PEOPLE'S DEMOCTATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University of Mohamed El-Bachir El-Ibrahimi–Bordj Bou Arreridj Faculty of Science and Technology

Department of Electronic

Master Thesis

Presented to get

THE MASTER'S DIPLOMA

Sector: Electronics

Specialty : **Embedded System Electronics**

By

- **Mr.GUERBOUKHA Achour**
- **Mr.BELKHIRI Cherif**

Entitled

Ensemble Learning for Super Resolution of Magnetic Resonance Images Based on Deep Learning

Supported on:

By the evaluation committee composed of:

First name & family name :	*Grade*	*Quality*	*Establishment*
M.F.HAMADACHE	***MAA***	***Chairman***	***Univ-BBA***
M.D.E. BOUDECHICHE	***MCB***	***Supervisor***	***Univ-BBA***
M.N.DIFFELLAH	***MCA***	***Examiner***	***Univ-BBA***

Academic year 2022/2023

بسم الله الرحمن الرحيم

Appreciation

ALHAMDULILLAH first of all for always helping me despite all the circumstances, to get this work done. I would like to Thank all the people who have contributed to my success And my training and who helped me during the writing of this thesis. I sincerely thank all the professors and all the people who have advised and supported me throughout my years of study

.

I hope ALLAH will help us in what is coming☺

Dedication

Cherif

I want to give this work to my parents, who have always been there for me, my little family Madjida, Ismahan, Ramzi, Besma, for all the encouragement they gave me,
To the little chicks Mohammed, Nouh and Taouba,
To all my friends in my school career and abroad,
To my friend Achour for all the moments and wars we faced...♥

THANKS

Achour

I dedicate this to my family. To my father, who was my first and most influential teacher, and to my mother, who embodied the essence of paradise. I also cherish the presence of my brothers, Moussa and Ilyes, and my sisters, Katia and Sara, who have been pillars of support throughout my life. The little ones in our family Youcef, the axis of our happiness, have constantly inspired me with their encouragement and assistance. I extend my warm greetings to all my friends, and to you, who have been a constant source of inspiration. I hope that one day we will meet and never be parted. A special mention goes to my dear friend, Senior Cherif Belkhiri, whose unwavering encouragement hasbeen a guiding light. Together, we shall strive for our eternal paradise.

Abstract:

our study focuses on using deep learning to improve the accuracy of Magnetic Resonance (MR) images. To begin, we start by reducing the size of High-Resolution (HR) images then employ interpolation techniques (Linear -Nearst Neighbor - bicubic) to enhance the quality of the Low-Resolution (LR) image. This process leads to a noticeable improvement in image resolution, allowing for the preservation of finer details. Subsequently, we leverage the power of a specialized Deep Convolutional Neural Network (CNN) known as the DNCNN, which is specifically designed to excel in super-resolution tasks. The DNCNN is applied to enhance three Super-Resolution (SR) images generated through the previous steps, resulting in further enhancement and refinement. These SR images are effectively combined using Ensemble Learning to produce a single high-quality SR image. Overall, this research presents a novel deep learning approach to enhance MR images, demonstrating the potential to significantly improve their precision and quality in medical imaging applications.

Résumé:

Notre étude vise à utiliser l'apprentissage profond pour améliorer la précision des Images par Résonance Magnétique (IRM). Pour commencer, nous réduisons la taille des images Haute Résolution (HR) puis utilisons des techniques d'interpolation (linéaire – plus proche voisin - bicubic) pour améliorer la qualité de l'image Basse Résolution (BR). Ce processus entraîne une amélioration notable de la résolution de l'image, permettant ainsi la préservation des détails plus fins. Ensuite, nous exploitons la puissance d'un réseau de neurones convolutifs profonds spécialisé appelé DNCNN, conçu spécifiquement pour exceller dans les tâches de super-résolution. Le DNCNN est utilisé pour améliorer trois images de Super-Résolution (SR) générées à partir des étapes précédentes, ce qui entraîne une amélioration et un affinement supplémentaires. Ces images SR sont ensuite combinées de manière efficace à l'aide de l'apprentissage en ensemble pour produire une seule image SR de haute qualité. Dans l'ensemble, cette recherche présente une nouvelle approche d'apprentissage profond pour améliorer les images d'IRM, démontrant ainsi le potentiel d'amélioration significative de leur précision et de leur qualité dans les applications d'imagerie médicale.

ملخص:

تركز دراستنا على استخدام التعلم العميق لتحسين دقة صور الرنين المغناطيسي. في البداية، نبدأ بتقليل حجم الصور عالية الدقة ثم نستخدم تقنيات الاستيفاء (الخطية -الجار القريب -بيكوبيك) لتعزيز جودة الصورة منخفضة الدقة تؤدي هذه العملية إلى تحسن ملحوظ في دقة الصورة، مما يسمح بالحفاظ على تفاصيل دقيقة. بعد ذلك، نستفيد من قوة شبكة عصبية تلافيفية عميقة متخصصة (س.ن.ن) تعُرف باسم د ن س ن ن، والتي تم تصميمها خصيصًا للتفوق في المهام فائقة الدقة. يتم تطبيق د.ن.س.ن.ن، لتعزيز ثلاث صور فائقة الدقة تم إنشاؤها من خلال الخطوات السابقة، مما أدى إلى مزيد من التحسين والصقل. يتم دمج صور هذه بشكل فعال باستخدام تعلم التجميع لإنتاج صورة واحدة عالية الجودة. بشكل عام، يقدم هذا البحث نهج تعلم عميق جديد لتعزيز صور الرنين المغناطيسي، مما يوضح القدرة على تحسين دقتها وجودتها بشكل كبير في تطبيقات التصوير الطبي.

Table of content

- **Chapter III: Implementation and results**

List of Figures:

List of Abbreviations

SR	super-resolution
LR	low resolution
HR	high resolution
MRI	magnetic resonance imaging
CNN	Convolutional Neural Networks
LN	layer normalization
ReLU	Rectified Linear Unit
DNCNN	denoising Convolutional Neural Network
GAN	generative adversarial network
PET	Positron Emission Tomography
CT	Computed Tomography
MLP	Multilayer perceptron
X-Ray	X-radiation
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent

General introduction

The importance of super resolution techniques has grown due to the increasing demand for highquality images in critical applications like medicine, security, and other domains[1]. These techniques aim to enhance the resolution of low-resolution (LR) images, enabling the acquisition of clearer and more visuals detailed.The methods of obtaining ultra-accurate images have evolved, after the advent of deep learning (DL) techniques which showed good results.

This master thesis deals with the enhancement algorithms of Magnetic Resonance Imaging (MRI) of poor quality due to low exposure time and low dose of radiation for the sake of healthy of the patient.Addressing these issues is critical to ensuring the accuracy of diagnostic interpretations and improving the usefulness of the MRI technique.

In our study, we will improve the quality of MRI images by some new techniques of deep learning, relying on a branch of CNN, which is DNCNN, which is considered the ideal solution in our study.

In chapter 1, we begin by explaining the concept of noise in images and the different medical image modalities. We cover Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Positron Emission Tomography (PET), microscopy images, and capsule imaging. We also touch upon conventional Super Resolution (SR) techniques such as Bicubic interpolation, Linear interpolation, and Nearest Neighbor interpolation. Additionally, we introduce Super Resolution with deep learning, particularly utilizing convolutional neural networks (CNNs). This chapter sets the stage for understanding the fundamentals of image processing and its applications in the medical field.

In chapter 2, our focus was on neural networks and deep learning concepts. We delved into the principal parts of neural networks, explaining their significance within the broader context of deep learning. We covered fundamental concepts such as activation functions, loss functions, and backpropagation, which are essential for understanding how neural networks operate. Additionally, we explored the Adam optimization algorithm, a crucial component for training neural networks and optimizing their performance. Furthermore, we discussed convolutional neural networks (CNNs) and their key components, including the convolutional layer, pooling layer, feature detector, and fully connected layer. Lastly, we highlighted the importance of selecting appropriate hyperparameters to achieve optimal performance. Additionally, we also touched upon ensemble learning, which involves combining multiple models to improve overall performance, as a promising technique to further enhance the capabilities of neural networks.

In the last chapter, we included the content of our topic and the results we achieved. We presented the methods that we used to obtain these results and showcased the data that our training program relied on, as well as the interface we worked on. Furthermore, this chapter provided a detailed explanation of our working methodology in the subject and discussed some of the challenges we encountered during the process.

Chapter I: Image Super Resolution

Abstract:

The chapter 1 of this study focuses on two main aspects: noise image and medical image modalities (MRI, CT, PET...), as well as conventional super resolution (SR) techniques (Bicubic, Linear, nearest Neghibor) and SR with deep learning.

Summary

I.1. Introduction:

Image SR refers to the process of generating a high resolution (HR) image from one or more low resolution (LR) inputs. The goal is to obtain a more detailed, clearer and visually pleasing image that is comparable to the one that would have been obtained from a high-resolution input. SR has applications [2]in a wide range of fields, including medical imaging [3], surveillance [4], and satellite imaging [5], the problem of SR is challenging, as it involves inferring missing high frequency information that is not present in the LR input. There are many approaches of super resolution, including conventional such as Bicubic interpolation, Linear interpolation, and Nearest Neighbor interpolation [6].

I.2. Noise image:

The noise image can have different characteristics depending on the noise type and intensity. For example: **[7]**

I.2.1. Gaussian noise:

The image may exhibit a fine graininess or speckling effect throughout, with no specific pattern. It can make the image look slightly blurred or distorted.

I.2.2. Salt and pepper noise:

Random black and white pixels, resembling grains of salt and pepper, may be scattered throughout the image. These pixels can disrupt the overall image quality and make it appear noisy and distorted.

I.2.3. Poisson noise:

In images with low light conditions, you might notice random variations in pixel intensity. This noise can cause the image to have a speckled appearance, especially in darker regions.

I.2.4. Speckle noise:

This type of noise can result in granular patterns of bright and dark regions, similar to a textured noise. It may appear as small, irregular spots or patches throughout the image.

These are just a few examples of how noise can affect an image. The specific characteristics and severity of the noise depend on the underlying cause and the image capture or transmission conditions.

I.3. Medical Image modalities:

A medical image refers to any visual representation of the internal structures or functions of the human body or specific body parts, obtained through various imaging techniques such as X- ray [32], ultrasound [8], Magnetic Resonance Imaging (MRI) [9], computed tomography (CT) [10]and positron emission tomography (PET) [11]. These images are used by medical professionals to help in diagnosis, treatment planning and monitoring of diseases and conditions [12].

I.3.1. Magnetic Resonance Imaging (MRI):

An MRI image is a type of medical image that is generated using a magnetic resonance technique. It uses a strong magnetic field and radio waves to create detailed cross-sectional images of the internal structures of the body. MRI images provide valuable information about organs, tissues, and abnormalities, helping medical professionals diagnose and monitor various medical conditions. MRI is commonly used for imaging the brain, spine, joints, abdomen, and other areas of the body.[9]

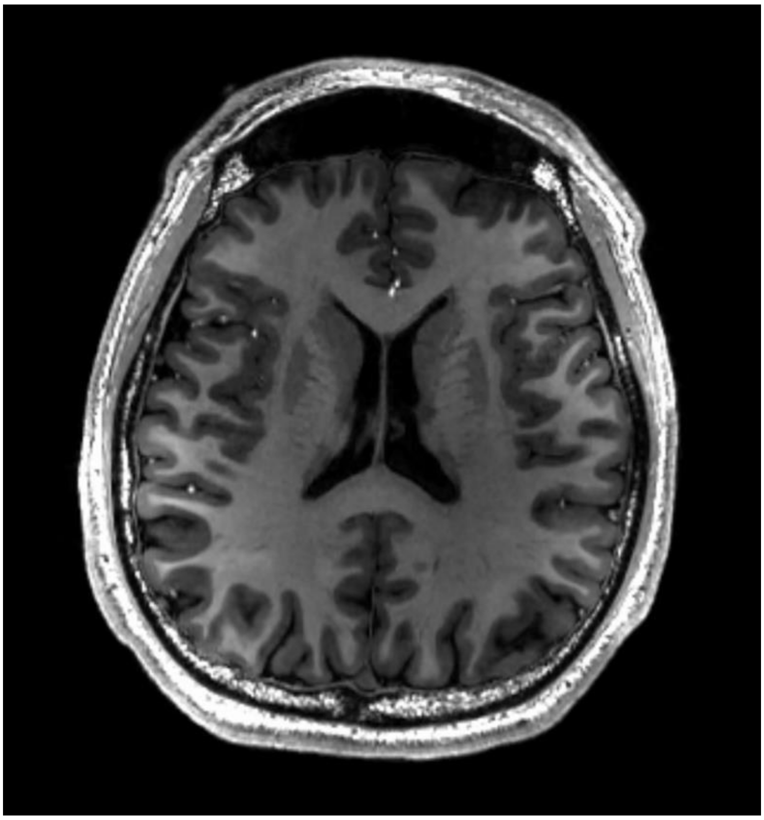


Figure I.1: Magnetic Resonance Imaging.**[38]**

I.3.2. Computed Tomography (CT):

Computed Tomography (CT) is a medical imaging technique that uses X-ray technology to generate cross-sectional images of the body. It provides detailed information about the internal structures, organs, and tissues in a non-invasive manner. CT scans involve taking multiple X-ray images from different angles around the body, which are then processed by a computer to create a series of cross-sectional images called "slices." These slices can be further reconstructed to create 3D images.

CT imaging is valuable in diagnosing and evaluating various medical conditions, such as detecting tumors, assessing injuries, examining blood vessels, and guiding interventional procedures. It offers superior visualization of structures compared to traditional X-rays and is particularly useful for brain imaging, chest, abdomen, pelvis, and bones. CT scans are typically performed in specialized imaging centers or hospitals, and the procedure is relatively quick, often lasting only a few minutes.[10]

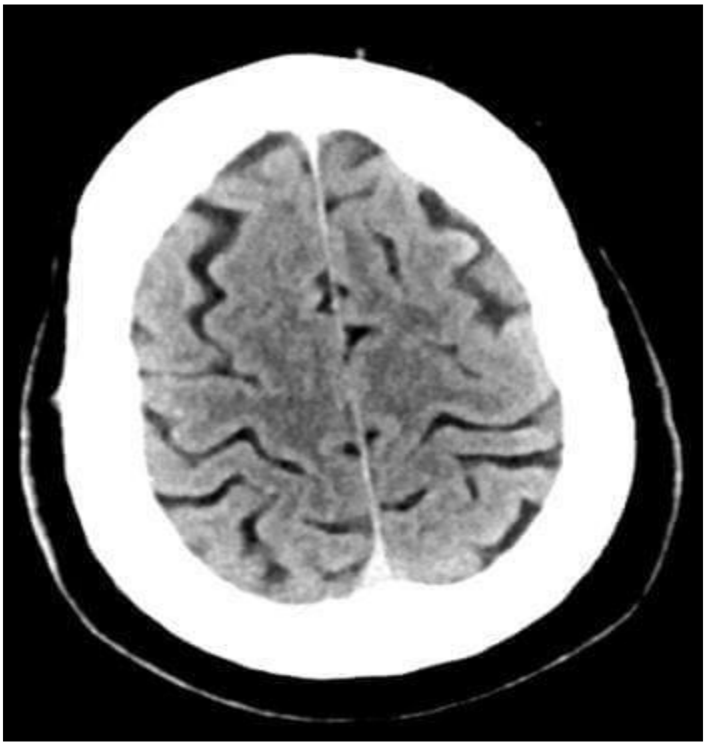


Figure I.2: CT Imaging in Neurologic Disorders. [33]

I.3.3. Positron Emission Tomography (PET):

Positron Emission Tomography (PET) is a medical imaging technique that uses a radioactive tracer to visualize and measure metabolic activity and biochemical processes in the body. It provides functional information by detecting gamma rays emitted from the decay of positrons produced by the tracer. PET scans are used to evaluate organ function, detect and stage diseases, monitor treatment response, and aid in research and clinical trials. [11]

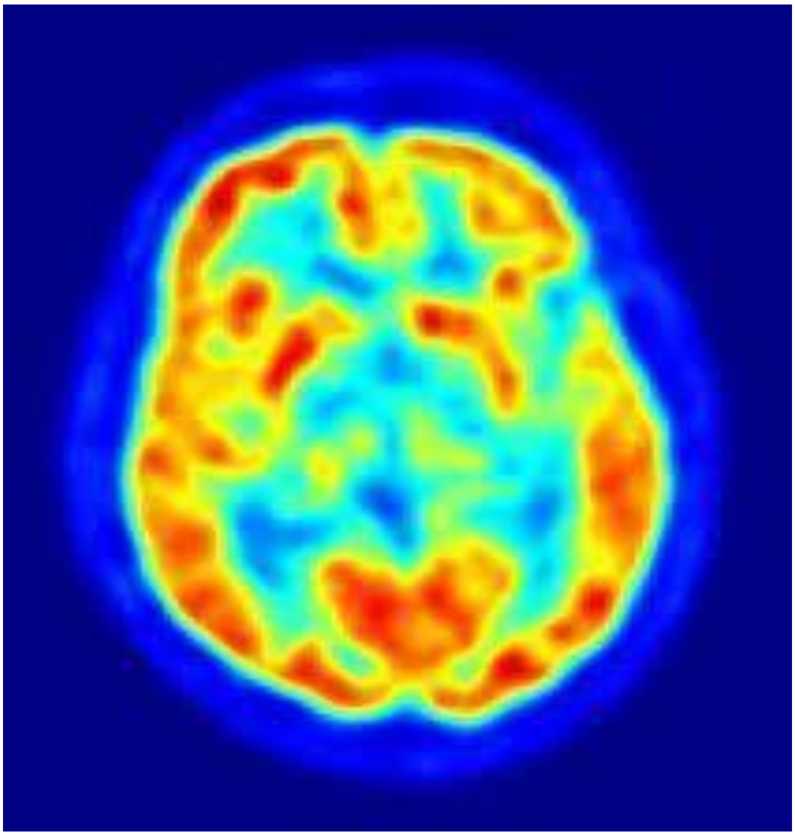


Figure I.3: Positron Emission Tomography. [35]

I.3.4. Microscopy images:

Microscopy images are visual representations of objects or samples that have been magnified and observed using a microscope. They provide detailed views of tiny structures, particles, or materials that are otherwise too small to be seen by the human eye. Microscopy images are obtained by capturing the interaction of light or electrons with the sample, allowing scientists and researchers to study and analyze its characteristics, such as shape, size, texture, and composition. These images are instrumental in various scientific disciplines for understanding and advancing knowledge in fields like biology, medicine, materials science, and nanotechnology.[13]

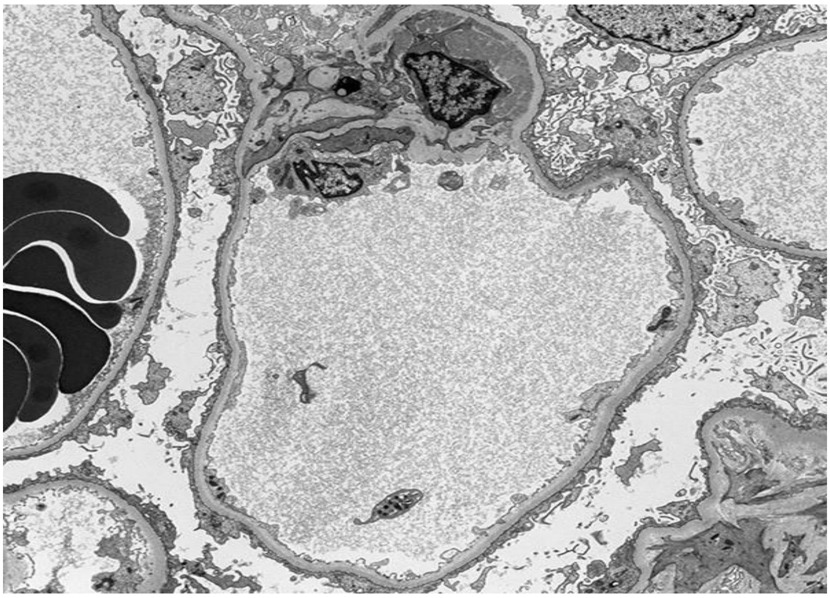


Figure I. 4: Electron Microscopy. [36]

I.3.5. Capsule image:

Image capsules are a type of deep learning model used for image recognition tasks. They are an alternative to traditional CNNs and aim to capture more detailed and context-aware information from images. Image capsules represent specific entities or concepts in an image and encode both their presence and properties. By considering the relationships between capsules, image capsules provide a more nuanced representation of objects and their spatial relationships, leading to improved image recognition capabilities.[14]

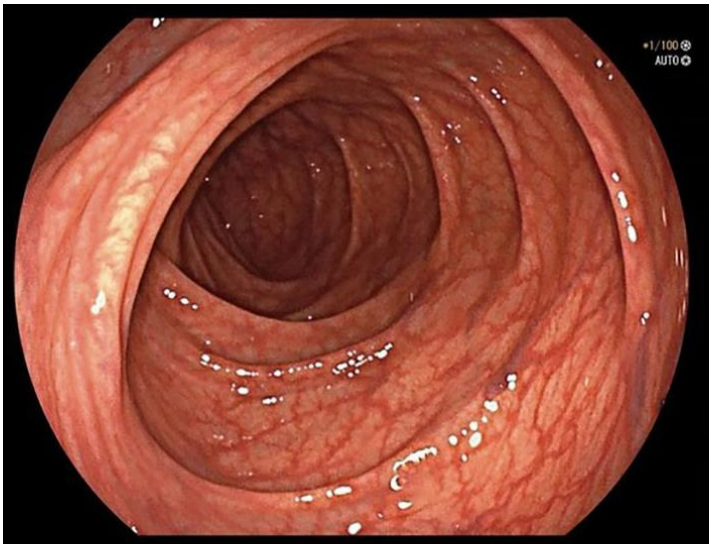


Figure I. 5: endoscopy image. [40]

I.4. Conventional SR techniques:

I.4.1. Bicubic interpolation:

Bicubic interpolation is a method of image interpolation used to estimate the values of new pixels in an image. This method involves fitting a cubic polynomial function to a 4x4 pixels neighborhood and using it to estimate the value of the pixel at a specified location. Bicubic interpolation can produce smoother and more accurate results than simpler interpolation methods, but it can also be more computationally expensive. It is commonly used in digital image processing, computer graphics and other applications where images need to be resized or transformed. Bicubic interpolation can help preserve the quality of an image when it is enlarged or reduced in size.

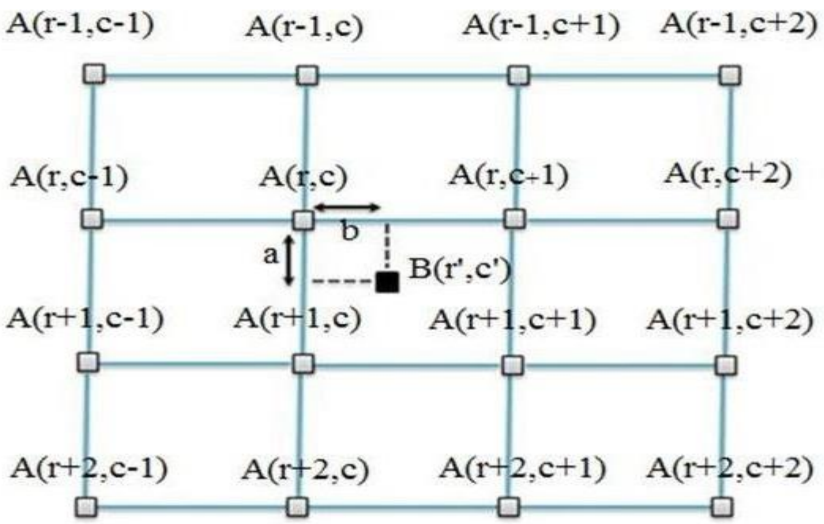


Figure I.6: Diagram for Bicubic Interpolation [15]

I.4.2. Linear interpolation:

Linear interpolation is a method of estimating a value that falls between two known values by assuming a linear relationship between the known values. In other words, it is the process of finding a value that lies on a straight line between two known values.

For example, suppose we have two data points $(x1, y1)$ and $(x2, y2)$ and we want to estimate the value of y at some point x that lies between $x1$ and $x2$. We can use linear interpolation to estimate y by assuming that the relationship between x and y is linear between $x1$ and $x2$.

The formula for linear interpolation is:

$$y = y1 + (x - x1) * (y2 - y1) / (x2 - x1) \quad \textbf{(I.1)}$$

Where, x is the value we want to estimate, y is the estimated value of y at x, $x1$ and $y1$ are the values of the first data point and $x2$ and $y2$ are the values of the second data point.
This formula calculates the value of y at x by finding the slope of the line between the two data points and using it to estimate the value of y at x. The slope of the line is given $by\,(y2 - y1) / (x2 - x1)$, and the value of y at x is then found by adding the offset from $y1$ to the slope times the offset from $x1$ to x.

I.4.3. Nearest neighbor interpolation:

Nearest neighbor interpolation is a method used to resize or resample an image. In this method, the value of each pixel in the output image is determined by the value of the closest pixel in the input image. This method is called "nearest neighbor" because it simply selects the pixel that is nearest to the desired location.

For example, let's say we want to increase the size of a 2x2 image to a 4x4 image using nearest neighbor interpolation. The new image would look like this:

In this Figure 9, each pixel in the new image is simply copied from the nearest pixel in the original image. The result is an image that has been scaled up without adding any new information.

Nearest neighbor interpolation is a simple method that is easy to implement, but it can result in a loss of image quality and a "blocky" appearance. It is generally not the best method to use for image resizing, but it can be useful in some specific applications where speed and simplicity are more important than image quality.

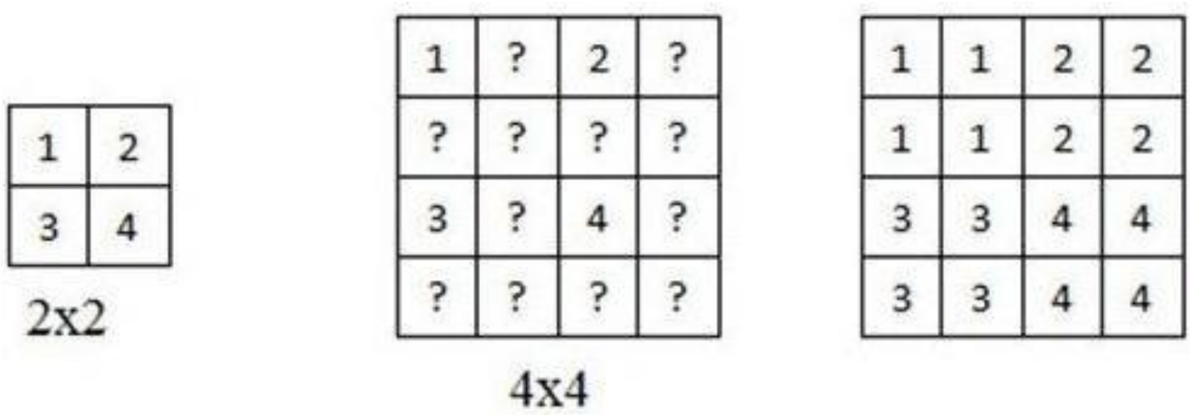


Figure I.7: Nearest Neighbor algorithms [15].

I.5. super resolution with deep learning:

Deep learning and SR have a symbiotic relationship. Deep learning techniques, particularly CNNs, have revolutionized the field of super resolution. By training CNN models on pairs of LR and high-resolution images, these models can learn to generate high quality, detailed outputs from low quality inputs. The hierarchical nature of deep learning allows the models to capture complex patterns and relationships in the data, resulting in visually appealing super resolved images. On the other hand, SR serves as a crucial application for deep learning, showcasing its ability to enhance image quality and extract fine details. Together, deep learning and SR have propelled advancements in image enhancement, offering practical solutions for various domains and applications.

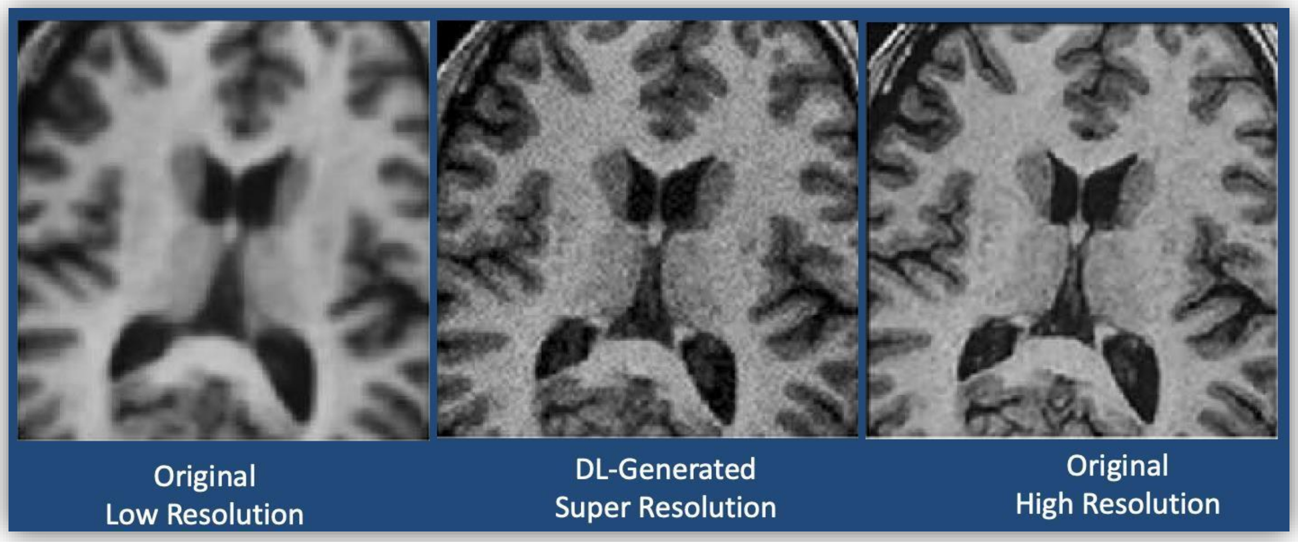


Figure I.8: Super resolution image. **[40]**

I.6. Conclusion:

In this chapter, we have presented the fundamental concepts of the image and the digital image and the color image, the different structures and the different forms of the image. We have presented the operation, the different architectures and the training algorithms as well as the classic techniques of SR. In addition ,we have defined the medical image, and Some areas of its use

The next chapter will detail a type of neural network which is the CNN and we will expose the basic rules on which it works, that the "CNN" architecture, this network will serve as the basis for our study to reduce noise in medical images.

ChapterII: Deep learning

Abstract:
This chapter provides an introduction to neural networks, deep learning techniques, optimization methods, and the role of convolutional neural networks (CNNs). It also introduces ensemble learning, which combines multiple models for improved predictions.

Summary

II.1. Introduction:

Deep learning is a subfield of machine learning that uses artificial neural networks to model and solve complex problems [16]. These neural networks consist of multiple layers of interconnected nodes, which are trained on large datasets to learn patterns and relationships in the data. Deep learning has made significant advances in a wide range of applications, such as computer vision [17], natural language processing [18], speech recognition [19], and robotics[20].Some of the most well-known applications of deep learning include image and object recognition, language translation, and self-driving cars.

Deep learning requires a large amount of data to be effective, as well as significant computational resources for training the models. However, the advances in hardware and software technologies have made it more accessible and widely used in various industries, including healthcare, finance, and marketing. Overall, deep learning has shown great promise in solving complex problems and advancing our understanding of artificial intelligence.

In summary, deep learning is a broad term that encompasses many different techniques and models, including generative adversarial network (GAN) and CNN. GAN are a type of deep learning model that can generate new data samples, while CNNs are a type of neural network that is commonly used for image analysis in deep learning and essentially based on convolutional layer, as we will see in this chapter. For the sake of clarity, we will first introduce neural network architecture.

II.2. Principal parts of Neural networks:

Neural networks, also known as artificial neural networks, are a type of machine learning model inspired by the structure and function of the human brain [21]. They are designed to recognize patterns in data and make predictions or decisions based on that data. At their core, neural networks consist of interconnected nodes or neurons, which are organized into layers. The process begins by inputting data into the first layer, which then passes this information through a series of interconnected processing nodes. These nodes are linked together by weighted connections, allowing for the integration and transformation of the data. Each node in the network receives input from the nodes in the previous layer and generates an output signal that is transmitted to the nodes in the subsequent layer. To better understand the concept, refer to figure II.1, which illustrates the comparison between a biological neuron and an artificial neuron. In the biological neuron, dendrites collect excitations from the cell body, contributing to the generation of the final output signal.

During training, the weights of the connections between the neurons are adjusted to optimize the network's performance. This adjustment process is accomplished by comparing the output generated by the network to the desired output and employing an algorithm known as backpropagation. Backpropagation enables the network to iteratively adjust the weights in a manner that minimizes the error. This error, often referred to as the "loss function," quantifies the discrepancy between the predicted output and the desired output. Neural networks have been successfully applied to a wide range of tasks [22], including image recognition, speech recognition, natural language processing, and many others they are particularly useful for tasks where the input data is complex or difficult to define using conventional algorithms.

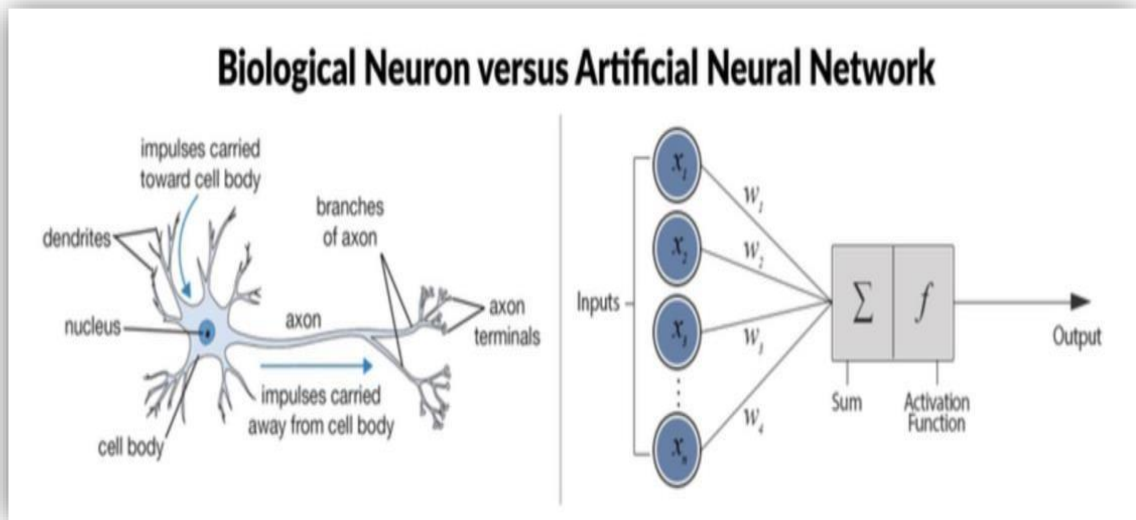


Figure II. 1: Biological neuron vs artificial neuron. [23]

II.3. Deep Learning Basic concepts:

Deep Learning is a machine learning technique. It teaches a machine to process the entries by layers in order to classify infer and predict the result as shown in figure II.2. Machine learning has the ability to learn through training with models. The end of this training gives the ability to distinguish things. In figure II.3, for example, the network tries to learn to differentiate things.

There is a distinct difference between them: “deep learning” is an area of artificial intelligence based on neural networks and requires no human interference to learn, as presented in figure II.2, however, machine learning is an area of artificial intelligence that requires human guidance, as illustrated in figure II.3.

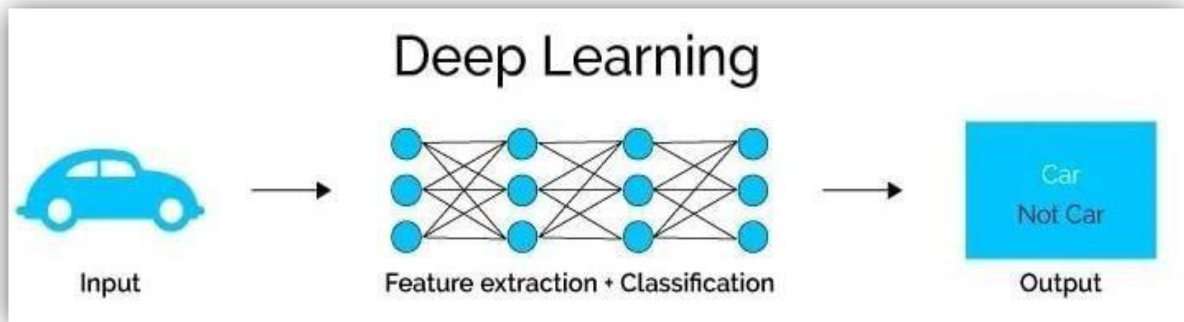


Figure II.2: Principle of deep learning [23]

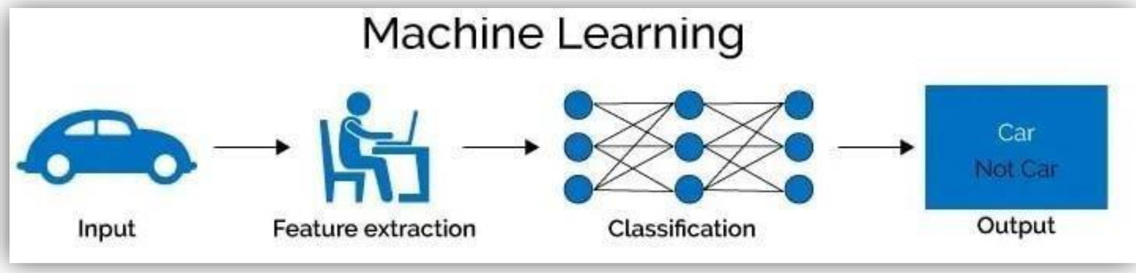


Figure II. 3: Principle of Machine Learning [23]

II.4. Optimization Algorithm Adam:

The Adam optimization algorithm optimizes stochastic functions using first-order gradients. In terms of big datasets and many parameters, it is a method that is easy to implement for any model. It is highly computationally efficient and requires less memory in terms of hardware resources. It also works well for sparse and noisy gradients for non-stationary goals and issues. Any successful model must be tuned, but the Adam optimization algorithm typically requires little tuning [24]. The Adaptive Moment Estimation (Adam) method maintains a single learning rate for all weight updates that do not change throughout the training while keeping adaptive learning rates distinct from each parameter [25]. Adam is also a combination of RMS propand Stochastic Gradient Descent (SGD), as Adam estimates the first and second moments of the gradient to balance the learning rate for each weight of the model network.

The figureII.4 illustrates the training cost of different optimizer algorithms versus the number of iterations for a specific data set.

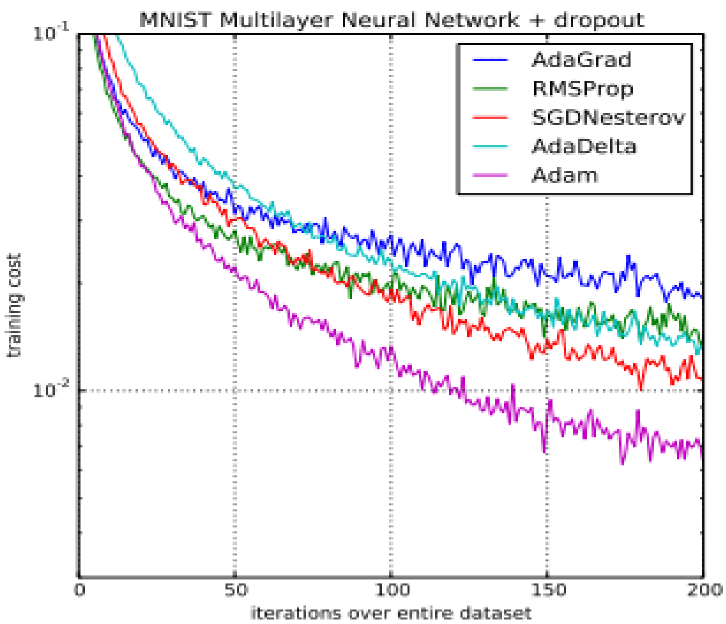


Figure II.4: Comparison of Training Cost for Different Optimizer Algorithms vs. Iterations on a Specific Dataset.**[39]**

The figureII.5 it is showing that Adam is performing well enough compared to other optimizer's algorithms. Then, Nadam came, which is even more promising result showing than Adam.

Despite Adam promising the result still after a while, researchers noticed that it does not converge to an optimal solution such as image classification on well-known CIFAR datasets. On the CIFAR dataset, state of the art results was achieved by the SGD with momentum. Though it has few drawbacks, researchers are still using Adam optimizer, and its popularity is growing day by day on par with SGD with momentum [26].

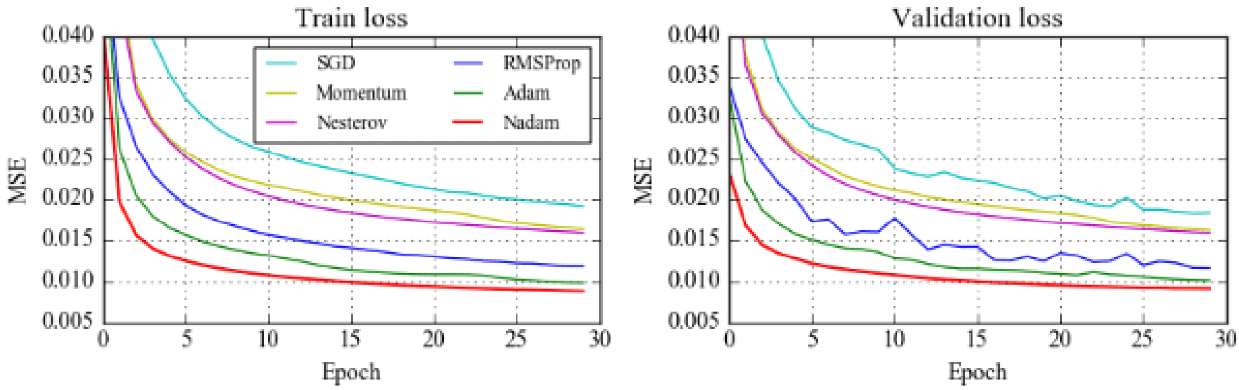


Figure II.5: Performance Comparison of Optimizer Algorithms: Adam vs. Nadam. **[42]**

II.5. Convolutional Neural Networks (CNN):

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for analyzing visual data, such as images and videos. CNNs have revolutionized various fields, including computer vision, image recognition, object detection, and medical imaging.

The structure and functionality of CNNs are inspired by the organization of the visual cortex in the human brain. They consist of multiple layers, each performing specific operations to extract features from the input data. The primary layers in a CNN are:

- Convolution Layer
- Pooling Layer
- Fully Connected (FC) Layer

The convolutional layer is the top layer of a convolutional network. The fully connected layer is the final layer, though additional convolutional or pooling layers may be added after convolutional layers. With each layer, the CNN gets and can detect larger portions of the image. As the image data advances through the CNN layers, it begins to recognize larger elements or shapes of the object before ultimately identifying the intended object. Earlier (colors and edges), layers focus on basic elements. [27]

II.5.1. Convolutional Layer:

The input will be a color image that is composed of a 3D pixel matrix. As a result, the input will have three dimensions height, width, and depth that are analogous to RGB in an image. Additionally, a feature detector, also referred to a kernel or a filter, is used to determine whether a feature is present by moving across the image's receptive fields. Convolution describes this process [30]. figure II.6 shows an example of this layer with a patch 4×4 and a specified kernel (filter 3×3). The result of the convolution is also shown.

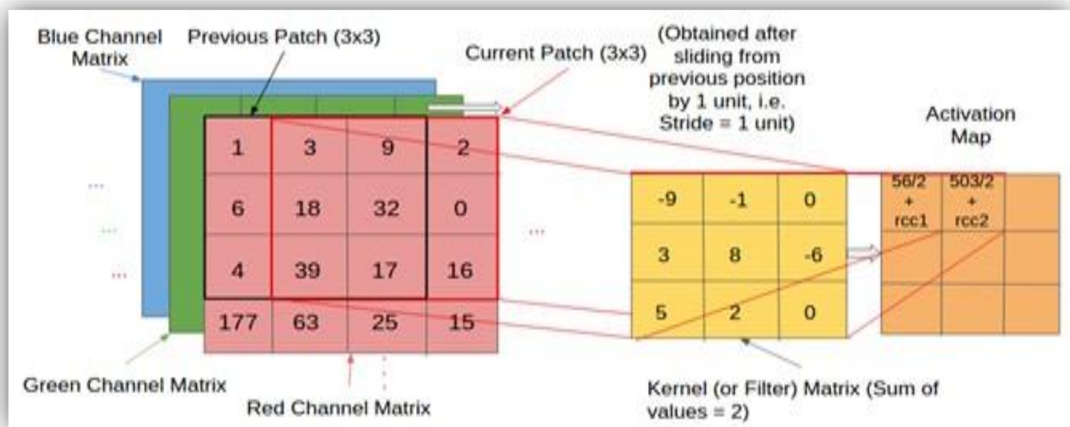


Figure II.6: Convolutional layer **[30]**

II.5.2.1. Feature detector:

The filter size, which also determines the size of the receptive field, is typically a 3x3 matrix, though they can vary in size.

- **Valid padding:**

This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.

- **Same padding:**

This padding ensures that the output layer has the same size as the input layer.

- **Full padding:**

This type of padding increases the size of the output by adding zeros to the border of the input

Figure II.7 show the different types of padding process used in CNN.

A CNN adds a Rectified Linear Unit (ReLU) adjustment to the feature map after each convolution operation, introducing nonlinearity to the model. [29]

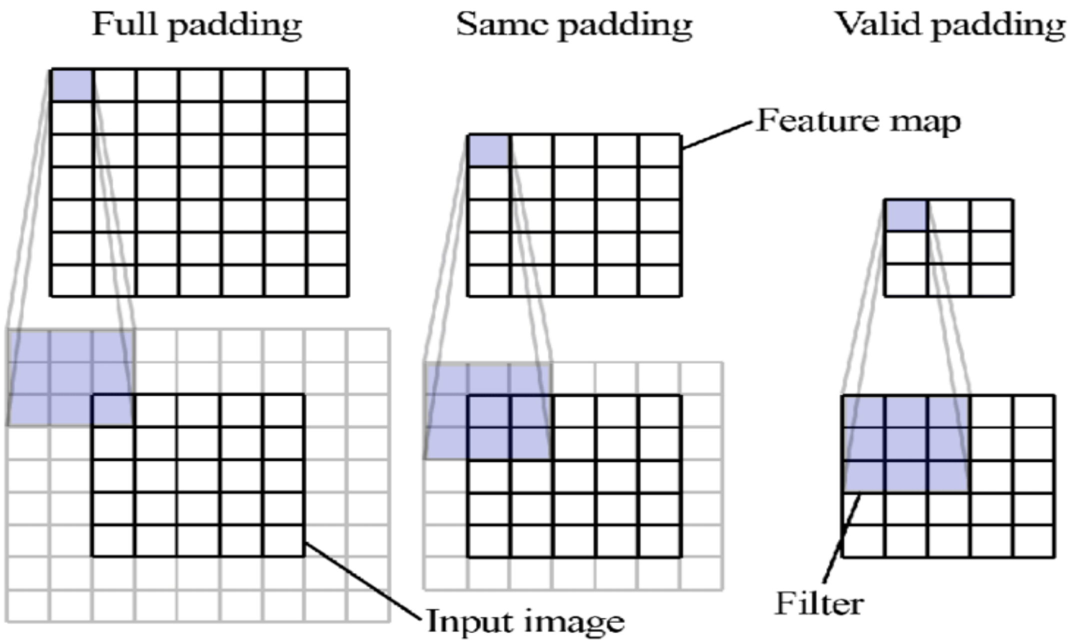


Figure II.7: Padding of a 2D image. [43]

II.5.2. Pooling Layer:

Down sampling, also referred to pooling layers, is a dimensionality reduction method that minimizes the amount of input factors. Similar to the convolutional layer, the pooling process sweeps a filter across all 13 inputs, but this filter lacks weights. Instead, the kernel populates the output array from the values in the receptive field using an aggregation function. Pooling can be performed through the computation of the maximum or the mean of a considered neighboring pixel.

- **Max pooling:**

As shown in figure II.7, as the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an, 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 [31].

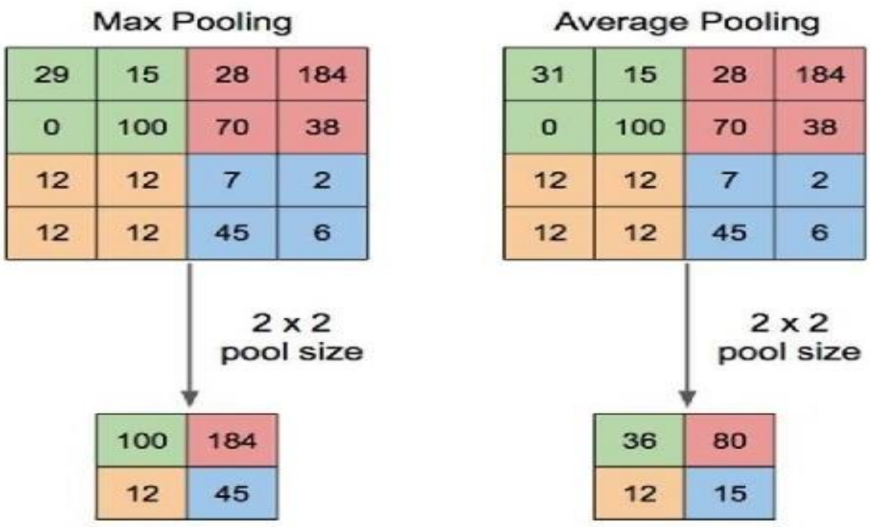


Figure II.8: Illustration of Max Pooling and Average Pooling [32]

II.5.3. Fully Connected (FC) Layer:

The name of the layer, full connected, is self-explanatory. The pixel values of the input image are not, as stated previously, directly connected to the output layer in partially linked layers. On the other hand, every node in the output layer has a direct connection to a node in the layer above it in the fully connected layer. Based on the features that were retrieved by the previous layers and their various filters, this layer performs classification tasks. FC layers typically use a soft max activation function to generate a probability ranging from 0 to 1, as opposed to convolutional and pooling layers, which typically use activation functions to classify inputs.

II.5.3.1. Choice of hyperparameters:

CNN uses more hyperparameters than a standard multilayer perceptron (MLP). Even though the rules usual for learning rates and regularization constants apply always, it is necessary to take into account the notions of number of filters, their form and the form of max pooling. [29]

II.6. Ensemble Learning:

Ensemble learning is a machine learning technique that involves combining multiple models to improve the overall performance and accuracy of the prediction.

The basic idea behind ensemble learning is that by combining the predictions of multiple models, the final prediction will be more accurate and robust than that of any individual model. This is based on the principle of the "wisdom of the crowd," where the collective prediction of a group of individuals is often more accurate than that of any individual.

Ensemble learning can be achieved in a variety of ways, such as:

- **Bagging (Bootstrap Aggregation):**

This involves creating multiple models, each trained on a random subset of the training data. The final prediction is then made by aggregating the predictions of all the models.

- **Boosting:**

This involves creating a sequence of models, each trained to improve upon the errors of the previous model. The final prediction is then made by aggregating the predictions of all the models.

- **Stacking:**

This involves training multiple models and then combining their predictions as inputs to a Meta model that makes the final prediction.

Ensemble learning has been shown to be an effective technique in improving the accuracy of machine learning models, particularly in complex tasks where no single model can achieve high accuracy on its own. **[34]**

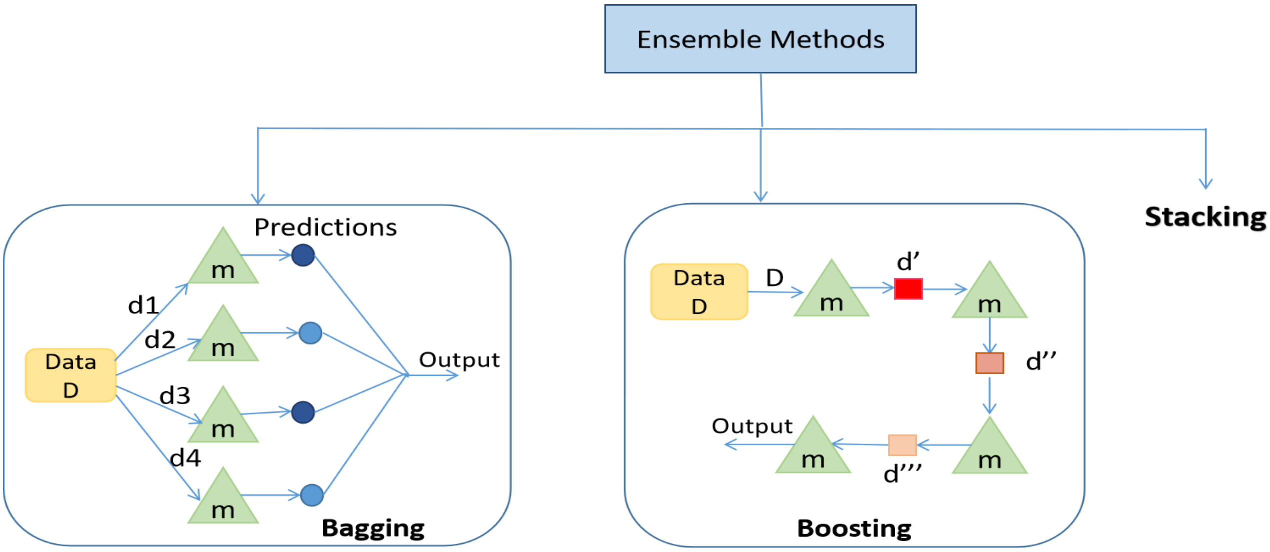


Figure II. 9: Representing the types of ensemble methods. Here, m represents a weak learner; d1, d2, d3, d4 are the random samples from Data D; d', d", d"' are updated training data based on the results from the previous weak learner. [37]

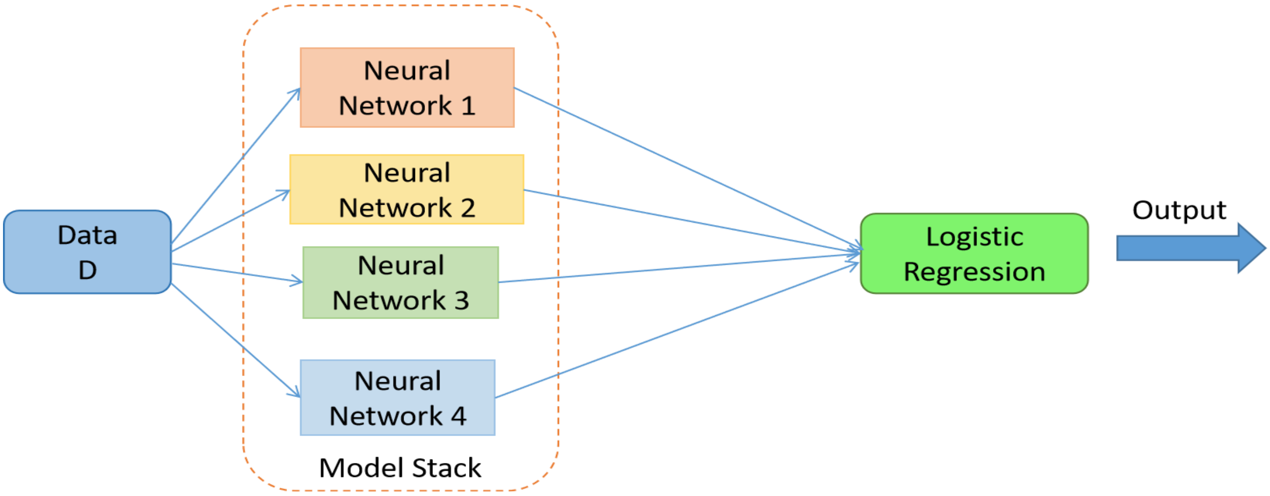


Figure II. 10: The stacked model with Meta learner = Logistic Regression and weak learners = 4 Neural Networks [37]

II.6. Conclusion:

In conclusion, our study has covered the fundamental aspects of neural networks, deep learning concepts, and optimization algorithms, with a specific focus on convolutional neural networks (CNNs) and their applications. By understanding these components and techniques, we have gained insights into the potential of neural networks in domains such as image recognition and computer vision. Moreover, the integration of ensemble learning techniques can further enhance the performance and robustness of CNN models. The knowledge acquired in this chapter equips us with a solid foundation to leverage the capabilities of CNNs and ensemble learning for various tasks, including image recognition and denoising.

In our study, we used the DNCNN model, it is a deep learning model specially designed for image denoising tasks. The DNCNN architecture uses the CNN to suppress the noise of the images, this is the principle of our research in the next chapter, and the simulation results will be explained and interpreted.

Chapter III: Implementation and results

Abstract:
This chapter focuses on the implementation of Ensemble Learning and denoiser CNN (DnCNN) these architectures in image super resolution (ISR) using specific data sets. We will describe the main steps of our simulation experiments. We will present and discuss the obtained results of the implemented methods, and concludes with remarks.

Summary

III.1. Introduction:

In this chapter, we will provide a detailed implementation of convolutional Super Resolution (SR) techniques, specifically focusing on DnCNN and ensemble learning. Our goal is to evaluate the performance of these methods in enhancing image resolution.

To conduct our experiments and assess the results, we use the capabilities of Google Colab, a cloud-based platform provided by Google. This platform allowed us to write and execute Python code through a browser-based interface. Additionally, we use the Keras library, which facilitated the implementation of deep learning models.

We first implemented convolutional SR techniques, which serve as the foundation for our study. These techniques involve using deep neural networks to upscale low-resolution images. We detailed the architecture of DnCNN, including the specific hyper parameters used for training the model. Subsequently, we trained the model using an appropriate dataset and analyzed the achieved results.

Next, we investigated ensemble learning, which involves combining multiple individual models to improve overall performance. We elucidated the process of constructing an ensemble, which encompassed selecting the conventional models, implementing diversification techniques, and employing fusion methods. In our implementation, we use the Stacking technique, where the predictions of individual models were used as input to a meta-learning model. Subsequently, we applied this ensemble approach to address our Super Resolution problem and meticulously evaluated the results achieved.

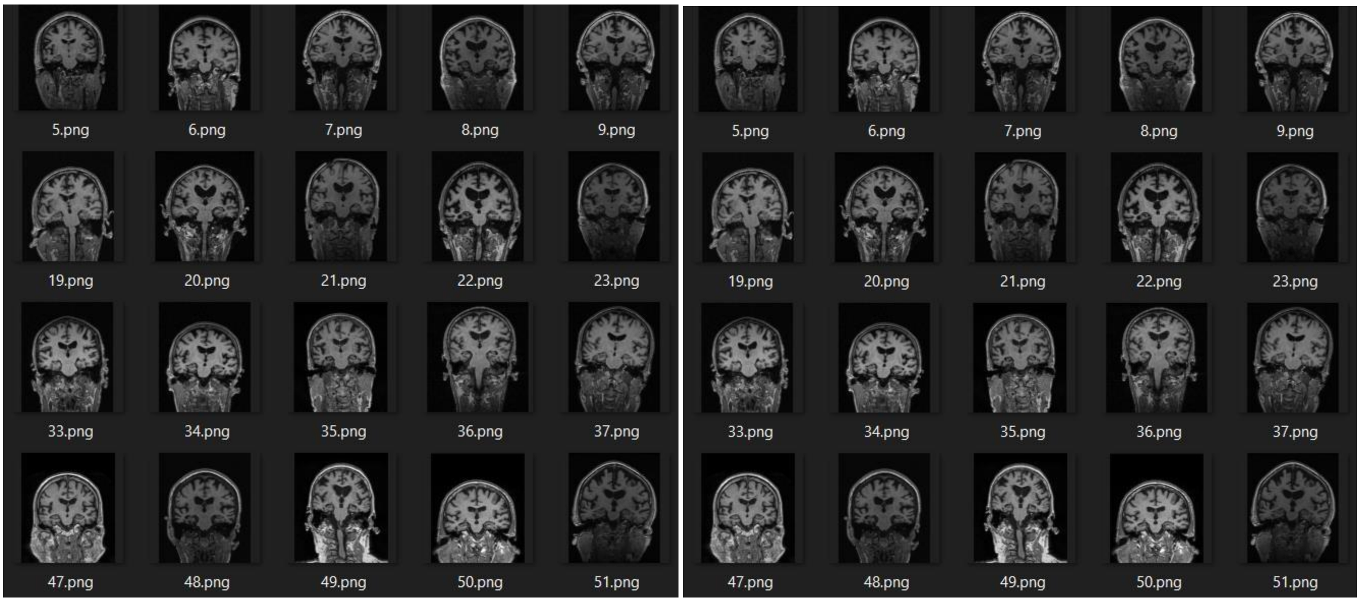
III.2 Used data sets:

For our machine learning experiments, we use a specific data set consisting of brain images in grayscale. The data set included a total of 1573 images, each with dimension of 256x256 pixels.

- **Training Data:** We used a subset of 1175 images (74.70%) from the data set for training our models. These images were used to teach the models to learn the underlying patterns and characteristics of the brain images.

- **Test Data:** For evaluating the performance and generalization ability of our models, we reserved a separate set of 398 images (25.30 %) from the data set as test data. These images were not used during the training phase and served as an independent evaluation set.

The use of this specific data set allowed us to focus on the task of Super Resolution, aiming to enhance the resolution and quality of the brain images. By training and testing our models on this data set, we were able to assess their performance in improving the visual clarity and details of the brain images.



(a) **(b)**

Figure III.1: (a) samples of training images and (b) samples of testing images.

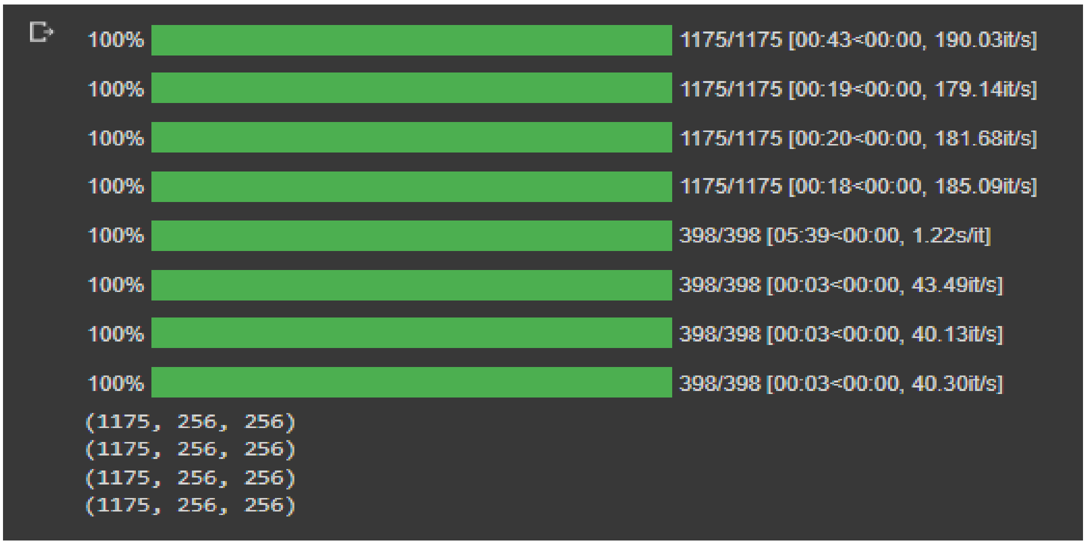


Figure III. 2: Image Loading and processing for super Resolution training and testing

III.3 Material and software:

For this project, we use Google Colab as our primary platform for conducting machine learning experiments. Google Colab offers several advantages that make it a convenient choice for machine learning projects.

III.3.1. Access to Powerful Hardware Resources:

Google Colab provides access to powerful hardware resources, including Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs). These hardware accelerators significantly speed up the training process for deep learning models.

We specifically use a GPU Tesla, which allowed us to leverage the parallel processing capabilities of the GPU to accelerate our computations. figure III.3 shows the details of the used GPU.

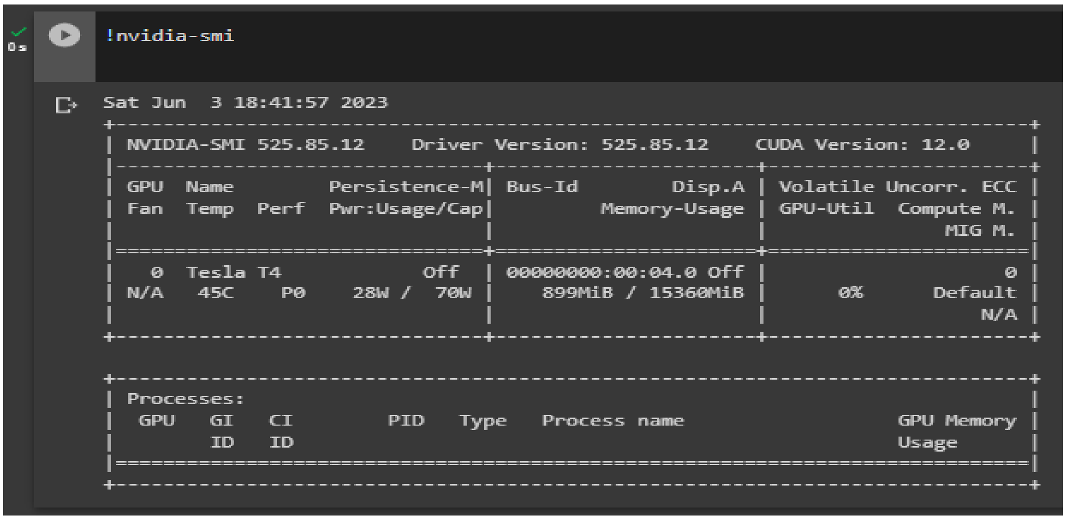


Figure III.3: NVIDIA GPU Monitoring

III.3.2. RAM and Disk Capacity:

In our experimentation, we had access to a system with 12.7 GB of RAM. During the training and evaluation processes, our implementation uses approximately 4.9 GB of the available RAM. Additionally, we had access to a disk with a total capacity of 78.2 GB. During the project, we use approximately 23.6 GB of disk space for storing datasets, models, and intermediate results. As shown in this figure III.4:

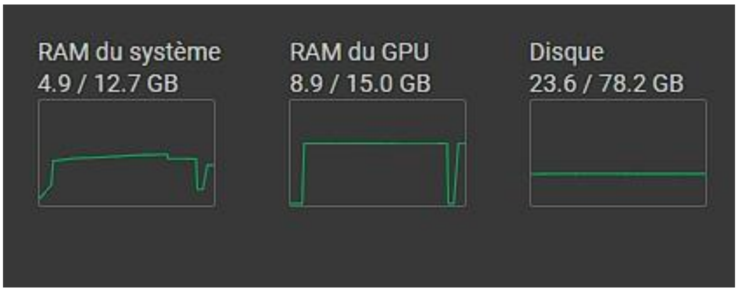


Figure III. 4: the current usage of system RAM, GPU RAM, and disk space, along with their respective maximum capacities.

Google Colab combination of powerful hardware resources, such as GPUs and TPUs, along with sufficient RAM and disk capacity, allowed us to efficiently perform our machine learning tasks without the need for setting up a local computing environment. These resources provided the computational power necessary for training complex deep learning models and conducting rigorous experiments.

III.4. Implementation:

Figure III.5illustrates the main steps of our ensemble learning process.

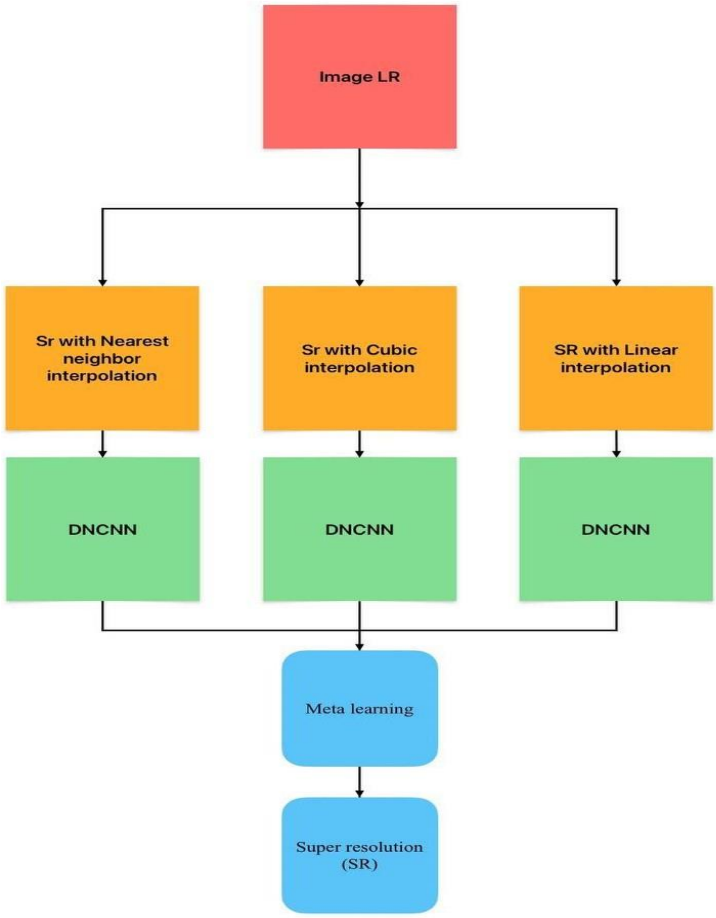


Figure III.5: schema of proposed work.

- **Down sampling of the High Resolution (HR) Image:**

The process begins by down sampling the original HR image. Down sampling reduces the resolution of the image, typically to a lower resolution version, to create a Low Resolution (LR) image. This step is done to simulate the effect of capturing images with lower quality cameras or to mimic the loss of detail in images due to compression or other factors.

- **Interpolation of the LR Image:**

After down sampling, the LR image is processed by three different interpolation methods: Cubic, Nearest Neighbor, and Linear. Interpolation is a technique used to estimate pixel values in between existing pixels to fill in the gaps and create a smoother image. Each method meets its own algorithm to estimate these values as shown in chapter I.

- **Denoising using the DNCNN:**

The three interpolated images are then fed through a denoising Convolutional Neural Network (DNCNN). The DNCNN is a deep learning model designed specifically to reduce noise and enhance image quality. It learns to remove artifacts and noise from images, which can occur during the down-sampling and interpolation steps.

- **Assembling with Ensemble Learning:**

Ensemble learning is a technique where multiple models are combined to make predictions. In this case, the outputs of the DNCNN for the three interpolated images are combined or assembled using ensemble learning. This can be done through various methods, such as averaging the pixel values or using a more sophisticated ensemble technique.

III.4.1. Implementation of DNCNN-Preparing the data sets Training step of the network:

- **Implementation of DNCNN:**

  As shown in figure III.6 for the implementation of DNCNN (Denoising Convolutional Neural Network), the following steps are taken:

- **Network Architecture:**

  The input image is processed through a convolutional layer, followed by a Rectified Linear Unit (ReLU) activation function. This initial step helps extract initial features from the input.

- **Loop Iteration:**

  A loop is initiated from **i= 1**to **i = 12**withineach iteration, the following operations are performed:

  a. Batch Normalization:

The output from the previous layer is subjected to batch normalization. This step helps normalize the activations, making the training process more stable.

b. ReLU Activation:

The batch-normalized output is then processed by through a ReLU activation function. This introduces non linearity and enhances the network's ability to learn complex representations.

c. Convolutional Layer:

Another convolutional layer is applied to the ReLU activated output. This layer performs filtering operations and extracts higher level features from the input.

- **Output Collection:**

Once the loop iterations are completed (i = 12), the output from the final convolutional layer is collected. This output represents the denoised image obtained from the DNCNN model.

- **Training Procedure:**

During the training phase, noisy images are fed into the DNCNN network. The output from the last convolutional layer is compared to the corresponding clean images using an appropriate loss function, such as mean squared error (MSE). Backpropagation is then used to update the network's weights and biases, optimizing them to minimize the loss and improve the denoising performance, as shown in figure III.6.

The implementation of DNCNN involves applying convolutional layers, batch normalization, and ReLU activation functions in a loop for 11 iterations. The final output obtained after the loop iterations represents the denoised image. The training process focuses on minimizing the difference between the network's output and the ground truth clean images. By iteratively updating the network's parameters, the DNCNN model learns to effectively remove noise from the input images.

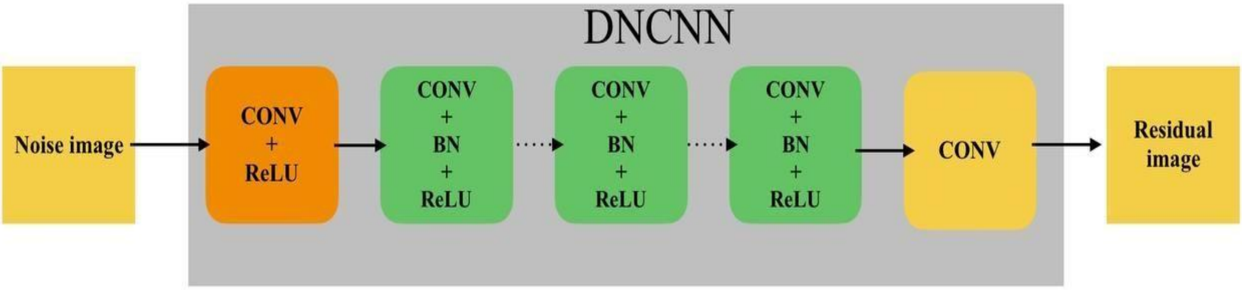


Figure III. 6: Architecture Overview of the Denoiser Convolutional Neural Network (DNCNN)

III.4.2. Implementation of Ensemble Learning:

As shown in the figure III.7in this section, we describe the details of ensemble learning implementation using concatenation and convolution in our project. The ensemble is constructed by combining three input images using concatenation and applying convolution operations. The overall process involves the following steps:

- **Input Preparation:**

We start by preparing the three input images that will be used in the ensemble. Each input image represents a different aspect or variation of the data, capturing diverse features.

- **Concatenation:**

The first two input images are concatenated together using an appropriate method, such as stacking or channel wise concatenation. This creates a fused image that combines the information from the first two inputs.

- **Concatenation with the Third Input:**

The fused image from step 2 is further concatenated with the third input image. This results in a final composite image that incorporates the information from all the three input sources.

- **Convolution:**

Convolutional operations are applied to the composite image obtained from step 3. This involves passing the image through one or more convolutional layers, which learn and extract relevant features from the concatenated input.

- **Ensemble Evaluation:**

The performance of the ensemble is evaluated using appropriate evaluation metrics, such as accuracy or error rates, depending on the specific task. This evaluation helps assess the effectiveness of the ensemble in capturing the combined information from the inputs.

- **Ensemble Refinement:**

If necessary, the ensemble can be refined by adjusting the architecture, hyperparameters, or weights of the convolutional layers. This refinement process aims to improve the overall performance and accuracy of the ensemble.

By implementing ensemble learning with concatenation and convolution, we aim to exploit the complementary information from multiple input sources and enhance the representation power of the ensemble. This approach allows us to capture and use different aspects of the data, leading to improved performance in various machine learning tasks.

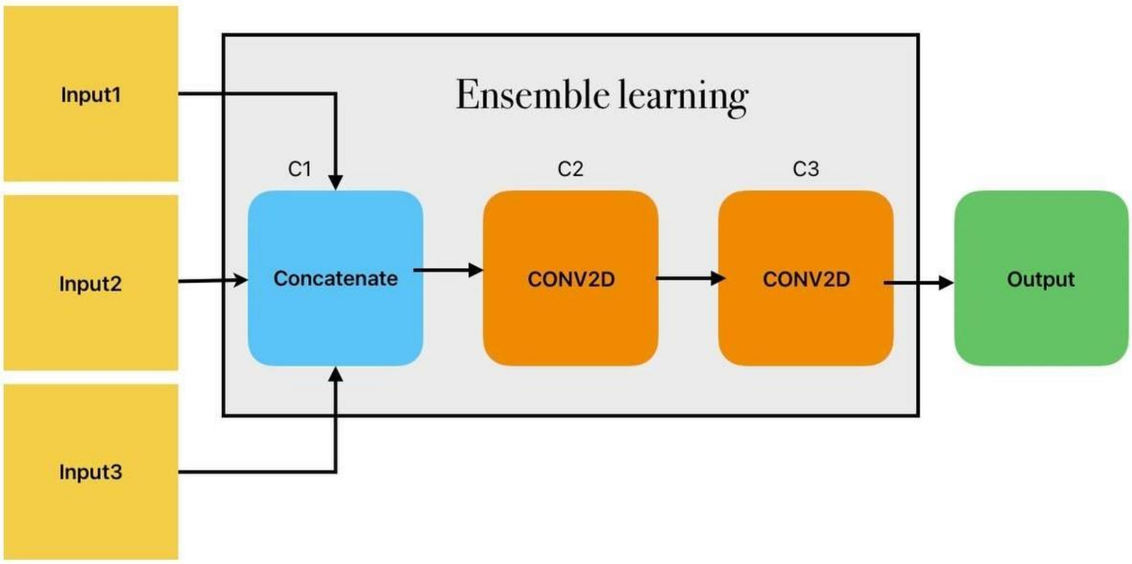


Figure III. 7: the proposed Ensemble Learning layers

III.5. Results and Discussion:

III.5.1Explanation of code operation:

The given program seems to be a Python script that performs image processing tasks using various libraries such as OpenCV, NumPy, TensorFlow, and Keras. Here is an explanation of the different sections and functions in the program:

1. Importing Libraries:

This section imports the necessary libraries for the program, including ***matplotlib, seaborn, pickle, pandas, numpy, cv2, skimage, tensorflow, PIL, and tqdm***.

2. Function Definitions:

The program defines several functions that will be used later. These functions include

- **Preprocessing data:** Import, resize and normalization of training images.
- **Batch Normalization:** Custom batch normalization layer implementation.

- **DNCNN:** Defines a deep neural network model for denoising images.
- **Ensemble:** Defines an ensemble model that combines the outputs of multiple models.

3. Data Preparation:

The program prepares the training and test image datasets by calling the previously defined functions with the appropriate arguments.

4. Training the DNCNN Model:

The program compiles and trains the DNCNN model using the training dataset and saves the model checkpoints after each iteration.

5. Ensemble Learning:

The program loads three different trained models (dncnn1, dncnn2, dncnn3) and predicts the images' outputs. Then it trains the ensemble model (Ensemble1) using the predicted outputs as inputs and the ground truth images as targets.

6. Image Visualization:

The program visualizes a set of images, including the ground truth image, low-resolution image, images generated using different interpolation methods (***SR_NEAREST_test_images, SR_LINEAR_test_images, SR_cubic_test_images*)**, and the output image generated by the ensemble model (predicted_image4).

- **Explication of the Figure III.8**

Figure III.8 illustrates the evolution of the loss function versus the number of epochs. It is evident that the loss decreases as the number of epochs increases. In the initial phase, the loss declines rapidly within the first 100 epochs, reaching an estimated value of 26. However, after this point, the decrease in loss becomes minimal, stabilizing at around 16 after 700 epochs. We stopped the training process after approximately 900 epochs, with each epoch taking between 89 to 109 seconds. The final loss achieved was 15.56. The batch size used during training was set to 8, indicating that 8 training examples were processed together in each iteration.

The learning rate used in the training process was 0.001, implying that the model's parameters were updated with relatively small increments during each training step. A smaller learning rate, such as 0.001, often results in a slower convergence rate. However, it can contribute to achieving a more precise and accurate solution.

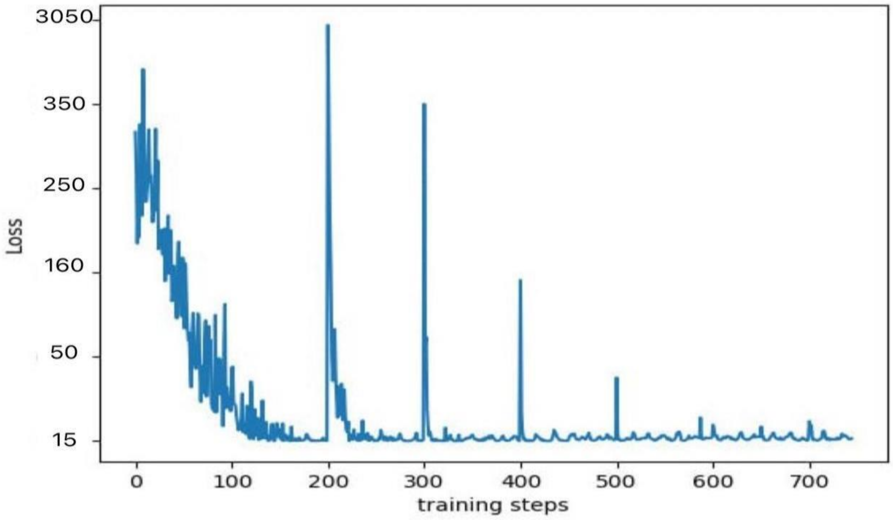


Figure III.8: Diminishing Loss over Training Steps (Epochs):
Visualizing Training Progress

Overall, this program performs image processing tasks such as image loading, resizing, interpolation, denoising process, and ensemble learning using deep neural networks.

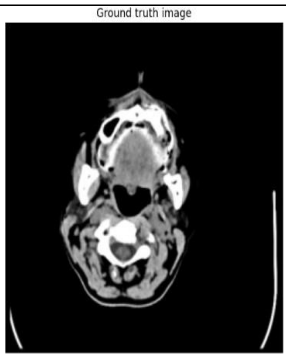
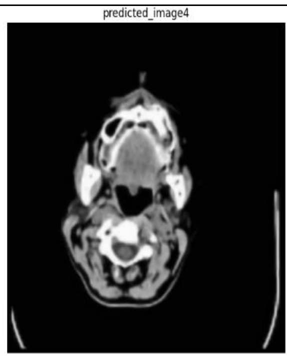
Images	Ground truth	LR	SR	PSNR (dB)
Image 1				**35.721**
Image 2				**37.643**

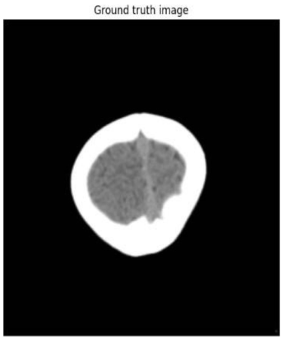
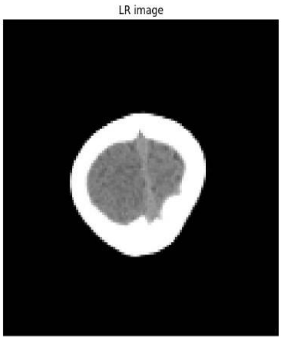
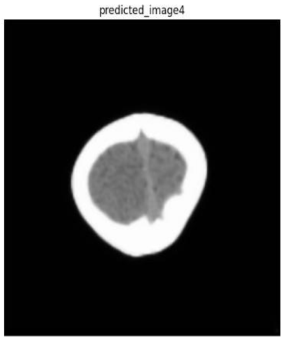
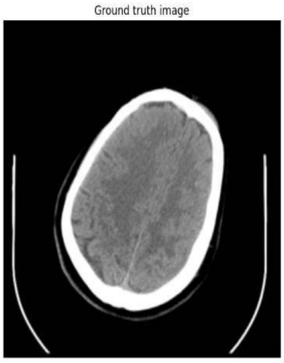
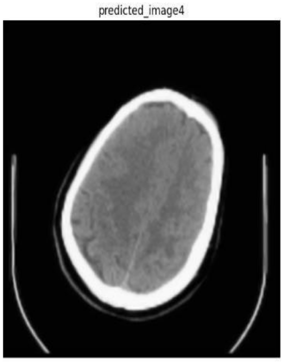
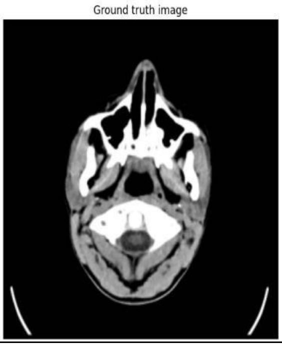
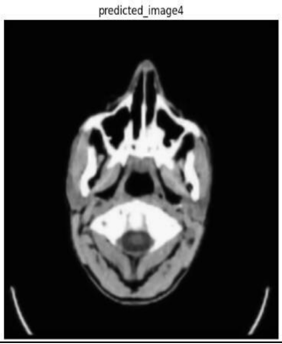
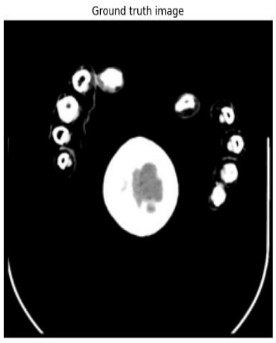
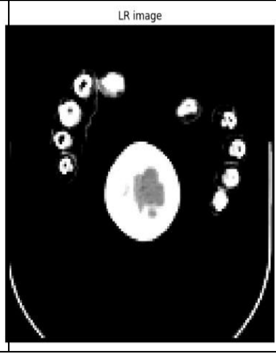
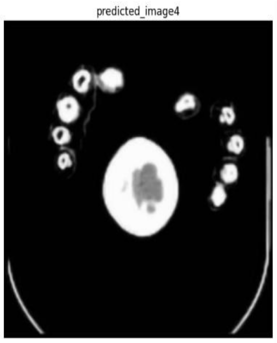
Image 3				**41.572**
Image 4				**35.649**
Image 5				**35.574**
Image 6				**35.525**
Image 7				**37.693**

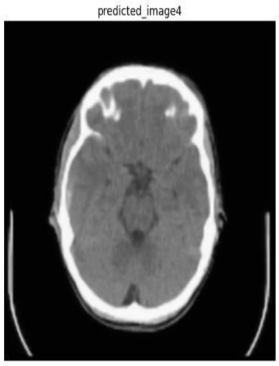
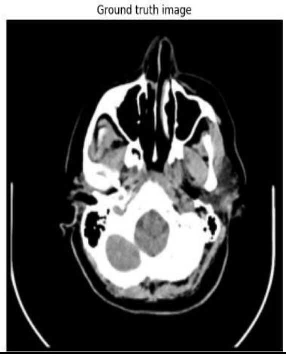
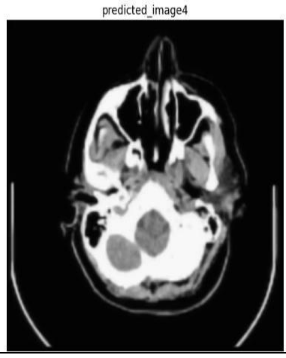
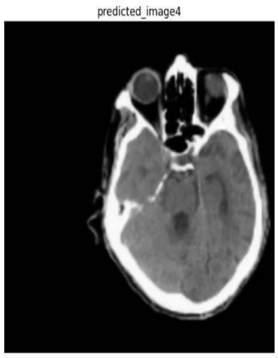
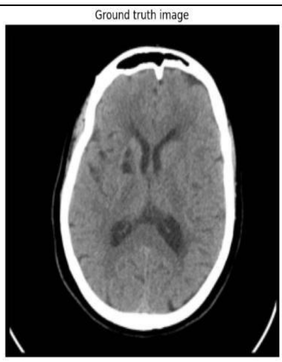
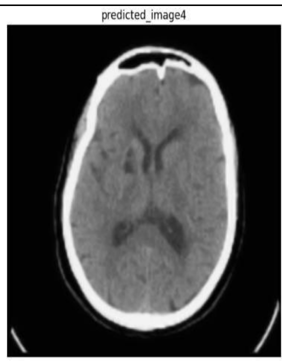
Image 8				**37.693**
Image 9				**34.094**
Image 10				**34.051**
Image 11				**35.544**
Image 12				**34.804**

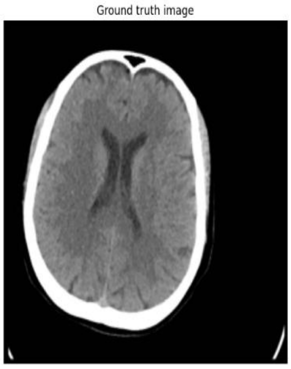
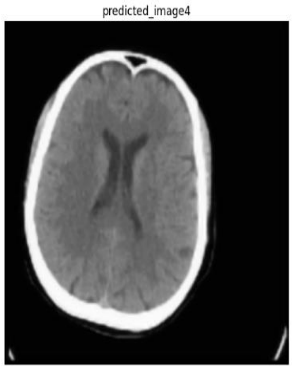
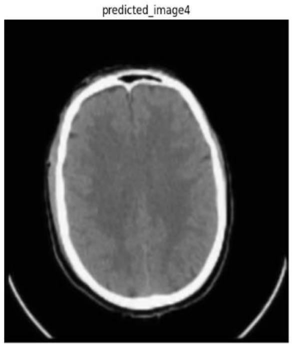
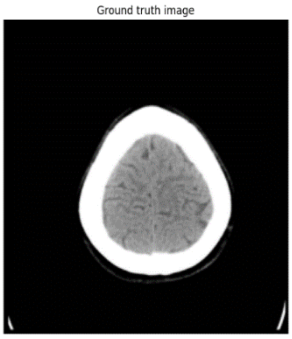
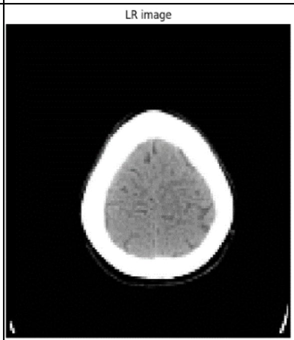
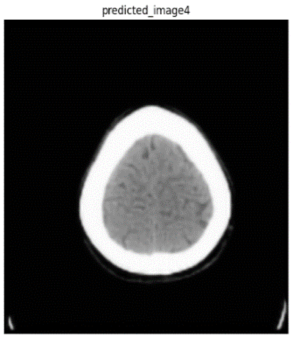
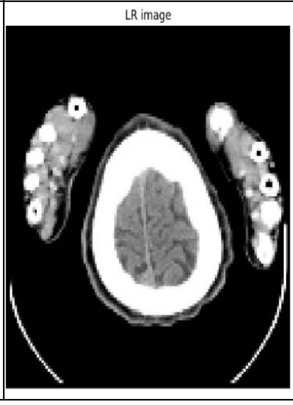
Image 13				**36.093**
Image 14				**35.845**
Image 15				**38.726**
Image 16				**34.798**

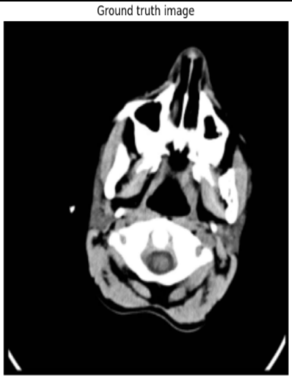
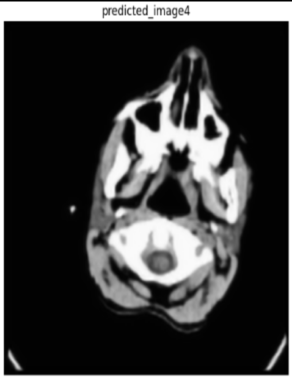
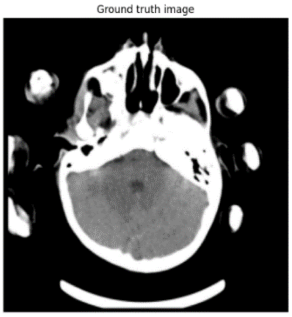
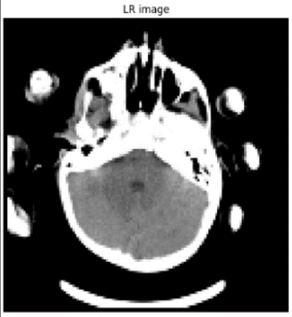
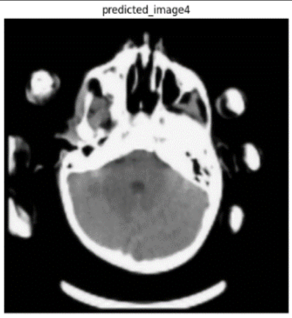
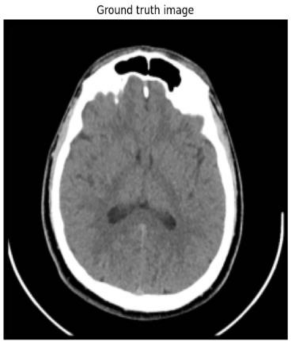
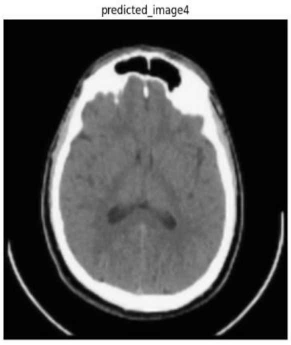
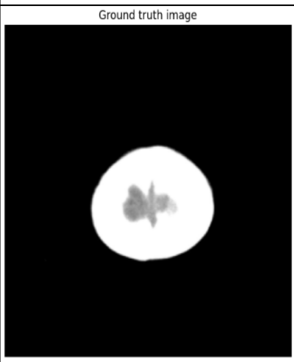
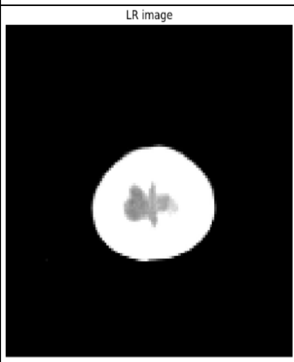
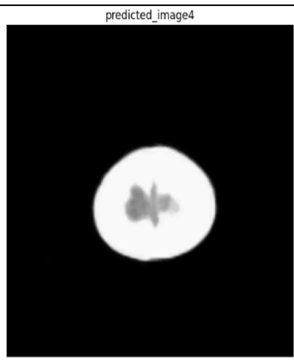
Image 17				**35.209**
Image 18				**35.145**
Image 19				**34.739**
Image 20				**34.488**
Image 21				**43.941**

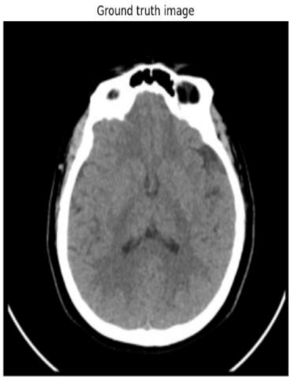
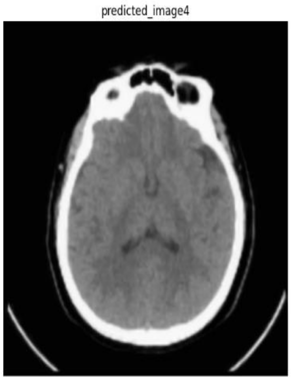
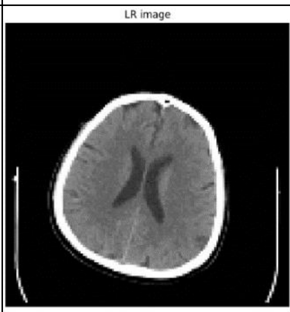
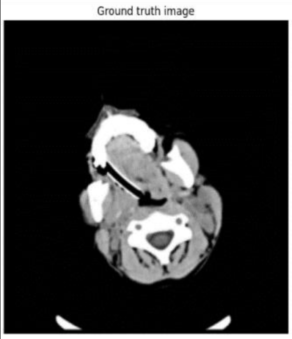
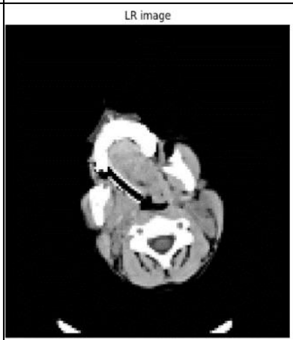
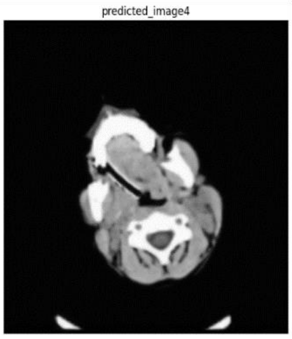
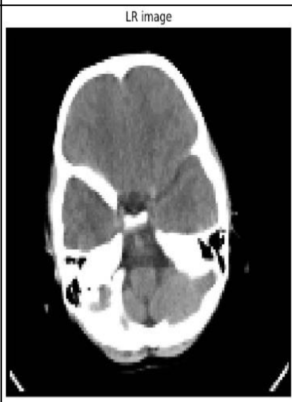
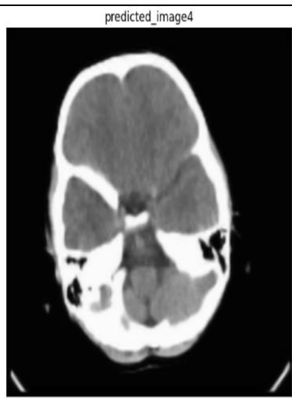
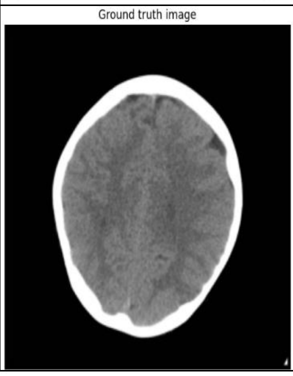
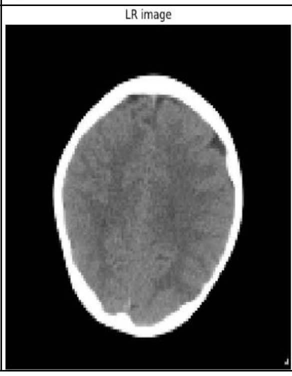
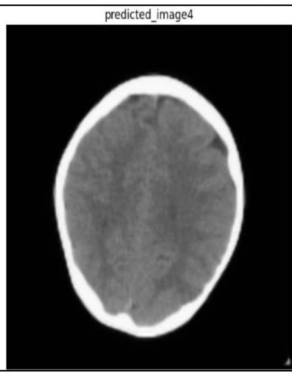
Image 22				**35.641**
Image 23				**36.336**
Image 24				**37.596**
Image 25				**36.355**
Image 26				**37.054**

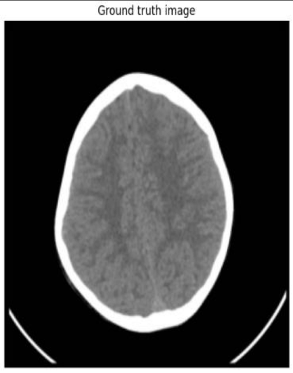
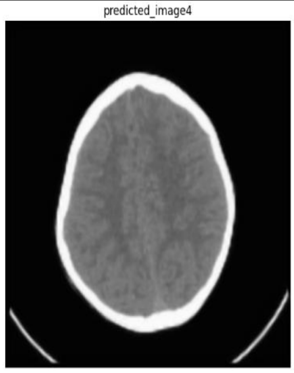
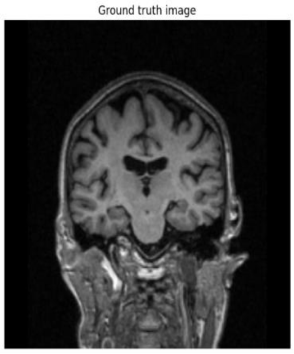
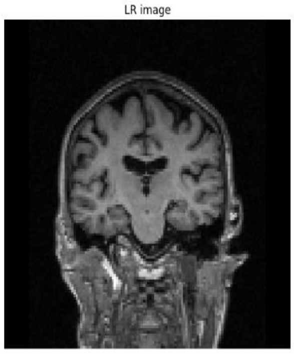
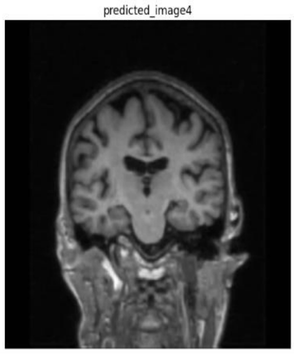
Image 27				**37.659**
Image 28				**34.702**

Figure III. 9: Comparative Analysis of Ground Truth, LR, and SR Images with PSNR Evaluation for 28 tests Images

"Figure III.9 represents a collection of 28 images (IRM) along with their corresponding PSNR (Peak Signal-to-Noise Ratio) values. The PSNR values indicate the quality or similarity of each image to its original version.

In this figure, we observe variations in PSNR values across the images. Some images have higher PSNR values, such as Image 3 with a PSNR of 41.572 and Image 21 with a PSNR of 43.941, indicating a closer resemblance to the original images. On the other hand, there are images with relatively lower PSNR values, such as Image 9 with a PSNR of 34.094 and Image 10 with a PSNR of 34.051, suggesting a higher level of distortion or difference from the original images.

By considering the PSNR values of each image, you can evaluate and compare their quality or fidelity. Adding these values to your memory will allow you to recall the specific PSNR values for future reference and analysis."

III.6. Conclusion:

In this program, we focused on image super resolution using deep learning techniques. We had a dataset of 1115 training images and 398 test images, initially downscaled to (128, 128, and 1). We applied three interpolation methods to resample the downscaled images. The denoising process was performed using a pre-trained deep neural network, the denoiser DNCNN. To further enhance the denoising performance, we have used ensemble learning techniques with a pre-trained ensemble model called Ensemble1. The denoised images were evaluated using peak signal to-noise ratio (PSNR). Despite the limitations of time and resources, the results were promising, highlighting the potential for further advancements in image denoising using deep learning.

General Conclusion:

As a general conclusion, we will try to establish a global synthesis of the work that was carried out in this thesis.

The field of image super resolution aims to enhance the resolution and quality of low-resolution images, enabling better visual interpretation and analysis. In various domains, including medicine, remote sensing, surveillance, and photography, there is a continuous demand for high-quality images. However, factors such as limited sensor capabilities, data transmission constraints, or hardware limitations can result in images with low resolution.

Image super resolution techniques offer a solution to this problem by utilizing advanced algorithms, particularly those based on deep learning. These techniques leverage the power of neural networks to learn intricate patterns and features from large datasets of high-resolution images. By training on pairs of low-resolution and corresponding high-resolution images, the models can effectively restore missing details and generate high resolution versions of the input images.

In first chapter, we delve into the fundamentals of image super resolution. We begin by discussing the different types of noise that can corrupt images, including Gaussian noise, salt-and-pepper noise, Poisson noise, and speckle noise. Understanding these noise types is crucial for accurately analyzing and addressing the challenges associated with low-resolution images.

Furthermore, we explore various medical imaging modalities, such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Positron Emission Tomography (PET), microscopy images, and capsule images. These modalities have unique characteristics and specific challenges when it comes to super resolution.

Additionally, we delve into conventional super resolution techniques, such as bicubic interpolation, linear interpolation, and nearest neighbor interpolation. While these techniques have been widely used, their limitations become evident when dealing with complex image structures and intricate details.

In the second, to overcome the limitations of conventional techniques, we focus on super resolution with deep learning. Deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized image processing and analysis. We discuss the core concepts of deep learning, including the principal parts of neural networks and optimization algorithms such as Adam. Furthermore, we dive into the specific layers of CNNs, such as convolutional layers, pooling layers, and fully connected layers. Additionally, we explore Recurrent Neural Networks (RNNs) and their applications in image super resolution.

In the last chapter, we shift our focus towards the implementation of super resolution architecture using neural networks. We discuss the datasets used for training and evaluation, the required materials and software, and the step-by-step implementation of deep learning models, such as DnCNN. Furthermore, we explore the concept of ensemble learning, which involves combining multiple models to enhance the overall performance and robustness of the super resolution system.

Finally, we conclude this chapter with some closing remarks and insights into the potential future developments and applications of image super resolution techniques. By the end of this thesis, readers will have gained a comprehensive understanding of image super resolution and the underlying deep learning methodologies used to achieve high-quality, high-resolution images.

References:

[1] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3365-3387, 2020.

[2] P. Milanfar (Ed.), "Super-resolution imaging," CRC Press, 2017.

[3] K. Suzuki, "Overview of deep learning in medical imaging," in Radiological Physics and Technology, vol. 10, no. 3, pp. 257-273, 2017.

[4] F. Lin, C. Fookes, V. Chandran, and S. Sridharan, "Investigation into optical flow super-resolution for surveillance applications," in WDIC 2005 : APRS Workshop on Digital Image Computing: Workshop Proceedings, University of QLD, 2005, pp. 73-78.

[5] L. M. Fonseca and M. H. Costa, "Automatic registration of satellite images," in Proceedings X Brazilian Symposium on Computer Graphics and Image Processing, IEEE, 1997, pp. 219-226.

[6] N. A. Dodgson, "Quadratic interpolation for image resampling," in IEEE Transactions on Image Processing, vol. 6, no. 9, pp. 1322-1326, 1997.

[7] A. K. Boyat and B. K. Joshi, "A review paper: noise models in digital image processing," arXiv preprint arXiv:1505.03489, 2015.

[8] T. G. Leighton, "What is ultrasound?," in Progress in Biophysics and Molecular Biology, vol. 93, no. 1-3, pp. 3-83, 2007.

[9] Z. P. Liang and P. C. Lauterbur, "Principles of Magnetic Resonance Imaging," Advances in Electronics and Electron Physics, 2000.

[10] J. Hsieh, "Computed Tomography: Principles, Design, Artifacts, and Recent Advances," 2nd edition, SPIE Press, 2009.

[11] Z. Cheng, H. Zhu, and S. S. Gambhi, "Principles and Applications of Positron Emission Tomography Imaging in Clinical," Molecular Imaging and Biology, Feb. 2018.

[12] R. L. Eisenberg and H. Hricak, "Medical Imaging," Radiology, May 2012.

[13] J. Smith, A. Johnson, and C. Brown, "Advances in Microscopy Techniques for Imaging Cellular Structures and Functions," Journal of Cell Biology, 2020.

[14] G. E. Hinton, S. Sabour, and N. Frosst, "Dynamic Routing Between Capsules," in Advances in Neural Information Processing Systems, 2017.

[15] A. K. Badrul hisham Mohamad, S. N. Yaakob, R. A. A. Raof, and M. W. Nasrudin, "An Analysis of Performance for Commonly Used Interpolation Method," in 3rd International Conference on Computer, Communication and Control Technology, April 2016. [Online]. Available: https://www.researchgate.net/publication/308980343_An_Analysis_of_Performance_for_Commonly_Used_Interpolation_Method

[16] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

[17] R. Szeliski, "Computer Vision : Algorithms and Applications," Springer, 2010.

[18] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition," 3rd edition, Pearson, 2019.

[19] D. Yu and L. Deng, "Automatic Speech Recognition : A Deep Learning Approach," Springer, 2015.

[20] J. J. Craig, "Introduction to Robotics: Mechanics and Control," 4th edition, Pearson, 2019.

[21] L. R. Medsker and L. C. Jain, "Recurrent neural networks," Design and Applications, vol. 5, pp. 64-67, 2001.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[23] A. M. Elzine and A. Belferoum, "Débruitage d'images médicales à faible SNR a l'aide de Réseau CycleGAN," Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj, 2022.

[24] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," arXiv preprint arXiv:2003.05991, 2020.

[25] P. Sharma, "A Comprehensive Tutorial to learn Convolutional Neural using in," Analytics Vidhya.

[26] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

[27] A. Lamia and N. H. Rouabeh, "Study of Image Super Resolution Algorithms," University of Mohamed El-Bashir El-Ibrahimi, 2022.

[28] Y. WaqarAhmad, "A new generative adversarial network for medical images super," Scientific Reports, [Online]. Available: https://www.nature.com/articles/s41598-022-13658-4

[29] S. Muhamad Yani, "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail," Journal of Physics Conference Series, May 2019.

[30] M. A. Djaballah, "Deep Learning based energy consumption prediction system," Université de 8 Mai 1945, Sept. 2021.

[31] P. J. Withers et al., "X-ray computed tomography," Nature Reviews Methods Primers, vol. 1, no. 1, p. 18, 2021.

[32] M. C. Levin, "Electron Microscopy," Stanford, [Online]. Available: https://web.stanford.edu/group/microscopy/cgi-bin/mediawiki/index.php/Electron_Microscopy

[33] Y. Khandelwal, "Ensemble Stacking for Machine Learning and Deep Learning," Analytics Vidhya, Aug. 2021, [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/ensemble-stacking-for-machine-learning-and-deep-learning/

[34] M. A. E. D. Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," South China University of Technology, City University of Hong Kong, Jun. 2018.

[35] Y. F. Tai and P. Piccini, "Applications of positron emission tomography (PET) in neurology," Journal of Neurology, Neurosurgery & Psychiatry, vol. 75, no. 5, pp. 669-676, 2004.

[36] N. Kambham, "Electron Microscopy," Stanford, [Online]. Available: https://web.stanford.edu/group/microscopy/cgi-bin/mediawiki/index.php/Electron_Microscopy

[37] Y. Khandelwal, "Ensemble Stacking for Machine Learning and Deep Learning," Aug. 2021, [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/ensemble-stacking-for-machine-learning-and-deep-learning/

[38] https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

[39]https://www.researchgate.net/publication/342424785_Predicting_Stock_Market_Prices_using_Deep_Learning_by_Tensor_Flow

[40] https://www.medimaging.net/general-imaging/articles/294790849/endoscope-visualization-system-eases-colonoscopy-procedures.html

[41] https://www.adam.com/

[42] https://openreview.net/pdf/OM0jvwB8jIp57ZJjtNEZ.pdf

[43] https://ai.stackexchange.com/questions/17004/does-each-filter-in-each-convolution-layer-create-a-new-image