PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

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

Faculty of Science and Technology

Department of Electronics

Thesis

Presented for the fulfillment of the MASTER degree

Field: Telecommunication

Specialty: Telecommunication Systems

By

Zitouni Neserine

Entitled

Study of Analytic vs Algebraic Image Tomography Reconstruction (Conventional and Deep Learning)

Supported on: 29/06/2025

Before the jury composed of:

Name	***Grade***	***Quality***	***Establishment***
Mrs. Samia LAIB	MCB	President	Univ-BBA
Mr. Abdellah BENREDOUANE	MCA	Examiner	Univ-BBA
Mrs. Messali Zoubeida	Pr	Supervisor	Univ-BBA
Ms. Soumia SID AHMED	MCA	Sub-Supervisor	Univ-BBA

School Year 2024/2025

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful. We extend our sincere gratitude to Allah, who blessed us with health, strength, patience, and determination to complete this work. His guidance and mercy have been the light that led us through every step of this journey.

We express our deep appreciation to our supervisor, Professor Messali Zoubida, for her remarkable efforts, continuous support, and unwavering dedication. Her guidance, availability, and valuable advice have played a fundamental role in the progress and success of this work. We are truly grateful for her commitment and the trust she placed in us throughout this experience.

We also sincerely thank our co-supervisor, **Dr. Sid Ahmed Soumia**, for her support and constructive guidance during this project. Her contributions and encouragement were greatly appreciated.

We are thankful to the honorable members of the examination committee for their time and thoughtful remarks, which enriched the quality of this work. We would also like to extend our gratitude to all the professors who have contributed to our academic development through their teaching and guidance.

Our thanks also go to all those who supported us, directly or indirectly, during the realization of this thesis. Special thanks to our friends, for their constant support, motivation, and encouragement. This accomplishment would not have been possible without the efforts and contributions of all those involved.

We pray that Allah rewards each one of you generously and accepts this work among our good deeds. Peace be upon you all.

Dedication

First and foremost,
To my father, Zitouni Abd-el-Kader, and my mother, Badia,
Words cannot fully express the depth of my gratitude and love.
You have been my first pillars, my models of courage, patience, and perseverance.
Your silent sacrifices, constant support, and unconditional love have shaped every step of my journey.
I love you deeply. This work, this path I dedicate it to you, from the bottom of my heart.

To my brother Sofiane, and my sisters Souhila and Ikram,
Thank you for your steady presence, your heartfelt encouragement, and your unwavering love.
You have been a quiet strength throughout this journey.

To my cousin Hamza,
Thank you for your invaluable help, your kindness, and your generosity.
Your support in crucial moments meant more than words can say. I am truly grateful.

To myself,
Thank you for not giving up. Thank you for believing, even when it was hard.
Thank you for holding on, for dreaming, for moving forward.
This thesis is more than just a project — it is a testament to willpower, growth, and a deep love for life.

To my dear friends: Houda Bechami, Sarah Remmache, Roumayssa Brahimi, Amel Boutaher, and Khadija Saoudi,
Thank you for the laughter, the memories we created, the kind words,

Table of Contents

List of Figures

List of Table

List of Acronyms

AI: Artificial Intelligence.

ART: Algebraic Reconstruction Technique.

BP: Back Projection.

CNN: Convolutional Neural Network.

CT: Computed Tomography.

$\boldsymbol{D_f}$: Difference Factor between Original and Reconstructed Images.

DL: Deep Learning.

$\boldsymbol{D_p}$: Deviation Proportion between Measured and Calculated Sinograms.

FBP: Filtered Back Projection.

FBP+DnCNN: Filtered Back Projection followed by a CNN denoising.

MLEM: Maximum Likelihood Expectation Maximization.

MLEM+DnCNN: Maximum Likelihood Expectation Maximization followed by CNN denoising.

ML: Machine Learning.

MSE: Mean Squared Error.

PSNR: Peak Signal-to-Noise Ratio.

PReLU: Parametric Rectified Linear Unit.

ReLU : Rectified Linear Unit function.

RT : Radon Transform.

SART : Simultaneous Algebraic Reconstruction Technique.

SSIM : Structural Similarity Index.

SIRT : Simultaneous Iterative Reconstruction Technique.

Abstract

Image reconstruction in tomography is a key challenge in medical imaging, especially in complex clinical situations involving incomplete data, noise, or low-dose acquisition. This work focuses on the study and implementation of the most efficient tomography reconstruction techniques based on a parallel beam model, ranging from conventional methods to modern approaches powered by artificial intelligence. Our contribution is the enhancement of the reconstructed images especially in noisy data, through the use of convolutional neural networks (CNN).

Two main categories were explored: analytical methods, represented by Back Projection (BP) and Filtered Back Projection (FBP), and iterative methods, including the Simultaneous Algebraic Reconstruction Technique (SART) and the Maximum Likelihood Expectation Maximization (MLEM) algorithm. To enhance reconstruction quality, these methods were extended with Convolutional Neural Networks (CNNs). Four hybrid architectures were developed: FBP + DnCNN, Learnable FBP, MLEM + DnCNN, and Learnable MLEM.

Experiments were conducted on both synthetic images and real medical scans, and evaluated in terms of quantitative metrics (MSE, PSNR, SSIM, D_f, D_p) in addition to visual quality. The obtained results demonstrated a significant improvement of hybrid models over conventional ones, particularly the Learnable MLEM, which yielded the most accurate reconstructions in terms of metrics and visual quality. Learnable schemes in both MLEM and FBP are more efficient than the combination with DnCNN.

This study highlights the relevance of CNN-based hybrid approaches in parallel-beam tomography, which in turn allow precision, robustness, and clinical useful reconstructions.

Keywords: image reconstruction, tomography, parallel beam, BP, FBP, SART, MLEM, CNN, deep learning, hybrid methods.

Résumé

La reconstruction d'images en tomographie constitue un enjeu majeur en imagerie médicale, notamment dans les contextes cliniques complexes (données incomplètes, bruit, faible dose). Ce mémoire porte sur l'étude et la mise en œuvre de techniques efficaces de reconstruction à partir de projections selon un modèle de faisceau parallèle, allant des méthodes classiques aux approches modernes basées sur l'intelligence artificielle.

Deux catégories principales sont analysées: les méthodes analytiques, représentées par la rétroprojection (BP) et la rétroprojection filtrée (FBP), et les méthodes itératives, illustrées par la Technique de reconstruction algébrique simultanée (SART) et l'algorithme Estimation du maximum de vraisemblance par maximisation (MLEM). Afin d'améliorer la qualité des reconstructions, l'intégration de réseaux de neurones convolutifs (CNN) est insérée entre les itérations des algorithmes conventionnels. Quatre architectures hybrides sont développées, à savoir: FBP + DnCNN, FBP apprenable, MLEM + DnCNN et MLEM apprenable. Les expériences, menées sur des images synthétiques et des images médicales réelles, sont évaluées en termes de métriques quantitatives (MSE, PSNR, SSIM,D_f, D_p) plus la qualité visuelle. Les résultats obtenus montrent une amélioration significative des modèles hybrides par rapport aux méthodes classiques, en particulier le modèle MLEM apprenable, qui a fourni les reconstructions les plus fidèles.

Ce travail met en évidence l'apport des approches hybrides basées sur les CNN dans le contexte de la tomographie à faisceau parallèle, ouvrant la voie à des reconstructions plus précises, robustes et adaptées aux exigences cliniques.

Mots-clés : reconstruction d'images, tomographie, faisceau parallèle, BP, FBP, SART, MLEM, CNN, apprentissage profond, méthodes hybrides

ملخص

تُعدّ إعادة بناء الصور في التصوير المقطعي من التحديات الأساسية في مجال التصوير الطبي، خصوصًا في الحالات السريرية المعقدة مثل البيانات غير المكتملة أو المشوشة أو ذات الجرعات المنخفضة. يركّز هذا العمل على دراسة وتطبيق عدة تقنيات لإعادة البناء اعتمادًا على نموذج الحزمة المتوازية، وذلك باستخدام طرق تقليدية وحديثة تعتمد على الذكاء الاصطناعي.

تم تحليل فئتين رئيسيتين: الطرق التحليلية، مثل الإسقاط العكسي (BP) والإسقاط العكسي المُرشح(FBP) ، والطرق التكرارية مثل خوارزمية SART وخوارزمية MLEM لتحسين جودة الصور المُعادة، تم دمج هذه الطرق مع الشبكات العصبية الالتفافية .(CNN) وقد تم تطوير أربع هياكل هجينةFBP + DnCNN ، FBP قابل للتعلّم، MLEM + DnCNN، و MLEMقابل للتعلّم ,أُجريت التجارب على صور تركيبية وصور طبية حقيقية، وتم تقييم النتائج باستخدام مقاييس كمية (MSE، PSNR، SSIM، D_f، D_p) وتحليل بصري نوعي. أظهرت النتائج أن النماذج الهجينة، وخاصة نموذج MLEMالقابل للتعلّم، تفوقت بشكل واضح من حيث الدقة وتقليل الضوضاء مقارنة بالطرق التقليدية.

يبرز هذا العمل أهمية الأساليب الهجينة المعتمدة على CNN في التصوير المقطعي بالحزمة المتوازية، مما يفتح المجال لإعادة بناء صور أكثر دقة وموثوقية وتوافقًا مع الاحتياجات السريرية.

الكلمات المفتاحية: إعادة بناء الصور، التصوير المقطعي، الحزمة المتوازية، BP، FBP، SART، MLEM، CNN، التعلم العميق، النماذج الهجينة

General Introduction

Medical imaging plays a crucial role in clinical diagnosis, therapeutic monitoring, and the planning of medical interventions. Among various imaging modalities, X-ray computed tomography (CT) stands out for its ability to provide high-resolution cross-sectional images of the human body, enabling detailed exploration of internal structures [1]. However, the quality of these images strongly depends on the reconstruction algorithm used, particularly in challenging clinical scenarios such as low-dose acquisitions, incomplete data, or noisy environments[2].

CT image reconstruction algorithms are traditionally divided into two main categories: analytical methods and iterative methods. Analytical methods, such as Filtered Back projection (FBP), rely on frequency filtering applied to the sinogram prior to back projection, which enhances the sharpness of the reconstructed image[2]. Although fast and straightforward to implement, these methods are sensitive to noise and artifacts, especially when data are limited or noisy.

Iterative methods, including Maximum Likelihood Expectation Maximization (MLEM) and Simultaneous Algebraic Reconstruction Technique (SART), aim to overcome these limitations. MLEM optimizes the likelihood function to progressively adjust the reconstructed image by comparing simulated projections to measured data, providing better noise handling at the cost of increased computational time [3,4]. SART, known for its fast convergence and effectiveness, simultaneously corrects differences between measured and simulated projections, offering a good trade-off between reconstruction quality and computation time [5].

With the advent of deep learning techniques, hybrid methods have been developed to enhance reconstruction quality by integrating convolutional neural networks (CNNs) into these traditional categories. Approaches such as FBP-CNN and MLEM-CNN leverage CNNs' ability to learn complex representations to reduce artifacts, suppress noise, and refine structural details, particularly in limited or noisy data conditions [6].

In this thesis, we focus on these two main categories, namely Filtered Back projection (FBP) for analytical methods and MLEM for iterative methods. We also investigate the integration of CNNs with these methods to assess improvements in image reconstruction quality. The goal is to compare the performance of FBP, MLEM, and their CNN-enhanced versions in terms of image quality, noise reduction, and structural fidelity, using quantitative metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), difference factor(D_f) and difference projection(D_p).

This thesis is structured into three main chapters:

- Chapter 1: Fundamentals of tomographic image reconstruction: presentation of basic concepts and analytical and iterative reconstruction methods.
- Chapter 2: Deep Learning and CNNs for tomography: exploration of neural networks applied to reconstruction enhancement.
- Chapter 3: Implementation, results, and algorithm comparison: experimental evaluation and discussions of obtained results.

This study aims to determine the real contribution of CNNs in the context of tomography, while identifying the optimal scenarios for each method in complex clinical environments.

Chapter1 Basic Concept of Image Computed Tomography (CT).

Summary

This chapter introduces the key concepts and techniques used in computed tomography (CT) image reconstruction. It begins with the basic definitions of projections, sinograms, and the Radon transform, followed by an explanation of how internal images are reconstructed from external X-ray measurements. Two main types of reconstruction methods are presented: analytical algorithms (like Back Projection and Filtered Back Projection) and iterative methods (such as ART, SIRT, SART, and MLEM). The analytical approaches offer fast, direct reconstruction, while iterative methods refine the image step by step for improved accuracy. The chapter concludes with a presentation of evaluation metrics MSE, PSNR, SSIM, DF, and DP which are used to objectively measure the quality of the reconstructed images and compare the performance of different algorithms.

Table of Contents

1.1 Introduction

Computed Tomography (CT) is a medical imaging technique that enables the reconstruction of cross-sectional images of an object from projection data acquired at multiple angles. It is based on the principle of transmitting X-ray beams through an object and measuring the attenuated intensities. The goal is to reconstruct the internal image of the object from this projection data, a process known as **image reconstruction** [7].

In our work, we focus on **parallel beam** projection, which is one of the simplest and most commonly used models in theoretical and simulation-based studies. In this geometry, all X-ray beams are assumed to be parallel to each other for a given projection angle, simplifying both the forward and inverse modeling of the imaging process [8].The reconstruction from projections lies at the heart of CT techniques. There are two main categories of reconstruction algorithms: **analytical methods**, namely Back Projection (BP) and Filtered Back Projection (FBP), and **algebraic (iterative) methods**, including **Algebraic Reconstruction Technique (ART)**, SART, SIRT, and MLEM.

This chapter introduces the essential terminology of computed tomography, explains the basic principles of image reconstruction from projections, and describes the conventional reconstruction methods along with the performance metrics used to assess reconstruction quality.

1.2 Basic Concepts and Terminology of CT

Fundamental ideas and parameters are necessary to comprehend the image reconstruction procedure in computed tomography (CT). These cover topics like projection geometry, sinogram, the Radon transform, and reconstruction algorithms methods. These key terms are fundamental to understanding the image reconstruction process.

1.2.1 Projection

In computer tomography (CT), a **projection** is the measurement of the X-rays' attenuation as they pass through an object at a particular angle. These projections are necessary to reconstruct images of the object under study in transverse slices [2]. During a CT scan, the scanner sends X-rays that go through the patient's body from various perspectives. Once the X-rays have passed through the tissues, the detectors measure their intensity. Every measurement

reflects the sum of the tissue's coefficients of attenuation during the course of the beam and corresponds to a projection[8].

Each projection is equivalent to an integral of the attenuation density along a ray, which is mathematically modeled by the Radon transform [9].

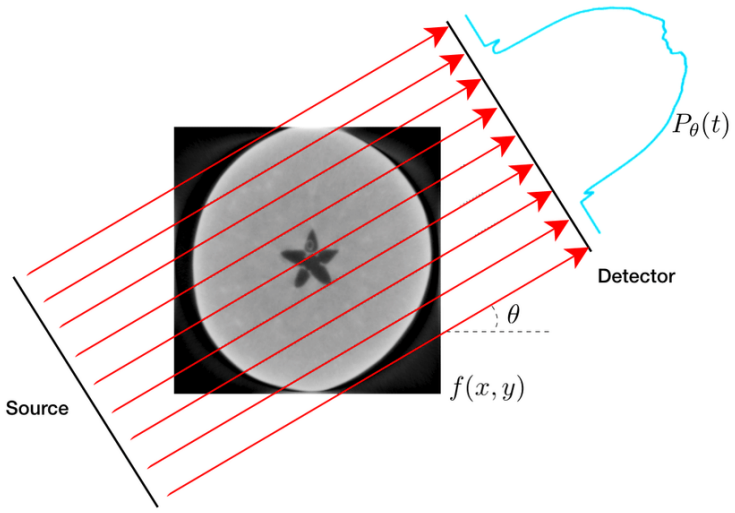


Figure 1-1 : Parallel beam projection. (10)

Fig 1.1 illustrates the principle of parallel beam projection used in tomography. Each ray passes through the object along a given direction (angle θ) and measures the attenuation through the material density $f(x,y)$. The set of projections at different angles forms the sinogram $P_\theta(t)$ obtained through the Radon transform.

1.2.2 Sinogram

The **sinogram** is a key intermediate representation in computed tomography. When an object is scanned at multiple angles, the scanner records a **projection** for each angle. This projection is the sum of the X-ray attenuations along lines passing through the object. These projections are arranged row by row based on the rotation angle to form an image known as the **sinogram** [8]. The sinogram thus contains all the information required to reconstruct the original image. Reconstruction algorithms, such as back projection (BP) or iterative techniques, take the sinogram as input to generate cross-sectional images of the scanned object [8]. Fig 1.2 shows a phantom sheep Logan image [11] and its sinogram; these images were obtained from the practical implementation of one of the reconstruction algorithms we applied.

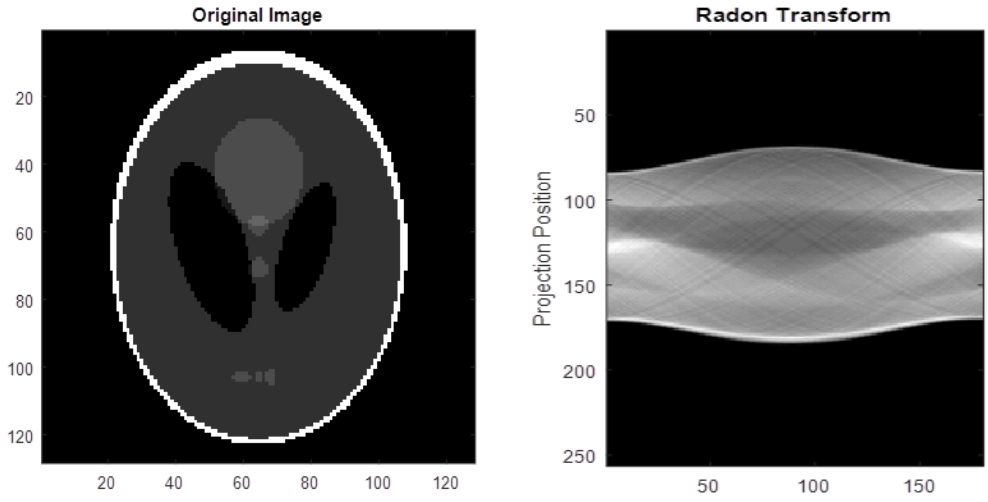


Figure 1-2 : original Phantom image and its corresponding sinogram.

1.2.3 Radon transform (RT)

The **Radon Transform** is a key mathematical tool used in tomography to model how projections are formed. When X-ray beams pass through an object, the detector records the sum of attenuation values along straight lines. The Radon Transform describes this summation process across all projection angles [2].

More precisely, it maps a 2D image $f(x, y)$ to a set of projections $p(\theta, s)$, where θ the projection angle and s is the position along the detector. It is defined as:

$$\boldsymbol{Rf(\theta, s) = \int_{-\infty}^{+\infty} f(s\ \cos\theta - t \sin\theta, s \sin\theta + t \cos\theta) dt} \quad (1.1)$$

This means that for each angle θ, the transform computes the line integrals of the image along all lines at that angle [9]. The Radon Transform is invertible, meaning that it is possible to reconstruct the original image from all its projections assuming a sufficient number of angles. This is the basis of reconstruction algorithms like BP and FBP[8].

1.3 Basic principles of object reconstruction through projection

Object reconstruction from projections is an essential technique used in computed tomography (CT). It enables the creation of an internal image of an object (such as the human body) based on measurements taken from multiple angles. In practice, parallel-beam X-rays pass through the object, and detectors capture the attenuation of the rays the energy lost as they traverse the object which provides partial views called projections. By combining these projections from various

angles, a complete image of the internal structure can be reconstructed. This process is mathematically modeled by the Radon transform, which relates the object's density to the acquired projection data. Image reconstruction is then carried out using various algorithms, including analytical methods and algebraic (iterative) techniques that aim to invert the projection process and retrieve the original image.

1.4 Conventional Methods of Computed Tomography (CT)

Computed Tomography (CT) image reconstruction has historically been divided into two major categories: **analytic methods** and **algebraic methods**. Each class approaches the reconstruction problem using different mathematical principles, offering distinct advantages and limitations depending on the application

1.4.1 Analytical CT algorithms

Analytical image reconstruction algorithms use direct mathematical inversion of projection data to reconstruct images. These techniques are commonly implemented in modern clinical CT scanners due to their speed and robustness [12]. Among these, the most widely used are the Back Projection (BP) and the Filtered Back Projection (FBP) algorithms. They constitute the foundational tools in tomographic image reconstruction and are critical for understanding the conversion of raw projection data into cross-sectional images.

- **Back projection algorithm (BP)**

The **Back Projection (BP)** algorithm is one of the simplest and most intuitive methods for reconstructing an image in computed tomography. Its principle is to take each projection and redistribute its values across the image along the direction in which it was acquired.

When this operation is repeated for all projection angles, the contributions accumulate, and the internal structure of the object begins to appear. Although the result is often blurry (because the projections are neither corrected nor filtered), this method provides a first approximation of the original image and helps to understand the basic concept of reconstruction [1].

Mathematically, the back projection of a projection $p(\theta, s)$ is written as:

$$f_{BP}(x, y) = \int_0^{\pi} p(\theta, \cos\theta + y \sin\theta) d\theta \quad (1.2)$$

Where:

$f_{BP}(x, y)$: The image reconstructed by back projection.

$p(\theta, s)$: The projection at point s for angle θ.

x, y: The spatial coordinates in the image.

- ## Filter back projection algorithm (FBP)

The **Filtered Back Projection (FBP)** algorithm is an improvement over the simple Back Projection (BP) method. While BP typically produces a blurry image, FBP applies a mathematical filter to the projection data before performing the back projection. This filtering step helps to sharpen the image and compensate for the excessive smearing introduced by the basic BP approach. The principle of FBP involves two main steps: first, the sinogram (the complete set of projections) is filtered in the frequency domain using filters, second the filtered projections are back projected to reconstruct the final image.

This method is widely used in clinical CT systems because of its speed, simplicity, and effectiveness in producing high-resolution images. It is expressed as follows:

$$f_{FBP} = \int_0^{\pi} [p_\theta(s) * h(s)] \, d\theta \tag{1.3}$$

In other words, FBP is a combination between filtering in the frequency domain, inverse Fourier transform and simple back projection of the filtered projections

The filters integrated into this study include:

- **Ram-Lak**: A basic ramp filter with a pure high-pass profile, which retains sharp image features but may also increase sensitivity to noise.

$$H(w) = |w| \tag{1.4}$$

- **Shepp-Logan**: A smoothed version of the Ram-Lak filter, obtained by multiplying it with a sinc function to dampen high-frequency noise while preserving structural details.

$$H(w) = |w| \sin\left(\frac{w}{2}\right) \tag{1.5}$$

- **Cosine**: This filter applies a cosine weighting to the ramp filter, achieving a compromise between noise reduction and edge clarity.

$$H(w) = |w| \cos\left(\frac{w}{2}\right) \tag{1.6}$$

- **Hamming**: Based on a window function, it significantly reduces high-frequency components, leading to smoother but potentially less detailed reconstructions.

$$H(w) = |w|(0{,}54 + 0{,}46\cos(\frac{w}{2})) \quad (1.7)$$

- **Hann**: Similar to the Hamming filter, with a different tapering curve, it further smooths the image by more aggressively reducing high frequencies.

$$H(w) = |w|(0{,}5 + 0{,}5\cos(\frac{w}{2})) \quad (1.8)$$

Synthesis

In summary, based on the previous sections, we can conclude that the **Back Projection (BP)** algorithm reconstructs the image by re-injecting each projection along the direction in which it was originally acquired. Although it provides an approximation of the original image, the result is often blurry, especially in the center of the image, due to the redundant accumulation of contributions.

To overcome these limitations, the **Filtered Back Projection (FBP)** algorithm was developed. It applies a frequency-domain filter to the sinogram data before performing the back projection. This filtering step compensates for the overestimation caused by BP and enhances the sharpness of the reconstructed image. The filtering operation is essential because it amplifies high-frequency components, which are often attenuated in the raw data.

In this study, several filters were tested within the FBP framework, including the **Ram-Lak, Cosine, Hamming, Hann, and Shepp-Logan** filters. Although these filters differ in terms of frequency response and smoothing capacity, the reconstructed images showed comparable results, with slight differences in contrast and noise management. Therefore, the choice of filter can be adjusted according to the desired trade-off between image sharpness and noise suppression. Due to its computational efficiency and accuracy, FBP remains the most widely used analytical reconstruction method in clinical practice today.

To more clearly illustrate the differences between the two analytical methods, Table 1.1 presents a comparative summary of their main characteristics.

Table 1.1 Comparative analysis between back projection and filtered back projection algorithms.

Criterion	**Back projection (BP)**	**Filtered back projection (FBP)**
Principle	Projects data back along the acquisition path	Projects data after frequency filtering
Image quality	Quality Blurred, lacks sharpness	High sharpness and detail
Noise handling	Poor, accumulates noise	Better noise suppression through filters
Filter usage	No	Yes (Ram-Lak, Hann, Hamming, etc.)
Limitation	Poor resolution, artifacts in center	Sensitive to noise depending on filter selection

1.4.2 Algebraic (iterative) algorithms

- **Algebraic Reconstruction Technique (ART)**

The **ART algorithm** is an iterative reconstruction method used in computed tomography (CT) to generate an image from angular projections (sinogram). It was first introduced by Gordon in 1970 [[13]. Unlike analytical method such as Back Projection, ART is based on a discrete model, where the reconstruction problem is formulated as a **system of linear equations**:

$$A \cdot x = b \tag{1.9}$$

Where:

A : is the projection matrix, modeling the acquisition process,

x: is the unknown image vector to be reconstructed,

b: is the measured projection data (the sinogram).

ART starts from an initial estimate of the image (often a uniform image), and at each iteration, it corrects the image step by step by comparing the calculated projections with the measured ones. This correction is done ray-by-ray, allowing for simple implementation and relatively fast

convergence. ART is particularly useful for noisy or incomplete datasets, but it can also be sensitive to measurement noise, which may affect reconstruction accuracy [1,12,14].

The algorithm updates the image at each iteration according to a formula of the type:

$$x^{(k+1)} = x^{(k)} + \lambda \cdot \frac{b_i - \sum_j a_{ij} x_j^{(k)}}{\sum_j a_{ij}^2} \quad (1.10)$$

Where:

λ is the **relaxation parameter** controlling convergence speed,

aij Represents the influence of pixel j on ray i,

b_i is the measured value for ray i.

The iteration continues until the difference between the simulated and measured projections is sufficiently small, indicating convergence.

- **Simultaneous Iterative Reconstruction Technique (SIRT)**

The Simultaneous Iterative Reconstruction Technique (SIRT) is an iterative method used in tomography that improves upon traditional algebraic methods like ART. It was introduced to reduce the instability and noise sensitivity of ART by using a global update strategy. Unlike ART, which updates the image ray by ray, SIRT performs simultaneous corrections across all pixels using all projection errors at once.

The problem is still represented by a linear system (equation (1.9)) but in SIRT, at each iteration, the difference between all measured and estimated projections is computed and then distributed across the image in a balanced way. This leads to smoother convergence, although it generally requires more iteration. The main benefit of SIRT is its ability to produce more uniform and less noisy images, especially when data is incomplete or affected by noise. However, it is computationally more demanding than ART[1,8,12].

The update equation is:

$$x^{(k+1)} = x^{(k)} + \lambda \cdot A^T \left(\frac{b - Ax^{(k)}}{W} \right) \quad (1.11)$$

Where:

$x(k)$: Current image estimate,

A^T: transpose of the projection matrix,

b: measured projection vector,

W: weighting vector (sum of projection coefficients per row),

λ: relaxation parameter.

- **Simultaneous Algebraic Reconstruction Technique (SART)**

The **Simultaneous Algebraic Reconstruction Technique (SART)** is an advanced iterative algorithm used in CT imaging. It was introduced by Andersen and Kak in 1984 to address the drawbacks of both ART (instability, noise) and SIRT (slow convergence) [5].

SART updates the image projection by projection, like ART, but instead of applying each correction immediately, it accumulates the corrections during each iteration and applies them simultaneously at the end. This leads to more stable and accurate convergence, especially for inconsistent or sparse data. Like ART and SIRT, SART uses the linear model equation (1.9). The image is updated using the following formula:

$$x_j^{(k+1)} = x_j^{(k)} + \lambda \cdot \frac{1}{\sum_i a_{ij}} \sum_i \frac{a_{ij}}{\sum_j a_{ij}} \cdot (b_i - \sum_j a_{ij} x_j^{(k)}) \quad (1.12)$$

Where:

$x_i(k)$: is the current estimate of pixel j,

a_{ij} : is the contribution of pixel j to ray$i$,

b_i : is the measured data for ray i,

λ : is a relaxation parameter.

SART is widely appreciated for balancing convergence speed, noise control, and robustness. It is particularly suitable for CT scanners using non-standard geometries or incomplete data [2,8,12].

- **Maximum Likelihood Expectation Maximization (MLEM)**

The MLEM (Maximum Likelihood Expectation Maximization) algorithm is an advanced iterative method used for image reconstruction in tomography. It is especially suitable for situations where the acquired data is noisy, incomplete, or obtained from complex acquisition

geometries. MLEM was introduced by Shepp and Vardi in 1982 as a statistical alternative to traditional algebraic reconstruction methods such as ART or SIRT [3].

Unlike classical techniques that attempt to solve the linear system eq1.9 directly, MLEM follows a probabilistic approach. It seeks to maximize the likelihood between the measured projection data and the estimated image, refining the image step-by-step using an Expectation-Maximization (EM) process [15].

MLEM relies on the EM algorithm to update the image iteratively. The update rule is expressed as:

$$x_j^{(k+1)} = x_j^{(k)} . \sum_i \frac{a_{ij}}{\sum_l a_{il} x_l^{(k)}} . b_i \tag{1.13}$$

Where:

$x_j(k)$: pixel intensity at position j at iteration k,

b_i: measured projection value for ray i,

a_{ij}: system matrix coefficient linking pixel j to detector i,

A: projection matrix modeling the acquisition system.

To make the process clearer, **Figure 1.3** illustrates the MLEM algorithm flowchart, showing the full iteration cycle for tomographic image reconstruction, the MLEM algorithm begins with an initial estimate of the image, typically uniform. At each iteration, it simulates the expected projections from the current image using the system matrix A. These simulated projections are then compared to the actual measured data (the sinogram) to calculate a correction factor in the form of a ratio. This ratio is then back projected across all the pixels of the image to adjust their values. Each pixel is updated by multiplying it by this factor, gradually improving the alignment between the estimated image and the measured data. This process is repeated until a reconstructed image is obtained that closely matches the acquired measurement

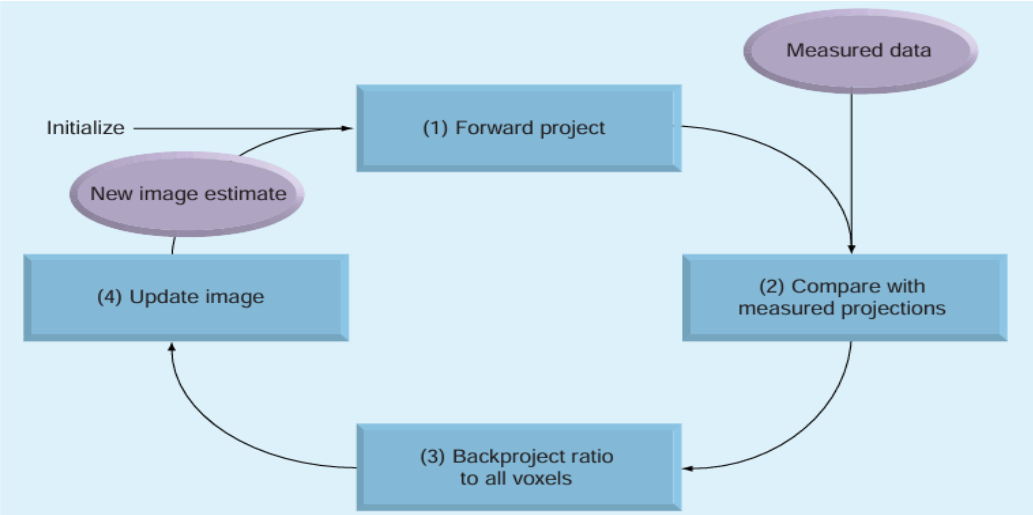


Figure 1-3 The MLEM algorithm flowchart [16]

Synthesis

Algebraic reconstruction methods are iterative techniques that aim to solve the inverse problem of image reconstruction from projection data, particularly when the data is incomplete or affected by noise. Among the most commonly used algorithms are ART (Algebraic Reconstruction Technique), SIRT (Simultaneous Iterative Reconstruction Technique), SART (Simultaneous Algebraic Reconstruction Technique), and MLEM (Maximum Likelihood Expectation Maximization).

The ART algorithm updates the image estimate sequentially using each projection line. While it converges quickly in early iterations, it is highly sensitive to noise and often produces unstable or artifact-prone results, making it less suitable for clinical or high-precision applications.

The SIRT algorithm improves on ART by updating the image estimate simultaneously using all projection data in each iteration. This global approach ensures more stable convergence and reduces noise amplification. However, SIRT is computationally more intensive and converges more slowly compared to ART.

The SART algorithm combines the benefits of both ART and SIRT. It processes one projection at a time, like ART, but applies normalization and weighting to each update to improve accuracy and stability. SART achieves a better compromise between convergence speed and image quality. It has been widely adopted in practical applications due to its ability to produce high-quality reconstructions while mitigating noise and artifacts effectively.

The MLEM algorithm is grounded in statistical modeling and maximizes the likelihood of the observed projection data under a Poisson noise model. This makes it particularly well-suited for low-dose or noisy acquisition scenarios. MLEM guarantees non-negativity of the image and can produce highly accurate reconstructions with fine structural detail. In order to summarize the characteristics of the four main algebraic reconstruction algorithms discussed in this work, the following table provides a comparative overview of ART, SIRT, SART, and MLEM. This summary highlights the key principles, convergence behavior, noise sensitivity, image quality, and the main advantages and disadvantages of each method.

Table1. 2 : comparative analysis between ART, SIRT, SART and MLEM algorithms.

Method	Principle	Convergence	Noise sensitivity	Image quality	Advantages	Disadvantages
ART	Pixel by pixel update for each ray	ray Fast but unstable	Very sensitive	Moderate, with possible artifacts	Easy to implement	High noise sensitivity, unstable
SIRT	update using all projections	Slow but stable	Low	Smoothed, stable	Noise reduction, stable results	Slow convergence
SART	Projection-by-projection update with normalization	Faster than SIRT	Moderate	Better than ART/SIRT	SIRT Good compromise between speed and quality	More complex to implement
MLEM	Based on statistical likelihood of measured data	Slow	Low (robust)	High, sharp results	Non-negativity ensured, ideal for noisy	Slow convergence computationally intensive

1.5 Quantitative Evaluation Metrics for Image Tomography Reconstruction Quality

To objectively evaluate the performance of image reconstruction methods, it is essential to use **quantitative metrics**. These criteria compare the reconstructed image to a reference image (commonly referred to as the *ground truth*) and assess the level of accuracy and similarity. The main evaluation metrics used in this work are presented below:

1.5.1 Mean Squared Error (MSE)

The **Mean Squared Error (MSE)** measures the average squared difference between the pixels of the original image $I(i,j)$ and the reconstructed image$R(i,j)$. It is mathematically defined as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I_0(i,j) - I_r\,(\mathrm{i,j}))^2 \tag{1.14}$$

- An MSE value of 0 indicates a perfect reconstruction, where there is no difference between the original and reconstructed images.
- Higher MSE values reflect greater errors, indicating lower image quality and higher deviation from the reference image.

1.5.2 Peak Signal-to-Noise Ratio (PSNR)

The **Peak Signal-to-Noise Ratio (PSNR)** is a widely used metric for evaluating the quality of image reconstruction, particularly in image compression and medical imaging applications. It expresses the logarithmic ratio between the maximum possible intensity value of a pixel and the power of the noise or error in the reconstructed image [17].

It is defined mathematically as:

$$\mathrm{PSNR} = 10 \cdot log_{10}(\frac{MAX_I^2}{MSE}) \tag{1.15}$$

Where:

MAX is the maximum possible value of the image pixels (usually 255 for 8-bit images),

MSE is the Mean Squared Error between the original image and the reconstructed image [17]

- A higher PSNR value typically indicates better reconstruction quality, as it reflects lower error between the two images [18].
- A PSNR above 30 dB is generally considered acceptable in many applications of image processing and compression [17]

1.5.3 Structural Similarity Index (SSIM)

The **Structural Similarity Index (SSIM)** is a perceptual metric used to measure the similarity between two images. Unlike MSE and PSNR, which only consider pixel-wise differences, SSIM evaluates structural information, which aligns better with human visual perception.The SSIM

metric compares three aspects: Luminance ,Contrast, and Structure between the original and reconstructed images [19]

The SSIM formula is given by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (1.16)$$

Where:

μ_y, μ_y are the average pixel intensities of images x and y,

σ_x^2 , σ_Y^2 are the variances,

σ_{xy} is the covariance between the two images,

C_1 , C_2 are small constants to stabilize the division[19].

- SSIM values range from -1 to 1, with 1 meaning perfect similarity between the original and reconstructed images [19]
- Higher SSIM indicates better structural fidelity, especially useful when the goal is to preserve image details and textures.

1.5.4 Difference Factor between Original and Reconstructed Images D_f

This metric evaluates the **overall relative error** between the reconstructed image $f_{\text{reconstructed}}$ and the original image f_{original} [8]

It is defined as:

$$D_f = \frac{\|f_{\text{original}} - f_{\text{reconstructed}}\|2}{\|f_{\text{original}}\|2} \quad (1.17)$$

- A value close to 0 means the reconstruction is faithful to the original.
- A value near 1 indicates a significant mismatch [2]

1.5.5 Deviation Proportion between Measured and Calculated Sinograms D_p

The DP metric quantifies the **consistency** between the measured sinogram p_{measured} and the sinogram recalculated from the reconstructed image $p_{\text{calculated}}$ [20].

It is defined as:

$$D_p = \frac{\| p_{\text{measured}} - p_{\text{calculated}} \|2}{\| p_{\text{calculated}} \|2} \tag{1.18}$$

- A low DP value implies good agreement between the reconstruction and the original data.
- A high DP value suggests a poor match and potentially inaccurate reconstruction.

1.6 Conclusion

This chapter introduced the fundamental concepts of image tomography reconstruction in computed tomography. We discussed the principles of parallel beam geometry, the Radon transform, and the main reconstruction techniques—both **analytical** (BP, FBP) and **iterative** (ART, SIRT, MLEM). Each method offers distinct advantages and limitations, and the choice of method depends on factors such as data quality, required accuracy, and computational resources.

To allow for an objective comparison of reconstruction performance, various quantitative metrics were also presented, including MSE, PSNR, SSIM, D_f, and D_p. These indicators will be essential in the following chapters to assess the quality of the reconstructed images under different methods and acquisition scenarios

Chapter2 Fundamental Principles of Deep Learning for Image Tomography Reconstruction.

Summary

This chapter provides a comprehensive introduction to the fundamental concepts of Deep Learning and neural networks, which constitute the core of modern Artificial Intelligence (AI) systems. Special emphasis is placed on Convolutional Neural Networks (CNNs), a powerful class of deep learning models particularly well-suited for image-related tasks. The chapter explores the underlying principles of CNN architectures, including essential operations such as convolution, activation, pooling, and fully connected layers. These components are explained in a simplified yet rigorous manner to highlight their role in enabling machines to learn and extract complex patterns from data. The relevance of CNNs to medical imaging, especially in image tomography reconstruction tasks, is also emphasized, setting the foundation for the advanced methods discussed in the following chapter.

Table of Contents

2.1 Introduction

Artificial Intelligence (AI) has emerged as a pivotal field in computer science, empowering machines to perform tasks that emulate human reasoning and decision-making. Within AI, Machine Learning (ML) and more specifically Deep Learning (DL) have achieved significant success in tackling complex challenges, particularly in the realm of image analysis. Among DL techniques, Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in medical imaging, providing robust tools for precise and efficient CT image reconstruction. This chapter introduces the fundamental concepts of AI, with a particular emphasis on Deep Learning and CNNs, and examines their growing applications in computed tomography.

This chapter begins with an introduction to the fundamentals of artificial intelligence, followed by an exploration of deep learning as a key approach in the field. It then provides an overview of artificial neural networks before examining in detail convolutional neural networks (CNNs) and their architecture. The chapter concludes with a general summary of the concepts discussed.

2.2 Artificial intelligence (AI)

Artificial intelligence (AI) refers to the set of techniques and systems capable of simulating human intelligence, including learning, reasoning, problem-solving, and decision-making [21]. As illustrated in Figure 2.1, AI is a broad and rapidly evolving field that encompasses various subfields such as machine learning and deep learning [22].

Machine learning (ML) enables machines to learn from data and improve their performance autonomously without explicit programming, allowing systems to adapt to new information and make predictions or decisions based on patterns [23]. Deep learning (DL), a subset of machine learning, relies on multilayer artificial neural networks inspired by the human brain's structure, which can process vast amounts of complex data and automatically extract meaningful features [24]. This hierarchical relationship highlights that deep learning is a specialized branch within machine learning, which itself forms a crucial part of the broader AI landscape.

Today, AI technologies are revolutionizing numerous sectors, including healthcare, finance, transportation, and manufacturing, by providing innovative, efficient, and adaptive solutions to complex real-world problems. As AI continues to advance, it holds the promise of enabling smarter systems that can augment human capabilities and drive transformative changes across society.

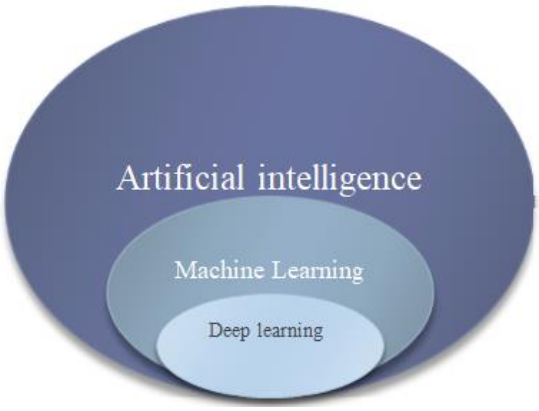


Figure 2-1 : Relation between AI, ML, deep learning.

2.3 Deep Learning (DL)

Deep Learning is a subfield of Machine Learning inspired by the structure of the human brain to process information (24). This approach uses artificial neural networks with multiple layers (*deep layers*) to automatically extract complex features from large amounts of data [22].

The depth of these networks refers to the number of input, output, and hidden neuron layers. Each layer transforms input data into meaningful information, which is then used by subsequent layers for task-specific predictions. This architecture allows the system to autonomously learn from data and excels in feature extraction, feature selection, and classification tasks.

2.4 Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) is models inspired by the human brain, designed to recognize complex patterns in data. They are composed of artificial neurons as shown in figure 2.2 organized in layers: an input layer, one or more hidden layers, and an output layer [23,25]. Typical neural network architecture includes the following:

A neural network is structured into three main types of layers:

- **Input Layer**: It receives the raw data, where each neuron corresponds to an input feature, such as a pixel in image processing [22].
- **Hidden Layers**: These intermediate layers transform inputs through multiple neurons. Their depth and architecture depend on the task's complexity [26]
- **Output Layer**: It produces the final prediction, with neurons representing target classes or output values like "dog" or "car" in classification [22].

- **Weights**: control how important the information passed between neurons is, and they play a key role in how input data is processed.
- **Bias** is an extra value that helps the neuron adjust better, even when the input is small. It makes the model more flexible.
- **Activation Function** applies a non-linear transformation to the neuron's output, allowing the network to understand more complex patterns [22].

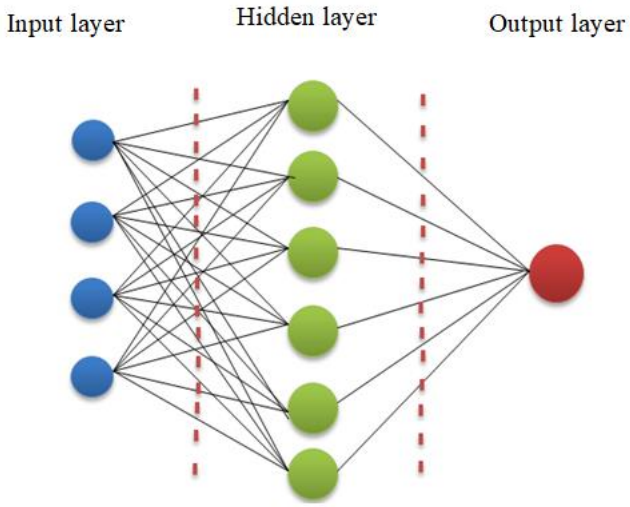


Figure 2-2 : architecture of Artificial Neural Networks (ANN).

2.4.1 Perceptron

The perceptron is a fundamental model of an artificial neuron, often regarded as the basic building block of neural networks. It was first introduced by **Frank Rosenblatt in 1958** as a simple computational model inspired by the biological neuron [27].

A perceptron takes a set of numerical inputs and produces a binary output (0 or 1) based on a linear combination of the inputs followed by an activation function.

Each neuron receives input value denoted as x_i which are multiplied by corresponding weight w_i. A **weighted sum** of the inputs, which can be mathematically expressed as:

$$\varepsilon = \sum_{x=1}^{n} x_i . w_i \qquad (2.1)$$

This weighted sum ε is then added by a bias b, the result is then passed through an **activation function (**such as **ReLU),** which introduce non-linearity into the model.

For example, a neuron may perform the following calculation

$$z = w_1 x_1 + w_2 x_2 + b \tag{2.2}$$

And then apply the ReLU activation function, this operation allows the network to learn complex relationships between the input and the output.

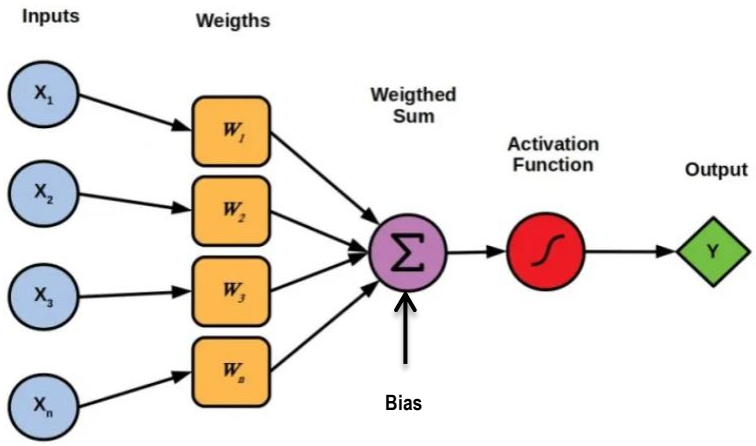


Figure 2-3 : architecture of the perceptron.

2.5 Neural Networks Vocabulary

Understanding the language and concepts used in neural network architectures is a key step toward grasping how deep learning models are built and improved. The more familiar we become with this terminology, the easier it is to make sense of how these models work, adapt to different problems, and evolve through training and optimization.

2.5.1 Backpropagation

Backpropagation is a fundamental method used to train neural networks by minimizing the error between predicted and actual outputs, when the neural network makes an incorrect prediction, back propagation calculates how wrong it was, and adjusts the weights to reduce future errors. It works in two main steps:

- **Forward Pass**: the input data is passed through the network, layer by layer, to produce an output.
- **Backward Pass:** The network calculates the difference (called the **loss**) between the predicted output and the true value. Then it uses this loss to compute the **gradient** of the error with respect to each weight, starting from the output layer and going backward to the input layer, (These gradients are used to update the weights using **gradient descent)**.

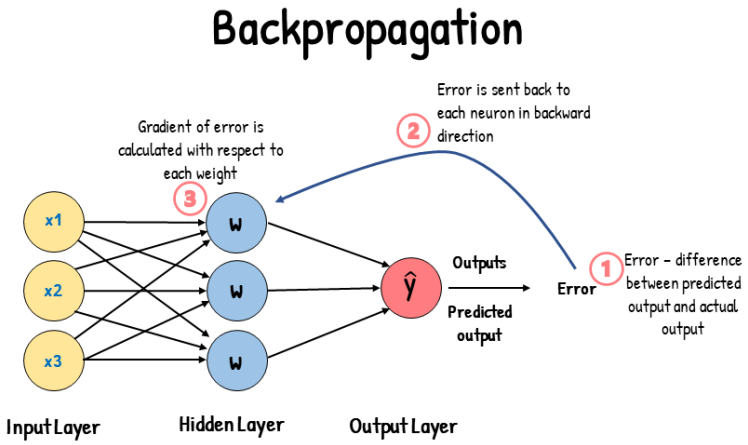


Figure 2-4 : Simple Illustration of how the backpropagation works by adjustment of weight (28).

2.5.2 Gradient Descent

Gradient Descent is one of the most widely used optimization algorithms in Deep Learning to tune the model's parameters [22]. It works by minimizing the loss function by adjusting the model weights in the opposite direction of the gradient [29]. At each iteration, the model makes a prediction and calculates the error (or loss) between the predicted output and the actual target [30].Then, the gradient of the loss with respect to each parameter is computed.

2.5.3 Learning Rate

The learning rate is a parameter that defines how much the model's weights are updated at each iteration during training [30]. It controls the speed of learning: a rate that is too high may cause the model to overshoot the minimum, while a rate that is too low may significantly slow down training [30]. Choosing the right learning rate is essential to ensure efficient and stable convergence of the gradient descent algorithm [31]. As shown in Figure 2.5, the initial weight is

updated using the gradient, and the size of the step depends on learning rate. This visual example helps to understand how the weight moves toward the minimum of loss function.

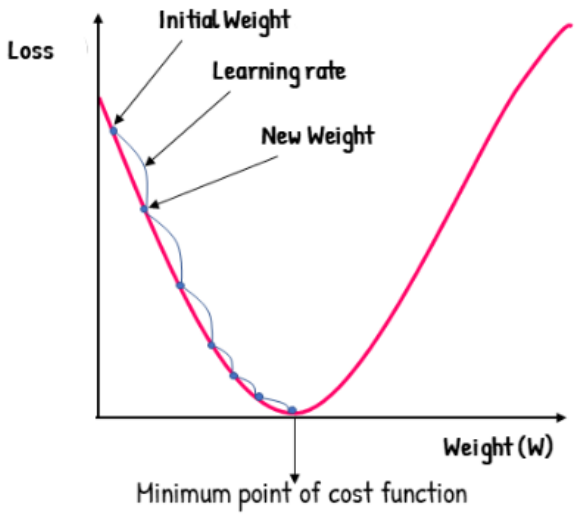


Figure 2-5 : illustration of weight update based on learning rate in gradient descent.

2.5.4 Batch

A batch is a group of data selected from the training set, used to update the model's weights [22]. Instead of using the entire dataset at each iteration, the data is split into smaller groups called mini-batches, making training faster and more efficient [32]. Using small batches allows for faster convergence and lower memory usage, but can introduce instability or noise during training. On the other hand, larger batches make the training more stable, but require more memory and may lead to slower convergence [33].

2.5.5 Epoch

An **epoch** refers to one complete pass of the entire training dataset through the neural network [22]. In other words, an epoch means that each example in the dataset has been seen once by the model. Usually, several epochs are required for the network to effectively learn the patterns present in the data. The number of epochs directly influences the model's performance: too few epochs may not allow the model to learn enough, while too many may cause the model to memorize the data and fail to generalize well [33].

2.5.6 Hyperparameters

Hyperparameters are configuration settings that control how a neural network is trained [22]. Unlike model parameters (such as weights and biases), hyperparameters are not learned during training they are set manually before the learning process begins. Examples of common hyperparameters include:

- the **learning rate**, which determines how fast the model updates its weights,
- the **batch size**, which specifies how many samples are processed at once,
- The **number of epochs**, which defines how many times the full dataset, is used during training.

Choosing good hyperparameters is crucial for achieving optimal performance. If they are poorly selected, the model may learn too slowly, fail to converge, or overfit the training data [34].

2.6 Convolution Neural Networks (CNN)

Convolutional Neural Network (CNN) is a type of deep neural network designed to process data structured in grid formats, such as digital images composed of pixels arranged in matrices [34]. Unlike fully connected networks, where each neuron is linked to all others, CNNs use **local filters (or kernels)** that slide over the image to automatically extract simple visual features such as edges, and then more complex patterns in the deeper layers, such as shapes and objects [22]. The main goal of CNNs is to reduce the model's complexity while maintaining a strong representational capacity. This is achieved through several specialized layers as represent in Figure 2.6 (**Convolutional layers**, **ReLU activation functions**, **pooling layers** and finally **fully connected layers)**.

Thanks to this efficient architecture, CNNs are widely used in fields such as facial recognition, automated medical anomaly detection, as well as in image reconstruction applications. In medical imaging and tomography, CNNs enhance the quality of reconstructed images from limited or noisy projections [22,24].

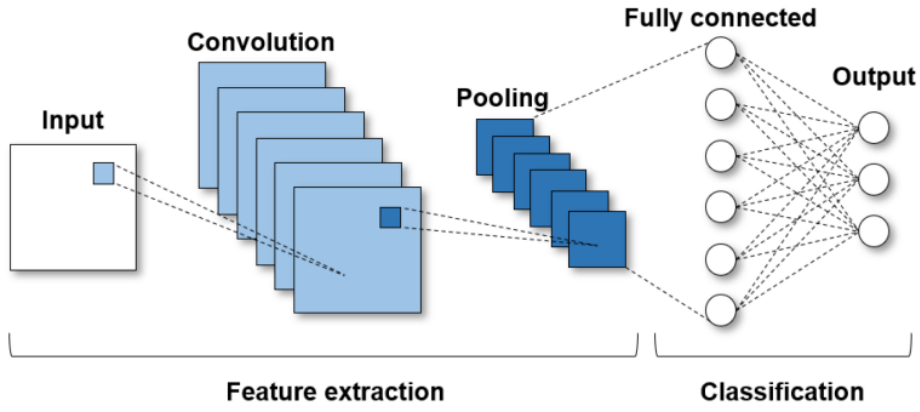


Figure 2-6 : Basic CNN architecture (35).

2.6.1 Convolution layer

The **convolutional layer** is the core component of a convolutional neural network. It uses a small **filter (or kernel)** that slides over the input image to automatically detect patterns such as edges, textures, or simple shapes [36]. This process is called **convolution**. It involves locally multiplying the filter values with those of the image, then summing the results to produce a new pixel in an image called a **feature map** [37].Each filter learns to recognize a specific type of pattern in the data, which allows the network to progressively understand more and more complex structures in the following layers [36]. Figure 2.7 illustrates the basic operation of a convolutional layer in a CNN. A small filter slides over the input image, performing convolution by multiplying and summing values at each position. The resulting values form a feature map that highlights important patterns such as edges and textures.

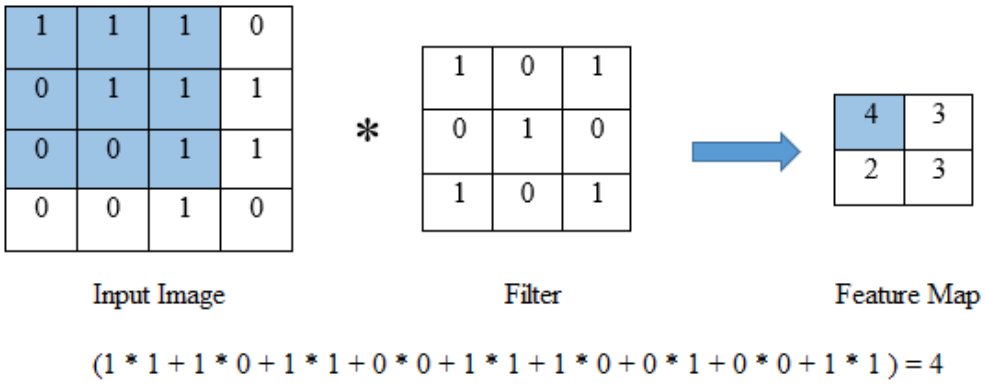


Figure 2-7: basic operation of a convolutional layer in a CNN (38).

2.6.2 Rectified Linear Unit function (ReLU)

After the convolution step, an activation function called **ReLU (Rectified Linear Unit)** is applied. This function introduces non-linearity, which is essential for the network to learn and model complex patterns in the data [39].Mathematically, the ReLU function is defined as:

$$ReLU\ (z) = \max(0, z) \tag{2.3}$$

In simple terms, the ReLU function replaces all negative values with zero and **retains positive** values as they are. This allows the model to remain computationally efficient while preserving important information from the convolution step [39].

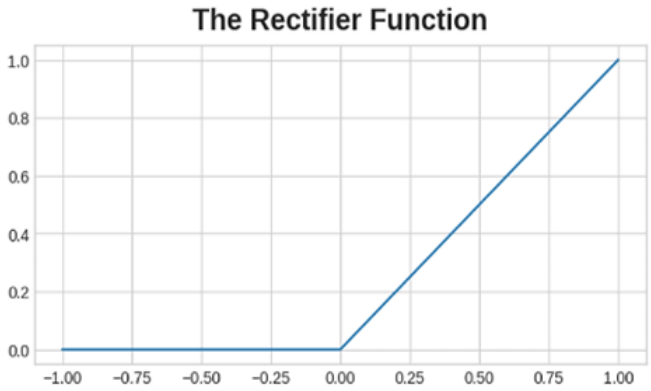


Figure 2-8 : Relu activation function (40).

As shown in **Figure 2.8**, the ReLU function outputs 0 for all negative inputs and increases linearly for positive inputs.

The following example in Figure 2.9 demonstrates the practical application of the ReLU activation unction to an input matrix. As shown, all negative values are replaced by zero, while positive values are retained. This operation allows the network to propagate only meaningful activations while discarding non-informative (negative) ones. Therefore, the ReLU function plays a key role in reducing computational complexity while preserving essential features for learning.

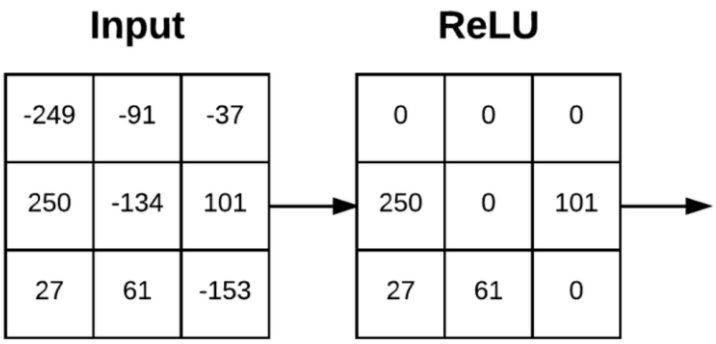


Figure 2-9 : illustration of the ReLU activation functions with a numerical example [41].

2.6.3 Pooling layer

The **pooling layer** is a key step in Convolutional Neural Networks (CNNs) applied after activation function (ReLU). It helps to reduce the size of the feature maps created after convolution, which lowers the number of parameters and speeds up computations . By reducing the data size, pooling also helps to prevent overfitting, meaning the model will generalize better instead of just memorizing the training [36]. **Max-pooling** is the most commonly used method. It selects the highest value in a small window (like 2×2) and discards the rest, which helps to keep the most important features [36].

Figure 2.10 illustrates the Max Pooling operation applied to a feature map using a 2×2 filter and a stride of 2. In this example, the input matrix is divided into non-overlapping 2×2 blocks. For each block, the maximum value is selected to form the output matrix. Max Pooling helps reduce the spatial dimensions of the data, retaining only the most important features while reducing computational load and controlling overfitting.

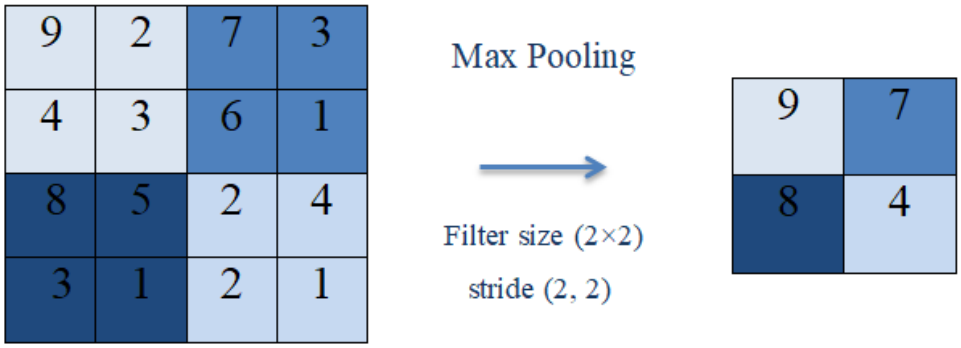


Figure 2-10 : Example of pooling layer.

2.6.4 Fully connected layer

The **fully connected layer** is a crucial final step in convolutional neural networks (CNNs). It connects every neuron from the previous layers to each neuron in the output layer, similar to a traditional feed forward neural network. This layer takes the features extracted by the convolutional and pooling layers and combines them to make a final prediction. For example, in an image classification task, it determines which class the image most likely belongs to.

The output of this layer is typically passed through a final activation function, such as the SoftMax function, which converts the raw scores into probabilities associated with each class [36].This architecture enables the network to make a global decision by integrating all the local information learned in the earlier stages.

Figure2.11 explains how a Convolutional Neural Network (CNN) processes an image to recognize a digit. It starts with an input image of size 28×28 pixels, in gray scale (1 channel, so depth = 1).The first convolution layer applies a 5×5 filter with "valid" padding (no extra pixels are added around the edges). This reduces the size to $24 \times 24 \times n1$, where n1 is the number of filters, and it becomes the new depth. Then, a 2×2 max pooling operation is applied, cutting the width and height in half, resulting in$12 \times 12 \times n1$. A second 5×5 convolution (also with no padding) followed by another 2×2 max pooling gives a feature map of size$4 \times 4 \times n2$, with n2 filters. These feature maps are then flattened into a 1D vector, passed through two fully connected layers, and finally to an output layer that predicts the digit class (from 0 to 9).

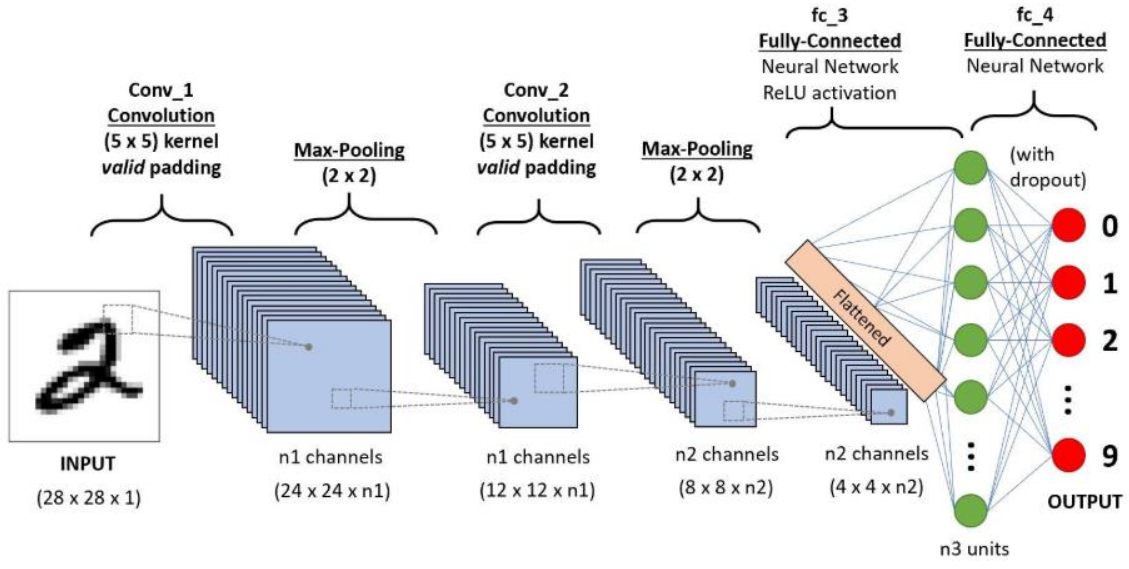


Figure 2-11 : convolutional neural network CNN from input image to output layer (42).

2.7 Conclusion

In this chapter, we explored the foundations of Deep Learning, including key learning mechanisms. Special attention was given to Convolutional Neural Networks (CNNs), which are a major breakthrough in image processing due to their ability to automatically extract meaningful features. These properties make CNNs particularly well-suited for image tomography reconstruction tasks, where the goal is to recover a clear and accurate image from incomplete or noisy projections.

Therefore, in the following chapter, we will leverage the power of CNNs to enhance the quality of tomographic image reconstructions and compare them with traditional methods. More precisely, we will implement four tomography reconstruction architectures based on CNN. The use of Deep Learning in this context paves the way for faster, more accurate, and more robust reconstructions.

Chapter 3 Implementation of CT Image Tomography Reconstruction Analytic and Iterative Algorithms: Conventional & Learnable by CNN.

Summary

This chapter is entirely devoted to presenting the main results obtained from our simulation study. It begins with the evaluation of conventional CT image reconstruction algorithms, including analytic approaches such as BP and FBP, as well as iterative methods such as SART and MLEM. To assess the robustness of these techniques, noisy projection data are also taken into account.

In the second part of the chapter, we focus on advanced methods that integrate convolutional neural networks (CNNs) into the reconstruction process, specifically within the frameworks of FBP and MLEM. Four hybrid algorithms are proposed and studied: FBP+DnCNN, a learnable FBP, MLEM+DnCNN, and a learnable MLEM approach. The description of the simulation steps and the architecture of the neural networks is first introduced, followed by the results obtained for each method. Finally, a comparative analysis of all the algorithms is carried out using widely recognized quantitative evaluation metrics.

Table of Contents

3.1 Introduction

In this chapter, we present the simulation experiments and the results obtained from different image reconstruction methods in computed tomography (CT). We begin with conventional techniques, including analytical methods such as Back projection (BP) and Filtered Back projection (FBP), as well as iterative methods like SART and MLEM. For FBP, several filters have been implemented: ramp, Hamming, Hann, Shepp-Logan, and cosine.

The second part of the chapter focuses on four reconstruction algorithms based on convolutional neural networks (CNNs): FBP+DnCNN, Learnable-FBP, MLEM+DnCNN, and Learnable-MLEM. The simulation process is structured into three main stages: data preparation, model training, and performance evaluation. A quantitative comparison is carried out using commonly used metrics in the literature, including Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Difference Factor between Original and Reconstructed Images$\boldsymbol{D_f}$, and Deviation Proportion between Measured and Calculated Sinograms$\boldsymbol{D_p}$. Noisy datasets are also included in our tests to evaluate the robustness of the proposed algorithms.

Only parallel beam projections are used in this work. We begin by describing the dataset, then present the main steps of the algorithm implementations. The improvements in the reconstructed images are demonstrated through the evaluation metrics, which are summarized in several tables

3.2 Used Data base

In our simulations, we used three types of images: the Shepp-Logan phantom, a brain image, and the data base thoracic computed tomography (CT) images extracted from publicly available Kaggle datasets [1].

- The Shepp-Logan phantom, illustrated in Figure 3.1, is a synthetic gray scale reference image with a resolution of 256×256 pixels. It is widely used in tomographic reconstruction research, as it provides a controlled and ideal noise-free environment for precise evaluation of reconstruction algorithms.

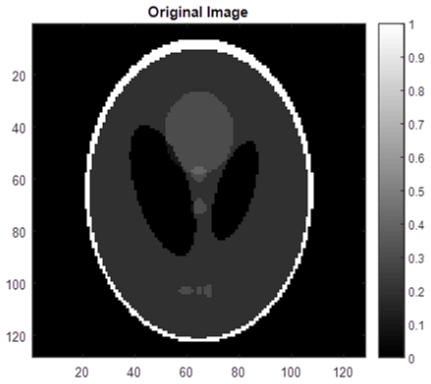


Figure 3-1 Phantom Shepp-Logan image

- **Brain** image, shown in Figure 3.2, is used to evaluate the performance of reconstruction methods on complex and realistic anatomical structures, thus providing a transition between synthetic phantoms and clinical data. The image is gray scale with a resolution of 256×256 pixels.

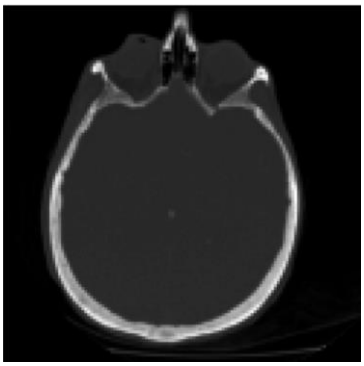


Figure 3-2 brain image.

- **Medical CT** image, illustrated in Figure 3.3, provide cross-sectional views of the human thorax, highlighting the lungs and other internal structures. These images are characterized by high contrast and high resolution, making them particularly relevant for evaluating the robustness and effectiveness of reconstruction methods under realistic clinical conditions. The dataset, obtained from the Kaggle platform, contains a total of 1000 images, divided into three subsets:
  - Training data: 70%.
  - Validation data: 10%.
  - Test data: 20%.

This dataset is particularly used for training and evaluating deep learning (DL) algorithms in medical image reconstruction.

Figure3-3: CT scan slices of different body regions.

- **Generated noisy images** to evaluate the robustness of the proposed algorithms against noise, we generated noisy versions of the original images. Specifically, Gaussian noise with different standard deviation values was added to the noise-free images to simulate various noise levels. For each noisy image, we generated the corresponding sinogram and reconstructed image, allowing us to assess the performance of the reconstruction methods under noisy conditions.

3.3 Materials and Software Tools

The development and simulation of tomographic reconstruction algorithms were performed on a personal computer under Windows 10, with the following hardware configuration:

- **Device name:** DESKTOP-3M7OTA9
- **Processor:** Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (up to 1.80GHz)
- **Installed RAM:** 8.00 GB (7.84 GB usable)
- **System type:** 64-bit operating system, x64-based processor
- **Touch input:** Supported with 10 touch points

Two main programming environments were used: **MATLAB** and **Python** (via **Spyder IDE** and **Google Colab**).

MATLAB R2015a was used to develop and test different versions of both analytical and algebraic reconstruction algorithms. Its interactive environment, efficient matrix operations, and image processing tools made it suitable for rapid prototyping and visualization.

Python was used in two configurations:

- **Google Colab**, for running large-scale simulations involving deep learning models (FBP and MLEM based).
- **Spyder IDE (version 3.12.7)** under the Anaconda distribution, installed locally on the PC, was used for algorithm development, data preprocessing, result visualization, and performance evaluation using CPU resources.

This dual-environment setup enabled flexible and efficient development, as well as cross-validation of the results between conventional and deep learning-based reconstruction techniques.

3.4 Implementation of Conventional CT Reconstruction Algorithms

This section presents the implementation and evaluation of conventional CT image reconstruction techniques, including both **analytical methods** such as Back Projection (BP) and Filtered Back Projection (FBP) with different filters and **algebraic (iterative) methods**, such as Simultaneous Algebraic Reconstruction Technique (SART) and Maximum Likelihood Expectation Maximization (MLEM).

3.4.1 Analytical algorithm

- **Back projection (BP)**

This algorithm implements the **Back Projection (BP)** to reconstruct an image from its projections. The image is projected at **180 different angles**, ranging from 1° to 180°, with a step size of 1°. These projections are then **back-projected** into the image space according to their respective angles. By **superimposing** all these back-projections, an estimation of the original image is obtained. This approach is based on the idea that each projection carries partial information, and their accumulation enables full image reconstruction. Finally, a **normalization** step is applied to scale the values of the reconstructed image between 0 and 1, which facilitates both visualization and quantitative evaluation.

Figure 3.4 back projection BP Reconstruction Process

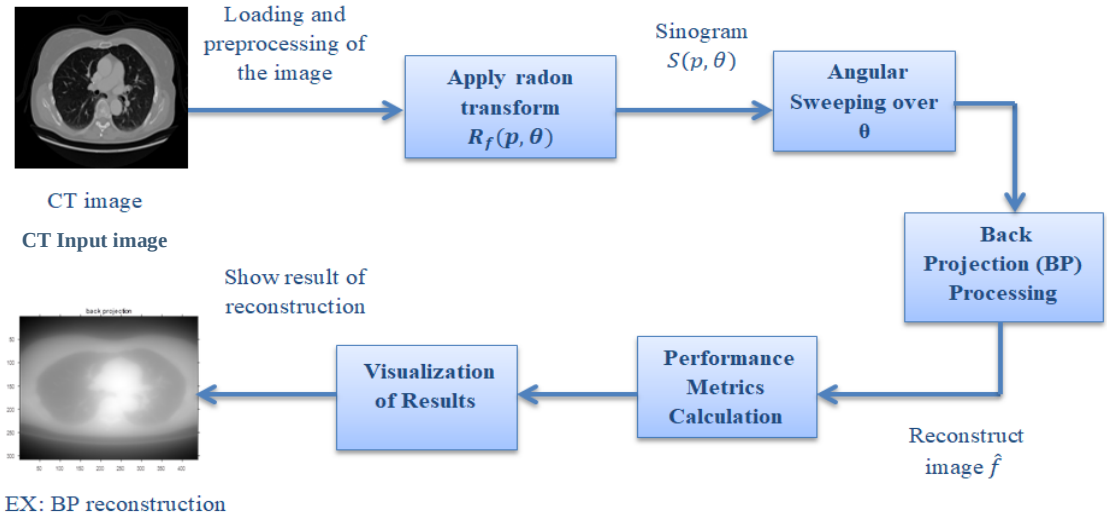


Figure 3-4: BP Reconstruction Process for CT Imaging

Where:

$\boldsymbol{R_f(\theta, s)}$ Denotes the **Radon transform** of the function $\boldsymbol{f(x, y)}$ as defined in Equation (1.1), $\boldsymbol{\theta}$ is the **projection angle** (with a total of 180 projections), and $\hat{f}(x, y)$ represents the **reconstructed image** from the projection data.

Figure.3.5 shows sinogram of Phantom sheep Logan image, and its result of reconstruction.

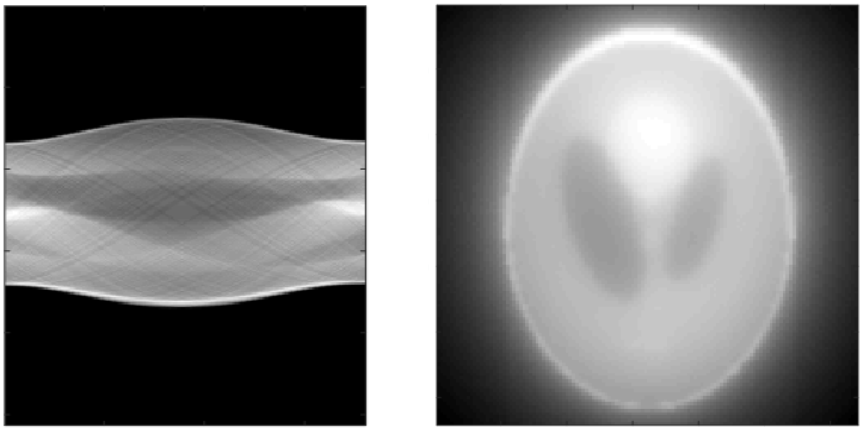


Figure 3-5 : sinogram of phantom sheep Logan image and its result of reconstruction.

The BP algorithm produces a basic reconstruction that retains structural information but suffers from significant blurring, particularly in low-contrast areas or fine details, even though we increase the number of projections. To better understand the limitations of the back-projection algorithm, we have also calculated the corresponding metrics shown in Table 3.3.

Table3.3 : performance metrics of reconstructed phantom image using BP.

Image	**MSE**	**PSNR(dB)**	**SSIM**	D_f	D_P
Phantom	0.3368	4.73	0.1275	4.6200	6.6371

The obtained metric values as shown in table 3.3 confirm that the reconstruction quality using back projection BP is low. The reconstructed image is blurry, with few visible details and poor structural similarity to the original image.

In the same way as with the phantom image, the Back Projection (BP) algorithm is applied to the remaining images in order to test its performance and analyze the quality of the reconstruction. Figure.3.6 shows sinogram of brain image, and its result of reconstruction.

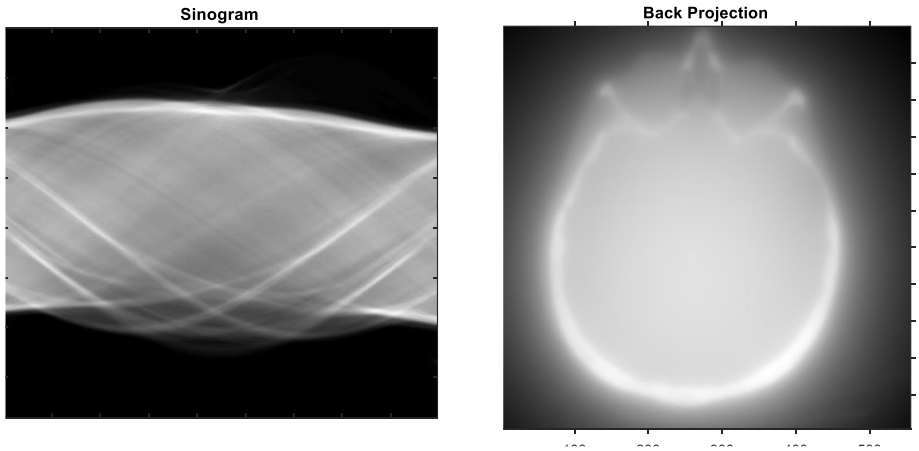


Figure 3-6: sinogram of brain image, and its result of reconstruction.

The reconstruction result of the **brain image using the Back Projection (BP) algorithm** appears significantly blurred compared to the original. The image suffers from poor edge definition, and internal anatomical details are barely distinguishable, indicating the limitations of BP in preserving fine structures.

Figure.3.7 shows sinogram of image (a), and its result of reconstruction.

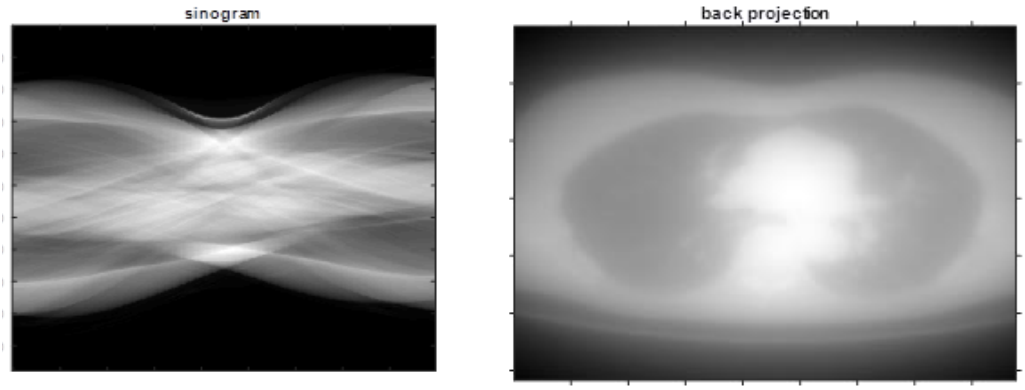


Figure 3-7: sinogram of image (a), and its result of reconstruction.

Table 3.4 presents all these results for each image, allowing for a quantitative comparison of the reconstruction BP images.

Table3.4 : perfermonce metric of BP reconstruction.

Metrics / images	**MSE**	**PSNR (dB)**	**SSIM**	D_f	D_p
Brain	0.236621	6.26	0.1978	0.486437	0.179349
Image (a)	0.160123	7.96	0.2385	0.400154	0.132413

Applying the Back Projection (BP) algorithm to various medical images, including a phantom image, highlights several limitations in the reconstruction quality. The most noticeable issue is overall blurring, as each projected line is spread across the entire image. This causes overlapping of projections, resulting in poorly defined contours.

These artifacts are due to the uncorrected accumulation of projections. To address these issues, it is essential to apply frequency domain filtering. Therefore, to overcome the limitations of BP, we proceed with the implementation of the **Filtered Back Projection (FBP)** algorithm, which we will discuss in the next algorithm.

- **Filtered Back projection (FBP)**

In order to better analyze the projected data for the same previously examined images, we use the improved back projection (BP) technique, which incorporates a filtering phase prior to reconstruction. We employed a variety of filters in the reconstruction process to assess their

impact on the reconstructed image quality. Figure3.8 presents the implementation of the filtered back projection (FBP), from loading input image to result analysis.

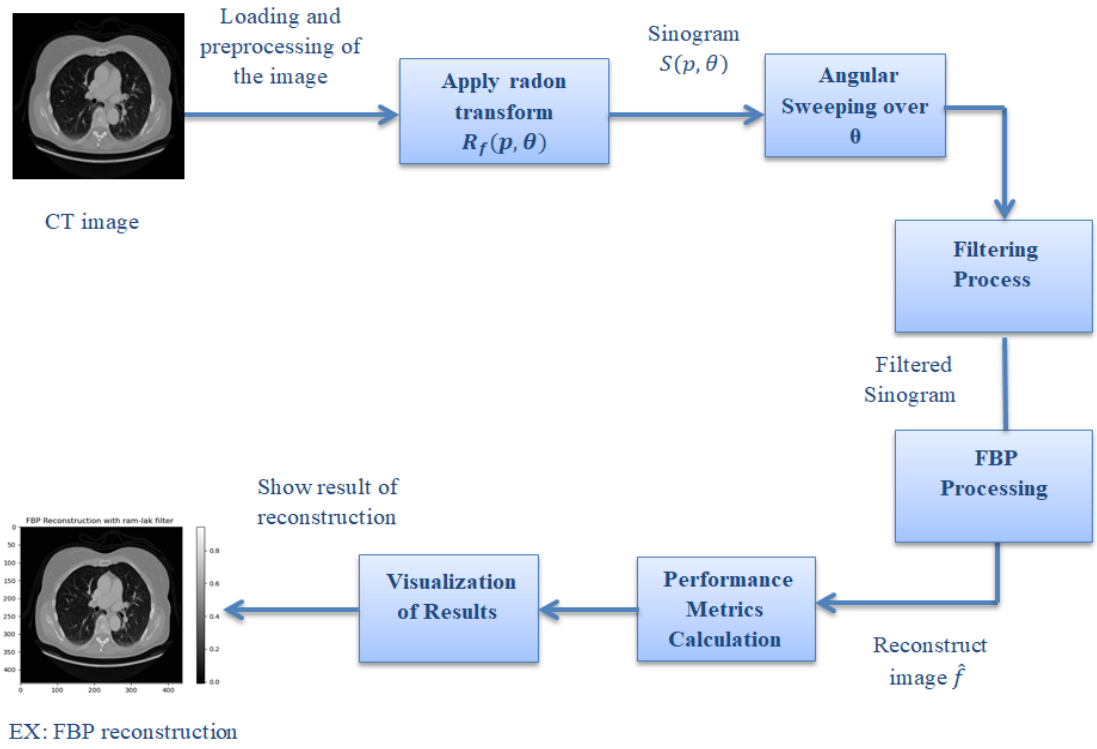


Figure 3-8: FBP Reconstruction Process for CT Imaging.

Where:

$\boldsymbol{R_f(\theta, s)}$ Denotes the **Radon transform** of the function $\boldsymbol{f(x, y)}$ as defined in Equation (1.1), $\boldsymbol{\theta}$ is the **projection angle** (with a total of 180 projections), and $\hat{f}(x, y)$ represents the **reconstructed image** from the projection data.

Figure.3.9 shows reconstruction result of phantom sheep Logan image using FBP with different filter

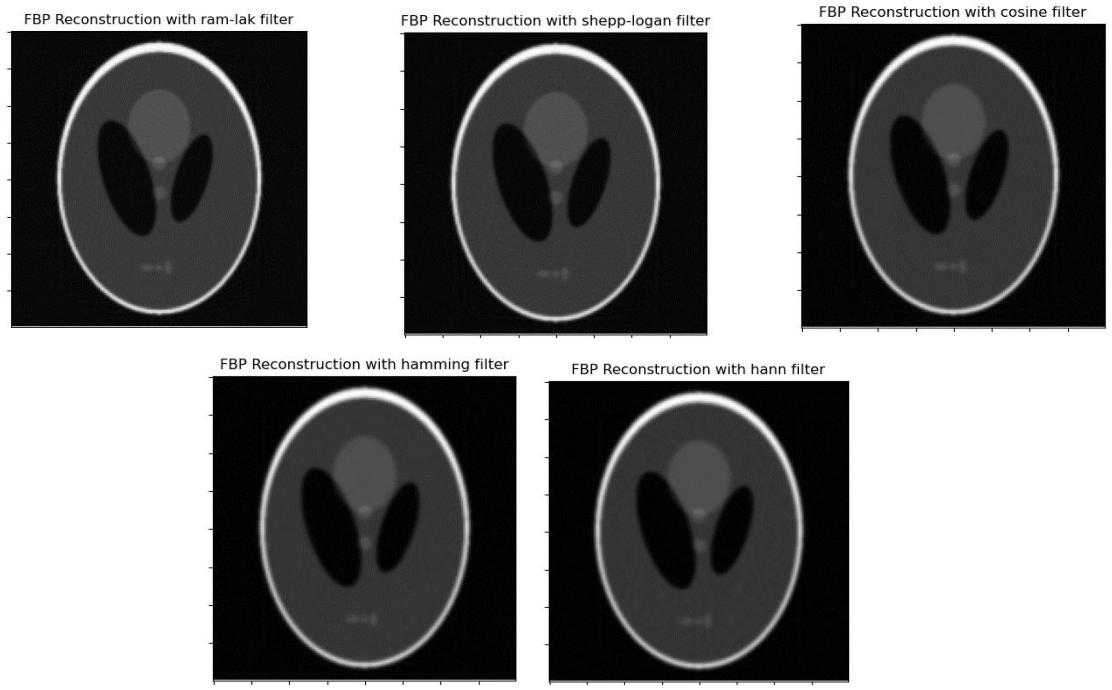


Figure 3-9 : FBP Reconstruction with: (a) ram-lak filter, (b) sheep-logan filter, (c) cosine filter, (d) hamming filter (d), (e) hanning filter

Applying different filters (Hamming, Shepp-Logan, Cosine, Hann, and Ram-Lak) to the same phantom image produces **relatively similar** reconstruction results overall.

Table 3.5 present the corresponding evaluation metrics of the different filters.

Table3. 5: Performance Comparison of Different Filters Applied to Phantom image.

Filters	**MSE**	**PSNR**	**SSIM**	D_f	D_p
Ram-lak	0.000799	30.972218	0.962401	0.028274	0.310928
Shepp-logan	0.001076	29.682187	0.963789	0.032801	0.377284
Cosine	0.001762	27.539138	0.959124	0.041980	0.508868
Hamming	0.002463	26.086015	0.947782	0.049625	0.638248
Hann	0.002660	25.751669	0.945026	0.051572	0.682484

The evaluation metrics in Table 3.5 show that the differences between the FBP filters are relatively small. Although the Ram-Lak and Shepp-Logan filters slightly outperform the others in terms of MSE, PSNR, and SSIM, the overall variations across all filters remain **moderate and**

not highly divergent. This indicates that all the tested filters produce comparable reconstruction quality, with only subtle distinctions in performance.

Figure.3.10 shows reconstruction result of brain image using FBP with different filter

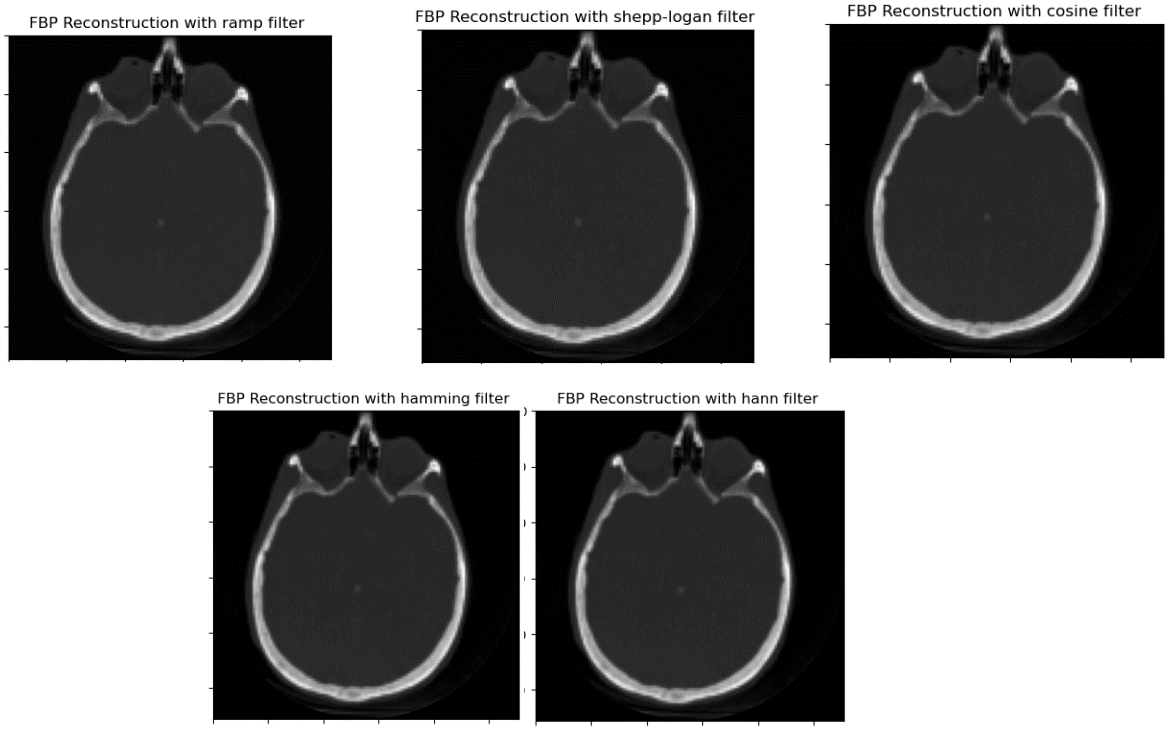


Figure 3-10: FBP Reconstruction with: (a) ram-lak filter, (b) sheep-logan filter, (c) cosine filter, (d) hamming filter (d), (e) hanning filter

Table 3.6 present the corresponding evaluation metrics of the different filters of brain image.

Table3. 6 : Performance Comparison of Different Filters Applied to brain image.

Filters	**MSE**	**PSNR**	**SSIM**	D_f	D_p
Ram-lak	0.000145	38.381659	0.962272	0.012048	0.594887
Shepp-logan	0.000154	38.120537	0.961983	0.012416	0.599096
Cosine	0.000178	37.499873	0.960600	0.013335	0.611222
Hamming	0.000203	36.922944	0.958770	0.014251	0.629442
Hann	0.000210	36.773770	0.958246	0.014498	0.633694

Figure.3.11 shows reconstruction result of image (a) using FBP with different filter

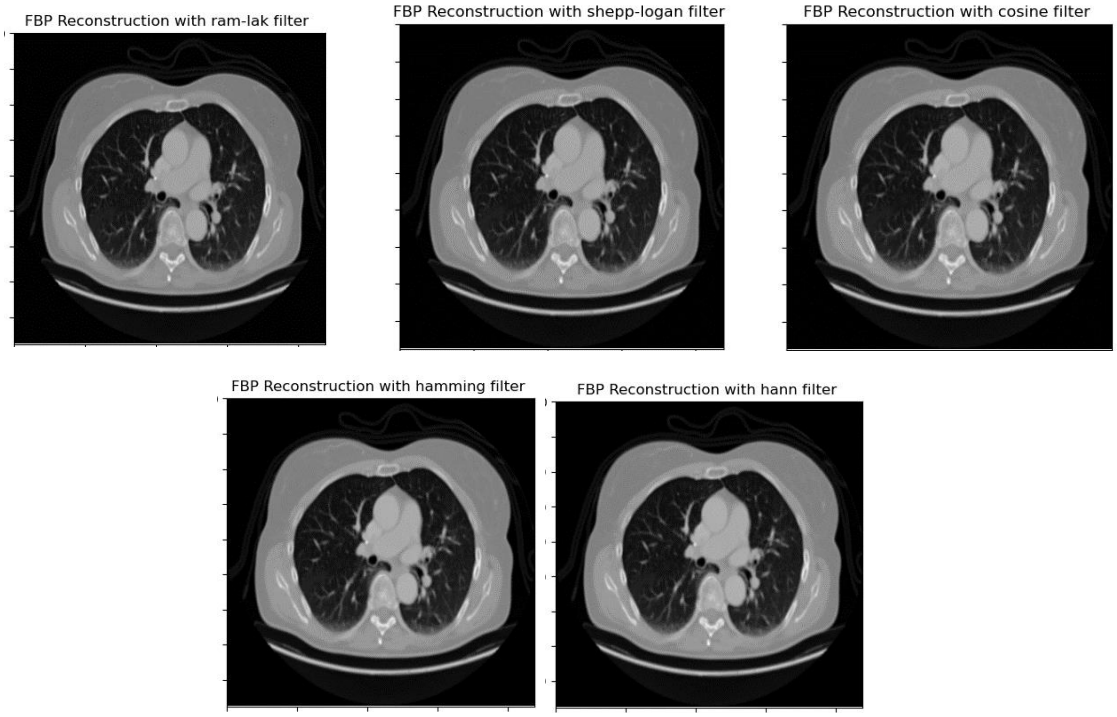


Figure 3-11: FBP Reconstruction with: (a) ram-lak filter, (b) sheep-logan filter, (c) cosine filter, (d) hamming filter (d), (e) hanning filter

After applying the Filtered Back Projection (FBP) algorithm with different filters, the reconstructed images appear visually similar in overall quality, exhibiting only slight variations. This indicates that all filters achieve effective image restoration, maintaining satisfactory levels of contrast and sharpness. The table 3.5 confirms this observation, as the evaluation metrics show very close values, underscoring the robustness of the FBP method regardless of the filter used. The reconstructed image is significantly enhanced compared to BP;

Table3.7 presents the corresponding evaluation metrics of the different filters of brain image.

Table3. 7: Performance Comparison of Different Filters Applied to image (a).

Filters	**MSE**	**PSNR**	**SSIM**	D_f	D_p
Ram-lak	0.000066	41.806708	0.979666	0.008122	0.213303
Shepp-logan	0.000082	40.841074	0.977473	0.009077	0.241246
Cosine	0.000125	39.036659	0.971474	0.011173	0.302051
Hamming	0.000179	37.471398	0.964240	0.013379	0.373336
Hann	0.000193	37.149369	0.962304	0.013885	0.388488

Syntheses

In this first part of our simulation study, we explored two fundamental analytical techniques in computed tomography (CT): **Simple Back Projection (BP)** and **Filtered Back Projection (FBP).** The objective was to compare their reconstruction performance using the Shepp-Logan Phantom Brain and medical images as a reference images well as and quantitative evaluation metrics. Back Projection (BP) method was directly applied to the sinogram obtained via Radon transform. Although it provides a quick initial estimate of the image, BP is well known for introducing significant circular artifacts and blurring effects, which degrade image quality and limit its applicability in precise medical or industrial imaging. The computed metrics, we have obtained: Mean Squared Error (MSE) and the Peak Signal-to- Noise Ratio (PSNR), revealed high error rate and low reconstruction accuracy, confirming the method's limitations. Then, the Filtered Back Projection (FBP) method was employed, integrating a crucial enhancement: frequency-domain filtering of the sinogram prior to back projection. This filtering step compensates for the excessive accumulation of low frequencies inherent in simple back projection, thus improving image sharpness and reducing artifacts .Various filters were tested, including Ram-Lak, Shepp-Logan, Cosine, Hamming, and Hann, each exhibiting unique characteristics in terms of noise suppression and detail preservation. The experimental results demonstrated that the Ram-Lak and Shepp-Logan filters achieved the best reconstructions, with PSNR values exceeding 30 dB and SSIM (Structural Similarity Index) values approaching 1, indicating a strong structural resemblance with the original image. In contrast, the Cosine, Hamming, and Hann filters provided smoother images with but at the cost of slightly diminished fine details. Additionally, the analysis of the distances D_f and D_p further validated these observations, with smaller discrepancies observed between the reconstructed and original images when employing the most effective filters. These findings highlight the importance of appropriate filter selection in optimizing the trade-off between blur reduction and image fidelity in FBP reconstructions.

This analysis provides a solid foundation for the next stage of our work, focusing on the implementation and evaluation of iterative optimization algorithms such as **SART (Simultaneous Algebraic Reconstruction Technique)** and **MLEM (Maximum Likelihood Expectation Maximization)**, which will be studied as part of the algebraic reconstruction methods.

3.4.2 Algebraic algorithm

In this section, we present the implementation of two algebraic iterative reconstruction algorithms: SART (Simultaneous Algebraic Reconstruction Technique) and MLEM (Maximum Likelihood Expectation Maximization). These algorithms are known for improving reconstruction quality, especially when the number of projections is limited or the data is noisy. For the sake of comparison, we use the same tested images.

❖ **Simultaneous Algebraic Reconstruction Technique**

We applied the SART algorithm over several iterations, to better illustrate the implementation of the SART algorithm, the following block diagram figure 3.12 summarizes its main steps

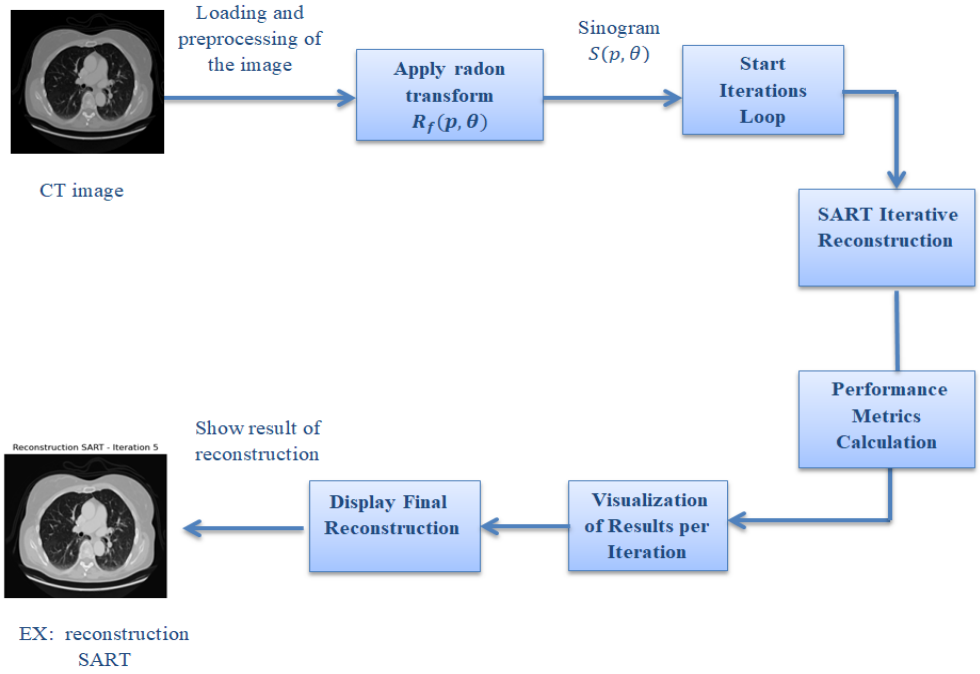


Figure 3-12 : SART Reconstruction Process for CT Imaging.

First, the original image is loaded, resized to 256x256 pixels, and then masked with a circular mask to remove areas outside the region of interest. Next, the Radon transform is applied to generate the sinogram, the reconstruction is performed using the iterative SART method over 5 iterations. At each step, the reconstructed image is displayed, and several quantitative metrics are calculated to evaluate the reconstruction quality. Figure 3.13 represent reconstruction result of the Phantom sheep-logan image using the SART algorithm

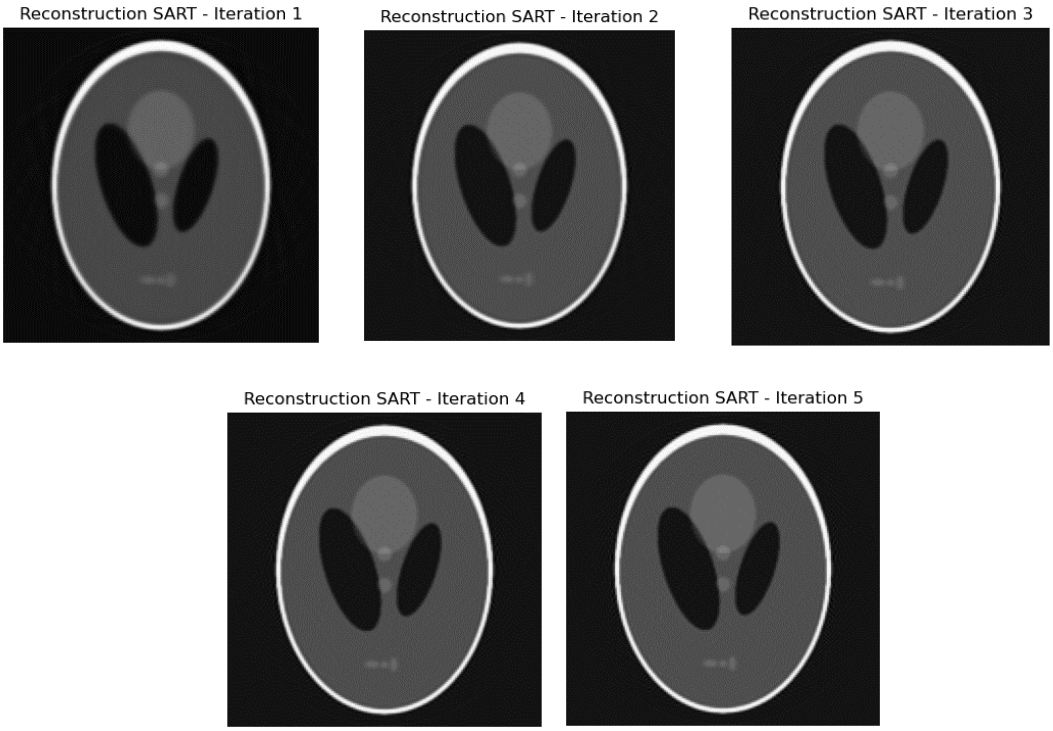


Figure 3-13: Reconstruction Progression of Phantom Shepp-Logan using SART Algorithm (Iterations 1 to 5)

We fixed the number of SART iterations to **5**, which corresponds to the optimal case observed in our experiments. Beyond this point, increasing the number of iterations tends to **degrade the image quality** rather than improve it. Excessive iterations may amplify noise and introduce artifacts, leading to image distortion. This confirms that **SART requires a limited number of iterations**, and exceeding this threshold negatively affects the accuracy and visual quality of the reconstructed image. Performance metrics were calculated for each iteration as shown in table 3.8

Table3. 8: Performance Comparison of Different iteration Applied to phantom Shepp-Logan image using SART.

Iteration	MSE	PSNR (dB)	SSIM	D_f	D_p
1	0.001084	29.650164	0.959590	0.032922	0.346770
2	0.000457	33.400720	0.971232	0.021378	0.159801
3	0.000313	35.050645	0.970357	0.017679	0.105494
4	0.000258	35.883899	0.967582	0.016062	0.081993
5	0.000232	36.351179	0.964571	0.015221	0.069368

The results in the table 3.8 show that **image quality improves steadily with each additional iteration up to iteration 5**. Metrics such as MSE decrease, while PSNR and SSIM increase, reflecting better reconstruction accuracy and visual fidelity. This indicates that up to five iterations, SART progressively enhances the image. Beyond this point, however, increasing the number of iterations tends to degrade the quality due to noise amplification or overcorrection. Therefore, **five iterations can be considered an optimal balance** between reconstruction quality and algorithm stability. Figure 3.14 represent reconstruction result of the brain image using the SART algorithm

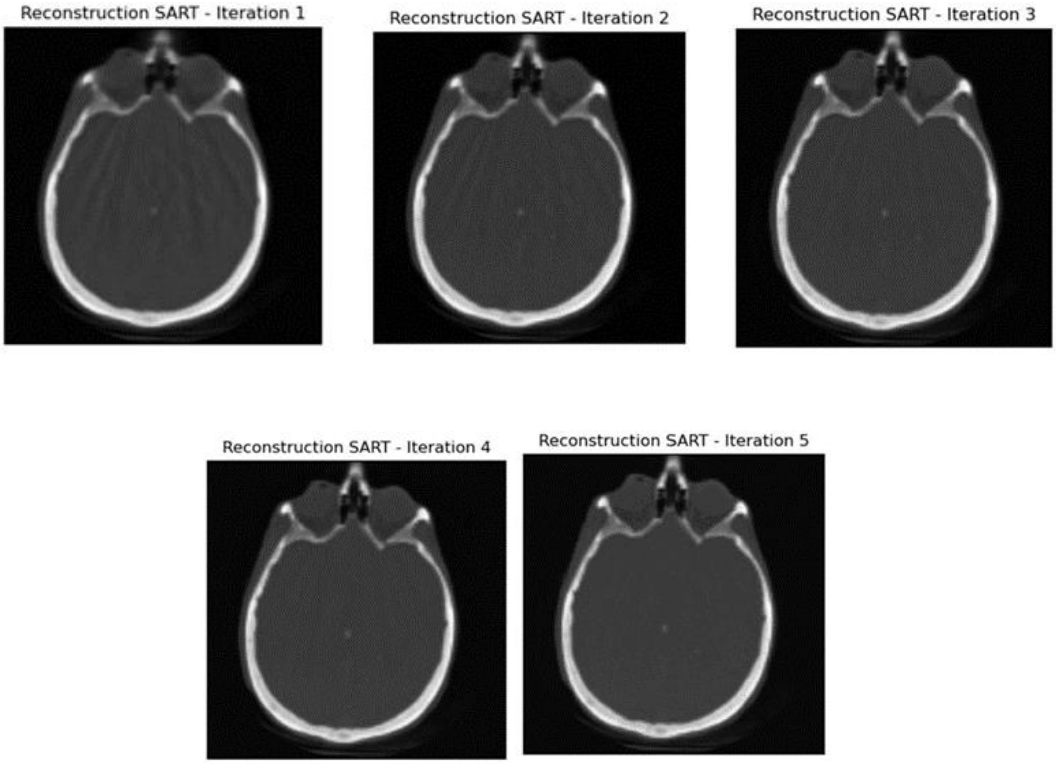


Figure.3-14: Reconstruction Progression of brain using SART Algorithm (Iterations 1 to 5).

Performance metrics were calculated for each iteration as shown in table 3.9

Table3.9 : Performance Comparison of Different iteration Applied to brain image using SART algorithm.

Iteration	MSE	PSNR (dB)	SSIM	D_f	D_p
1	0.000393	33.952114	0.962587	0.019815	0.583161
2	0.000183	37.267595	0.980272	0.013528	0.319012
3	0.000116	39.253679	0.986009	0.010763	0.207116
4	0.000086	40.553825	0.987901	0.009267	0.149191
5	0.000070	41.454894	0.988658	0.008353	0.115724

Figure 3.15 represent reconstruction result of the image (a) using the SART algorithm

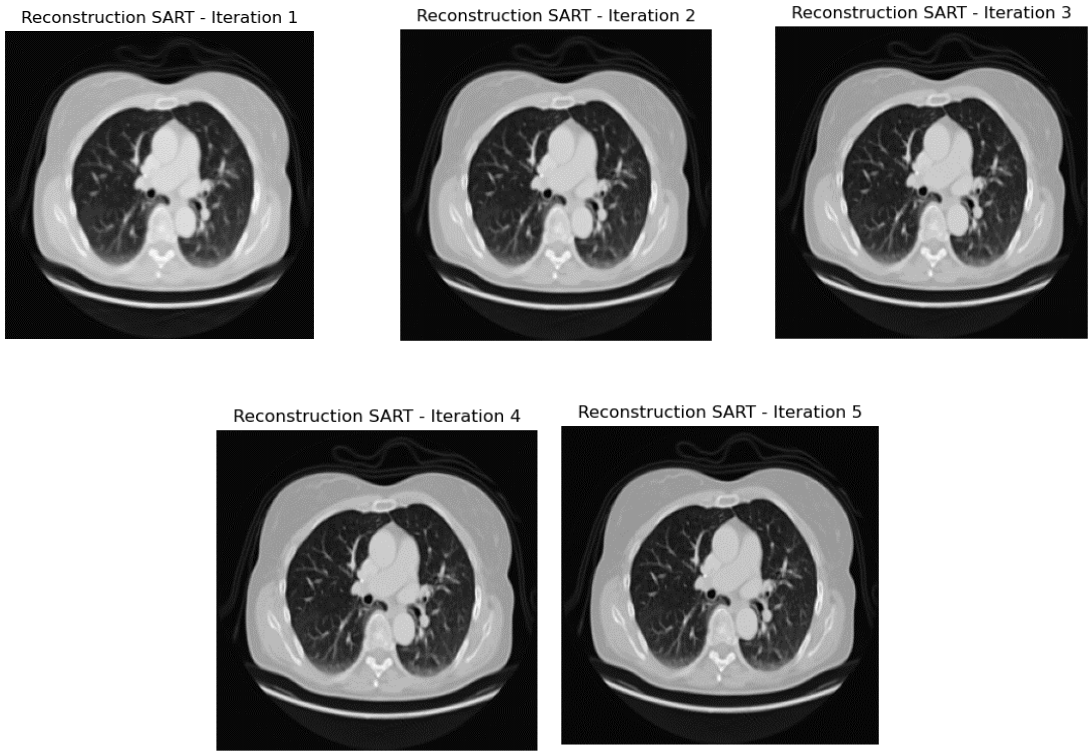


Figure 3-15: Reconstruction Progression of image (b) using SART Algorithm (Iterations 1 to 5).

Performance metrics were calculated for each iteration as shown in table 3.10

Table3.10: Performance Comparison of Different iteration Applied to image (a) using SART algorithm.

Iteration	MSE	PSNR(dB)	SSIM	D_f	D_p
1	0.000413	33.441428	0.940709	0.020320	0.400779

2	0.000156	37.663179	0.969957	0.012498	0.201729
3	0.000090	40.046239	0.978624	0.009499	0.132038
4	0.000064	41.523489	0.982309	0.008014	0.098624
5	0.000052	42.476181	0.984171	0.007181	0.079965

The results obtained with the SART method demonstrate that this technique enables progressive image reconstruction, with quality improving at each iteration. However, several limitations become evident. The method is sensitive to noise in the input data, which can cause the amplification of artifacts if the number of iterations is excessive. Additionally, selecting the optimal number of iterations is challenging: too few iterations lead to incomplete reconstructions, while too many can degrade image quality by enhancing noise and artifacts. Furthermore, SART may struggle when dealing with incomplete or highly noisy data, which limits its robustness in practical acquisition scenarios. These observations underscore the necessity to enhance or complement SART with more robust and better-suited algorithms, such as the Maximum Likelihood Expectation Maximization (MLEM) method, which offers improved noise management and superior reconstruction fidelity.

3.4.3 Maximum Likelihood Expectation Maximization (MLEM)

The MLEM algorithm is then implemented using an optimized iterative method. As before, the same sinogram is used, and the reconstruction is performed over 500 iterations, the following block diagram figure 3.16 summarizes its main steps:

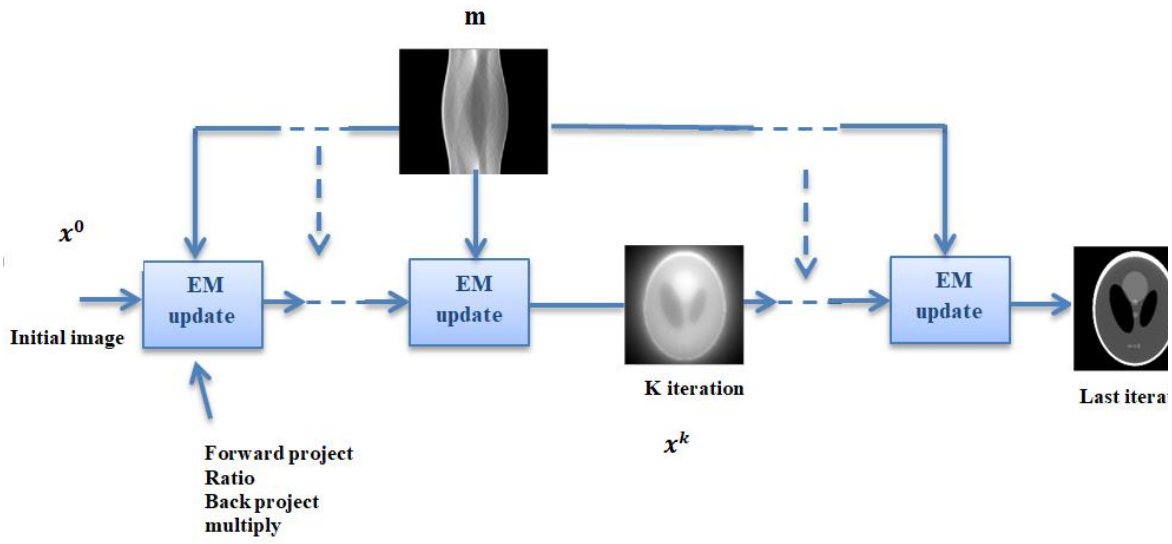


Figure 3-16 : block diagram of MLEM reconstruction image.

After loading and preprocessing the image (resizing, and applying a circular mask to limit the region of interest), the sinogram is generated using the Radon transform. It starts with an initial image (often a uniform image) and improves it step by step in each iteration. In every step, the current image is projected to estimate the sinogram. Then, a ratio is calculated between the measured sinogram and the estimated one. This ratio is back projected and used to update the image. The update is normalized. This cycle is repeated several times (as shown in the diagram) to get a higher-quality final image. In the diagram, each "EM update" block represents one iteration, and we can observe how the image evolves from the initial guess to the final reconstructed version.

For the sake of clarity, we report the basic MLEM equation for the actual iteration as follows:

$$x^{(k+1)} = \frac{x^k}{A^T 1} A^T \frac{m}{Ax^k} \tag{3.1}$$

Where:

m: the measured data (sinogram).

A: the forward projection matrix.

A^T: The back projection operator (transpose of$A$).

x^k : The current image estimate at iteration k.

$x^{(K+1)}$: The updated image estimate at iteration $k + 1$.

1: a vector of ones (used to compute the sensitivity image).

Figure 3.17 represent reconstruction result of the phantom Shepp Logan image using the MLEM algorithm.

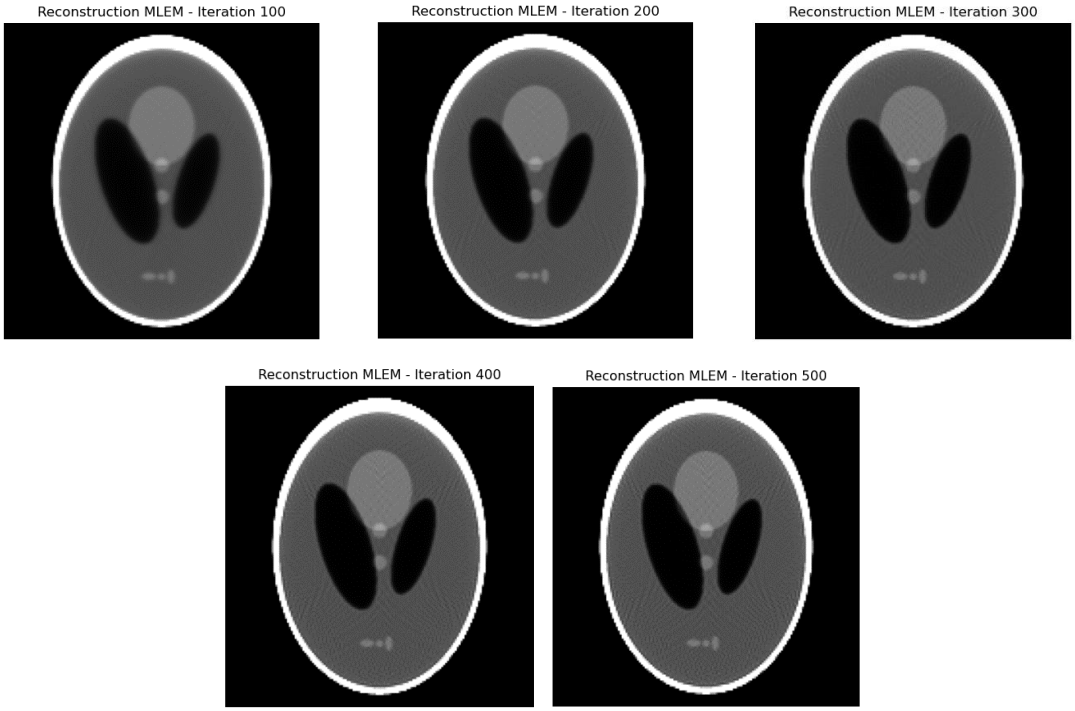


Figure 3-17 : Reconstruction Progression of phantom shepp Logan image using MLEM Algorithm (Iterations 100 to 500).

Performance metrics were calculated for each iteration as shown in table 3.11

Table3. 11 : Performance Comparison of Different iteration Applied to phantom shepp Logan image using MLEM algorithm.

Iteration	MSE	PSNR	SSIM	D_f	D_p
100	0.000511	32.918357	0.933407	0.022599	0.394337
200	0.000229	36.409740	0.958278	0.015119	0.215541
300	0.000142	38.485315	0.970658	0.011905	0.150779
400	0.000100	39.987988	0.977837	0.010014	0.116367
500	0.000077	41.137241	0.982307	0.008773	0.094890

The data in Table 3.13 highlight the efficiency of the MLEM algorithm in reconstructing high-quality images. As the number of iterations increases from 100 to 500, the image quality improves steadily, with lower MSE and higher PSNR and SSIM values. Although MLEM is computationally more demanding and requires more time than other methods, it offers **superior reconstruction performance**. Its main strengths lie in its **robustness to noise**, **gradual convergence**, and **excellent detail preservation**, making it highly suitable for accurate medical image reconstruction.

Figure 3.18 represent reconstruction result of the brain image using the MLEM algorithm

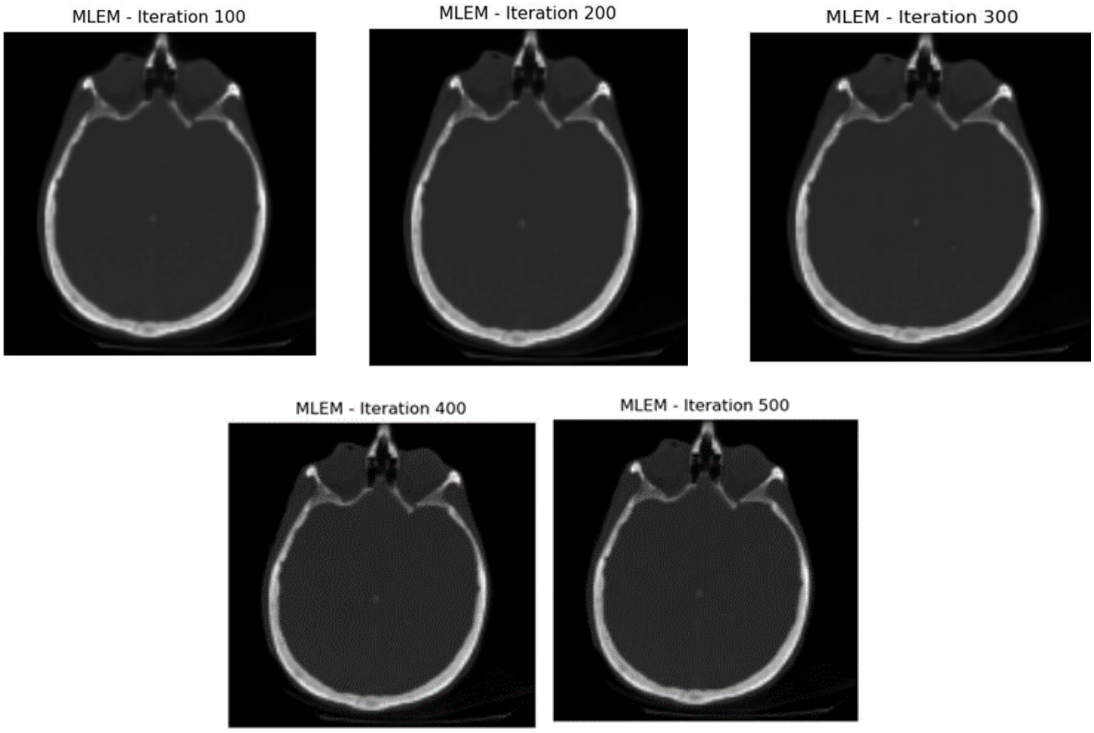


Figure 3-18 : Reconstruction Progression of brain image using MLEM Algorithm (Iterations 100 to 500).

Performance metrics were calculated for each iteration as shown in table 3.12

Table3. 12 : Performance Comparison of Different iteration Applied to brain image using MLEM algorithm.

Iteration	MSE	PSNR	SSIM	D_f	D_p
100	0.000223	36.524456	0.970467	0.014599	0.299411
200	0.000099	40.024591	0.986780	0.009119	0.148988
300	0.000064	41.911568	0.990073	0.008905	0.101506
400	0.000048	43.227852	0.992381	0.006814	0.078114
500	0.000038	41.137241	0.993704	0.006141	0.063928

Figure 3.19 represent reconstruction result of the image (a) using the MLEM algorithm

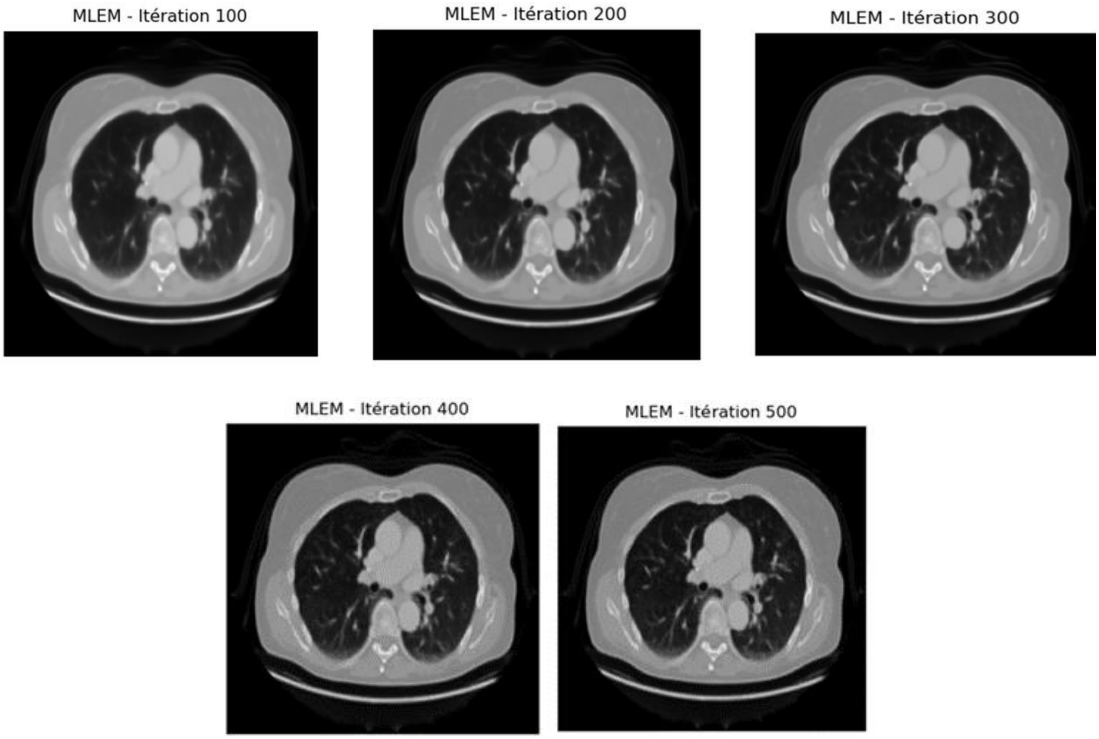


Figure 3-19 : Reconstruction Progression of image (a) using MLEM Algorithm (Iterations 100 to 500).

Performance metrics were calculated for each iteration as shown in table 3.13

Table3. 13 : Performance Comparison of Different iteration Applied to image (a) using MLEM algorithm.

Iteration	**MSE**	**PSNR**	**SSIM**	D_f	D_p
100	0.000380	34.289343	0.941387	0.019260	0.324555
200	0.000140	37.225484	0.974470	0.011441	0.158337
300	0.000080	39.565299	0.980959	0.009662	0.106742
400	0.000058	41.270995	0.983473	0.007362	0.071117
500	0.000045	42.670122	0.984542	0.006672	0.063923

Based on the results obtained and the literature, the Maximum Likelihood Expectation Maximization (MLEM) algorithm stands out as one of the most effective iterative reconstruction methods in tomography. MLEM is particularly efficient at producing high-quality images even when the projection data (sinogram) are noisy or incomplete. Unlike simpler methods, MLEM

incorporates a statistical model of the data acquisition process, allowing it to better manage noise and reduce artifacts in the reconstructed images. Although it requires more computational time due to slower convergence compared to methods like SART, MLEM consistently delivers superior image fidelity by preserving fine details while suppressing noise. Overall, its robustness, noise resilience, and high reconstruction quality make MLEM the preferred choice for accurate tomographic imaging, despite its higher computational cost.

Syntheses

In this study, we focused on two major iterative image reconstruction techniques widely used in computed tomography: the **Simultaneous Algebraic Reconstruction Technique (SART)** and the **Maximum Likelihood Expectation Maximization (MLEM)** algorithm. The SART algorithm, an improvement of the Algebraic Reconstruction Technique (ART), achieves fast convergence through successive pixel updates based on angular projection errors, providing good reconstruction quality with a moderate number of iterations. This speed makes it suitable for scenarios where computational efficiency is critical.

However, **SART** has several notable drawbacks. It is sensitive to noise in the input data, which can lead to artifact amplification if the number of iterations is too high. Choosing the optimal number of iterations is challenging: too few iterations result in incomplete reconstruction, while too many can degrade image quality. Additionally, **SART** may lack robustness when dealing with incomplete or highly noisy data, limiting its effectiveness in real acquisition conditions.

Conversely, the **MLEM** algorithm, although slower and more computationally demanding, stands out for its superior noise-handling capabilities and ability to produce high-quality images. By incorporating a precise statistical model of the acquisition process, **MLEM** significantly reduces artifacts and preserves fine details even under challenging conditions. This trade-off between computation time and image quality makes MLEM the preferred method when accuracy and robustness are essential, particularly in demanding clinical applications.

Thus, despite its limitations, **SART** remains useful for rapid reconstructions, but **MLEM** establishes itself as the best method for optimal results in computed tomography.

3.5 Algorithms of Deep Learning FBP

This section presents and compares the performance of Filtered back projection (FBP) CT reconstruction algorithms augmented with deep learning approaches. Two hybrid techniques are compared:

- Filtered Back Projection followed by a CNN denoisng **(FBP + DnCNN).**
- Learnable Filtered Back Projection integrated with CNN (**Learnable FBP).**

Each approach is evaluated under varying levels of Gaussian noise to assess its robustness and image quality.

3.5.1 Architecture of FBP+DnCNN

In order to improve the quality of obtained reconstructions using the analytical Filtered Back Projection (FBP), we have adopted a hybrid approach combining FBP with a convolutional neural network (CNN) that plays the role of a denoiser. This technique makes it possible to compensate for the noise and artifacts caused by direct reconstruction. Figure 3.20 shows the block diagram for this FBP-CNN architecture.

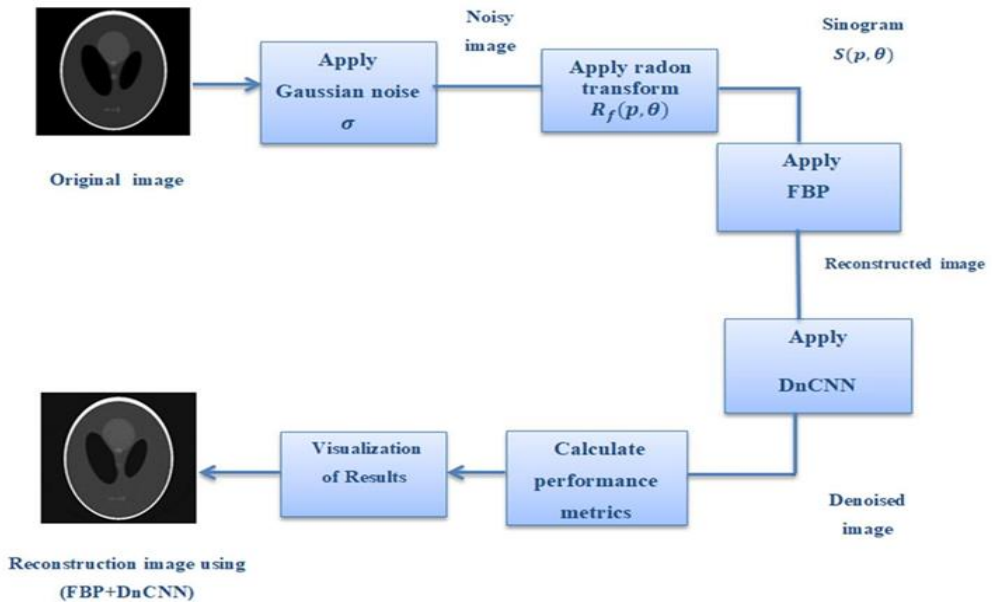


Figure 3-20 : the block diagram for FBP+DnCNN architecture

This algorithm begins by generating original image and adding **Gaussian noise** to simulate different noise levels. It then applies the **Filtered Back Projection (FBP)** method to reconstruct

the image from its noisy sinogram. To enhance the quality of the reconstruction, a **Convolutional Neural Network (CNN)** is trained to learn the mapping between the noisy FBP output and the original clean image. Finally, several **quantitative metrics** (MSE, PSNR, SSIM,D_f,D_p) are computed to evaluate and compare the performance of both the FBP and CNN results.

The convolutional neural network (CNN) used in this study is simple and effective for removing noise from CT images reconstructed with FBP, as shown in Figure 3.21. It has four convolutional layers. The first three use 32 filters of size 3×3. The last layer also uses a 3×3 filter but with no activation and gives output image. This CNN acts like a trainable filter that reduces noise and artifacts from the FBP result.

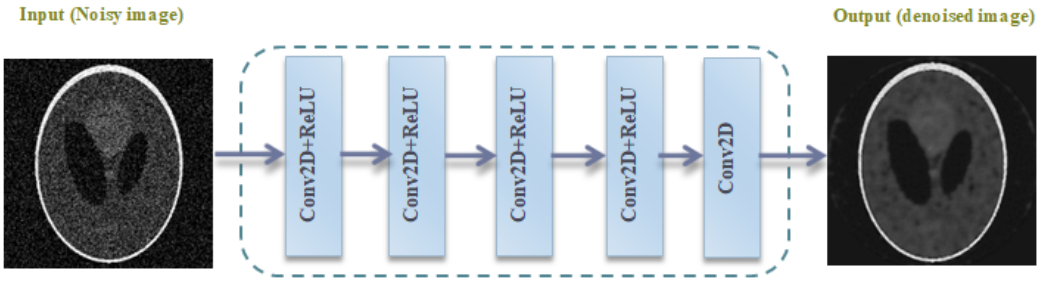
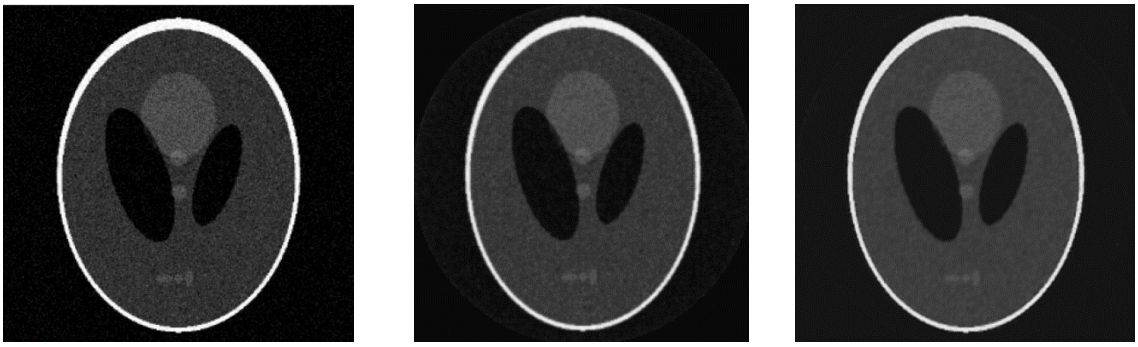


Figure 3-21 : Archirecture of CNN used in FBP+DnCNN.

The figures below illustrate the reconstruction results of the Phantom Sheep-Logan image using the conventional FBP algorithm and the enhanced FBP+DnCNN method.

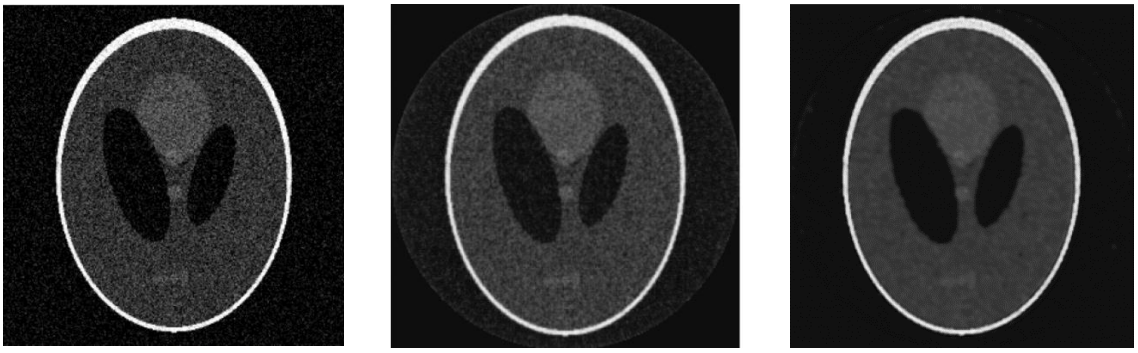
- **Case 1: Gaussian noise with standard deviation σ= 0.05**



Noisy image σ **FPB reconstruction** **CNN denoiser**

Figure 3-22 : reconstruction results of the Phantom Sheep-Logan image with Gaussian noise (σ=0.05), showing the noisy image, FBP reconstruction, and CNN denoising.

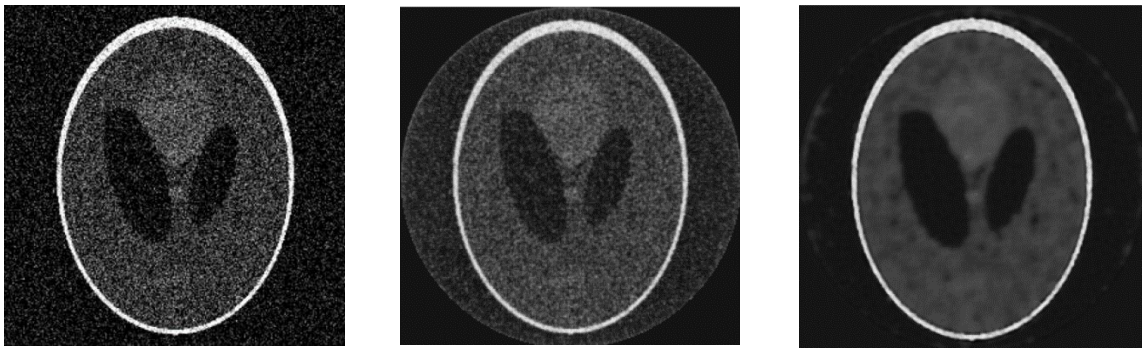
- **Case 2: Gaussian noise with standard deviation σ= 0.1**



Noisy image σ **FBP reconstruction** **CNN denoiser**

Figure 3-23 : reconstruction results on the Phantom Sheep-Logan image with Gaussian noise (σ=0.1), showing the noisy image, FBP reconstruction, and CNN denoising.

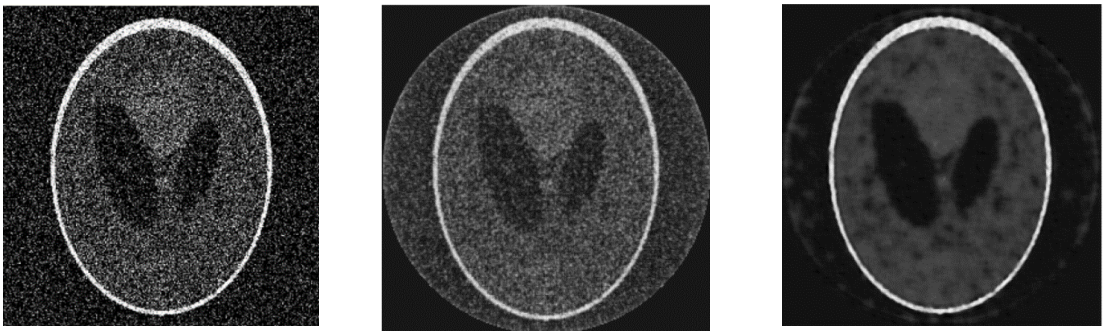
- **Case 3: Gaussian noise with standard deviation σ= 0.2**



Noisy image σ **FBP reconstruction** **CNN denoiser**

Figure 3-24: reconstruction results on the Phantom Sheep-Logan image with Gaussian noise (σ=0.2), showing the noisy image, FBP reconstruction, and CNN denoising.

- **Case 4: Gaussian noise with standard deviation σ= 0.3**



Noisy image σ **FBP reconstruction** **CNN denoiser**

Figure 3-25 reconstruction results on the Phantom Sheep-Logan image with Gaussian noise (σ=0.3), showing the noisy image, FBP reconstruction, and CNN denoising.

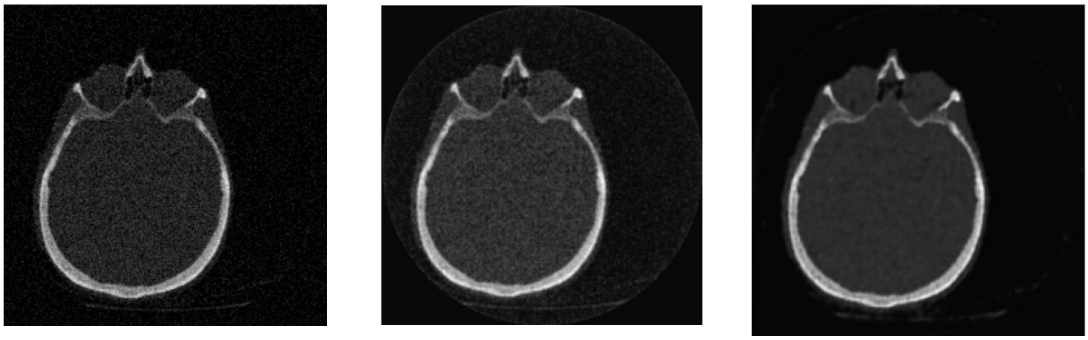
Visually, it is observed that the images reconstructed using the FBP method degrades as the noise level increases: edges become blurred and internal details fade, especially for σ = 0.2 and 0.3. In contrast, the CNN-based reconstructions maintain significantly better visual quality, with sharper edges and more faithful structure, even under noisy conditions. This demonstrates the **CNN's ability to suppress noise while preserving important image details**. This visual observation is **confirmed by the table3.14**.

Table3.14: Quantitative comparison between FBP and FBP+DnCNN reconstruction for phantom Sheep-logan image.

Methods	**FBP reconstruction**					**CNN denoised reconstruction**				
Metrics	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
σ=0.05	0.001521	28.18	0.5596	0.1322	0.1609	0.000465	33.32	0.9108	0.0763	0.0890
σ=0.1	0.003472	24.59	0.4102	0.2430	0.1821	0.000885	30.53	0.8896	0.1227	0.1037
σ=0.2	0.010365	19.84	0.3201	0.4199	0.3185	0.001477	28.31	0.7744	0.1585	0.1211
σ=0.3	0.021145	16.75	0.2836	0.5997	0.4721	0.002737	25.63	0.6408	0.2158	0.1651

The figures below illustrate the reconstruction results of the brain image using the conventional FBP algorithm and the enhanced FBP+DnCNN method.

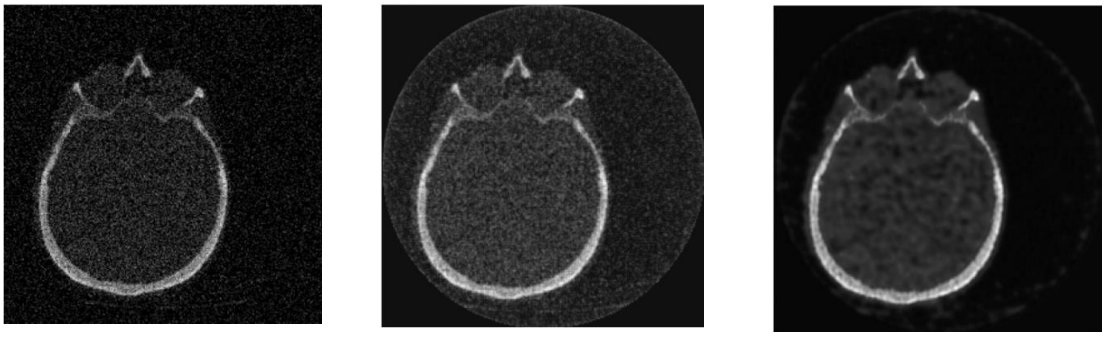
- **Case 1: Gaussian noise with standard deviation σ= 0.05**



Noisy image σ **FBP reconstruction** **CNN denoiser**

Figure 3-26: reconstruction results on the brain image with Gaussian noise (σ=0.05), showing the noisy image, FBP reconstruction, and CNN denoising.

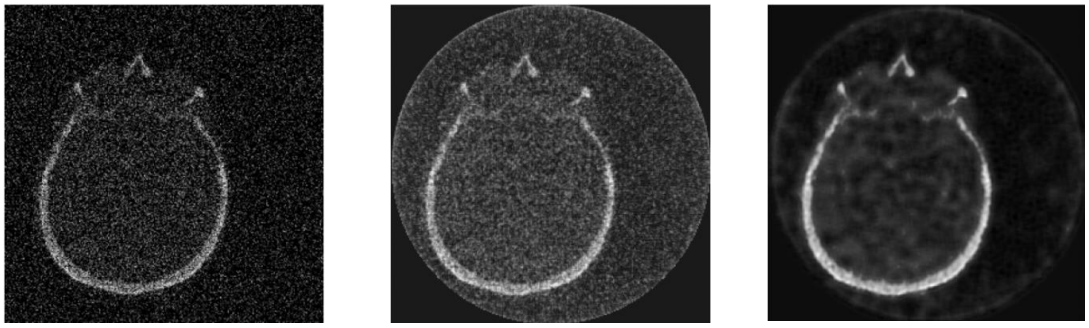
- **Case 2: Gaussian noise with standard deviation σ= 0.1**



Noisy image **FBP reconstruction** **CNN denoiser**

Figure 3-27: reconstruction results on the brain image with Gaussian noise (σ=0.1), showing the noisy image, FBP reconstruction, and CNN denoising.

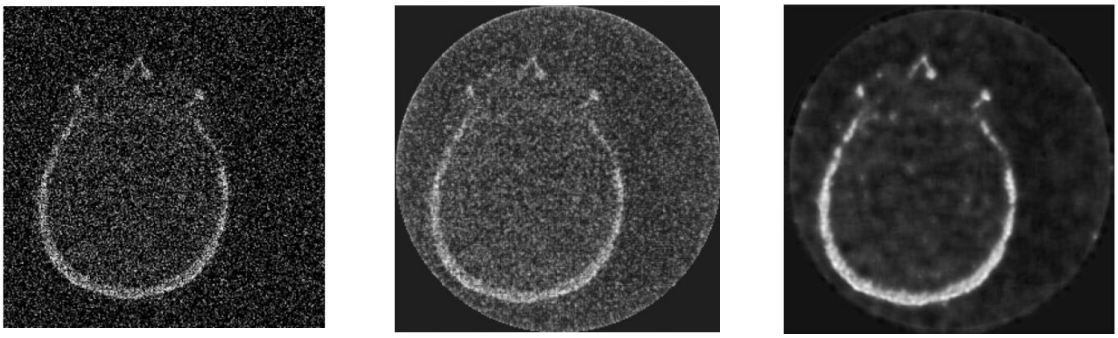
- **Case 3: Gaussian noise with standard deviation σ= 0.2**



Noisy image **FBP reconstruction** **CNN denoiser**

Figure 3-28 : reconstruction results on the brain image with Gaussian noise (σ=0.2), showing the noisy image, FBP reconstruction, and CNN denoising.

- **Case 4: Gaussian noise with standard deviation σ= 0.3**



Noisy image **FPB reconstruction** **CNN denoiser**

Figure 3-29 : reconstruction results on the brain image with Gaussian noise (σ=0.3), showing the noisy image, FBP reconstruction, and CNN denoising.

Table3.15: Quantitative comparison between FBP and FBP+DnCNN reconstruction for Brain image.

methods	**FBP reconstruction**					**CNN denoised reconstruction**				
metrics	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
σ=0.05	0.000665	31.77	0.5539	0.2217	0.1692	0.000128	38.94	0.9376	0.0972	0.0718
σ=0.1	0.002409	26.18	0.3935	0.4219	0.3320	0.000303	35.19	0.8740	0.1495	0.1046
σ=0.2	0.009104	20.41	0.2833	0.8202	0.6785	0.001211	29.17	0.6722	0.2991	0.2164
σ=0.3	0.020220	16.94	0.2360	1.2223	1.0462	0.002001	26.99	0.5494	0.3845	0.2623

Simulated images like the Shepp-Logan phantom are perfect, which allows us to introduce large levels of noise to test algorithm robustness. Real CT pictures, on the other hand, already have noise due to patient movements, metal artifacts, low radiation dose, etc. The image is severely deteriorated by adding extra noise, which makes restoration more challenging. Since noise is cumulative, even a tiny σ number can significantly influence genuine photos. For this reason, we did not apply the same noise levels to the brain image, the Shepp-Logan phantom, and the real CT scans.

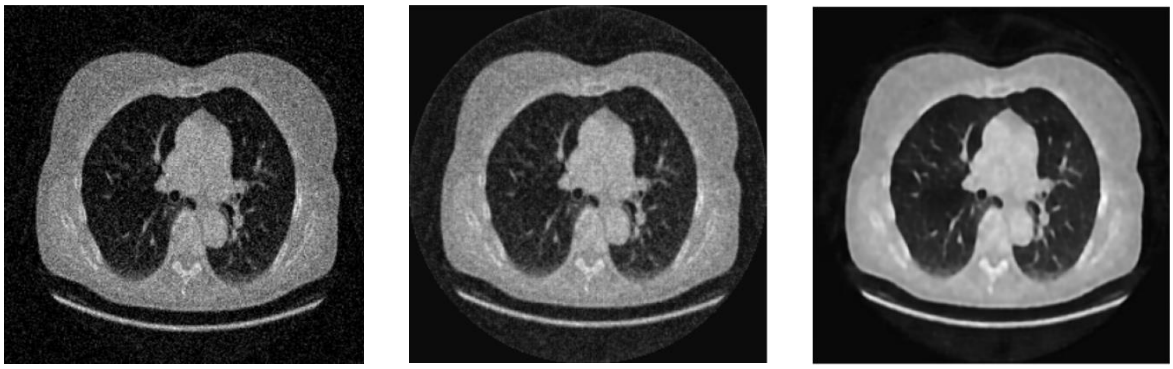
The figures below illustrate the reconstruction results of the image (a) using the conventional FBP algorithm and the enhanced FBP+DnCNN method.

- **Case 1: Gaussian noise with standard deviation σ= 0.01**

Noisy image **FPB reconstruction** **CNN denoiser**

Figure 3-30 : reconstruction results on the image (a) with Gaussian noise (σ=0.01), showing the noisy image, FBP reconstruction, and CNN denoising.

- **Case 2: Gaussian noise with standard deviation σ= 0.04**



Noisy image **FPB reconstruction** **CNN denoiser**

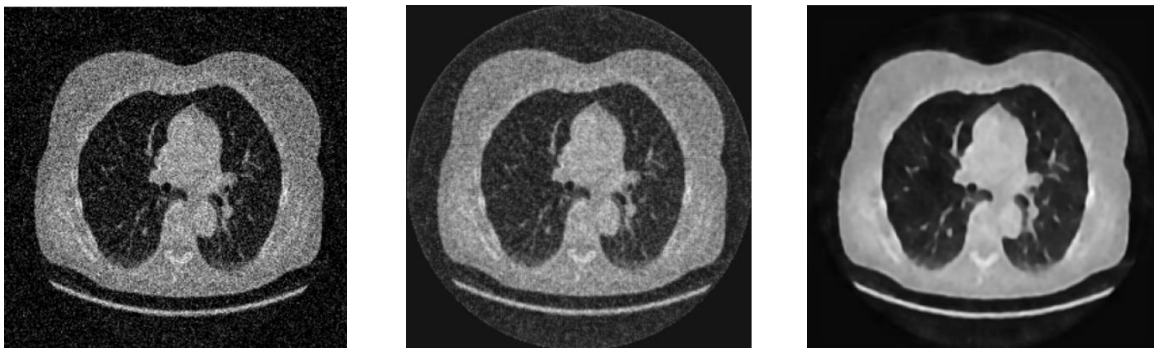
Figure 3-31 : reconstruction results on the image (a) with Gaussian noise (σ=0.04), showing the noisy image, FBP reconstruction, and CNN denoising.

- **Case 3: Gaussian noise with standard deviation σ= 0.07**

Noisy image **FPB reconstruction** **CNN denoiser**

Figure 3-32 : reconstruction results on the image (a) with Gaussian noise (σ=0.07), showing the noisy image, FBP reconstruction, and CNN denoising.

- **Case 4: Gaussian noise with standard deviation σ= 0.1**



Noisy image **FPB reconstruction** **CNN denoiser**

Figure 3-33 : reconstruction results on the image (a) with Gaussian noise (σ=0.1), showing the noisy image, FBP reconstruction, and CNN denoising.

Table3.16: Quantitative comparison between FBP and FBP+DnCNN reconstruction for image (a).

Methods	**FBP reconstruction**					**CNN denoised reconstruction**				
Metrics	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
σ=0.005	0,001455	28,37	0,6760	0,0850	0,1234	0,000500	33,01	0,8750	0,0531	0,0724
σ=0.009	0,002409	26,18	0,6013	0.1122	0.1588	0.000589	32,30	0.8580	0.0569	0.0786
σ=0.01	0.002687	25,71	0.5854	0,1190	0.1678	0.000602	32,20	0.8531	0.0575	0.0794
σ=0.02	0.005075	22,95	0.4974	0.1706	0.2306	0.000912	30,40	0.8140	0.0718	0.0977

The FBP+DnCNN method demonstrates a significant improvement in reconstruction quality, even under high noise conditions. Thanks to the denoising step using a convolutional neural network (CNN), the reconstructed images show greater sharpness, structural fidelity, and contrast. Metrics such as PSNR and SSIM confirm this enhanced performance. This approach effectively combines the speed of FBP with the strength of deep learning to overcome the limitations of traditional FBP in noisy environments.

3.5.2 Learnable FBP architecture

In this simulation section, we implement **Learnable FBP**, which combines the filtered FBP with an appreciable filter using a convoluted neural network (CNN). In contrast to the traditional approach, where the filter (such as the Ram-Lak filter) is fixed and determined analytically, here CNN learns to optimally filter the sinogram from the data. Figure3. 34 show the principal steps in this algorithm:

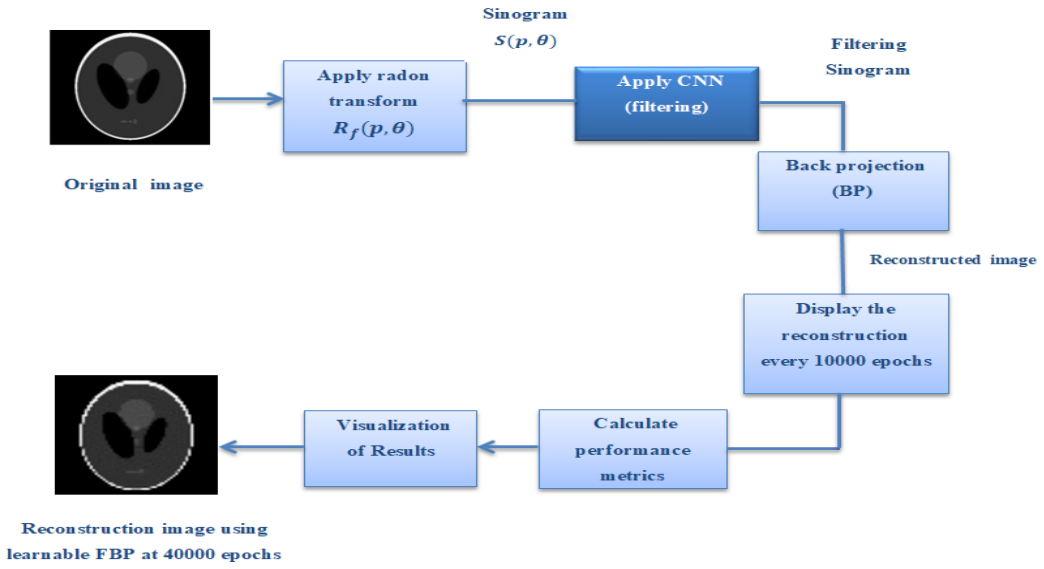


Figure 3-34 : block diagram for learnable FBP architecture.

This method enhances CT image reconstruction by combining Filtered Back projection (FBP) with a Convolutional Neural Network (CNN). A CT image is resized and projected using a simplified Radon-based system. A deep CNN with PReLU activations is trained to filter the sinogram before back projection. The network learns by minimizing the MSE between the reconstructed and original images over 10,000 iterations. This deep learning-based FBP reduces artifacts and improves image quality.

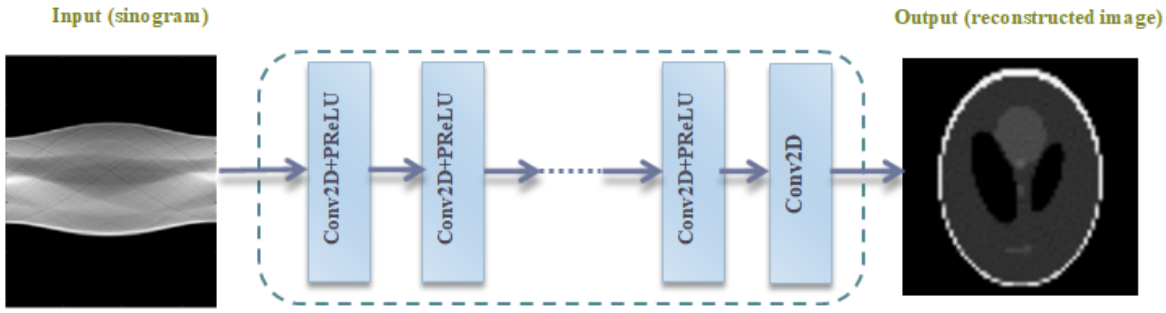


Figure 3-35 : architecture of CNN used in Learnable FBP

The CNN network as shown in figure 3.35, takes as input a sinogram, this architecture consists of an initial convolutional layer with 32 filters of size 3×3, followed by eight successive convolutional layers, each also using 32 filters of size 3×3. Each layer is followed by a PReLU (Parametric Rectified Linear Unit) activation, which introduces an adaptive non-linearity

allowing the network to learn more effectively. The final layer applies a 3×3 convolution with a single filter, thus reducing the output to a filtered sinogram. The goal of the network is to process and refine the input sinogram by reducing noise and artifacts before filtered back projection reconstruction. The expected output is therefore an enhanced sinogram which, once back projected, leads to a higher-quality reconstructed CT image.

The PReLU activation function plays a key role in this architecture: unlike the classical ReLU that clamps negative values to zero, PReLU learns a coefficient for the slope in the negative region, providing greater flexibility and promoting better convergence during network training.

Figure 3.36 show the result of reconstruction of phantom sheep-Logan image using learnable FBP

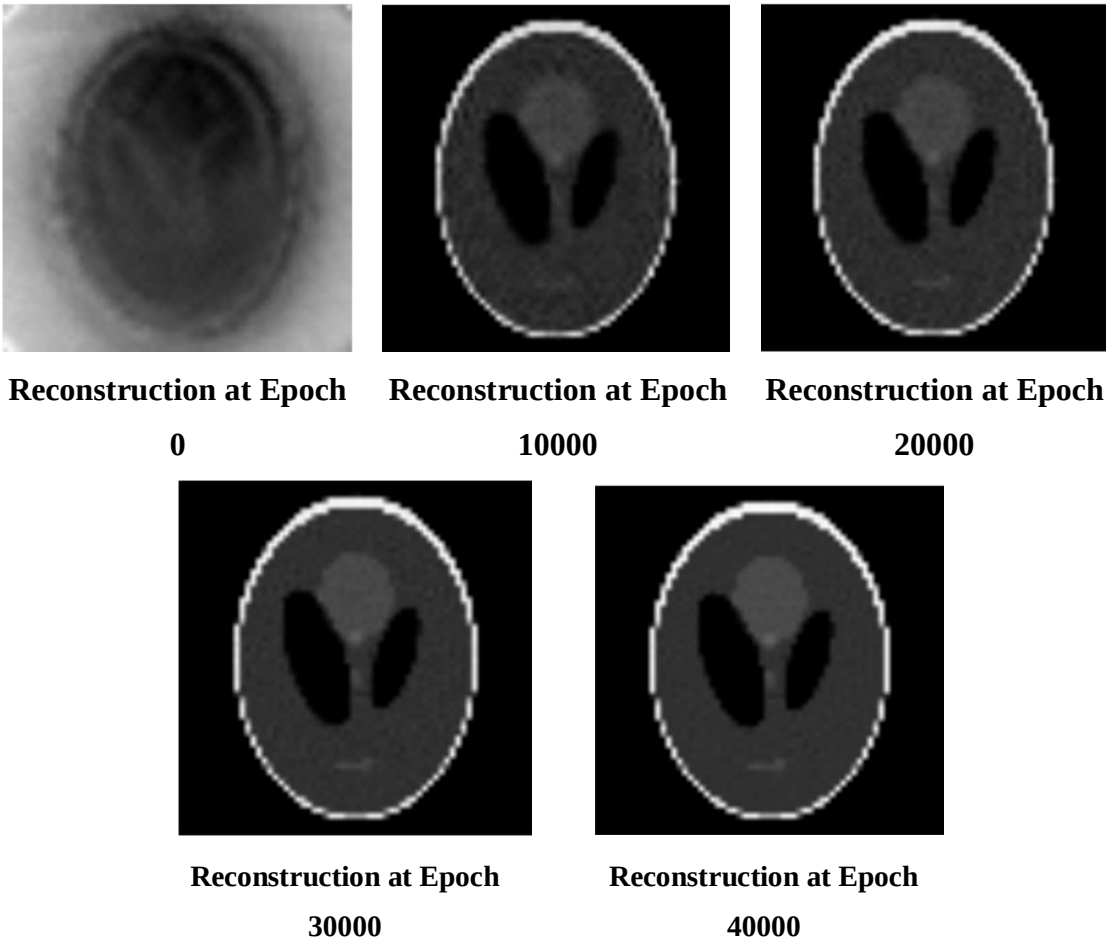


Figure 3-36 : reconstruction result of the Phantom sheep-Logan image during training using learnable FBP.

It can be observed that the reconstruction quality gradually improves with the number of training epochs. The initial image at epoch 0 is very blurry, but as the training progresses, the contours become sharper and more details appear. This highlights the effectiveness of the **Learnable FBP** with CNN model in learning to suppress noise and reconstruct the image accurately from noisy projections.

Figure 3.37 show the result of reconstruction of brain image using learnable FBP

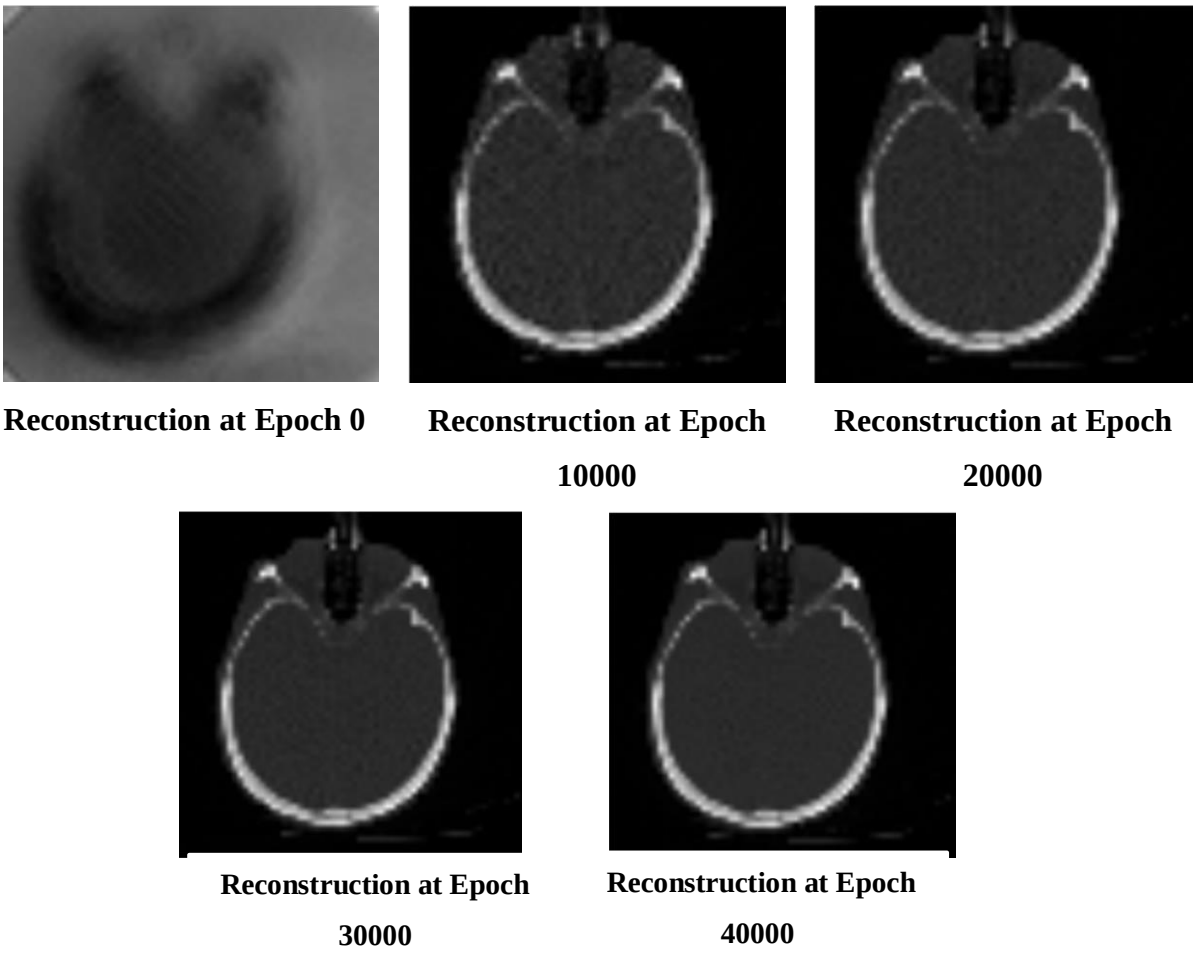


Figure 3-37 : result of reconstruction of the brain image during training using learnable FBP.

Table 3.17 presents the performance metrics of the Shepp-Logan phantom and brain image across different epochs (from 0 to 40,000) during the training of the learnable reconstruction model.

Table3. 17 : Performance of Shepp-Logan Phantom and Brain Image at Different Training Epochs using Learnable FBP.

Image	**Phantom sheep-Logan image**					**brain image**				
Metrics	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
Epoch 0	0.011318	19.46	0.610	0.320	0.410	0.060176	12.20	0.420	0.700	0.0750
Epoch 10000	0.000272	34.76	0.948	0.056	0.070	0.000930	30.43	0.910	0.085	0,098
Epoch 20000	0.000082	39.85	0.974	0.031	0.040	0.000818	30.99	0.918	0.077	0.089
Epoch 30000	0.000056	41.25	0.981	0.024	0.033	0.000150	38.23	0.970	0.034	0.045
Epoch 40000	0.000042	42.45	0.985	0.020	0.028	0.000099	40.20	0.980	0.025	0.033

The Learnable FBP method, which integrates filtered back projection with a trainable convolutional network, demonstrates strong potential to enhance reconstruction quality. However, it has a significant requirement: **it needs a relatively large number of training epochs** to reach optimal performance. Notably, clear improvements in metrics such as MSE, PSNR, and SSIM become evident only after tens of thousands of iterations. This highlights the strength of the model, but also its computational expense. Figure 3.38 illustrates the Mean Squared Error (MSE) loss evolution during the 40,000 training epochs of the Learnable FBP

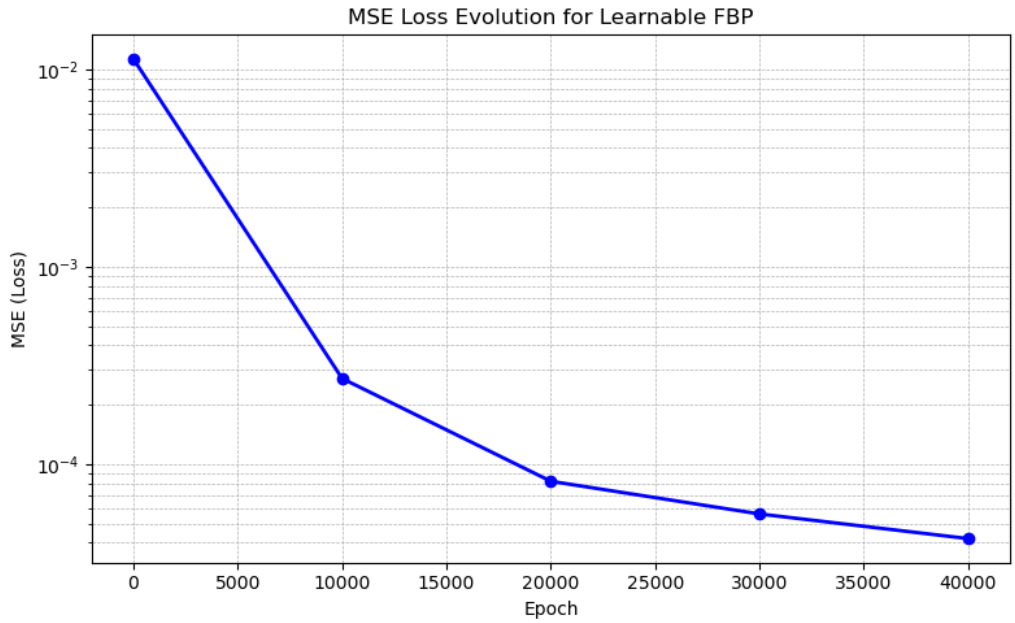


Figure 3-38 : Loss curve (MSE) evolution for the Learnable FBP model.

Synthesis

In this work, two reconstruction methods based on the FBP (Filtered Back projection) algorithm were implemented and compared: the first applies a denoising CNN (DnCNN) to the image reconstructed by FBP, while the second integrates a CNN directly into the reconstruction process by learning to filter the sinogram before back projection. The FBP + DnCNN technique serves as a post-processing phase to diminish noise and artifacts in the reconstructed image, while the Learnable FBP CNN alters the back projection filter in a learnable method.

About the quality of images, FBP combined with DnCNN produces superior visual and quantitative outcomes (PSNR, SSIM), along with expedited and more consistent training. The Learnable FBP CNN, on the other hand, is more inventive because it learns to optimize the reconstruction process itself, but its training is more involved and takes longer. Therefore, the trade-off between learning complexity and the intended final image quality determines which of these two approaches is best.

3.6 Algorithms of Deep learning MLEM

This section presents and compares the performance of various CT reconstruction algorithms augmented with deep learning approaches. Two hybrid techniques are compared:

- Maximum Likelihood Expectation Maximization followed by CNN denoising **(MLEM + DnCNN).**
- Learnable Maximum Likelihood Expectation Maximization integrated with CNN **(Learnable MLEM).**

Each approach is evaluated under varying levels of Gaussian noise to assess its robustness and image quality.

3.6.1 Architecture of MLEM+DnCNN denoiser

This method combines the iterative **MLEM algorithm**, known for its ability to reconstruct images faithfully under noisy or incomplete data, with a **convolutional neural network (DnCNN)** for denoising. The process involves applying MLEM to generate an initial reconstruction, which is then refined by a CNN trained to reduce residual noise and artifacts. This hybrid approach leverages both the statistical robustness of MLEM and the learning power of deep networks. Figure 3.39 show the main step of this algorithm

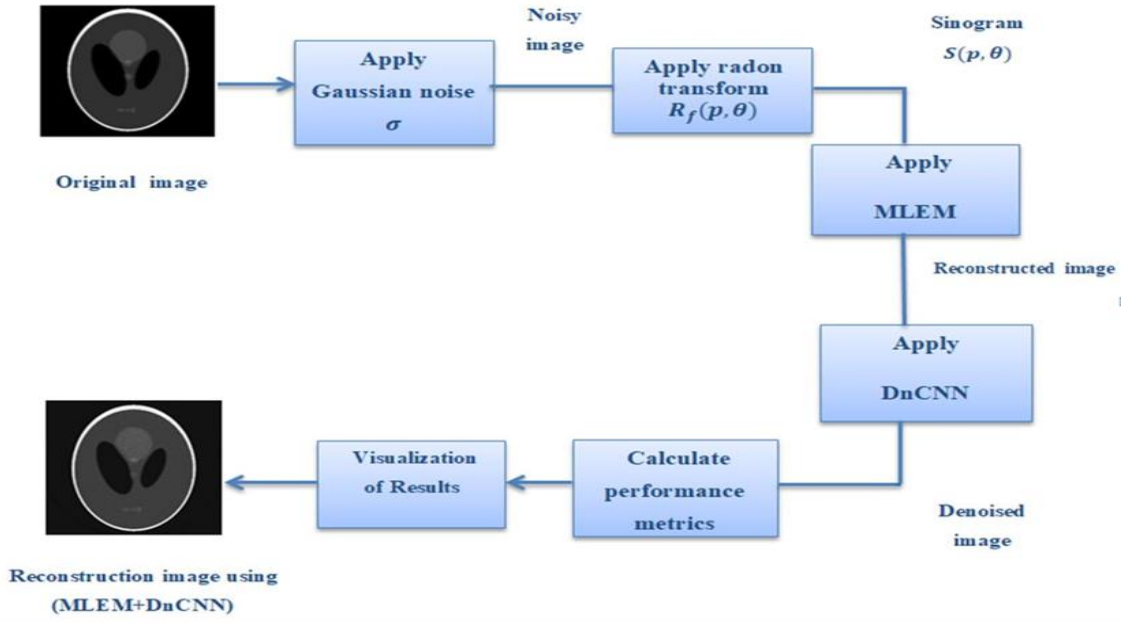


Figure 3-39 : block diagram for MLEM+DnCNN architecture.

The process begins by applying the iterative **MLEM** algorithm on the noisy sinogram to obtain an initial image reconstruction. This algorithm updates the reconstructed image by comparing simulated projections with measured data, aiming to maximize the likelihood of the observations. Next, this reconstructed image serves as input to a deep convolutional neural network **DnCNN,** which is responsible for reducing residual noise and artifacts. The network's output estimates the noise, which is subtracted from the input image to produce a denoised version. The network is trained by minimizing the Mean Squared Error (MSE) loss between the denoised image and the original reference image.

Figure 3.40 represents the CNN architecture used, which consists of 5 successive convolutional layers; each layer employs 3×3 filters. The first four layers are followed by a ReLU activation function, while the last layer produces the final output without any activation. This design, which excludes pooling layers, allows the network to maintain the fine spatial resolution necessary for accurate tomographic image reconstruction.

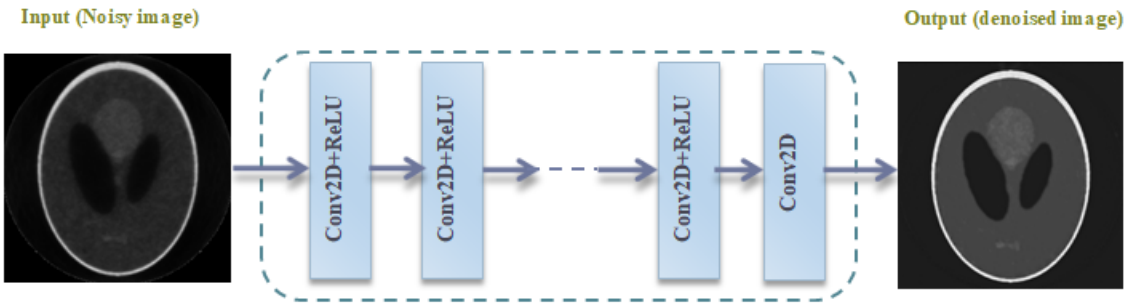
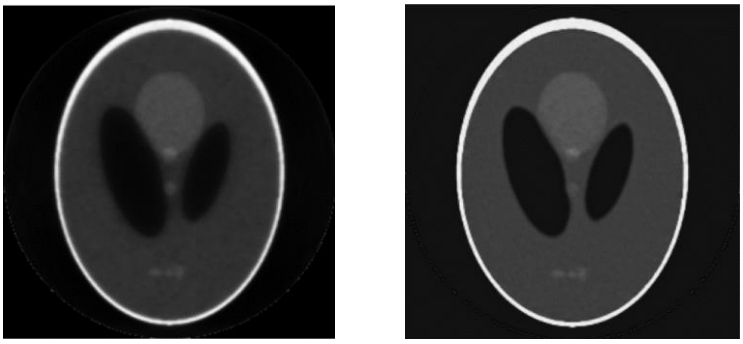


Figure 3-40 : Architecture of CNN used in MLEM+DnCNN.

The figures below illustrate the reconstruction results of the Phantom Sheep-Logan image using the conventional MLEM algorithm and the enhanced MLEM+DnCNN method

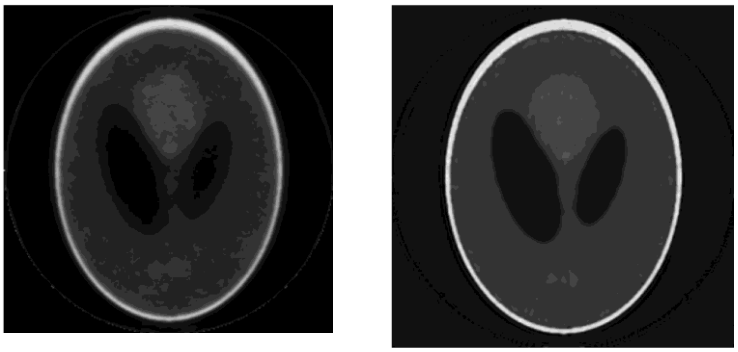
- **Case 1: Gaussian noise with standard deviation σ= 0.05**



MLEM Reconstruction **CNN denoiser**

Figure 3-41 .reconstruction results on the Phantom Sheep-Logan image with Gaussian noise (σ=0.05), showing the MLEM Reconstruction, and CNN denoising.

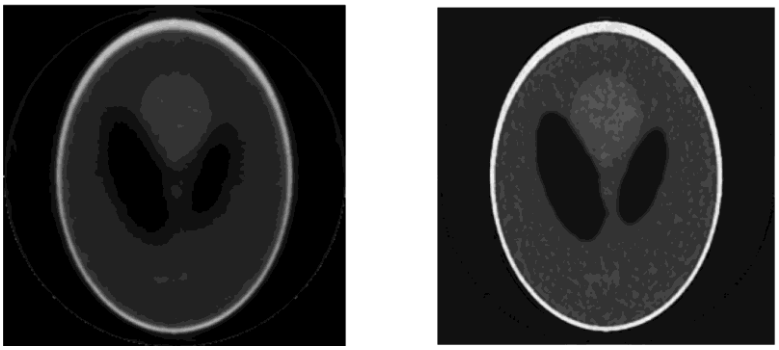
- **Case 2: Gaussian noise with standard deviation σ= 0.1**



MLEM Reconstruction **CNN denoiser**

Figure3-42 : reconstruction results on the Phantom Sheep-Logan image with Gaussian noise (σ=0.1), showing the MLEM Reconstruction, and CNN denoising.

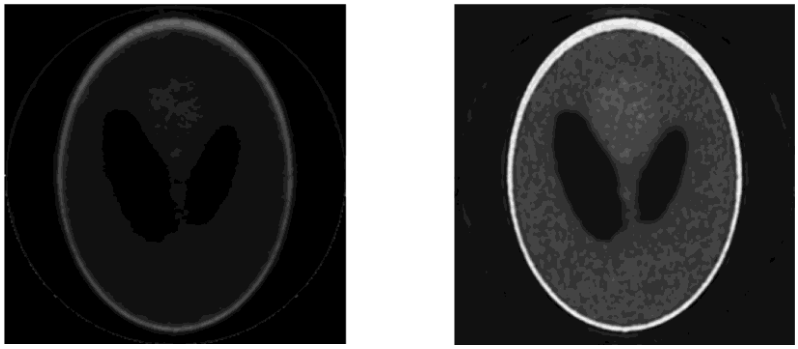
- **Case 3: Gaussian noise with standard deviation σ= 0.2**



MLEM Reconstruction **CNN denoiser**

Figure 3-43: reconstruction results on the Phantom Sheep-Logan image with Gaussian noise (σ=0.2), showing the MLEM Reconstruction, and CNN denoising.

- **Case 4: Gaussian noise with standard deviation σ= 0.3**



MLEM Reconstruction **CNN denoiser**

Figure 3-44 : reconstruction results on the phantom shepp Logan image with Gaussian noise (σ=0.3), showing the MLEM Reconstruction, and CNN denoising.

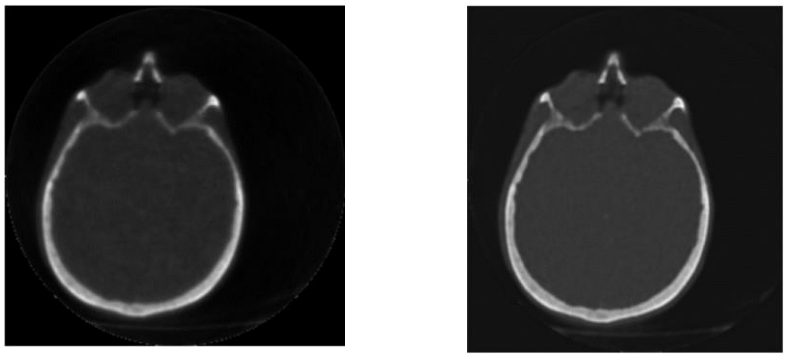
Table 3.18 represents Quantitative comparison between FBP and FBP+DnCNN reconstruction for Phantom Sheep-Logan image

Table3. 18 : Quantitative comparison between FBP and FBP+DnCNN reconstruction for Phantom Sheep-Logan image.

Methods	**MLEM reconstruction**					**CNN denoised reconstruction**				
Metrics	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
σ=0.05	0.002766	25.58	0.6435	0.0271	4.76	0.000397	34.01	0.9867	0.0075	0.82
σ=0.1	0.004447	23.52	0.5715	0.2472	0.2957	0.000562	32.50	0.9298	0.0755	0.0973
σ=0.2	0.010259	19.89	0.4981	0.3511	0.4297	0.001226	29.11	0.8731	0.1155	0.1470
σ=0.3	0.019291	17.15	0.4456	0.4470	0.5424	0.002148	26.68	0.7829	0.1534	0.1929

We can see that using MLEM alone does not give very good results, especially when the noise level is high. This is because we only used **20 iterations**, which is not enough to get a good-quality image. However, when we use **DnCNN after MLEM**, the image quality improves a lot. The DnCNN network removes noise well and makes the image clearer and more accurate. Metrics confirm this improvement. This shows that combining MLEM with a network like DnCNN is a good solution to improve reconstruction, even when the data is noisy. The figures below illustrate the reconstruction results of the brain image using the conventional MLEM algorithm and the enhanced MLEM+DnCNN method:

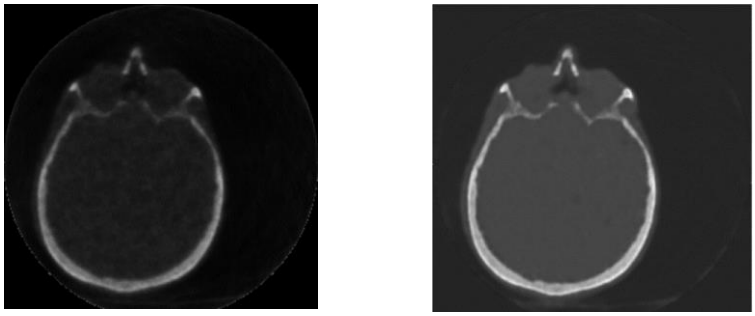
- **Case 1: Gaussian noise with standard deviation σ= 0.05**



MLEM Reconstruction **CNN denoiser**

Figure 3-45 : reconstruction results on the Brain image with Gaussian noise (σ=0.05), showing the MLEM Reconstruction, and CNN denoising

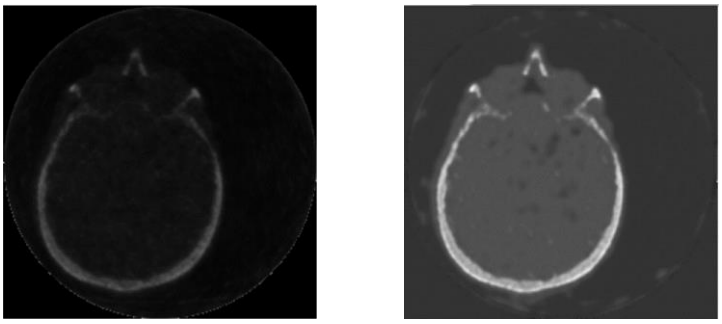
- **Case 2: Gaussian noise with standard deviation σ= 0.1**



MLEM Reconstruction **CNN denoiser**

Figure 3-46: reconstruction results on the Brain image with Gaussian noise (σ=0.1), showing the MLEM Reconstruction, and CNN denoising.

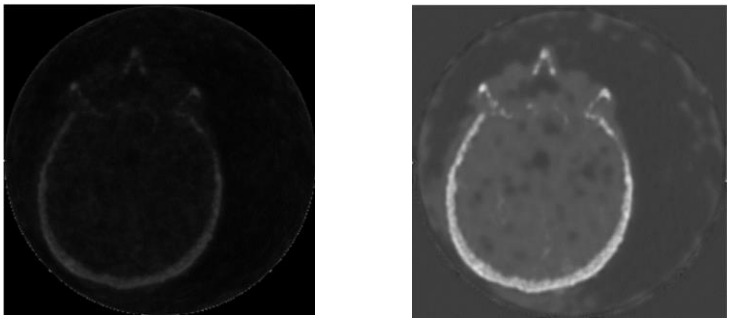
- **Case 3: Gaussian noise with standard deviation σ= 0.2**



MLEM Reconstruction **CNN denoiser**

Figure 3-47 : reconstruction results on the Brain image with Gaussian noise (σ=0.2), showing the MLEM Reconstruction, and CNN denoising

- **Case 4: Gaussian noise with standard deviation σ= 0.3**



MLEM Reconstruction **CNN denoiser**

Figure3. 3-48 : reconstruction results on the Brain image with Gaussian noise (σ=0.3), showing the MLEM Reconstruction, and CNN denoising.

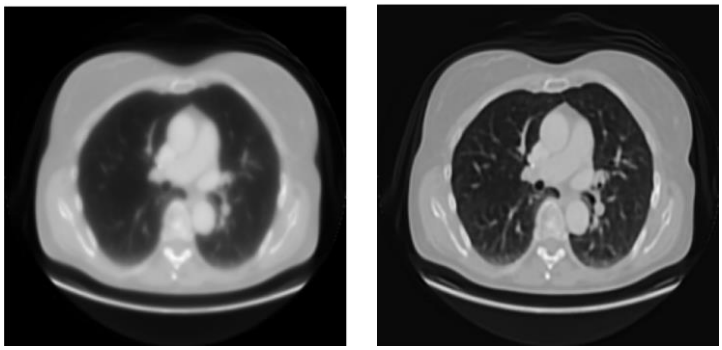
Table 3.19 represents Quantitative comparison between MLEM and MLEM+DnCNN reconstruction of brain image.

Table3. 19: Quantitative comparison between MLEM and MLEM+DnCNN reconstruction of brain image.

Methods	**MLEM reconstruction**					**CNN denoised reconstruction**				
Metrics	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
σ=0.05	0.000872	30.60	0.6468	0.1572	0.1948	0.000192	37.16	0.9285	0.0638	0.0897
σ=0.1	0.002309	26.37	0.5601	0.2402	0.3070	0.000291	35.36	0.9195	0.0789	0.1105
σ=0.2	0.007739	21.11	0.4787	0.3896	0.4937	0.000695	31.58	0.8485	0.1209	0.1718
σ=0.3	0.017113	17.67	0.4244	0.5131	0.6269	0.001706	27.68	0.7008	0.1872	0.2740

The figures below illustrate the reconstruction results of the image (a) using the conventional MLEM algorithm and the enhanced MLEM+DnCNN method

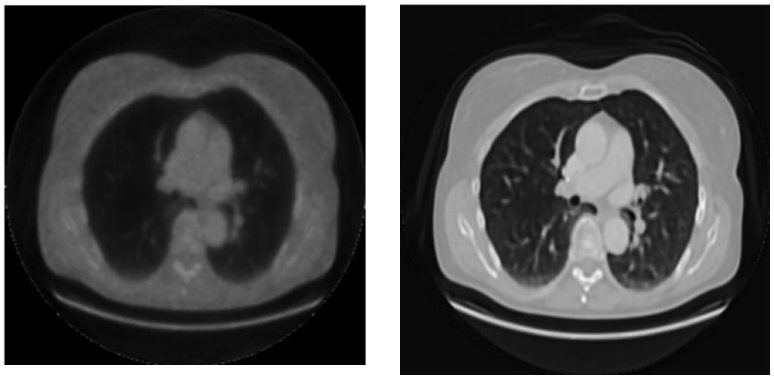
- **Case 1: Gaussian noise with standard deviation σ= 0.005**



MLEM Reconstruction **CNN denoiser**

Figure3-49 : reconstruction results on the Image (a) with Gaussian noise (σ=0.005), showing the MLEM Reconstruction, and CNN denoising.

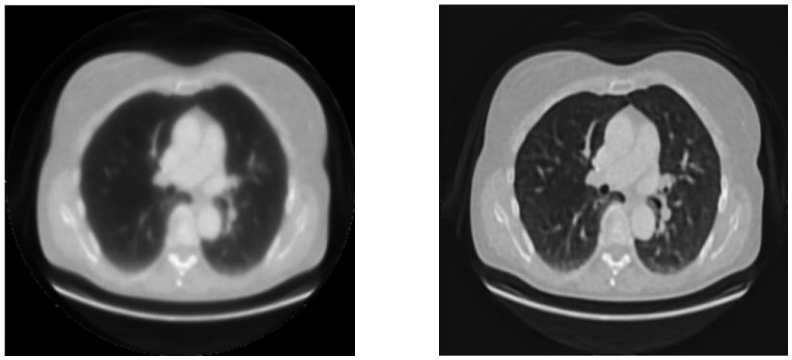
- **Case 2: Gaussian noise with standard deviation σ= 0.1**



MLEM Reconstruction **CNN denoiser**

Figure 3-50 : reconstruction results on the Image (a) with Gaussian noise (σ=0.1), showing the MLEM Reconstruction, and CNN denoising.

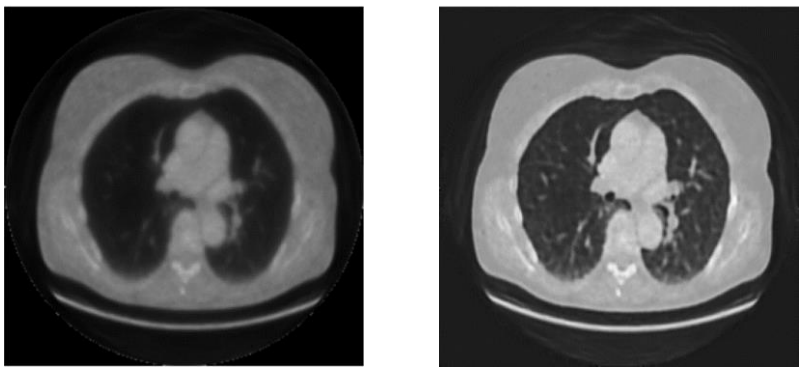
- **Case 3: Gaussian noise with standard deviation σ= 0.03**



MLEM Reconstruction **CNN denoiser**

Figure 3-51 : reconstruction results of the Image (a) with Gaussian noise (σ=0.03), showing the MLEM Reconstruction, and CNN denoising.

- **Case 4: Gaussian noise with standard deviation σ= 0.07**



MLEM Reconstruction **CNN denoiser**

Figure 3-52 reconstruction results of the Image (a) with Gaussian noise (σ=0.07), showing the MLEM Reconstruction, and CNN denoising.

Table3. 20 : Quantitative comparison between MLEM and MLEM+DnCNN reconstruction for Image (a).

Methods	**MLEM reconstruction**					**CNN denoised reconstruction**				
Metrics	MSE	PSNR (dB)	SSIM	D_f	D_p	MSE	PSNR (dB)	SSIM	D_f	D_p
σ=0.005	0.002194	26.59	0.7972	0.1375	0.1500	0.000290	35.37	0.9375	0.0402	0.0529
σ=0.03	0.003932	24.05	0.6535	0.1616	0.1994	0.000942	30.26	0.7246	0.0746	0.0974
σ=0.07	0.002389	26.22	0.7397	0.1402	0.1567	0.000382	34.18	0.9077	0.0448	0.0605
σ=0.1	0.003064	25.14	0.6882	0.1499	0.1773	0.000592	32.27	0.8597	0.0551	0.0758

Based on the result obtained, the MLEM + DnCNN method provides a powerful solution for image reconstruction in noisy environments. MLEM ensures statistically accurate reconstruction, but may leave visible noise when limited iterations are used. By combining it with DnCNN a neural network specialized in denoising the final image becomes cleaner, sharper, and closer to the ground truth. This hybrid approach merges the mathematical robustness of iterative algorithms with the adaptive capabilities of deep learning, making it an effective tool for enhancing medical image quality, even when dealing with incomplete or highly noisy data.

3.6.2 Architecture Learnable MLEM

Learnable MLEM is an innovative approach that enhances the classical iterative MLEM algorithm by integrating adaptive filtering layers based on convolutional neural networks. This method automatically learns optimal filtering parameters at each iteration, enabling better noise control and improved reconstruction quality. We show in figure 3.53 block diagram of the main steps of this method.

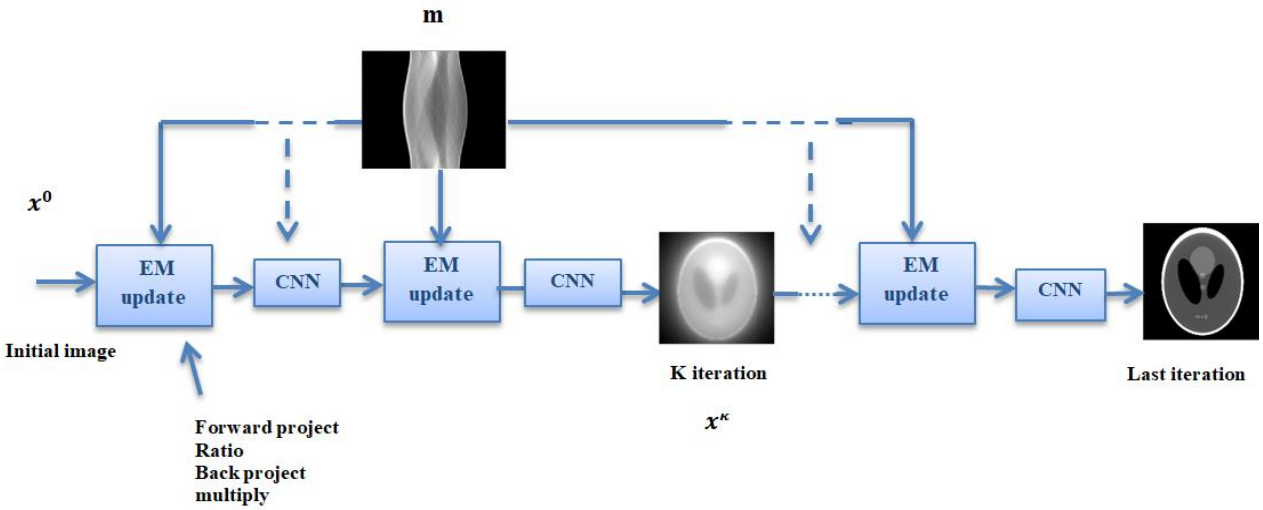


Figure 3-53 : the block diagram for learnable MLEM architecture.

In this algorithm, the first step is to load the original image, then preprocessed, notably resized to a standard size to standardize dimensions and facilitate subsequent processing. This resizing is crucial to ensure compatibility with the system matrix used to simulate projections and back projections. After this preprocessing, the sinogram is generated from the resized image by simulating CT projections at various angles. This raw sinogram, often noisy, is then filtered by a convolutional neural network (CNN) designed to reduce noise and correct artifacts present in the data. Finally, image reconstruction is performed through an iterative MLEM loop enhanced by this adaptive filtering, where the image is updated at each iteration to converge toward a more accurate and less noisy reconstruction. This complete pipeline ensures improved reconstruction quality by combining the strengths of deep learning and classical methods.

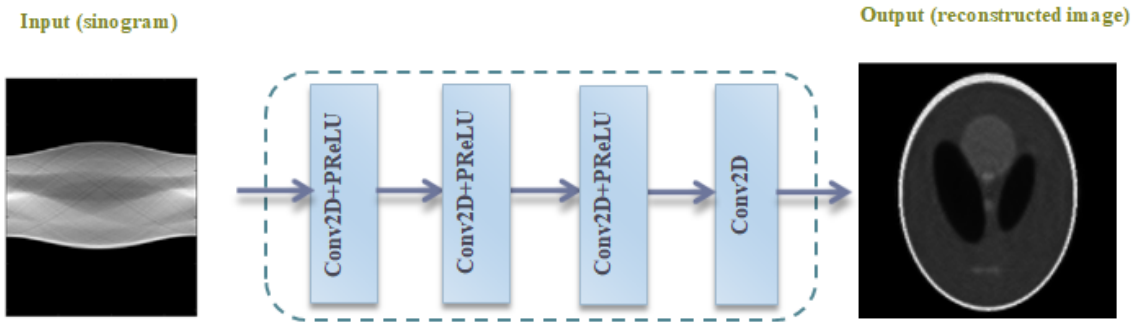


Figure 3-54 : CNN architecture for learnable MLEM.

Figure 3.55 shows reconstruction result of phantom sheep Logan image using learnable MLEM

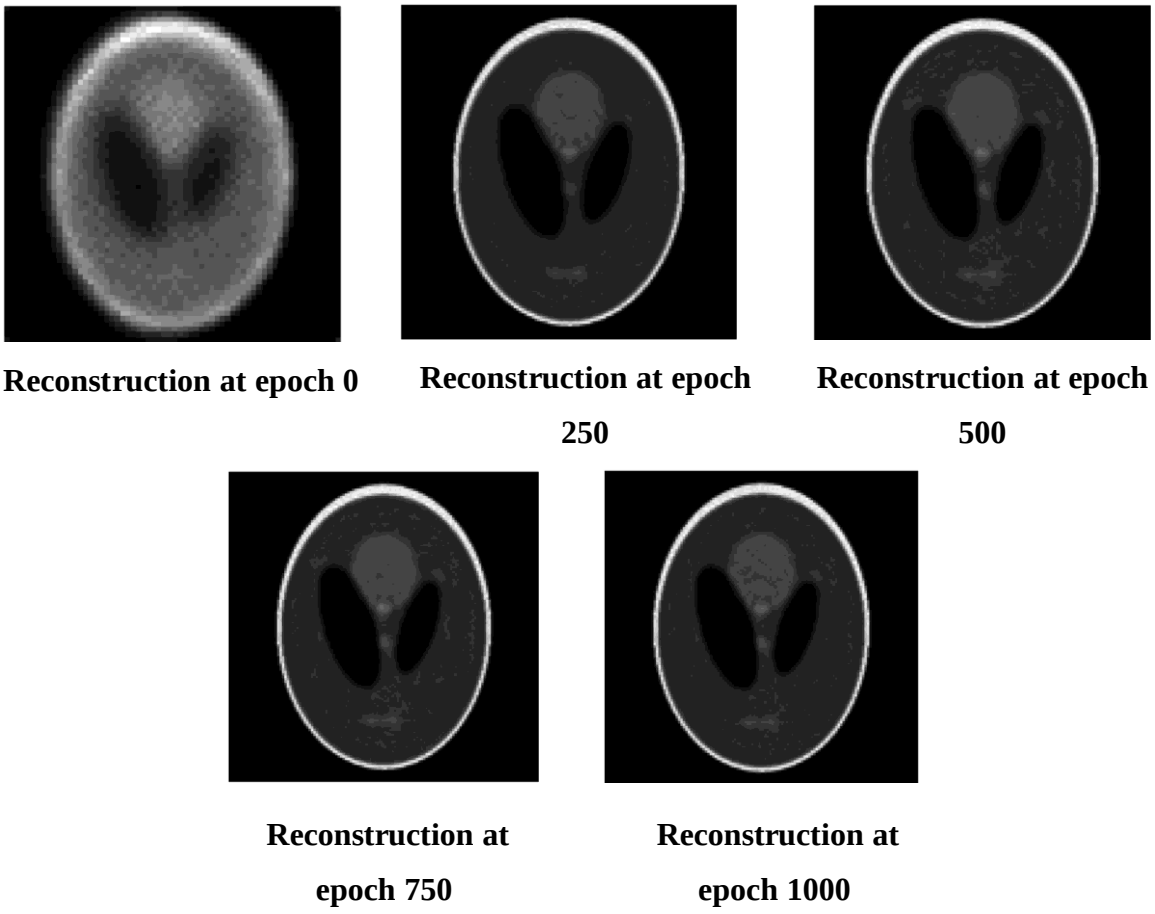


Figure 3-55 : reconstruction result of phantom sheep logan image using Learnable MLEM algorithm (0 to 1000) epochs.

Based on the results obtained, the image reconstructed using the Learnable MLEM model shows rapid and significant improvement. Starting from the first epochs, the reconstruction improves rapidly, with a clear reduction of noise and enhancement of structural details. Unlike methods such as Learnable FBP, which often require more than 10,000 epochs to converge, this model achieves high-quality reconstruction in just 1000 epochs or less. This highlights the efficiency and fast learning capability of the CNN applied to the sinogram, combined with the strength of the MLEM algorithm.

Figure 3.56 show reconstruction result of brain image using learnable MLEM

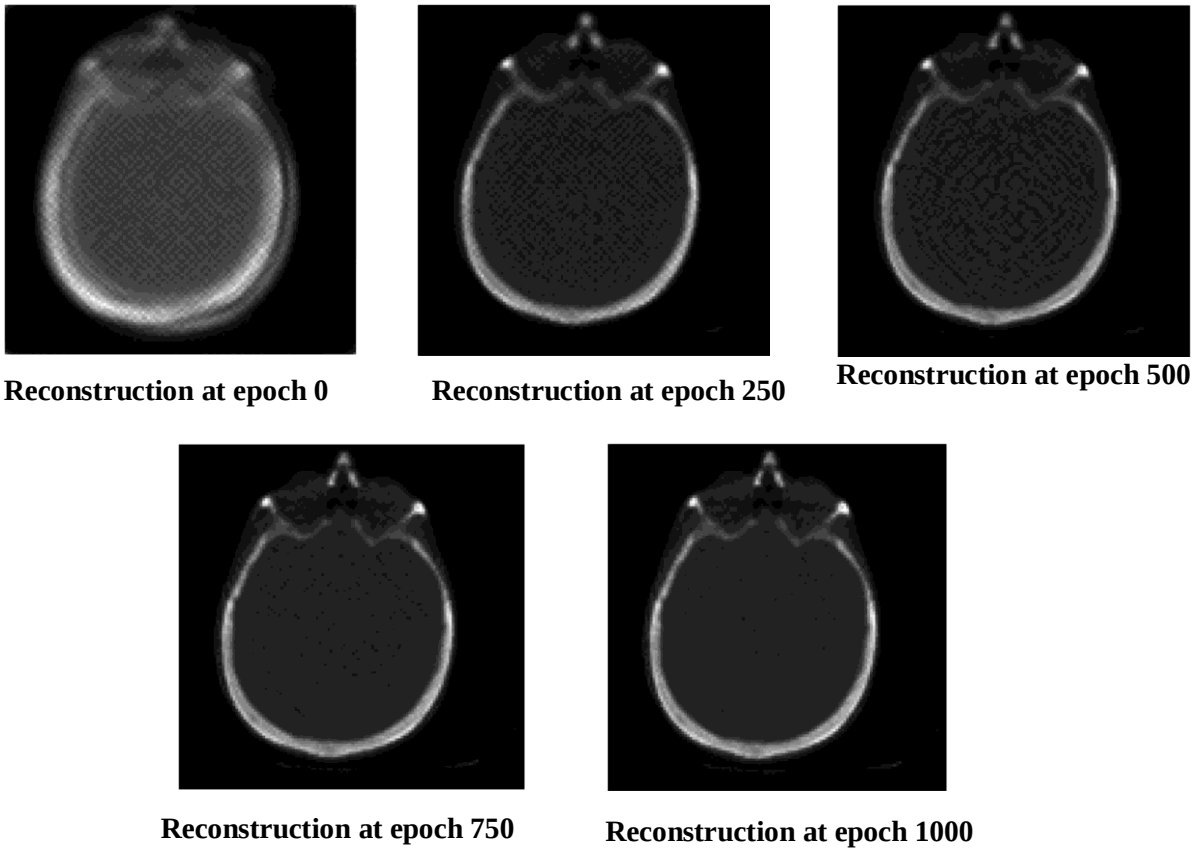


Figure 3-56 : reconstruction result of brain image using Learnable MLEM algorithm (0 to 1000) epochs.

Table3.21 represents Performance of Shepp-Logan Phantom and Brain Image at Different Training Epochs using Learnable MLEM

Table3. 21 : Performance of Shepp-Logan Phantom and Brain Image at Different Training Epochs using Learnable MLEM.

Methods	**Phantom shepp logan image**					**Brain image**				
Metric	MSE	PSNR	SSIM	D_f	D_p	MSE	PSNR	SSIM	D_f	D_p
epoch0	0.015272	18.16	0.3719	0.5787	0.4213	0.054191	12.66	0.3790	0.0508	0.9492
Epoch250	0.000194	37.12	0.9258	0.9883	0.0117	0.000663	31.78	0.9002	0.9902	0.0098
Epoch500	0.000126	38.99	0.9563	0.9909	0.0091	0.000331	32.75	0.9102	0.9908	0.0092
Epoch750	0.000102	39.89	0.9684	0.9913	0.0087	0.000168	33.29	0.9129	0.9918	0.0082
Epoch1000	0.000089	40.51	0.9718	0.9918	0.0082	0.000095	33.42	0.9136	0.9927	0.0073

The **Learnable MLEM (Maximum Likelihood Expectation Maximization)** model, enhanced by a **Convolutional Neural Network (CNN)**, represents a significant advancement in CT image reconstruction. Unlike Learnable FBP, which often require over 10,000 training epochs, Learnable MLEM achieves optimal reconstruction quality in fewer than 1,000 epochs. This efficiency stems from two main strengths: the statistical robustness of MLEM in handling noisy projections, and the CNN's ability to refine details and reduce artifacts. Together, they form a powerful synergy that enables fast, accurate, and noise-resilient reconstruction, making Learnable MLEM one of the most effective models available. Figure3.58 show the loss curve demonstrates a rapid convergence, dropping from 0.0152 to 0.000089 within just 1000 epochs. This indicates that the model successfully learns and refines the reconstruction through iterative updates.

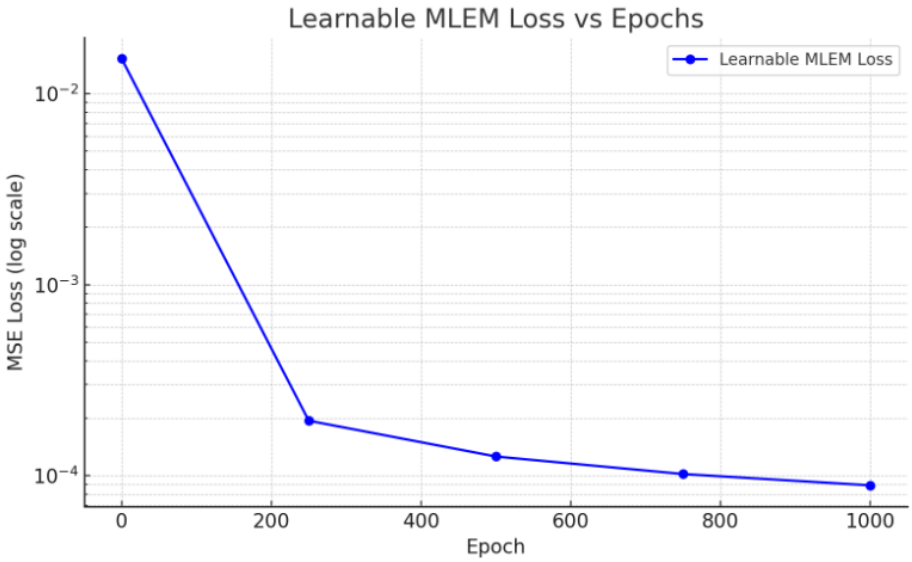


Figure 3-57 : Loss curve (MSE) evolution for the Learnable MLEM model.

Synthesis

Based on the results obtained, a clear distinction can be made between the performances of the four tested reconstruction algorithms. The **Filtered Back Projection (FBP) + DnCNN** approach combines a traditional analytical reconstruction with a denoising network, offering a slight enhancement but remaining limited in highly noisy conditions. The **Learnable FBP** improves upon this by integrating a trainable network, yet requires a large number of epochs to reach acceptable quality, reducing its training efficiency.

The **MLEM + DnCNN** method achieves better image quality due to the statistical strength of MLEM and the denoising power of CNN, but the CNN is applied after reconstruction, which limits joint optimization. In contrast, the **Learnable MLEM** algorithm integrates the CNN directly within the iterative reconstruction loop, allowing for simultaneous learning and correction. This fusion significantly improves both reconstruction quality and training efficiency, achieving strong results with fewer epochs. Therefore, **Learnable MLEM stands out as the most powerful and accurate method among all four models**.

Table3. 22 represent the difference between learnable FBP and Learnable MLEM

Table3. 22 : Difference between learnable FBP and Learnable MLEM.

Criterion	Learnable FBP	Learnable MLEM
CNN	8 layers	4 layers
Number of filter	32	64
Noise sensitivity	high	low
Reconstruction quality	Good	Excellent
Required epochs	More than 10000	Less than 1000
Time for epoch	Fast	Slower (due to MLEM iteration)

3.7 Conclusion

In this chapter, two major families of tomographic reconstruction methods were studied: analytical methods, represented by Back projection and Filtered Back projection (FBP), and iterative methods, represented by SART and MLEM.

A comparative performance analysis highlighted that FBP and MLEM were the most effective within their respective categories. Based on this observation, these two methods were selected

for enhancement using deep learning techniques, with the aim of combining the mathematical rigor of classical models with the adaptability of neural networks.

As a result, four hybrid approaches were developed: the first combines FBP with a denoising convolutional neural network (FBP + DnCNN), and the second applies the same concept to MLEM (MLEM + DnCNN). The other two approaches go a step further by integrating learning directly into the reconstruction process: this is the case for Learnable FBP, where the filters are optimized during training, and Learnable MLEM, which combines the iterative structure of MLEM with trainable CNN modules.

Experimental results have shown that these hybrid methods provide significant improvements over classical approaches, both quantitatively (lower MSE, higher PSNR and SSIM) and qualitatively (visually enhanced reconstructions). In particular, Learnable MLEM emerged as the most effective method, successfully leveraging the expressive power of deep learning and the robustness of maximum likelihood estimation.

These findings underscore the potential of hybrid approaches in the field of tomography, especially in challenging scenarios involving noisy, sparse, or incomplete data. Integrating deep learning into classical reconstruction frameworks paves the way for more accurate, robust, and application-oriented imaging solutions in clinical and industrial contexts.

General Conclusion

This thesis presented a comprehensive study, detailed implementation, and optimization of various image reconstruction techniques in parallel tomography, integrating both conventional methods (analytical and iterative) and recent approaches based on deep learning.

Noisy data are considered in this study. The noise is additive Gaussian with various levels, to check the robustness of the algorithms against noise. The implementations of the conventional algorithms are first performed; namely: for analytic tomography algorithms:

- Back projection (BP): this algorithm is tested on different data sets. the quality of reconstruction is discussed versus the number of projections (max=180 projections).
- FBP: in addition to the number of projections, three type of filters filter are used in our simulations.

The obtained results highlight the superiority of FBP allowing a significant enhancement of the reconstructed image quality. the calculated metrics are in total agreement with the visual quality.

For iterative tomography algorithms:

- Maximum Likelihood Expectation Maximization **(MLEM)** algorithm proved to be more effective than the Simultaneous Algebraic Reconstruction Technique **(SART)** and analytic BP and FBP**,** especially in scenarios involving noisy data.
- Simultaneous Algebraic Reconstruction Tomography (SART): allows a significant quality enhancement comparing to FBP and BP. The robustness against noise of SART is also proved.

Based on these observations, **FBP** and **MLEM** were selected as solid foundations for the development of hybrid methods integrating Convolutional Neural Networks (CNNs). Four hybrid architectures were implemented in this study:

1. **FBP + DnCNN**, combining fast analytic reconstruction FBP with learned CNN denoising;
2. **Learnable FBP (FBP_CNN_Net)**, where the filters are learned during training; so CNN is integrated in the FBP scheme.
3. **MLEM + DnCNN**, embedding the CNN after the iterative loop (post processing);
4. **Learnable MLEM (MLEM_CNN_Net)**, merging the statistical formulation of MLEM with deep learning optimization (CNN) in a fully trainable framework.

Experiments conducted on both ideal synthetic images (Shepp-Logan phantom) and real medical images (brain scans) demonstrated significant improvements in reconstruction quality across all hybrid methods, both quantitatively (MSE, PSNR, SSIM, D_f and D_p) and qualitatively (visual evaluation). Among the tested approaches, **Learnable MLEM** stood out as the best-performing method, successfully combining the accuracy of iterative algorithms with the adaptive power of neural networks.

In conclusion, this work confirms that the **future of tomographic reconstruction lies in the intelligent hybridization** of classical physical models with the adaptive capabilities of artificial intelligence, paving the way for more precise, robust, and clinically relevant reconstructions which in turn gives precise medical decisions..

References

References

1. Herman GT. Fundamentals of Computerized Tomography [Internet]. London: Springer London; 2009 [cited 2025 Jun 23]. (Singh S, editor. Advances in Pattern Recognition). Available from: http://link.springer.com/10.1007/978-1-84628-723-7

2. Kak AC, Slaney M. 1. Introduction. In: Principles of Computerized Tomographic Imaging [Internet]. Society for Industrial and Applied Mathematics; 2001 [cited 2025 Jun 23]. p. 1–4. (Classics in Applied Mathematics). Available from: https://epubs.siam.org/doi/10.1137/1.9780898719277.ch1

3. Shepp LA, Vardi Y. Maximum Likelihood Reconstruction for Emission Tomography. IEEE Transactions on Medical Imaging. 1982 Oct;1(2):113–22.

4. Hudson HM, Larkin RS. Accelerated image reconstruction using ordered subsets of projection data. IEEE Transactions on Medical Imaging. 1994 Dec;13(4):601–9.

5. Andersen AH, Kak AC. Simultaneous Algebraic Reconstruction Technique (SART): A superior implementation of the ART algorithm. Ultrasonic Imaging. 1984 Jan 1;6(1):81–94.

6. Jin KH, McCann MT, Froustey E, Unser M. Deep Convolutional Neural Network for Inverse Problems in Imaging. IEEE Trans on Image Process. 2017 Sep;26(9):4509–22.

7. Kak AC, Slaney M. Principles of Computerized Tomographic Imaging [Internet]. Society for Industrial and Applied Mathematics; 2001 [cited 2025 Jun 23]. 335 p. (Classics in Applied Mathematics). Available from: https://epubs.siam.org/doi/book/10.1137/1.9780898719277

8. Buzug T, editor. Introduction. In: Computed Tomography: From Photon Statistics to Modern Cone-Beam CT [Internet]. Berlin, Heidelberg: Springer; 2008 [cited 2025 Jun 23]. p. 1–13. Available from: https://doi.org/10.1007/978-3-540-39408-2_1

9. Deans, S. R. The Radon Transform and Some of Its Applications. 2nd ed. Dover Publications; 2007.

10. ResearchGate [Internet]. [cited 2025 Jun 26]. Figure. Parallel beam projection geometry used to generate Datasets A... Available from: https://www.researchgate.net/figure/Parallel-beam-projection-geometry-used-to-generate-Datasets-A-and-B_fig2_347881088

11. Shepp LA, Logan BF. The Fourier reconstruction of a head section. IEEE Trans Nucl Sci. 1974 Jun;21(3):21–43.

12. Zeng GL. Basic Principles of Tomography. In: Zeng GL, editor. Medical Image Reconstruction: A Conceptual Tutorial [Internet]. Berlin, Heidelberg: Springer; 2010 [cited 2025 Jun 23]. p. 1–19. Available from: https://doi.org/10.1007/978-3-642-05368-9_1

13. Gordon R, Bender R, Herman GT. Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography. Journal of Theoretical Biology. 1970 Dec 1;29(3):471–81.

14. Natterer F. The Mathematics of Computerized Tomography [Internet]. Society for Industrial and Applied Mathematics; 2001 [cited 2025 Jun 23]. 240 p. (Classics in Applied Mathematics). Available from: https://epubs.siam.org/doi/book/10.1137/1.9780898719284

15. Vardi Y, Shepp ,L. A., and Kaufman L. A Statistical Model for Positron Emission Tomography. Journal of the American Statistical Association. 1985 Mar 1;80(389):8–20.

16. Tong S, Alessio AM, Kinahan PE. Image reconstruction for PET/CT scanners: past achievements and future challenges. Imaging in Medicine. 2010 Oct;2(5):529–45.

17. Gonzalez RC, Woods RE. Digital image processing [Internet]. 3rd ed. Upper Saddle River, N.J: Prentice Hall; 2008. 954 p. Available from: https://www.r-5.org/files/books/computers/algo-list/image-processing/flat/Rafael_Gonzalez_Richard_Woods-Digital_Image_Processing-EN.pdf ``` :contentReference[oaicite:1]{index=1}

18. Huynh-Thu Q, Ghanbari M. Scope of validity of PSNR in image/video quality assessment. Electron Lett. 2008 Jun 19;44(13):800–1.

19. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing. 2004 Apr;13(4):600–12.

20. De Man, Benoit ; Basu, Samarjit. Distance-driven projection and backprojection in three dimensions. Physics in Medicine and Biology. 2002;49(11):2463–75.

21. Russell SJ, Norvig P. Artificial intelligence: a modern approach. Fourth Edition. Hoboken, NJ: Pearson; 2021. 1115 p. (Pearson Series in Artificial Intelligence).

22. Goodfellow I, Bengio Y, Courville A. Deep learning [Internet]. Cambridge, Massachusetts: The MIT Press; 2016. 775 p. (Adaptive computation and machine learning). Available from: https://www.deeplearningbook.org/

23. Mitchell TM. Machine Learning. New York: McGraw-Hill; 1997. 414 p. (McGraw-Hill series in computer science).

24. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 May;521(7553):436–44.

25. Bishop, C. M. Pattern Recognition and Machine Learning [Internet]. Springer New York; 2006 [cited 2025 Jun 24]. Available from: https://link.springer.com/deleted

26. Murphy KP. Machine learning: a probabilistic perspective [Internet]. Cambridge, MA: MIT Press; 2012. 1067 p. (Adaptive computation and machine learning series). Available from: https://www.cs.ubc.ca/~murphyk/MLbook/pml-toc-9apr12.pdf

27. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review. 1958;65(6):386–408.

28. Bansal, S. Gradient Descent vs. Backpropagation: What's the Difference? [Internet]. Analytics Vidhya. 2023 [cited 2025 Jun 25]. Available from:

https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/

29. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986 Oct;323(6088):533–6.

30. Sutton RS, Barto A. Reinforcement learning: an introduction [Internet]. Second edition. Cambridge, Massachusetts London, England: The MIT Press; 2018. 526 p. (Adaptive computation and machine learning). Available from: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

31. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization [Internet]. arXiv; 2014 [cited 2025 Jun 24]. Available from: https://arxiv.org/abs/1412.6980

32. Bottou L. Stochastic Gradient Descent Tricks. In: Montavon G, Orr GB, Müller KR, editors. Neural Networks: Tricks of the Trade [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012 [cited 2025 Jun 24]. p. 421–36. (Lecture Notes in Computer Science; vol. 7700). Available from: http://link.springer.com/10.1007/978-3-642-35289-8_25

33. Shrestha M. Data-centric Image Super-resolution in Magnetic Resonance Imaging: Challenges and Opportunities. Graduate Theses and Dissertations [Internet]. 2023 Dec 1; Available from: https://scholarworks.uark.edu/etd/5196

34. Bengio Y. Practical Recommendations for Gradient-Based Training of Deep Architectures. In: Montavon G, Orr GB, Müller KR, editors. Neural Networks: Tricks of the Trade [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012 [cited 2025 Jun 25]. p. 437–78. (Lecture Notes in Computer Science; vol. 7700). Available from: https://arxiv.org/pdf/1206.5533

35. Kim J, Lee W, Hong J, Kim S. Efficient Integrity-Tree Structure for Convolutional Neural Networks through Frequent Counter Overflow Prevention in Secure Memories. Sensors. 2022 Nov 13;22(22):8762.

36. Goodfellow I, Bengio Y, Courville A. Deep learning [Internet]. Cambridge, Massachusetts: The MIT Press; 2016. 775 p. (Adaptive computation and machine learning). Available from: https://www.deeplearningbook.org/

37. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998 Nov;86(11):2278–324.

38. Song, Shuhan. Exploring the effective application of color elements in graphic design based on a convolutional neural network model. ResearchGate [Internet]. 2023 [cited 2025 Jun 25];9. Available from: https://www.researchgate.net/publication/374552194_The_use_of_color_elements_in_graphic_design_based_on_convolutional_neural_network_model

39. Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. 2010;807–14.

40. Bega D, Gramaglia M, Fiore M, Banchs A, Costa-Perez X. DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications [Internet]. Paris, France: IEEE; 2019 [cited 2025 Jun 25]. p. 280–8. Available from: https://ieeexplore.ieee.org/document/8737488/

41. Byers, B.; Sheta, A. (PDF) Design of Convolutional Neural Networks for Fish Recognition and Tracking [Internet]. [cited 2025 Jun 25]. Available from: https://www.researchgate.net/publication/374113332_Design_of_Convolutional_Neural_Networks_for_Fish_Recognition_and_Tracking

42. Shivam. Convolutional Neural Networks : Understand the Basics [Internet]. Analytics Vidhya. 2021 [cited 2025 Jun 25]. Available from: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-understand-the-basics/