PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University of Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculty of Science and Technology

Department of Electronics

Master Thesis

Presented to obtain

THE MASTER'S DEGREE

Filiere: Telecommunications

Specialty: Telecommunication Systems

By

- **Ms. Bouaziz Chaima**
- **Mr. Sobhi Mohammed Ali**

Intitled

A Comparative Study of Conventional Interpolation Image Super Resolution Algorithms

Before the Jury composed of:

First Name & Family Name	*Tytle*	*Quality*	*University*
M. Hacine Gharbi Abdennour	***MCA***	***President***	***Univ-BBA***
***M.* Asbai Nassim**	***MCA***	***Examinator***	***Univ-BBA***
***Mss.* Messali Zoubeida**	***Prof.***	***Supervisor***	***Univ-BBA***
M. Bouechiche D.Eddine	***MCB***	***Co-supervisor***	***Univ BBA***

The Academic Year 2022/2023

Acknowledgment

Tous d’abord , nous remercions ***Allah*** qui m’a donné la force et la patience pour terminer ce travail dans le temps

Nous adressons nos sincéres remerciments à : nos encadreurs ***‘ Pr MESSALI ZOUBEIDA’***et ***‘Dr BOUDECHICHE DJAMEL’*** et ***‘Dr SID AHMED SOUMIA’***, qui nous ont supporté dans les moments les plus difficiles pour terminer ce travail avec leurs disponibilité et leurs precieux conceils, pour leurs savoir faire sur le plan théorique et pratique, pour savoir débloquer les difficultés, un grand merci.

Pour tous les enceignants ***du départements d’électronique*** et en particulier la branche ***systéme des télécomunications***.

Pour tous les membres de ***la famille*** qui m’ont supporté moralement et pour leurs priéres sincéres . Tous ***nos amis*** et ***cammarades de la promotion***.

Nous exprimons tout notre respect ***aux jurys*** qui feront l’honneur d’apprécier ce travail

Dedicaces

Je dédie ce modeste travail ***à mes chers parents*** qui n'ont pas cessé de mencourager durant toutes mes études et qui m'accompagnés dans toutes ma vie , s'inquiétant énormémément pour m'offrir une meilleure vie . Que Dieu les protége

A mes fréres Abd almadjid et messeaud

A mes sœurs Samira et Chafia

Pour leurs soutiens moral et leurs conseils précieux tout au long de mes études

A toute ma famille

A mes chers amis Ihab et Ansar et Racha , Belquess

Pour leurs aides et supports dans les moments difficiles

Un merci spécial a mon binome ***Sobhi Mohamed Ali***

Enfin , je dédie ce travail à tous ceux qui me connaissents de prés ou de loin

Mlle Bouaziz Chaima

Dedicaces

Je dédie ce modeste travail ***à mes chers parents*** qui n'ont pas cessé de mencourager durant toutes mes études et qui m'accompagnés dans toutes ma vie , s'inquiétant énormémément pour m'offrir une meilleure vie . Que Dieu les protége

A Mes Soeurs et leurs fils Meriem et ***Lilya*** et ***Nour*** et ***Yahia***

Pour leurs soutiens moral et leurs conseils précieux tout au long de mes études

A mes chers amis Mohammed,Ala,Hamza,Chouaib,Hamouka,Aymen,Fouad ,Islem *et* ***Abdou***

Pour leurs aides et supports dans les moments difficiles

La Famille de { ELECTRO MINDS CLUB-نادي العقول الالكترونية}

{Rassemblement des Etudiants Algériens Libres- تجمع الطلبة الجزائريين الاحرار }

{ SHINGTHAI KARATE DO- شينغتاي كاراتي دو }

Pour leur soutien et assistance lors des périodes difficiles.

Au **Cheikh Abd al-Aziz al-Khatib al-Hassani** en Syrie

Pour son soutien spirituel et religieux

Un remerciement spécial à mon binôme **Bouaziz Chaima**

Enfin , je dédie ce travail à tous ceux qui me connaissents de prés ou de loin

Sobhi Mohammed Ali

Abstract

This master thesis deals with image super resolution (ISR) problem. We focus on the differences between conventional image interpolation algorithms and deep learning-based algorithms. The considered SR algorithms are applied on the most used datasets in this field. Many simulation experiments are conducted to higlight the efficiency of deep learning SR algorithms. The original images are considered as High resolution (HR) images. Low resolution (LR) images are generated from HR ones, by simple downsampling. Performance assessment of the different SR algorithms is in terms of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) in addition to the visual quality of the obtained SR image. From the obtained results, deep learning SR algorithms allow a significant enhancement of the LR images and are the best algorithms in terms of PSNR and SSIM.

Résume :

Ce mémoire de maîtrise traite le problème de la super résolution d'image (ISR). Nous nous concentrons sur les différences entre les algorithmes d'interpolation d'images conventionnels et les algorithmes basés sur l'apprentissage profond. Les algorithmes SR considérés sont appliqués sur les ensembles de données les plus utilisés dans ce domaine. De nombreuses expériences de simulation sont menées pour évaluer l'efficacité des algorithmes SR d'apprentissage profond. Les images originales sont considérées comme des images haute résolution (HR). Les images basse résolution (LR) sont générées à partir des images HR, par simple échantillonnage. L'évaluation des performances des différents algorithmes SR se fait en termes de Peak Signal to Noise Ratio (PSNR) et Structural Similarity Index (SSIM) en plus de la qualité visuelle de l'image SR obtenue. A partir des résultats obtenus, les algorithmes SR de deep learning permettent une amélioration significative des images LR et sont les meilleurs algorithmes en termes de PSNR et SSIM..

ملخص:

تتناول هذه الرسالة الرئيسية مشكلة الدقة الفائقة للصورة (ISR). نركز على الاختلافات بين خوارزميات استيفاء الصور التقليدية والخوارزميات القائمة على التعلم العميق. يتم تطبيق خوارزميات SR المدروسة على مجموعات البيانات الأكثر استخدامًا في هذا المجال. يتم إجراء العديد من تجارب المحاكاة لتسليط الضوء على كفاءة خوارزميات التعلم العميق SR.

تعتبر الصور الأصلية صورًا عالية الدقة (HR). يتم إنشاء صور منخفضة الدقة (LR) من صور الموارد البشرية، عن طريق التقليل البسيط. يتم تقييم أداء خوارزميات SR المختلفة من حيث نسبة إشارة الذروة إلى الضوضاء (PSNR) ومؤشر التشابه الهيكلي (SSIM) بالإضافة إلى الجودة المرئية لصورة SR التي تم الحصول عليها. من النتائج التي تم الحصول عليها، تسمح خوارزميات التعلم العميق SR بتحسين كبير لصور LR وهي أفضل الخوارزميات من حيث PSNR و SSIM..

List of Figures

List of Tables

List of Abbreviations

AI Artificial intelligence

BCI Cubic Interpolation

BLI Bilinear Interpolation

CAR Content Adaptive Resample

CNN Convolutional Neural Network

DL Deep Learning

EDSR Enhanced Deep Super Resolution

HR High Resolution

LR Low Resolution

MSE Mean Square Error

ML Machine Learning

MOS Mean Opinion Score

NN's Neural Networks

NIQE Naturaleness Image Quality Evaluator

PSNR Peak Signal-to-Noise Ratio

ReLU Rectified Linear Unit

SR Super Resolution

SSIM Similarity Structural Index Metric

VDSR Very Deep Super Resolution

Table of Contents

General introduction

In recent years [1], the demand for high-resolution images has significantly increased due to the rapid advancement of imaging devices and the proliferation of multimedia applications. However, capturing high-resolution images can be challenging and costly. This challenge has led to the development of image super-resolution (SR) techniques, which aim to enhance the resolution of low-resolution images and provide visually appealing and detailed results.

Interpolation-based algorithms have long been used as conventional approaches for SR [1] as they are computationally efficient and easy to implement. These methods rely on interpolation techniques to estimate the missing high-frequency information in low-resolution images. However, their performance is often limited in preserving fine details and generating realistic textures.

Deep learning-based approaches [2] on the other hand, have revolutionized various computer vision tasks, including image super-resolution. By leveraging the power of deep neural networks, these algorithms have demonstrated remarkable performance improvements in terms of reconstructing high-resolution details and producing visually enhanced results. They learn from large-scale datasets (big data) and exploit the inherent patterns within images to generate high-resolution outputs (estimated version of Low Resolution (LR) images).

The objective of this End of our Study Project is to conduct a comparative study between conventional interpolation-based algorithms and deep learning-based algorithms for image super-resolution. By evaluating and analyzing the performance of these techniques, we aim to gain insights into their strengths, weaknesses, and areas of applicability.

The manuscript is organized on this general introduction and three chapters in addition to a general conclusion.

In chapter one: we present the basic concepts of the filed of Image super resolution

In chapter two: we give the most used terminology of the field of deep learning;

In chapter three: we explain our imlementation to three deep learning image super resolution networks, namely: very deep learning SR VDSR, Enhanced SR EDSR and Content adaptive resampler (CAR netwoks in addition to three conventional SR algorithms: bicubic, bilinear and nearest neigbord. We give the global remarques in the general conclucion.

Chapter 1

Basic Concepts of Super-Resolution

Abstract

This chapter focuses on the process of image super-resolution (SR) enhancement. Initially, we will provide an introduction to the key concepts underlying the SR process. Subsequently, we will delve into the explanation of three SR algorithms that are rooted in deep learning principles.

Summary

1.1 Introduction :

Image super-resolution (SR) is an important task in image processing methods that improve the resolution of input images. In the last two decades [3] significant progress has been made in the super-resolution (SR) field [3], especially by using deep learning methods.
So, Super-resolution is a challenging problem in image processing that involves increasing the spatial resolution of an image beyond the original resolution. There are many applications of super-resolution, including medical imaging, surveillance, and video processing. Recent advances in deep learning have led to the development of powerful super-resolution methods based on different deep learning networks such as: Very deep Learning Network (VDSR) [4], Convolutional Neural Networks (CNNs) [5], Enhanced deep learning SR (EDSR) [6] and Content Adaptive Resolution (CAR) [7] networks. Overall, Super-Resolution is an important and active area of research with many practical applications. While there is no one-size-fits-all solution to the super-resolution problem, a combination of different methods can be used to achieve high-quality and efficient super-resolution for various applications.
In our work, we have first implemented the conventional interpolation algorithms, namely: bicubic, bilinear and nearest niegbord algorithms. Then, we have implemented VDSR, EDSR and CAR networks as we will detailed in the next sections.

So, this chapter describes the basic concepts of image super-resolution. We first present problem formulation and the conventional Image super-resolution algorithms. A focus on the Bicubic SR algorithm will be discussed. The SR conventional algorithms will be classified into three categories, conventional algorithms, supervised learning-based algorithms, and unsupervised learning-based algorithms. Finally, we introduce the mostly used metrics in image super-resolution, namely: Mean square error (MSE), Peak Signal to Noise Ratio (PSNR), Similarity structural Index Metric (SSIM). and Natureleness Image Quality Evaluation (NIQE). PSNR and SSIM will be used in our simulation study to establish a quantitative qualitative comparison of the considered SR algorithms.

1.2 Problem formulation of image super-resolution process :

Image super-resolution (SR) refers to the process of reconstructing a high-resolution (HR) image from a low-resolution (LR) image input [8]. The LR image, denoted as $\boldsymbol{I_{xLR}}$, is typically the output of a degradation function, as demonstrated in the equation 1.1:

$$\boldsymbol{I_{xLR}} = \boldsymbol{d}(\boldsymbol{I_{yHR}}, \boldsymbol{\delta}) \quad 1.1$$

d Is the degradation function in image super-resolution (SR), represents the operator for converting a high-resolution (HR) image to a low-resolution (LR) image. The input HR image, or reference image, is denoted as $\boldsymbol{I}_{\mathrm{yHR}}$, while $\boldsymbol{\delta}$ represents the input parameters of the degradation function, such as scaling factor, blur, or noise. In practice, the degradation process and its parameters are typically unknown [1], and only LR images are available to obtain SR images through HR algorithms. The SR process aims to predict the inverse of the degradation function, $\boldsymbol{d}$, to reconstruct the HR image from the LR input.

$$\boldsymbol{g}\ (\boldsymbol{I_{xLR}}, \boldsymbol{\delta}) = \boldsymbol{d^{-1}}(\boldsymbol{I_{xLR}}) = \boldsymbol{I_{yE}} \approx \boldsymbol{I_{ySR}} \quad 1.2$$

The function $\boldsymbol{g}$ represents the SR function (or the super resolution algorithm),, where $\boldsymbol{d}$ represents the input parameters of the function $\boldsymbol{g}$, and $\boldsymbol{I_{yE}}$ denotes the estimated HR value corresponding to the input $\boldsymbol{I_{xLR}}$ image.

The degradation process of the input LR images is influenced by various factors like sensor-induced noise, loss compression-induced artifacts Gaussian or, speckle noise, motion blur (in video frames), and misfocused images, and the exact process is unknown. In our study only, a single downsampling function is considered as the image degradation function, such that:

$$\boldsymbol{d}(\boldsymbol{I_{yHR}}, \boldsymbol{\partial}) = (\boldsymbol{I}_{\mathbf{yHR}}) \downarrow \boldsymbol{s_f}\ , \{\boldsymbol{s}\} \subseteq \boldsymbol{\partial} \quad 1.2$$

↓: denotes the downsampling operator by a scalar factor $\boldsymbol{s_f}$ **(**$\boldsymbol{s_f = 2\ or\ 4}$**).**

1.3 Conventional SR Algorithms:

Image interpolation, also known as image upscaling, is a commonly used technique in various image-related applications. Interpolation methods, such as nearest neighbor interpolation, linear interpolation, and cubic interpolation, are conventional methods used for this purpose.

Bicubic interpolation (BCI) is another form of cubic interpolation that takes into account 4×4 pixels on both axes, resulting in smoother results with fewer artifacts than bilinear interpolation (BLI). However, BCI is slower due to its increased computation complexity [1].

1.3.1 Bilinear interpolation :

Bilinear interpolation is a conventional interpolation method commonly used in image processing to increase the resolution of an image. It is a relatively simple method that estimates the color value of a missing pixel by using the color values of the four closest pixels to the missing pixel.

To perform bilinear interpolation, the color values of the two closest pixels in each direction (horizontal and vertical) are first averaged. The resulting two averages are then averaged again to produce the estimated color value for the missing pixel. This technique works well for images with smooth color gradients or transitions.

While bilinear interpolation can produce smoother images than nearest-neighbor interpolation, it can result in blurred edges or loss of details. This is because bilinear interpolation assumes that the color values change linearly between pixels, which may not always be the case in real-world images.

However, modern super-resolution techniques based on, deep learning-based methods, have surpassed conventional interpolation algorithm in terms of generating high-quality images with enhanced details

Nearest neighbor interpolation, as shown on Figure 1.1, is a simple and fast method of interpolation used in image processing and super-resolution. In this method, the value of a new pixel in the high-resolution image is assigned to be the value of the nearest pixel in the low-resolution image. While nearest neighbor interpolation is fast and easy to implement, it can result in a blocky or pixelated appearance, especially when the scale factor is high. This is because the high-resolution image is simply a replication of the low-resolution image, resulting in a loss of detail and resolution [9].

Therefore, nearest neighbor interpolation is generally not recommended for super-resolution tasks where high image quality is desired. However, it may be suitable for certain applications where speed is more important than image quality, such as real-time video processing. Figure 1.1 shows an example of nearest neighbor interpolation

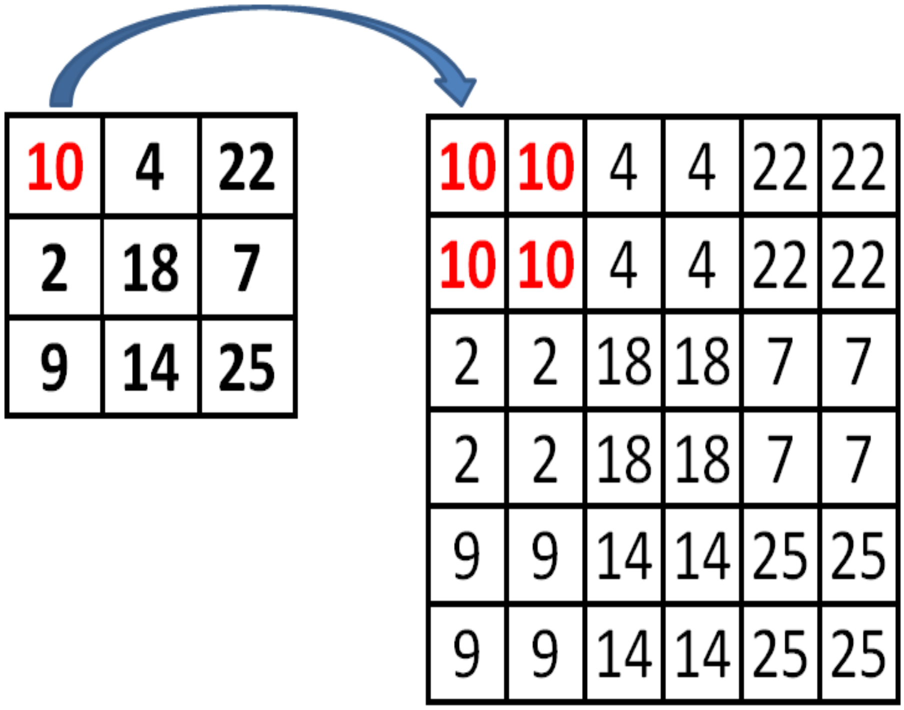


Figure 1.1 Nearest Neighbor Interpolation [10]

1.3.2 Linear interpolation

Linear interpolation is a common method of interpolation used in image processing and super-resolution. In this method, a new pixel in the high-resolution image is estimated as a weighted average of the surrounding pixels in the low-resolution image.

Linear interpolation is a simple and computationally efficient method of interpolation that can provide good results when the scale factor is small. However, it can result in a loss of detail and resolution when the scale factor is large, as it does not take into account higher-order information such as texture or patterns in the image [11]. Equation 1.5 describes the rule of computing the new pixel:

$$\boldsymbol{y = y_1 + (x - x_1)\frac{(y_2 - y_1)}{(x_2 - x_1)}} \qquad 1.3$$

Whe

- x_1 and y_1 are the first coordinates
- x_2 and y_2 are the second coordinates

- x is the point to perform the interpolation
- y is the interpolated value

1.3.3 Spline interpolation

Spline interpolation is a method of interpolation used in image processing and super-resolution that involves fitting a piecewise polynomial curve through the image pixels to estimate the missing values.

In spline interpolation, a set of control points is selected from the low-resolution image, and a piecewise polynomial curve is fitted through these points. The resulting curve is then used to estimate the missing pixel values in the high-resolution image. It Can provide smoother and more accurate results than linear or cubic interpolation, especially when the scale factor is large. This is because the piecewise polynomial curve which can capture higher-order information such as texture and patterns in the image, resulting in a more realistic and detailed super-resolved image.

However, spline interpolation can be computationally intensive and may require a larger number of control points than other methods. It can also introduce artifacts and noise in the image if the control points are not well chosen [12].

1.4 Super-Resolution (SR) Algorithms Based on Deep Neural Networks

Deep learning has numerous advantages when it comes to super-resolution (SR) image processing [2]. It enables the learning of complex features at various levels of abstraction, can handle large datasets, can handle non-linear relationships, and can be optimized end-to-end. These advantages allow deep learning models to produce more accurate, visually appealing, and natural-looking super-resolved images. Consequently, deep learning has significantly improved the accuracy and speed of SR algorithms and has revolutionized the field of SR image processing [13].

In the following, we will describe three of deep SR algorithms, namely: the Enhanced deep learning SR (EDSR) network, it’s improved version Content adaptive resampler (CAR) and the basic very deep learning SR network (VDSR). These networks have been implemented in our simulation experiments, in order to extract the limitations and advantages of each one.

1.4.1 Very Deep Super Resolution VDSR

VDSR is a deep learning-based method for image super-resolution based on Convolutional Neurol network architecture (CNN). It uses deep convolutional neural networks to generate high-resolution images from low-resolution inputs. By stacking numerous convolutional layers, VDSR captures complex relationships between LR and HR images, enhancing fine textures and sharp edges. During training, it minimizes the difference between predicted and ground truth images. VDSR is versatile, handles various upscaling factors, and benefits from parallel computing for faster processing. It has shown exceptional performance, making it valuable for tasks like digital photography, medical imaging, and video processing [4]. Figure 1.2 shows the architecture of VDSR network, where $\boldsymbol{x}, \boldsymbol{r}$ *and* $\boldsymbol{y}$ are LR, residual and SR image respectively. It consists on a consecutive layers of convolution (Convi) and Non linear layers (activation function ReLui). LR image is added to the extracted risual image to obtain the SR image.

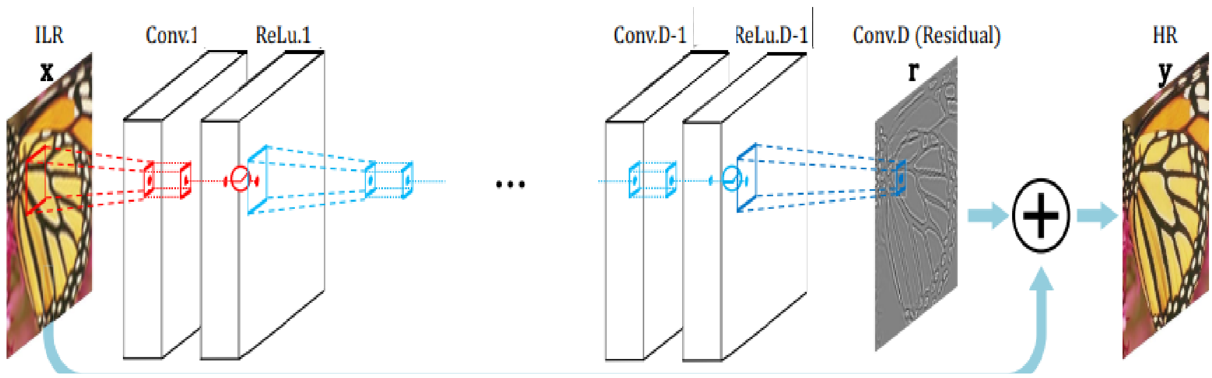


Figure 1.2: Architecture of VDSR

1.4.2 Enhanced Deep Super-Resolution Network Algorithm (EDSR)

The EDSR algorithm is a deep learning-based super-resolution network that has achieved notable performance improvements. These enhancements are attributed to the optimization process, which involves removing redundant modules from conventional residual networks. Additionally, the EDSR algorithm benefits from the expansion of its model size, while ensuring stable training procedures, which further improves its performance [14]. Figure 1.3 shows the EDSR network architecture.

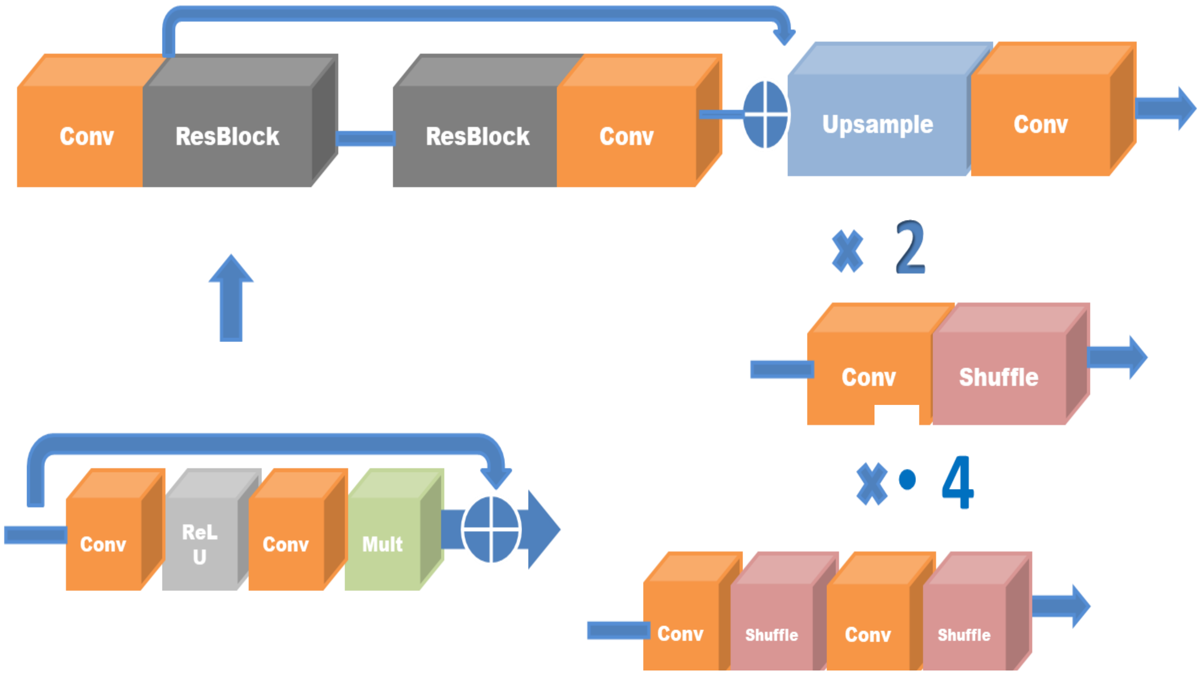


Figure 1.3: Architecture of EDSR

The framework consists of three main components as shown in Figure 1.3: the ResamplerNet, the Downscaling module, and the SRNet. The task of the ResamplerNet is for estimating content-adaptive resampling kernels and corresponding offsets for each pixel in the downscaled image. To train the ResamplerNet, the SRNet, which can be any differentiable upsampling operation, is used to guide the training process by minimizing the SR error (MSE). The entire framework is trained end-to-end by backpropagating error signals through the differentiable downscaling module. The composition of each building block is further detailed within the blue dashed frame [5] in Figure 1.4**.**

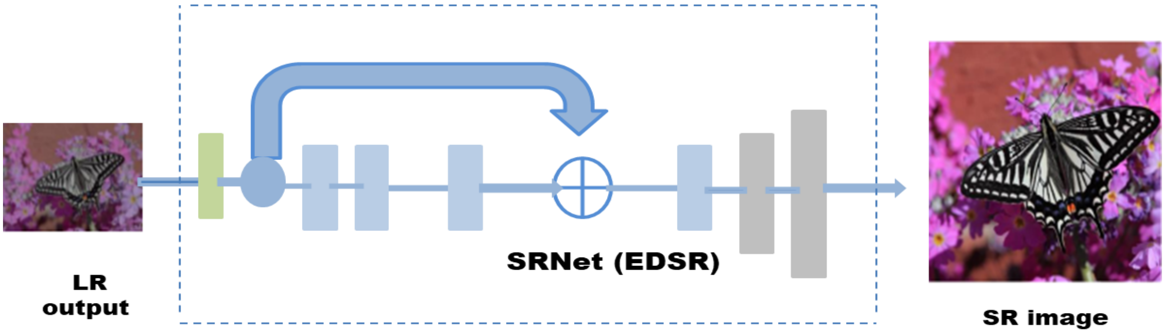


Figure 1.4: Training Objectives (Objective Function)

1.4.3 CAR Algorithm: Content Adaptive Resampler

In this section, we describe the Enhanced version of EDSR network:-Content Adaptive resamler (CAR) network,–in which the image upscaling module is used to guide the training of the Content Adaptive Resampler (CAR) model. CAR is a learned image downscaling technique that takes into account the upscaling process. The CAR model generates content adaptive resampling kernels for the original high-resolution (HR) input to generate pixels on the downscaled image. Additionally, a differentiable upscaling (SR) module is used to upscale the low-resolution (LR) result to its corresponding HR counterpart. By backpropagating the reconstruction error throughout the framework is used to adjust model parameters. This framework achieves a new state-of-the-art SR performance through upscaling-guided image resamplers, which adaptively preserve detailed information that is essential to the upscaling process [15]. Figure 1.5 shows the CAR network architecture, and Figure 1.6 illustrates the relationship between EDSR and CAR networks.

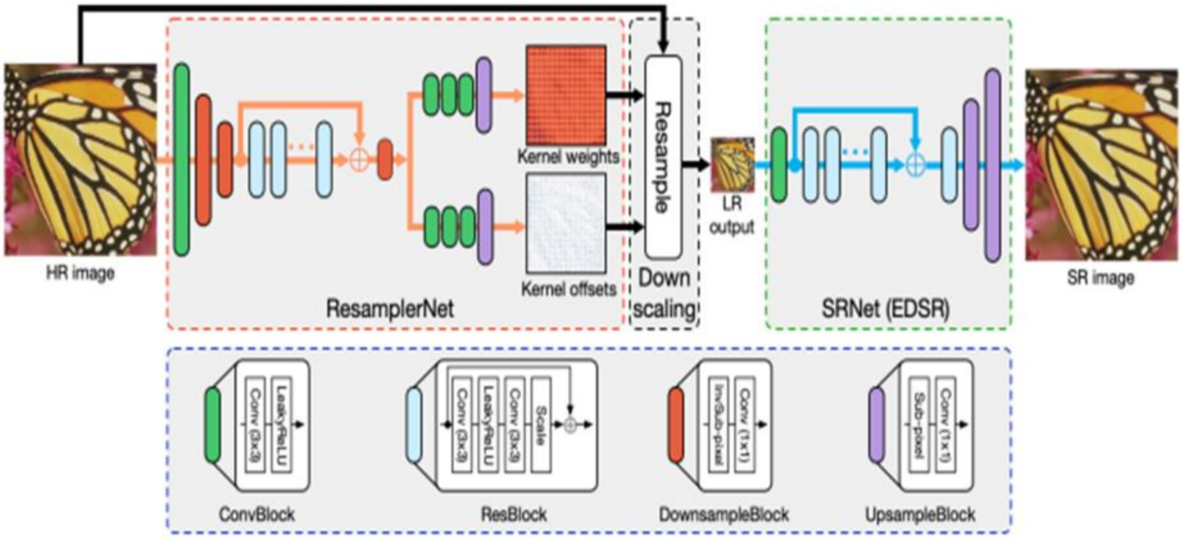


Figure 1.5: Architecture of CAR algorithm [16]

HR image → Generation of LR image → EDSR → Obtained SR

Figure 1.6: CAR architecture based on EDSR network

1.5 Performance Assessment

- **Mean Squared Error (MSE):** This metric measures the average squared difference between the original image and the processed image. It is a simple and widely used metric but does not always correlate with human perception.

$$\boldsymbol{MSE} = \frac{\sum \boldsymbol{M,N}[\boldsymbol{I_1(m,n)} - \boldsymbol{I_2(m,n)}]^2}{\boldsymbol{M \times N}} \qquad 1.4$$

Where $\boldsymbol{I_1}$ ***and*** $\boldsymbol{I_2}$ denote the originale HR image and the estimated SR image respectivelly of size $M \times N$ each.

- **Peak Signal-to-Noise Ratio (PSNR)**: This metric measures the ratio between the maximum possible value of a signal and the noise that affects the fidelity of its representation. It is commonly used to compare the quality of compressed images.

$$\boldsymbol{PSNR\ (dB) = 10 \times \log_{10} \frac{255^2}{MSE}} \qquad 1.5$$

- **Structural Similarity Index (SSIM):** This metric measures the structural similarity between the original image and the processed image. It takes into account the luminance, contrast, and structural information of the image and is considered to be a more accurate measure of image quality than MSE.

$$\boldsymbol{SSIM(x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad 1.6$$

With:

- $\boldsymbol{\mu_x}$ the pixel sample mean of x
- $\boldsymbol{\mu_y}$ the pixel sample mean of y
- $\boldsymbol{\sigma_x^2}$ the variance of x
- $\boldsymbol{\sigma_y^2}$ the variance of y
- $\boldsymbol{\sigma_{xy}}$ the covariance of x and y
- $\boldsymbol{C_1}$**:** Constant Correlation coefficient
- $\boldsymbol{C_2}$**:** Constant Correlation coefficient

- **Mean Opinion Score (MOS):** This metric measures the perceived quality of an image by asking human observers to rate the image on a scale from 1 to 5. It is a subjective metric and requires a large number of observers to obtain statistically significant results.

- **Natural Image Quality Evaluator (NIQE):** This metric is a no-reference metric that measures the quality of an image based on its naturalness and complexity. It is designed to be more correlated with human perception than other no-reference metrics.

1.6 Conclusion

In conclusion, this chapter has covered the basic concepts of Super-Resolution (SR) and provided an overview of different algorithms used in this process. The chapter began with the problem formulation of image super-resolution, highlighting the challenges and goals involved. It then explored conventional SR algorithms, which employ various techniques such as interpolation methods to enhance image resolution.

Furthermore, the chapter delved into the advancements brought about by deep neural networks in SR algorithms. These deep learning-based approaches leverage the power of machine learning to learn complex mappings between low-resolution and high-resolution image pairs. This enables the generation of visually plausible and detailed high-resolution outputs.

Overall, this chapter has laid the foundation for understanding the core principles and techniques used in the field of super-resolution. It has highlighted the importance of image information and machine learning algorithms in achieving high-quality super-resolution results. With further research and advancements in deep learning and computational resources, the field of super-resolution holds great potential for addressing real-world challenges and advancing various applications in computer vision and image processing.

Chapter 2

Principles of Deep Learning

Abstract

This chapter serves as an introduction to the fundamental concepts of machine learning and deep learning. Specifically, we will provide clear definitions of the terminology commonly used in these fields.

Summary

2.1 Introduction

Deep learning is a type of machine learning that uses artificial neural networks to learn from complex data and make predictions. It is especially useful for tasks involving large amounts of data, such as image and speech recognition, natural language processing, autonomous driving, and image super-resolution as we will use in our study. Deep learning models consist of layers of interconnected nodes that automatically learn features from raw data so they can recognize patterns and make accurate predictions. However, developing and optimizing deep learning models requires massive computing power, expertise, and training data [2]. In this Chapter, we will describe the basic concepts of deep learning. For this task, we will first introduce machine learning, then we study the most used architectures of deep learning (DL). We will define the different parameters of a DL network.

2.2Machine Learning (ML)

Machine learning (ML) refers to the process of teaching computers to learn from data using a variety of algorithms that can iteratively improve, explain, and predict outcomes. By absorbing and analyzing training data, machine learning algorithms are able to generate increasingly precise models. The output of this process is a machine learning model, which is the result of training the algorithm on the data [17]. The link between Artificial Intelligence (AI), ML, and DL with data science and computer vision is shown in Figure 2.1.

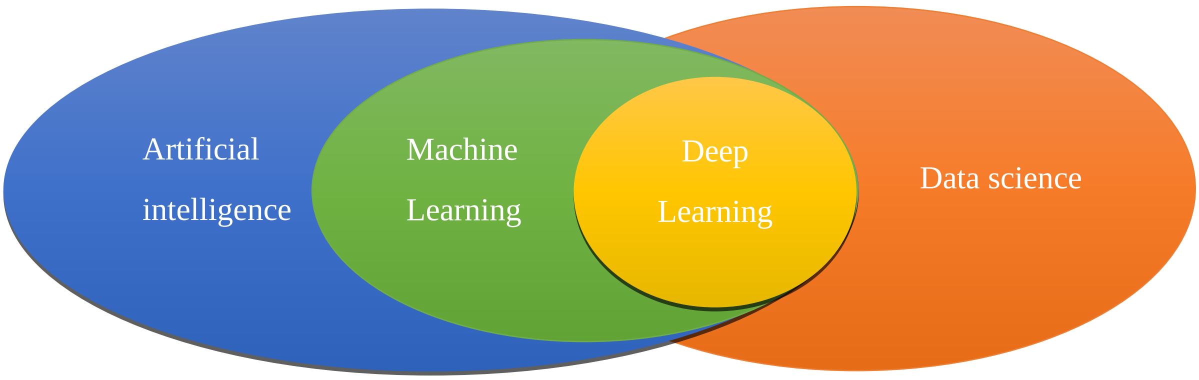


Figure 2.1: Diagram representing the relationships of AI-related domains.

There are two main categories of ML: supervised ML and unsupervised ML.

2.2.1 Supervised learning

Supervised Learning is the dominant learning paradigm in the fields of Machine Learning (ML) and Deep Learning (DL). This approach involves providing the machine learning algorithm with examples (data) of the task it needs to perform, allowing it to learn from these examples in a supervised manner. Supervised learning has a wide range of applications, including computer vision, regression, classification, and more. The majority of problems in Machine Learning and Deep Learning rely on supervised learning as the primary approach [18]

2.2.2Unsupervised learning

Unsupervised learning is particularly useful when dealing with vast amounts of unlabeled data. This approach is especially well-suited to applications such as social media platforms (e.g. Twitter, Snapchat) that generate massive amounts of unstructured data without pre-existing labels or categories.

Figure 2.2 shows supervised ML vs unsupervised ML.

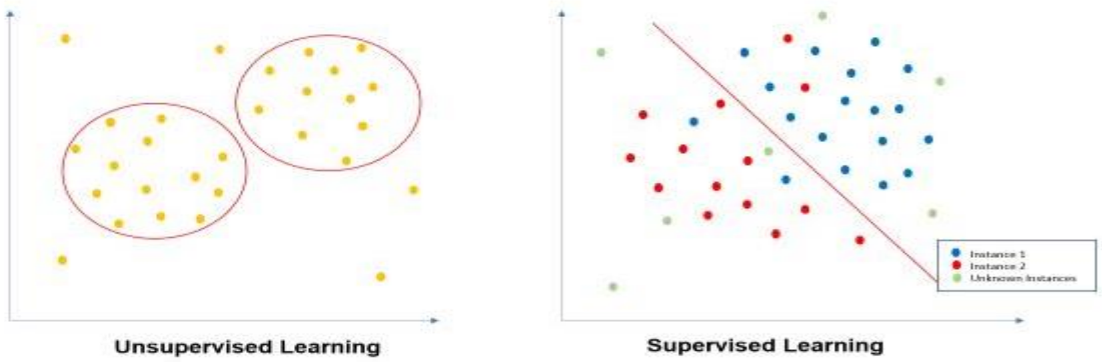


Figure 2.2: The dissimilarity between the two forms of learning.

2.3 Deep Learning (DL)

Deep learning is a type of artificial intelligence (AI) [3] that has evolved from machine learning, which is a form of automatic learning. Unlike following predetermined rules to the letter, deep learning allows machines to learn on their own. DL is a subfiled of ML.

2.4Neural Networks (NNs)

A neural network is composed of an input layer, one or more hidden layers, and an output layer. The input layer receives data, which is then transformed in the hidden and output layers using weights assigned to the nodes. A standard neural network can have thousands or even millions of interconnected simple processing nodes [19]

2.4.1The architecture of neural networks

The architecture of a neural network refers to its overall structure and how its components are organized and connected. At the most basic level, a neural network is made up of layers of interconnected nodes or neurons. These neurons receive input data, perform mathematical operations on that data, and then pass the results to other neurons in the network [19]**.** Figure 2.3 shows this architecture with one hidden layer.

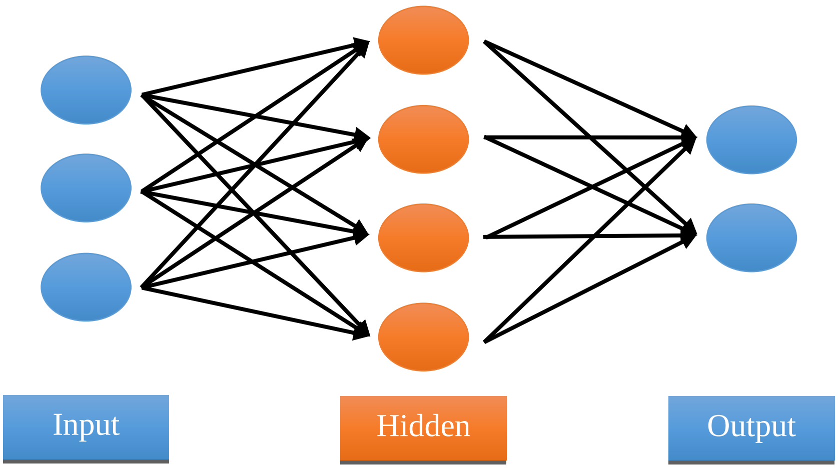


Figure 2.3: The architecture of neural networks

2.4.2Basic Principles of Neural Networks

Neural networks are computational models that are inspired by the structure and function of biological neural networks, such as the human brain. At their core, neural networks consist of interconnected nodes or neurons that process and transmit information. As shown in Figure 2.4, the basic principles of neural networks include:

- **Activation function**: The activation function of a neuron determines the output of the neuron based on the weighted sum of its inputs. Common activation functions include the sigmoid and rectified linear unit (ReLU) functions

- **Weight and bias**: The connections between neurons in a neural network are represented by weights, which determine the strength of the connection. A bias term is also added to each neuron to allow for shifting the activation function.
- **Feedforward**: In a feedforward neural network, information flows in one direction, from the input layer through one or more hidden layers to the output layer.
- **Backpropagation**: a learning algorithm used to train neural networks. It involves calculating the error between the predicted output and the actual output and then adjusting the weights and biases of the network to minimize the error [20]
- **Overfitting and regularization**: Overfitting occurs when a neural network is too complex and fits the training data too closely, resulting in poor performance on new data. Regularization techniques, such as L1 and L2 regularization, prevent overfitting.
- **Hyperparameters**: Neural networks have many hyperparameters that need to be set, such as the number of layers, the number of neurons in each layer, the learning rate, and the regularization strength [21]**.**

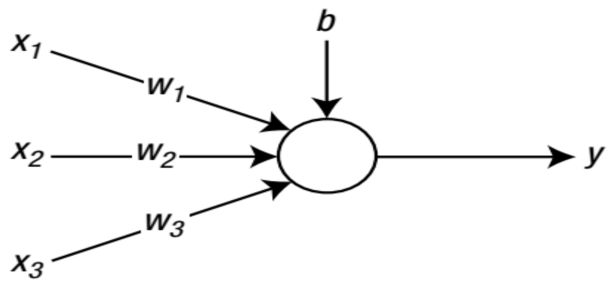


Figure 2.4: Typical Artificial Neural Network

Where:

$\boldsymbol{x_i}$: the inputs of the network

w_i denote the weights to be learned by the network,

$\boldsymbol{b}$: the bias of the network

y the out put

Figure 2.5 shows the details of the different functions of an artificial neurone.

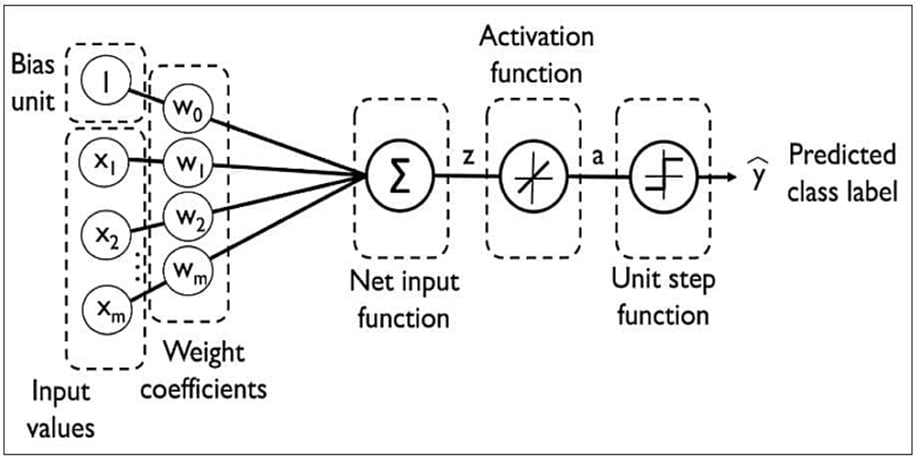


Figure 2.5: Block diagram of artificial neuron.

2.4.3Forward propagation

It is the process by which information flows through a neural network, from the input layer through one or more hidden layers to the output layer. During forward propagation, each neuron in the network receives input from the neurons in the previous layer, processes that input using its activation function, and then passes the output to the neurons in the next layer. The forward propagation process can be summarized as follows [22] as illustrated in Figure 2.6:

- **Input layer**: The input layer of the neural network receives the input data, which is usually represented as a vector [22]**.**
- **Hidden layers**: The input data is then passed through one or more hidden layers, each consisting of multiple neurons. Each neuron in the hidden layer receives input from the neurons in the previous layer, processes that input using its activation function, and then passes the output to the neurons in the next layer.
- **Output layer**: The final layer of the neural network is the output layer, which consists of one or more neurons. Each neuron in the output layer receives input from the neurons in the previous layer, processes that input using its activation function, and produces an output [23].
- **Output**: The output of the neural network is the final output produced by the output layer neurons.

- **Epoch**: The number of epochs is a hyperparameter that specifies how many times the learning algorithm will process the complete training dataset. During an epoch, the internal model parameters are updated for each sample in the training dataset, with an epoch typically consisting of one or multiple batches [24].
- **Batch**: The batch size is a hyperparameter that determines the number of samples to process before updating the internal model parameters. If a single batch contains all training samples, the learning algorithm is known as batch gradient descent. Stochastic gradient descent is the term used when a batch consists of one sample. Mini-batch gradient descent refers to the learning algorithm where the batch size ranges between one sample and the size of the training dataset

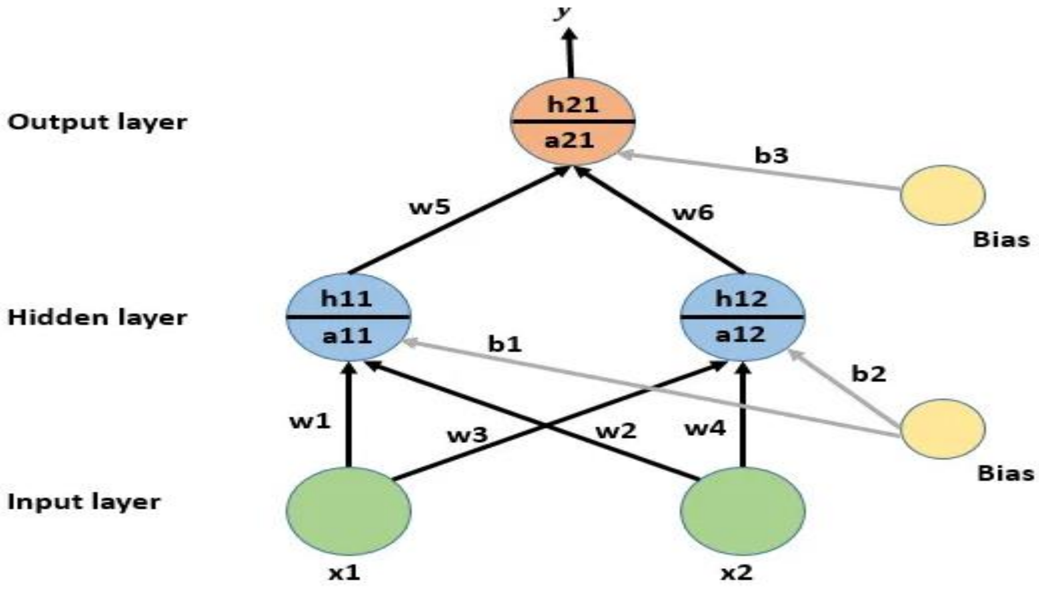


Figure 2.6: Demonstration of Layer Neural Network

2.4.4Loss function

A loss function is a mathematical function that measures the difference between the predicted output and the actual output in a neural network. It quantifies the error between the predicted output and the actual output for a given input and is used to optimize the weights and biases of the neural network during the training process. The choice of loss function depends on the nature of the problem being solved and the type of output produced by the neural network [14]. The loss function $\mathcal{L}$ defined based on norme 1 (N1) as follows:

$$\mathcal{L}(P) = \frac{1}{N}\sum_{p \in P}|x(p) - y(p)| \qquad 2.1$$

Where x denotes the predicted output.

y Denotes the actual output

Loss function can be defined based on norm2 (N2) as follows:

$$\mathcal{L}(P) = \frac{1}{N}\sum_{p \in P}(x(p) - y(p))^2 \qquad 2.2$$

2.4.5Training the neural network

Training a neural network involves adjusting the weights and biases of the network so that it can accurately map inputs to outputs, as shown in Figure 2.7. The process typically involves feeding the network a set of inputs (generally big data sets) and comparing its predicted output to the true output. The difference between the predicted output and the true output is used to adjust the network's weights and biases so that it can make better predictions in the future [25] and then converges.

The goal of training a neural network is to minimize the difference between its predicted output and the true output for a given input. This is achieved by using an optimization algorithm, such as stochastic gradient descent, to adjust the network's weights and biases. During training, the network is typically evaluated on a validation set to ensure that it is not overfitting the training data. Once the network has been trained, it can be used to make predictions on new data [25].

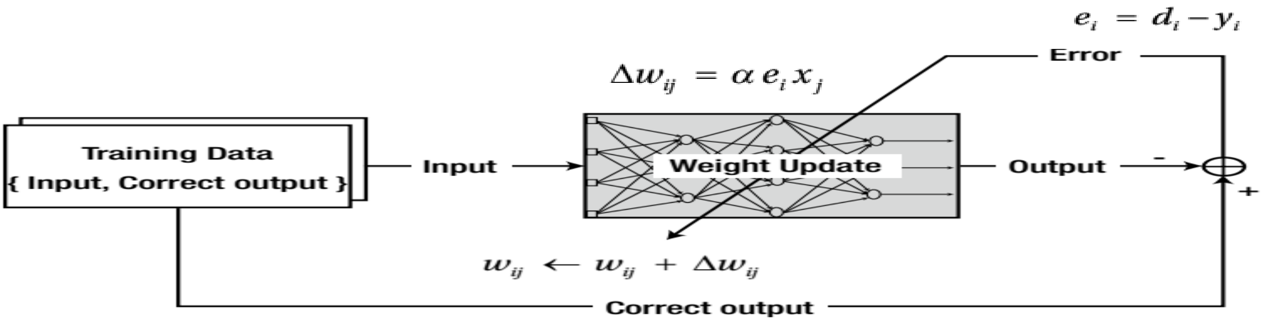


Figure 2.7:Training procedure of neural networks [25]

Where:

x_j: the jth input vector,

e_i: the error between the ith predicted and the true output

w_{ij}: the weights to be determined

Δw_{ij} the term of correction to update the weights

α: the learning rate (generally choosen very small)

- **Gradient descent:** is a popular optimization algorithm used in machine learning to find the optimal weights or parameters of a model by minimizing a loss function. It works by iteratively adjusting the weights in the direction of the steepest descent of the loss function until it reaches a minimum.

- **Regression**: is a type of supervised learning where the goal is to predict a continuous numerical output variable based on a set of input variables or features. Examples of regression problems include predicting housing prices, stock prices, or the amount of rainfall.

2.5Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of neural network that is designed to process data with a grid-like structure, such as images or time-series data. The key feature of a CNN, as illustrated in Figure 2.8 is the convolutional layer, which applies a set of learnable filters to the input data to extract features. The filters are learned during the training process and are optimized to detect specific patterns in the data [26]**.**

CNNs are typically composed of several layers, including convolutional layers, pooling layers, and fully connected layers, as shown in Figure 2.8. The convolutional layers are responsible for extracting features from the input data (giving feature maps as the number of filters), while the pooling layers downsample the output of the convolutional layers to reduce the dimensionality of the feature maps. The fully connected layers are used to classify the input data on the considered output, based on the features extracted by the convolutional layers.

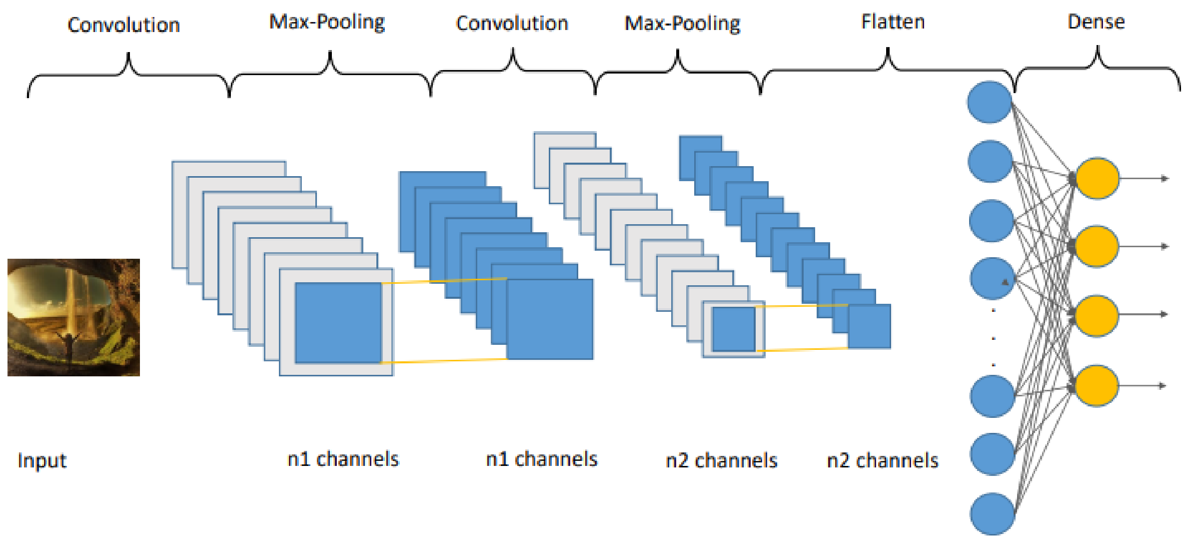


Figure 2.8 :Conventional architecture of a convolutional neural network

The hidden layers after filtering and Maxpooling processes are represented by n1 channels and n2 channels, respectively. The Dense layer consists of a simple network, followed by four outputs.

The architecture of a CNN is highly flexible, and there are many variations on the basic architecture that have been developed over the years. The choice of architecture depends on the specific task being solved, the size and complexity of the input images, and the available computational resources [27]**.**

2.5.1CNN Components

The key components of a CNN include the input layer, convolutional layers, activation layers, pooling layers, fully connected layers, and output layers.

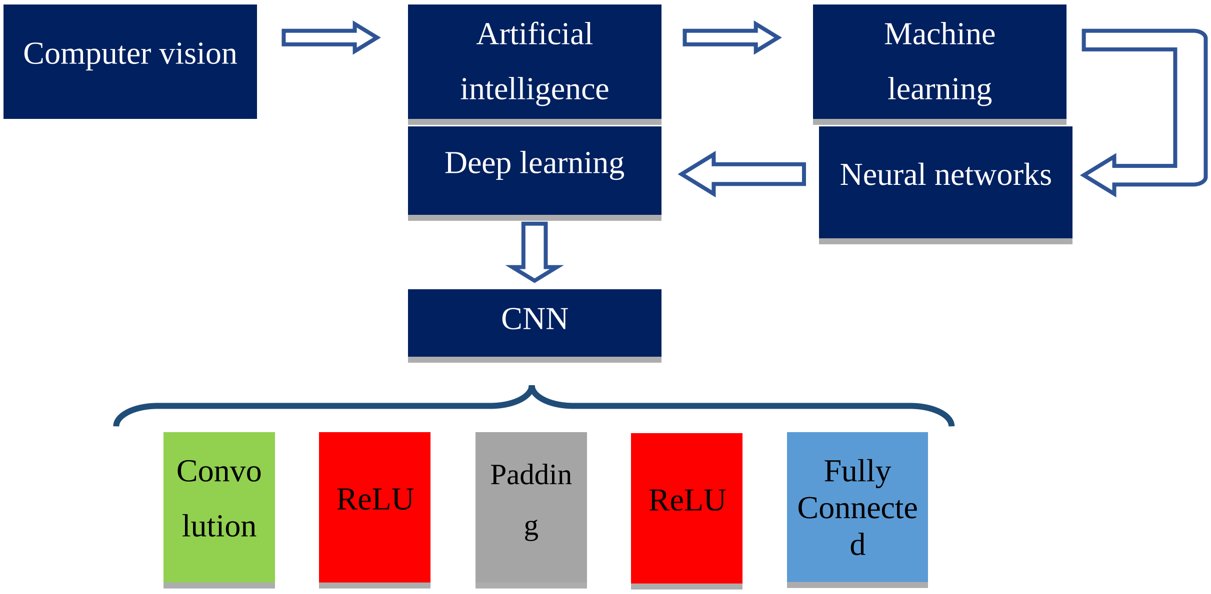


Figure 2.9: CNN architecture

- **Fully connected layer**: This layer takes the output of the pooling layer and assigns it to a specific class. A fully connected layer is similar to a layer in a traditional neural network, where each neuron in the layer is connected to every neuron in the previous layer.

- **The convolutional layer**: Convolutional Neural Networks (CNNs) are very effective at recognizing the spatial and temporal relationships between pixels in images. They do this by using a convolutional layer that applies a series of filters to the input image. These filters create feature maps by convolving a small weight matrix with the input image, thereby extracting important features such as edges, lines, textures, and shapes. Each feature map represents a specific feature or pattern that the CNN has learned to recognize in the input image. Multiple filters generate multiple feature maps, each representing a unique set of learned features. The output of the convolutional layer undergoes a ReLU-like activation to introduce nonlinearity into the training process and then enters the pooling layer. The pooling layer downsamples the feature map to reduce its size while preserving the most important information [27].
- **Pooling layers**: CNNs typically include pooling layers that downsample the feature maps generated by convolutional layers. This helps reduce the dimensionality of

feature maps, making them more computationally efficient to process and reducing the risk of overfitting. Max pooling works by dividing the feature map into non-overlapping regions and taking the maximum value of each region. This operation preserves the most salient features of the feature map and discards the rest, resulting in a compressed representation of the input image. The pooling operation also helps enforce a degree of translation invariance in the features, meaning the network can recognize an object regardless of its position in the input image. Convolutional layers are used for feature detection and max pooling layers are used for feature selection [27].

- **Activation layers**: These layers introduce non-linearity into the model by applying an activation function to the output of the convolutional layer. The most commonly used activation function is ReLU (Rectified Linear Unit).

2.6Conclusion

In this chapter, we have introduced the fundamental concepts of machine learning and deep learning. We have explored the basics of neural networks, including their structure and the role of activation functions. Additionally, we have discussed the application of convolutional neural networks in image processing tasks. By understanding these concepts, we gained insight into the foundations of deep learning and its potential for solving complex problems in various fields.

Chapter3

Implementation of Image Super-Resolution Algorithms based on Conventional and Deep Learning SR Algorithms: a quantitative comparative study

Abstract

In this chapter, we first implement the conventional image super resolution algorithms; namely: Bicubic, Nearest Neighbor, and Bilinear interpolation. Then we implement the deep learning SR algorithms: VDSR, EDSR, and CAR. The comparative study is in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) in addition to the visual quality. We have performed many simulation experiments with the most used datasets in this field.

Summary

3.1 Introduction:

In this chapter, we focus on presenting all our simulation results obtained in this study. Our experiments involved implementing various super-resolution (SR) algorithms, including conventional SR algorithms: bicubic, bilinear, and nearest neighbor. Then, we implement SR algorithms based on deep learning process: Very Deep Learning SR (VDSR), EDSR, and CAR. The quantitative comparative study is in terms of two well known metrics: PSNR and SSIM, in addition to evaluating the visual quality of the recovered images. Note that these two metrics are computed for each obtained test image. The mean values of PSNR and SSIM are computed at the end of all the simulation experiments. For the sake of objective comparison, we use the same datasets for all the considered SR algorithms

3.2 Used material:

The code sources were developed and executed in Python 3.9.13 using Spyder Navigator in Anaconda environment on a TOSHIBA PC with an i7-5600 vPro processor running Windows 10 Professional. PyTorch was the main programming package used in Python, specifically designed for deep learning implementations, in addition to the packages of image processing such as scikit-image. The primary code sources were run on Google Colab and MATLAB R2022a. The big datasets are processed thanks to the use of GPU online in Google Colab.

3.3 Used data sets:

The developed SR algorithms are tested on the datasets used in the recent literature; namely: **LIVE1**, **Set5** and **Set14** as summarized in Table 3.1

Table 3.1 Used Datasets

Data sets	Quantity (nbr of images)	Format	Resolution
Set5	**5**	**PNG**	**313.548**
Set14	**14**	**PNG**	**497.409**
LIVE1	**29**	**BMP**	**688.548**

For the sake of clarity, we give in Tables 3.2-3.4 the content of each base.

Table 3.2 Dataset Set5

Image	Resolution
Butterfly	**(256x256)**
Head	**(280x280)**
Woman	**(228x344)**
Baby	**(512x512)**
Bird	**(288x288)**

Table 3.3 Datset Set14

Image	Resolution
Flowers	**(500x362)**
Coastguard	**(352x288)**
Bridge	**(512x512)**
Zebra	**(586x391)**
Foreman	**(352x288)**
Pepper	**(512x512)**
Lenna	**(512x512)**
Face	**(276x276)**
Ppt3	**(529x656)**
Barbara	**(720x576)**
Monarch	**(768x512)**
Comic	**(250x361)**
Baboon	**(500x480)**
Man	**(512x512)**

Table 3.4: LIVE1

IMAGE	Resolution
Caps	**(768x512)**
Manfishing	**(634x438)**
Sailing	**(768x512)**

Carnivaldolls	**(610x488)**
Coinsinfountain	**(640x512)**
Rapids	**(768x512)**
Ocean	**(768x512)**
Churchandcapitol	**(634x505)**
Woman	**(480x720)**
Womanhat	**(480x720)**
Buildings	**(768x512)**
Statue	**(480x720)**
Sailing3	**(480x720)**
Building2	**(640x512)**
Plan	**(768x512)**
Bikes	**(768x512)**
Sailing4	**(768x512)**
Stream	**(768x512)**
Dancers	**(618x453)**
Lighthouse3	**(768x512)**
Cemetry	**(627x482)**
Parrots	**(768x512)**
House	**(768x512)**
Flowersonih35	**(640x512)**
Studentsculpture	**(632x505)**
Monarch	**(768x512)**
Lighthous2	**(768x512)**
Sailing2	**(480x720)**
Paintedhouse	**(768x512)**

3.4 Original Images:

In our study, the original images are considered as High resolution (HR). The corresponding original images are shown in Figure 3.1. Many simulation experiments have been conducted. however due to limitation space, we show few of the obtained results.

Set5

Set14

LIVE1

Figure 3.1 : Original HR images of the used datasets

3.5 Implementation of SR Algorithms

3.5.1 Conventional SR Algorithms

The flowchart shown in Figure 3.2 summarizes the main steps of our implementation. First, we generate the low-resolution images (LR) from the HR ones. Note that we have only considered the downsampling as image degradation in our study. Both noise and blur are not considered. The two values of the scale factor s_f are X2, X4.

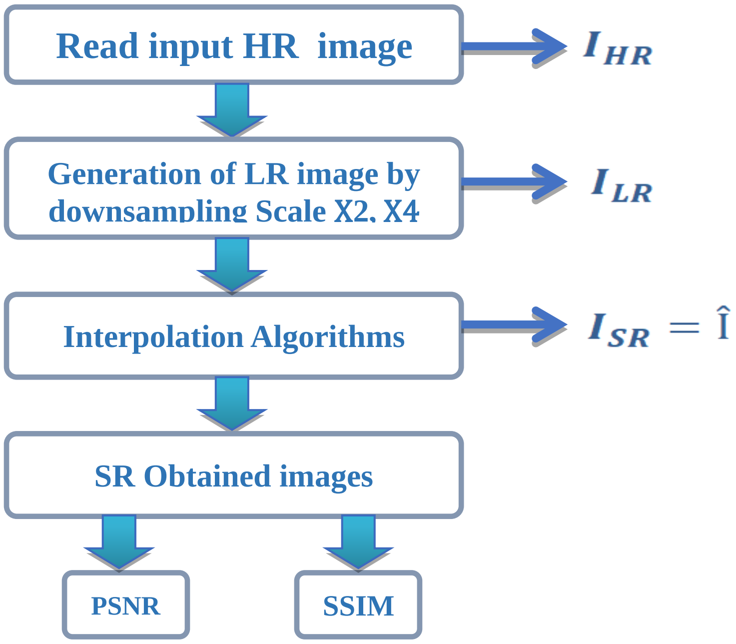


Figure 3.2 : Flowchart of interpolation algorithms

3.5.1.1 Bicubic Interpolation:

Tables 3.5-3.7 show the obtained PSNR and SSIM. Test images are from the different data sets.

Table 3.5 PSNR & SSIM of Bicubic Interpolation

Image	Datasets	Scale	PSNR(dB)	SSIM
Butterfly	**Set5**	**2**	**32.1329**	**0.9178**
		4	**30.1263**	**0.7294**
Baby		**2**	**37,0088**	**0,9556**
		4	**33.1127**	**0.9406**
Barbara	**Set14**	**2**	**32.5425**	**0.8546**
		4	**31.0359**	**0.7039**
		2	**33.0060**	**0.9219**

Zebra		4	30.2537	0.7053
Caps	LIVE1	2	36.4164	0.9316
		4	34.3064	0.8421
Bikes		2	31.5294	0.8635
		4	30.0087	0.6158

one corresponding obtained image is shown in Figure 3.3

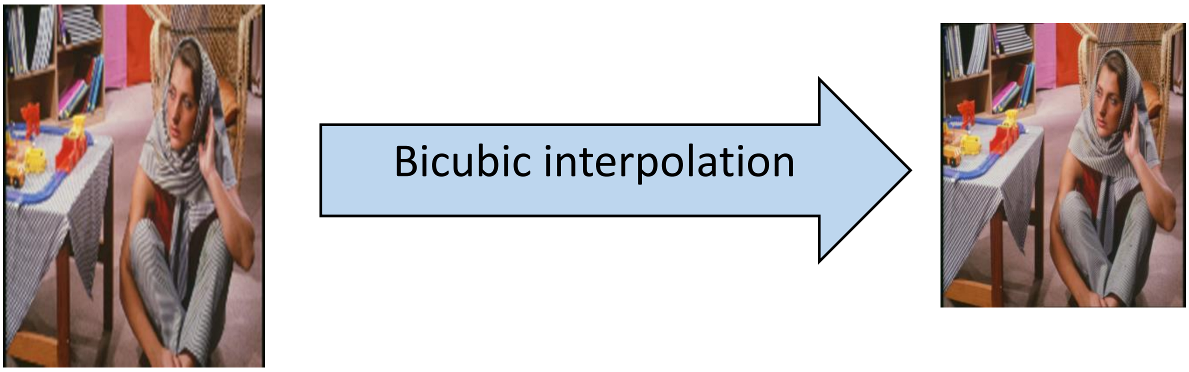


Figure 3.3: Bicubic interpolation

Table 3.6: PSNR & SSIM Nearest Neighbor

Image	Datasets	Scale	PSNR(dB)	SSIM
Butterfly	Set5	2	32.7210	0.8768
		4	31.5540	0.6587
Baby		2	35.1127	0.9306
		4	32.7996	0.6491
Zebra	Set14	2	31.4953	0.8185
		4	30.0531	0.5945
Barbara		2	31.9785	0.8422
		4	30.8719	0.6491
		2	36.2172	0.9227

Caps	LIVE1	4	34.4064	0.8219
Bikes		2	31.5210	0.8305
		4	30.2379	0.5712

One corresponding SR image (baby) is shown on Figure 3.4;

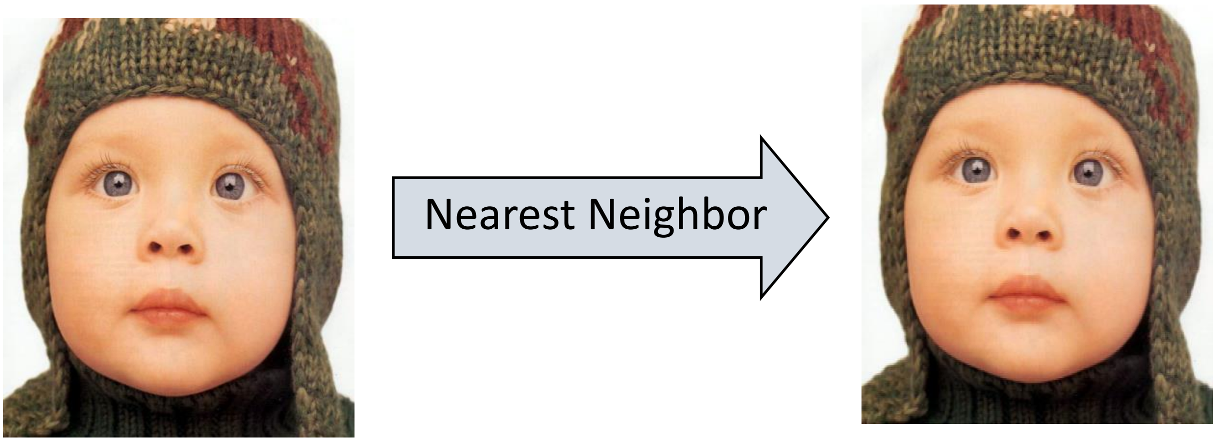


Figure 3.4: Nearest Neighbor

Table 3.7: Bilinear Interpolation

Image	Datasets	Scale	PSNR(dB)	SSIM
Butterfly	Set5	2	32.0372	0.8849
		4	30.4777	0.7212
Baby		2	35.7198	0.9378
		4	33.2417	0.8527
Barbara	Set14	2	31.9668	0.8221
		4	30.9543	0.6941
Zebra		2	31.9825	0.8845
		4	30.1066	0.6815
		2	35.9810	0.9146

Caps	**LIVE1**	**4**	**34.3632**	**0.8432**
Bikes		**2**	**31.1710**	**0.8106**
		4	**30.0225**	**0.6002**

One corresponding SR image (Zebra) is shown on Figure 3.5.

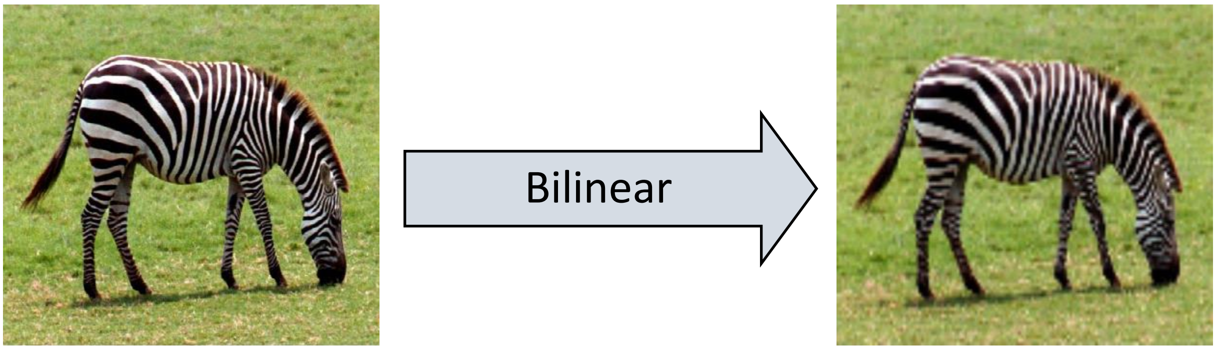


Figure 3.5: Bilinear Interpolation

From the obtained results, we easily observe that bicubic highlights the nearest and bilinear in terms of PSNR and SSIM. However, the resolution in bicubic is reduced. These remarques are reproduced for all the images.

3.5.2 Implementation of SR Deep Neural Networks:

Note that VDSR is the first SR deep learning network. For both VDSR and EDSR, we first generate the LR image by downsampling. The flowchart shown in Figure 3.6 summarizes the main steps of the implementation.

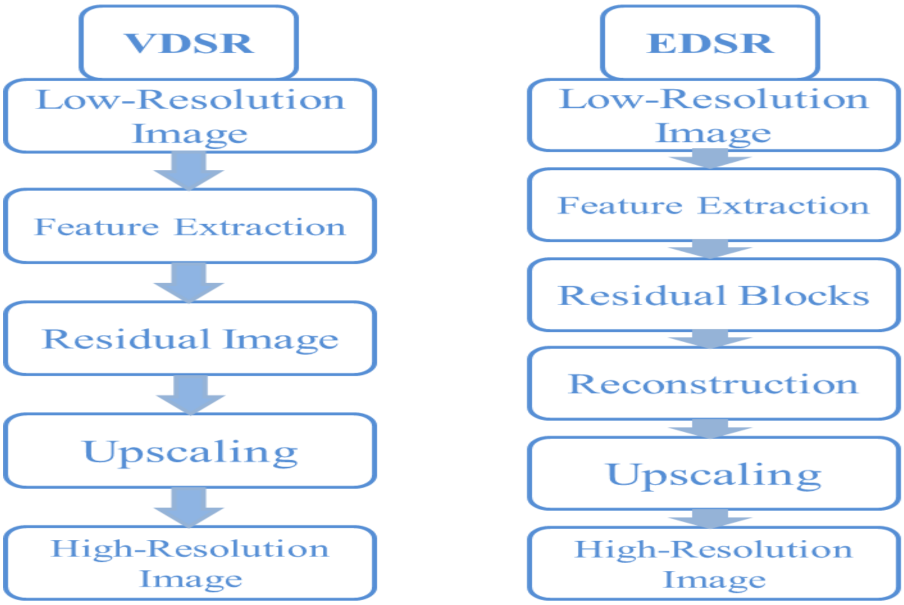


Figure 3.6: Flowchart of VDSR and EDSR algorithms

Tables 3.8-3.9 show the obtained PSNR & SSIM of VDSR and EDSR respectively.

VDSR

Table 3.8: PSNR & SSIM of VDSR

image	Datasets	Scale	PSNR(dB)	SSIM
Butterfly	**Set5**	**2**	**34.1875**	**0.7808**
		4	**26.5469**	**0.7835**
Baby		**2**	**38.6875**	**0.7080**
		4	**33.3750**	**0.7254**
Zebra	**Set14**	**2**	**34.2188**	**0.6012**
		4	**26.7344**	**0.6137**
Barbara		**2**	**28.1875**	**0.8987**
		4	**25.7500**	**0.7968**
		2	**30.8438**	**0.9495**

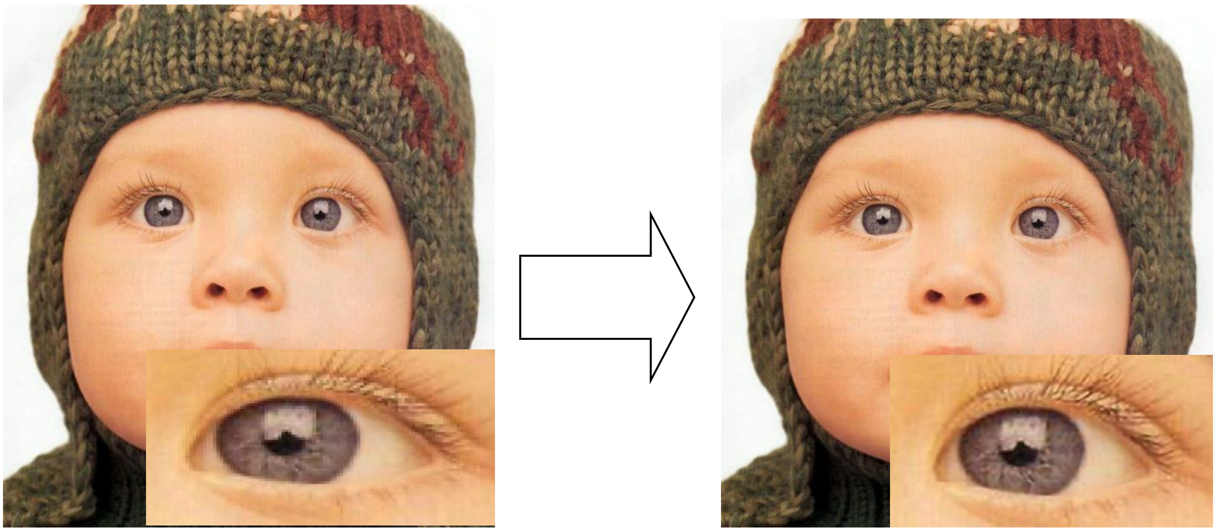
Caps	**LIVE1**	**4**	**33.1562**	**0.7773**
Bikes		**2**	**38.1250**	**0.3471**
		4	**24.5625**	**0.1522**

EDSR

Table 3.9: PSNR & SSIM of EDSR

image	Datasets	Scale	PSNR(dB)	SSIM
Butterfly	**Set5**	**2**	**31.3049**	**0.9826**
		4	**30.2462**	**0..9826**
Baby		**2**	**42.1512**	**0.9909**
		4	**42.1021**	**0.9828**
Zebra	**Set14**	**2**	**36.7189**	**0.9885**
		4	**36.3591**	**0.9790**
Barbara		**2**	**32.132**	**0.9681**
		4	**31.5598**	**0.9583**
Caps	**LIVE1**	**2**	**37.0144**	**0.9759**
		4	**36.5893**	**0.9736**
Bikes		**2**	**31.98**	**0.9643**
		4	**31.7855**	**0.9494**

The corresponding obtained SR images ‘baby of VDSR and EDSR are shown on Figure 3.7-3.8 respectively. Furthmore, we show the region of interest (ROI) to more view the difference between HR and SR images. We easely seen, that the visual quality of HR and SR images are approximatevely the same, eather in ROI. However, EDSR results is slightly better in terms of visual quality.



HR **SR**

Figure 3.7: EDSR

LR **HR**

Figure 3.8: VDSR

CAR(Content-Adaptive Resampled):

As outlined in Chapter 1, the content-adaptive resampled (CAR) model comprises two main components: the ResamplerNet and SRNet blocks. This framework consists of the following primary elements.

The ResamplerNet is responsible for predicting content-adaptive resampling kernels based on the input high-resolution (HR) image. These kernels are then applied to the HR image to generate a downscaled image.

On the other hand, the SRNet accepts the downscaled image as input and aims to reconstruct the original HR image. Recall that in CAR, there isn't need to generate the LR images frome the HR ones, because this step is including in the CAR network, in fact this increases the accuracy of CAR. We summarize our implementation of CAR in the the flowchart shown in Figure 3.9.

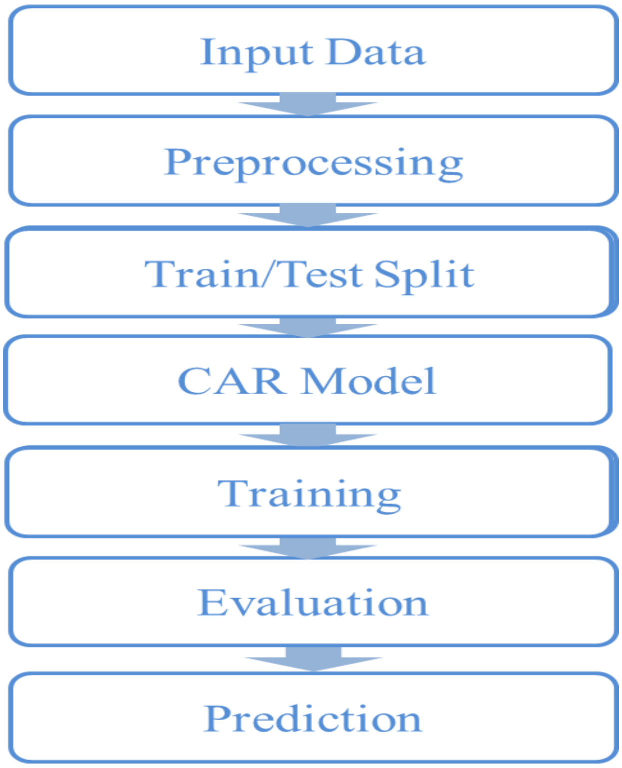


Figure 3.9: Flowchart of CAR

For the sake of objective comparison , we use the same datasets as before.

Table 3.10 displays the computed average values of the two metrics PSNR and SSIM, after computing PSNR and SSIM of each image as summarized in Table 3.11. We fixe the scale factor to two values: x2 and x4 It showcases the original images, the generated downsampling images, and the resulting SR images. The corresponding obtained SR images are shown on Figures 3.10-3.11 (where we show: the original image, generated LR version with two values of the scale factor and SR reconstructed image respectively. To more highlight the difference between the tthree versions, we zoom region of interest (ROI) which contains more details. It is evident that the quality of the SR images has significantly improved and is close to that of the original images. We have included one image of our faculty (Figure 3.10 (b)). We can easely seen that CAR allows us to significantly enhance the quality of the reconstructed image and increase the values of PSNR and SSIM. The obtained results are satisfactory and promising.

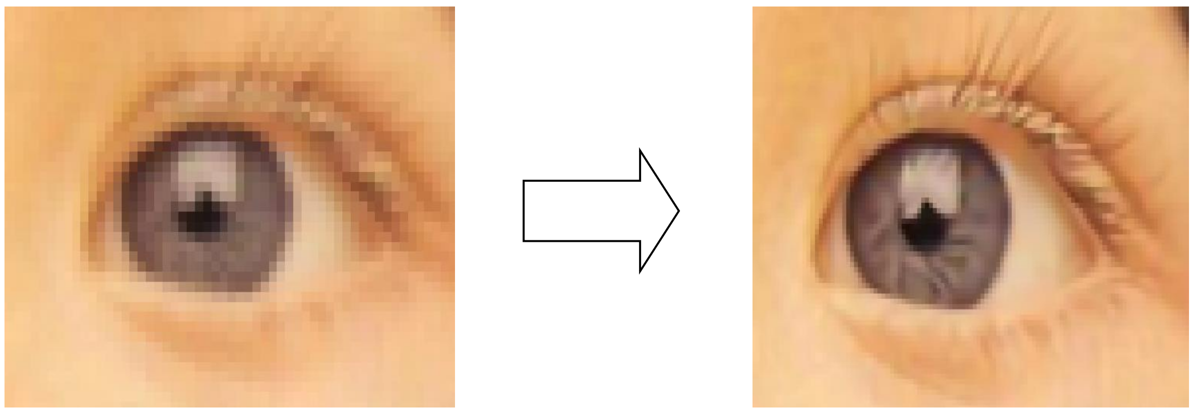
Table 3.10: meanPSNR & meanSSIM values of CAR

Datasets	Scale	MEAN PSNR(dB)	MEAN SSIM
Set5	2	38.96	0.9643
	4	34.17	0.9196
Set14	2	35.84	0.9394
	4	30.61	0.8427
LIVE1	2	34.90	0.9446
	4	29.71	0.8393

Table 3.11: PSNR & SSIM by CAR, for each image

image	Datasets	Scale	PSNR(dB)	SSIM
Butterfly	Set5	2	36.8531	0.9808
		4	30.9527	0.9434
Baby		2	39.6441	0.9725
		4	35.2766	0.9230
Zebra	Set14	2	35.9430	0.9558
		4	30.6106	0.8597
Barbara		2	36.6782	0.9648
		4	27.9461	0.8404
Caps		2	35.4539	0.9194
		4	40.9641	0.9761

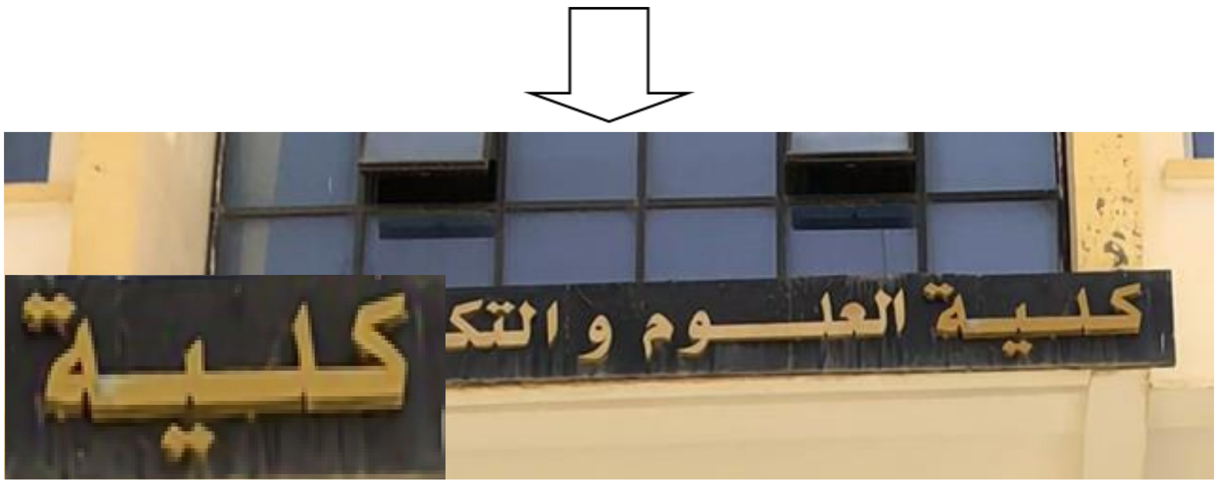
Bikes	LIVE1	2	27.7551	0.8254
		4	34.5244	0.9592



ROI LR image SR image

(a)

LR



(b) SR

Figure 3.10: First results of CAR

We show the results where the scale factor is fixed to 2, followed by the results of the scale factor equals to 4. The value of the corresponding PSNR is given on right

Figure 3.11: CAR results Set 5, scale factor=2

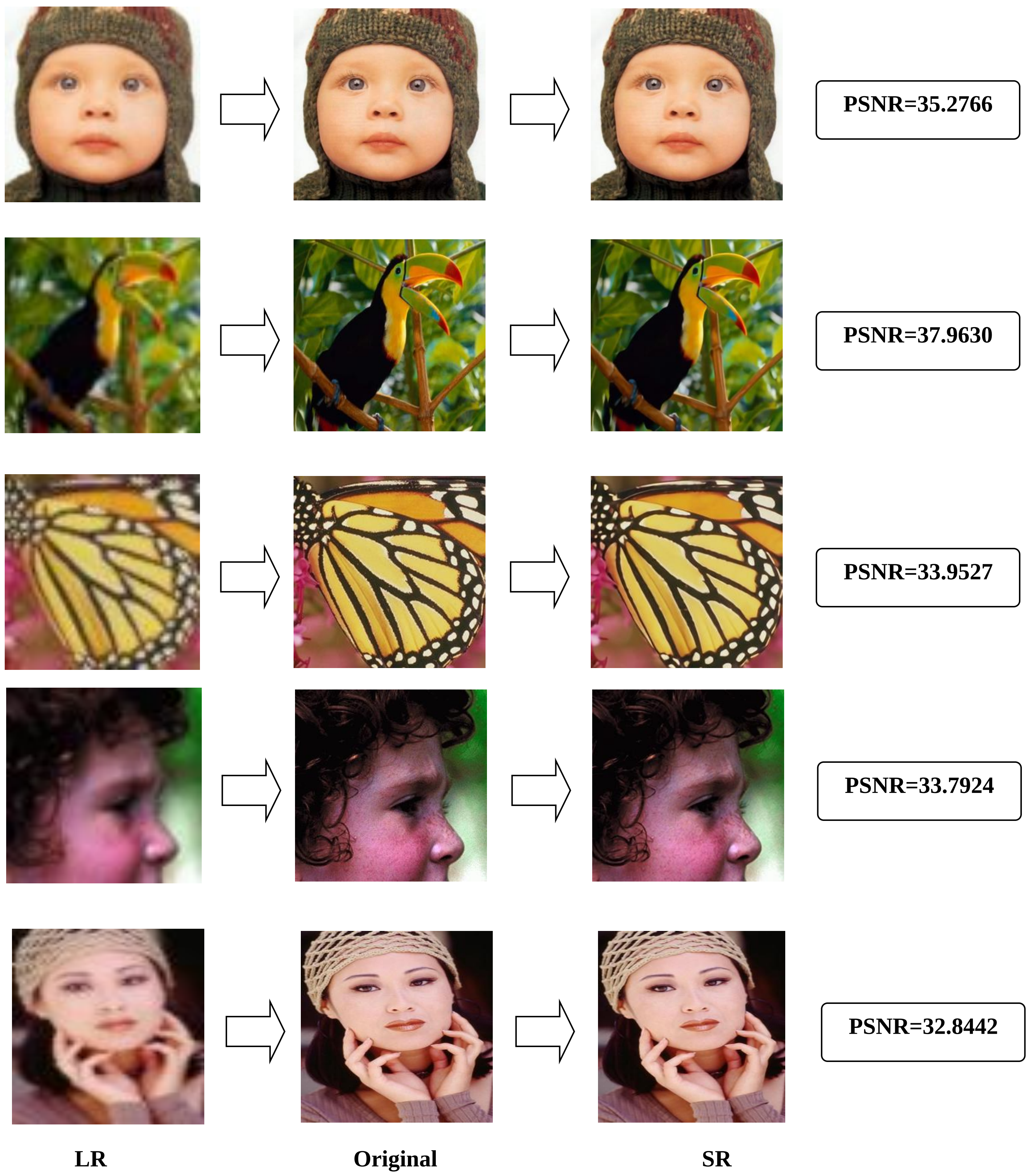


Figure 3.12: CAR results Set 5, scale factor x4

The results of Set14, scale factor fixed at 2 are shown on Figure 3.13.

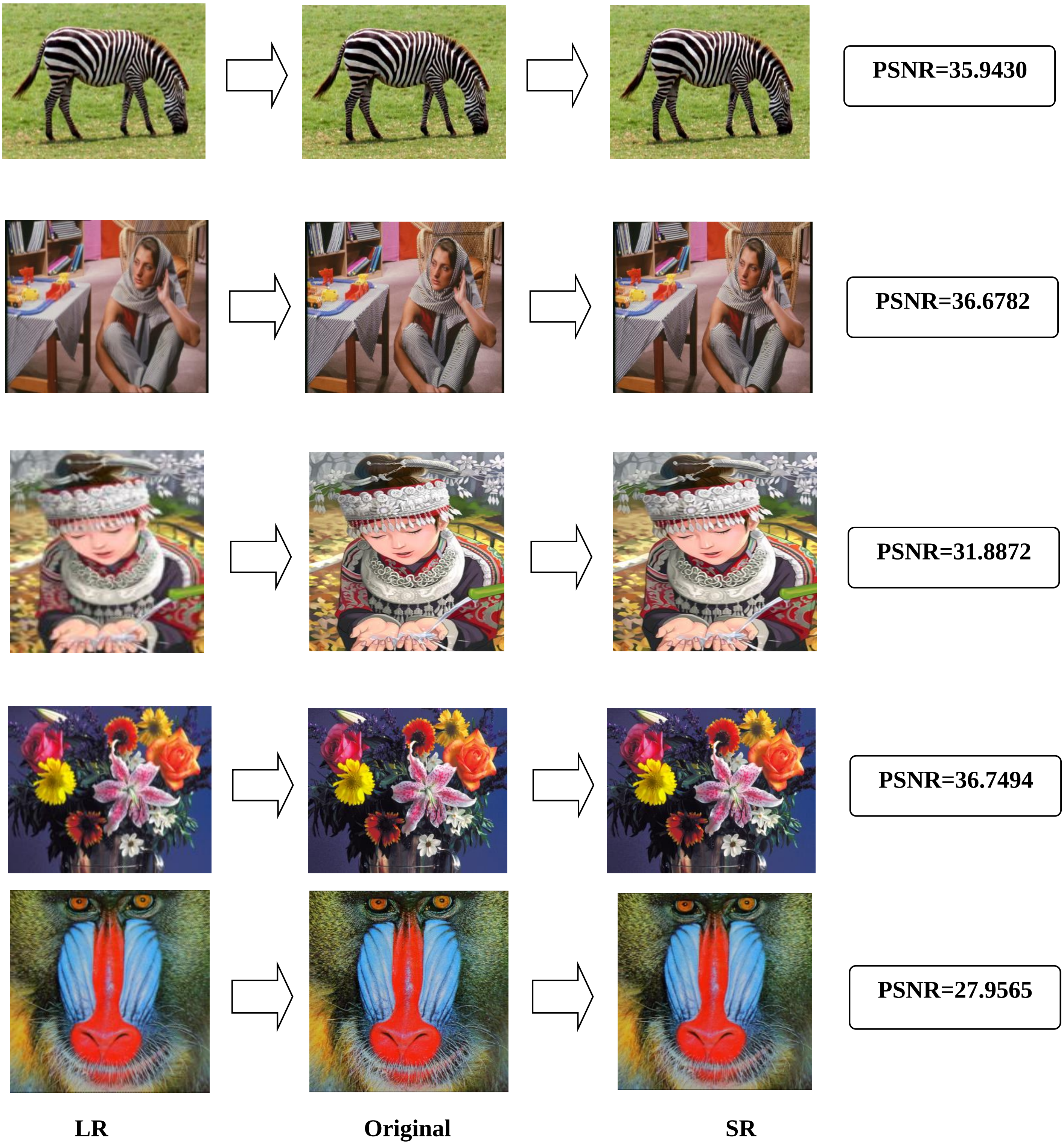


Figure 3.13: CAR results Set 14, scale factor x2

For $s_f = 4$, we show the obtained results on Figure 3.15.

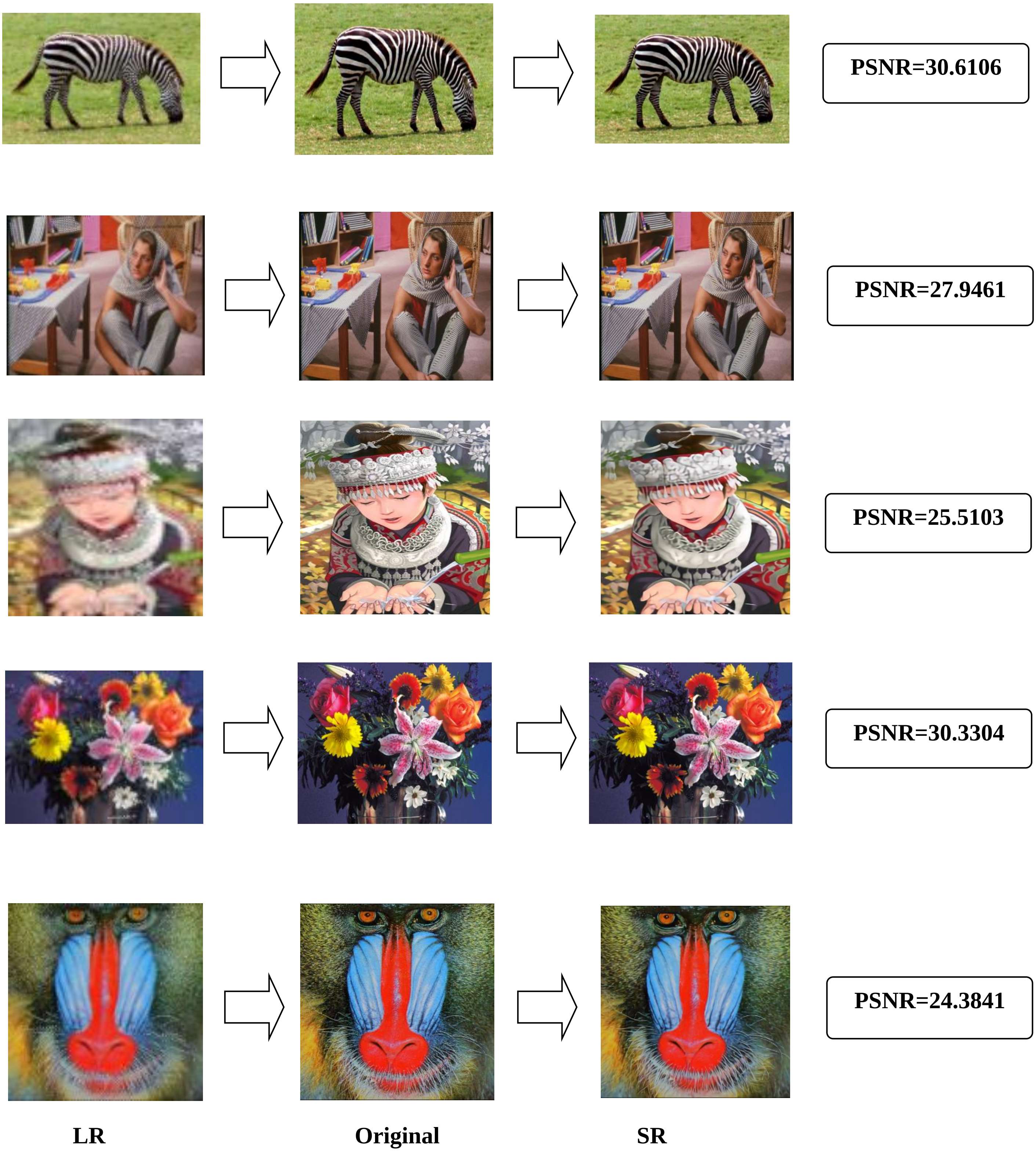
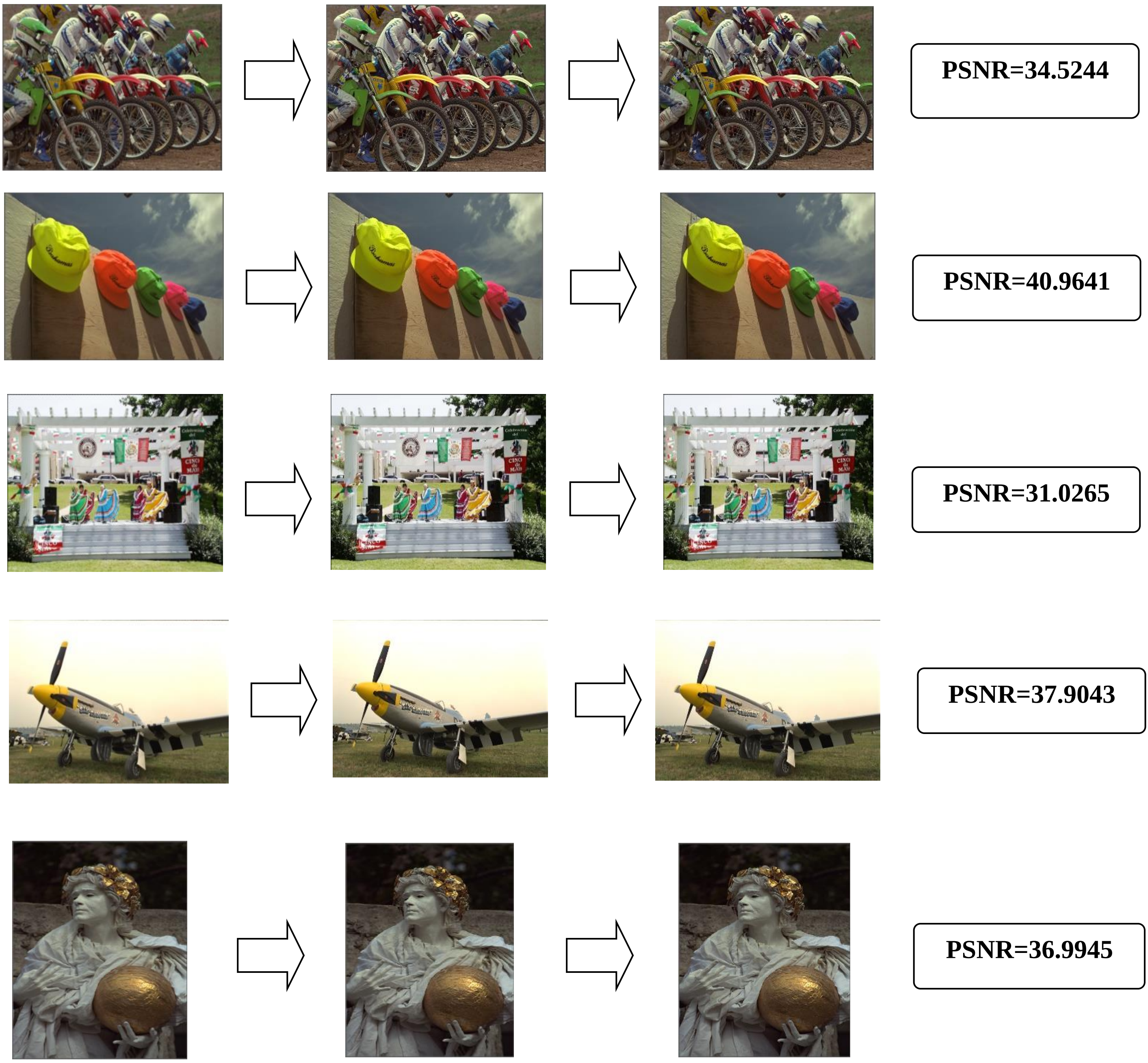


Figure 3.14: CAR results Set 14, scale factor x4

We can observe, from the obtained values of PSNR, that PSNR is inversely proportioned to the scale factor. PSNR dcreases, when s increases., i.e when the deterioration is significant. Figure 3.15 and 3.16 show the obtained CAR results for Live 1 with s= x2,   x4, respectively. The corresponding value of PSNR is indicated at the right of the figures.



LR **Original** **SR**

Figure 3.15: CAR results for LIVE 1, s=x2

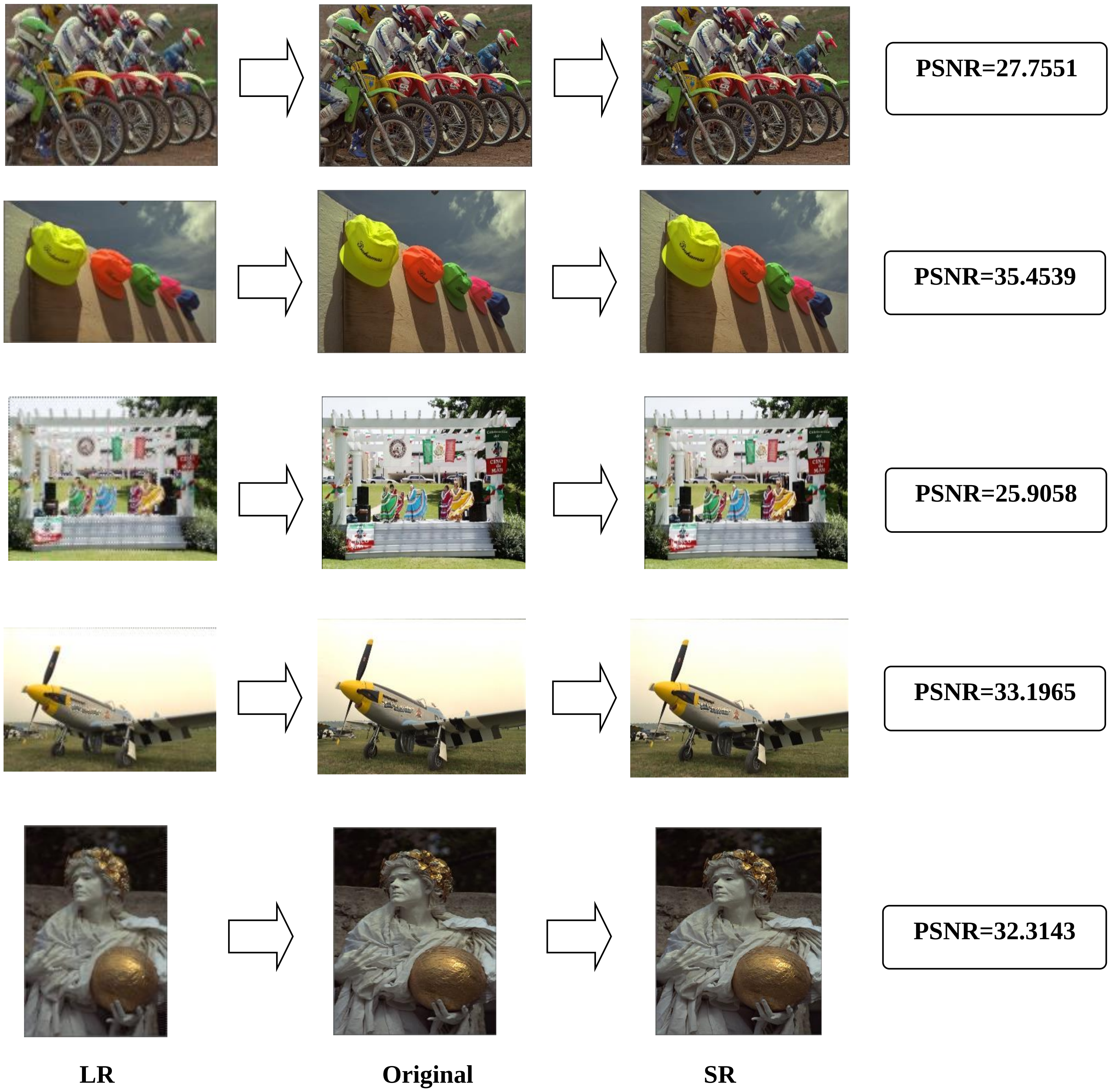


Figure 3.16: CAR results, LIVE 1, scale=x4

both two values of the scale factor. This is justified by the satisfactory values of PSNR. From many simulation experiments, CAR is better than EDSR and VDSR. This enhancement is done thanks to the generation of LR images by the network it self.

3.6 Conclusion

In our study, we examined three deep learning-based super-resolution (SR) image algorithms: VDSR, EDSR and CAR which is an enhanced version of EDSR. We have also applied conventional interpolation algorithms: Bicubic, bilinear and nearest neigbord. We have used the well known datatsets in the filed of SR problem: Set5, Set14, and LIVE1. Our assessment focused on two metrics, PSNR and SSIM, as well as the visual quality of the SR reconstructed images.

To conduct each experiment, we initially generated low-resolution (LR) images from the original ones. This step resulted in either X2 or X4 scaled images. Based on the results obtained, it is evident that the CAR algorithm outperforms the other SR algorithms in terms of visual quality and computed metrics. However, it is worth noting that deep learning SR algorithms allow the process of big data and large amount of images in short time, thanks to the use of GPU and the network architecture. Finaly, our goals have been succefully achieved.

General conclusion

The thesis titled "A Comparative Study of Conventional and Deep Learning Interpolation Image Super Resolution Algorithms" focuses on comparing two approaches for enhancing the resolution of images: conventional interpolation algorithms and deep learning-based interpolation algorithms. After conducting a comprehensive study, several key conclusions can be drawn from this thesis.

Firstly, the thesis highlights that conventional interpolation algorithms, such as bicubic interpolation, have been widely used for image super resolution tasks. These algorithms are based on mathematical principles and are relatively straightforward to implement. However, the results obtained from conventional interpolation algorithms may lack fine details and fail to produce high-quality super-resolved images.

Secondly, the thesis emphasizes the emergence of deep learning-based interpolation algorithms, which utilize neural networks to learn complex mappings between low-resolution and high-resolution image pairs. These algorithms have shown remarkable performance in generating visually appealing and highly detailed super-resolved images. They can capture intricate patterns and textures that conventional algorithms struggle to reproduce.

Furthermore, the thesis concludes that deep learning-based interpolation algorithms generally outperform conventional interpolation algorithms in terms of perceptual quality and objective metrics. The ability of deep learning models to learn from large-scale datasets enables them to capture intricate image features, leading to superior super-resolution results.

However, the thesis also acknowledges that deep learning-based approaches may have some limitations. They often require significant computational resources and extensive training data to achieve optimal performance. Additionally, the selection of appropriate network architectures, loss functions, and training strategies significantly affects the final results.

In conclusion, the thesis provides valuable insights into the comparative study of conventional and deep learning interpolation image super-resolution algorithms. It highlights the advantages of deep learning approaches in generating high-quality super-resolved images, while acknowledging the challenges and considerations involved in their implementation.

References

[1] K. Nakanishi, S. Maeda, T. Miyato, and D. Okanohara, "Neural Multi-scale Image Compression," 2018 : https://doi.org/10.48550/arXiv.1805.06386.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015. [En ligne]. Disponible : https://doi.org/10.1038/nature14539

[3] Z. Wang, J. Chen and S. C. H. Hoi, "Deep Learning for Image Super-Resolution: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3365-3387, 1 Oct. 2021, doi: 10.1109/TPAMI.2020.2982166.

[4] J. Kim, J. K. Lee, and K. M. Lee, "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016," pp. 1646-1654

[5] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281

[6] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017, pp. 136-144

[7] W. Sun and Z. Chen, "Learned Image Downscaling for Upscaling Using Content Adaptive Resampler," in IEEE Transactions on Image Processing, vol. 29, pp. 4027-4040, 2020, doi: 10.1109/TIP.2020.2970248.

[8] R. Keys, "Cubic convolution interpolation for digital image processing," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 6, pp. 1153-1160, December 1981, doi: 10.1109/TASSP.1981.1163711.

[9] Olivier Rukundo and Hanqiang Cao, “Nearest Neighbor Value Interpolation” International Journal of Advanced Computer Science and Applications(IJACSA), 3(4), 2012. http://dx.doi.org/10.14569/IJACSA.2012.030405

[10] i. processing, «image processing,» [En ligne]. Available: imageeprocessing.com/p/contact-me.html.

[11] C. S. Tong and K. T. Leung, "Super-resolution reconstruction based on linear interpolation of wavelet coefficients," Multidimensional Systems and Signal Processing, vol. 18, pp. 153-171, 2007.

[12] M. Jalali, H. Behnam, F. Davoodi, and M. Shojaeifard, "Temporal super-resolution of 2D/3D echocardiography using cubic B-spline interpolation," Biomedical Signal Processing and Control, vol. 58, p. 101868, 2020.

[13] H. Liu, J. Xu, Y. Wu, Q. Guo, B. Ibragimov, and L. Xing, "Learning deconvolutional deep neural network for high resolution medical image reconstruction," Information Sciences, vol. 468, pp. 142-154, 2018

[14] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, 2016

[15] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV, vol. 13, pp. 184-199, Springer International Publishing.

[16] W. Sun and Z. Chen, "Learned image downscaling for upscaling using content adaptive resampler," IEEE Transactions on Image Processing, vol. 29, pp. 4027-4040, 2020.

[17] J. H. a. D. Kirsch, Machine learning for Dummies., 2018.

[18] B. Qian, J. Su, Z. Wen, D. N. Jha, Y. Li, Y. Guan, et al., "Orchestrating the development lifecycle of machine learning-based IoT applications: A taxonomy and survey," ACM Computing Surveys (CSUR), vol. 53, no. 4, pp. 1-47, 2020

[19] M. A. Nielsen, "Neural networks and deep learning," in Neural Networks and Deep Learning, San Francisco, CA, USA: Determination Press, 2015, pp. 15-24.

[20] C. Bento, « Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment,» 2021.

[21] N. Karayiannis and A. N. Venetsanopoulos, "Artificial neural networks: learning algorithms, performance evaluation, and applications," Springer Science & Business Media, vol. 209, 1992.

[22] IBM, «IBM,» 2021. [En ligne]. Available: https://www.ibm.com/topics/neural-networks.

[23] S. Walczak, "Artificial neural networks and other AI applications for business management decision support," International Journal of Sociotechnology and Knowledge Development (IJSKD), vol. 8, no. 4, pp. 1-20, 2016.

[24] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Artificial Neural Networks and Deep Learning for Genomic Prediction of Continuous Outcomes," in Multivariate Statistical Machine Learning Methods for Genomic Prediction, Cham: Springer International Publishing, 2022, pp. 427-476.

[25] H. Z. a. L. Z. J. Zhang, «An Overview of Neural Network Training Methods and Their Applications,» 2019.

[26] J. D. Bodapati and N. Veeranjaneyulu, "Feature extraction and classification using deep convolutional neural networks," Journal of Cyber Security and Mobility, pp. 261-276, 2019.

[27] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights into Imaging, vol. 9, pp. 611-629, 2018.

[28] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646-1654.