be more possibility of high spatial resolution.

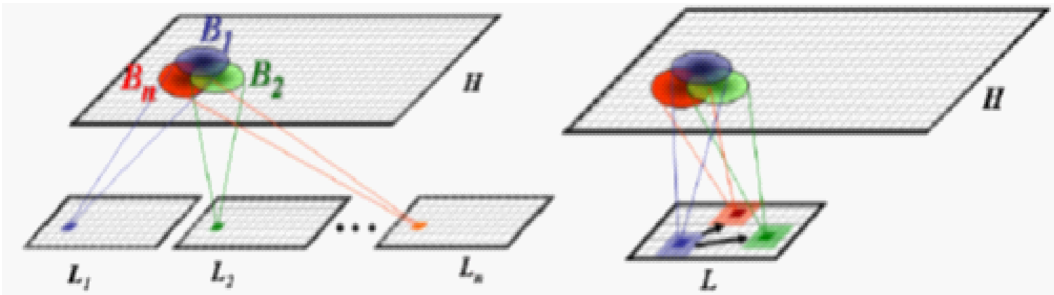
An imaging system having detectors which have insufficient features will produce images with low resolution and high blocky effects and the reason for this is aliasing which occurs by low spatial sampling frequency. One simple way to increase the spatial resolution in the imaging devices is to increase the density of sensors by minimizing the size of sensor. But as we decrease the size of the sensor, the amount of light that incidents on every sensor will also decrease and this will cause noise which is known as shot noise. Also, when we increase the density of sensors or the density of pixels in an image, the hardware cost corresponding to each sensor also increases. Optics is the other factor which restricts the details in an image i.e. the bands with high frequency and the reason for this is blurring of lens which is associated with the point spread function (PSF) of sensor, aberration effects of the lens and the optical blur which is due to the motion. In most of the real and practical applications to construct image chips for imaging system and the optical components for capturing images with high resolution is very expensive for example widely used cameras for surveillance and the cameras in the cell phones. The high resolution for the surveillance camera is not only limited by the cost but also by the speed of camera and the hardware storage. In other case like in satellite imagery, due to the physical constraints, it is very difficult to use sensors with high resolution.

The way to solve this problem is to accept degradations in an image and then use the signal processing for the post processing of the images which has been captured. This method will decrease the computational and the hardware cost. And this method is specifically known as Super-Resolution (SR) reconstruction. The technique used in Super-Resolution is to construct an image with high- resolution (HR) from number of images with low-resolution Therefore, it increases the components with high frequency and remove degradations which are produced by the imaging process of camera having low-resolution. Non-redundant information is combined that is present in various low-resolution frames to produce an image with high-resolution. This is the basic idea behind the Super Resolution (SR) technique.

Other technique which is costly related to the SR technique is single image interpolation, and this approach is also used to increase the size of an image. But as there is no added information provided, the quality of an image by using single image interpolation is very less and the reason for it is ill-posed nature of problem, and the frequency components which are lost can't be recovered. While SR makes the problem better constrained because in SR technique multiple low-resolution frames for reconstruction are available.

Subpixel shifts between the LR images typically introduces the non-redundant information. The reason for these subpixel shifts can be uncontrolled motion between imaging device and the scene for example object's movement or because of controlled motions, e.g., the imaging system in the satellites has predefined path and speed for the orbit of an earth. Every low-resolution image is the decimated, blurred or the aliased observation of the original or the true scene. If subpixel motion exists between the low-resolution frames then only SR is possible, that's why ill posed problem is better conditioned. In the process of imaging, camera records several low- resolution frames, frames are down sampled from high resolution scene with the shifting of subpixels between each other. In super resolution this process is reversed. In this there is alignment of LR observations to accurate subpixel and combining them in high resolution image grid. SR algorithms depends on the number of input and output images used in the process.

When single low-resolution image is used to produce high resolution image, then it is single-image single output (SISO) SR. And in multiple-input single output (MISO), multiple low-resolution frames are used to estimate single high-resolution frame. SR problem is considered as an inverse problem in MISO so in algorithms of multi-frame SR, observation model is made and by taking reverse of that observation model we can get the output image. Due to the fusion stages and high computational complexity during image registration these methods are not that popular, so SISR is preferred over multi-frame super resolution.



(a) Classical Multi-Image SR (b) Single-Image Multi-Patch SR

Figure 2.1 Difference between Multi-Image SR and Single-Image SR [5].

2.3 MACHINE LEARNING

Machine learning gives potential to the computers to learn features and patterns corresponding to the given input and provide desired output without the need of explicit programming. Any program which observes things and on basis of those observations, it is improving its ability to perform the task, then the program is said to be learning. Machine-learning is of great importance in real life scenario. The factors showing machine learning's importance are as follows:

- There are many situations that may occur during the run-time and the program should have ability to handle these situations but sometimes these changes are so rapid that it's not possible to detect these changes with the help of traditional formulas and programing.

- Its very hard to develop a program for the computer which detects objects and patterns in the same way as the human brain does. So, there is need of learning algorithms by which computer learns to detect objects and patterns.

2.4 LEARNING TYPES

Depending on the type of data that is given and the problem statement, machine learning is mainly divided into three categories [6] and every category has advantages as well as disadvantages. In artificial neural network, network's weights are altered in accordance with these learning algorithms. The purpose of this operation is to find the weight matrices set that can be apply to the network so that input can be mapped to the correct output.

2.4.1 Supervised learning

In supervised learning we are given with the desired output. As we are given with both input and desired output, we are in position to calculate the error by using any loss functions. The most common loss function that is used to calculate the error is mean square error (MSE). In mean square error, difference between desired output and the actual output is calculated and then this difference is back propagated and make corrections in the network by updating the weights present in the hidden layers of the network with the help of optimization techniques. This process is repeated till we get an output equivalent to the desired output.

2.4.2 Unsupervised learning

In unsupervised learning, neural network is given with input, but desired output is not specified. So, it is the responsibility of the network to find out some specific pattern for the input provided without the help of any external support. Data mining uses these kind of learning algorithms. Also, encoders use unleveled data to provide output for the given input.

2.4.3 Reinforcement learning

In case of reinforcement leaning, desired output is not given for guidance, but an independent agent is present in the architecture which perform the tasks by learning different patterns with the help of trial and error method. Due to its learning abilities, reinforcement learning is achieving great success in robotics field.

2.5 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are organized in the form of layers. These layers are composed of interconnecting nodes and have some activation function. In the network there is an input layer which communicates with one or more hidden layers where further processing is done using the weights.

2.6 KEY ELEMENTS OF DEEP LEARNING/NEURAL NETWORKS

On basis of construction and training we can categorize deep neural networks:

1. Deep Neural Networks (DNNs)
2. Convolutional Neural Networks (CNNs)
3. Recurrent Neural Networks (RNNs)
4. Emergent architectures

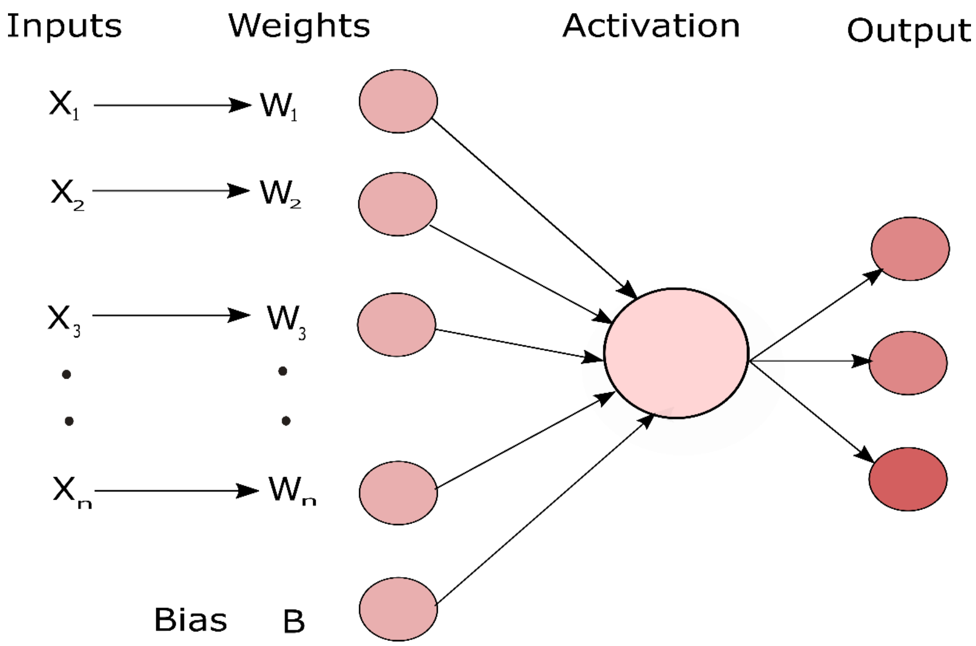


Figure 2.2 General representation of neural networks [7]

2.6.1 Deep Neural Networks

DNNs architecture consists of an input layer, an output layer and in between multiple hidden layers. As the input data is given to this type of network, output value is computed sequentially across the layers. In each layer, weighted sum is produced as each unit in the layer is multiplied by the weight vector to produce the output values. Different nonlinear functions for example sigmoid, rectilinear unit or hyperbolic tangent are applied to weighted sum to calculate the output values of layer. DNNs are further classified as MLP, SAE and DBN [8]

- **MLP:** Multilayer Perceptron (Figure 1.3) has similar structure to the neural networks but it has more number of stacked layers. Supervised learning is done for training the network which uses labeled data. Multilayer Perceptron are used in cases where amount of leveled data is very high.

- **SAE and DBM:** AEs and RBMs are the building blocks for the architecture of SAE (Figure 1.4) and DBN respectively. MLP is different from these two as in both of these blocks, training is carried out in two steps: unsupervised Pre-training and after that fine-tuning is done in supervised manner. So, in first phase of training, we sequentially stacked the layers and then training is done in a layer-wise manner as an Auto Encoder in case of Stack Auto Encoder and as Restricted Boltzmann Machine in case of Deep Belief Network by using unlabeled data. In the second phase i.e. supervised fine-tuning, optimization is done by retraining with the help of the labeled data.

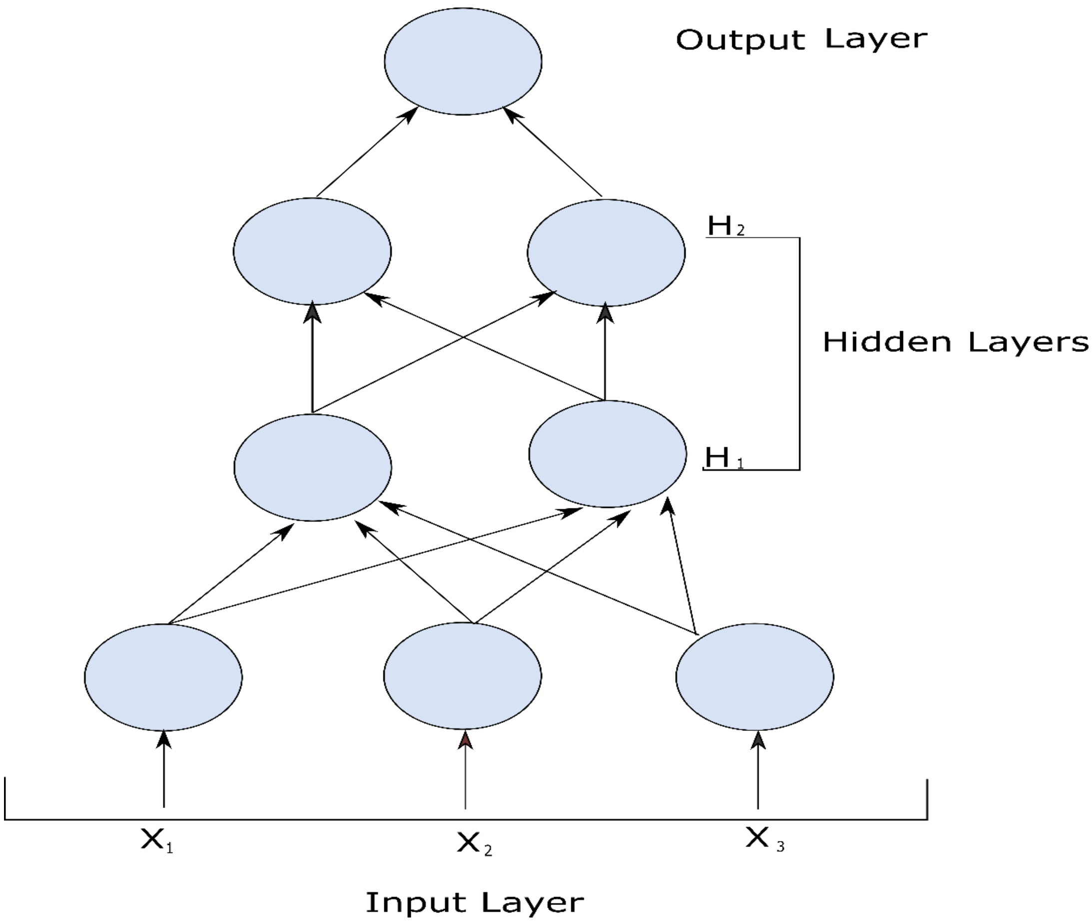


Figure 2.3 Architecture of MLP having X_1,X_2,X_3 as three inputs and H_1,H_2 as two hidden layers [9]

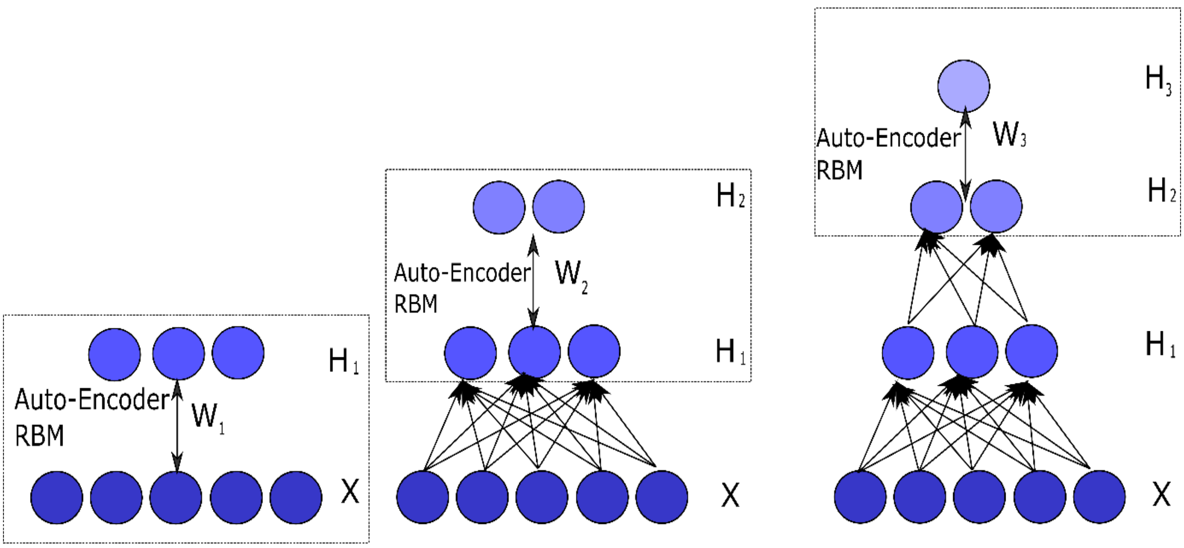


Figure 2. 4 Architecture of SAE and DBN [10]

2.6.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are type of deep learning networks used to process data points which are related to each other by spatial-relation which is also known as grid-like topology [11]. The operation performed by CNN is purely linear. These networks are different from other deep learning networks in a way that other deep learning methods perform matrix multiplication, but CNN perform convolution operation in some layers. To improve the way of learning in machines convolution network imposes three very important features. Interactions between the sparse, equivariant representation and sharing of the parameters.

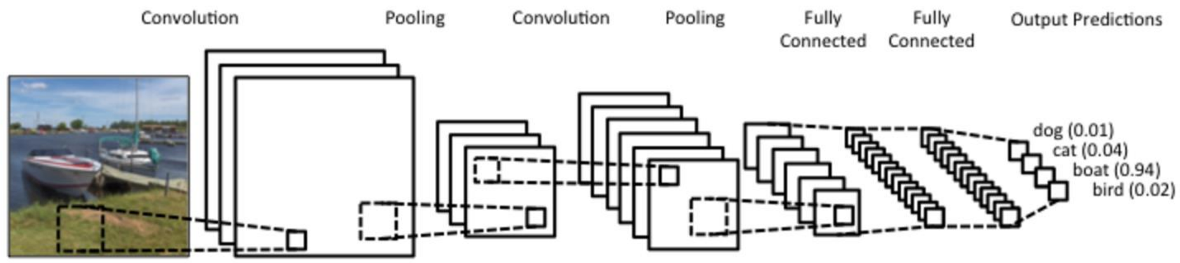


Figure 2.5 Basic Architecture of CNN [12]

Convolution layer is further divided into some layers which are described below:

- **Input layer:** The input image which is given to the CNN architecture is the input layer and it is represented in [A*B*C] form in which raw pixel values of the image are present. A is the height and B is the width of an image while C is color dimension that means its value will be 1 in case of gray image and 3 in case of RGB image.

- **Convolutional layer:** In this layer output of neurons will be computed which are connected with local regions of the input. And this is done by calculating the dot product of the weights and connected small region with in the input volume. For example: if 6 filters are used in first layer then output for the first layer will be [A*B*R] where R will be 6 in this case that means R is number of convolutional filters and filter size is given by (s*t*C). Where s, t are the filter size and C will be the third dimension for the image.

- **Stride:** In stride, over the input matrix in accordance with the number of pixels filter matrix slides or jumps to produce feature maps of lesser size. If we take stride as 1 then the filter is moved by 1 pixel at a time. When it is 5, then filter will jump five pixels at a time. If the stride is larger, then it will produce feature maps which are smaller in size.

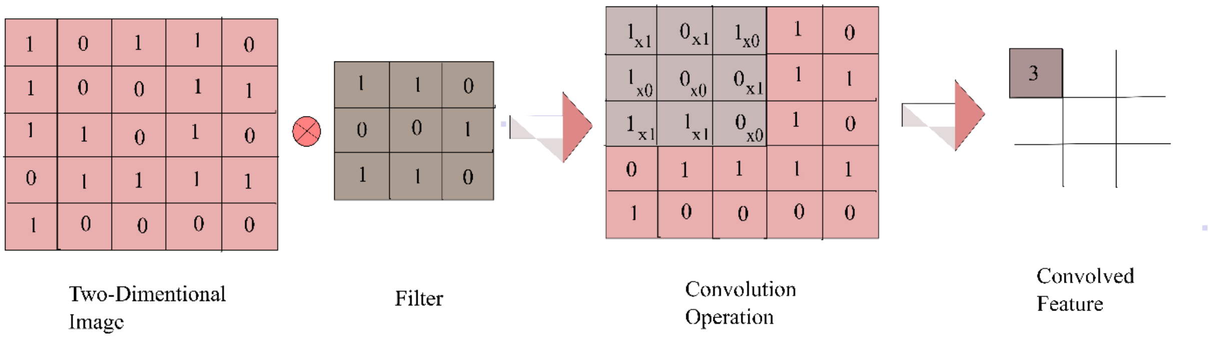


Figure 2.6. Operation performed by convolution layer [13]

- **Zero-padding:** To apply filters to the border elements of the input image matrix, padding with zeroes is done around the input matrix. We can control the feature map size by zero padding. Wide convolution is the other name for adding zero padding. And if we are not adding zero padding then convolution is known as narrow convolution.

- **Activation layer:** After every convolution layer we use the activation layer and activation layers has the non-linear function which controls the output for each layer. It is based on the biological neurons firing mechanism. The non- linearity added by this layer decides about the contribution of which part of input signal to be used in the class label. Non-linearities which are mainly added in Convolutional Neural Networks are Tanh, Rectified Linear Unit, Sigmoid. ReLU is applied per pixel i.e. an element wise operation. The output image which we get from the convolution layer contains positive as well as negative values. ReLU converts the negative value of the output image to zero. Almost all the data that is present in real world has some non-linearity in it, so ReLU is very advantageous to map real-world data.

- **Pooling layer:** Sub-sampling and down-sampling are the other names for spatial pooling. By keeping the important information in pooling layer, we reduce the dimensionality of feature maps. Various types of pooling are: Max, Sum, Average etc. In first case i.e. in max pooling spatial neighborhood is defined and then largest element is taken from feature map within the selected window. In case of Average Pooling we can get average and in Sum pooling sum of the elements is taken.

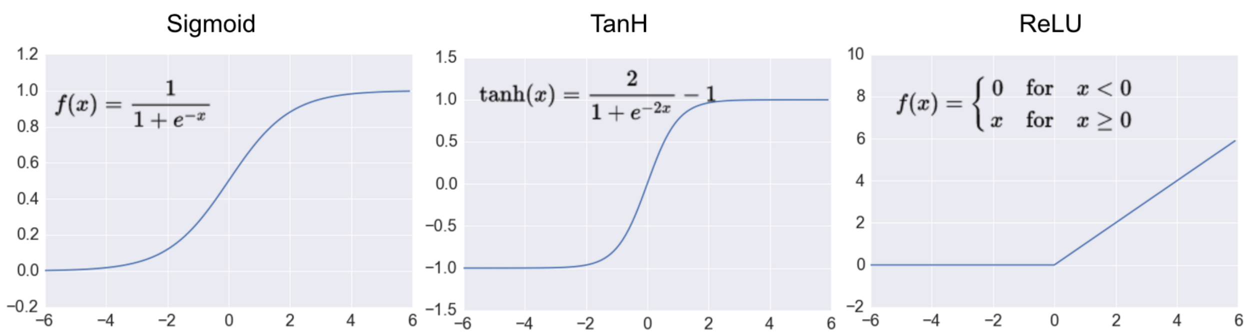


Figure 2.7 Sigmoid, TanH, Rectified Linear Unit (ReLU) [14]

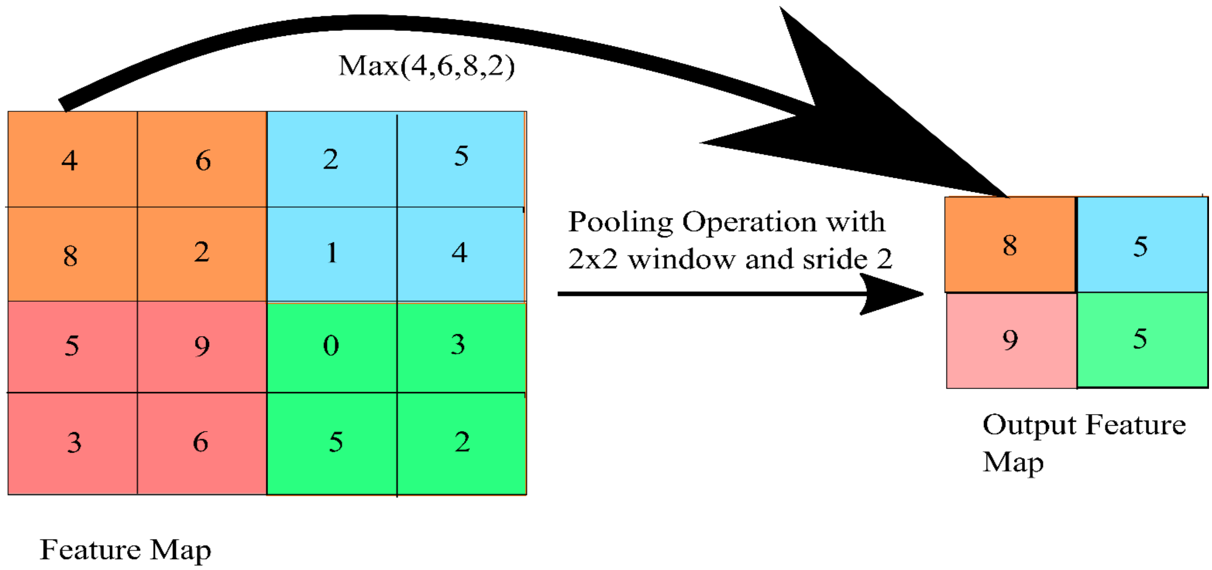


Figure 2.8. Max Pool operation performed by pooling layer [15]

- **Fully connected layer:** "Fully Connected" means that all the neurons of the adjacent layers are interconnected to each other. High-level features of an input image are represented by all the layers discussed above. The aim of the Fully Connected layer is to use the high-level features for the classification purpose. This classification is done on the basis of training dataset.

2.6.3 Recurrent Neural Network

The most important feature of the Recurrent Neural Network (RNN) is that these networks have minimum one feed-back connection. This makes the activations to flow round in a loop. Hidden layers of the network are connected by cyclic connections, so the input data which is passed to these layers is processed sequentially. There is one implicit feature present in recurrent neural networks called as state vector that stores all the information from the past. So, to calculate output for the given input all the past inputs are taken into consideration with the help of these state vectors.

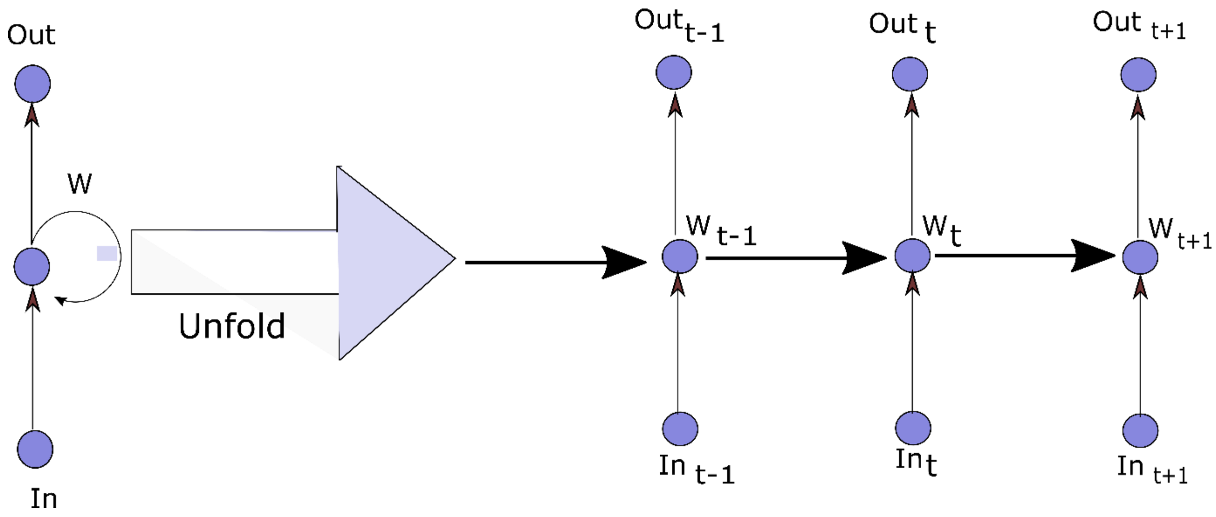


Fig 2.9 Basic architecture of Recurrent Neural Network [16]

To calculate output for the given input, sometimes there is need of both present as well as future inputs. So, for these types of outputs bidirectional RNN's are preferred. Since number of layers in RNN are very less as compared to CNN it has some disadvantages. One of those disadvantages is vanishing problem in gradient descent.

2.6.4 Emergent Architectures

Hidden layers in case of emergent architecture are multi-dimensional. DST-NN is an example of emergent architecture. Progressive refinement is the key feature for this type of architecture. Each layer is DST-NN consists of spatial and temporal features [8]. Original input that is given to the network corresponds to the spatial features. To carry out processes in the higher layers, temporal features are used.

MD-RNN is the other example of emergent architectures. RNN can be applied to sequential data only. So, MD-RNN provided power to RNN to use non-sequential data which is multi-dimensional in nature. CAE also comes in category of emergent architectures. It is the combination of Auto-Encoders and Convolutional Neural Networks. These types of architectures have various applications in image processing like in segmentation of images, recognizing different patterns, prediction of protein structure etc.

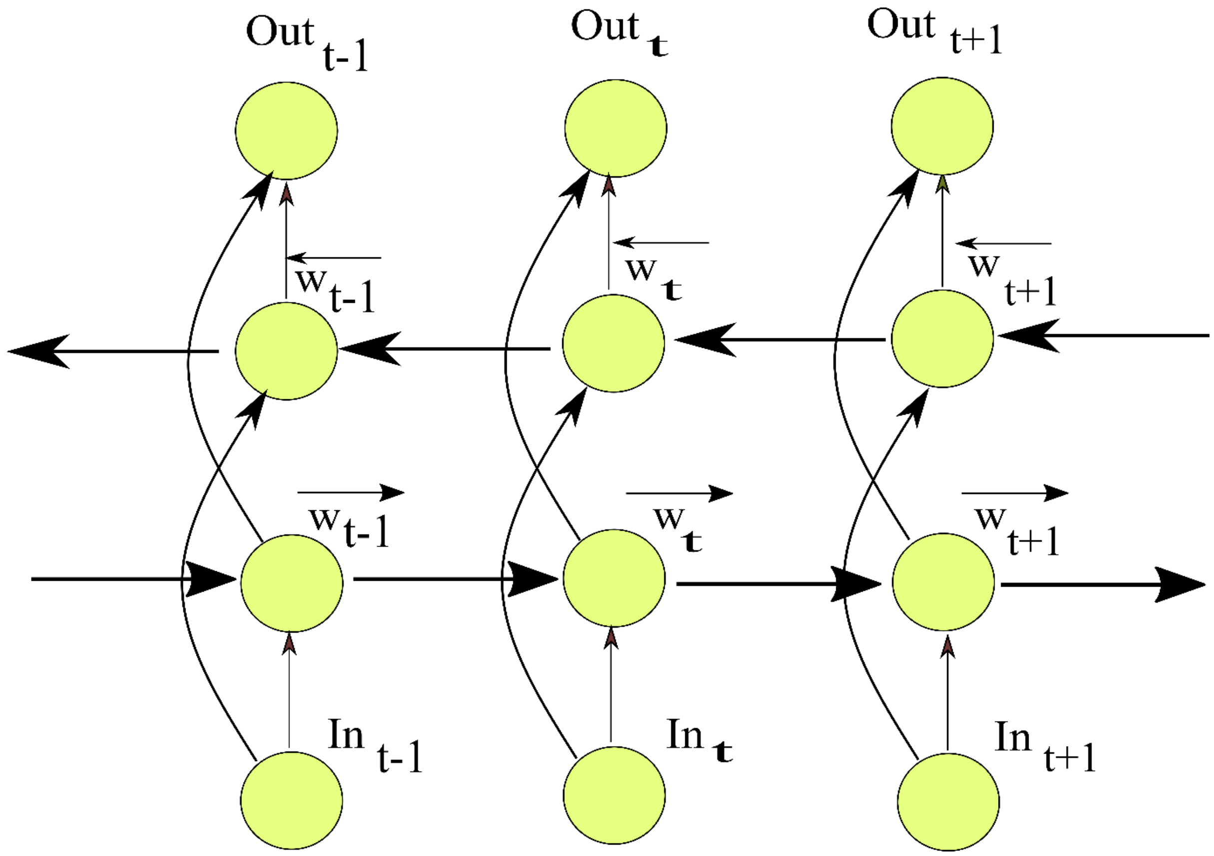


Figure 2.10 Bidirectional RNN [17]

CHAPTER 3

LITERATURE REVIEW

Traditional methods are introduced in this section to resolve problem of Super-Resolution. Basic signal processing techniques and some concepts of machine learning are used in the traditional methods. Traditional methods have mainly three categories.

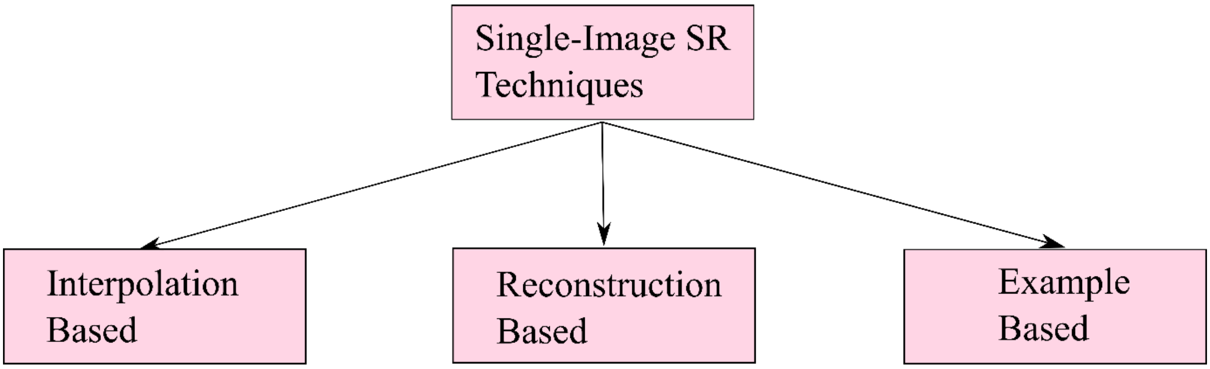


Figure 3.1 Categories of SISR

3.1 INTERPOLATION-BASED METHODS

To convert a low-resolution image to high-resolution the easiest way is to interpolate the output image that we get from the sensors. Digital zoom is the application present in digital cameras which is used for the interpolation purpose and linear filtering is the basic operation on which this whole application relies [18]. Different types of interpolation methods used to resolve super-resolution problem are:

3.1.1 Nearest Neighbor interpolation

In this type of interpolation, nearest neighbor pixel point is determined and then some unknown intensity value is assigned to it [19]. The technique used in this interpolation is the simplest way to achieve interpolation. In nearest neighbor interpolation the kernel used for doing interpolation is:

$$h(y) = \begin{cases} 0, |y| > 0 \\ 1, \ |y| < 0 \end{cases} \tag{1}$$

Frequency Response to the kernel defined above is:

$$H\ (\omega) = \mathrm{sinc}\ (\omega/2) \tag{2}$$

where H (ω) is the frequency response for the kernel h(y).

Although the technique used in this interpolation is very simple and efficient, but the image quality is very low.

3.1.2 Bilinear interpolation

In this interpolation, for a given input coordinate four nearest neighbor pixels are determined and then weighted average of these pixels is calculated and assigned to the unknown value [20]. This interpolation gives better results in terms of image quality as comparison to the technique defined above [21,22]. Kernel used for the interpolation is:

$$v(y) = \begin{cases} 0, |y| > 1 \\ 1 - |y|, |y| < 1 \end{cases} \tag{3}$$

where v(y) is the kernel used in the bilinear interpolation for doing interpolation.

3.1.3 Bicubic interpolation

In this interpolation, for a given input coordinate sixteen nearest neighbor pixels are determined and then weighted average of these pixels is calculated and assigned to the unknown value [23]. This interpolation technique gives more accurate results than the techniques discussed above and this technique is also used as a pre-processing step in many models designed for the purpose of super-resolution. The kernel used in this interpolation is:

$$\text{Q(y)} = \begin{cases} (b+2)|y|^2 - (b+3)|y|^2 + 1 \ \ for\ |y| \leq 1 \\ b|y|^2 - 5b|y|^2 + 8b|y| - 4b\ for\ 1 < |y| < 2 \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases} \tag{4}$$

Where value of b varies from -0.5 to -0.75 and Q(y) is the kernel used for doing interpolation in bicubic interpolation.

3.1.4 Lanczos interpolation

In this interpolation, windowed sinc function [18] is used to find the value of an unknown pixel. This operation can be applied to 4 x 4, 6 x 6 and 8 x 8 cells. The accuracy of 6 x 6 and 8 x 8 filters is more as compare to the bicubic interpolation. But the computation time taken by these two filters to calculate the resultant output image is little longer than all the techniques discussed above. Also, the filters that uses 4 x 4 window to interpolate an image shows results similar to the bicubic interpolation with one dissimilarity that there is no sharpening effect in these images like in the bicubic interpolation.

There are some other techniques based on interpolation:

Gross *et al.* [24] presented another interpolation technique in which they have used multichannel sampling theorem to get a high-resolution image from the low-resolution image. The interpolation technique used here is basically a non-uniform interpolation. The computational load is very less in this technique, so this technique can be used for real-time applications. Disadvantage of this technique is that errors occurred during interpolation are not considered so the reconstruction process is not that optimal.

Boss *et al*. [25] presented another approach which finds out the value of unknown pixel by using moving least square method. To find pixel values of high-resolution image corresponding to the low-resolution image polynomial approximation is done to the neighborhood pixels of the unknown pixel. Adaptive learning is used in this paper to adjust the order of approximation.

Irani *et al*. [26] presented another algorithm known as iterative back projection in which difference is taken between the low and stimulated low-resolution image. And then this difference is used to estimate the pixel values of high-resolution image. But this algorithm has some disadvantages as it does not give unique solution to the SR problem.

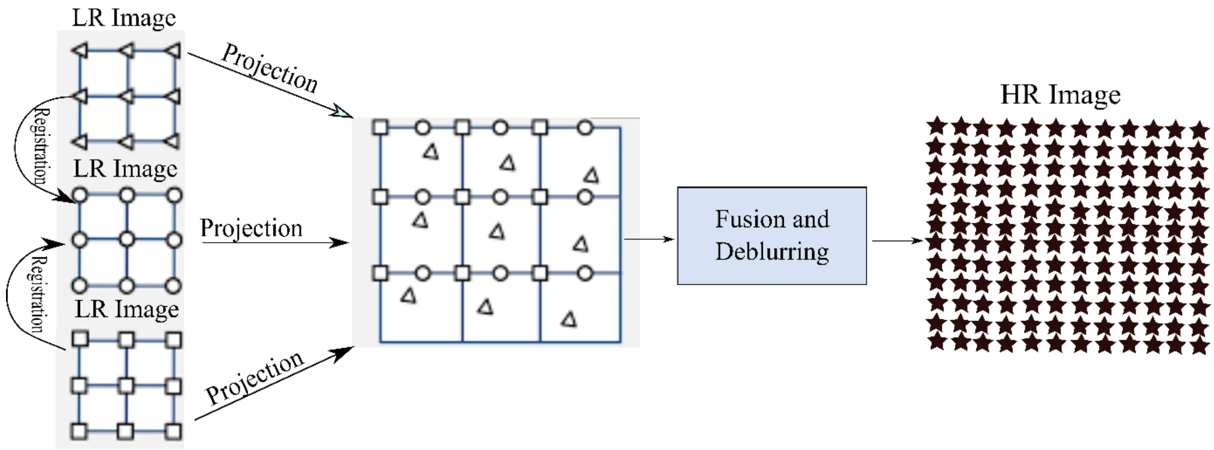


Figure 3.2 The interpolation-based approach for SR image reconstruction [26].

Patti *et al*. [27] presented another algorithm called as projection of convex set. In this algorithm, the information which we get by constantly observing prior image model and low-resolution image is used to construct high-resolution image. Constraint sets are made up by using this information and these sets intersection gives us the solution. But there are two disadvantages of this algorithm: 1) it's computation cost is very high 2) solution given to the problem is not unique.

3.2 RECONSTRUCTION-BASED METHODS

In reconstruction-based methods, there should be knowledge about the high-resolution image and on the basis of that knowledge a reconstruction constraint is framed. In these types of algorithms low-resolution images are considered as the smoothened and down sampled version of high-resolution images.

Chau kao *et al*. [28] presented an edge-based approach to resolve the problem related to super-resolution. In this approach, firstly detection of an edge is done i.e. location and direction of an edge is identified and on basis of this identification interpolation method is selected. This approach produces better results than the direct interpolation techniques.

Fattal *et al*. [29] presented another approach to increase resolution of an image which is based on statistical edge dependency. Problems based on grid-related artifacts are solved using this technique by finding the correct relationships between the edges. Also input intensities are conserved in this technique as output image is the up sampled version of the input image. In this paper as the edge features are enhanced, so visually the results are more appealing.

Sun *et al*. [30] proposed a method in which there is a requirement of image gradients prior knowledge which is gained using large number of images. With the help of this prior knowledge gradient field transformation is done to form gradient fields for the resultant image. Artifacts and ringing effects are removed by using this technique by forming sharp edges.

Protter *et al*. [31] suggested probabilistic motion estimation for task of super-resolution. The results of this technique are same as the NLR-SR. But this technique is comparatively easy than the NLR-SR. also this technique can be adopted for the task of resampling. De-interlacing's example is also shown in the paper for this adaption.

Dai *et al.* [32] presented another approach in which firstly smoothening of edges is done. By doing this image artifacts are reduced. By using the geo-cuts method, a novel technique is developed that characterizes the smoothness for soft edges. For the task of super-resolution in color images alpha-matting technique is used in combination with this technique. So different contrasts and scales edges can be formed using these techniques for color image enhancement.

3.3 EXAMPLE-BASED SR

Machine learning algorithms are used for the example based super-resolution. With the help of examples images having high-resolution are recovered from the images having low-resolution. Learning based approach is the other name for example-based methods. Example-based methods are further divided into two parts:

3.3.1 Parametric Methods

In parametric methods, less number of parameters are required for mapping an image to higher-resolution image [18]. As number of parameter required for estimation are less, so these models are more efficient as compare to the non-parametric models. With the help of examples parameter learns mapping from low-resolution to high-resolution image.

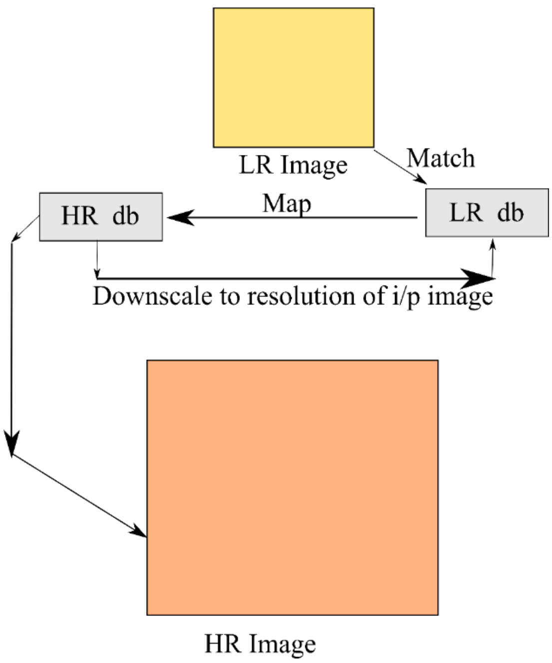


Figure 3.3 Basic idea for example-based SR [33]

Li *et al.* [34] presented a model for super-resolution in which local covariance parameters from the input image are used for the estimation of high-resolution image. With the help of these parameters interpolation is done to achieve high-resolution.

Fattal [35] another approach based on parametric methods for the task of super-resolution. In this approach edge models are used to learn parameters of an image and then these learned parameters are used to get a high-resolution image. As output image is the up sampled version of an input image, so statistical relation is used to find the parameters for the edges and back projection is also used to avoid the problem related to the constrained optimization.

3.3.2 Non-Parametric Methods

These models are completely dependent on the training data. And in these models, prior information about the structure of the model is not given [18]. Parameters in non-parametric methods are more flexible as they are trained according to the training data. Further non-parametric methods are divided into two parts.

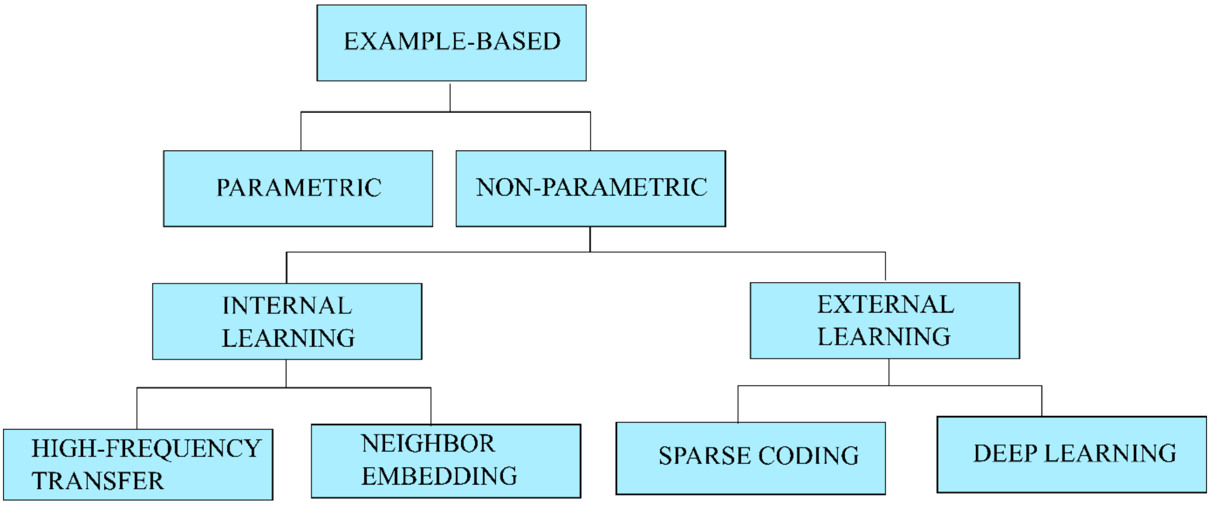


Figure 3.4 Categorization of example-based SR

Internal learning

In case of internal learning, only the input images are used for training i.e. patches are extracted from the input image. As learning is directly from the image, so these types of models give better results for upscaling and noise is also less in these models. Further types of internal learning:

High Frequency Transfer

During interpolation process when low-resolution image is converted to the high-resolution image some of the high frequency components may lost. So, to preserve high frequency components in the output image high frequency transfer methods are used.

Freedman *et al*. [36] presented a high-frequency transfer method. A high-resolution image which is the result of interpolation of low-resolution image has a narrow frequency spectrum. So high-frequency details are missing in the high-resolution image and quality of image is also poor due to this reason. To overcome this problem high frequency components which are extracted from the low-resolution image are merged with the interpolated low-resolution image.

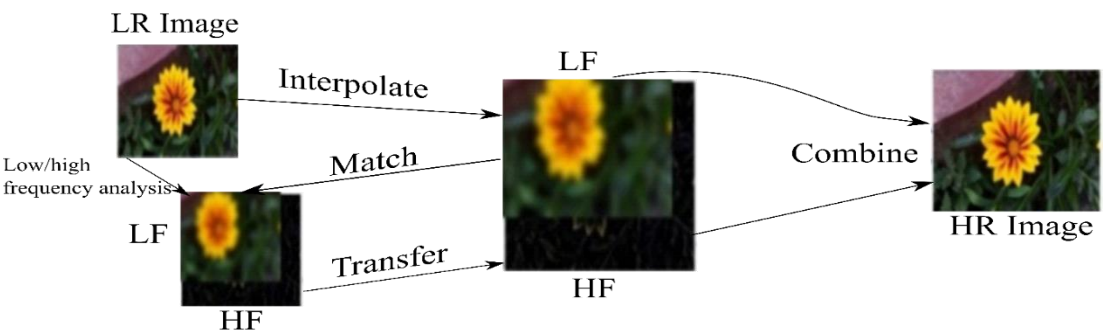


Figure 3.5 Basic idea for high-frequency transfer [33]

Glanser *et al.* [37] proposed another high-frequency method which is based on the property named as cross-scale self-similarity. According to this property, if the low-resolution image is upscaled by some scaling factor then there is very high probability that low-resolution image and upscaled version of this image have identical small patches. Equations describing this method are as follows:

$$U_L = Wr * (V \uparrow r) \tag{5}$$

Where U_L is the upscaled image, Wr is the interpolation kernel, $\uparrow r$ is the factor by which image is upscaled and V is the frequency band spectrum containing low frequencies.

Low-frequency band V_L and high-frequency band V_H can be obtained by using equations shown below:

$$V_L = Wr * V \tag{6}$$

$$V_H = V - V_L \tag{7}$$

As these are following cross-scale self-similar property, final output image (U) equation will be:

$$U = U_L + U_H - Wr * U_H \tag{8}$$

Neighbor Embedding

Basic definition of neighbor embedding is that by linearly combining nearest neighbors, an input data vector can be formed.

Chang *et al.* [38] proposed a novel method based on neighbor embedding to resolve super-resolution problem. In this paper, firstly low-resolution image is down sampled. Then small patches are formed from this down sampled image. Similarly, large patches are formed from the original image. Now these small and large patches are combined to form databases for low-resolution and high-resolution respectively. And with the help of weighted combination of nearest neighbors for small and large patches high-resolution image is reconstructed.

Yu *et al.* [39] proposed a method which is an improvisation to the neighbor embedding method introduced above. This method basically covers two problems which are present in neighbor embedding :1) under and over-fitting 2) in nearest neighbor method only k nearest neighbors are taken into consideration. So, in this method empirical threshold is used to find nearest neighbors. Results obtained from this method are far better than the neighbor embedding technique.

External Learning

In external learning examples are not internal i.e. in this type of learning external datasets are used to train the network. Efficiency of these models is better than the internal learning models.

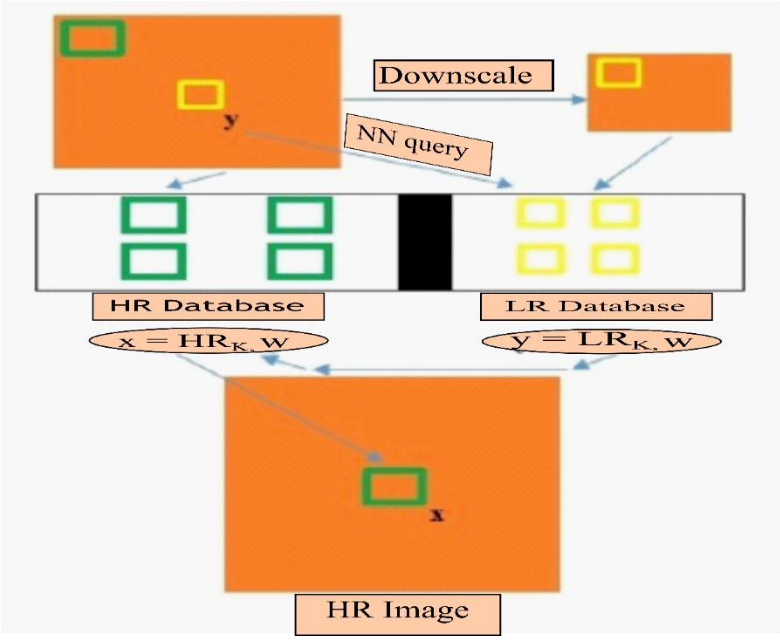


Figure 3.6 Neighbor Embedding Technique [33]

Sparse coding

The disadvantages of the neighbor embedding technique are removed in sparse coding. Unsupervised learning is used in the sparse coding to make dictionaries to extract a high-resolution image from the input image.

Yang *et al*. [40] presented a sparse coding method for super-resolution purpose. When a high-resolution image is downscaled by some factor s, low-resolution image is formed. Sparse dictionaries i.e. low-resolution and high-resolution dictionaries are generated from the down sampled version and the original image patches respectively. Now to correctly recover down sampled signal's sparse representation, compressed sensing is used. Finally, output image can be formed by merging each patch's contribution. Also, in this paper sparse coding technique is compared with the neighbor embedding technique. From there it is concluded that sparse coding technique has some similarities with neighbor embedding technique, but this technique is more optimized because dictionaries in this technique are obtained through training.

Wright *et al*. [41] proposed a sparse signal method to obtain a high-resolution image from the low-resolution image. In this method firstly, low-resolution image patches are represented by sparse coding. Now the coefficients calculated from these low-resolution image patches are used to form high-resolution output. Basically, the patch dictionary and the coefficients are gathered to form a high-resolution image.

Aharon *et al*. [42] introduced the concept of adaptive dictionaries to represent input image patches. The algorithm used here is known as K-SVD which is an iterative method. So, in this method alternating dictionaries are used as K means clustering in K-SVD. Two dictionaries are used in combination in which first is sparse coding from current dictionary and the second one is the dictionary i.e. updating with the data. This algorithm gives better results than the other sparse based algorithms as this algorithm is more flexible.

Deep Learning

Supervised learning in used in this case to get a high-resolution image. Error is calculated and on basis of that error by using back propagation algorithm an output image is formed from the input image.

Chao Dong *et al*. [1] presented a deep learning method for the super resolution for single images. In this paper there is direct mapping from low resolution image to high resolution image with the help of convolutional neural networks. Firstly, a low- resolution image is converted from ground truth image by down sampling and then that low resolution is made to same size as that of the ground truth image by using bicubic interpolation. This image is passed through the CNN network to get the high - resolution image. Comparison is also done with the other state-of-arts method used for the super resolution. Speed and performance are taken into consideration is this paper.

Dong *et al.* [2] further done some modifications in their paper and presented other paper whose main purpose is to accelerate the SRCNN network. As the network used in above paper is better than the hand-crafted models but computational cost of this model is still very high. This model takes lot of time for the calculation of results. So, in this paper low resolution image is directly mapped to the high-resolution image without doing any preprocessing such as interpolation in the starting and directly putting the low-resolution image to the CNN network. In the end of the network deconvolution layer is used for making the size of image equal to the ground truth image. This model has speed more than 40 times as compare to the model used above and has good restoration quality.

Wenzhe Shi *et al.* [3] presented another approach for doing super resolution in images and videos. In this network an efficient subpixel layer is introduced at the end of the network to make size same as that of ground truth image. In this paper deconvolution layer is replaced by the subpixel layer. Phase shift is introduced in the subpixel layer instead of introducing zeroes to increase the size. Rest the network used above the subpixel layer is same as the network used in super resolution convolution neural network i.e. in SRCNN. PSNR value is more accurate in case of this paper, when this network is applied to a large dataset as compared to the SRCNN network.

Daniel Glasner *et al*. [43] presented a paper in which super resolution is done by combining two methods first one is super resolution using classical multi image in which we combine images that are obtained from the misalignments of subpixels and the second approach is super resolution which is done on basis of examples from a database. Image patches are learned by correspondence of low resolution to high resolution. This paper combines these two approaches in a unified framework.

Zhen Cu *et al*. [44] presented a deep network cascade model also known as DNC. In this model a low-resolution image is upscaled to the higher one using layer by layer structure and every layer has the small scale-factor. Cascading of auto-encoders in multiple stacked form in done in DNC. Firstly, non-local self-similarity searching is done for enhancing texture details having high frequency. These patches give very good results in visual quality. Also, it's quantitative performance is very high.

David Eigen *et al*. [45] presented a very effective solution for removing localized rain and dirt artifacts from image. This is a post-capture solution for image processing. A convolutional neural network is trained by using datasets which have clean as well as corrupted image pairs. With the help of CNN, network learns a mapping from corrupted image to the clean image. For outdoor test conditions this model is actually very effective.

Viren Jain *et al*. [46] presented a very effective method for denoising natural images. Two approaches are combined in this paper: first one is the architecture used for processing an image which is convolutional neural network. And the second one is learning procedure for the training purpose which is unsupervised learning. Then the results of this network are compared to the Markov Random Field (MRF) and from the comparison we get to know that the results from both the models are comparable and this is an easy approach as compared to the MRF approach.

J. Schuler *et al*. [47] explains that the sharp image deconvolution of images is an ill-posed problem as it gives blurred images by convolution. So, another very effective technique which can be used is space-invariant non-blind deconvolution. In the first step, images are converted to fourier domain and in Fourier domain there is regularized inversion of blur is done. As the result of this, noise is colored and amplified which then distort the information of the image. After this, the next step is to remove this colored noise with the help of algorithm called non-blind deconvolution.

Yann Lecun *et al*. [48] presented a successful gradient based learning technique in which neural network having multilayers is trained with back propagation algorithm. Convolutional neural network architecture is used with gradient-based learning algorithm to make complex decision for the classification of high-dimensional patterns for example handwritten characters. This paper also gives brief review over the various approaches for the handwritten character recognition and then comparison is done over the standard techniques for hand written character recognition task.

Alex Krizhevsk *et al*. [49] presented a deep convolutional neural network which is trained for the classification of 1.2 million of high-resolution images in the ImageNet contest having 1000 different classes. In comparison to the state-of-the-art method it achieves top-1 or top-5 ranks with error rate of 37.5% and 17.0% respectively on the test data. Architecture of neural network has five convolution layers having 60 million parameters and 650,000 neurons followed by pooling layers, activations functions and the fully connected layer and in the end a 1000-way softmax activation function.

Kaiming He *et al*. [50] presented a time constrained as a cost function for investigating the accuracy of the convolutional neural network. Under the time constraint, architecture of network i.e. CNN architecture need to show trade-off between different factors for example between size of filters, number of filters, depth etc. So, by preserving the time complexity and with the help of controlled comparisons, baseline model is modified progressively. This model in totality helps us in understanding the important factors for the network designing.

Radu Timofte *et al*. [51] presented a method for fast super resolution, assuming a condition that there will be no compromise on the quality of an image. Two methods are used in this technique in a combination: first one is sparse learned dictionaries and second one is neighbor embedding methods. In this method Euclidean distance is not calculated rather nearest neighbors are calculated by finding the correlation with in the dictionary atoms. Also, as global collaborative coding has been used it has more speed due to the reduction of the SR mapping to projective matrix which is precomputed.

Emily Denton *et al*. [52] presented techniques to speed up the evaluation of convolutional neural networks which are designed for the recognition of objects in test-time. This model is very much accurate as comparison to the other models, but it has some disadvantages also. As for evaluating each image this model requires large number of floating point operations, which is problematic in case of their deployment in Internet-scale clusters or in cell phones. Convolution operations are performed in lower layers for this model. Speed of convolution layers in both CPU and GPU can be increased by using large state-of-the-art models.

CHAPTER 4
CONVOLUTION NEURAL NETWORKS FOR SR

Now a day, learning based methods are more preferred for SR algorithms. These methods basically try to find a non-linear relationship between low and high-resolution images. CNN's have recently shown their excellent performance in different image processing tasks such as in image classification and in recognition of objects [53,54,55]. As the learning capabilities of these networks are very high, first network based on CNN for the task of super resolution is proposed by Dong et al. [1]. Detailed discussion of these networks in presented below.

4.1 ARCHITECTURE

Convolutional Neural Networks (CNNs) are type of deep learning networks used to process data points which are related to each other by spatial-relation which is also known as grid-like topology. The operation performed by CNN is purely linear. These networks are different from other deep learning networks in a way that other deep learning methods perform matrix multiplication, but CNN perform convolution operation in some layers. To improve the way of learning in machines convolution network imposes three very important features: Interactions between the sparse, equivariant representations and sharing of the parameters.

CNN architecture is basically build of four types of layers: Convolutional layer, activation layer, pooling layer, fully connected layer. Pooling layer is generally used for the purpose of dimension reduction and fully connected layer is way of learning non-linear combinations of the features. So, these two layers are basically used for object classification and detection purposes. Only the convolution layer and activation layer are used for the SR purpose.

The reference model ESPCNN [35] which we are using for our work consists of three convolution layers having ReLU as activation function and last layer is the subpixel convolution layer having one or three channels according to the image as it is a gray scale image or the colored image respectively. Subpixel layer is basically derived from the convolution layer. To make a subpixel layer from the convolution layer an operator is added which periodically shuffles the elements so that rearrangement of elements with the shape of N x (W x S) x (H x S) from the shape (N x S^2) X W x H is done. Where the upscaling factor is denoted by S and number of feature maps are denoted by N. W and H are the height and width of an image respectively. S is very important in the model as LR feature space is connected to the HR feature space by this factor.

4.2 RECONSTRUCTION

In SR for reconstruction of low-resolution images into high-resolution images we need to have a better CNN architecture and to make CNN architecture better, there is a need to minimize Euclidean distance loss in between the image that is predicted and original HR image. In [3] for the computation of the similarity between low-resolution and high-resolution image mean square error (MSE) is used as a metric.

General formula for the MSE in case of SR:

$$\mathrm{MSE} = \frac{1}{\mathrm{W\,X\,H}} \parallel \mathrm{X} - \mathrm{Y} \parallel^2 \tag{9}$$

here W is the width and H is the height of an image, X is the ground-truth image and Y is the predicted image.

And for evaluating the restoration quality of the SR image, the most common metrices which are used are Peak signal-to-noise ratio(PSNR) and the Structural Similarity (SSIM) index. PSNR is basically used for the quality measurement of an image that is reconstructed or restored with reference to the ground truth image. If the PSNR value in between images is high, then the CNN model is better and vice versa

$$\mathrm{PSNR} = 20 \log_{10} \frac{\mathrm{MAX}}{\sqrt{\mathrm{MSE}}} \tag{10}$$

here MAX is the maximum pixel value that is possible of an image. If the representation of pixels is using eight bits per sample, then the value of MAX will be 255. And if have normalized the image pixels then the value of MAX will be 1.

SSIM is basically a method for measuring the similarity between reconstructed or restored image with the ground truth image. Three terms are computed to calculate SSIM and these terms are luminance, contrast and structural. The overall function is a multiplication of these three terms.

$$\mathrm{SSIM(X,Y)} = [\mathrm{I(X,Y)}]^{\alpha}.\,[\mathrm{C(X,Y)}]^{\beta}.\,[\mathrm{S(X,Y)}]^{\gamma} \tag{11}$$

$$\mathrm{SSIM(X,Y)} = \frac{(2\mu_x\mu_\gamma + C_1) + (2\sigma_{x\gamma} + C_2)}{(\mu_x{}^2 + \mu_\gamma{}^2 + C_1)\,(\sigma_x{}^2 + \sigma_\gamma{}^2 + C_2)} \tag{12}$$

where X is the ground truth image, Y is the predicted image, μ_x and μ_γ are means of ground image and predicted image respectively, σ_x and σ_γ are standard deviations of ground truth and predicted image respectively, $\sigma_{x\gamma}$ is the covariance of ground truth and predicted image.

At the training time, the calculation of the loss in CNN architecture is done by equation shown below:

$$\mathrm{LOSS} = \frac{1}{2\mathrm{M}} \sum_{\mathrm{I}=1}^{\mathrm{M}} \parallel \mathrm{X} - \mathrm{Y} \parallel^2 \tag{13}$$

Here M is the total number of samples of the images used during training. Therefore, with the help of these equations we can reconstruct a high-resolution image from the low-resolution one.

4.3 OPTIMIZERS

The difference between the desired image and the predicted image is known as the cost function. One such example is mean square error. The function of optimizers is to reduce this cost function by applying some formula. So, in machine learning there are different types of optimizers to minimize the cost function based on their functionality. All these optimizers have different formulas for updating weights. Types of optimizers [57] used in neural networks are as follow:

4.3.1 Stochastic Gradient Descent

If the size of data that is used for training is very large, then it is not feasible to train network with gradient descent algorithm, so in this case stochastic gradient descent is used. Speed of this technique is much greater than the gradient descent algorithm. To each training sample with in the training data, stochastic gradient descent carries out parameter update. It has some disadvantages also. As the fluctuation caused by the frequent update of parameters are very high, it makes the process of convergence to minima very difficult. This problem also leads to high variance.

4.3.2 Adagrad

Learning rate in case of adagrad is adaptive in nature. This means that learning rate depends upon the occurrence of parameters. If the occurrence of the parameter is not that frequent then the learning rate makes big update and similarly if the occurrence of the parameter is very frequent then the learning rate makes very small updates. The disadvantage of adagrad is the decaying nature of learning rate. As the learning rate is decreasing with each update so after some time the network will not be able to learn new features.

4.3.3 Adadelta

Adadelta is the advanced version of adagrad. It basically overcome the problem of decreasing learning rate caused by adagrad. It put a constrained by fixing gradients to some value instead of adding all the past squared gradient. So, learning rate in this is adaptive in nature but will not reduce to zero with increasing updates. The momentum in case of adadelta is fixed.

4.3.4 Adam

The rate of convergence and speed of learning in case of adam is very quick. The main of adam is its ability to store the past gradients momentum. So, in adam with the adaptive learning rate, adaptive momentum is the additional feature. It overcomes nearly all the drawbacks of the optimizers defined above. That’s why adam is preferred over all other optimizers.

4.4 RESIDUAL LEARNING

Although ESPCNN is more effective than the other learning methods but high-frequency details are missing in this architecture also. Because in CNN during training, architecture needs to preserve each input detail of the image and according to the CNN architecture output image which is produced at the end is totally depends upon the learned features.

With increase in the number of convolution layers for end to end mapping there is need of long-term memory which worsen the problems related to gradients. So high-frequency components are vanished in this process of training which reduces performance of the architecture. Therefore, to get more effective performance in image SR, Kim *et al*. [58] introduces residual learning for the training purpose. In residual learning input is merged with the resultant image to get output image.

CHAPTER 5

PROPOSED METHODOLOGY

The first proposed architecture is shown in figure 5.1 and we named it as DLSPCNN (Double-layered Subpixel Convolutional Neural Network).

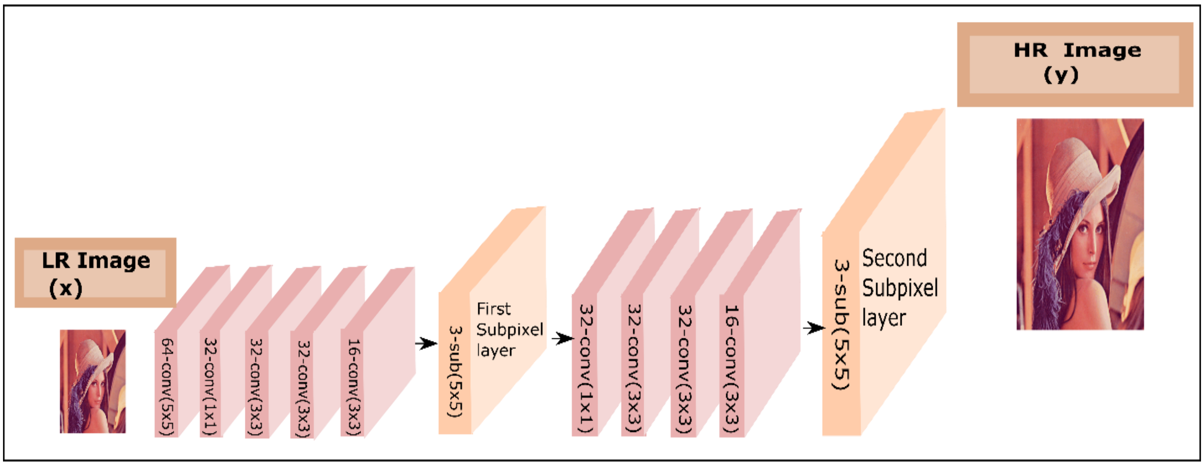


Figure 5.1 DLSPCNN architecture for Super Resolution of an input low-resolution image with upscaling factor of four

As shown in the figure 5.1 LR image is passed to the first convolution layer and this convolution layer is followed by series of convolution layers which have ReLU [57] as their activation function. Operation on first convolution layer is described by equation written below:

$$\mathcal{F}^1(\mathrm{I}^{\mathrm{LR}}; \mathrm{W}_1, \mathrm{B}_1) = \varnothing(\mathrm{W}^1 \otimes \mathrm{I}^{\mathrm{LR}} + \mathrm{B}_1) \tag{14}$$

Now first convolution layer is followed by four hidden convolution layers having ReLU as activation function and this can be written as below:

$$\mathcal{F}^{\mathrm{u}}(\mathrm{I}^{\mathrm{LR}}; \mathrm{W}_{1:\mathrm{u}}, \mathrm{B}_{1:\mathrm{u}}) = \varnothing(\mathrm{W}_{\mathrm{u}} \otimes \mathcal{F}^{\mathrm{U}-1}(\mathrm{I}^{\mathrm{LR}}) + \mathrm{B}_{\mathrm{u}}) \tag{15}$$

where W_u are learnable weights,B_u are biases and values of u ranges from (1: U-1) where U is 5. Size of learnable weight W_u is given by n_{u-1} x n_u x k_u x k_u, where number of features are represented by n_u at layer u and filter size is given by k_u. To add non-linearity, we have used ReLU as activation function and in equation $\varnothing$ is denoting activation function. Now output of Equation (15) is given as input to the first subpixel layer which is described in the equation written below:

$$\mathrm{I}^{\mathrm{SP}} = \mathcal{F}^{\mathrm{U}}\left(\mathrm{I}^{\mathrm{LR}}\right) = \mathrm{SP}(\mathrm{W}_{\mathrm{U}} \otimes \mathcal{F}^{\mathrm{U}-1}\left(\mathrm{I}^{\mathrm{LR}}\right) + \mathrm{B}_{\mathrm{U}}) \tag{16}$$

Here SP is denoting the subpixel layer. Output image from Eq. (16) is now upscaled by a factor of 2 and this layer consists of only one or three channels corresponding to which type of image we are using as if it is a gray scale image or a colored image respectively. Output image from the first subpixel layer is passed to the series of convolution layers which are followed by the activation function ReLU as discussed below:

$$\mathcal{F}(I^{SP}; W_2, B_2) = \varnothing(W_2 \otimes I^{SP} + B_2) \tag{17}$$

$$\mathcal{F}^{v}\left(I^{SP}; W_{1:v}, B_{1:v}\right) = \varnothing(W_v \otimes \mathcal{F}^{v-1}(I^{SP}) + B_v) \tag{18}$$

where W_v are learnable weighs in architecture, B_v are biases and value of v is in range (1: V-1) and V is 4 here. The output from the last hidden convolution layer is passed to the second subpixel layer that upscales the image by a factor of two as described below:

$$I^{SP1} = \mathcal{F}^{V}\left(I^{SP}\right) = SP(W_V \otimes \mathcal{F}^{V-1}\left(I^{SP}\right) + B_V) \tag{19}$$

Output of the Eq. (19) is the final image that is the predicted image.

The second architecture proposed by us is shown in Figure 5.2 which is double-layered subpixel Convolutional Neural Network with two-stage residual learning(DLSPCNN-2SRL).

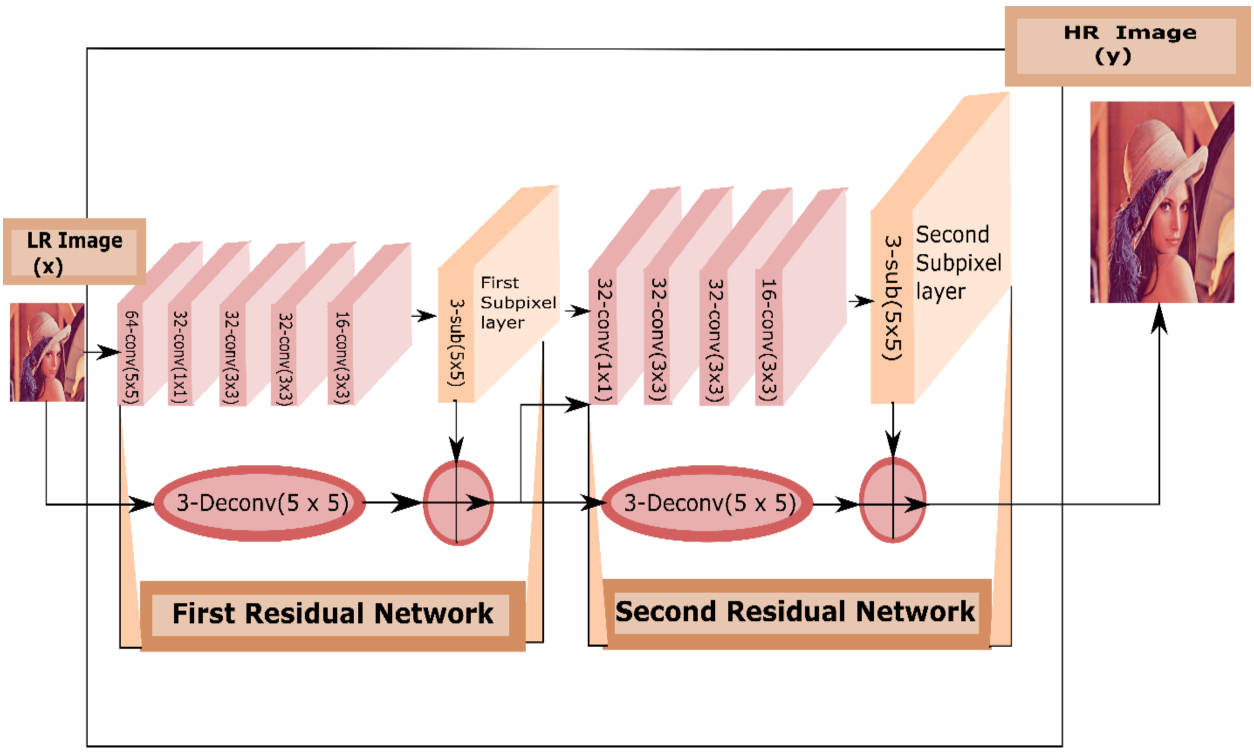


Figure 5.2 DLSPCNN with 2-SRL architecture with residual learning to super-resolve an input low-resolution image with upscaling factor of four

In this architecture we have introduced residual learning to our network. Basic idea of residual learning is as follows:

$$Y=F(x)+x \tag{20}$$

where Y is the output image, x is the input image and F(x) is the image obtained after applying some operations on the input image x.

In residual learning, input image x is merged with output of the convolution layers to get final predicted image. In our architecture, we have introduced residual learning in two stages, so we have divided our architecture in two networks and named them as first residual network and the second residual network.

LR image is passed to the first convolution layer and this convolution layer is followed by series of convolution layers which have ReLU as their activation function. Operation on first convolution layer is described by equation written below:

$$\mathcal{F}^1(\mathrm{I}^{\mathrm{LR}}; \mathrm{W}_1, \mathrm{B}_1) = \varnothing(\mathrm{W}^1 \otimes \mathrm{I}^{\mathrm{LR}} + \mathrm{B}_1) \quad (21)$$

Now first convolution layer is followed by four hidden convolution layers having ReLU as activation function and this can be written as below:

$$\mathcal{F}^{\mathrm{u}}(\mathrm{I}^{\mathrm{LR}}; \mathrm{W}_{1:\mathrm{u}}, \mathrm{B}_{1:\mathrm{u}}) = \varnothing(\mathrm{W}_{\mathrm{u}} \otimes \mathcal{F}^{\mathrm{U}-1}(\mathrm{I}^{\mathrm{LR}}) + \mathrm{B}_{\mathrm{u}}) \quad (22)$$

where W_u are learnable weights,B_u are biases and values of u ranges from (1: U-1) where U is 5. Size of learnable weight W_u is given by n_{u-1} x n_u x k_u x k_u, where number of features are represented by n_u at layer u and filter size is given by k_u. To add non-linearity, we have used ReLU as activation function and in equation $\varnothing$ is denoting activation function. Now output of Equation (21) is given as input to the first subpixel layer which is described in the equation written below:

$$\mathrm{I}^{\mathrm{SP}} = \mathcal{F}^{\mathrm{U}}\left(\mathrm{I}^{\mathrm{LR}}\right) = \mathrm{SP}(\mathrm{W}_{\mathrm{U}} \otimes \mathcal{F}^{\mathrm{U}-1}\left(\mathrm{I}^{\mathrm{LR}}\right) + \mathrm{B}_{\mathrm{U}}) \quad (23)$$

Then the image from the subpixel layer is merged with the input image to preserve the high-frequency details. As residual learning can only be done with same size of images so before merging input image with the first subpixel layer, image scaling of the input image has been done with a factor of two by using deconvolution layer.

$$\mathrm{I}^{\mathrm{DLR}} = \mathrm{DC}(\mathrm{I}^{\mathrm{LR}}) \quad (24)$$

And then this deconvoluted input image is merged with the image coming from the subpixel layer as described below:

$$\mathrm{I}^{\mathrm{SPx}} = \mathrm{I}^{\mathrm{SP}} + \mathrm{I}^{\mathrm{DLR}} \quad (25)$$

where I^{SP} is the output image from first subpixel layer and $\mathrm{I}^{\mathrm{DLR}}$ is low-resolution input image which is upsampled by using deconvolution layer.

Now in the second residual network, the resultant image from the merging layer is passed to the other convolution layer followed by three more hidden convolution layers having ReLU as an activation function as shown below:

$$\mathcal{F}(\mathrm{I}^{\mathrm{SPx}}; \mathrm{W}_3, \mathrm{B}_3) = \varnothing(\mathrm{W}_3 \otimes \mathrm{I}^{\mathrm{SPx}} + \mathrm{B}_3) \quad (26)$$

$$\mathcal{F}^{\mathrm{j}}\left(\mathrm{I}^{\mathrm{SPx}}; \mathrm{W}_{1:\mathrm{j}}, \mathrm{B}_{1:\mathrm{j}}\right) = \varnothing(\mathrm{W}_{\mathrm{j}} \otimes \mathcal{F}^{\mathrm{J}-1}(\mathrm{I}^{\mathrm{SPx}}) + \mathrm{B}_{\mathrm{j}}) \quad (27)$$

where W_j and B_j are the learnable weights and biases of the architecture and range of j is (1:J-1) and value of j is 4.

Now the output image from Eq. (27) is passed to the second subpixel layer where the image is again upscaled with the factor of two as described below:

$$I^{SP2} = \mathcal{F}^{V}\left(I^{SPx}\right) = SP(W_J \otimes \mathcal{F}^{J-1}\left(I^{SPx}\right) + B_J) \tag{28}$$

Eq. (25) is upscaled by factor of 2 using deconvolution layer as shown below:

$$I^{DSPx} = DC(I^{SPx}) \tag{29}$$

Then we merge these two images as follows:

$$I^{SR} = I^{SP2} + I^{DSPx} \tag{30}$$

The resultant image from Eq. (30) is the final output that is HR version of the given LR image.

For optimization we have used adam optimizer. For every parameter adam calculates adaptive learning rates. Weight updation rule for adam is given as:

$$Q_{t+1} = Q_{t} - \frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon}\widehat{m_t} \tag{31}$$

where $\widehat{m_t}$ and $\widehat{v_t}$ are the estimates of mean and variance respectively and default value of ϵ is 10^{-8}.By using these two architectures we are getting better results and significant increase in the PSNR values as comparison to the ESPCNN.

CHAPTER 6
EXPERIMENTS, RESULTS AND ANALYSIS

In experiments section, we are evaluating our model performance on different datasets. Firstly, there is brief introduction to the datasets we have used for training and testing purpose. And finally, comparison is done between our models and some state-of-the art methods.

6.1 DATASETS

Training and testing dataset

From Pascal VOC we have randomly selected 4,000 images and then patches with size 300 x 300 are cropped from these images. After this we have divided these images into training and testing dataset. First 3000 patches are used for the training dataset. The patches in the training dataset are then resized into size of 25 x 25 and 100 x 100 patches. Patches with size 25 x 25 are given to the model as an input. The images given for training are different from the images given in the testing and validation datasets. From the 4,000 images last 1,000 images are used for testing dataset. We have also tested our networks on publicly available benchmark datasets- **set5**, **set14** and **B500**.

6.2 EXPERIMENTAL SETUP

We have evaluated the performance of our model for the upscaling factor of 4. To obtain LR image, HR image is down sampled with a factor of 4 i.e. 3,000 images having size 300 x 300 are resized by using bicubic kernel to the patches with size 25 x 25. These patches are given to the model as an input. Also, these 300 x 300 images are resized into the patches with size of 100 x 100. We have used these image patches to train our network. To optimize our network, we have used Adam as an optimizer with the learning rate of 0.001 and to calculate loss we have used mean square error as a metric. As phenomenon of overfitting is not there, so we haven’t used dropout and weight decay. We have trained our model for 50 epochs. We have used the default batch size i.e. 32. To quantitatively evaluate our model, we have used Structural Similarity (SSIM) index measurement and Peak signal to noise ratio as metrices. These two metrices are mostly used for the quantitative evaluation in image SR techniques.

6.3 COMPARISONS WITH STATE-OF-THE-ART APPROACHES

Finally, we have compared our both model’s performance with bicubic interpolation and some other state-of-the-art methods: SRCNN, ESPCNN. In table 1,2,3 and 4 we have presented a quantitative analysis on three most common datasets that are B500, set5, set14 and on our dataset respectively. The analysis summarized in the tables shows that our both models that are DLSPCNN and DLSPCNN with 2-SRL have better PSNR value and SSIM. So, it implies that these two models are better than the other state-of-the-art method for SR of images. Also, in Figures 6.1,6.2,6.3,6.4,6.5,6.6 visual results are compared between our models and other state-of-the-art models including SRCNN, ESPCNN, and bicubic interpolation and we are getting better results visually as well.

6.4 RESULTS

Resultant figures after applying different techniques to achieve Super-Resolution are shown below:

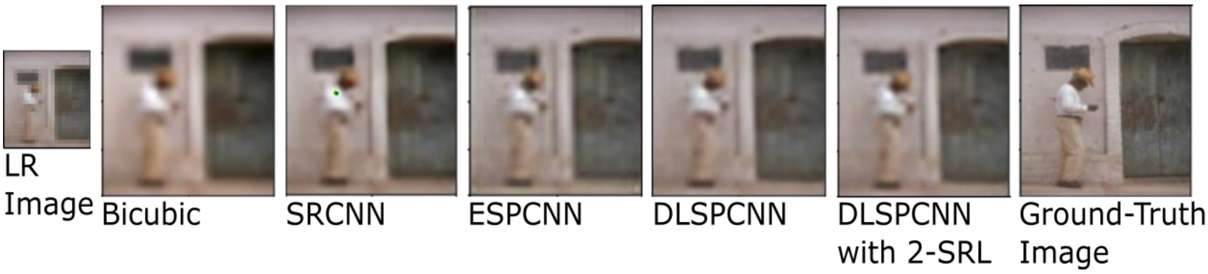


Figure 6.1 SR results. The ‘64061’ image from the B500 is upscaled by the factor of 4 using different state-of-the art algorithms.

In this figure first image is a low-resolution image. First technique applied to increase resolution of this image is bicubic interpolation. PSNR value for the bicubic interpolation is 18.75 dB. Second method applied to achieve high resolution is Super Resolution Convolutional Neural Network (SRCNN). By applying SRCNN, PSNR value is increased to 20.95 dB but the computational time is very high for this method. Third technique is Efficient Subpixel Convolutional Neural Network (ESPCNN) which have PSNR value 30.38 dB. Fourth technique is Double Layered Subpixel Convolutional Neural Network (DLSPCNN) whose PSNR value is coming out to be 30.68 dB. And the last technique is Double Layered Subpixel Convolutional Neural Network with 2-Stage Residual Learning (DLSPCNN with 2-S RL). This technique has the highest PSNR value which is 31.27dB.

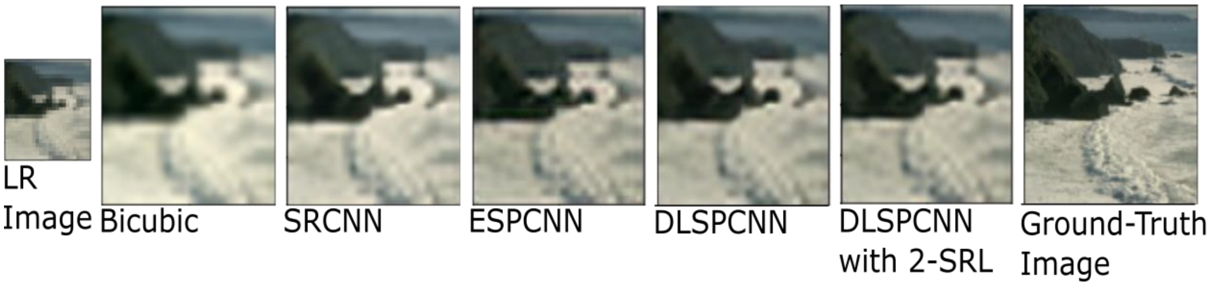


Figure 6.2 SR results. The ‘117025’ image from the B500 is upscaled by the factor of 4 using different state-of-the art algorithms.

In the figure shown above firstly bicubic interpolation is done to convert a low-resolution image of dimensions 25 x 25 to a high-resolution image having dimensions 100 x 100. But the PSNR value after applying bicubic interpolation is only 24.03 dB which is very less. So, the next technique applied to have high-resolution is SRCNN. PSNR value in this case is 27.51 dB which is greater than the previous technique. The third technique used to get a high-resolution image is ESPCNN and the PSNR value coming out to be 29.40 dB in case of this technique. The fourth technique used to achieve Super-Resolution is DLSPCNN. PSNR value for this technique is 29.99 dB. Fifth technique used is DLSPCNN with 2-S RL and this technique has the highest PSNR value and i.e. 30.07 dB.

Table 6.1: Performance comparison for various SR methods of '11725' **(B500)** and '64061'**(B500)** test set with scale factor of 4. The highest scores are in **red bold** and second highest are in blue bold.

B500	Metric	Bicubic	SRCNN	ESPCNN	DLSPCNN	DLSPCNN With 2-SRL
11725	PSNR	18.75	20.95	30.38	30.68	31.27
	SSIM	0.83	0.86	0.89	0.90	0.90
	MSE	0.0013	0.0080	0.00091	0.00085	0.00074
64061	PSNR	24.03	27.51	29.40	29.99	30.07
	SSIM	0.87	0.81	0.89	0.88	0.90
	MSE	0.0039	0.0017	0.0014	0.0010	0.00098

Figure 6.3 SR results. The 'face' image from the SET5 is upscaled by the factor of 4 using different state-of-the art algorithms.

The face figure taken from the SET5 is firstly converted to a low-resolution image which have size 25 x 25 by down sampling. Then on down sampled image, bicubic interpolation technique is applied. PSNR value after applying bicubic interpolation is 27.21 dB. As the results are not very good so SRCNN is the next technique used to get an image with high-resolution. PSNR value is case of SRCNN is 27.63 dB. After SRCNN next technique used is ESPCNN. PSNR value for ESPCNN is 28.09 dB. By applying next technique which is DLSPCNN there is an increase in the PSNR value from the previous one 0.6 dB. The PSNR value is 28.15 in this case. And after applying DLSPCNN with 2-S RL the results are more accurate i.e. the PSNR value for this technique is 29.18 which is greater than all the other techniques discussed above.

Table 6.2: Performance comparison for various SR methods of ‘face’ (**Set5)** test set with scale factor of 4. The highest scores are in **red bold** and second highest are in blue bold

Metric	**Bicubic**	**SRCNN**	**ESPCNN**	**DLSPCNN**	**DLSPCNN With 2-S RL**
PSNR	27.21	27.63	28.09	28.15	29.18
SSIM	0.86	0.88	0.88	0.89	0.90
MSE	0.0019	0.0017	0.0015	0.00152	0.0012

Figure 6.4 SR results. The ‘barbara’ image from the SET14 is upscaled by the factor of 4 using different state-of-the art algorithms.

Bicubic interpolation is the first technique applied to a low-resolution image to get a high-resolution image. PSNR value is 23.03 dB for this technique. Second technique used is SRCNN and the PSNR value corresponding to this technique is 23.49 dB. To get a more accurate image, third technique used is ESPCNN and the PSNR value for this technique is 23.52 dB. For Double Layered Subpixel Convolutional Neural Network, the PSNR value is 23.86 dB. There is a significant increase in the PSNR value after applying last technique i.e. DLSPCNN with 2-S RL and the PSNR value is 24.50 dB.

Table 6.3: Performance comparison for various SR methods of ‘Barbara’ (**Set14)** test set with scale factor of 4. The highest scores are in **red bold** and second highest are in blue bold.

Metric	**Bicubic**	**SRCNN**	**ESPCNN**	**DLSPCNN**	**DLSPCNN With 2-SRL**
PSNR	23.03	23.49	23.52	23.86	24.50
SSIM	0.79	0.81	0.80	0.81	0.83
MSE	0.0049	0.0044	0.0043	0.0041	0.0035

Figure 6.5 SR results. The image from our dataset upscaled by the factor of 4 using different state-of-the art algorithms.

In this figure first image is a low-resolution image. First technique applied to increase resolution of this image is bicubic interpolation. PSNR value for the bicubic interpolation is 22.69 dB. Second method applied to achieve high resolution is Super Resolution Convolutional Neural Network (SRCNN). By applying SRCNN, PSNR value is increased to 26.77 dB but the computational time is very high for this method. Third technique is Efficient Subpixel Convolutional Neural Network (ESPCNN) which have PSNR value 28.32 dB. Fourth technique is Double Layered Subpixel Convolutional Neural Network (DLSPCNN) whose PSNR value is coming out to be 29.04 dB. And the last technique is Double Layered Subpixel Convolutional Neural Network with 2-Stage Residual Learning (DLSPCNN with 2-S RL). This technique has the highest PSNR value which is 29.11 dB.

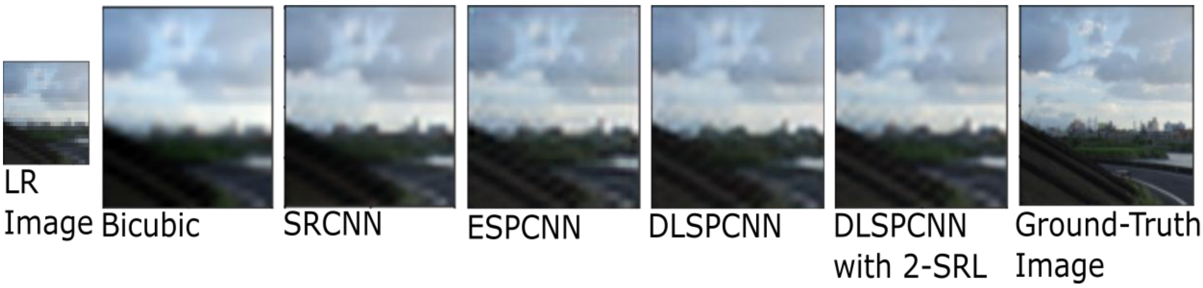


Figure 6.6 SR results. The image from our dataset upscaled by the factor of 4 using different state-of-the art algorithms.

In the figure shown above firstly bicubic interpolation is done to convert a low-resolution image of dimensions 25 x 25 to a high-resolution image having dimensions 100 x 100. But the PSNR value after applying bicubic interpolation is only 16.72 dB which is very less. So, the next technique applied to have high-resolution is SRCNN. PSNR value in this case is 18.54 dB which is greater than the previous technique. The third technique used to get a high-resolution image is ESPCNN and the PSNR value coming out to be 26.86 dB in case of this technique. The fourth technique used to achieve Super-Resolution is DLSPCNN. PSNR value for this technique is 27.46 dB. Fifth technique used is DLSPCNN with 2-S RL and this technique has the highest PSNR value and i.e. 27.80 dB.

Table 6.4: Performance comparison for various SR methods on images of our own dataset with scale factor of 4. The highest scores are in **red bold** and second highest are in blue bold

Own dataset	Metric	Bicubic	SRCNN	ESPCNN	DLSPCNN	DLSPCNN with 2-SRL
1	PSNR	22.69	26.77	28.32	29.04	29.11
	SSIM	0.83	0.87	0.89	0.89	0.90
	MSE	0.0053	0.00131	0.00091	0.0012	0.0012
2	PSNR	16.72	18.54	26.86	27.46	27.80
	SSIM	0.86	0.88	0.89	0.90	0.91
	MSE	0.0212	0.0139	0.0020	0.0017	0.00165

- For Super Resolution Convolutional Neural Network, variation of mean square error and validation loss with the increasing number of epochs are shown below with the help of graphs.

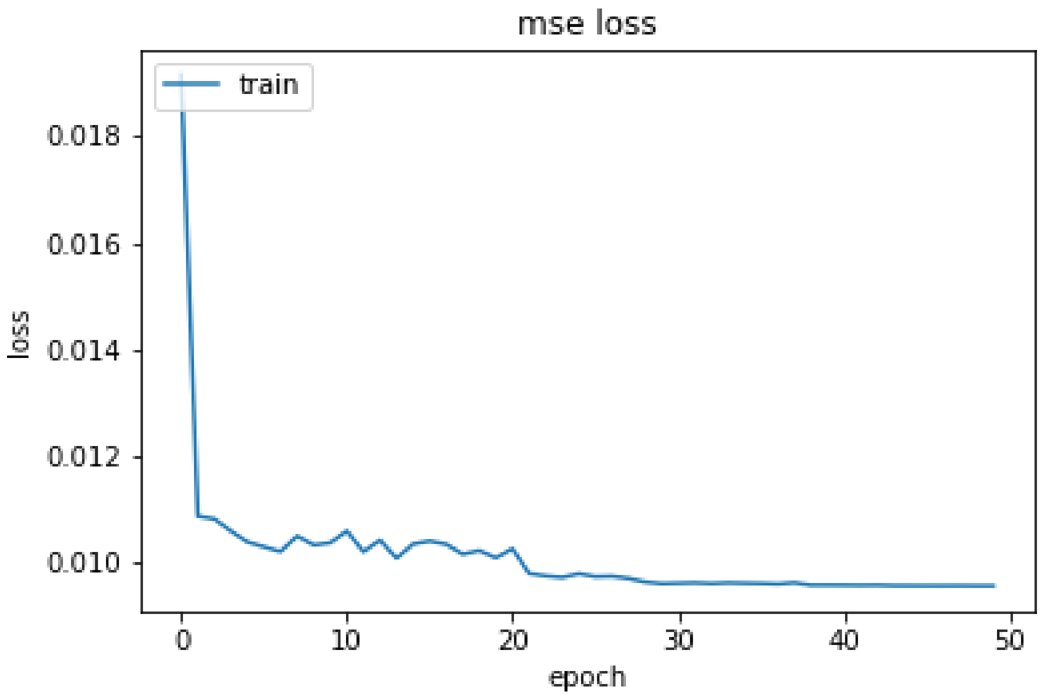


Figure 6.7 Graph showing convergence of mean square error with increasing number of epochs.

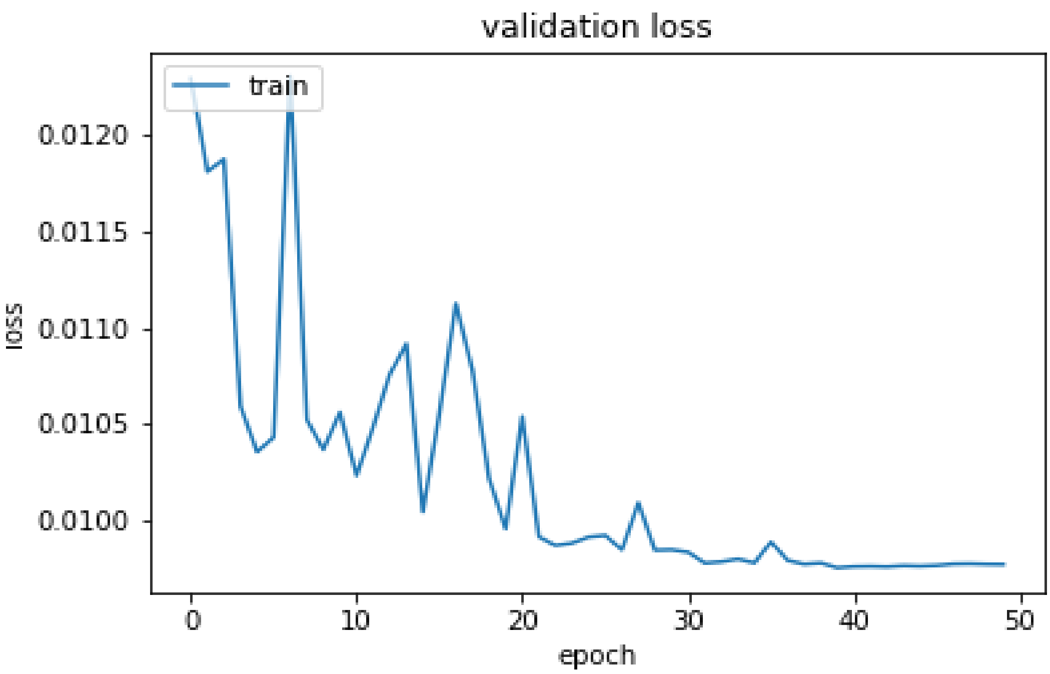


Figure 6.8 Graph showing variation of validation data with the increasing number of epochs.

- For Efficient Subpixel Convolutional Neural Network, variation of mean square error and validation loss with the increasing number of epochs are shown below with the help of graphs

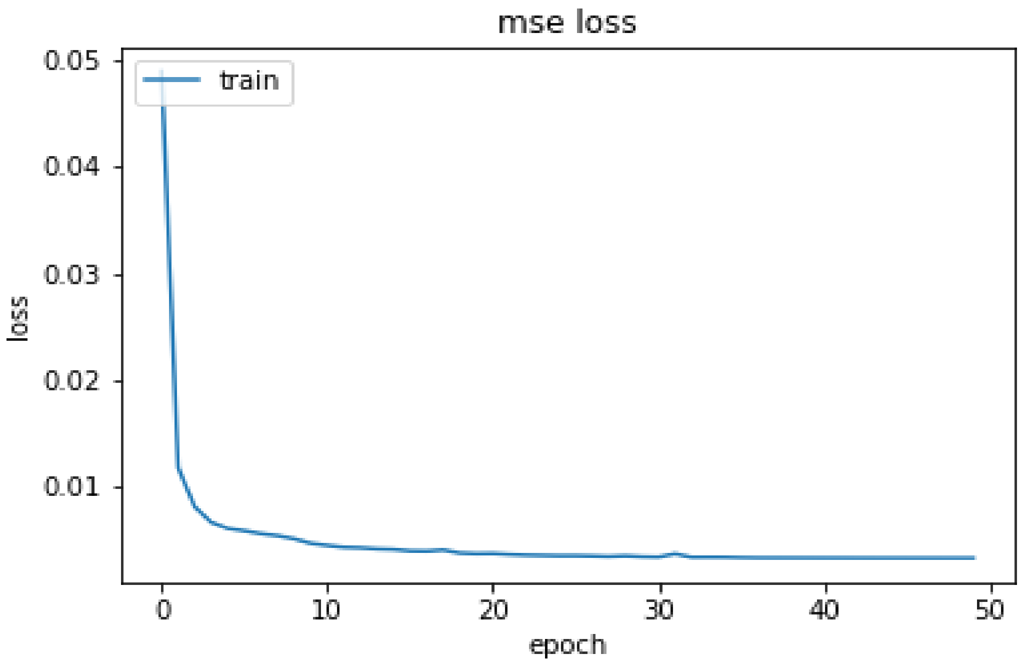


Figure 6.9 Graph showing convergence of mean square error with increasing number of epochs.

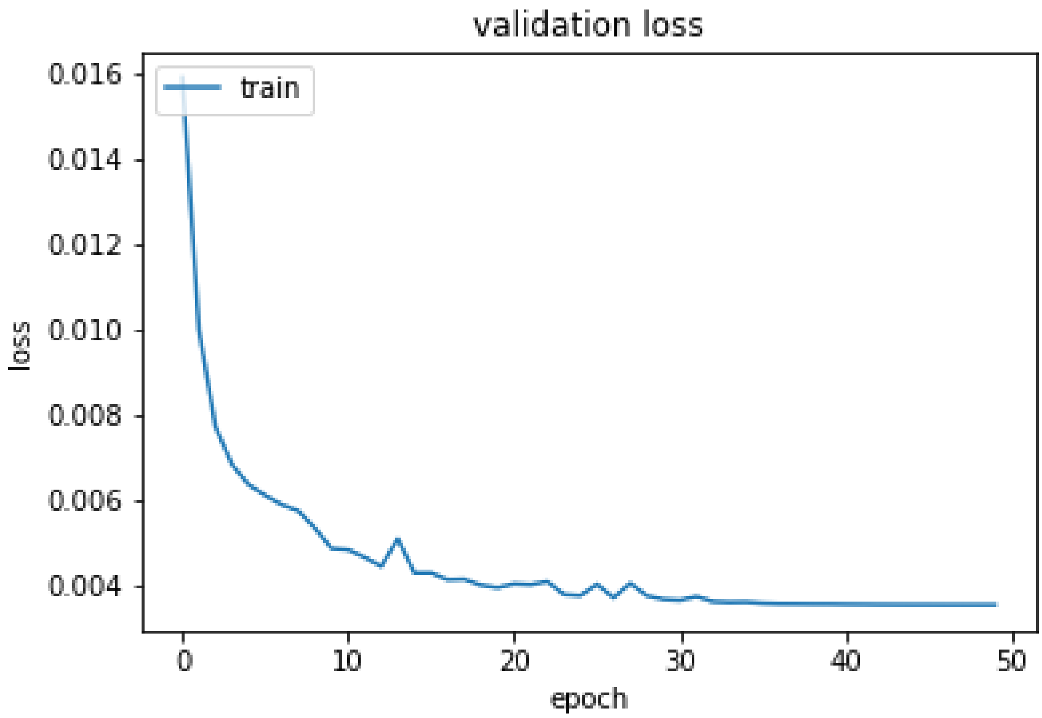


Figure 6.10 Graph showing variation of validation data with the increasing number of epochs.

- For Double Layered Subpixel Convolutional Neural Network, variation of mean square error and validation loss with the increasing number of epochs are shown below with the help of graphs.

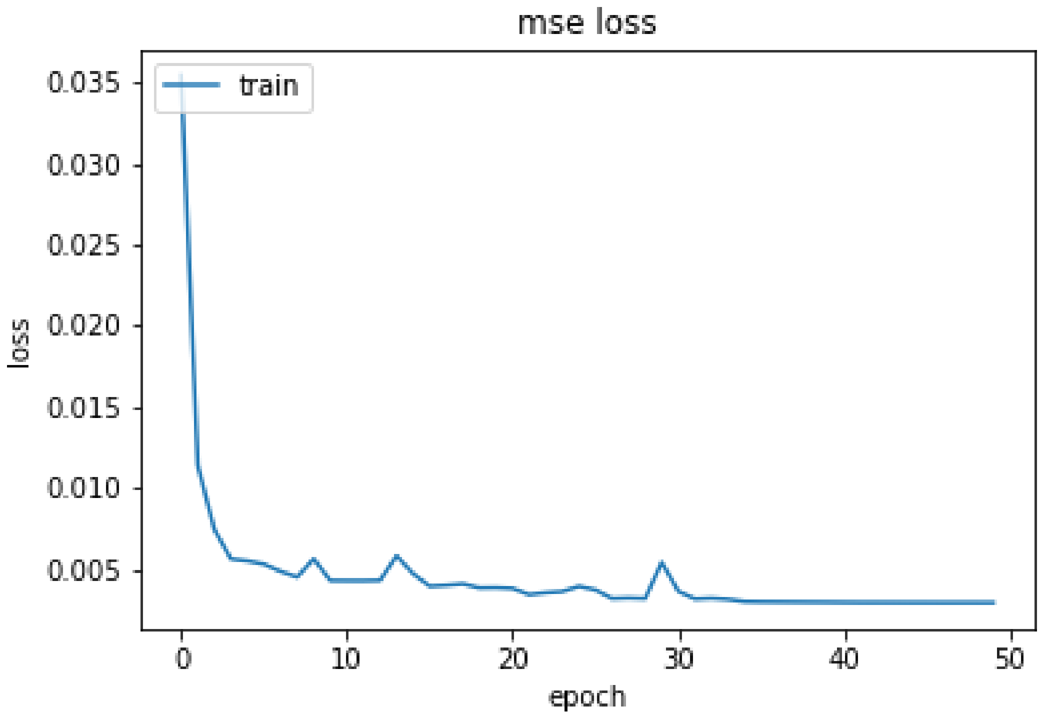


Figure 6.11 Graph showing convergence of mean square error with increasing number of epochs.

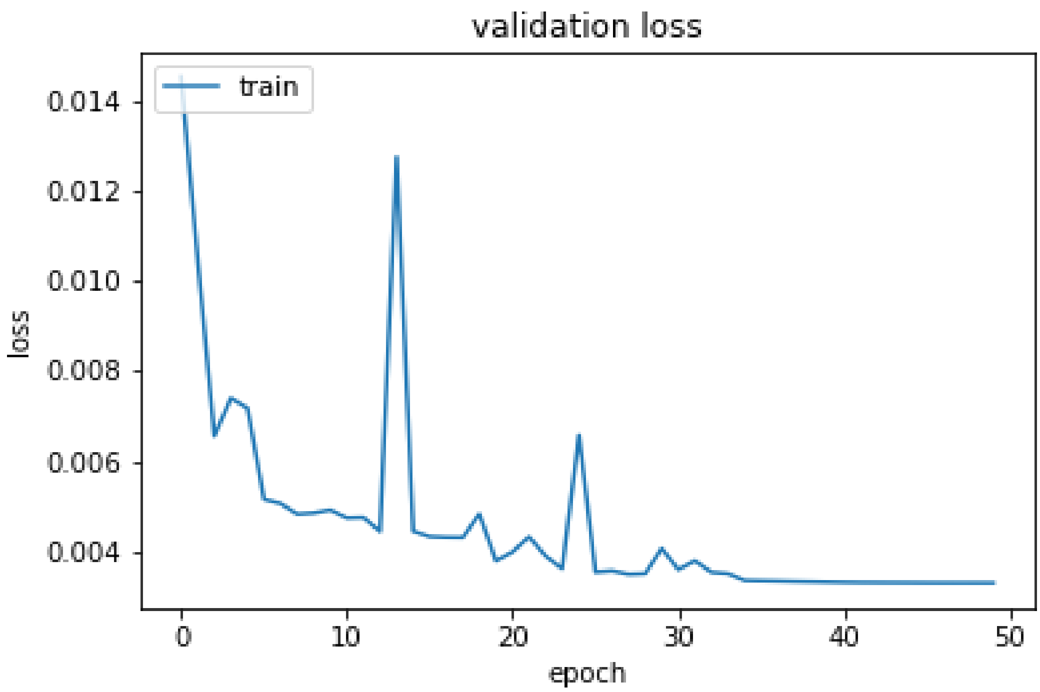


Figure 6.12 Graph showing variation of validation data with the increasing number of epochs

- For Double Layered Subpixel Convolutional Neural Network with 2-Stage Residual Learning, variation of mean square error and validation loss with the increasing number of epochs are shown below with the help of graphs

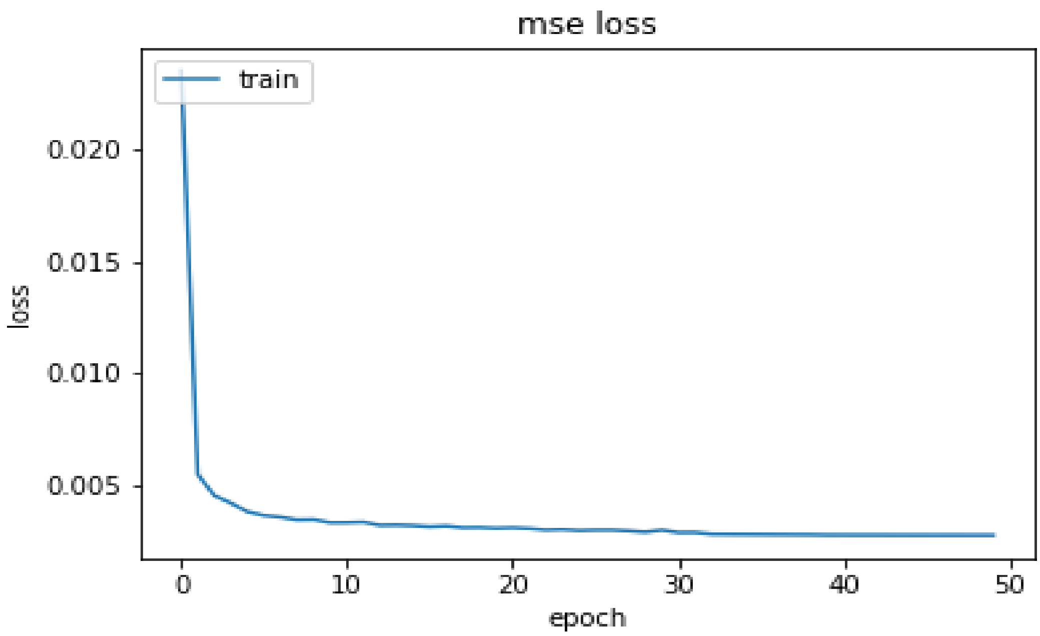


Figure 6.13 Graph showing convergence of mean square error with increasing number of epochs.

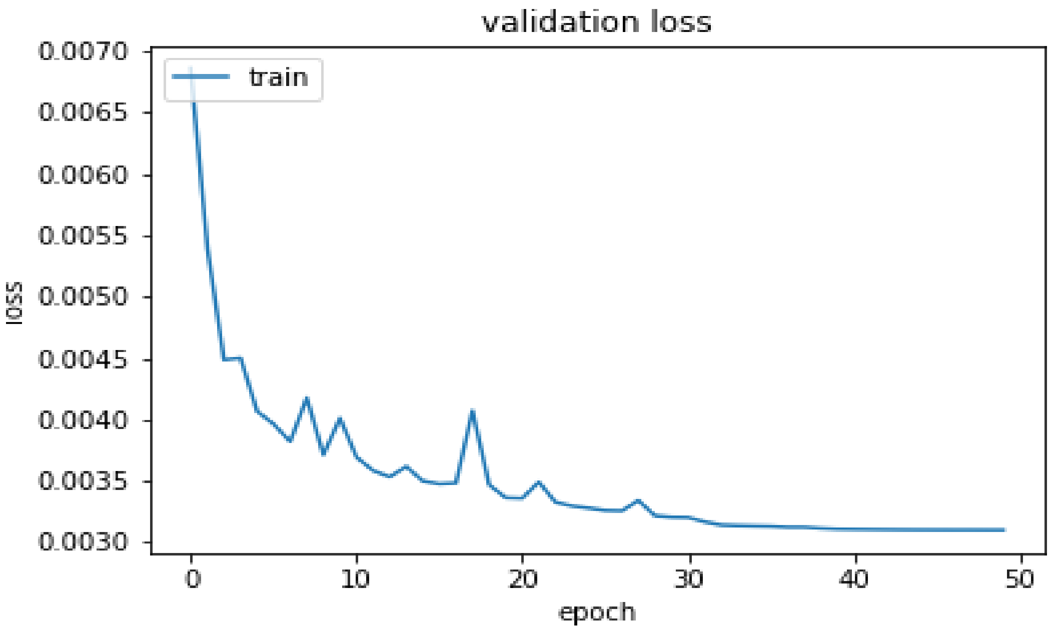


Figure 6.14: Graph showing variation of validation data with the increasing number of epochs.

CHAPTER -7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

As there are some limitations in the SR models based on deep learning, so we have proposed a novel two stage residual learning convolutional neural network for image super resolution which is more efficient model then the previous CNN models. We have approached our goal by re-designing the ESPCNN structure and further we have added residual learning to our architecture. so that our architecture is able to preserve high frequency details of the image. Experiments shows that the proposed model yields superior results in PSNR and SSIM values as comparison to the previous state-of-the-art methods. DLSRCNN have achieved an average gain of +0.43dB and DLSRCNN with 2-SRL have achieved an average gain of +0.89dB as compared to ESPCNN.

7.2 FUTURE SCOPES

- Images having high-resolution are highly required in every field like in medical imaging, remote sensing and surveillance etc.
- For diagnoses purpose in field of medical images like in Magnetic Resonance Imaging (MRI), Computer Tomography (CT) etc. DLSPCNN with 2-S RL architecture can play major role to obtain images with high-resolution.
- To analyze an object in an image has a requirement of zooming that part of an image which contains the object specifically in case of forensic, satellite imaging and surveillance, so DLSPCNN with 2-SRL can be applied here also to get a high-resolution image.
- SR techniques can also be applied to increase the resolution of video sequences.

REFERENCES

[1] Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence, 38*(2), 295-307.

[2] Dong, C., Loy, C. C., & Tang, X. (2016, October). Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision* (pp. 391-407). Springer, Cham.

[3] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., ... & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1874-1883).

[4] Huang, T., & Yang, J. (2010). Image super-resolution: Historical overview and future challenges. In *Super-resolution imaging* (pp. 19-52). CRC Press.

[5] Glasner, D., Bagon, S., & Irani, M. (2009, September). Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on* (pp. 349-356). IEEE.

[6] Available at: https://en.wikipedia.org/wiki/Machine_learning (Accessed on 20^{th} April 2018)

[7] Available at: https://www.researchgate.net/profile/Daniel_Caissie/Figure-1-Artificial-Neural-Network-representation (Accessed on 25^{th} April 2018)

[8] Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in bioinformatics*, *18*(5), 851-869.

[9] Available at: https://www.researchgate.net/profile/Nouman_Rasool/76357634@1455683835300/The-four-layer-architecture-of-MLP-with-backpropagation.png (Accessed on 25^{th} April 2018)

[10] Available at: https://encryptedtbn0.gstatic.com/images?q= tbn (Accessed on 30^{th} April 2018)

[11] Available at: http://pydeeplearning.weebly.com/ (Accessed on 15^{th} May 2018)

[12] Available at: http://d3kbpzbmcynnmx.cloudfront.net/wp-content/uploads/2015/11/Screen-Shot-2015-11-07-at-7.26.20-AM.png (Accessed on 19^{th} May 2018)

[13] Available at: http://cse19-iiith.vlabs.ac.in/theory.php (Accessed on 24^{th} May 2018)

[14] Available at: http://www.stratio.com/wp-content/uploads/2017/10/3.jpg (Accessed on 24th May 2018)

[15] Available at: https://qph.fs.quoracdn.net/main-qimg-8afedfb2f82f279781bfefa269bc6a90 (Accessed on 25th May 2018)

[16] Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-unrolled.png (Accessed on 25th May 2018)

[17] Available at: https://www.semanticscholar.org/paper/A-Unified-Tagging- %3A-Bidirectional-LSTM-with-Wang-Hong/191dd7df9cb91ac22f56ed0dfa4a5651e8767a51/ (Accessed on 26th May 2018)

[18] Salvador, J. (2016). *Example-Based Super Resolution*. Academic Press.

[19] Lehmann, T. M., Gonner, C., & Spitzer, K. (1999). Survey: Interpolation methods in medical image processing. *IEEE transactions on medical imaging*, *18*(11), 1049-1075.

[20] Courtesy of MathWorks - MATLAB and Simulink for Technical Computing. Available at :http://www.mathworks.com/ (Accessed on 26th May 2018)

[21] Sachs, J. (2001). Image Resampling. *Digital Light & Color*, *14*.

[22] C. Gonzalez, E. Woods, L. Eddins.(2010). Digital Image Processing Using Matlab. 2e. Gatesmark Publishing.

[23] Gavade, Anil & Sane, Prasana. (2013). super resolution image reconstruction by using bicubic interpolation. 10.13140/RG.2.1.4909.0401.

[24] Ur, H., & Gross, D. (1992). Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*, *54*(2), 181-186.

[25] Bose, N. K., & Ahuja, N. A. (2006). Super-resolution and noise filtering using moving least squares. *IEEE Transactions on Image Processing*, *15*(8), 2239-2248.

[26] Irani, M., & Peleg, S. (1991). Improving resolution by image registration. *CVGIP: Graphical models and image processing*, *53*(3), 231-239.

[27] Patti, A. J., Sezan, M. I., & Tekalp, A. M. (1997). Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE transactions on image processing*, *6*(8), 1064-1076.

[28] Chi-Chou Kao, Yen-Tai Lai & Chao-Feng Tseng (2016) Improved edge-directed super resolution, International Journal of Computers and Applications, 37:3-4, 160-167.

[29] Fattal, R. (2007, August). Image upsampling via imposed edge statistics. In *ACM transactions on graphics (TOG)* (Vol. 26, No. 3, p. 95). ACM.

[30] Sun, J., Xu, Z., & Shum, H. Y. (2011). Gradient profile prior and its applications in image super-resolution and enhancement. *IEEE Transactions on Image Processing*, *20*(6), 1529-1542.

[31] Protter, M., & Elad, M. (2009). Super resolution with probabilistic motion estimation. *IEEE Transactions on Image Processing*, *18*(8), 1899-1904.

[32] Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y., & Katsaggelos, A. K. (2009). Softcuts: a soft edge smoothness prior for color image super-resolution. *IEEE Transactions on Image Processing*, *18*(5), 969-981.

[33] Moitra (2016), S. Single-Image Super-Resolution Techniques: A Review. In *International Journal of Scientific & Engineering Research* (Vol. 3), 2395-1052

[34] Li, X., & Orchard, M. T. (2001). New edge-directed interpolation. *IEEE transactions on image processing*, *10*(10), 1521-1527.

[35] Fattal, R. (2007, August). Image upsampling via imposed edge statistics. In *ACM transactions on graphics (TOG)* (Vol. 26, No. 3, p. 95). ACM.

[36] Freeman, W. T., Jones, T. R., & Pasztor, E. C. (2002). Example-based super-resolution. *IEEE Computer graphics and Applications*, *22*(2), 56-65.

[37] Lin, W. T., & Lai, S. H. (2013, November). Single image super-resolution based on local self-similarity. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on* (pp. 191-195). IEEE.

[38] Chang, H., Yeung, D. Y., & Xiong, Y. (2004, June). Super-resolution through neighbor embedding. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (Vol. 1, pp. I-I). IEEE.

[39] Yu, W. S., & Chen, S. Q. (2011). An Improved Neighbor Embedding Method to Super-resolution Reconstruction of a Single Image. *Procedia Engineering*, *15*, 2418-2422.

[40] J. Yang, J. Wright, T. Huang, Y. Ma. "Image super resolution as sparse representation of raw image patches".In Proc. CVPR, 2008.

[41] Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE transactions on image processing*, *19*(11), 2861-2873.

[42] Aharon, M., Elad, M., & Bruckstein, A. (2006). $ rm k $-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, *54*(11), 4311-4322.

[43] Glasner, D., Bagon, S., & Irani, M. (2009, September). Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on* (pp. 349-356). IEEE.

[44] Cui, Z., Chang, H., Shan, S., Zhong, B., & Chen, X. (2014, September). Deep network cascade for image super-resolution. In *European Conference on Computer Vision* (pp. 49-64). Springer, Cham.

[45] Eigen, D., Krishnan, D., & Fergus, R. (2013, December). Restoring an image taken through a window covered with dirt or rain. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (pp. 633-640). IEEE.

[46] Jain, V., & Seung, S. (2009). Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems* (pp. 769-776).

[47] Burger, H. C., Schuler, C. J., & Harmeling, S. (2012, June). Image denoising: Can plain neural networks compete with BM3D?. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 2392-2399). IEEE.

[48] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

[49] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[50] He, K., & Sun, J. (2015, June). Convolutional neural networks at constrained time cost. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (pp. 5353-5360). IEEE.

[51] Dai, D., Timofte, R., & Van Gool, L. (2015, May). Jointly Optimized Regressors for Image Super-resolution. In *Computer Graphics Forum* (Vol. 34, No. 2, pp. 95-104).

[52] Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., & Fergus, R. (2014). Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in neural information processing systems* (pp. 1269-1277).

[53] Du, B., Xiong, W., Wu, J., Zhang, L., Zhang, L., & Tao, D. (2017). Stacked convolutional denoising auto-encoders for feature representation. *IEEE transactions on cybernetics*, *47*(4), 1017-1027.

[54] Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(10), 6232-6251.

[55] He, K., Zhang, X., Ren, S., & Sun, J. (2014, September). Spatial pyramid pooling in deep convolutional networks for visual recognition. In *european conference on computer vision* (pp. 346-361). Springer, Cham.

[56] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1646-1654).

[57] Available at: https://towardsdatascience.com/types-of-optimization-algorithms-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f (Accessed on 27th May 2018)

[58] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).

m

ORIGINALITY REPORT

8%	4%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ifp.illinois.edu Internet Source	1%
2	"Image and Graphics", Springer Nature, 2017 Publication	1%
3	Jae-Seok Choi, Munchurl Kim. "A Deep Convolutional Neural Network with Selection Units for Super-Resolution", 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 Publication	<1%
4	Liu, Zhang Huang, Qi Li, Jian Wang, Qi. "Single image super-resolution via [L.sub.0] image smoothing.(Research Article)(Report)", Mathematical Problems in Engineering, Annual 2015 Issue Publication	<1%
5	Yuan Yuan, Xiangtao Zheng, Xiaoqiang Lu. "Hyperspectral Image Superresolution by Transfer Learning", IEEE Journal of Selected Topics in Applied Earth Observations and	<1%