Mohamed Khider University of Biskra
Faculty of Science and Technology
Department of Electrical Engineering

MASTER THESIS

Science and Technology
Electronics
Embedded Systems

Réf.:

Presented and defended by:
Moustafa Remadna
Bilal Lahlali

On: Thursday, June 05, 2025

Enhancing Image Quality with Deep Learning-Based Super-Resolution

Jury:

Dr.	Zehani Soraya	MCA	University of Biskra	President
Dr.	Meriem fedias	MCB	University of Biskra	Examiner
Dr.	Medouakh Saadia	MCB	University of Biskra	Supervisor

Academic year: 2024 – 2025

Mohamed Khider University of Biskra
Faculty of Science and Technology
Department of Electrical Engineering

MASTER THESIS

Science and Technology
Electronics
Embedded Systems

Réf.:

Submitted and defended by:
Moustafa Remadna
Bilal Lahlali

On: Thursday, June 05, 2025

Enhancing Image Quality with Deep Learning-Based Super-Resolution

In: Thursday, June 05, 2025

Presented by:	Favorable opinion of the supervisor:
Moustafa Remadna	Dr. MEDOUAKH Saadia
Bilal Lahlali	

Favorable opinion of the jury president

Stamp and signature

I thank God for granting me the strength to accomplish this work and move forward.

I would also like to express my sincere thanks and deep gratitude to my beloved family, who have been a constant source of support and strength throughout my studies.

I would like to thank everyone who contributed to the success of my internship and helped me in writing this thesis.

I express my deepest appreciation to my thesis supervisor, Mrs. Saadia Madouakh, for her supervision, guidance, assistance, and valuable advice.

My special thanks go to Professor ilyas Ben Aissa for his continuous support, insightful guidance, and encouragement throughout this work.

I also extend my heartfelt thanks to all the professors, and to everyone who has guided me through their words, writings, advice, and constructive criticism.

﴿وَقُلْ رَبِّ زِدْنِي عِلْمًا﴾

— سورة طه، الآية 114 —

To the ones whose prayers were the foundation of our success,

whose hands toiled to build our future...

***We dedicate this work to our beloved parents**, symbols of unconditional love and sacrifice.*

***To our dear brothers and sisters**, our constant source of support and joy this achievement is as much yours as it is ours.*

***To our friends**, who shared with us moments of hardship and triumph, thank you for your shared effort and dedication.*

***To our esteemed professors**, who generously shared their knowledge and guidance,*

*and **to our supervisor** your support and insight have meant the world to us.*

With deepest gratitude and respect

Abstract:

In recent decades, the field of image processing has received increasing attention, particularly in image super-resolution techniques, due to their significant importance in various critical applications such as medicine, surveillance, satellite imagery, and digital media. This technique involves converting a low-resolution image into a high-resolution one while preserving and realistically restoring its details. Although traditional methods, such as interpolation-based upscaling, exist to achieve this goal, they often fail to recover fine details. For this reason, recent research has turned to deep learning techniques, which have shown remarkable efficiency in this domain thanks to their ability to learn from large datasets and understand complex visual patterns. In this work, we aim to study the application of deep learning techniques for image super-resolution using the Enhanced Deep Super-Resolution (EDSR) model, which is considered one of the most efficient deep learning models for image super-resolution due to its deep and streamlined architecture. This design helps enhance performance and stability by eliminating unnecessary layers. Despite its high effectiveness in recovering fine image details, the model may face challenges in certain complex scenarios—such as images with high noise, images with extremely fine details missing from the original, or in the presence of severe distortions in the low-resolution input. This is due to its reliance on visual features extracted from the input data, which may not always be sufficient to perfectly reconstruct a high-resolution image in all cases.

Résumé :

Au cours des dernières décennies, le domaine du traitement d'images a suscité un intérêt croissant, en particulier pour les techniques de super-résolution d'images, en raison de leur importance significative dans diverses applications critiques telles que la médecine, la surveillance, l'imagerie satellitaire et les médias numériques. Cette technique consiste à convertir une image de basse résolution en une image de haute résolution tout en préservant et en restaurant de manière réaliste ses détails. Bien que des méthodes traditionnelles, telles que l'agrandissement basé sur l'interpolation, existent pour atteindre cet objectif, elles échouent souvent à récupérer les détails fins. Pour cette raison, les recherches récentes se sont tournées vers les techniques d'apprentissage profond, qui ont montré une efficacité remarquable dans ce domaine grâce à leur capacité à apprendre à partir de vastes ensembles de données et à comprendre des motifs visuels complexes. Dans ce travail, nous visons à étudier l'application de techniques Deep Learning pour la super-résolution d'images en utilisant le modèle EDSR (Enhanced Deep Super-Résolution), considéré comme l'un des modèles d'apprentissage profond les plus efficaces pour la super-résolution d'images, en raison de son architecture à la fois profonde et simplifiée. Ce design contribue à améliorer les performances et la stabilité en éliminant les couches non essentielles. Malgré sa grande efficacité dans la récupération des détails fins d'une image, le modèle peut rencontrer des défis dans certains scénarios complexes, tels que les images avec un bruit élevé, celles contenant des détails très fins absents de l'original, ou encore en présence de fortes distorsions dans l'image d'entrée à basse résolution. Cela est dû à sa dépendance aux caractéristiques visuelles extraites des données d'entrée, qui ne sont pas toujours suffisantes pour reconstruire parfaitement une image en haute résolution dans toutes les situations.

الملخص :

في العقود الأخيرة، حظي مجال معالجة الصور باهتمام متزايد، خاصة في تقنيات تحسين دقة الصور (Image Super-Resolution)، لما لها من أهمية بالغة في العديد من التطبيقات الحيوية مثل الطب، والمراقبة، وصور الأقمار الصناعية، والوسائط الرقمية. وتتمثل هذه التقنية في تحويل صورة منخفضة الدقة إلى أخرى عالية الدقة، مع الحفاظ على التفاصيل واستعادتها بشكل واقعي .ورغم وجود طرق تقليدية لتحقيق ذلك، كالتكبير عبر الاستيفاء(Interpolation) ، إلا أن هذه الأساليب غالبًا ما تفشل في استرجاع التفاصيل الدقيقة. ولهذا، توجهت الأبحاث الحديثة إلى الاستفادة من تقنيات التعلم العميق، والتي أظهرت كفاءة ملحوظة في هذا المجال بفضل قدرتها على التعلم من البيانات الضخمة وفهم الأنماط البصرية المعقدة .في هذا العمل، نهدف إلى دراسة تطبيق تقنيات التعلم العميق لتحسين دقة الصورة باستخدام نموذج (Enhanced Deep Super-Resolution) EDSR، الذي يُعد من أكثر نماذج تحسين دقة الصور كفاءةً في تطبيقات المعالجة الحديثة، بفضل بنيته العميقة وخلوه من الطبقات غير الضرورية، مما يساهم في تعزيز الأداء والاستقرار .وعلى الرغم من فعاليته الكبيرة في استعادة التفاصيل الدقيقة، إلا أن النموذج قد يواجه تحديات في بعض السيناريوهات المعقدة، مثل الصور ذات الضوضاء العالية، أو الصور التي تحتوي على تفاصيل دقيقة جدًا مفقودة في الأصل، أو عند وجود تشويهات كبيرة في الصورة المنخفضة الدقة .ويُعزى ذلك إلى اعتماده على الخصائص البصرية المُستخلصة من البيانات المُدخلة، والتي قد لا تكون كافية دائمًا لإعادة بناء صورة عالية الدقة بشكل مثالي في جميع الحالات.

Summary

Chapter III: Experimental Setup and Performance Evaluation

List of Figures

Chapter I: An Overview of Image Super-Resolution

Chapter II: Image super-resolution using deep learning

Chapter III: Experimental Setup and Performance Evaluation

Listes of tables :

List of abbreviations:

SR: super-resolution

HR: high resolution

LR: low resolution

AI: Artificial Intelligence

DL: Deep Learning

ML: Machine Learning

ANN: A neural network

CNN: Convolutional Neural Networks

EDSR: Enhanced deep super-resolution network

SRCNN: Super-Resolution Convolution neural network

GAN: generative adversarial network

SRGAN: Super-Resolution generative adversarial network

SRResNet: Super-Resolution Residual Network

PSNR: Peak Signal-to-Noise Ratio

SSIM: Structural Similarity

General introduction

With the arrival of the contemporary digital age, high-resolution images have applications in a majority of fields, such as medical imaging, surveillance and security, video enhancement, and more. However, limitations within imaging sensors, storage, and bandwidth typically result in low-resolution images, with negative impact on the functionality of computer vision-based systems. this is here where super-resolution restoration comes to the rescue, and its function is to reconstruct a high-resolution image from a low-resolution image. Super-Resolution (SR) technology aims to enhance the image resolution using single or multiple data acquisitions.

Image super-resolution (SR) is one of the vital image processing techniques that enhances the resolution of an image in the field of computer vision. In the last two decades, significant progress has been made in the field of super-resolution, especially utilizing deep learning approaches. In the past, most conventional approaches in this field have a tendency to rely on interpolation techniques or hand-crafted features, but they are not able to preserve the fine details and inherent texture of the image. In contrast, deep learning-based approaches have made significant progress, due to their ability to learn the complex relationship between low- and high-resolution images from large datasets.

A variety of deep learning methods have been developed to address super-resolution (SR) tasks, starting from early convolutional neural network (CNN) approaches, such as SRCNN to recent promising SR approaches using generative adversarial network (GAN), such as SRGAN. Among these, the Enhanced Deep Super-Resolution network (EDSR) stands out as one of the most prominent and high-performing deep learning models for SR tasks, It uses a convolutional neural network architecture based on deep residual learning. EDSR removes redundant units, such as batch normalization, to improve performance and accuracy and reduce computational complexity. The model has demonstrated effectiveness in reconstructing lost fine details in low-resolution images, making it suitable for applications in real-world situations that require high-quality images.

The main objective of this work is to study single-image super-resolution using Enhanced Deep Super-Resolution (EDSR) model. We chose this technique because it represents a giant leap forward for image improvement based on deep learning methods. EDSR is built upon deep Convolutional Neural Networks (CNNs) and is designed to reconstruct high-resolution images from their low-resolution counterparts with remarkable accuracy and visual quality.

We have chosen to organize our study around three main chapters as follows:

- the first chapter introduces the concept of image super-resolution (SR), its importance, and applications. It contrasts traditional methods with deep learning, highlighting the superiority of deep learning and current limitations. It provides a comparison between deep learning and traditional super-resolution (SR) methods.

- the second chapter presents the basics of image super-resolution algorithms by deep learning and focuses on convolutional neural networks (CNNs) and which form the basis of the EDSR model we have chosen in this work .
- In the third chapter, we present experiments on image super-resolution using the EDSR model with various datasets, and we discuss the obtained results. In addition, we explain how to build a website to apply our method.

We conclude this thesis with a general conclusion and the perspectives.

Chapter I:

An Overview of Image Super-Resolution

I.1 Introduction

In this chapter, we will present an overview of image super-resolution (SR), a technique in computer vision that focuses on reconstructing high-resolution images from low-resolution inputs. The main goal of super-resolution is to enhance the clarity and detail of images, which is crucial in various fields such as medical imaging, satellite imagery, and surveillance.

Traditional SR methods often relied on interpolation techniques, which frequently resulted in blurred images and loss of fine details. With the advent of deep learning, more advanced approaches have been developed, leading to significant improvements in image quality. These modern methods utilize neural networks to predict and generate high-resolution images, capturing intricate details more effectively. Despite these advancements, challenges remain, particularly in ensuring that SR models generalize well to real-world images and in managing computational demands.

I.2. Image Super-Resolution

I.2.1. Definition of Image:

An image is a visual representation of something, such as a photograph, drawing, scene, a person, or an idea. It is captured or created using visual means such as photography, drawing, or digital computing. An image can be two-dimensional (such as photographs or drawings) or three-dimensional (such as models and sculptures).

In digital form and the field of computer science and computer vision, an image is characterized as a matrix of numerical values that indicate the color intensity or gray levels for each pixel it contains. Image it is a two-dimensional array of size M x N where M is the number of rows and N is the number of columns in the array. A digital image is made up of a finite number of discrete picture elements called a pixel. The location of each pixel is given by coordinates (x, y) and the value of each pixel is given by intensity value f. Hence, the elements in a digital image can be represented by f (x, y). These images are used in several applications such as artificial intelligence, image processing, and pattern recognition. For this purpose, an image is a picture that was created or copied and stored in electronic form [1]

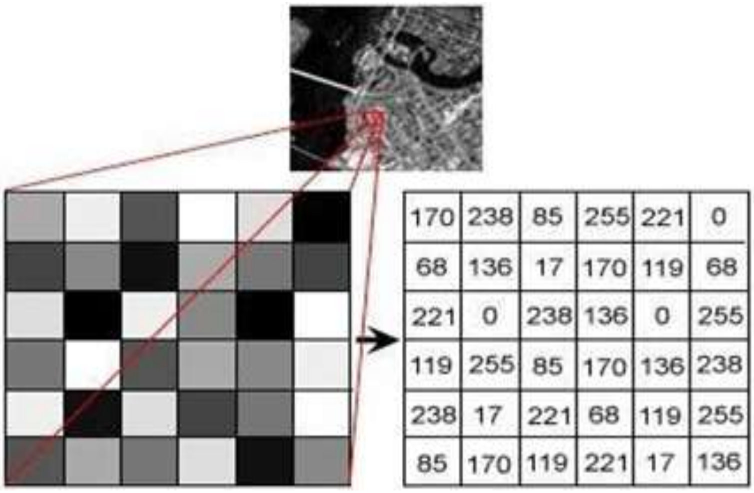


Figure I.1: digital image [2]

I.2.2. Definition of Image Resolution:

Image resolution is the level of detail of an image. The term applies to digital images, film images, and other types of images. "Higher resolution" means more image detail. Image resolution can be measured in various ways. Figure 1.2 shows how the same image can appear at different pixel resolutions [3].

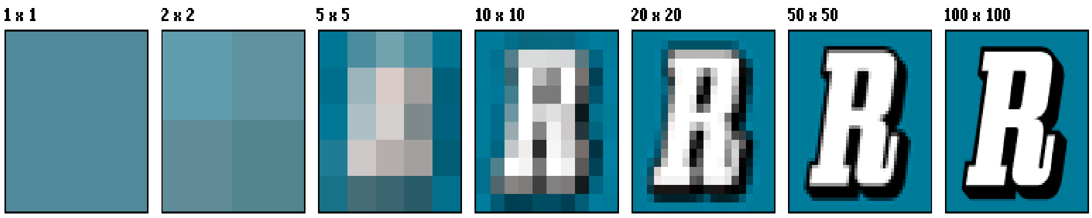


Figure I.2 : Resolution illustration [3]

In digital photography, resolution is the level of detail contained in an image. More specifically, it refers to the number of pixels that exist within that image. The higher the resolution, and the richer the pixel count, the more detail and definition you will see. (See figure 1.2) [4].

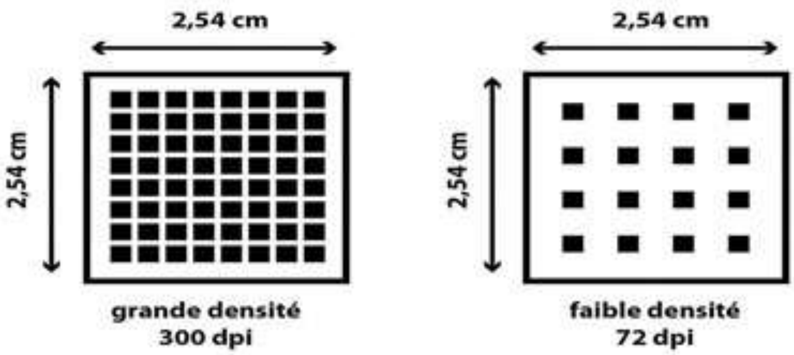


Figure I.3 : diagram of digital image resolution [5]

I.2.3. How is Image Resolution Measured:

The resolution of a digital image is measured using its pixels; specifically in pixels per inch (PPI). For printing, picture resolution is measured by dots per inch (DPI) (see figure 1.4).

- The higher the image resolution, the more pixels are bunched together — which creates a smoother, more detailed image.
- The phrase 'high-resolution' is often used synonymously with quality — as vivid, crisp images are what we typically associate with good photography. However, it's important to remember that resolution is just one factor that can affect image quality [4].

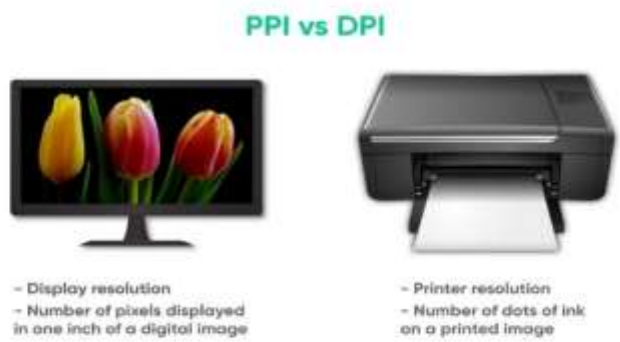


Figure I.4: difference between PPI and DPI [6]

I.2.4. Comparison between Low-Resolution and High-Resolution Images:

When comparing high and low resolutions, there are two characteristics to consider [7] :

- High-resolution images provide higher quality, less distortion and more pixels but come in larger files sizes.
- Low-resolution images are lower in quality and pixels and tend to have smaller file sizes.

Figure I.5 : Comparing a low resolution and a high-resolution image [8]

I.2.5. Limitations of low-resolution images:

Low-resolution images suffer from a lack of detail and clarity because they contain fewer pixels to represent the visual information. This result in images that appear blurry, pixelated, and less sharp compared to high-resolution counterparts, making it difficult to discern fine details such as text, facial features, or intricate patterns. Low-resolution images have several significant limitations that affect their use in various fields like computer vision, photography, medical imaging, and more. Here are some key drawbacks:

I.2.5.1. Loss of Details:

Blurry or pixelated images often occur when the resolution is too low to accurately represent the details in the original content. When you enlarge a low-resolution image, the limited number of pixels get stretched, resulting in a loss of clarity. Each pixel becomes more noticeable, causing the image to appear blocky or pixelated [9].

I.2.5.2. Pixelation:

Image with low quality have fewer individual pixels, making them look pixelated, hazy, or lacking in detail. This makes watching image less fun and make it harder to see small details [10].

I.2.5.3. Poor Print Quality:

On screen, your images are usually displayed at 72dpi, while the optimal resolution for a good print is 300dpi. The risk of printing an image at a low resolution is that it may produce unsightly pixelation or noise [11].

I.2.5.4. Restricted Editing:

Editing may be harder with low-quality images because fewer pixels exist. This may restrict your ability to improve the image’s quality or add extra effects [10].

I.2.5.5. Limited Potential for Scaling:

Low resolution image cannot be scaled up to larger displays or higher resolutions without losing quality [10].

I.2.6. Definition of Image Super-Resolution:

Super resolution is an advanced image processing technique used to enhance the resolution of an image, making it clearer and more detailed. It works by reconstructing or predicting finer details that may not be explicitly present in the original lower-resolution image [12]. It is the technique of reconstructing high-resolution (HR) images from low-resolution (LR) images, as shown in the above figure 1.5.

Super resolution is an important class of image processing techniques in computer vision and has numerous real-world applications, such as medical imaging, satellite imaging to surveillance and security, astronomical imaging.... With the advancement in deep learning techniques in recent years, deep learning-based SR models have been intensively studied and often achieve state-of-the-art performance on various benchmarks of SR. A number of deep learning methods have been developed to address SR issues, from the earliest CNN-based method to the latest promising GANs-based SR methods [13].

I.2.7. Importance of enhancing image quality:

High resolution photographs are ideal when you need to deliver crisp and clear content—the degree to what you need depends on your project and goal, but here are some examples: [7]:

- **Posters:** If you're creating a life-size poster, you want to ensure you're working at a high-quality PPI or DPI – since larger images are going to require more details to avoid looking blurry. That said, if it is going to be viewed from a distance, intermediate resolution values can sometimes strike the right balance.
- **Artwork and fine photography:** Depending on the type of artwork, a high resolution may better convey the subtleties of what you've created. It may be worth considering what the artwork will be used for and how it will be presented as well—for example will your fine art photography work be on display at showcases or exhibitions? Or will it go into a quality online portfolio?
- **Nature photography:** If you're producing landscapes or images of nature you want to catch all the details that bring those scenes to life. High-res photos can bring out the wonder and professionalism in what you capture.
- **Printing:** When creating work for print, a high resolution is essential to ensure quality imagery. Images that go into print are usually a minimum of 300 DPI.

I.2.8. Applications of super-resolution

Super-resolution has numerous applications across a wide range of fields where enhancing image or video resolution is crucial. The following are some of the most significant applications [14]:

I.2.8.1. Satellite image processing:

Researchers are looking to leverage super-resolution capabilities with images captured from satellites stationed in Earth's orbit to monitor geographical changes over time. This could also be useful for governments, as it could enable them to observe certain area of interest.

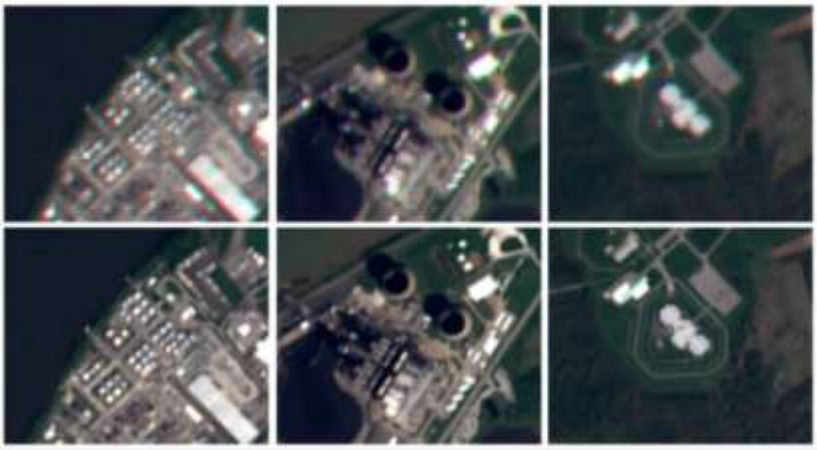


Figure I.6 : Satellite image before and after super-resolution [12]

I.2.8.2. Medical Image Processing:

Several medical image types require massive scan times that can get exponentially larger with higher resolution requirements. This trade-off is short-circuited with super-resolution to get a better look, allowing doctors to maintain enough resolution to identify details while reducing the amount of time the patient must spend in the process. It also helps in identifying objects at a microscopic level, allowing us to go further than what most optical microscopes allow and enhance details in biological samples.

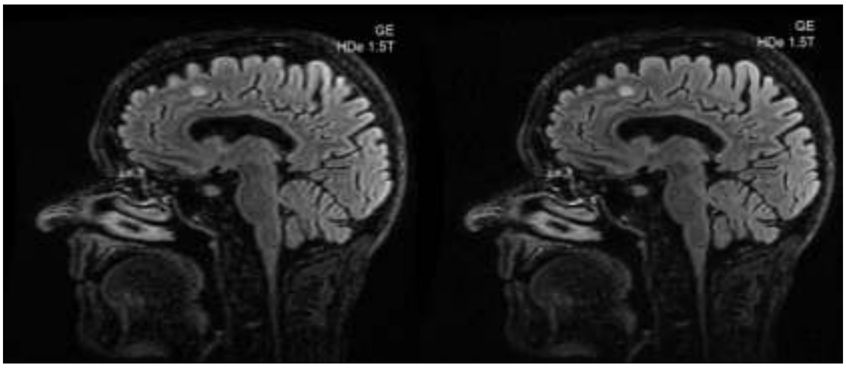


Figure I.7: Medical MRI scans before and after super-resolution [14]

I.2.8.3. Multimedia Industry and Video Enhancement:

Super-resolution is actively used in games to improve rendered textures and images and provide a better experience. It also helps in upscaling the game screen to the user's display resolutions to give the best experience available.

A lot of computer-generated imagery is often rendered in real-time in games and the animation industry. Offsetting this with pre-generated images upscaled to a particular resolution saves time and compute for the end-user.

Figure I.8: Stills from a game before and after super-resolution [14]

I.2.8.4. Astrological Studies:

Super-resolution can also be made to work beyond Earth. Images captured from probes and satellites orbiting other planets or celestial bodies sometimes undergo super-resolution to help researchers identify important objects. These kinds of images often require capturing faint objects with long exposure times, leading to unwanted artifacts or blurs. Super-resolution can reduce this effect to a large extent.

Figure I.9: Astronomy images before and after super-resolution [14]

I.2.8.5. Surveillance & Security:

It's common to see movies in which the boss in a police control room asks the officers to zoom into an image and sharpen it to identify important clues and pieces of information that may be relevant to the case.

This is an interesting example of how super-resolution is abstracted away, while being an important step to make the image clearer. In security cameras, super-resolution can be used to improve the quality of already-captured or live footage, allowing for better identification of objects and people. This can be crucial in forensic analysis and monitoring public areas for signs of distress.

Figure I.10 : Security camera image before and after super-resolution [14]

I.2.9. Image Super-Resolution Challenges:

Image super-resolution (SR) has the goal to enhance the image resolution, facilitating clearer and higher-detail images. Despite significant advancements, several challenges persist in this field [14][16][17][18]:

I.2.9.1. Evaluation Metrics:

The standard evaluation metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) may conflict with human visual perception of image quality. The conflict may result in SR models that are good under quantitive comparison but produce images that are not even rated as high-quality by humans.

I.2.9.2. Computational Efficiency:

High-performance SR models tend to demand considerable computation effort, rendering them less practical for real-time applications or implementation on less powerful devices. Sustaining model performance while being computationally efficient continues to be a major challenge.

I.2.9.3. Lack of High-Quality Training Data:

Enhanced resolution of image requires low-resolution images and their corresponding high-resolution counterparts, which is difficult to obtain in some fields (such as medical imaging or satellite imagery). In some cases, synthetic data is used that may not represent reality well.

I.2.9.4. Demosaicing in Color Images:

Demosaicing, or reconstruction of full-color images from raw sensor data, is a critical step in color images. High-quality SR requires proper demosaicing because poor demosaicing can lead to color artifacts and degrade image quality.

I.3 Image Super-Resolution Approaches:

Super-resolution is a technique to improve the wide-ranging quality, sharpness of images, video frames, and multidimensional images with low resolution (LR) by increasing resolution [19]. The main approaches can be categorized as follows:

I.3.1 Traditional Approaches to Image Super-Resolution:

Super-Resolution reconstruction has been one of the most active research areas since the seminal work by Tsai and Huang in 1984. Many techniques have been proposed over the last two decades representing approaches from frequency domain to spatial domain, and from signal processing perspective to machine learning perspective. Early works on super-resolution mainly followed the theory of by exploring the shift and aliasing properties of the Fourier transform. However, these frequency domain approaches are very restricted in the image observation model they can handle, and real problems are much more complicated [20].

Below, we review the most popular traditional computer vision algorithms for super-resolution.

I.3.1.1 Interpolation-Based Methods:

Interpolation is the process of using known data to estimate values at unknown locations. This works in two directions and tries to achieve the best approximation of a pixel's intensity based on the values of surrounding pixels. As it's an approximation method image will always lose some quality when interpolated [21].

Image interpolation occurs in all digital photos at some stage whether this be in Bayer demosaicing or in photo enlargement. It happens anytime you resize or remap (distort) your image from one pixel grid to another. Image resizing is necessary when you need to increase or decrease the total number of pixels, whereas remapping can occur under a wider variety of scenarios: correcting for lens distortion, changing perspective, and rotating an image Even if the same image resize or remap is performed, the results can vary significantly depending on the interpolation algorithm, The most widely used methods for image interpolation are nearest neighbor, bilinear, and bicubic interpolation [22].

a) Nearest Neighbor Interpolation:

The simplest form of interpolation. It assigns the value of the nearest pixel to the new pixel. This method is fast but can result in a blocky, less smooth image [23].

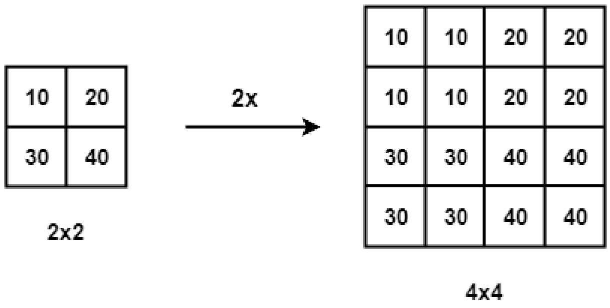


Figure I.11 : Nearest neighbor interpolation upsampling [24]

b) Bilinear Interpolation:

Considers the closest 2x2 neighbor hood of known pixel values (total 4 pixels) surrounding the unknown pixel and then takes the weighted average of these values to assign the unknown pixel, this will create smoother-looking images than the nearest neighbor and needs more processing[23].

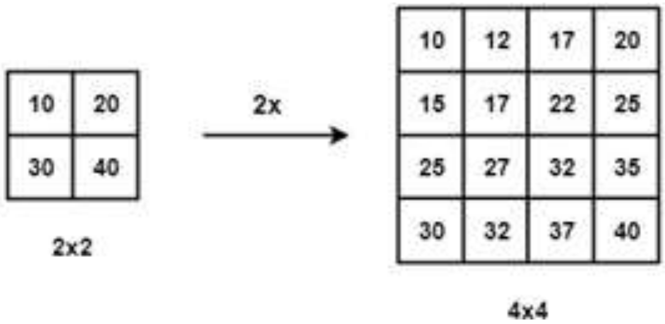


Figure I.12: Bilinear interpolation upsampling [23]

c) Bicubic Interpolation:

Bicubic interpolation considers the closest 16 pixels (a 4x4 environment) to estimate the new pixel values using cubic polynomials. This results in a higher quality image with smoother edges and finer detail compared to bilinear and nearest neighbor methods [23].

This produces noticeably sharper images than the nearest neighbor and bilinear interpolations. Bicubic interpolation is an ideal combination of processing time and output quality [21].

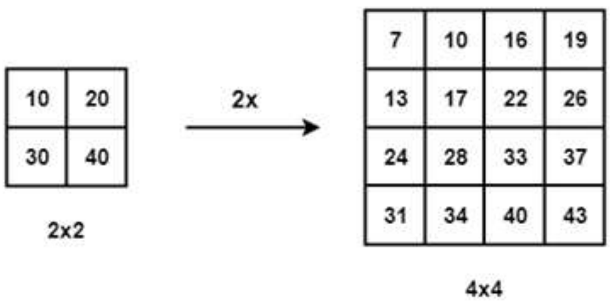


Figure I.13: bicubic interpolation upsampling [25]

I.3.2. Deep Learning Approaches for Image Super-Resolution:

Super-resolution has long been a fundamental problem in computer vision. Early methods involving explicit HR image reconstruction models were inadequate for real-world scenarios. With the rapid development of deep learning techniques in recent years, deep learning-based super-resolution methods have shown great potential in addressing super-resolution tasks, and often achieve the state-of-the-art performance on various benchmarks of super-resolution. A variety of deep learning methods have been applied to tackle super-resolution tasks, ranging from the early Convolutional Neural Networks (CNN) based method (e.g., SRCNN [27]) where Dong et al. [27] pioneered the use of a super-resolution convolutional neural network (SRCNN) for single image super-resolution, to recent promising super-resolution approaches using Generative Adversarial Nets (GAN) (e.g., SRGAN [30]). In general, the family of SR algorithms using deep learning techniques differ from each other in the following major aspects: different types of network architectures, different types of loss functions, different types of learning principles and strategies, etc [26]. Below are the key deep learning-based Image super-resolution techniques:

I.3.2.1. SRCNN (Super-Resolution Convolutional Neural Network):

SRCNN model appeared in 2014, it's the first CNN-based SISR model. In the method, the input LR image is mapped to the HR image by learning the end-to-end mapping, as shown in figure 1.14. This technique employs bicubic interpolation as a pre-processing step. After that, it extracts feature vectors from the image patches by convolution, which are then non-linearly mapped to find the most representative patches to reconstruct the HR image. SRCNN only uses convolutional layers, so it's possible to input images of any size, and its algorithm is not patch-based. The SRCNN model outperforms many" traditional" models [37].

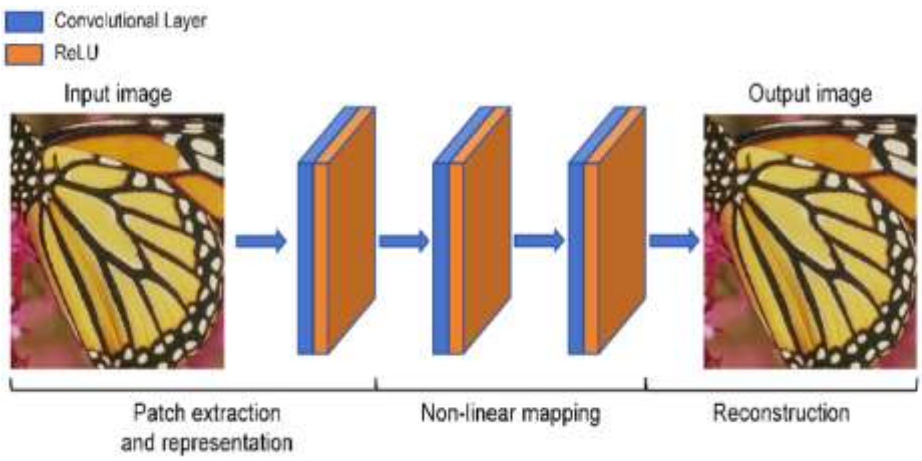


Figure I.14 : Overview of SRCNN [28]

I.3.2.2. SRResNet (Super-Resolution Residual Network):

SRResNet model appeared in 2016, it's based on a convolutional neural network added to a residual learning network structure. The main body includes two parts: a deep residual network and a sub-pixel convolution network. It uses the MSE as the loss function. The deep residual network adds a residual learning module to the convolutional neural network, which effectively solves problems of accuracy degradation and gradient dispersion in the deep network, greatly deepens the number of network layers, and ensures precision. Thus, the depth and precision of the training is effectively improved, which aids efficient feature extraction and reduces image noise. The main function of the sub-pixel convolution model is to increase the size and the accuracy of the enlarged image through sample learning. SRResNet uses a low-resolution image as its input and outputs a reconstructed high-resolution image, this method gives very good results [29].

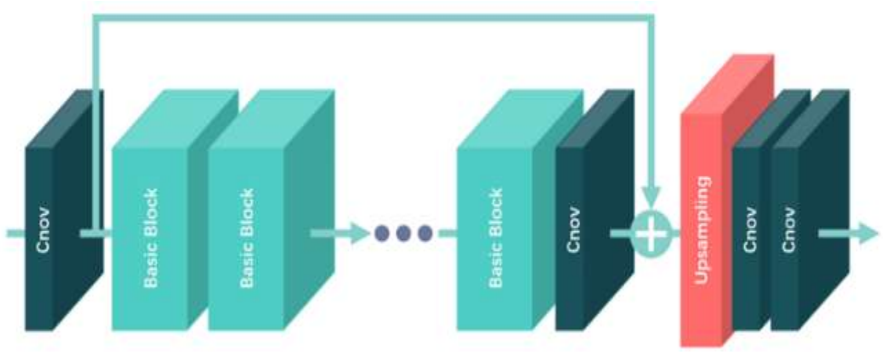


Figure I.15 : SRResNet Basic Structure [30]

I.3.2.3. SRGAN (Super-Resolution Generative Adversarial Network):

SRGAN model appeared in 2017, the concept of SRGAN is one of the first techniques that allows the model to achieve an upscaling factor of almost 4x for most image visuals. The idea of estimating and generating a high-resolution image from a low-resolution image is a highly challenging task. CNNs were earlier used to produce high-resolution images that train quicker and achieve high-level accuracy. However, in some cases, they are incapable of recovering finer details and often generate blurry images. The proposed SRGAN architecture combats most of these issues for generating high-quality, state-of-the-art images [31].

SRGAN architecture consists of two neural networks [32]: Generator Network (Figure 1.16); creates synthetic data from random noise to produce data so realistic that the discriminator cannot distinguish it from real data. Discriminator Network (Figure 1.17); acts as a critic, evaluating whether the data it receives is real or fake.

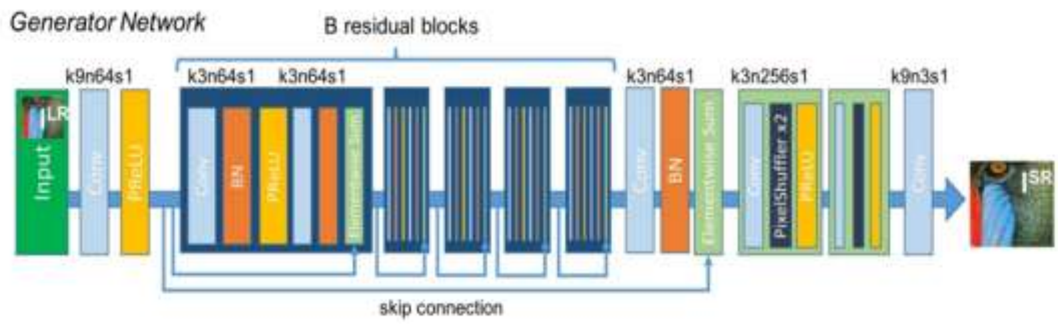


Figure I.16 : Architecture of Generator Network [31]

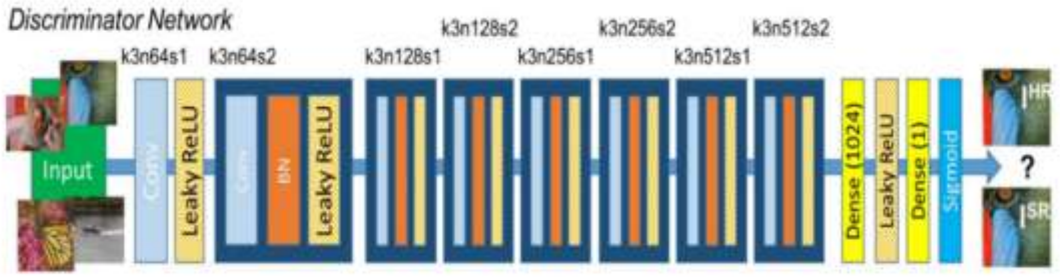


Figure I.17 : Architecture of Discriminator Network [31]

I.3.2.4. EDSR (Enhanced Deep Super-Resolution Network):

EDSR model appeared in 2017, it's a deep learning-based super-resolution technique. Super-resolution aims to enhance the resolution of an image or video, effectively upscaling it while trying to maintain or even improve image quality. EDSR is an enhancement of the earlier Deep Super-Resolution Network (DSRCNN). It's known for achieving state-of-the-art results in image super-resolution.

The EDSR architecture is based on the SRResNet architecture and consists of multiple residual blocks. It uses constant scaling layers instead of batch normalization layers to produce consistent results (input and output have similar distributions, thus normalizing intermediate features may not be desirable). Instead of using a L2 loss (mean squared error), the authors employed an L1 loss (mean absolute error), which performs better empirically [33].

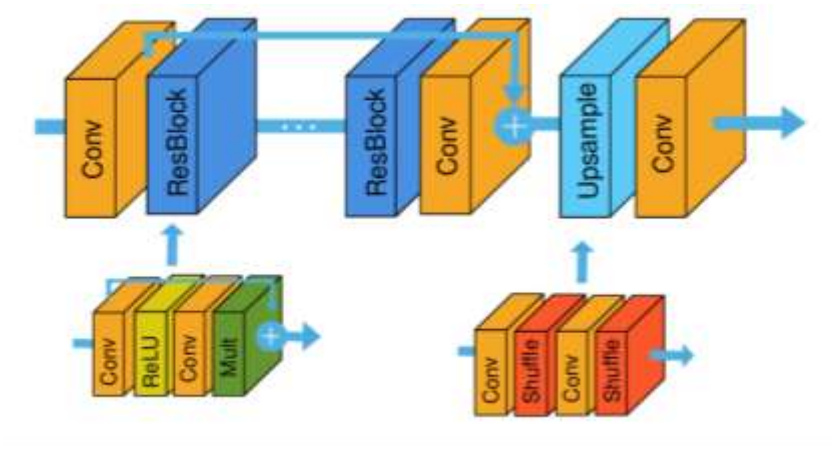


Figure I.18: EDSR Model Architecture blocks [34]

I.3.3. Comparison of Classical and Deep Learning-Based Super-Resolution methods:

A comprehensive comparison of image super-resolution techniques is essential for understanding their strengths, weaknesses, and suitability for various applications. Traditional methods, rooted in interpolation-based approaches and optimization algorithms, have long served as the cornerstone of image enhancement. These techniques, including bicubic interpolation, are known for their computational efficiency and straightforward implementation. However, their efficacy often diminishes when it comes to producing high-quality, visually appealing results.

In contrast, deep learning-based approaches have emerged as a transformative force in the field of image super resolution. Leveraging advanced neural network architectures such as Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), these methods have demonstrated remarkable performance in generating high resolution images with superior visual quality and perceptual similarity. By learning complex patterns and relationships from large-scale datasets of low- and high-resolution image pairs, deep learning-based techniques excel in preserving fine details and textures, thus overcoming the limitations of traditional methods.

Despite their impressive capabilities, deep learning-based approaches come with their own set of challenges. They require extensive computational resources and large-scale datasets for training, making them computationally intensive and resource-demanding. Additionally, the implementation of deep learning models necessitates expertise in neural network architectures and optimization techniques, adding to the complexity of deployment. Moreover, the performance of deep learning models is heavily reliant on the quality and quantity of training data, which may not always be readily available or representative of diverse real-world scenarios. The choice of a super-resolution technique depends on various factors, including computational constraints, application requirements, and the availability of training data. For scenarios where computational resources are limited, traditional methods may offer a practical solution due to their efficiency and simplicity. However, in applications where high-quality, visually pleasing results are paramount, deep learning-based approaches prove to be more effective despite their computational demands [35].

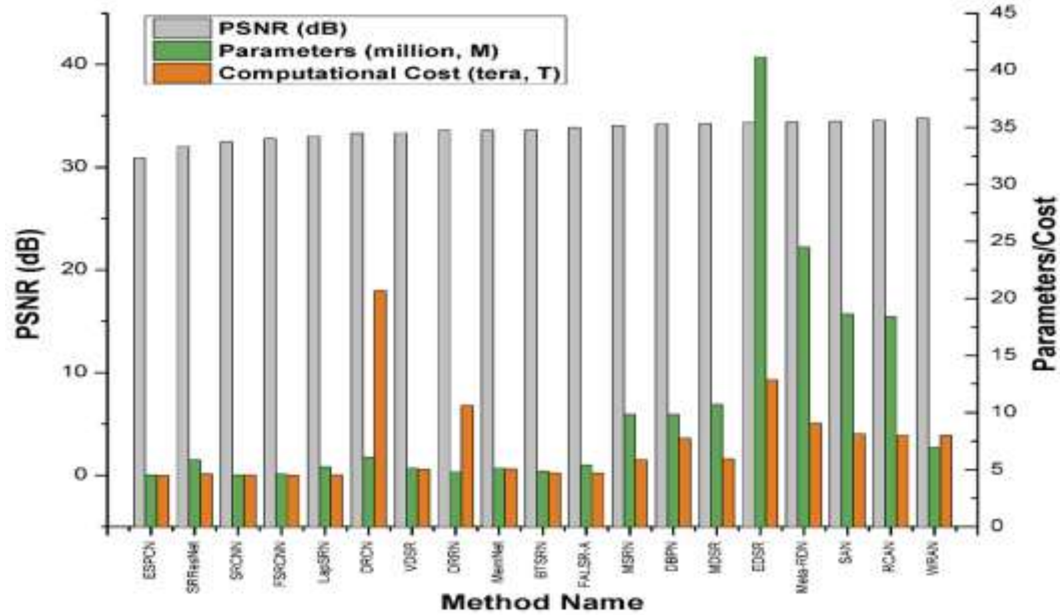


Figure I.19: Comparison of different method on scale PSNR [33]

I.4. Conclusion

The first chapter provided an overview of image super-resolution. Image super-resolution (SR) has taken huge transformations during the past seven decades, starting from traditional interpolation methods to contemporary deep learning paradigms. All this made through various steps resulted in amazing leaps forward regarding enhancement of image quality and resolution. Integration of SR techniques within various applications like medical imaging, astronomy, and security depicts how significant its presence has become for multiple domains. Despite all these advancements, there are still regions of challenge, such as developing models of good generalization to real-world examples and managing the computational resources involved in managing high-resolution image processing.

Chapter II: Image super-resolution using deep learning

II.1 Introduction:

In the last few years, machine learning and deep learning have revolutionized the way computers read and process visual data. Among all the computer vision tasks, image super-resolution (SR) has gained a very high level of interest due to its applications in real-world scenarios such as medical imaging, satellite imaging, video upscaling, and more.

This chapter provides a theoretical background necessary to study the super-resolution algorithms using deep learning techniques. The chapter begins with a discussion of the fundamental concepts of machine learning, then moves on to deep learning, highlighting its differences from conventional machine learning methods and emphasizing the significance of neural networks, particularly convolutional neural networks (CNNs) in image processing and image enhancement. Next, it presents the general background of the Enhanced Deep Super-Resolution (EDSR) model, a recent state-of-the-art deep learning architecture specifically designed to solve the single image super-resolution (SISR) problem.

II.2 Intelligence Artificial:

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy. Applications and devices equipped with AI can see and identify objects. They can understand and respond to human language. They can learn from new information and experience. They can make detailed recommendations to users and experts. They can act independently, replacing the need for human intelligence or intervention (a classic example being a self-driving car) [36].

II.3 Machine Learning (ML):

Machine learning is a branch of artificial intelligence focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their performance and accuracy through experience and exposure to more data. (see figure II.1) [37].

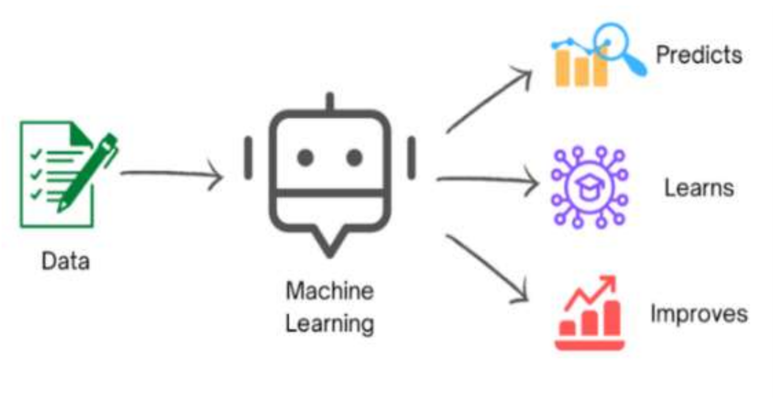


Figure II.1: Basic of Machine Learning [38]

II.3.1 Machine learning types:

Here's a quick look at some of the commonly used types in machine learning (ML):

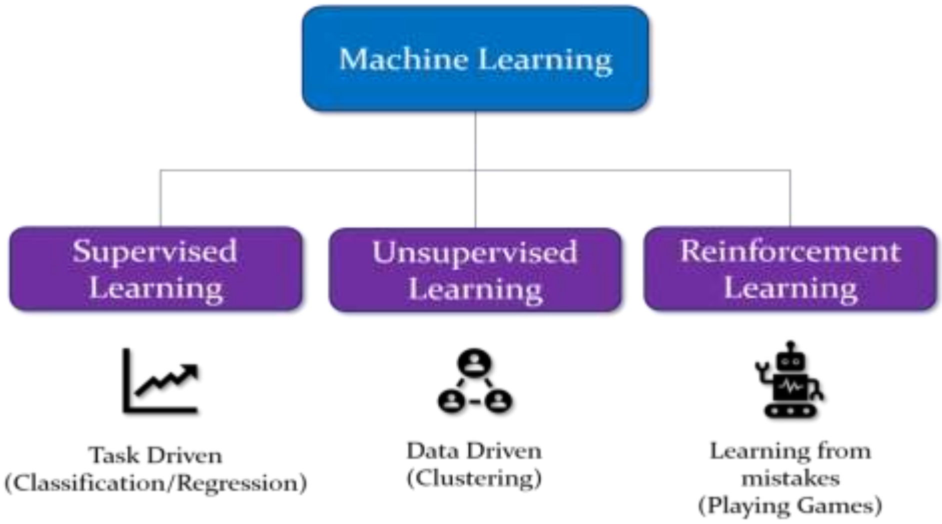


Figure II.2: Types of Machines Learning [39]

II.3.1.1 Supervised Learning:

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output (see figure II.3).

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

The vast majority of Machine Learning and Deep Learning problems use supervised learning [40].

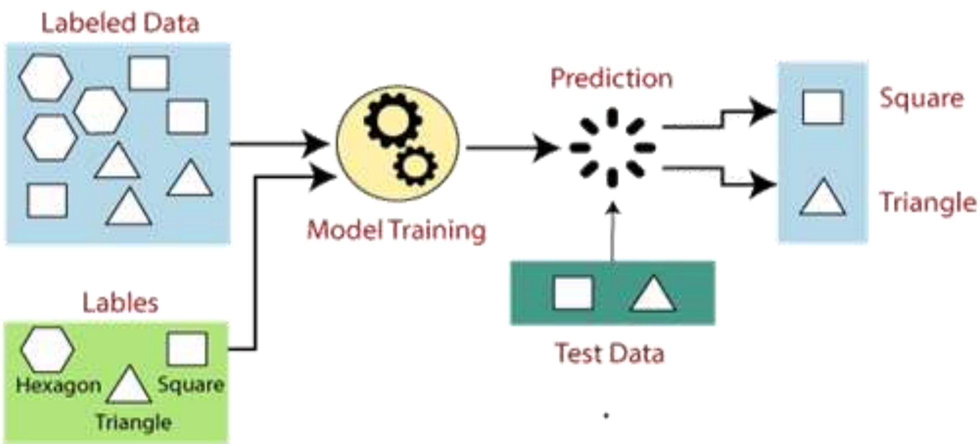


Figure II.3: how supervised Learning Works [40]

II.3.1.2 Unsupervised Learning:

Unsupervised learning is machine learning where the models are not supervised with the training dataset. Rather, the models itself discover the hidden information and patterns in the provided data. It's similar to the learning that happens in the brain of a person when learning new things [41].

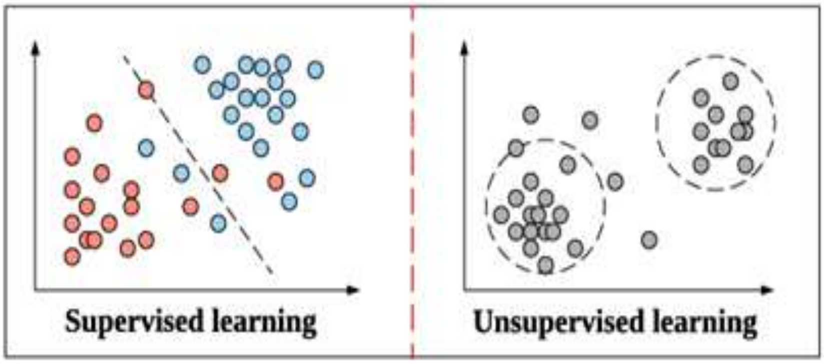


Figure II.4: Examples of Supervised Learning (Linear Regression) and Unsupervised Learning (Clustering) [42]

II.3.1.3 Reinforcement Learning:

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty. In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning. Since there is no labeled data, so the agent is bound to learn by its experience only. RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as game-playing, robotics, etc… [43].

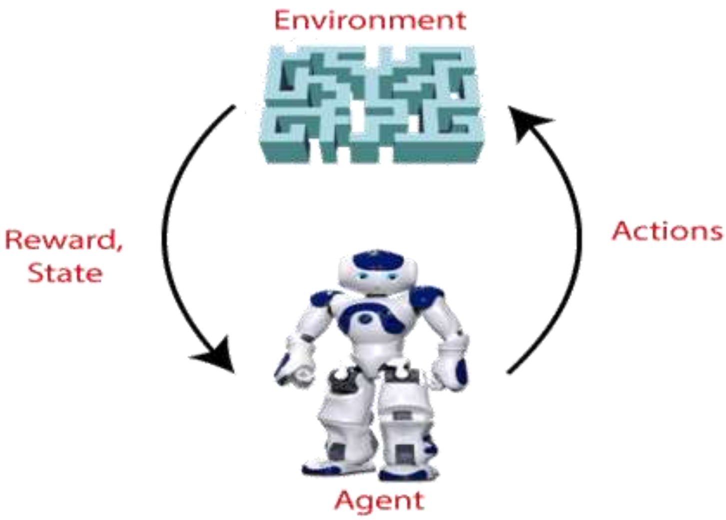


Figure II.5: Reinforcement Learning [43]

II.4 Deep Learning (DL):

Deep learning is one of the main technologies of machine learning. With Deep Learning, we are talking about algorithms capable of mimicking the actions of the human brain thanks to artificial neural networks. The networks are composed of dozens or even hundreds of "layers" of neurons, each receiving and interpreting the information of the previous layer [44].

A neural network consists of three or more layers: an input layer one or many hidden layers, and an output layer, as shown in figure II.*6*. Data is ingested through the input layer. Then the data is modified in the hidden layer and the output layers based on the weights applied to the se nodes. The typical neural network may consist of thousands or even millions of simple processing nodes that are densely interconnected [45].

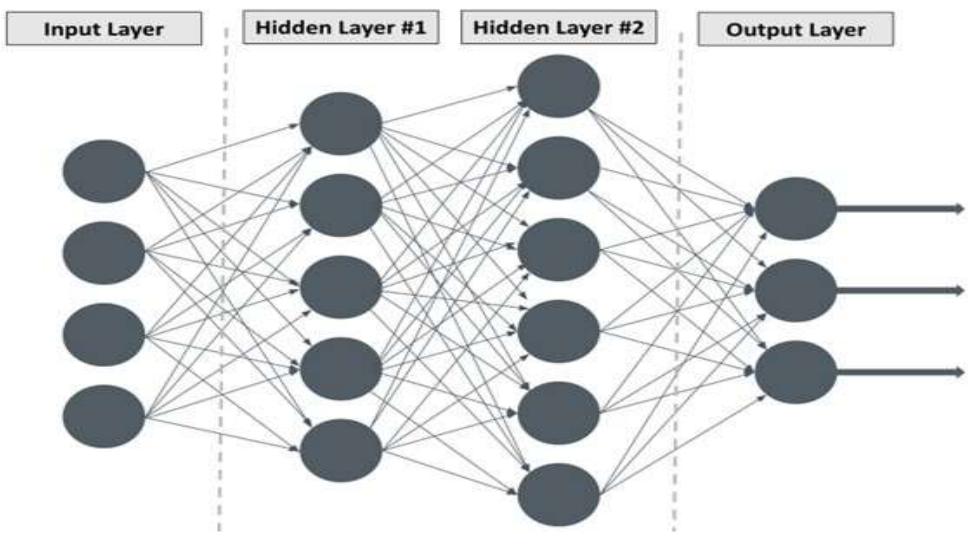


Figure II.6: A simple artificial neural network [46]

Various neural networks exist, each with a unique structure and function. The commonly neural networks used in today's technology are:

II.4.1 Convolutional Neural Networks (CNNs):

They are a specialized type of deep neural network used for processing data that has a grid-like topology, such as images. Hence, they excel at image recognition and analysis tasks. It is the best and most excellent in the field of identifying objects from images and videos, in addition to improving the resolution of images, as such EDSR model [47]. We will discuss the CNN architecture in detail in section II.6.

II.4.2 Generative Adversarial Networks (GANs):

GANs are a unique class of deep-learning architectures used for generative modeling through unsupervised learning. They utilize an adversarial training process to generate new, realistic data. GANs involve two main components: Generator Network (G), which creates synthetic data that is indistinguishable from real data but resembles its distribution; and the Discriminator Network (D), which plays the role of a critic, distinguishing between genuine and synthesized data [47].

II.5 Differences between Machine Learning and Deep Learning:

Machine Learning means computers learning from data using algorithms to perform a task without being explicitly programmed. Deep Learning uses a complex structure of algorithms modeled on the human brain. This enables the processing of unstructured data such as documents, images, and text. To break it down in a single sentence: Deep Learning is a specialized subset of Machine Learning which, in turn, is a subset of Artificial Intelligence [46].

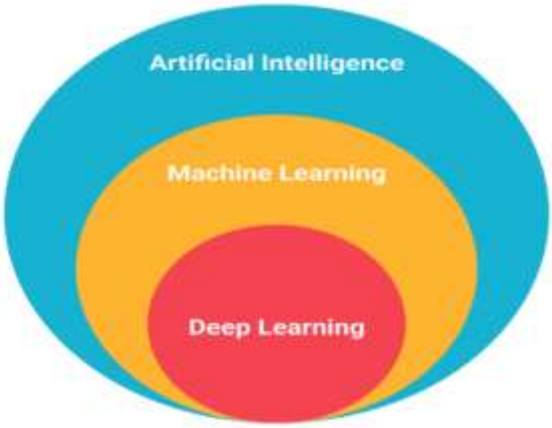


Figure II.7: The relationship between AI, machine learning, and deep learning [48].

II.6 Convolutional Neural Networks (CNNs):

Convolutional neural network (CNN) is one of the most popular and used of DL networks. It is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. Most super-resolution models based on CNN, so we have dug in deep with CNN by presenting the main components of it. Convolutional neural network indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers [49]. The role of the convolutional networks is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction [50]. The higher performance of convolutional neural networks with image, speech, or audio signal inputs set them apart from conventional neural networks.

A convolution neural network has multiple hidden layers that help in extracting information from an image. It becomes more complicated with each layer, detecting larger areas of the image. The important layers in CNN are (see figure II.8) [49]:

- Layer of convolution
- Layer of pooling
- Fully connected layer (FC).

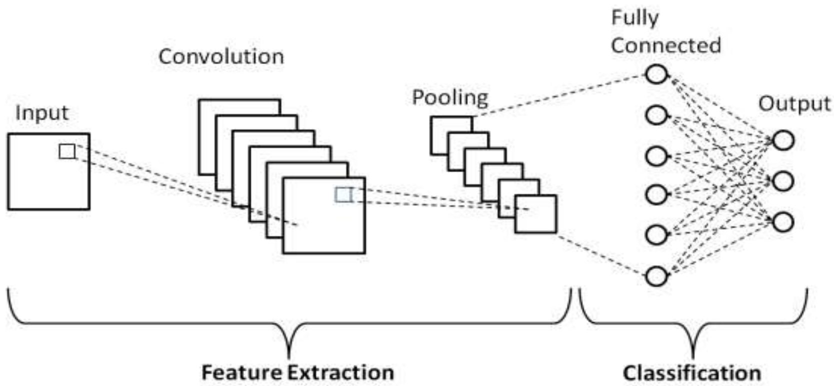


Figure II.8: Architecture of a Convolutional Neural Network (CNN) [51]

II.6.1 Convolutional Layer:

The input is going to be a color image, and it is a 3D matrix of pixels. Therefore, the input is going to be three-dimensional a height, a width, and a depth that corresponds to RGB in an image. We also have a feature detector, also referred to as a kernel or a filter, and this will slide over the receptive fields of the image, looking for if the feature is present. This process is referred to as a convolution. At each step, the product between the filter and the corresponding part of the image is calculated, as illustrate in figure II.9. This process produces a map called a feature map. The convolution process aims to extract the initial features from the image such as edges, shapes, corners... etc. It is the main layer in CNN [52].

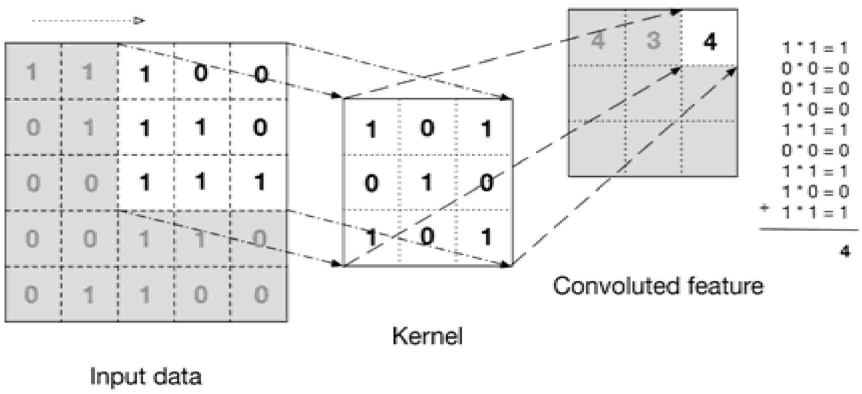


Figure II.9: how Convolutional layer Works [50]

II.6.2. Pooling Layer:

Pooling layer is used in CNNs to reduce the spatial dimensions (width and height) of the input feature maps while retaining the most important information. There are two (2) pooling approaches (figure II.10) [53] :

- **Max pooling:** As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- **Average pooling:** As the filter moves across the input, it calculates the average value within the receptive field to send to the output array

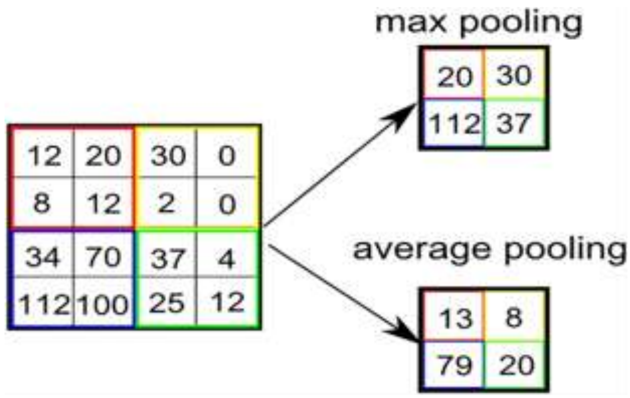


Figure II.10: Max pooling vs average pooling [54]

II.6.3. Fully-Connected Layer:

Fully connected (FC) layers are layers in a convolutional neural network (CNN) whose every neuron is connected to all neurons in the previous layer. It is utilized as the CNN classifier. These layers are used to transform extracted features of an image to the final output decision, e.g., classification of an image to classes (cat, dog, car, etc.). Convolutional layers are only good at extracting features. However, FC layers are what determine the final outcome. Without them, the model cannot make classification or predictive decisions [55].

II.6.4. Activation Function:

Activation functions of neural networks, including Convolutional Neural Networks (CNNs), play a critical role in determining the output of a neural node and the learning process. The activation functions introduce non-linearity into the model to enable it to learn and abstract complex data like images, video. Activation functions refer to mathematical functions utilized for all neurons in the neural network to determine whether to "activate" the neuron or not based on its input [56].

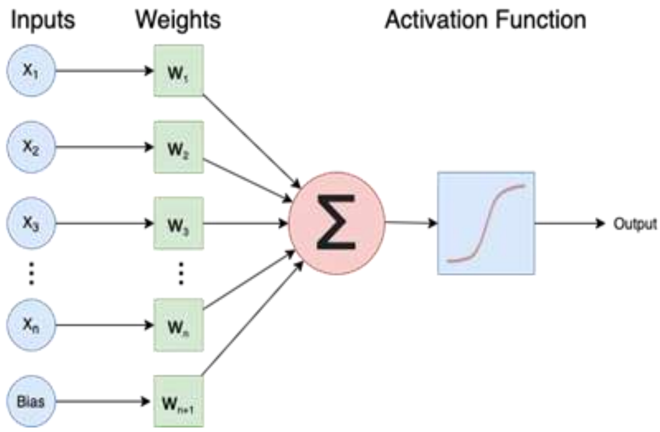


Figure II.11: Activation functions schema [57]

The following types of activation functions are most commonly used in CNN and other deep neural networks [56]:

- **The sigmoid activation function** maps input values to the range (0, 1), making it useful for binary classification problems.
- **The tanh activation function** is similar to sigmoid but outputs values between (-1, 1).
- **The ReLU activation function** is one of the most commonly used activation functions in CNNs. It returns the input value directly if it's positive, and zero otherwise. The main catch here is that the ReLU function does not activate all the neurons at the same time.

II.7. Super resolution by deep learning:

Image super-resolution (SR) problem, particularly single image super-resolution (SISR), has gained increasing research attention for decades. SISR aims to reconstruct a high-resolution image from a single low-resolution image. Recently, significant progress has been made in this field, especially utilizing deep learning methods. With the rapid development of deep learning techniques, deep learning-based SR models have been actively explored and often achieve the state-of-the-art performance on various benchmarks of SR.

Image super-resolution networks began with convolutional neural networks (CNNs), which were excellent in extracting image features. Over time, the need for deeper and more accurate networks set in, and ResNet emerged. ResNet uses residual blocks to allow deep networks to be trained and is even more image-classification-specific. SRResNet later emerged exclusively for image super-resolution. This was subsequently enhanced by the EDSR model, which significantly improved performance by eliminating redundant layers and deepening and refining the network to make it one of the best-performing models in the super-resolution domain.

In this section, we will elaborate the ResNet network, then we will discuss how it development into SRResNet network, and then will elaborate in detail EDSR network which we used to enhancing super-resolution images in our work.

II.7.1 Residual Networks (ResNet):

Residual Network is a deep Learning model used for computer vision applications [58]. After the first CNN-based architecture (Alex Net) that win the ImageNet 2012 competition, every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when we increase the number of layers, there is a common problem in deep learning associated with that called the Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus, when we increase number of layers, the training and test error rate also increases [59]. ResNet outperformed other architectures by winning the image classification task in ILVRSC 2015 by a substantial margin with top 5 error rate of 3.57 % and achieved remarkable results [58]. We can see from the figure II.12 that a 20-layer CNN architecture performs better on training and testing datasets than a 56-layer CNN architecture [60].

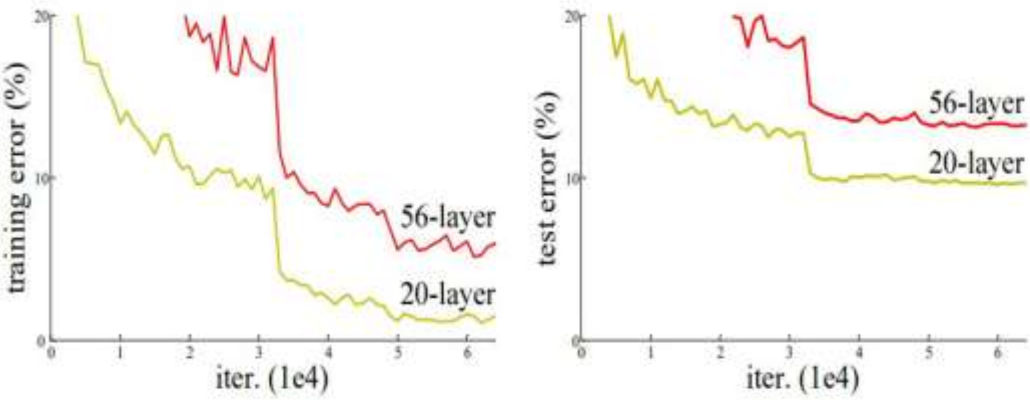


Figure II.12: Comparison of 20-layers vs 56-layers ResNet Architecture [61]

II.7.1.1 Why Residual Networks Matter in Deep Learning:

Residual Networks (ResNets) have transformed the science of deep learning by overcoming fundamental constraints in training very deep neural networks Among them [61]:

- Allows easier training of very deep neural networks, even those with hundreds of layers.
- Eradicates issues like vanishing/exploding gradients during training.
- This design has gone on to become the building block for most of today's leading models, including SRResNet, EDSR, and numerous others in computer vision.

II.7.2 Super-Resolution Residual Network (SRResNet):

SRResNet is an abbreviation for Super-Resolution Residual Network, a deep convolutional neural network (CNN) model, it was presented by the DeepMind team in 2017 [62]. SRResNet was utilized as the generic backbone for the building of SRGAN model. However, SRResNet is the core part that is responsible for enhancing the resolution of an image. To train deep neural networks more efficiently, the concept of residual blocks and skip-connections has been introduced in deep neural networks such as SRResNet.

SRResNet it is a customized ResNet, is a deep residual neural network specifically designed for super-resolution. The major strength of the SRResNet is the use of residual layers based on a technique of skip connections between two subsequent layers, which can effectively manage vanishing gradient problem encountered in deep neural networks [63].

II.7.2.1 SRResNet Architecture:

Figure II.13 illustrates the architecture of the SRResNet network, and its components are described in the Following [63]**:**

- The input image is convolved with filters of size 9 × 9 × 3 (This layer is used to extract the main features from the image.) followed by Parametric Rectified Linear Unit (PReLU) then passed from residual blocks.
- The output of each residual block is elementwise added with the block input to make a skip-connection (Elementwise sum)
- Each residual block has a convolutional layer with 3 × 3 filters followed by batch normalization and PReLU and then passed from another convolutional layer with filter size 3 × 3 and batch normalization.
- A final skip connection is built by passing the residual block's output from a convolution layer with filters of size 3 × 3 and batch normalization layer and the result is added with the input to the convolutional layer.
- Then the result is passed from UpSampling block
- UpSampling block consists of convolutional layer followed by UpSampling (Usually using PixelShuffle Methode) Layer and then ReLU.
- Output of the upsampling block is passed from a final convolutional layer with 3 filters of size 9 × 9. The output of three filters corresponds to RGB color values

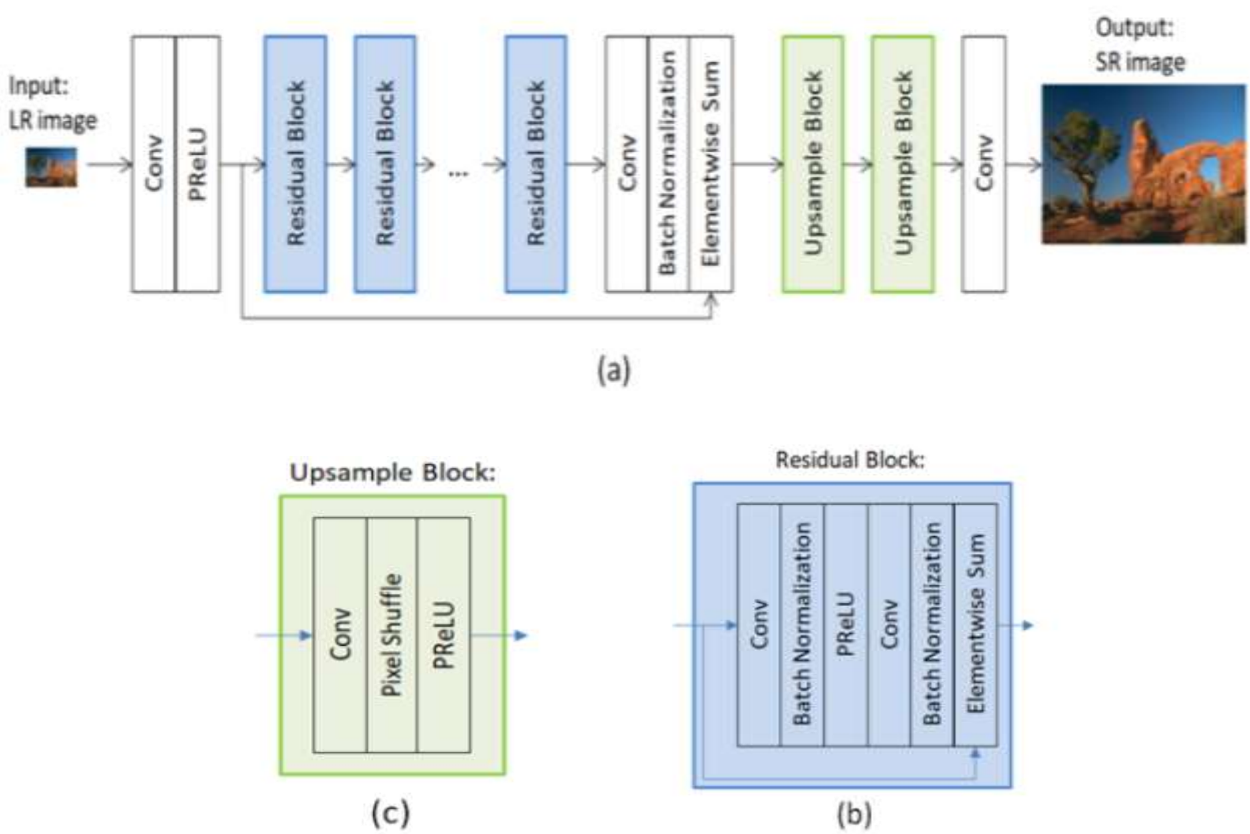


Figure II.13: (a) Architecture of the SRResNet network. (b) Residual block Architecture of SRResNet. (c) Upsampling block Architecture of SRResNet [64]

a) Residual Block:

Residual blocks are the main components of Residual Neural network. In a classical neural network, the input is transformed by a set of convolutional layers then it is passed to the activation function. In a residual network the input to the block is added to the output of the block creating a residual connection, as shown in figure II.13 (b). The goal of these blocks is to learn deep features without forgetting the original features [58].

b) UpSampling block:

The Upsampling Block is one of the core components of the SRResNet model, as shown in figure II.13. Once we pass the image through the remaining blocks and feature processing, we move on to the upsampling image

The Upsampling Block performs the upscaling of the spatial resolution of the image - meaning it increases the width and height dimensions. This is usually done using (figure II.13 (c)):

- A convolutional layer to increase the number of channels in a way that prepares them for upscaling.

- Pixel Shuffle, also known as sub-pixel convolution, which rearranges the channels into higher spatial dimensions, effectively converting channel depth into image size.

The pixel shuffle method will be explained in the section on EDSR Architecture [65].

II.7.2.2 Skip Connection:

Skip connection helps in forming the residual blocks. Skip connection consists of the input of the residual block that is bypassed over the convolutional layer and added to the output of the residual block (figure II.14).

When neural networks become too deep, the following problems begin to arise [61]:

- Learning becomes slower.
- Information is lost as it passes through layers.
- Gradients gradually fade during training.

That's why we need to use skip connections:

- Help the model retain the original information.
- Maintain a smooth flow of gradients during training.

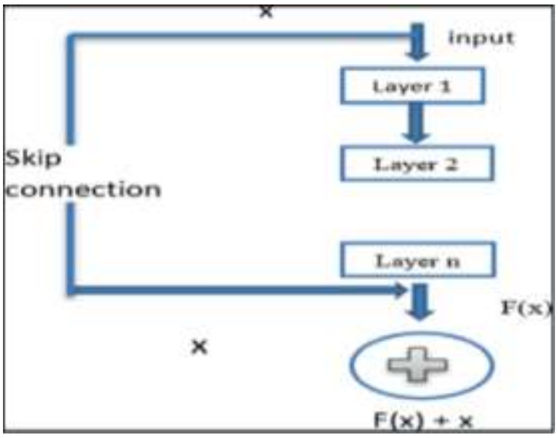


Figure II.14: SKIP connection [66]

In image super resolution tasks, the network is designed to preserve the details of the original image and improve resolution. Skip connections allow the network to pass the details to the next stage without deforming or modifying them. Any changes the network learns are directed towards improving the quality of the details or reconstructing lost details. Skip connections help in optimization process which leads to faster convergence and less training time. The skip connections in the residual networks leads to better generalization on the unseen data as the network can skip unnecessary or irrelevant information [58].

II.7.3 Enhanced Deep Super-Resolution Network (EDSR):

Deep Residual Networks (EDSR) for Single Image Super-Resolution is a deep learning algorithm created in July 2017 that significantly increases the resolution and visual quality of low-resolution images. Based on deep Convolutional Neural Networks (CNNs), EDSR is actually built on the Residual Network (ResNet) architecture, which has been optimized and refined especially for single image super-resolution (SISR).

In this section, we interest to study the architectural enhancements of EDSR compared to previous models ResNet and SRResNet. We also highlight how EDSR enhances the quality of image reconstruction without compromising computational efficiency, and this makes it a powerful tool when it comes to image super-resolution.

II.7.3.1 EDSR Architecture:

The EDSR architecture is based on the SRResNet architecture, consisting of multiple residual blocks, as illustrate in figure II.15. EDSR is a super-resolution model proposed after SRResNet. SRResNet successfully solved the problems of processing time and memory consumption, but ResNet used in SRResNet is a model architecture for image classification, which is not optimal for super-resolution [67].

Therefore, EDSR builds a more optimal model for super-resolution by removing unnecessary modules from ResNet. For example, Batch Normalization (BN) is removed because it loses range flexibility, removal of BN results in an improvement in accuracy. The BN layers consume memory, and removing them leads to up to a 40% memory reduction, making the network training more efficient [68], based on the research in [62]

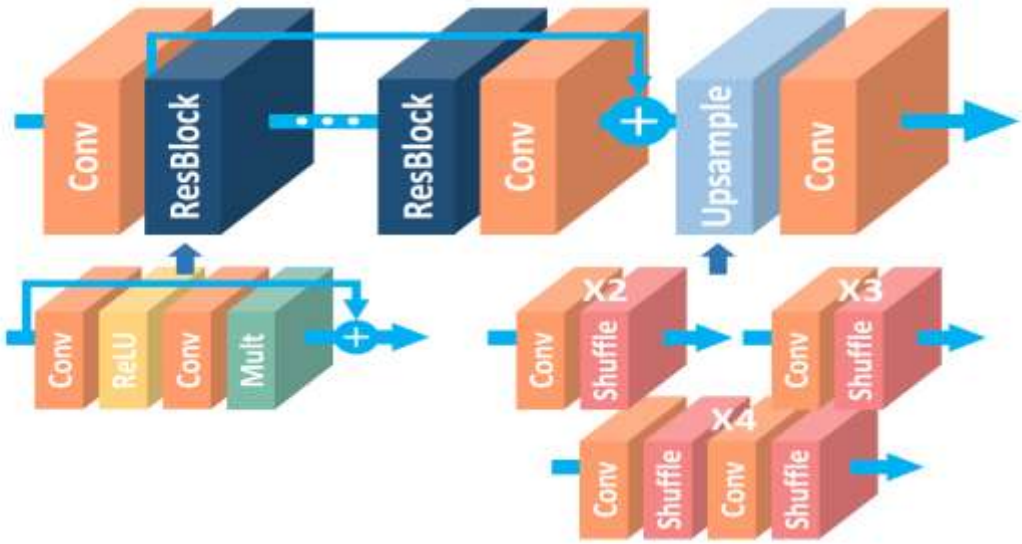


Figure II.15: The EDSR model architecture [62]

a) Residual blocks of EDSR Model:

The figure below compares the residual block architectures used within three different deep neural network models: the ResNet (a), SRResNet (b), and the proposed EDSR model (c). Each model illustrates a gradual reduction in the architecture to achieve better efficiency and performance.

In figure II.20 (a), the fundamental ResNet block has two convolution layers each followed by batch normalization (BN) and in between a ReLU activation. In figure II.20 (b), SRResNet has the same structure except it drops the final ReLU after the residual addition. In figure II.20 (c), the proposed model used in EDSR, goes one step further by removing all batch normalization layers and shrinking the block to just two convolutional layers with one ReLU in between [69].

As we can see from figure II.16, batch normalization is removed [70]:

- Since batch normalization layers normalize the features, they get rid of range flexibility from networks by normalizing the features, it is better to remove them.
- GPU memory usage is also sufficiently reduced.
- The proposed baseline model without batch normalization layer saves approximately 40% of memory usage during training, compared to SRResNet.

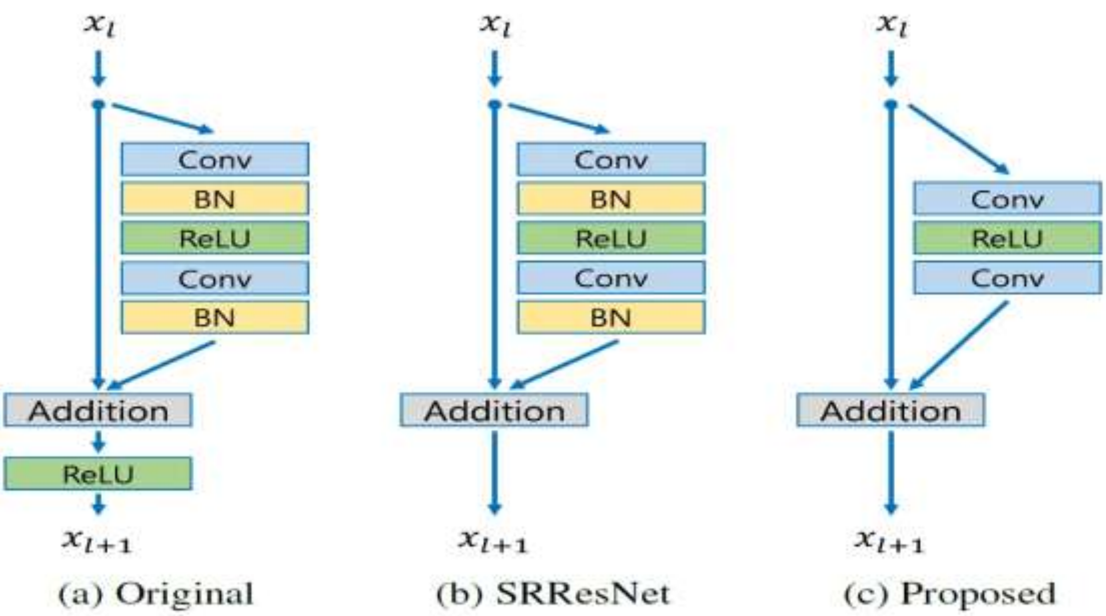


Figure II.16: Comparison of residual blocks in: (a) original ResNet, (b) SRResNet, (c) EDSR [62].

b) UpSampling block of EDSR Model:

The EDSR architecture differs from SRResNet in the remaining blocks after the elimination of the batch normalization layers BN, but otherwise similar architecture in the upsampling blocks. The Upsampling Block of the EDSR model is the block responsible for rescaling low-resolution feature maps to high-resolution images. This is done by : Conv2D → Pixel Shuffle.

This architecture utilizes Sub-pixel Convolution (Pixel Shuffle) instead of traditional upsampling methods like Transposed Convolution and bicubic interpolation to avoid artifacts [62].

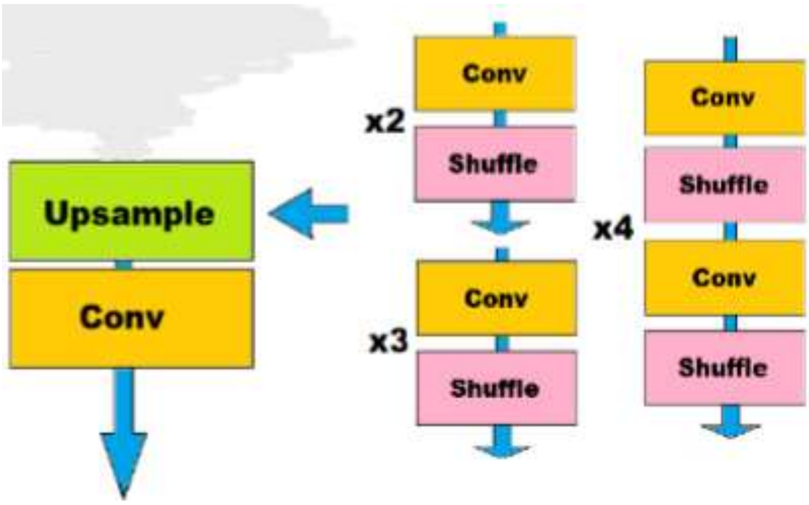


Figure II.17: Upsampling block EDSR [71].

c) Pixel Shuffle layer (sub-pixel convolution):

Pixel Shuffle (or Sub-pixel Convolution) is a neural network layer that increases the spatial resolution of an image by reordering data from channels to spatial dimensions (Height × Width). After the ResBlocks, EDSR use Pixel Shuffle upsampling to increase the resolution at the end of the network, which avoids the usage of bicubic interpolation prior to the network and its disadvantages like Loss of Fine Details and Checkerboard Artifacts and Slow training.

Pixel Shuffle layer consists of two main computations, as shown in the figure II.21.

1) A convolution layer to increase the number of filters to the size required by the next step.
2) A shuffle operation that rearranges the activations in the feature maps to obtain shallower but larger feature maps, with the same number of activations.

Pixel Shuffle layer is applied once for x2 super-resolution and twice for x4 super-resolution [72]. Pixel Shuffle Rearranges elements in a tensor of shape $(*, C\times r^2, H, W)$ to a tensor of shape $(*, C, H\times r, W\times r)$ where r is an upscale factor. This process redistributes the depth (channel) information into the spatial dimensions (height and width), effectively increasing the resolution of the feature map [73]. As shown in Figure II.22, we have a feature map of H×W shape with 4C channels, we flatten the 4 pixels of each position to 2×2 shape, and finally get a feature map of 2H×2W shape with C channel. This upsampling method has no learnable weights, but the high-resolution features after upsampling are completely obtained from the low-resolution features, without the need for the network to derive by itself [74].

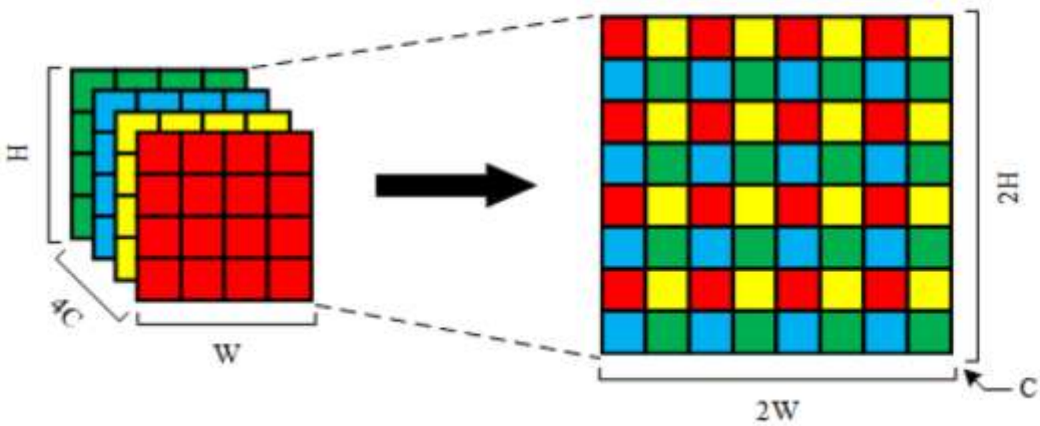


Figure II.18: Schematic diagram of the pixel shuffle [74].

II.8. Enhancing EDSR Model Performance:

II.8.1 Optimization Method:

Optimization algorithms are very important while training any deep learning models, by modifying the model parameters to minimize the loss function, there are many optimization algorithms, including like Gradient Descent (GD), SGD and Momentum, RMSProp, Adam, Nesterov Accelerated Gradient (NAG) In our model we used

II.8.1.1. Adam Optimizer:

Adam optimizer (adaptive moment estimation) is one of the most popular deep network optimization algorithms. It was first introduced in 2015, and has since become widely adopted because of its fast convergence, robustness, and computational efficiency [75]. Adam optimizer combines the strengths of two other well-known algorithms (Momentum and RMSprop) to deliver a powerful method for adjusting the learning rates of parameters during training.

Adam is highly effective, especially when working with large datasets and complex models, because it is memory-efficient and adapts the learning rate dynamically for each parameter [76]. Adam maintains two moving averages m_t et v_t for each parameter:

- **m_t**: the exponentially decaying average of past gradients (first moment estimate, like momentum)

- **v_t:** the exponentially decaying average of past squared gradients (second moment estimate, like RMSProp)

The equations of two moving averages in Eq. II.1 and Eq. II.2.

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \qquad \text{.................} \text{ (II.1)}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2 \qquad \text{.................} \text{ (II.2)}$$

Where: g_t: is the gradient at time step t. β_1: (commonly 0.9) controls the decay rate for the first moment. β_2: (commonly 0.999) controls the decay rate for the second moment

II.8.2 loss function:

The loss function is a key component in training deep learning models, as it determines how well the model performs at each training step. In high-resolution tasks, the loss function aims to measure the difference between the image generated by the model and the (original high-resolution ground truth image). By gradually reducing this difference during training, the model's performance improves, resulting in higher-quality images with more accurate details.

In the EDSR model, the researchers used L1 Loss instead of L2 Loss, because using L1 Loss produces better visual results compared to L2 Loss. The main reason is that L1 focuses on enhancing fine details and visual texture, while L2 tends to reduce overall numerical error at the expense of some loss of sharpness [77].

II.8.2.1. Mean Absolute Error (MAE) / L1 Loss:

Calculates the average absolute value of the difference between each pixel in the predicted image and the original High-Resolution Ground Truth Image. The mathematical equation for Mean Absolute Error (MAE) or L1 Loss is:

$$L1 = \frac{1}{N} \sum_{i=0}^{N} |Yt - Yp| \qquad \text{.......................................} \text{(II.3)}$$

Where: Yt: The true pixel value in the original (HR) ground truth image. Yp: Expected pixel value in the predicted image. N: Number of pixels

II.8.2.2. Mean Square Error (MSE) / L2 Loss:

Calculates the mean squared difference between each pixel in the predicted image and the original High-Resolution Ground Truth Image. The mathematical equation for Mean Square Error (MSE) or L2 Loss is:

$$L2 = \frac{1}{N}\sum_{i=1}^{N}(Yt - Yp)^2 \qquad \text{(II.4)}$$

Where: Yt: The true pixel value in the original (HR) ground truth image. Yp: Expected pixel value in the predicted image. N: Number of pixels

II.8.2.3 Importance of Minimizing the Loss Function for Model Performance:

In general, the loss function is best when it is small. This is because the loss function measures the error or difference between the model's output and the correct prediction (Ground Truth), The smaller this error, the more accurate the model's prediction. Because the goal of training any model is to minimize errors so that the model can predict or classify more accurately.

If the loss is large, it means that the model:

- Has not learned the correct patterns from the data.
- It produces weak or random results.

If the loss is small, it means that the model:

- Has become able to understand the relationship between the inputs and outputs.
- It produces results that are close to the truth.

II.9. conclusion:

This chapter addressed the fundamental theoretical and technical foundations related to machine learning and deep learning, by presenting a comprehensive and systematic overview of the evolution of concepts and techniques used in image processing and resolution enhancement. The focus was placed particularly on deep learning, as it represents the most prominent approach for achieving accurate and efficient results in image-related tasks, especially in super-resolution. In the theoretical context, a large part of the chapter is devoted to understanding how the simple CNN model evolved into the EDSR model. Both ResNet and SRResNet were presented, followed by an in-depth exploration of the EDSR model, which is considered one of the most efficient and accurate models in this domain. The chapter detailed its internal architecture, residual blocks, upsampling mechanism, and the Pixel Shuffle technique used to reconstruct high-resolution images. Furthermore, the methods used to enhance the performance of the EDSR model were discussed, particularly optimization algorithms such as Adam, and loss functions, with a specific emphasis on L1 Loss, which has proven superior in preserving visual details compared to L2 Loss. The importance of minimizing the loss function was also clarified, as it plays a crucial role in achieving higher accuracy and model stability during training.

Overall, this chapter provides the essential scientific foundation for understanding deep models applied to super-resolution tasks and sets the stage for the third chapter, which will focus on the practical and experimental aspects of the EDSR model.

Chapter III: Experimental Setup and Performance Evaluation

III.1 Introduction:

This chapter focuses on the development tools and software used for model development. For the purpose of knowing the technical background and resources utilized for model building and testing, it is necessary to be well aware of all the instruments and software utilized. Therefore, it is necessary to introduce the tools utilized, prove their relevance, and mention the outcomes obtained through them. Performance metrics were used to evaluate the quality of the reconstructed images using our model, including PSNR and SSIM. The results are discussed in detail, highlighting the significant improvement of the model. We present and discuss qualitative results that showing the superior quality of our generated high-resolution outputs, with direct comparisons of low-resolution inputs, generated high-resolution outputs.

III.2 Used dataset:

In this work, the DIV2K (short for DIVerse 2K) dataset was used to train and evaluate the performance of the EDSR model. DIV2K is one of the most popular high-quality datasets in the field of image super-resolution and was provided by the Computer Vision Laboratory at ETH Zurich as part of the NTIRE Challenge.

This dataset contains [78]:

- 800 high-resolution (HR) training images, representing a wide range of scenes and objects rich in detail.
- 100 validation images.
- 100 test images (without baseline data for use in benchmark evaluations). Each image in the DIV2K dataset has a resolution close to 2K (approximately 2040 x 1080 pixels), providing a variety of textures and structures within the images.

Figure III.1: Div2k images dataset

III.3 Development tools:

III.3.1 Hardware used:

This work was done on my PC which has these features:

- CPU: I5 10400f
- RAM : 16gb
- GPU : RTX 3050 8gb VRAM
- ROM : 1tb SSD (Nvme)

III.3.2 Software:

Here we will talk and explain the tools and software that we used to write and run the code, as well as the development environment used and other auxiliary software.

III.3.2.1 Operating System:

Windows 11 (64-bit) is a good choice for AI and image processing applications. Because of its broad support for programming libraries like Python, PyTorch, and CUDA, which rely on GPU processing, and its good compatibility with development environments like Visual Studio Code.

III.3.2.2 Visual Studio Code (VS Code):

Visual Studio Code, or VS Code, is a lightweight, open-source code editor from Microsoft that has become a popular tool used by developers at all levels across a wide range of programming languages. It's used for developing Python applications and deep learning applications. We use it for its ease of use, flexibility, enhanced features, advanced capabilities for efficiently tracing and executing code, and its user-friendly interface.

Figure III.2: Logo of Visual Studio Code

III.3.2.3 Python:

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991, It is used for web development (server-side), software development, mathematics, system scripting. Python is one of the most widely used and popular programming languages in the world, outperforming C, C++, Java, and JavaScript in many areas [79].

Figure III.3: Python logo

The Advantages of Python in deep learning Environments are:

- **Readability:** The code written in Python is simple and clear, making it easy to understand and write even for non-experts.
- **Extensive Libraries and Frameworks:** Python supports many important libraries, such as: TensorFlow, PyTorch, Keras, NumPy, Pandas, and Scikit-learn.
- **Compatibility with Other Languages:** Python can interact with languages like C/C++, Java, and others.
- **Strong Community Support:** Python has a large community of developers and researchers.
- **Scalability and Performance:** Python supports working with big data and processing models on multiple devices or with GPUs.

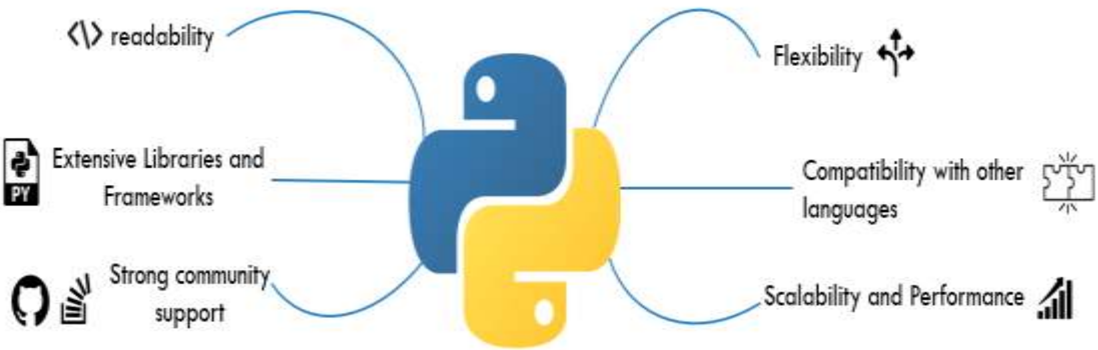


Figure III.4: Advantages of Python

III.3.2.4 CUDA Toolkit and cuDNN:

a) CUDA Toolkit:

Compute Unified Device Architecture, a.k.a. CUDA is a parallel computing platform developed by NVIDIA with an initial release date of 23 June 2007. It allows developers to use GPUs' power for general-purpose tasks, not just graphics rendering. NVIDIA CUDA made it possible to use GPUs for various applications, including scientific research, engineering simulations, and, eventually, AI and deep learning. By around 2015, the development of CUDA's focus shifted towards neural networks and AI. We used CUDA to speed up the training process of the EDSR model using a graphics processing unit (GPU). CUDA allowed us to perform calculations faster on special NVIDIA hardware, which contributed to reducing the training time and improving the overall performance of the model. We used PyTorch because it is a library that supports CUDA to achieve the greatest benefit and speed up the training time of our model, as training large models on the CPU consumes a large amount of time, which may waste days and days of waiting [80].

b) cuDNN:

cuDNN (CUDA Deep Neural Network library) is a specialized, GPU-accelerated library that provides essential building blocks for deep neural networks. It's designed to deliver high-performance components for convolutional neural networks and other complex deep learning algorithms to speed up the execution of repetitive mathematical operations like Convolution, Pooling Normalization. By implementing cuDNN, frameworks such as TensorFlow and PyTorch can take advantage of optimized GPU performance [81].

NVIDIA's CUDA installation lays the groundwork for GPU computing, whereas cuDNN provides targeted resources for deep learning. This combination enables remarkable GPU acceleration for tasks that a traditional CPU could otherwise require days or weeks to complete.

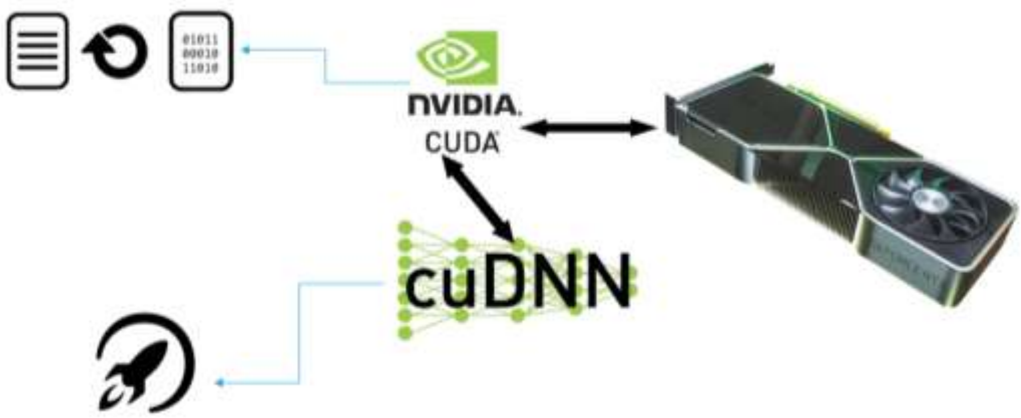


Figure III.5: Accelerating Neural Networks Using CUDA and cuDNN on NVIDIA GPUs

c) System Requirements for installation CUDA and cuDNN:

Before you start the NVIDIA CUDA installation or cuDNN installation steps, your system fulfils the following requirements:

- **Hardware Requirements:**

– NVIDIA graphics card (GPU) The card must be CUDA supported. While most recent

– NVIDIA GPUs support CUDA. Examples: RTX 30xx, RTX 20xx, GTX 16xx, GTX 10xx, Tesla, Quadro

– At least 8GB of RAM (16GB or more recommended).

– Setting up CUDA, cuDNN, and the necessary drivers may require several gigabytes of storage. You must have a minimum of 5–10 GB of free disk space available.

- **Software Requirements:**

– Windows 10/11 The system must be (64-bit) or Linux (Ubuntu ،CentOS ،RHEL)

– NVIDIA Driver: It must match the CUDA version you want to install.

– CUDA Toolkit Release: Choose the version that suits your graphics card and system.

– cuDNN: must be compatible with the installed CUDA version.

– Visual Studio (Windows only): Visual Studio 2019 and 2022 are not supported. Install "Desktop development with C++" during installation.

III.3.2.5. Installing CUDA and cuDNN on Windows :

This section provides a detailed guide on installing CUDA and cuDNN on a Windows system.

- **Step 1: Verify GPU Compatibility:**

To determine your GPU model and check if it is compatible with CUDA, right-click on the Start Menu, choose Device Manager, and then expand the Display Adapters section to locate your NVIDIA GPU. After finding it, head over to the NVIDIA CUDA-Enabled GPU List to verify whether the specific GPU model supports CUDA for GPU acceleration.

- **Step 2: Install NVIDIA GPU Drivers:**

if u don't have a NVIDIA drivers go to download and set up the latest NVIDIA drivers, go to the official NVIDIA website and download it from there. ' https://www.nvidia.com/en-us/drivers/ ' and choose the correct driver for your GPU and Windows version.

In my case :RTX 3050 8G VRAM.

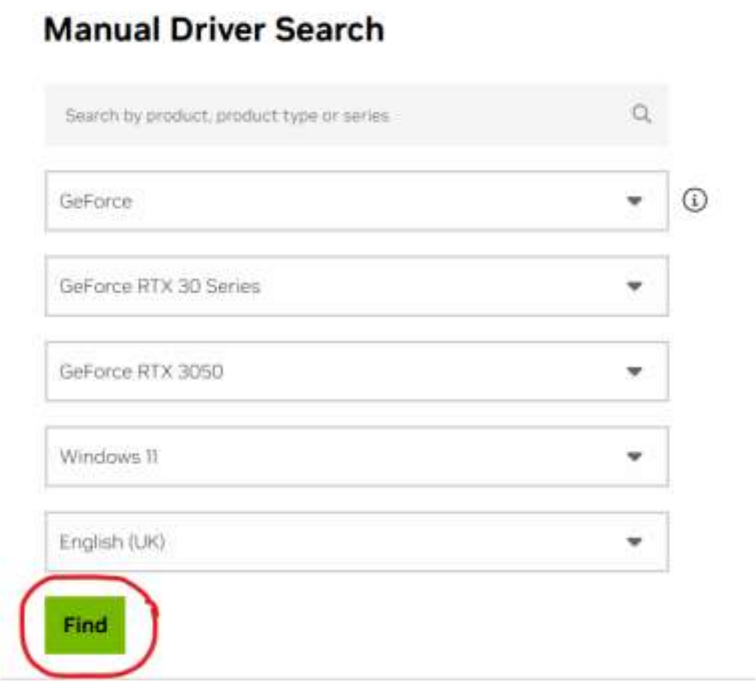


Figure III.6: Manual driver search

- Click on FIND and you will get the download interface for the version.

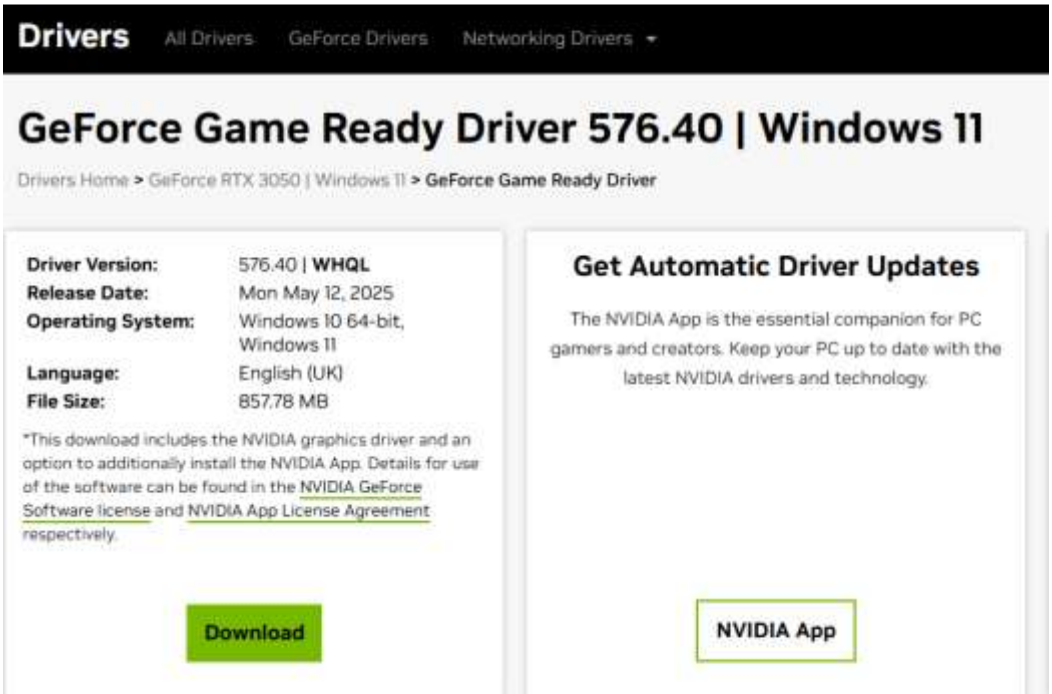


Figure III.7: Choosing and download the driver

- Download the driver.
- Follow the installation instructions.
- Reboot your computer.
- Open CMD and run 'nvidia-smi' command:

The command will give you information and tables about the version you installed like this :

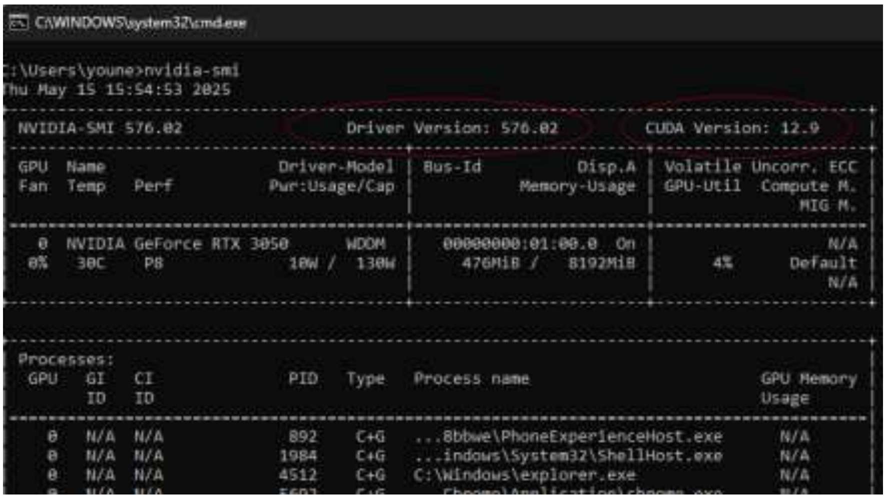


Figure III.8: open CMD interface

It means that the version NVIDIA Driver was installed successfully.

<u>**Note:**</u> CUDA version number in the table represents the latest CUDA Toolkit version your current NVIDIA Driver supports. It does not represent your currently installed CUDA Toolkit version, or even if you have it installed.

- **Step 3: Installing CUDA Toolkit:**

– Open following link in your browser: ' https://developer.nvidia.com/cuda-toolkit-archive '

Figure III.9: Choosing the version of CUDA

– Choose the version number you want to download from CUDA. It is preferable to install older versions because they are more stable and without problems. In my case I chose <u>11.8.0</u>

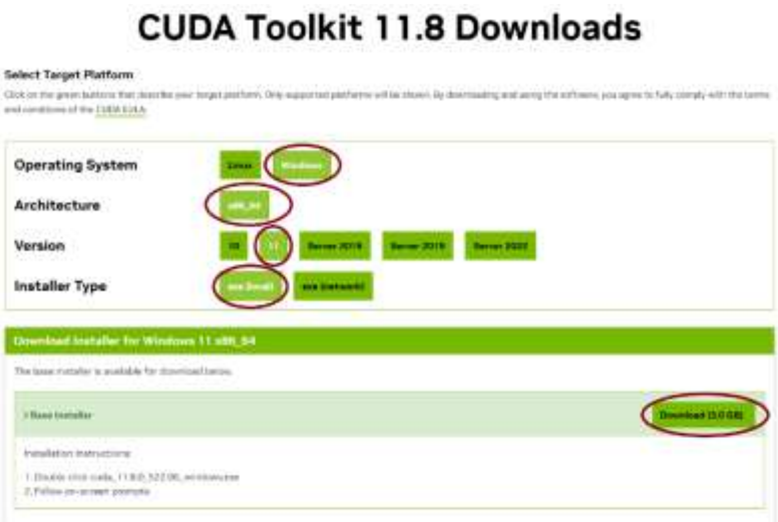


Figure III.10: Choosing configuration of your PC

– After downloading the version of CUDA, you must install it on the PC. These are simple steps, just follow the instructions.

Figure III.11: Installation CUDA version

- **Step 4: Installing cuDNN library:**

Open following link in your browser:' https://developer.nvidia.com/rdp/cudnn-archive '

It's important to ensure the cuDNN version aligns with your installed CUDA version. Download the latest version of cuDNN compatible with the CUDA you downloaded 12.x or 11. x. in my case I chose cuDNN version v8.9.7. that compatible with CUDA 11.8.

– Click to download A compressed file will be downloaded. Extract the files.

After downloading cuDNN from NVIDIA's website, you need to copy its files to the CUDA folder you have on your device, so that PyTorch or TensorFlow can use it.

- Extract the cuDNN file you downloaded (usually in ZIP format).
- Contains 3 main folders: bin / include / lib
- Go to the CUDA installation folder: Copy the contents of the folders from cuDNN to the CUDA folder

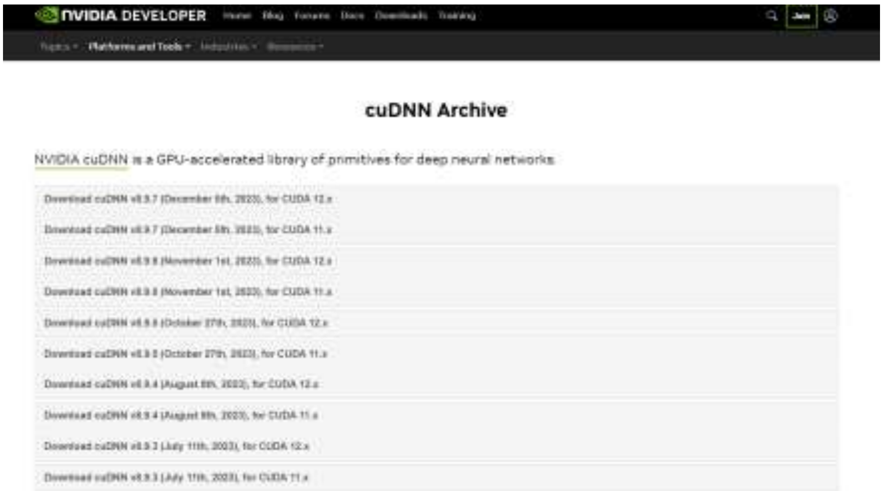
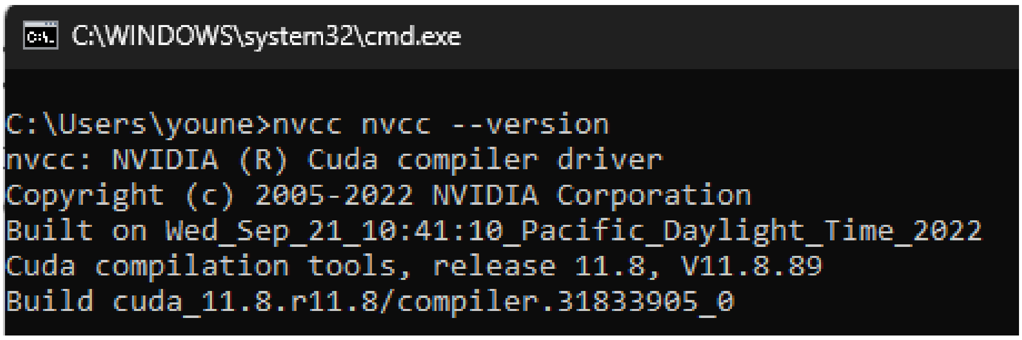


Figure III.12: Find a CUDA-compatible version of cuDnn.

Figure III.13: Choosing local installers

- **Step 5: Check installation:**

- Open CMD and type: ' nvcc –version '



- Now it means CUDA and cuDNN are installed correctly.

III.3.3 Libraries Used:

In this project, we used a lot of libraries with Python, such as numpy, skimage, and matplotlib, as well as the OpenCV 4.1.0 library for image processing, and the PyTorch 2.2 library to build and train our deep learning model, but we will talk about these two libraries for a reason, as they form the core of the project.

III.3.3.1 PyTorch :

PyTorch is an open-source machine learning (ML) framework based on the Python programming language and the Torch library. Torch is an open-source ML library used for creating deep neural networks and is written in the Lua scripting language. It's one of the preferred platforms for deep learning research [82]. Pytorch is the most optimized deep-learning tensor library mostly used for applications that use GPUs and CPUs. It is often preferred over its alternatives like TensorFlow and Keras because it is completely Pythonic and uses dynamic computation graphs.

Figure III.14: Logo of PyTorch

III.3.3.2 OpenCV:

OpenCV (Open Computer Vision Library) is an open-source software library for computer vision and machine learning. It is designed to provide a common infrastructure for computer vision applications and accelerate the use of machine perception in commercial products. The OpenCV library is used in the EDSR model for reading, transforming, displaying, and saving images. It also facilitates interaction with images before and after deep processing [83].

Figure III.15: Logo of OpenCV

III.4. Evaluation metrics:

Evaluating the performance of SR algorithms is essential for identifying their effectiveness and potential improvements. The PSNR and SSIM measure is used as one of the basic indicators to measure how close the resulting image is to the original reference image, where higher values reflect better quality.

III.4.1 Peak Signal-to-Noise Ratio (PSNR):

PSNR (Peak Signal-to-Noise Ratio) is one of the most popular numerical metrics used to evaluate image and video quality, especially in Super-Resolution tasks that aim to improve image resolution. by comparing the maximum possible signal strength to the noise that affects its fidelity. A higher PSNR value indicates better image quality, as it means that the differences between the original and the processed image are minimal, it is expressed in decibels (dB) [84].

The mathematical representation of the **PSNR** is as follows :

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{\mathrm{MAX_f}}{\sqrt{\mathrm{MSE}}}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots(\text{III.1})$$

Where $\mathrm{MAX_f}$: The largest possible pixel value in an image. If the image is 8-bit, the value is 255. If it is 16-bit, the value is 65535, and **MSE** (Mean Squared Error) is :

$$\mathrm{MSE} = \frac{1}{\mathrm{mn}}\sum_0^{m-1}\sum_0^{n-1}[f(i,j) - g(i,j)]^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(\text{III.2})$$

Where $f(i,j)$: The pixel value at position *i, j* in the original image (Ground Truth). $g(i,j)$: The pixel value at position *i, j* in the output image of the model. *m*: Number of pixels rows in the image, image height. *n*: The number of columns (pixels) in the image, the width of the image [85].

III.4.2 Structural Similarity Index (SSIM):

The SSIM index is a common metric used to evaluate image quality from the perspective of human visual perception, and aims to measure the structural similarity between two images, such as the original image and enhanced image. SSIM is as an alternative to traditional metrics such as Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR), which are based on pixel-wise differences between images and may not always correlate well with human visual perception of image quality [86].

The SSIM algorithm compares two images by measuring their luminance, contrast, and structural information. Specifically, it calculates three terms:

1. Luminance term: Measures the difference in brightness between the two images.
2. Contrast term: Measures the difference in contrast between the two images.
3. Structure term: Measures the correlation between the two images' structures.

The mathematical representation of the **SSIM** is as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (\text{III.3})$$

Where μ_x and μ_y : are the average luminance of the two images being compared. σx^2 and y^2 : are the variances of the pixel intensities in the two images. σ_{xy} : is the covariance of the pixel intensities between the two images. C1 and C2: are constants added to prevent instability when the denominators are close to zero [87].

The index produces a value between -1 and 1, where 1 indicates perfect similarity, 0 indicates no similarity, and negative values suggest dissimilarity.

- SSIM = 1 The images match perfectly.
- SSIM ≈ 0 There is no noticeable similarity.
- SSIM < 0 The images are very different.

III.5 Model Training and Optimization:

III.5.1 EDSR model Training:

In this section we detail the procedures of how we trained the EDSR model and what are the most important points and parameters [62].

- **Dataset and Hardware:**

The model was trained using training dataset of 800 high-resolution images from the DIV2K collection we talked about at the beginning of the chapter and using an RTX 3050 graphics card with (8GB Vram) to speed up the training process.

- **Input Preparation:**

At the beginning of the model training process, small squares (patches) taken from low-resolution images (LR) (96 x 96 pixels) are matched with corresponding squares from high-resolution images (HR).

– **Data Augmentation:**

To increase the size and diversity of the training data, we apply operations such as (data augmentation) through random horizontal flips and 90-degree image rotation for both low-resolution and high-resolution images. This process gives the model a broader view of the complex and different patterns within images and detects many angles and details.

– **Optimization Algorithm:**

We trained our model using the Adam optimizer, one of the most popular deep network optimization algorithms, which combines speed and stability. Its parameters were set to β1=0.9, β2=0.999, and ε=10^-8 (a small stability parameter to avoid division by zero).

– **Training Settings:**

We set the mini batch size to 16. The learning rate was initialized to 10^-4, and divided in half every two minibatch updates, to ensure continuous improvement

– **Loss Function:**

The L1 Loss function was used instead of the L2 Loss. The L1 Loss provides faster convergence and better results, which made it the preferred choice in our project.

– **Training Time:**

Training the EDSR model on x2 took about 8 hours, and 20 hours on x4 where we used 300 epochs.

All of the above is summarized in this table:

Table III.1 : Training the EDSR model

Parameter	**Value**
Dataset	DIV2K (800 HR images for training)
Optimizer	Adam (β1=0.9, β2=0.999, ε=10^-8)
Loss Function	L1 Loss
Batch Size	16
Initial Learning Rate	10^{-4}
Scaling Factors	2x, 4x
Training Time (2× scale)	Approximately 8 hours
Training Time (4× scale)	Approximately 20 hours

III.5.2 Loss function L1 :

Through our experience and training the EDSR model , we obtained the Curve below shows the L1 Loss progression over 300 epochs during the training of this model on the DIV2K dataset.

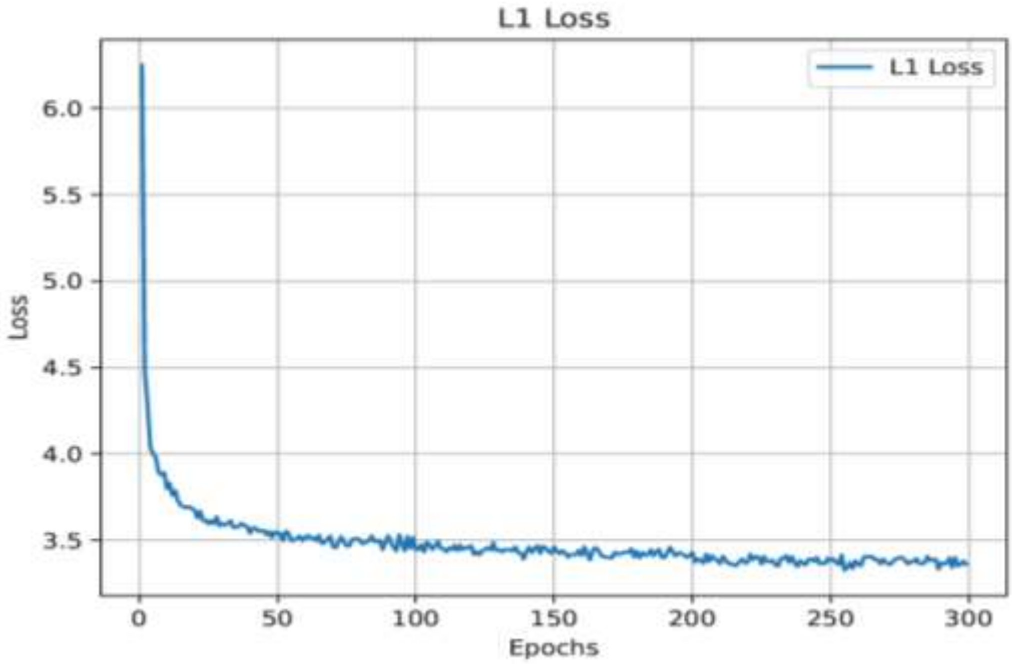


Figure III.16**:** L1 Loss Curve During EDSR Model Training

From the loss curve, we notice that the loss value starts at a very high level (~6.4) at the beginning of training, then drops rapidly over the first 50 training cycles or so, reflecting that the model is initially learning rapidly. After that, the loss value gradually begins to stabilize around a value of approximately 2.8, with small fluctuations, indicating that the model has approached a steady state. The model was able to significantly reduce the error in the early stages, after which it stabilized at a relatively low value, which reflects that the training is going well.

III.5.3 Evolution EDSR model using PSNR:

In this section, we evaluate the performance of the EDSR model to measure the quality of the resulting super-resolution images using PSNR on DIV2K Dataset for ×2 and ×4 Upscaling. From the results we obtained during training, we arrived at the PSNR score. The figure below shows curves of the evolution of the PSNR on DIV2K for 2x and 4x Scale index over 300 training (epochs). The horizontal axis represents the number of epochs , while the vertical axis expresses the PSNR value, a measure of the quality of the restored images compared to the original images.

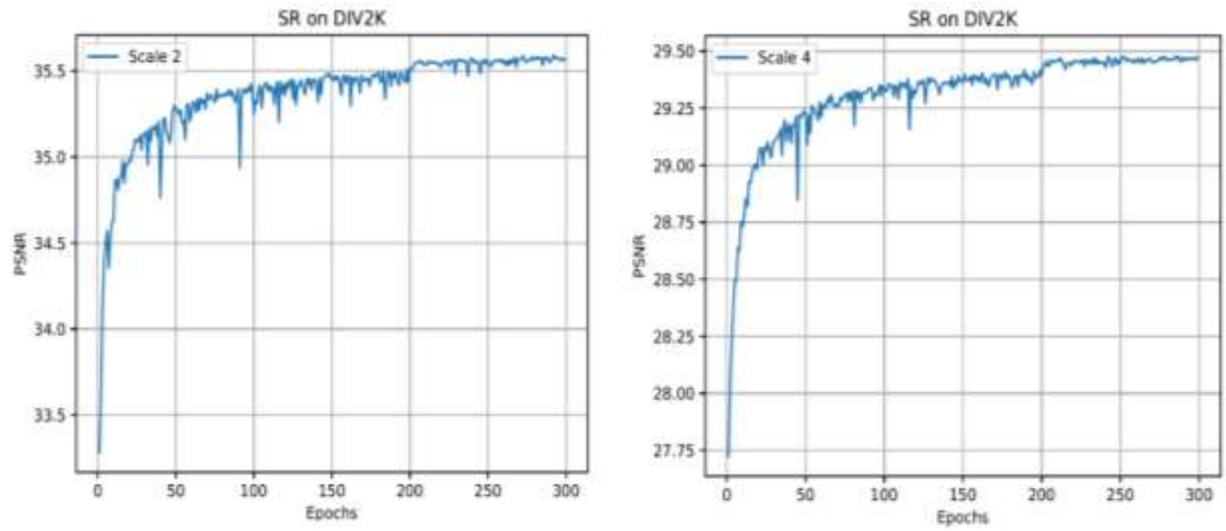


Figure III.17**:** PSNR Evolution on DIV2K for ×2 and x4 Scale during Training

The curves 2x and 4x presented clearly illustrate a gradual improvement in the performance of the EDSR model throughout the training process. In both curves, the PSNR value starts relatively low during the early training epochs and increases steadily as the number of epochs progresses. This upward trend reflects the model's effective learning process, where the loss is reduced progressively, leading to enhanced image reconstruction quality over time.

A notable observation is the superior performance achieved at the 2× scaling factor compared to the 4× scale. In the left curve (Scale 2×), the PSNR reaches approximately 35.6 dB by the end of training, indicating a high-quality reconstruction of the original high-resolution image. In contrast, the right curve (Scale 4×) shows a maximum PSNR of around 29.5 dB, which is significantly lower. This highlights the challenge in recovering fine details when dealing with higher scaling factors, as more structural and textural information is lost in the low-resolution inputs.

Additionally, both curve s exhibits a stabilization in PSNR values after approximately 150 epochs. This relative consistency suggests that the model has reached a convergence phase, where further performance improvements become more gradual and less significant. The model's parameters appear to have settled into an optimal configuration under the current training conditions.

Finally, the performance gap between the two scales underscores the technical challenges associated with larger upscaling factors. A 4× enlargement leads to greater information loss in the input image, making it harder for the model to reconstruct fine details accurately. In contrast, a 2× scale preserves more structural information, allowing for a more faithful and detailed reconstruction by the model.

III.6 Results and discussions:

In this section, we will present the results of the EDSR model we trained to improve image resolution. We will evaluate the model's performance using the quantitative metrics we discussed earlier, PSNR and SSIM, as well as a visual comparison between the enhanced images (SR), the low-resolution images (LR), and the original high-resolution images (Ground Truth). We will also provide a quantitative comparison of the results. The evaluation is performed on the DIV2K validation set images, using PSNR and SSIM, metrics as well as on the set5 and set14 datasets.

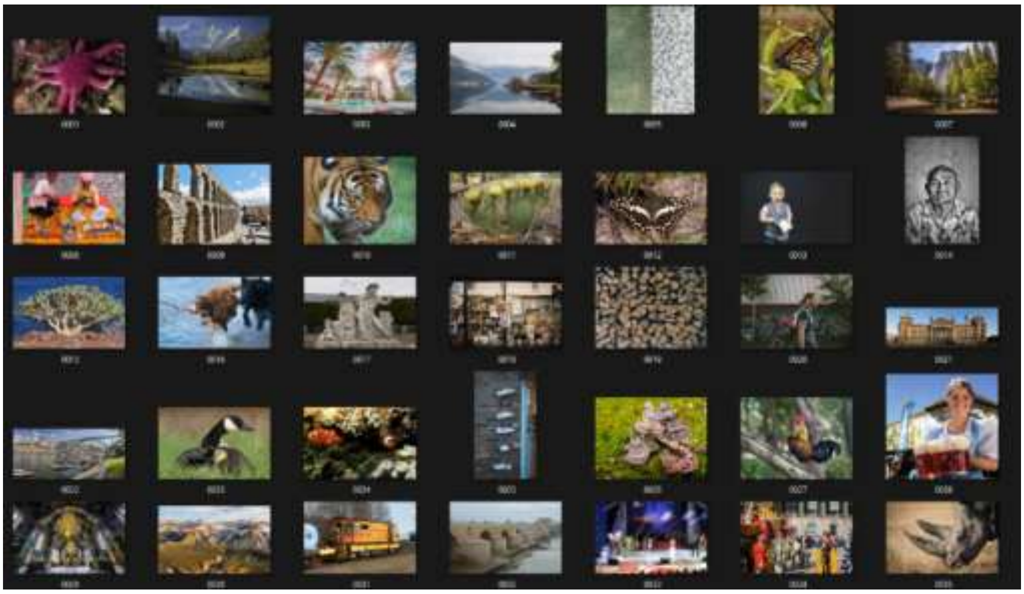


Figure III.18**:** DIV2K images test

Figure III.19**:** Set14 dataset for test

Figure III.20**:** Set5 dataset for test

The table III.2 shows the number of test images in each dataset used in the model testing process. We will select two images from set5 and set13 dataset, and 3 images from DIV2K dataset.

Table III.2 : showing the images used in the model test

Dataset	Amount	Format
Div2k validation	100	PNG
Set 5	5	PNG
Set 14	14	PNG

III.6.1 Quantitative and Qualitative Results:

We conducted a practical evaluation of the EDSR model using low-resolution input images generated by downscaling high-quality original images with the Bicubic algorithm at two reduction scales: 2× and 4×. For each image in the dataset, two downscaled versions were prepared one at 2× and the other at 4×while preserving the original high-resolution version for comparison. The low-resolution images were then fed into the model, which regenerated the high-resolution version of them. The results were compared visually with the original images, and quantitatively using metrics PSNR and SSIM to assess reconstruction accuracy and detail quality. This evaluation allowed us to understand the effectiveness of the model in enhancing images.

III.6.1.1 Qualitative Results:

Figures III.13-19 show the visual comparison between the original images, the 2x and 4x low-resolution images, and the resulting images for each test image.

In all of these figures, in the first row, we see original high-resolution Ground Truth images, with lower-resolution versions obtained by 2x and 4x downscaling using the bicubic algorithm. We clearly see that these lower-resolution images suffer a tremendous detail loss, especially in areas of fine texture such as feathers, leaves, or highly textured surfaces. This is very much strengthened by downscaling to 4x, where the details are less sharp and the image appears grainy and artificial in some areas.

In the second row, we show the results of reconstructing images at higher resolution using the EDSR model. As can be seen, there is a drastic improvement in the quality of the reconstructed images over low-resolution images, especially at 2x magnification, where many previously lost details are recovered. At 4x, and with the added challenge, the model is still able to reconstruct many visual features successfully, and in some cases, acceptably.

In the bottom row, selected areas of the images are magnified to show finer detail in the model's work. From these close-up areas, it can be seen that EDSR's ability to recreate accurately details like fine edges, which were almost lost in lower-resolution images. Quantitative measures such as PSNR and SSIM (measuring image quality as compared to the original image) confirm the model's high performance since these are indeed measures of visual quality enhancement and detail restoration.

Image 0853 from Div2k HR (Ground Truth) 2040 x1356 3.52 MB

LR (downscaled by bicubic 2x) 1020 x 678 1 MB

LR (downscaled by bicubic 4x) 510 x 339 268.07 KB

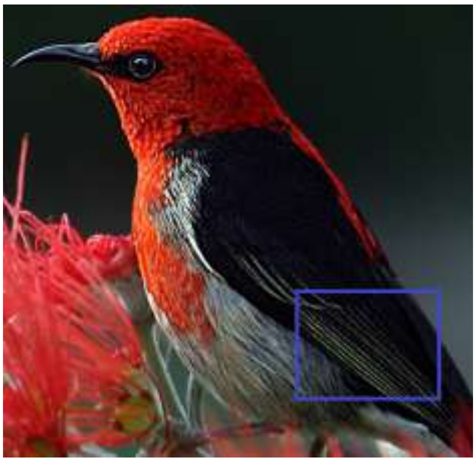


Image HR generated using EDSR with 2x upscaling from a image LR (downscaled by bicubic 2x) 2040 x 1356 3.14 MB

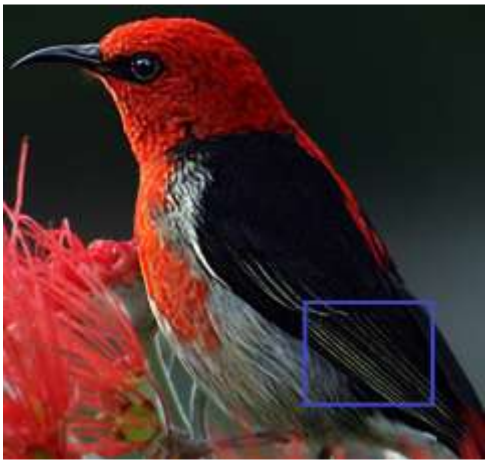


Image HR generated using EDSR with 4x upscaling from a image LR (downscaled by bicubic 4x) 1020 x 678 891.19 KB

HR 2x Resolution (41.122 dB / 0.9796)

HR 4x resolution (34.001 dB / 0.9341)

Figure III.21: Super-resolution results for Image 0853 from DIV2K dataset

Figure III.22: Super-resolution results for Image 0013 from DIV2K dataset

Image 0309 from Div2k HR (Ground Truth)*2040 x1356   4.20 MB*

LR (downscaled by bicubic 2x)
1020 x 678   1.13 MB

LR (downscaled by bicubic 4x)
510 x 339   328 KB

Image HR generated using EDSR with 2x upscaling from a image LR (downscaled by bicubic 2x)
2040 x 1356   3.59 MB

Image HR generated using EDSR with 4x upscaling from a image LR (downscaled by bicubic 4x)
2040 x 1356   3.20 KB

HR with EDSR 2x upscaling
(37.188 dB / 0.9753)

HR with EDSR 4x upscaling
(29.403 dB / 0.9157)

Figure III.23: Super-resolution results for Image 0309 from DIV2K dataset

flowers from Set14 HR (Ground Truth) 500 x 362   341 KB

LR (downscaled by bicubic 2x) 250 x 182   90.6 KB

LR (downscaled by bicubic 4x) 125 x 90   24.4 KB

Image HR generated using EDSR with 2x upscaling from a image LR (downscaled by bicubic 2x) 500 x 362   298 KB

Image HR generated using EDSR with 4x upscaling from a image LR (downscaled by bicubic 4x) 500 x 360   249 KB

HR with EDSR 2x upscaling (32.433 dB / 0.9365)

HR with EDSR 4x upscaling (23.117 dB / 0.7497)

Figure III.24: Super-resolution results for Image flowers from Set14 dataset

baboon from Set14 HR (Ground Truth)
500 x 480 518 KB

LR (downscaled by bicubic 2x)
250 x 240 134 KB

LR (downscaled by bicubic 4x)
125 x 120 33.3 KB

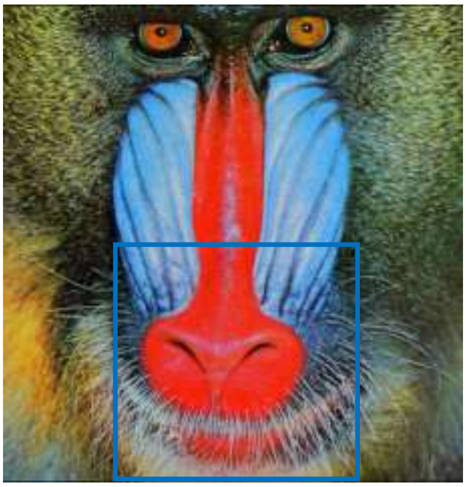


Image HR generated using EDSR with 2x upscaling from a image LR (downscaled by bicubic 2x)
500 x 480 479.9 KB

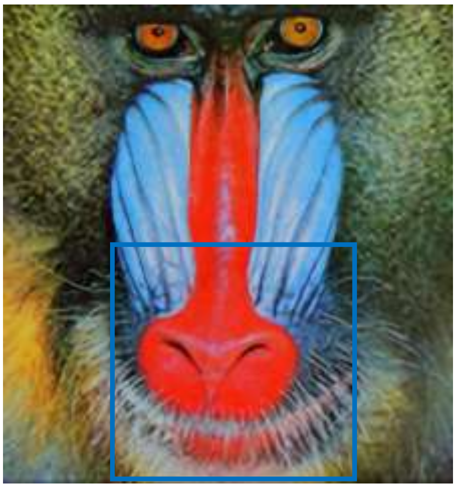


Image HR generated using EDSR with 4x upscaling from a image LR (downscaled by bicubic 4x)
500 x 480 356 KB

HR with EDSR 2x upscaling
(24.010 dB/ 0.7615)

HR with EDSR 4x upscaling
(20.946 dB/ 0.5056)

Figure III.25: Super-resolution results for Image baboon from Set14 dataset.

baby from Set5 HR (Ground Truth)
512 x 512   362 KB

LR (downscaled by bicubic 2x)
256 x 256   106 KB

LR (downscaled by bicubic 4x)
128 x 128   30.4 KB

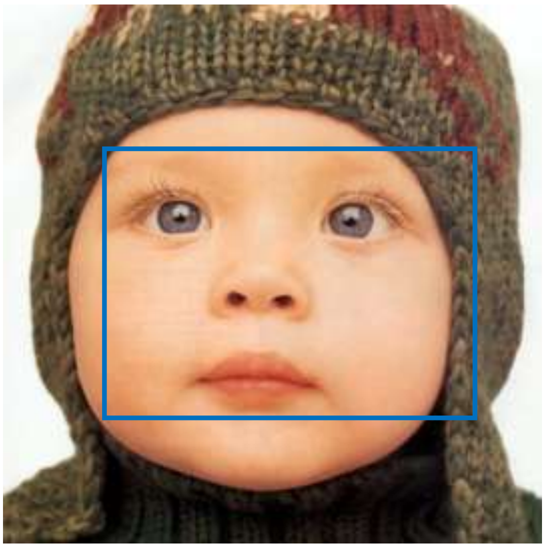


Image HR generated using EDSR
with 2x upscaling from a image LR
(downscaled by bicubic 2x)
512 x 512   340 KB

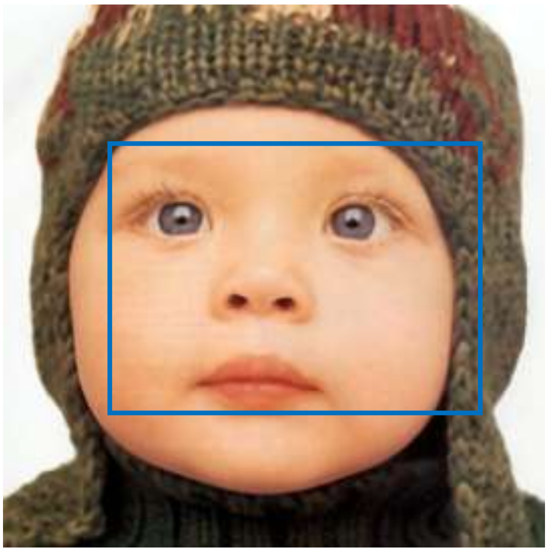


Image HR generated using EDSR
with 4x upscaling from a image LR
(downscaled by bicubic 4x)
512 x 512   284 KB

HR with EDSR 2x upscaling
(37.275 dB/ 0.9662)

HR with EDSR 4x upscaling
(32.131 dB/ 0.8828)

Figure III.26: Super-resolution results for Image baby from Set5 dataset

butterfly from Set5 HR (Ground Truth)
256 x 256 124.5 KB

LR (downscaled by bicubic 2x)
128 x 128 35.6 KB

LR (downscaled by bicubic 4x)
64 x 64 10.3 KB

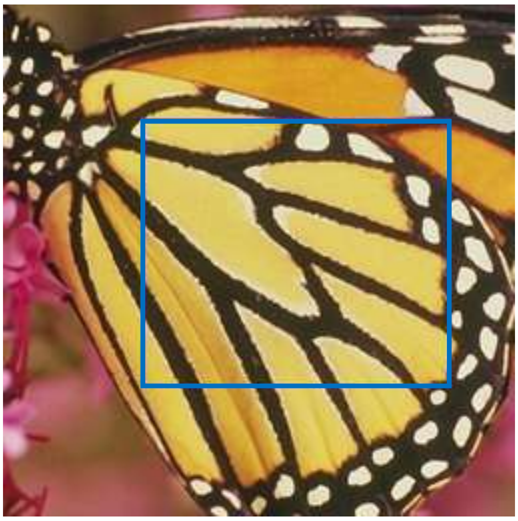


Image HR generated using EDSR with
2x upscaling from a image LR
(downscaled by bicubic 2x)
256 x 256 115.69 KB

Image HR generated using EDSR with
4x upscaling from a image LR
(downscaled by bicubic 4x)
256 x 256 110.74 KB

HR with EDSR 2x upscaling
(33.599 dB/ 0.9714)

HR with EDSR 4x upscaling
(27.087 dB/ 0.8978)

Figure III.27: Super-resolution results for Image butterfly from Set5 dataset.

III.6.1.2 Quantitative Results:

The table III.3 showing the quantitative results obtained by testing the EDSR model on a set of images from different datasets, using 2× and 4× upscaling factors. The table includes both PSNR and SSIM values, as well as the processing time for each case, allowing for an assessment of the model's accuracy and quality, along with its time efficiency.

Table III.3 : Evaluating the Quantitative Results of the EDSR Model

Image	Size of image (Ground Truth)	Scale	PSNR (dB) / SSIM	Processing time (S)
Image 0853 from DIV2K dataset	2040x1356	2x	41.122 dB / 0.9796	1.62 s
		4x	34.001 dB / 0.9341	1.93 s
Image 0013 from DIV2K dataset	2040 x1368	2x	28.799 dB / 0.9687	0.94 s
		4x	25.747 dB / 0.9298	0.26 s
Image 0309 from DIV2K dataset	2040 x1356	2x	37.188 dB / 0.9753	1.65 s
		4x	29.403 dB / 0.9157	1.32 s
Image flowers from Set14 dataset	500 x 362	2x	32.433 dB / 0.9365	0.11 s
		4x	23.117 dB / 0.7497	0.10 s
Image baboon from Set14 dataset	500 x 480	2x	24.010 dB / 0.7615	0.13 s
		4x	20.946 dB / 0.5056	0.14 s
Image baby from Set5 dataset	512 x 512	2x	37.275 dB / 0.9662	0.17 s
		4x	32.131 dB / 0.8828	0.15 s
Image butterfly from Set5 dataset	256 x 256	2x	33.599 dB / 0.9714	0.05 s
		4x	27.087 dB / 0.8978	0.11 s

To objectively evaluate the performance of the EDSR model, a series of experiments were conducted using a selection of images from well-established benchmark datasets such as DIV2K, Set14, and Set5. Two different upscale factors were tested: 2× and 4×. The quality of the reconstructed images was assessed using two widely recognized metrics PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index). Additionally, the processing time for each image was recorded to gauge the model’s efficiency in terms of computation time.

As shown the results table above, the model achieved noticeably better performance at the 2× scale compared to the 4× scale. Higher PSNR and SSIM values were consistently observed at 2×, indicating the model's superior ability to recover fine details at lower scaling factors. For instance, image "0853" from the DIV2K dataset achieved a PSNR of 41.122 dB and an SSIM of 0.9796 at 2×, while these values dropped to 34.001 dB and 0.9341 at 4×. The performance gap became more evident in images with complex textures, such as "baboon" from the Set14 dataset, where the 4× reconstruction yielded a PSNR of just 20.946 dB and an SSIM of 0.5056.

The use of an NVIDIA RTX 3050 GPU with 8GB VRAM drastically reduced image processing times, making the execution times considerably smaller. Surprisingly, the processing time was not constant, but varied with the zoom and what was in the image.

At 2x upscaling, it took longer to process some images. This is due to the fact that the original images did not lose much detail when downscaling, therefore their restoration was tougher for the model, which had to restore details with high precision. Conversely, the time taken to process was decreased at 4x upscaling in some cases. The reason for this is that the original images had lot of details which were lost when they were downscaling, limiting the model from retrieving those details with high precision. The processing operation was therefore relatively simpler computationally even if the final output quality was low.

Overall, the quantitative results demonstrate that the EDSR model performs effectively in reconstructing high-quality details at 2× scaling. However, its performance tends to decline at 4×, primarily due to the increased difficulty in recovering fine details that are significantly lost at higher downscaling levels.

III.6.1.3 Comparison of the Results between Our EDSR Model and the Original Model:

In this section, we compare the results of the EDSR-baseline model in terms of PSNR values as reported in the authors' original paper in 2017 [62], with the results obtained through training that we performed using the same model architecture but with slight differences in the training settings. We will display the original settings of the model along with the settings we trained our model in a table.

Table III.4 presents a detailed comparison between the original EDSR Baseline configuration (as described in the 2017 paper) [62] and the configuration used in this work. While most of the parameters such as the number of residual blocks (residual block = 16), the number of feature maps (Feature maps = 64), the scale factor (×2), and the loss function (L1) remain the same, there is one key difference: Patch Size in original paper is (48x48) but in our work (96x96).

Table III.4 : Differences Between EDSR Baseline (Paper) and Our Configuration

Settings	our Configuration	Original paper [62] (EDSR Baseline)
Patch size	**96**	48
residual block	16	16
Feature maps	64	64
scale	2x / 4x	2x / 4x
epochs	300	300
loss	L1	L1

Table III.5 shows the PSNR results between the original paper and the model we trained. The slight improvement in performance (PSNR) of the model I trained can be explained by the following:

- **More contextual information in each training patch :** A 96×96 patch contains more pixels (four times more than 48×48), providing the model with a wider spatial context during training. This helps it better understand details such as edges and patterns. Often results in improved signal-to-noise ratio (PSNR).
- **Improved generalization capability:** Larger patches may help the model generalize better when tested on new data, as it has seen more diverse and content-rich examples during training.

Table III.5: Quantitative Evaluation of EDSR Baseline and our Trained Model

our (Trained Model)	Original EDSR Baseline [62]	Scale
35.6 dB	34.55dB	2x
29.45 dB	28.94dB	4x

a) Why didn't the authors use patch size (96x96) in the Original EDSR 2017?

In the original EDSR paper (2017), a training patch size of 48×48 was used instead of 96×96, and there are several logical reasons for this in the context of that period, including:

- **Limited computing resources available at the time:**

In 2017, the capabilities of graphics cards (GPUs) and memory capacity were significantly lower than those available today. Using a larger patch, such as 96x96, required more memory and longer training time, which was impractical at the time, especially when training a deep model like EDSR for several days.

- **Standard practices at the time:**

In the context of SR (super-resolution) research at that time, small bits (such as 32, 48, 64) were the norm due to the limited computing resources available to the research community.

- **Balance between performance and cost:**

The 48×48 size was considered a good balance between providing sufficient information for learning and consuming resources. The goal of the original paper was to present an efficient model with high performance within relatively limited resources, especially in the context of competitions like NTIRE

III.6.2 Evaluation of the EDSR Model on Real External Images:

In this section, we present a selection of real-world outdoor images tested using the EDSR model. For each image, we show the low-resolution input and the enhanced output produced by the model (figure III.20 et figure III.21), as well as zoomed-in shots to highlight the visual differences and quality of the super-resolution results.

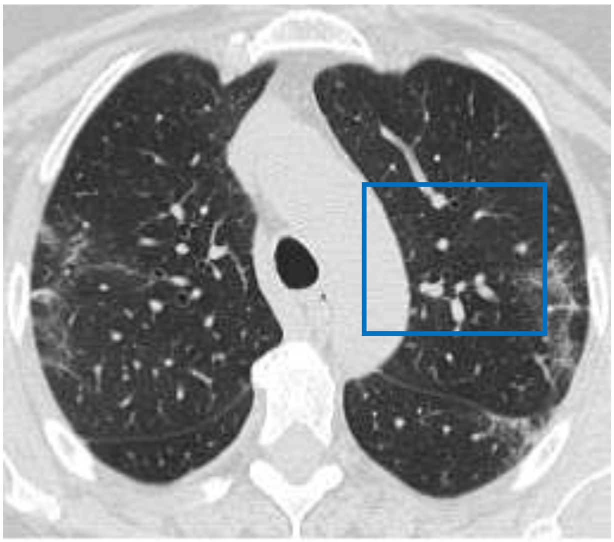


Image before super-resolution
256x202 63.3KB

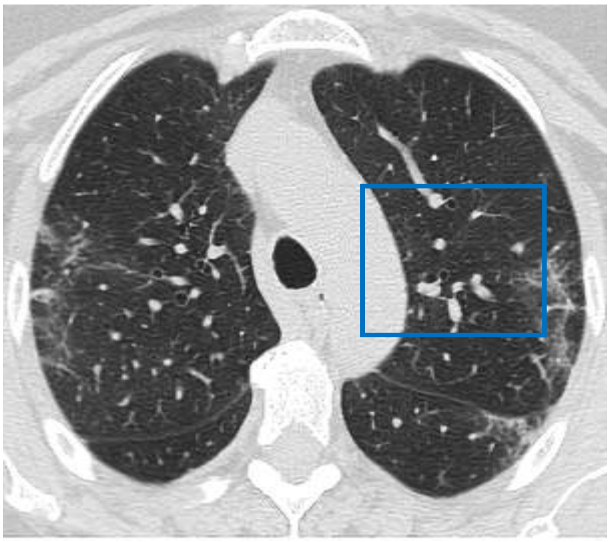


Image After super-resolution with EDSR
upscaling 2x 512x404 321.3KB

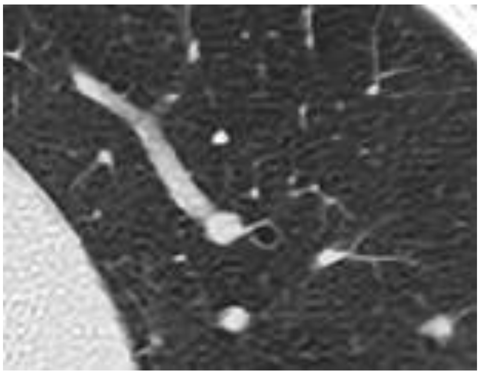


Figure III.28: Super-resolution results for CT (Computed Tomography) image of COVID-19 Lungs

The left image represents the original low-resolution input (256×202, 63.3KB), where anatomical structures such as blood vessels and lung textures appear blurred and less defined.

The right image shows the output after applying 2× upscaling with the EDSR model (512×404, 321.3KB), where finer details, especially in the pulmonary vessels and parenchyma, are noticeably sharper and more distinct

Image before super-resolution
960x540 724KB

Image After super-resolution with EDSR
upscaling 2x 1920x1080 2.32MB

Figure III.29: Super-resolution results for Image “chetma Intensive Language Learning Center

The image on the left represents the low-resolution version (960 x 540, 724 KB), where the text and architectural details appear sparse, especially when zoomed in.The image on the right is the output after zooming in 2x using the model (1920 x 1080, 2.32 MB), and it shows higher resolution and improved clarity. Two areas have been zoomed in to show the differences: the first focuses on the signage bearing the Arabic and French text, and the second shows the windows and air conditioning unit, both of which show significant improvements in edges, texture, and text clarity.

III.7 An Overview of the Web Application for Model Deployment:

In order to make the trained model accessible to end users, we created a web application that offers an interactive tool for users to upload a low-resolution image and see the high-resolution image directly through the application of the EDSR model.

The web app was built using the following tools:

- **Frontend:** Built using HTML, CSS, and JavaScript to offer a user-friendly interface.
- **Backend:** We used Python with the Flask framework to connect the frontend to the trained model (Backend). Images are sent to the server for processing and then the result is returned to the user.

Web app was run locally (Localhost) during development, as this ensures data privacy and ease of testing.

By designing a web interface with Flask, I was able to make the model easier to use and test in practice, which helped display results visually and interactively and made it easier to experience the project without having to run code manually.

In order to use the webapp for all people, we used and share it online, we used ngrok. It is a tool used to create a secure tunnel from the Internet to your local device, allowing you to easily share local web applications over the Internet.

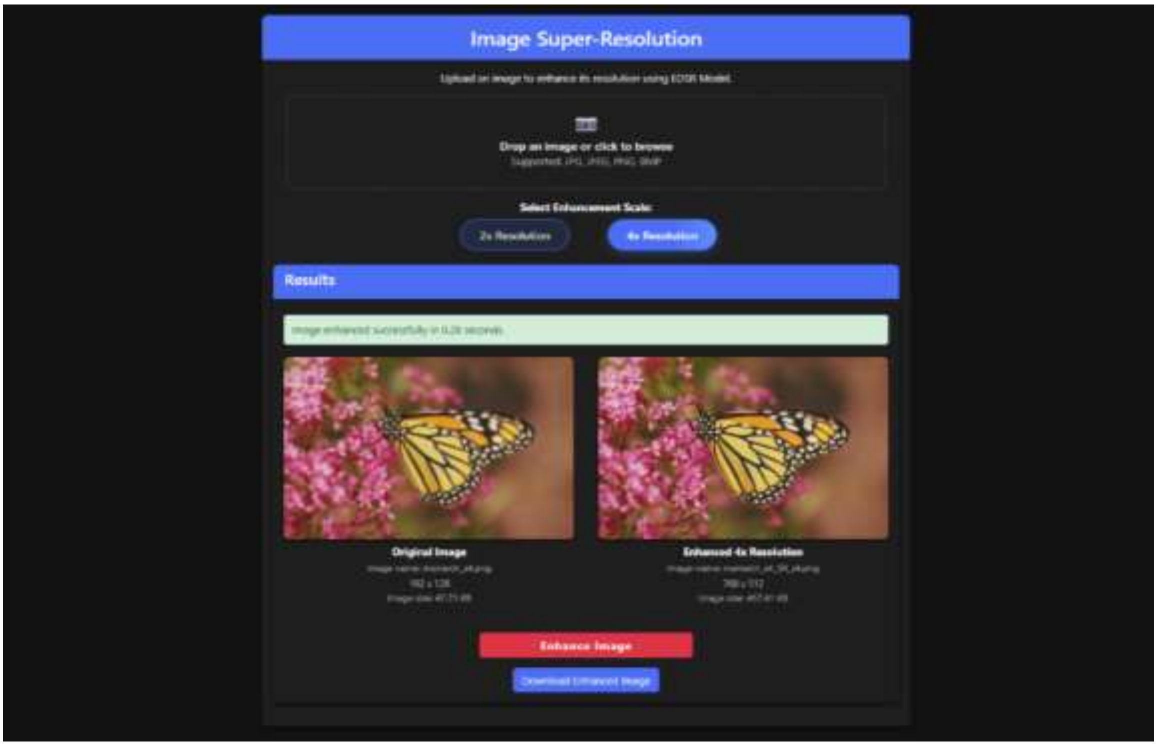


Figure III.30**:** web app interface from a pc

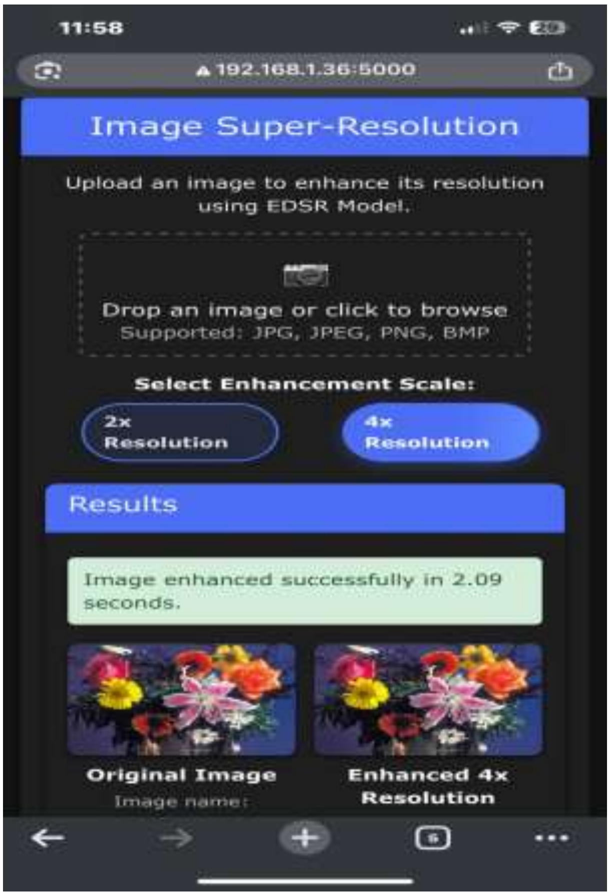


Figure III.31: web app interface from a mobile

III.8 Conclusion:

In this chapter, we have presented the complete setup and methodology used for developing and evaluating the EDSR (Enhanced Deep Super-Resolution) model. We detailed the dataset selection, particularly the use of the high-quality DIV2K dataset, which provided diverse and rich image content for effective model training. We outlined the hardware and software environments, including the use of Visual Studio Code, Python, CUDA, and the PyTorch library, which enabled us to implement and accelerate deep learning tasks efficiently. The training process was carefully designed, leveraging optimization techniques such as the Adam optimizer, patch-based training, and data augmentation to maximize learning while minimizing overfitting and computational costs. For evaluation, we relied on objective metrics like PSNR, SSIM which demonstrated that the EDSR model significantly improved the resolution of low-resolution images. Despite these promising results, several challenges were encountered, such as long training times and high GPU memory demands, particularly at higher scaling factors like 4×. However, the training strategy and parameter tuning helped mitigate many of these difficulties. Overall, the EDSR model proved to be an effective solution for image super-resolution, showcasing strong quantitative and qualitative results

General Conclusion:

The problem of image super-resolution (SR) remains one of the most active and important research topics in the field of computer vision. One of the core challenges of this task lies in realistically recovering the lost information, which requires models capable of learning rich visual representations and effectively inferring missing data. To this day, image super-resolution still faces several challenges that prevent it from achieving perfect performance in all scenarios.

In this thesis, we focused on the problem of single-image super-resolution using only deep learning techniques. After conducting a comparative study of state-of-the-art methods, we selected the Enhanced Deep Super-Resolution (EDSR) model as one of the most efficient and suitable approaches for addressing this type of problem. EDSR is characterized by a deep architecture that omits non-essential components such as Batch Normalization layers, which improves both performance and stability while reducing memory consumption. This model allowed us to precisely study its behavior and evaluate its performance using standard benchmark datasets and quality metrics such as PSNR and SSIM, in addition to testing it on real-world images.

The EDSR architecture relies on repeated residual blocks and avoids the use of Batch Normalization, which helps to improve gradient flow during training and enhances the extraction of visual features. Furthermore, the use of the Pixel Shuffle algorithm for the final upscaling step contributes to reducing artifacts and ensures better detail distribution across the upscaled image, compared to traditional methods such as bilinear or bicubic interpolation.

To evaluate the model both quantitatively and qualitatively, we adopted standard metrics such as PSNR and SSIM, along with visual analysis of the results on well-known datasets (e.g., DIV2K, Set5, Set14). The findings demonstrate that EDSR delivers high-quality results—particularly in terms of preserving edge sharpness and reducing visual noise—making it a promising option for applications that require high-resolution imagery, such as computer vision, medical diagnostics, and surveillance systems. We also analyzed the impact of the upscaling factor on the quality of the generated image and found that the model's performance varies depending on the scale level. Specifically, a scale factor of ×2 produced more accurate results than higher scaling factors such as ×4, primarily because less information is lost in the smaller-scale scenario. The results of this study showed that EDSR's deep convolutional architecture enables it to effectively exploit embedded visual patterns within the image, thereby reconstructing intricate features of the original high-resolution content.

Despite the promising results achieved by the EDSR model in enhancing image resolution and recovering fine details, this work opens several avenues for future research that could further improve performance and broaden its applicability. These include:

- Hybrid Model Integration: One promising direction is to combine EDSR with other advanced approaches, such as Generative Adversarial Networks (GANs), as in SRGAN, to generate more perceptually realistic images. This would shift the focus beyond purely numerical metrics like PSNR and aim for improved perceptual quality.
- Expanded Experimental Scope: Applying the model to a wider range of real-world data—such as medical images, satellite imagery, or low-quality videos—could provide insight into its generalization capabilities and allow for targeted improvements based on the type of image.
- Computational Efficiency and Optimization: Although EDSR performs well, it requires significant computational resources. A future direction could involve developing lightweight versions of the model (e.g., EDSR-lite) or using model compression techniques to enable its deployment on embedded systems and mobile devices.
- Multi-Feature Representation: Model performance may be further enhanced by incorporating additional visual features—such as texture and edge information—into the representation, especially to handle challenging cases involving fine detail loss or structural ambiguity.

I. Bibliography

[1] Spatial Resolution (down sampling and up sampling) in image processing. (2023, jan 03). Récupéré sur geeksforgeeks.org: https://www.geeksforgeeks.org/spatial-resolution-down-sampling-and-up-sampling-in-image-processing/

[2] Swamy, S. A. (2019, May 5). introduction to digital image processing. Récupéré sur sunnybellary: https://sunnybellary.com/course/intro-to-dip/tutorial1/

[3] https://en.wikipedia.org/wiki/Image_resolution

[4] https://www.adobe.com/uk/creativecloud/photography/discover/image-resolution.html

[5] nicolasroger.fr. (s.d.). Récupéré sur Définition et résolution d'une image numérique: https://www.nicolasroger.fr/2015/01/definition-et-resolution-dune-image-numerique/

[6] Kenney, K. (2024, February 22). DPI vs PPI: What's the Difference? Retrieved from greatbigphotographyworld: https://greatbigphotographyworld.com/dpi-vs-ppi/

[7] https://www.adobe.com/uk/creativecloud/photography/discover/high-resolution.html

[8] Hoyrup, A. (2011, nov 15). Récupéré sur anvydigital: http://blog.anvydigital.com/2011/11/getting-most-out-of-your-next-print-job.html?m=1

[9] Qu'est-ce que la basse résolution? (2025). Récupéré sur canada.lenovo: https://www.lenovo.com/ca/en/glossary/low-resolution/

[10] High Resolution vs Low Resolution: How it Matters. (2023, 03 30). Récupéré sur castr: https://castr.com/blog/high-resolution-vs-low-resolution-how-it-matters

[11] Attention aux images en basse résolution ! (2025). Récupéré sur copy-media: https://www.copy-media.net/attention-aux-images-basse-resolution/

[12] Márques, A. (2025, January 24). A Comprehensive Introduction to Super Resolution for Satellite Images. Récupéré sur digitalsense.: https://www.digitalsense.ai/blog/introduction-to-super-resolution-for-satellite-images

[13] Yash. (2021, May 27). Deep Learning for Image Super-Resolution. Récupéré sur analyticsvidhya: https://www.analyticsvidhya.com/blog/2021/05/deep-learning-for-image-super-resolution/

[14] Mukherjee, S. (2024). Unlocking Image Clarity: A Comprehensive Guide to Super-Resolution Techniques. Récupéré sur pyimagesearch:

https://pyimagesearch.com/2024/04/29/unlocking-image-clarity-a-comprehensive-guide-to-super-resolution-techniques/

[15] Olšan, J. (2023, 03 06). Récupéré sur hwcooling: https://www.hwcooling.net/en/nvidia-rtx-video-super-resolution-out-what-you-need-to-know/

[16] Chen, Zheng, et al. "Ntire 2024 challenge on image super-resolution (x4): Methods and results." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[17] Li, Yufei, Bruno Sixou, and Francois Peyrin. "A review of the deep learning methods for medical images super resolution problems." *Irbm* 42.2 (2021): 120-133.

[18] Yang, Jianchao, and Thomas Huang. "Image super-resolution: Historical overview and future challenges." Super-resolution imaging. CRC Press, 2017. 1-34.

[19] Jakub Ke¸dzierski, & Andrzej Rusiecki. (2024). Deep Learning Approach to Improve Image Super-Resolution with. 28th International Conference on Knowledge-Based and Intelligent Information & Engineering (p. 1). ScienceDirect.

[20] Jianchao Yang, & Thomas Huang. (s.d.). Image super-resolution: Historical overview. Récupéré sur mclab.unipv: http://mclab.unipv.it/Master/Image%20Super-Resolution%20Hystorical%20Overview%20and%20Future%20Challenges.pdf

[21] Rajapaksha, N. (2022, Aug 6). Image Interpolation. Récupéré sur medium: https://medium.com/@nuwanthidileka/image-interpolation-5e4cbe90603a

[22] cambridgeincolour. (s.d.). Récupéré sur DIGITAL IMAGE INTERPOLATION: https://www.cambridgeincolour.com/tutorials/image-interpolation.htm

[23] Jorgecardete. (2024, feb 8). The Art and Science of Interpolation. Récupéré sur medium: https://medium.com/thedeephub/the-art-and-science-of-interpolation-b12b99f2e053

[24] G, vedi. Image Processing – Nearest Neighbour Interpolation. 03 mar 2020. <https://theailearner.com/2018/12/29/image-processing-nearest-neighbour-interpolation/>.

[25] atul, k. &. (2018, Dec 29). Image Processing – Bicubic Interpolation. Récupéré sur theailearner: https://theailearner.com/2018/12/29/image-processing-bicubic-interpolation/

[26] Wang, Zhihao, Jian Chen, and Steven CH Hoi. "Deep learning for image super-resolution: A survey." IEEE transactions on pattern analysis and machine intelligence 43.10 (2020): 3365-3387.

[27] Dong, Chao, et al. "Image super-resolution using deep convolutional networks." IEEE transactions on pattern analysis and machine intelligence 38.2 (2015): 295-307.

[28] Su, Yuanxin, et al. "Efficient FPGA Binary Neural Network Architecture for Image Super-Resolution." Electronics 13.2 (2024): 266.

[29] Li, Wenhao, et al. "Super-resolution reconstruction, recognition, and evaluation of laser confocal images of hyperaccumulator Solanum nigrum endocytosis vesicles based on deep learning: Comparative study of SRGAN and SRResNet." Frontiers in Plant Science 14 (2023): 1146485.

[30] Su, Kai, Xueming Li, and Xuewei Li. "Single Image Super-Resolution Based on Generative Adversarial Networks." Image and Graphics Technologies and Applications: 15th Chinese Conference, IGTA 2020, Beijing, China, September 19, 2020, Revised Selected Papers 15. Springer Singapore, 2020.

[31] K, B. (2024). SRGAN: Super Resolution Generative Adversarial Networks. Récupéré sur digitalocean: https://www.digitalocean.com/community/tutorials/super-resolution-generative-adversarial-networks

[32] Generative Adversarial Network (GAN). n.d. <https://www.geeksforgeeks.org/generative-adversarial-network-gan/>.

[33] Chawda, G. (2022, 04 07). Enhanced Deep Residual Networks for single-image super-resolution. Récupéré sur keras: https://keras.io/examples/vision/edsr/

[34] Schmidt, Luca, and Nicole Ludwig. "Wind Power Assessment based on Super-Resolution and Downscaling--A Comparison of Deep Learning Methods." arXiv preprint arXiv:2407.08259 (2024).

[35] Kumar, D., et al. "A comprehensive review and comparison of image super-resolution techniques." International Journal of Advanced Engineering, Management and Science 10: 40-45.

[36] Cole, Stryker and kavlakoglu Eda. *What is artificial intelligence (AI)?* 9 august 2024. <https://www.ibm.com/think/topics/artificial-intelligence>.

[37] *What is machine learning?* 22 sep 2021. <https://www.ibm.com/think/topics/machine-learning>.

[38] Fatmanurkutlu. Introduction to Machine Learning. 5 jan 2024. <https://medium.com/@fatmanurkutlu1/introduction-to-machine-learning-d3de5694d274>.

[39] n.d. <https://www.newtechdojo.com/3-types-of-machine-learning/>.

[40] Supervised Machine Learning. 17 mar 2025. https://www.tpointtech.com/supervised-machine-learning.

[41] Unsupervised Machine Learning. 17 mar 2025. https://www.tpointtech.com/unsupervised-machine-learning.

[42] BIN QIAN,, et al. "Orchestrating the Development Lifecycle of Machine Learning-Based IoT." 29 may 2020. arxiv.org. <https://arxiv.org/pdf/1910.05433>.

[43] Reinforcement Learning Tutorial. 17 mar 2025. <https://www.tpointtech.com/reinforcement-learning>.

[44] Daniel. Deep Learning: All about this concept. 28 apr 2021. <https://datascientest.com/en/all-about-deep-learning>.

[45] Michael A. Nielson, "Neural Network and Deep Learning", Determination Press, 2015

[46] Wolfewics, Arne. *Deep Learning vs. Machine Learning – What's The Difference?* 30 septembre 2024. <https://levity.ai/blog/difference-machine-learning-deep-learning>.

[47] Mahmood, Huda. *5 Main Types of Neural Networks and their Applications*. 11 june 2024. <https://datasciencedojo.com/blog/5-main-types-of-neural-networks/>.

[48] *Chapter 1. Exploring the Landscape of Artificial Intelligence*. n.d. <https://www.oreilly.com/library/view/practical-deep-learning/9781492034858/ch01.html#drdot_may_carson>.

[49] "Chapter 9Convolutional Networks." n.d. <https://www.deeplearningbook.org/contents/convnets.html>.

[50] Manav. *Convolutional Neural Networks (CNN) in Deep Learning*. 18 feb 2025. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/#be44>.

[51] Xiang Fan and Phuong Truong. *An Introduction to Convolutional Neural Network (CNN)*. 11 feb 2022. <https://medium.com/sfu-cspmp/an-introduction-to-convolutional-neural-network-cnn-207cdb53db97>.

[52] Sharma, Pulkit. *A Comprehensive Tutorial to learn Convolutional Neural Networks from Scratch (deeplearning.ai Course #4)*. 16 dec 2024. <https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/>.

[53] Zhao, Lei, and Zhonglin Zhang. "A improved pooling method for convolutional neural networks." *Scientific Reports* 14.1 (2024): 1589.Buckner, Cameron. "Deep Learning: A Philosophical Introduction." *Philosophy Compass* (2019).

[54] Buckner, Cameron. “Deep Learning: A Philosophical Introduction.” Philosophy Compass (2019).

[55] Unzueta, Diego. *Fully Connected Layer vs. Convolutional Layer: Explained*. 24 feb 2025. <https://builtin.com/machine-learning/fully-connected-layer>.

[56] Vishnuam. *Activation Functions in Convolutional Neural Networks (CNN)*. 6 oct 2024. <https://medium.com/@vishnuam/activation-functions-in-convolutional-neural-networks-cnn-b1114a1b9b4d>.

[57] *Build a Convolutional Neural Network*. 30 may 2024. <https://carpentries-incubator.github.io/intro-image-classification-cnn/instructor/03-build-cnn.html#top>.

[58] I, Azeem -. *Understanding ResNet Architecture: A Deep Dive into Residual Neural Network*. 14 nov 2023. <https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9>.

[59] *Residual Networks (ResNet) - Deep Learning*. n.d. <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/>.

[60] Bangar, Siddhesh. *Resnet Architecture Explained*. 05 jul 2022. <https://medium.com/@siddheshb008/resnet-architecture-explained-47309ea9283d>.

[61] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[62] Lim, Bee, et al. "Enhanced deep residual networks for single image super-resolution." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017.

[63] Ullah, Safi, and Seong-Ho Song. "SRResNet Performance Enhancement Using Patch Inputs and Partial Convolution-Based Padding." *Computers, Materials & Continua* 74.2 (2023).

[64] Chao, Heng-Sheng, et al. "Generating high-resolution Ct slices from two image series using deep-learning-based resolution enhancement methods." *Diagnostics* 12.11 (2022): 2725.

[65] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[66] Prabakaran, J., and P. Selvaraj. "Implementation of ResNet-50 with the Skip Connection Principle in Transfer Learning Models for Lung Disease Prediction." *International Conference on Intelligent Systems and Sustainable Computing*. Singapore: Springer Nature Singapore, 2022.

[67] Cochard, David. *EDSR : A Machine Learning Model for Super-resolution Image Processing*. 18 may 2021. <https://medium.com/axinc-ai/edsr-a-machine-learning-model-for-super-resolution-image-processing-9deaf36b24ed>.

[68] Mukherjee, Shaoni. *A Complete Guide to Image Super-Resolution in Deep Learning and AI*. 1 may 2025. <https://www.digitalocean.com/community/tutorials/image-super-resolution>.

[69] Theerthala, Sai Pranav. (2023). Benefits of Implementing Super Resolution to Video and Photo Evidences in Court Cases. 10.47611/harp.262.

[70] Tsang, Sik-Ho. *Review: EDSR & MDSR — Enhanced Deep Residual Networks for Single Image Super-Resolution (Super Resolution)*. 24 apr 2020. <https://sh-tsang.medium.com/review-edsr-mdsr-enhanced-deep-residual-networks-for-single-image-super-resolution-super-4364f3b7f86f>.

[71] https://www.researchgate.net/figure/Workflow-of-EDSR-x4-image-enhancement-b-Feature-Extraction-Layers-Utilizes_fig3_376182043

[72] Galar, Mikel, et al. "Super-resolution of sentinel-2 images using convolutional neural networks and real ground truth data." *Remote Sensing* 12.18 (2020): 2941.

[73] *PixelShuffle*. n.d. <https://docs.pytorch.org/docs/stable/generated/torch.nn.PixelShuffle.html>.

[74] Yang, Xuan, et al. "A block shuffle network with superpixel optimization for Landsat image semantic segmentation." *Remote Sensing* 14.6 (2022): 1432.

[75] Kingma, Diederik P. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

[76] *What is Adam Optimizer?* n.d. <https://www.geeksforgeeks.org/adam-optimizer/>.

[77] Alake, Richmond. *Loss Functions in Machine Learning Explained*. 4 dec 2024. <https://www.datacamp.com/tutorial/loss-function-in-machine-learning>.

[78] Timofte, Radu, et al. "DIV2K dataset: DIVerse 2K resolution high quality images as used for the challenges@ NTIRE (CVPR 2017 and CVPR 2018) and@ PIRM (ECCV 2018)." 2018,

[79] *Python Introduction*. n.d. <https://www.w3schools.com/python/python_intro.asp>.

[80] —. *Introduction to NVIDIA CUDA Achieving Peak Performance with H100 for AI and Deep Learning*. 30 nov 2024. <https://www.digitalocean.com/community/tutorials/intro-to-cuda>.

[81] Payong, Adrien. *Step-by-Step Guide to Installing CUDA and cuDNN for GPU Acceleration*. 10 feb 2025. <https://www.digitalocean.com/community/tutorials/install-cuda-cudnn-for-gpu>.

[82] Yasar, Kinza. *Definition PyTorch*. n.d. <https://www.techtarget.com/searchenterpriseai/definition/PyTorch>.

[83] https://opencv.org/about/

[84] Fiveable. "Peak Signal-to-Noise Ratio (PSNR) – Biomedical Engineering II." Edited by Becky Bahr, Fiveable, 2024, https://fiveable.me/key-terms/biomedical-engineering-ii/peak-signal-to-noise-ratio-psnr. Accessed 21 May 2025.

[85] *Peak Signal-to-Noise Ratio as an Image Quality Metric*. 28 jun 2024. <https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html>.

[86] WANG, Zhou, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004, 13.4: 600-612.

[87] Pandey, Abhishek Kumar. *Structural Similarity Index(SSIM)*. 9 mar 2024. <https://medium.com/@akp83540/structural-similarity-index-ssim-c5862bb2b520>