Deep learning for tomographic reconstruction : Study and application to computed tomography and positron emission imaging

Théo Leuliet

N°d'ordre NNT : 2022ISAL0093

THESE de DOCTORAT DE L'INSA LYON, membre de l'Université de Lyon

Ecole Doctorale N° 160
Électronique, Électrotechnique, Automatique (EEA)

Spécialité/discipline de doctorat:
Traitement du Signal et de l'Image

Soutenue publiquement le 04/11/2022, par :
Théo Leuliet

Deep Learning for tomographic reconstruction: study and application to computed tomography and positron emission imaging

---

Devant le jury composé de :

Reader, Andrew	Professor	King's College London	Rapporteur
Talbot, Hugues	Professeur des Universités	CentraleSupélec	Rapporteur
Lartizien, Carole	Directrice de recherche	CNRS	Examinatrice
Sabet, Hamid	Assistant Professor	Harvard Medical School	Examinateur
Maxim, Voichita	Maître de Conférences HdR	INSA Lyon	Directrice de thèse
Sixou, Bruno	Maître de Conférences HdR	INSA Lyon	Co-directeur de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**Mme Sandrine CHARLES** Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX **sandrine.charles@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Bruno MILLY** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **bruno.milly@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Remerciements

Je voudrais tout d'abord remercier mes encadrants Bruno et Voichita pour m'avoir fait confiance afin de mener à bien cette thèse, ainsi que pour le temps qu'ils m'ont consacré au cours de ces trois dernières années.

Bruno a sans cesse cherché à enrichir de ses connaissances les travaux effectués. Son implication dans les études réalisées ainsi que sa disponibilité - à toute heure - m'ont permis d'avancer au mieux dans mes recherches et je l'en remercie.

Voichita s'est toujours assurée du bon déroulement de ma thèse. Elle m'a grandement facilité la vie en faisant en sorte dès le début de me fournir un cadre idéal, que ce soit au niveau scientifique mais également en terme d'environnement de travail. Je la remercie également pour tous les précieux conseils qu'elle m'a apportés.

Je remercie le LabEx PRIMES (ANR-11-IDEX-0007) pour avoir financé cette thèse.

J'adresse mes sincères remerciements aux membres du jury pour avoir accepté d'en faire partie. Je remercie en particulier Andrew Reader et Hugues Talbot pour avoir accepté d'être rapporteurs et pour les retours détaillés qu'ils m'ont transmis.

Je remercie également Hamid Sabet qui a été un acteur essentiel de la deuxième partie de ma thèse. Il a pris le temps de m'expliquer de façon détaillée un domaine d'étude que je connaissais très peu. Nos échanges, malheureusement à distance, ainsi que ses nombreux retours clairs et bienveillants ont largement contribué aux résultats qui ont pu être obtenus.

Je voudrais aussi remercier tous les membres de CREATIS que j'ai pu côtoyer, pour leur accueil et leur participation à l'environnement agréable qui règne au laboratoire.

Je remercie en particulier Fabrice Bellet et Pierre Ferrier pour leur aide et leur disponibilité lors de mes demandes liées aux aspects informatiques.

Thomas Baudier et Ane Etxebeste m'ont eux aussi aidé de façon significative, notamment pour la mise en place des simulations GATE. Sans eux, une partie importante des résultats présentés dans cette thèse n'auraient pu être obtenus et je les en remercie grandement.

Je remercie mes co-bureaux de thèse Charles, Florian, Julia, Ludmilla, Nicolas, Raoul, Zexian et Yulei. Ce bureau 13 est une véritable réussite et c'est (presque) uniquement grâce à vous: parole d'expert !

Je tiens particulièrement à remercier Louise pour ces trois années au laboratoire. Nous avons beaucoup travaillé ensemble au début, et si nous avons finalement pris différentes directions dans nos recherches, elle a toujours su être un soutien considérable dans les bons comme dans les mauvais moments. Je lui souhaite beaucoup de réussite pour la suite car elle le mérite.

Je remercie aussi tous mes amis qui, parfois malgré la distance, m'ont énormément aidé tout au long de mon parcours. Une mention spéciale à Jean, qui m'a notamment

permis de m'aérer l'esprit lors de nos sorties en montagne, et Mathias dont le soutien sans faille depuis des années m'a fait constamment avancer.

J'ai une pensée particulière pour Achille, Vincent et Moktar pour leur soutien moral à toute épreuve.

Enfin, je ne remercierai jamais assez ma famille pour m'avoir permis d'être où j'en suis aujourd'hui. Sans eux, cette thèse n'aurait jamais pu être effectuée. Je remercie mes frères; ils sont pour moi des exemples et leur prèsence pendant ces trois années - et bien avant - a été essentielle pour en arriver à ce résultat. Je remercie aussi ma mère, dont les sacrifices, le soutien, la patience et bien d'autres éléments encore m'ont ôté nombre d'obstacles, me permettant d'avoir le parcours qui est le mien. Sa force est une source d'inspiration infinie et je lui dois tout.

À Papa

Contents

List of Figures

List of Tables

Introduction

This thesis has been conducted in the TOMORADIO team at CREATIS laboratory in Lyon. Part of the work has been performed in collaboration with the Radiation Physics Instrumentation Laboratory (RPIL), which is part of Department of Radiology, Massachusetts General Hospital, Harvard Medical School, in Boston.

Context and objectives

Discovery of X-rays in 1895 by Wilhelm Röntgen marked the beginning of medical imaging history. The hand of the German physician's wife was, at the end of that year, the first part of the human body to ever be visualized with radiography. Visualization of organs and blood vessels was made possible few years later thanks to the use of pharmaceutical contrast agents. In the 1950's, nuclear medicine came out as an other tool to inspect the interior of a body with imaging. In the 1970's the concept of Computed Tomography (CT) was developed, combining X-rays acquisition with computer-based technology to produce images showing slices of the body.

More than 120 years after Wilhelm Röntgen's discovery, the performance of medical imaging has drastically improved. This has helped physicians establish accurate and evidence-based decisions regarding a patient's health. Today, medical imaging assists radiologists for administering radiation therapy, overseeing surgical interventions or performing diagnosis. Innovations in this field have paved the way for earlier diagnoses, better treatments, better recovery outcomes, and overall decreased mortality.

Imaging techniques are for instance an essential part of cancer clinical protocols as they are able to furnish morphological, structural, metabolic and functional information [Fass, 2008]. Early detection of cancer, mostly allowed by imaging, is probably the major contributor to a reduction in mortality for certain cancers. Imaging is involved in staging of cancer, evaluating the response of treatment, and it can also be a tool to guide cancer treatment.

Cancer care is far from being the only application of medical imaging; neuroscience [Jacobs et al., 2003], fracture detection [Markhardt et al., 2009, Ito et al., 2005] are other examples among the numerous applications for which it is an asset of choice.

There are different imaging modalities, each of them being more or less adapted depending on the clinical case. Those modalities include CT, Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), Magnetic Resonance Imaging (MRI), Ultrasound imaging (US). In this thesis we mainly consider CT and PET.

Imaging techniques in general rely on several key components for which innovation has lead to more accuracy in screenings: contrast agents and radiopharmaceuticals, in-

strumentation (scanners, detectors), acquisition methods, tomographic reconstruction, and finally image processing.

In this thesis we focus on tomographic reconstruction. In the medical context, tomography allows observing the interior of a patient from the acquisition data. The design of efficient reconstruction algorithms is essential as it is one of the key components within the whole imaging process to assist radiologists. Numerous methods exist; some of them are general, others are specific to a particular imaging modality. In all cases, tomographic reconstruction amounts mathematically to solve an inverse problem. .

Over the last few years, a lot of research in the tomographic reconstruction field has been driven towards methods based on artificial intelligence (AI). AI was originally performant to solve problems that can be described by a list of formal rules, like playing chess. It had however issues solving tasks that human beings solve intuitively, such as recognizing words in speech or objects in images. A solution appeared with machine learning, which allowed computers to learn from experience: in this realm, there is no longer the need to formally specify all the knowledge required to solve a specific task. Machine learning allows computers understanding the world in terms of a hierarchy of concepts [Goodfellow et al., 2016]. If one draws a graph to show how these concepts are built with respect to each other in real life, the graph is "deep", i.e with many layers. Methods using such structural information are part of a particular field of methods: deep learning. They rely on specific structures that are called neural networks.

The objective of this thesis is therefore to study the interest of deep learning based methods for image reconstruction in PET and CT imaging, as compared to traditional methods. We put the emphasis on evaluating the performance of neural networks when considering criteria that are relevant for medical application. We also propose a deep learning based method for solving blind deconvolution problems. As the point spread function is hard to determine in emission imaging, our method is of interest in this context. We show preliminary results on a particular PET imaging system that aims at improving the detection of lymph nodes in the context of breast cancer imaging.

Contributions and organization of the manuscript

The first chapter allows understanding the basic elements of inverse problems. We also describe in this chapter the two modalities of interest in this thesis, namely CT and PET. Our aim is to understand how tomographic reconstruction can be modeled as an inverse problem, and we give analytical methods for its resolution.

In Chapter 2 we give an overview of iterative methods for performing image reconstruction from measurements. Especially, we perform a comparative study on simulated and real data to show limits of such methods for practical use, and show part of the extent to which deep learning based methods might be a relevant alternative.

Chapter 3 is a state of the art review of deep learning based methods for tomographic reconstruction for both CT and PET.

In Chapter 4 we study the impact of the way neural networks learn from data on the accuracy of quantitative and qualitative parameters in the reconstructed image; the parameters that we consider are of major significance from a medical point of view. More specifically, we study the influence of the loss function used for training neural networks for bone μ-CT imaging. The parameters for which accuracy is evaluated are particularly

relevant for diagnosis of bone-related diseases such as osteoporosis.

Chapters 5 and 6 are specific to PET, as we perform a study on a time-of-flight (TOF) PET intraoperative imaging system, in collaboration with the RPIL. The objective of this system is to allow better detection of lymph nodes in the context of breast cancer imaging. In Chapter 5 we detail state of the art methods for PET imaging, with a focus put on both image reconstruction and blind deconvolution techniques. In Chapter 6 we show our experiments in order to assess the performance of neural networks for the particular application and to validate the effectiveness of the network that we propose, called PAVENET.

Chapter 1

Tomographic reconstruction as an inverse problem

The task that is at the core of this thesis is tomographic reconstruction. As we mentioned in the introduction, performing this task amounts to solving an inverse problem. The objective of this first chapter is therefore twofold: in Section 1.1 we introduce the notion of inverse problems along with standard methods for their numerical resolution. This allows one to lay a theoretical foundation for all methods presented in the thesis. In Section 1.2, we present the two main imaging modalities that will be considered in the next chapters, namely CT and PET. We give practical and theoretical elements for the associated reconstruction task. In this second section, we especially formulate the inverse problem associated to CT and PET image reconstruction, and we explain the main methods for solving it analytically.

1.1 Inverse problems

In this section we formulate the general form of an inverse problem. We show the corresponding variational formulation, which represents the inverse problem as a functional to minimize. The choice of methods to minimize this functional depends on its characteristics: smooth, convex, etc. Depending on these characteristics we describe common optimization methods for the minimization task. Finally we will see a different approach, considering an inverse problem from a statistical point of view. General methods for resolution will be discussed in this case.

1.1.1 General formulation

Some physics problems rely on predicting the state of a system from a set of initial conditions. For instance, knowing the initial speed of a car, the deceleration generated by the driver's action on the brakes and information on the grip between the tires and the road, a relatively accurate prediction of the car's stopping distance can be obtained. Now if we consider the opposite scenario where we observe the stopping distance of the car and we want to determine the initial parameters, the task becomes more complex. The second situation belongs to the category of inverse problems, which purpose is to determine the causes of observed effects. Source reconstruction in acoustics or astronomy, seismic

tomography in geophysics and image super-resolution in computer vision are among the numerous examples of such problems. They indeed cover a wide range of applications including optics, oceanography, non destructive testing and of course, medical imaging.

We will consider the general formulation of an inverse problem as

$$p = A(f) \tag{1.1}$$

where p are the observed measurements, A is some operator that transforms the quantity of interest f that we aim to retrieve. In the sense of Hadamard, the problem (1.1) is said to be well-posed if it satisfies the three following conditions :

- a solution exists : $p \in Im(A)$
- the solution is unique : if A is a linear operator, we have $Ker(A) = 0$
- the solution depends continuously on the data.

In this thesis, the tomographic reconstruction task that we will consider involves a linear inverse problem. Also, in Chapters 5 and 6, we will introduce the notion of deconvolution which also corresponds to a linear inverse problem. Therefore in what follows, A is considered as linear.

1.1.2 Variational formulation

When at least one of the three mentioned conditions is not fulfilled, the problem is said to be ill-posed. It appears to be the case for most of the real inverse problems. Variational approaches are generally preferred to direct inversion methods when ill-posedness implies a too important deterioration of the estimate of the object f. Variational methods consist in finding f^* that minimizes an energy function J such that

$$J(f) = d(A(f), p) + \alpha R(f) \tag{1.2}$$

where d is a data-consistency term, R is a regularization term that allows giving *a priori* information on the solution and α is a weighting parameter. In what follows we go through a non-exhaustive list of optimization methods which can be used to reach an estimate of the solution f^* depending on the characteristics of J for the variational formulation (1.2).

1.1.3 Optimization methods

We consider here cases where J is either smooth or convex.

1.1.3.1 Smooth optimization

Here we consider the discretized problem as this will be the case in practice, with an energy function $J \in C^2(\mathbb{R}^p, \mathbb{R})$. We denote the gradient of J as ∇J and the Hessian as H_J. We have $\forall f \in \mathbb{R}^p, \nabla J(f) \in R^p$ and $H_J(f) \in \mathbb{R}^{p \times p}$. We also consider that there is a constant L such that[1] $\forall f \in \mathbb{R}^p, H_J(f) \preceq L.I$, where I is the identity matrix, which implies

[1] For matrices $U \in \mathbb{R}^{p \times p}$ and $V \in \mathbb{R}^{p \times p}$, $U \preceq V \iff \forall x \in \mathbb{R}^p, x^T U x \leq x^T V x$

that ∇J is L-Lipschitz on $\mathbb{R}^p$. The first and second-order optimality conditions for a local minimizer f^* are

$$\begin{cases} \nabla J(f^*) = 0 \\ H_J(f^*) \geq 0 \end{cases} \tag{1.3}$$

From these optimality conditions, several optimization schemes can be derived such as gradient descent or second order methods.

Gradient descent The iteration scheme for the gradient descent given an initialization $f^0 \in \mathbb{R}^p$ is

$$f^{n+1} = f^n - \eta \nabla J(f^n) \tag{1.4}$$

where η is the step size of the descent. Convergence of the algorithm can be shown when $\eta \leq \frac{2}{L}$ [Nocedal and Wright, 1999]. It is also possible to consider a descent direction d^n such that $\langle d^n , \nabla J(f^n) \rangle < 0$ so that $f^{n+1} = f^n + \eta d^n$. Also, methods for finding an optimal step size η (or η^n as it might change over iterations) are known as linesearch strategies and they are used in order to accelerate convergence.

Second order methods Second order methods involve the computation of H_J to find the next iterate. An example is the Newton's method applied to the gradient of J in order to find where it equals to zero. The resulting iteration scheme is given by

$$f^{n+1} = f^n - H_J(f^n)^{-1} \nabla J(f^n). \tag{1.5}$$

1.1.3.2 Convex optimization

In this paragraph we consider cases where J is a convex functional, that might be not differentiable. The vector $g \in \mathbb{R}^p$ is a subgradient of J in f if

$$J(v) \geq J(f) + \langle g , v - f \rangle , \ \forall v \in \mathbb{R}^p. \tag{1.6}$$

The notion of subgradient generalizes the notion of gradient. The set of vectors satisfying (1.6) is called the subdifferential of J in f and is denoted as $\partial J(f)$.

Subgradient descent It is possible to extend (1.4) to find a minimizer with the following subgradient descent

$$f^{n+1} = f^n - \eta^n g^n , \ g^n \in \partial J(f^n) \tag{1.7}$$

where η^n are the stepsizes. Convergence is ensured as long as $\lim_{N \to +\infty} \sum_{n=0}^{N} \eta^n = +\infty$ and $\lim_{N \to +\infty} \sum_{n=0}^{N} (\eta^n)^2 < +\infty$ [Nocedal and Wright, 1999].

Proximal methods The proximal operator of a function F, computed in z with parameter σ is defined as

$$prox_\sigma[F](z) = \arg\min_{z'} \left\{ F(z') + \frac{1}{2\sigma} \|z - z'\|_2^2 \right\}.$$

In practice, it can be useful to split an objective function into two terms, with one of the terms that is non-smooth. In such a case, the interest of the proximal operator is to make this term smooth. A reference method in the tomographic reconstruction research field is the Chambolle-Pock algorithm [Chambolle and Pock, 2011], which we detail in what follows.

We consider the objective function as composed of two convex functions $F : Y \mapsto \mathbb{R}$ and $G : X \mapsto \mathbb{R}$ with X and Y two Hilbert spaces, a linear operator $K : X \mapsto Y$ such that the general optimization problem can be written as

$$\min_f F(Kf) + G(f). \tag{1.8}$$

The conjugate F^* of F is such that $F^*(y) = \max_{y'} \{\langle y, y' \rangle_Y - F(y')\}$. Considering F as a proper, lower semi-continuous, and convex function (see e.g [Beck, 2017] for definitions), the Moreau-Fenchel theorem holds and we have $F = F^{**}$. Replacing F with F^{**} in (1.8) gives the equivalent problem

$$\min_f \max_y \langle y, Kf \rangle - F^*(y) + G(f) \tag{1.9}$$

which is a saddle point - or minmax - problem. The problem (1.8) corresponds to the minimization part of (1.9): it is the primal problem. The dual problem corresponding to (1.9) consists in solving the maxmin problem, which is

$$\max_y \min_f \langle y, Kf \rangle - F^*(y) + G(f). \tag{1.10}$$

We have

$$\begin{aligned}
&\max_y \left\{ \min_f (\langle y, Kf \rangle + G(f)) - F^*(y) \right\} \\
&= \max_y \left\{ - \max_f (-\langle y, Kf \rangle - G(f)) - F^*(y) \right\} \\
&= \max_y \left\{ - \max_f (\langle -K^T y, f \rangle - G(f)) - F^*(y) \right\}
\end{aligned}$$

so that (1.10) can be written as

$$\max_y \left\{ -G^*(-K^T y) - F^*(y) \right\}. \tag{1.11}$$

Problem (1.11) is referred to as the dual problem of (1.9).

The Chambolle-Pock (CP) algorithm - also known as the primal-dual hybrid gradient method (PDHG) - aims at simultaneously maximizing the dual (1.11) and minimizing the primal (1.8). In practice this amounts to performing these two steps:

1. A maximization step on y for (1.10), assuming an estimate $\hat{f}$ that minimizes $\langle y, Kf \rangle - F^*(y) + G(f)$ over f.

2. A minimization step on f for (1.9), assuming an estimate $\hat{y}$ that maximizes $\langle y, Kf \rangle - F^*(y) + G(f)$.

The difference between the value of the cost function in steps 1 and 2 is called the duality gap; convergence of the algorithm can be assessed following the decrease of this gap.

In the CP algorithm, a proximal gradient algorithm is used to solve both tasks. This corresponds to a method that can be applied to the general minimization problem $\min_y g(y) + h(y)$ where g is differentiable and h is a function for which the proximal operator is usually not computationally expensive. The proximal algorithm finds a solution as

$$\begin{aligned} y^{n+1} &= prox_\sigma(h)(y^n - \sigma\nabla g(y^n)) \\ &= \arg\min_u h(u) + \frac{1}{2\sigma}||u - y^n + \sigma\nabla g(y^n)||_2^2 \end{aligned}$$

which, by considering terms that do not depend on u, is equivalent to solving

$$\begin{aligned} &\arg\min_u h(u) + \frac{1}{2\sigma}||u - y^n||_2^2 + < u - y^n, \nabla g(y^n) > \\ &= \arg\min_u h(u) + g(y^n) + < u - y^n, \nabla g(y^n) > + \frac{1}{2\sigma}||u - y^n||_2^2 \end{aligned}$$

so that one can observe that y^{n+1} is actually a minimizer of $h(u)$ plus a quadratic local model of $g(u)$ around y^n.

Let us now apply the proximal gradient method to (1.10) considering an estimate $\hat{f}$. The problem can be written as

$$\min_y g(y) + h(y) \tag{1.12}$$

with $g(y) = -\langle y, K\hat{f}\rangle$, so that $\nabla g(y) = -K\hat{f}$, and $h(y) = F^*(y)$.

Applying the proximal gradient method therefore gives

$$y^{n+1} = prox_\sigma(F^*)(y^n + \sigma K\hat{f}) \tag{1.13}$$

Assuming an estimate $\hat{y}$, minimization of (1.9) can also be written as $\min_f g(f) + h(f)$ where this time $g(f) = \langle \hat{y}, Kf\rangle$, so that $\nabla g(f) = K^T\hat{y}$, and $h(f) = G(f)$ so that

$$f^{n+1} = prox_\tau(G)(f^n - \tau K^T\hat{y}) \tag{1.14}$$

where, similarly to σ, τ is a step size.

The pseudo-code of the general Chambolle-Pock algorithm is given in Algorithm 1. Note that the last update aims at getting faster convergence.

Algorithm 1: Pseudo-code for N-steps of the original CP algorithm

```
initialize τ, σ, θ
initialize f^0 and y^0 to zero
n ← 0
f̂^0 ← f^0
while n ≤ N do
    y^{n+1} ← prox_σ[F*](y^n + σK f̂^n)
    f^{n+1} ← prox_τ[G](f^n − τK^T y^{n+1})
    f̂^{n+1} ← f^{n+1} + θ(f^{n+1} − f^n)
    n ← n + 1
```

To sum up, there are five steps required in order to run the Chambolle-Pock algorithm for an optimization problem :

1. Put the problem into the form $F(Kf) + G(f)$.
2. Compute the convex conjugate F^*.
3. Compute the proximal mappings of F^* and G.
4. Substitute the results into the original Chambolle-Pock algorithm.
5. Run the algorithm monitoring the primal-dual gap for convergence.

More generally there are plenty of algorithms for optimization problems [Nocedal and Wright, 1999, Beck, 2017], and here we described only a tiny part of them. It is also to be noted that we considered an energy function J that is either smooth or convex. When it is not the case, solving the optimization problem involve different methods; we can for instance mention stochastic gradient methods such as Adam optimization [Kingma and Ba, 2014] that is commonly used for deep learning based methods as we will see in Chapters 3 and 4.

We also point out the fact that the *scipy* library in Python offers a large number of implemented optimization algorithms and thus can be an interesting tool for such problems.

1.1.4 Statistical formulation

Inverse problems can also be considered with a statistical point of view [Dashti and Stuart, 2017]. This represents a different paradigm compared to what was detailed before. In that case the object of interest f and the measurements p are assumed to be random variables. We denote the likelihood function as $\pi_{data}(p|f)$; it corresponds to the probability of observing p knowing f. The prior distribution is the probability of having f, under no other assumption; we denote it as $\pi_{prior}(f)$. These two distributions can be combined to get the posterior distribution with Bayes rule as $\pi_{post}(f|p) \propto \pi_{data}(p|f)\pi_{prior}(f)$.

Let us illustrate these notions with a simple example. Say one wants to estimate the age of an ibex located in a certain mountain area by measuring the length of the animal's honks. In this case f represents the unknown age and p is the length of the honks. Knowing $\pi_{data}(p|f)$ corresponding to the honks' length distribution depending on the age of an ibex, and knowing the distribution $\pi_{prior}(f)$ of all these animals' age in the searched area allows obtaining the posterior distribution $\pi_{post}(f|p)$. An example is given in Figure 1.1; it illustrates the importance of the prior distribution since it allows getting more information on the solution one is looking for. Here in our simple example we observe that the knowledge of the fact that ibexes are generally old in the investigation area allows one to obtain a higher estimation of the age of the observed animal given the length of its honks. The prior knowledge thus shifts the distribution towards higher values of the ibex age f. This is a major difference between using the likelihood and using the posterior distribution to estimate a parameter from measurements.

A solution to our inverse problem can be obtained by solving

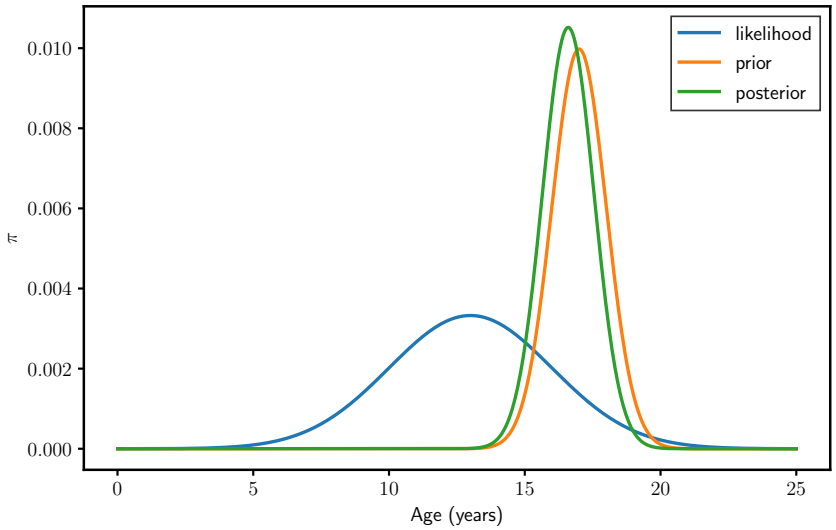


Figure 1.1: Illustration of the difference between likelihood and posterior distribution. We consider a measure of $p_{obs} = 65$ cm for the honks length. We represent here the likelihood $\pi_{data}(p|f)$ for $p = p_{obs}$ so it corresponds to the probability that such a measure is obtained for every age f. The prior $\pi_{prior}(f)$ is the age distribution over the searched area, with a mean of $\bar{f} = 17$y.o. The posterior distribution $\pi_{post}(f|p_{obs})$ is the probability of the ibex age given the measurement p_{obs}.

$$\max_f \pi_{post}(f|p). \tag{1.15}$$

The resulting estimate $\hat{f}$ is called the *maximum a posteriori* or MAP. It is common to solve the minimization problem that consists of taking the negative logarithm of (1.15). The variational methods discussed in the previous part can then be used to solve the equivalent problem

$$\min_f(-\log \pi_{data}(p|f) - \log \pi_{prior}(f)) \tag{1.16}$$

We derive in what follows some simple examples to illustrate how to obtain an estimate of the MAP depending on the noise considered on the data and the prior assumed on the solution.

Gaussian noise, Gaussian prior Considering Gaussian noise we can write

$$p = Af + \epsilon \tag{1.17}$$

with $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is an additive Gaussian noise with zero mean and variance σ. Assuming a Gaussian prior such that $f = (f_1, ..., f_n)$ with f_i i.i.d that are normally distributed with zero mean and unit variance, we have

$$\pi_{post}(f|p) = \frac{1}{2\pi\sigma} e^{-\frac{1}{2\sigma^2}||Af-p||_2^2 - \frac{1}{2}||f||_2^2} \tag{1.18}$$

Taking the negative logarithm we obtain the equivalent problem

$$\min_f ||Af - p||_2^2 + \sigma^2 ||f||_2^2 \tag{1.19}$$

which corresponds to a Tikhonov-regularized least squares problem. Denoting the adjoint of A by A^*, the solution f_{opt} satisfies the equation $(A^*A+\sigma^2 I)f_{opt} = A^*p$; it is then possible to solve this linear system numerically.

Poisson noise, uniform prior Let us now consider Poisson noise on the data and a uniform prior. We have a vector of measurements p that are represented as random variables following a Poisson distribution with mean $\lambda = Af$ (element-wise). Here the likelihood can be written

$$\pi_{data}(f|p) = \prod_i \frac{(Af)_i^{p_i}}{p_i!} e^{-(Af)_i}, \tag{1.20}$$

so maximizing the (log-)likelihood amounts to minimizing the Kullback-Leibler divergence

$$KL(Af, p) = \sum_i (Af)_i - p_i + p_i \log(p_i) - p_i \log((Af)_i) \tag{1.21}$$

since the term $-p_i + p_i \log(p_i)$ does not depend on f.

A common way to solve such a problem when A is the operator for tomographic reconstruction is to use MLEM algorithm [Shepp and Vardi, 1982] that will be described in Chapter 2. When A is a convolution operator, the equivalent method is known as the Richardson-Lucy algorithm [Richardson, 1972, Lucy, 1974] which we will refer to in Chapter 5.

We see that considering either a variational or a statistical formulation for a given inverse problem, methods for resolution involve minimizing some functional. Many algorithms for performing the minimization task exist and are more or less efficient depending on the considered noise and "quality" of the measurements p. In Chapter 2 we will describe some algorithms that are specific to PET and CT modalities. In what follows we describe both applications so that we can formulate the inverse problem that is common to the corresponding tomographic reconstruction task.

1.2 Tomographic reconstruction for CT and PET

We split this section as follows: we first describe computed tomography (CT), then we focus on positron emission tomography (PET). The last sub-section unifies the reconstruction task for both modalities by formulating a common inverse problem, for which analytical methods for resolution are given.

1.2.1 Computed tomography

CT scanners use a rotating X-ray tube and a row of detectors placed in a gantry to measure X-ray attenuations by different tissues inside the body (Figure 1.2). The multiple

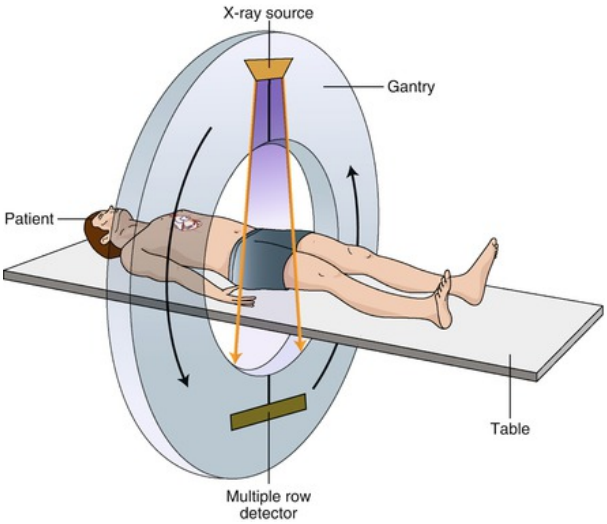


Figure 1.2: Scheme of a CT scan.[1]

X-ray measurements taken from different angles are then processed on a computer using tomographic reconstruction algorithms to produce tomographic (cross-sectional) images (virtual "slices") of a body. CT scan can be used in patients with metallic implants or pacemakers, for whom magnetic resonance imaging (MRI) is contraindicated.

Tomography is an imaging technique that reconstructs a volume from external measurements. It is thus a non-invasive technique to observe the inside of an object without any manual intervention on it. This is for instance the principle that is used in airports to visualize the content of a passenger's bag at the gates. There are plenty of applications for tomography : geophysics, astrophysics, optics or even archeology. The most common use for such a technique is biology and especially medical imaging. Tomography usually involves a detector that measures radiation data - photon counts for instance - after these radiations have interacted with the object of interest and/or with the environment. In emission tomography, the source of the radiations is the object of interest; in the contrary for transmission tomography the source is placed outside of the object, so that the measurements of the detector can give information on the attenuated/transmitted radiation.

We focus here on Computed Tomography which is an imaging technique that consists in 1: making use of X-rays measurements and 2: processing them with tomographic reconstruction algorithms to produce cross-sectional images of a body. CT has various applications but the most common and the one that we are interested in here is the use for medical scanners. CT scans are used for preventive medicine or screening for disease. They are a tool for almost every part of the body; for the head it is used to detect stroke, tumors, haemorrhage; for the neck it helps evaluate the presence of thyroid cancer for instance. CT scans for the lungs have numerous applications such as detecting changes in the lung parenchyma - the tissue of the lungs - or lung cancer. There are many more applications of CT scans including angiography, visualization of the heart, the abdomen, etc.

Overall, X-rays and Computed Tomography have been a breakthrough in the medical

[1] `https://thoracickey.com/basic-principles-in-computed-tomography-ct`

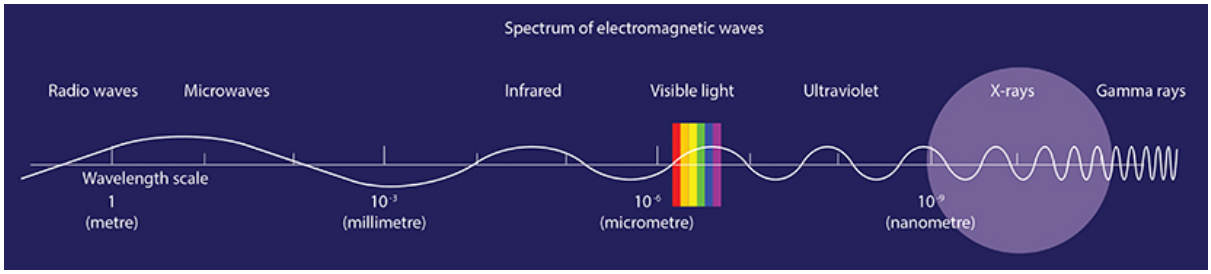


Figure 1.3: Electromagnetic spectrum. Scheme from the European Synchrotron Radiation Facility website.[1]

field by alleviating the systematic need for clinical intervention to visualize the tissues and organs of a patient. It is also to be noted that the radiation undergone by the patients might cause serious damage to the their health when the dose is high [United Nations Scientific Committee on the Effects of Atomic Radiation et al., 1996, Rehani and Berry, 2000, Staniszewska, 2002], which has lead to an increasing amount of research towards reducing the radiation dose while keeping the image quality exploitable for clinicians. We will cover this topic more in depth in Chapter 3.

We present here the physics of CT and then we focus on the mathematical framework associated to tomographic reconstruction in CT.

1.2.1.1 Physics of CT

From X-rays to CT images X-rays[2] are electromagnetic waves with a wavelength ranging from picometers to nanometers. They are located at the high energy/short wavelength end of the electromagnetic spectrum, between ultraviolet light and gamma rays as represented in Figure 1.3. For medical purpose, hard X-rays - as opposed to soft X-rays - are commonly used thanks to their ability to penetrate the matter. Their wavelength ranges from 0.01 nm and 0.1 nm. Radiography is based on the fact that X-rays are more or less absorbed depending on the density of the material they penetrate; more density means more attenuation/less transmission by the matter, so a detector placed after such an object will measure less radiation.

Let us consider an object placed between a X-ray source and a detector. The measured intensity of the X-ray beam I is related to the intensity of the incident beam I_0 according to the Beer-Lambert law as

$$I = I_0 e^{-\mu\rho l} \tag{1.22}$$

with l the thickness of the material in cm, μ the mass coefficient (cm^2/g) and ρ the volumic mass of the material (g/cm^3), $\mu_X = \mu\rho$ thus corresponding to its linear absorption - or attenuation - coefficient (cm^{-1}).

The volume of interest can therefore be characterized by its linear attenuation coefficient, which varies across the space. It is common for radiologists to represent CT images

[1] `https://www.esrf.fr/about/synchrotron-science/synchrotron-light`

[2] X-rays were first described by Wilhelm Röntgen in 1895. He called them X-radiation referring to the letter X often used for the unknown in mathematical formulas as their true nature was unclear to him at that time.

Tissue	Hounsfield unit (HU)
Air	−1000
Lung	−500
Fat	−100 to −50
Water	0
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
Soft tissues	+100 to +300
Cancellous bones	+700
Cortical bones	up to +3000

Table 1.1: Examples of Hounsfield units for different tissues.

in a discretized way with Hounsfield Units (HU)[1]. The HU value can be obtained with a linear transformation of the linear attenuation coefficient of the X-ray beam as

$$HU(X) = 1000 \times \frac{\mu_X - \mu_{water}}{\mu_{water}} \tag{1.23}$$

where μ_X is the linear attenuation coefficient of the studied tissue, and μ_{water} the one for water. As shown in Table 1.1, bones have a HU value ranging from 700 for cancellous bone and up to 3000 for dense bones, etc. CT images are thus represented with grayscale, where dark areas correspond to less dense areas, and more dense tissues are brighter. Examples of X-ray radiography are given in Figure 1.4.

Physics of X-rays creation The objective here is to cover the physical aspects of X-rays for CT. Especially, we detail the process for X-ray creation and explain their interactions with the matter so that they are absorbed by the tissues or detected by the CT system.

X-rays are produced in a Coolidge tube[2]. It consists of a glass tube with a cathode and an anode. The process of X-rays creation is as follows:

1. A metal filament is heated by Joule effect level with the cathode. By thermionic emission, the superficial electrons leave the cathode to form an electronic cloud. The intensity of the electron beam is proportional to the intensity of the heating current and the heating duration.

2. Those electrons are then ejected from the filament and sent to the anode with the electrical field which results from the cathode/anode difference of potential. Note

[1]The Hounsfield unit was named after Sir Godfrey Hounsfield who received the Nobel Prize in Physiology or Medicine in 1979, for his part in the invention of CT.

[2]The first Coolidge tube was produced in 1913 by William Coolidge, as an improvement of the Crookes tube.

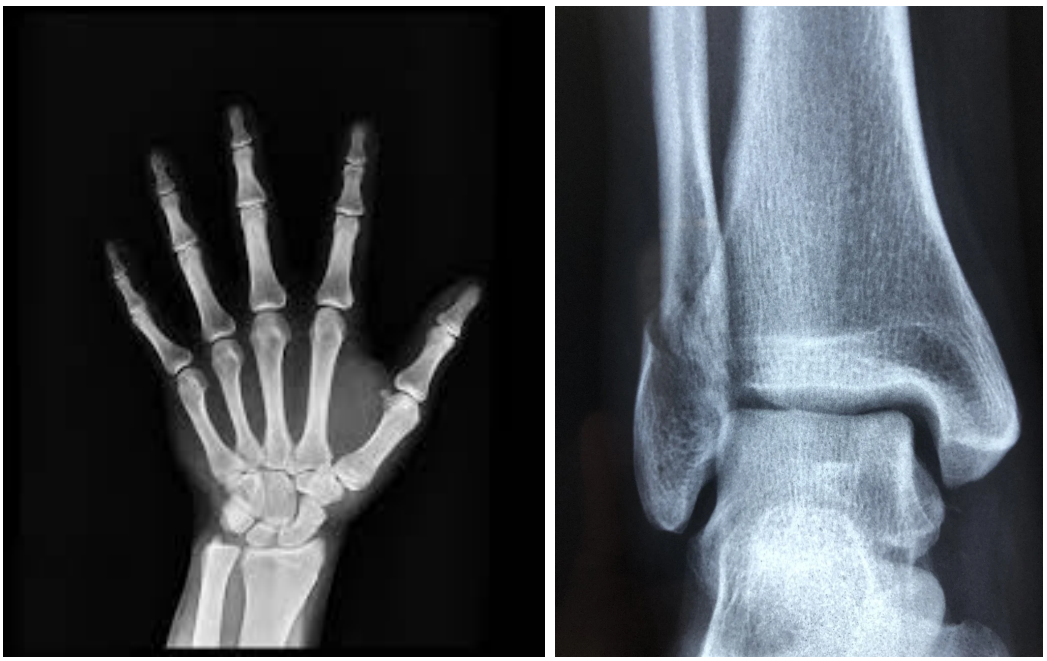


Figure 1.4: Radiography of a hand (left) and a knee (right)

that the Coolidge tube is placed in an empty space so that the electrons are not deviated from their trajectory.

3. At the end, the electrons interact with the anode: either they are deviated or they collide with atoms - we will detail those processes -, but in any case this is where X-rays are created.

In the case where the electrons are deviated, the emitted X-rays are the consequence of the Bremmstrahlung radiation[1] : the electron is attracted by the nuclei of an atom - Coulomb's law - and its trajectory is modified. As the electron is slowed down by the strong electric field near the nuclei, a photon is emitted due to the loss of the electron's kinetic energy. The energy of the emitted photon depends on the loss of kinetic energy.

The second possibility is that the incident electrons collide with an electron of an anode's atom. The electron that is hit is then ejected of its layer and the atom is in an excited state. An electron from an upper layer thus replaces the ejected electron to stabilize the atom; this results in a loss of energy for the electron of the upper layer which leads to the emission of a radiation which energy depends on the layers that are involved during the interaction.

Throughout this process, only a tiny part of the incident energy is converted into X-rays; 99% of the energy is dissipated into heat. The energy efficiency is low and is proportional to both the atomic number of the anode and the potential difference between the electrodes. Generally, the anode is made of tungsten, molybdenum or copper.

As any value can be taken by the loss of energy during the interaction resulting in the emission of X-ray, the braking radiation has a continuous spectrum. Such a spectrum

[1]from bremsen "to brake" and Strahlung "radiation"; i.e., "braking radiation" or "deceleration radiation"

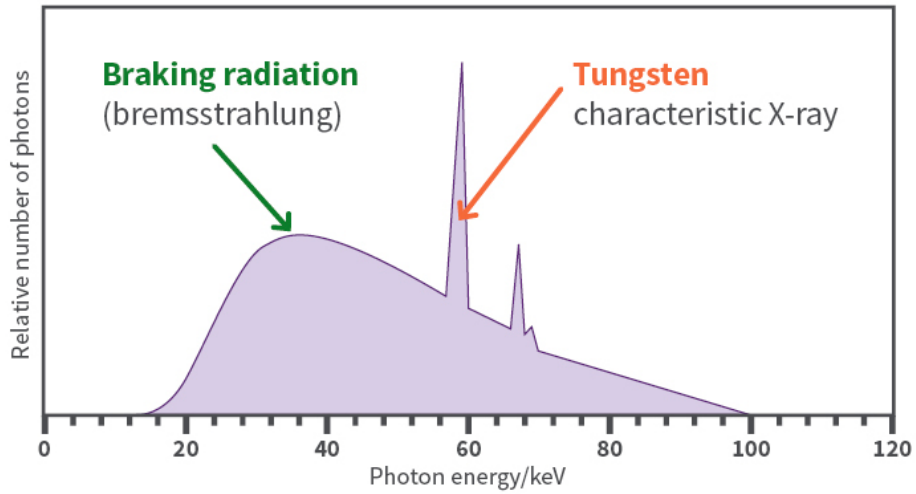


Figure 1.5: Example of the energy spectrum for a tungsten anode. Scheme from the Australian Radiation Protection and Nuclear Safety Agency.[1]

can be observed in Figure 1.5 for a tungsten anode. The peaks that can be observed in the X-rays spectrum corresponds to the quantization of the energy resulting from the changes of state of the anode's atoms.

In the X-rays spectrum there are low-energy photons than can be emitted; these are absorbed by soft tissues when going through the patient, so they are of no interest for imaging, and could even be a source for scattering. For this reason aluminium is used to filter those low-energy photons while letting the higher-energy ones go through.

Once it has been emitted from the Coolidge tube, the X-ray beam passes through the patient. During that pass, photons interact with the matter in different ways. The nature of the photon-matter interaction depends on the energy of the X-rays and on the atomic number of the atoms that constitute the matter. We detail here those different interactions:

- The photoeletric effect : the incident photon ejects an electron of the collided atom and is completely absorbed. The atom is thus ionized. This interaction happens when the photon energy is higher than the electron's binding energy. For a fixed energy, the probability of a photon being absorbed is higher for high value of the atom's atomic number.

- The Compton effect : the incident photon collides with an atom's electron but is not completely absorbed. The electron is ejected and the incident photon is scattered in a random direction with a lower energy. This happens in a similar range of energy compared to the photoelectric effect, which is a source of noise for medical imaging.

- The Rayleigh scattering : the photon excites an atom without colliding with an electron. Getting back to a stable state, the atom emits a photon with similar energy. This however happens for low energy and is less relevant for medical imaging.

[1] https://www.arpansa.gov.au/understanding-radiation/what-is-radiation/ionising-radiation/x-ray

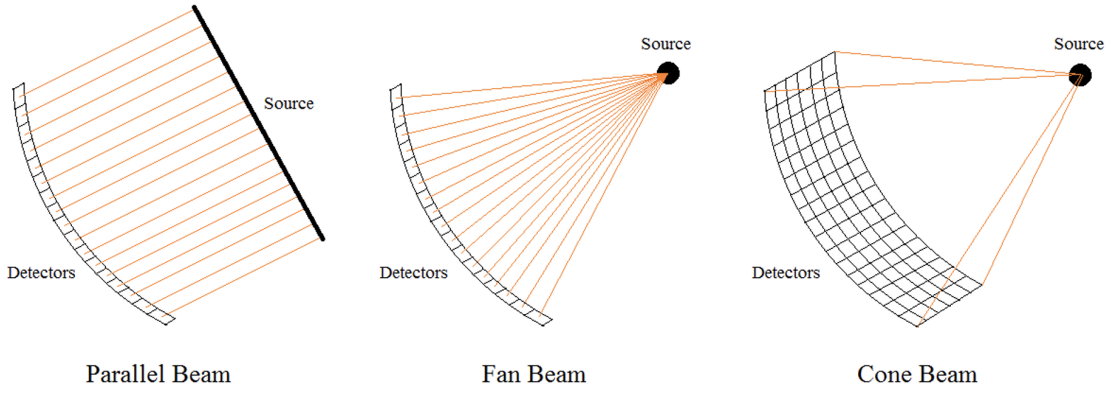


Figure 1.6: Parallel beam (left), fan beam (center) and cone beam (right) CT geometries [Schlifske and Medeiros, 2016].

- Pair production and photodisintegration are the two other interactions but they only appear for high energies that are not used in medical imaging.

As we mentioned, the principle of radiography is to make use of X-rays in order to measure the attenuation between a source and a detector. CT scanners actually use a rotating X-ray tube in addition to a row of detectors that measure the intensity of the beam after going through the body, as illustrated in Figure 1.2. Measurements taken from different angles are then processed with tomographic reconstruction to produce a 3D image of the body: the tomographic reconstruction step represents the core of interest of this thesis.

As X-rays are a ionizing radiation, their administration to biological tissue is harmful and needs to be well justified. Different CT geometries exist and their evolution over the years have improved the image quality with the less radiation for the patient as possible. Fan beam geometry gradually replaced parallel beam and nowadays cone beam geometry is widely used. Improvements of the scanner geometry were made to improve the image quality while reducing the required radiation dose delivered to the patient. Illustrations of those different geometries are presented in Figure 1.6.

1.2.1.2 Modeling

The objective now is to model the aforementioned physics so that we have a mathematical framework to work with. We thus need to describe the attenuation of the intensity of a photon beam going through a material. Let us assume that these photons are emitted in $r = 0$ with intensity I_0 and detected in $r = +\infty$. Let us consider $u(r) = \mu(r)\rho(r)$ the linear absorption coefficient in r. If we consider a small interval δr, the Beer-Lambert law states that the attenuation in δr is proportional to the intensity $I(r)$, $u(r)$ and the width of this interval such as

$$I(r + \delta r) = I(r) - u(r)I(r)\delta r \tag{1.24}$$

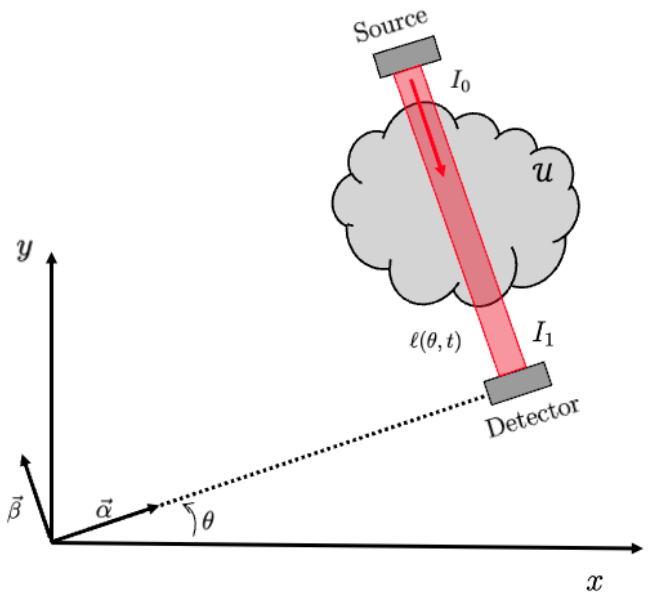


Figure 1.7: Source-detector pair. The detector is located in $(t,0)$ in the $(\vec{\alpha},\vec{\beta})$ basis. I_0 is the source beam intensity, I_1 is the received intensity. $\ell(\theta,t)$ is the line that is perpendicular to $\vec{\alpha}$ and intercepts $(0,\vec{\alpha})$ in t. The distance between source and detector is L.

which gives, dividing by δr and taking the limit when $\delta r \to 0$

$$\frac{d}{dr}I(r) = -u(r)I(r),\ I(0) = I_0. \tag{1.25}$$

Solving (1.25) we get

$$I(r) = I_0\, \mathrm{e}^{-\int_0^r u(r')dr'}. \tag{1.26}$$

If we consider a beam source at $r = 0$ and a received intensity I_1 at the detector located in $r = L$, we can introduce the notion of projection P_ℓ that corresponds to the following measure

$$P_\ell = \log(\frac{I_0}{I_1}) = \int_0^L u(r)dr. \tag{1.27}$$

In other words, taking the logarithm of the ratio between the initial intensity and the received intensity amounts to measuring the integral of the unknown attenuation $u(r)$ along the line of length L for each source-detector pair as illustrated in Figure 1.7. Note that such a model is valid considering a mono-energetic beam.

1.2.1.3 Radon transform

The Radon transform maps a function into the set of its integrals on hyperplanes of $\mathbb{R}^n$. For all unit vectors $\vec{\alpha} \in \mathbb{R}^n$ and all $t \in \mathbb{R}$, the Radon transform of u can be written as

$$Ru(\vec{\alpha},t) = \int_{\vec{v}.\vec{\alpha}=t} u(\vec{v})d\vec{v}. \tag{1.28}$$

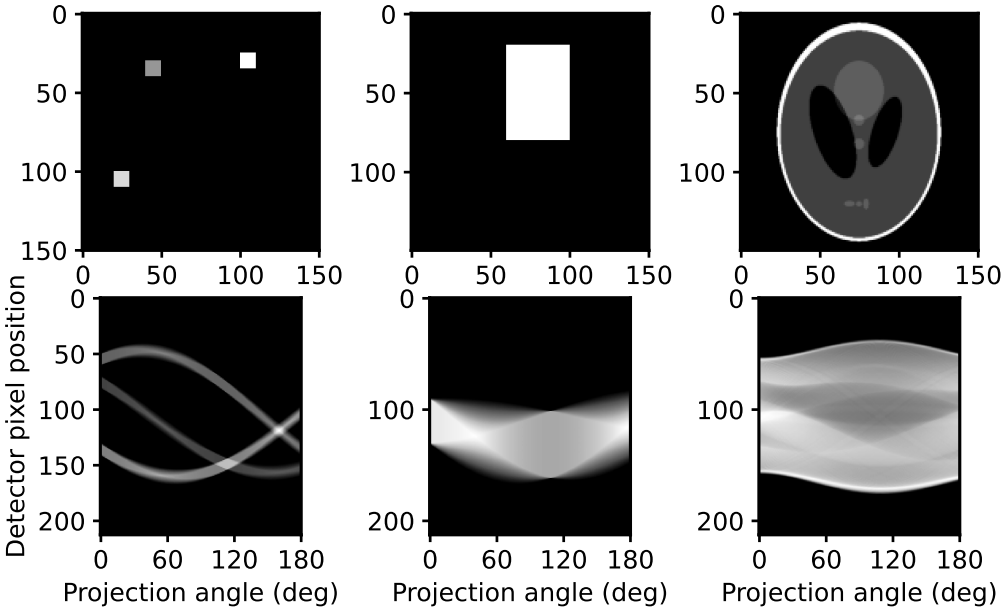


Figure 1.8: Examples of sinograms.

In other words, $Rf(\vec{\alpha}, t)$ is the integral of f on a hyperplane orthogonal to the vector $\vec{\alpha}$.

Considering $n = 2$, we suppose that $\vec{\alpha}$ makes an angleθ with the $0x$ axis; we can in this case write the Radon transform as:

$$Ru(\theta, t) = \int_{l(\theta,t)} u(r)dr \tag{1.29}$$

where $\ell(\theta, t)$ is the line that is perpendicular to $\vec{\alpha}$ and intercepts $(0, \vec{\alpha})$ in t, see Figure 1.7. In this case we obtain an expression that is similar to (1.27), considering that $u(r) = 0$ for $r \notin [0, L]$, with the Radon transform parametrized by an orientation θ and a real number t that describes all possible X-ray measurements of u. The set of measurements $(Ru(\theta_i, .))_i$ for a finite number of angles θ_i is referred to as the set of projections of u.

The graphical representation of the Radon transform $Ru(\theta, t)$ as a function of the projection angle θ looks like a superposition of sine and cosine waves; this is the reason why the set of projections is also called a sinogram. Examples of such sinograms are represented in Figure 1.8.

Since we can now describe all possible X-ray measurements of an image $u(x, y)$ thanks to the Radon transform Ru, the question is whether we can obtain u from the measurements Ru and thus exactly solve the inverse problem. This question involves several elements: the Fourier slice theorem, backprojection and filtered backprojection. We detail these in Section 1.2.3 as there are associated to both CT and PET modalities.

1.2.2 Positron emission tomography

PET is a medical imaging technique that provides functional information to the radiologists. It has a lot of applications in oncology, cardiology and neuropsychiatry. Especially,

it is a method of choice for cancer diagnosis [Coleman, 1999] and for radiation treatment planning [Macmanamus et al., 2009].

PET is a nuclear medicine technique that relies on the injection of a vector-radionuclide pair to the patient. The vector's objective is to get fixed on regions of interest; the radionuclide's decay then allows one to localize it through imaging. The choice of such a pair is motivated by the objective of the imaging protocol: pathology detection, molecular process to track, etc. As tumors consume a relatively high amount of glucose, a common used pair for cancer detection is the ^{18}F-fluorodeoxyglucose (^{18}F-FDG) [Reske and Kotzerke, 2001].

Similarly to the previous section, we first describe the physics of PET imaging and then we describe the model for the of the reconstruction task for PET.

1.2.2.1 Physics of PET

There are different radio-tracers used in PET: we mentioned ^{18}F-FDG, but there are other choices such as fluorothymidine (FLT), rubidium chloride etc. Isotopes for PET are β+ emitter, meaning that their radioactive decay consists in the transformation of a proton into a neutron while a positron and a neutrino are also emitted:

$$\mathrm{p}^{+} \longrightarrow \mathrm{n} + \mathrm{e}^{+} + \nu_{e}. \tag{1.30}$$

After a short distance of a few millimeters maximum [Levin and Hoffman, 1999], the emitted positrons collide with an electron, leading to their annihilation which gives a pair of photons. The total quantity of motion being zero considering the center of gravity of the electron-positron system as referential, the photons are emitted in opposite directions. The positron's kinetic energy being negligible, the photons energy corresponds to their mass : 511 keV.

As illustrated in Figure 1.9, a PET camera is composed of rings composed of elementary detectors surrounding the patient. Those detectors are optimized for the detection of 511 keV photons. An event is stored when two photons are detected in coincidence, which in practice corresponds to the fact that the time between two detections is shorter than a certain threshold, generally a few nanoseconds.

A single detector is composed of a scintillator and a photomultiplier tube. The former converts high-energy photons to visible light, i.e photons with energy between 1.6 eV and 3.3 eV. The photomultiplier tube converts those photons into an electric current.

It is the - almost - simultaneous detection of these coincidence photons that enables the estimation of tracer distribution with image reconstruction. Indeed, we assume that two photons that are detected in coincidence come from a unique electron-positron annihilation. This annihilation takes place in the line of response (LOR) that can be drawn between the two detectors. Contrary to CT imaging where one measures the attenuation of the imaged tissues thanks to an external source, the PET image corresponds to the distribution of the radiotracer injected inside the patient.

1.2.2.2 Modeling

The first step of the modeling in PET imaging is to represent the imaged object in a discretized way. As for CT, the space is divided into voxels, but here the value of each

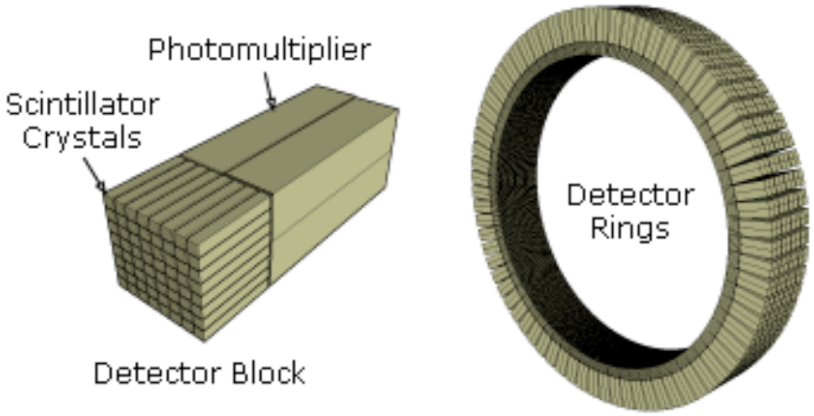


Figure 1.9: PET camera and photomultiplier tube.

voxel is the activity - in Becquerel - of the radiotracer at the corresponding location.

Let us consider two different detectors d_1 and d_2 and let α be the direction perpendicular to the LOR. We denote the tracer concentration as $\lambda(\vec{\alpha}, r)$ for each position r in the LOR $L(d_1, d_2)$ that has a direction perpendicular to the vector $\vec{\alpha}$. We denote the sensitivity function of the system as ψ: it corresponds to the probability of detection for a photon emitted, per time unit. The number of coincidences detected by d_1 and d_2 for an acquisition time τ can be expressed as

$$P_{d_1,d_2} = \tau \int_{L(d_1,d_2)} \lambda(\vec{\alpha}, r)\psi(r, \vec{\alpha})dr. \tag{1.31}$$

The reconstruction task then consists in estimating the tracer concentration λ, given all LORs. Considering the sensitivity function as uniform, the distance between the LOR and the center of the detector ring as t and the angle formed by $\vec{\alpha}$ w.r.t the origin as θ, we can retrieve an expression similar to (1.29) for the Radon transform of the tracer concentration along the FOV:

$$R\lambda(\theta, t) = \int_{\ell(\theta,t)} \lambda(r)dr \tag{1.32}$$

where $\ell(\theta, t) = L(d_1, d_2)$ so that the problem has strong similarities with the one discussed for CT.

In practice one needs to consider corrected PET data. Those corrections result from different phenomena:

- Noise: among the stored coincidences, a large part should be discarded for image reconstruction. Indeed, the Figure 1.10 illustrates the notion of scatter, random and multiple coincidences. In the first case, at least one of the photons is scattered before reaching the detector. For random coincidences, two photons coming from a different annihilation are detected in the considered time window. Multiple coincidences appear when two annihilations happen almost simultaneously and more than two photons are detected in the coincidence window. In all of those cases, the detected events should not be taken into account as the LORs do not contain the position of annihilation.

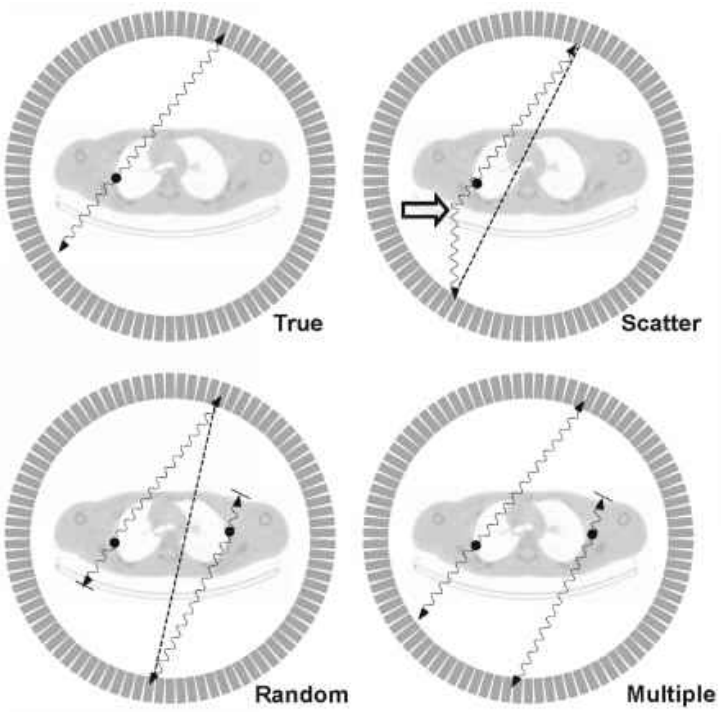


Figure 1.10: Illustration of scatter, random and multiple coincidences [Bailey et al., 2005].

- Attenuation: when a pair of photons is emitted, it is possible that one of the photons interacts with the matter and does not hit the detectors, or it does with an energy lower than the detectability threshold. The probability of such an interaction depends on the nature of the tissues and can be different for different lines of response. This is modeled with an attenuation sinogram.

- Normalization: in order to reduce the amount of data to be stored, the closest lines of response are binned. However the scanner geometry leads to the fact that all bins do not contain the same number of detector pairs. The normalization sinogram accounts for such differences, as well as the variations in the crystals efficiency.

All of these corrections - random, scattered, attenuation and normalization - are estimated and stored before the image reconstruction.

The coincidences can be stored into a sinogram with the x-axis representing the distance between the LOR and the center of the scanner, and the y-axis corresponds to the angle between the LOR and a given direction.

However in PET, the coincidences can also be stored in the so-called list-mode format. In that case each event is recorded into a file that contains the position of the detected event as well as the corresponding time and energy information. Coincidence time is of particular importance for time of flight (TOF) PET imaging. We will go back to the particular case of TOF PET imaging in Chapters 5 and 6 as it represents a major application of our work.

1.2.3 Analytical reconstruction

We will go back later in this thesis on the particularities of image reconstruction for the different imaging modalities that we showed. We focus here on the analytical resolution of the problem that is common to both PET and CT: retrieving the object which is at the origin of the measurements.

1.2.3.1 Backprojection

Since we can describe all measurements of an image $f(x,y)$ - whether it represents the attenuation $u(x,y)$ or the activity $\lambda(x,y)$ - with the Radon transform (1.29), the question is whether R is invertible.

The backprojection consists in mapping the projections $Rf(\theta,t)$ into the image space. For every angle θ, we add the value $Rf(\theta,t)$ to each point on the line $l(\theta,t)$ i.e to all $(x,y) \in \mathbb{R}^2$ such that $t = x\cos\theta + y\sin\theta$ so that the backprojection can be written

$$f_{BP}(x,y) = \int_0^\pi Rf(\theta, x\cos\theta + y\sin\theta)d\theta. \tag{1.33}$$

The backprojection corresponds to the adjoint R^* of the Radon transform but not the inverse R^{-1}. It can be shown that

$$R^*Rf(s) = \int_{\mathbb{R}^2} \frac{f(v)}{||s-v||_2}dv \tag{1.34}$$

i.e the backprojection retrieves the unknown function f convolved to a smoothing kernel. There are too many low frequencies that are backprojected as compared to the high frequencies; these low frequencies correspond to the smooth features of the image. We will see in what follows how it is possible to actually invert the Radon transform.

1.2.3.2 Fourier slice theorem

The Fourier slice theorem states that the Fourier transform of a projection $Rf(\theta,.)$ corresponds to a line in the Fourier transform of the image that we denote as $\tilde{f}$. This line intercepts the origin and forms an angle of θ with the horizontal axis.

This theorem theoretically allows reconstructing the image. One just needs to compute the Fourier transform of each projection and report those values for every corresponding line in the frequency domain. We denote the Fourier transform of a projection for an angle θ as $\widetilde{Rf}(\theta,\nu)$ for a frequence ν. Writing $\nu_x = \nu\cos\theta$, $\nu_y = \nu\sin\theta$, the value of the Fourier transform of the image $\tilde{f}(\nu_x,\nu_y)$ can be computed as

$$\tilde{f}(\nu_x,\nu_y) = \tilde{Rf}(\theta,\nu). \tag{1.35}$$

The inverse Fourier transform then gives the corresponding slice: for all $(x,y) \in \mathbb{R}^2$,

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{f}(\nu_x,\nu_y)e^{2i\pi(x\nu_x+y\nu_y)}d\nu_x d\nu_y \tag{1.36}$$

In that sense, R is invertible - since we can use the inverse Fourier transform - but in practice we only have a finite number of projections so we have missing values in the Fourier domain of the image, especially for high frequencies.

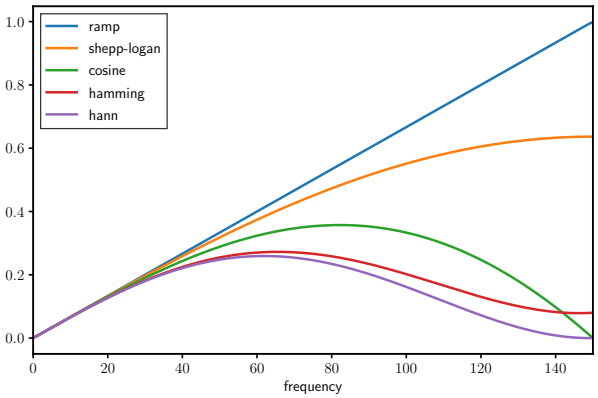


Figure 1.11: Filters for backprojection

The missing values can be completed with interpolation but the computation of the Fourier transform as well as the interpolation on complex numbers make this method inefficient in practice from a computational point of view.

1.2.3.3 Filtered backprojection

The filtered backprojection (FBP) allows getting rid of the interpolation and resampling steps in the frequency domain. To put it simply, it consists of two steps: filtering the projections then performing the backprojection.

Injecting (1.35) into (1.36) and performing the variable change $(\nu_x, \nu_y) \mapsto (\nu, \theta)$ we have the Jacobian value that is $|\nu|$ which gives

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} \widetilde{Rf}(\theta,\nu) e^{2i\pi\nu(x\cos\theta + y\sin\theta)} |\nu| d\nu d\theta. \tag{1.37}$$

The inner integral is the inverse Fourier transform of the filtered projection's Fourier transform. The formulation (1.37) corresponds to the inversion of the Radon transform R with the FBP. Indeed, if we denote the ramp filter operator as H, the FBP corresponds to R^*H as written in (1.37) and we do have $R^*HR = I$ where I is the identity. If one has access to all the projections in all possible directions, then it is possible to exactly reconstruct f.

An interpretation of the FBP is that is counter-balances the over-representation of the low frequencies in (1.33) by giving more importance to high frequencies. This is made possible by applying the ramp filter to the projections before backprojecting.

In practice there are several different filters that can be used instead of the ramp filter. We represent some of them in Figure 1.11. The general idea is that the ramp filter accentuates the high frequencies, which are generally dominated by noise. The choice of filters that cut those high frequencies can then be relevant in the case of noisy data, but the actual choice of the filter is dependent on the finality of the image: detection, quantification, etc. Also, the choice of cut-offs along with the potential loss of information that comes with them make the practical use of such filters more complicated compared to simply using the ramp filter.

We show in Figure 1.12 examples of reconstructions with simple backprojection along with filtered backprojections with ramp filter. We observe that when the "quality" of the measurements data decreases - few projection angles, noise -, even the filtered backprojection is unefficient to get a satisfying reconstruction. This represents the starting point of our main objective in this thesis which is to study and propose algorithms for image reconstruction - for both PET and CT - that enhance the quality of the reconstruction compared to analytical methods.

1.3 Objectives

In this thesis we consider several inverse problems for the tomographic reconstruction task: a CT application, a PET application and we also have a transmission electronic microscopy application in the next chapter. Some problems are specific to each application, but in this chapter we have shown that there is a common basis for all of the reconstruction methods. All of them have for objective to solve an inverse problem with a linear operator - the Radon transform. Analytical reconstructions represent a first solution to retrieve an image given measurements data. Besides, the quality of a FBP-reconstructed image might be satisfying enough for some applications. Over the years, research in tomographic reconstruction has however allowed to improve the quality of CT or PET images; especially, methods that are more robust to noise or incomplete data compared to analytical reconstructions received a lot of interest. In the next chapters we will focus on such methods, namely iterative and deep learning-based methods.

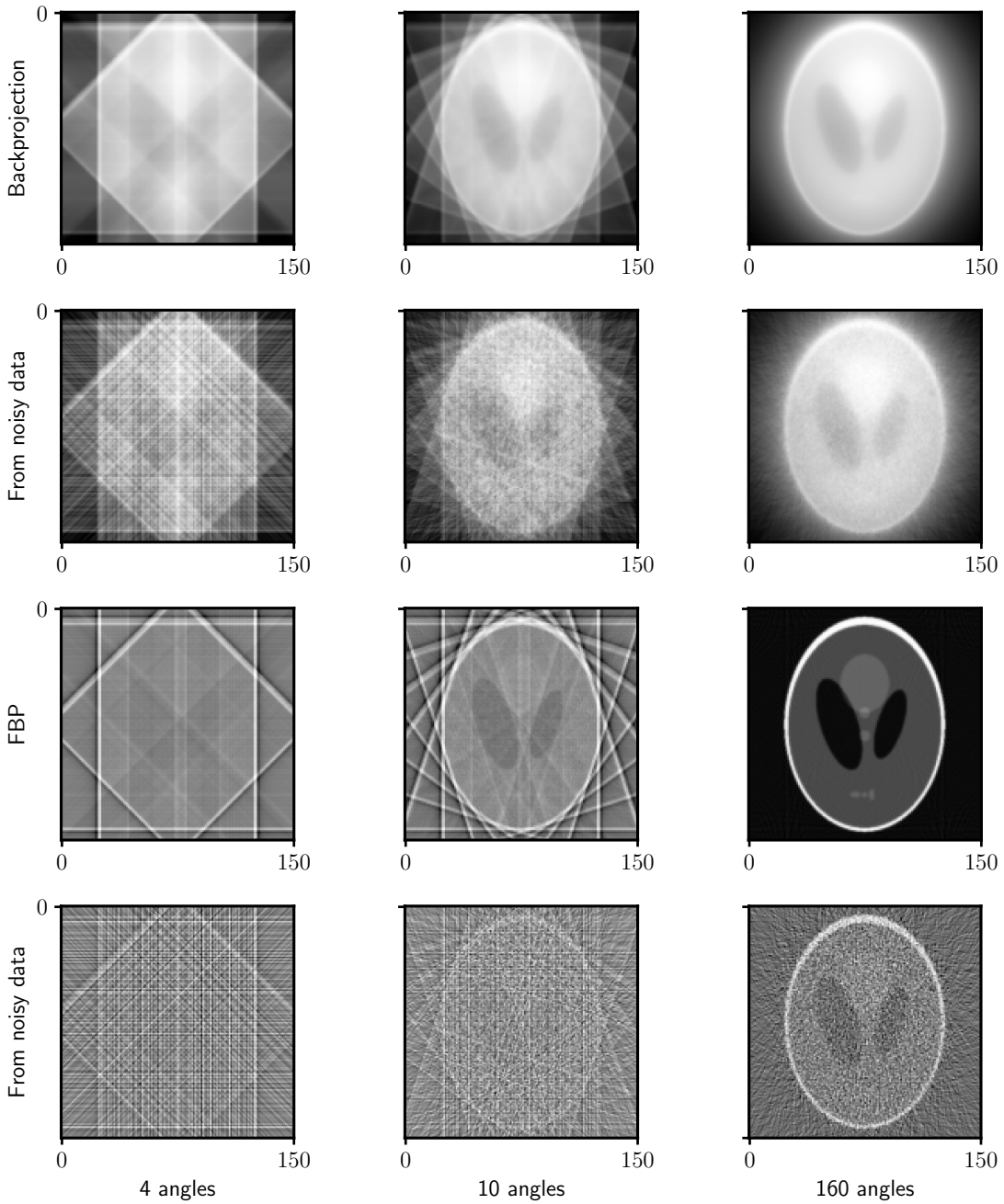


Figure 1.12: Backprojection (first and second row) and Filtered backprojection (third and fourth row) with different measurements. The second and fourth rows correspond to reconstructions obtained when Poisson noise is applied to the projections. The x-label corresponds to the number of projection angles in the sinogram, with angles that range from 0 to 180 degrees.

Chapter 2

Model-based iterative methods for tomographic reconstruction

An advantage of analytical methods such as the ones presented in the previous chapter is that they can be expressed in a closed form. In practice their solutions are also fast to compute. These methods however suffer from a deteriorated quality of the image as soon as the measurements are not ideal: noise, missing projection angles, etc. When it is the case, algorithms that consist in iteratively estimating a solution $\hat{f}$ to minimize a cost function are generally more robust to data perturbations. We refer to these methods as iterative algorithms.

Iterative algorithms allow to take into account the physics of the acquisition, the noise model for the data and potentially an *a priori* on the unknown solution in the case of regularized methods. There are plenty of iterative algorithms for tomographic reconstruction. The choice of one of them depends on the cost function to minimize, but also on the computation time, convergence speed and numerical stability. We present in this chapter some of the common iterative algorithms that have been applied to tomographic reconstruction. The idea is then to focus on specific examples and show how such algorithms can be compared with each other on simulated and real data. In the last section of this chapter, we discuss the limitations of these methods based on the experiments we performed.

2.1 Examples of iterative algorithms for tomographic reconstruction

2.1.1 ART

ART (Algebraic Reconstruction Technique) [Gordon et al., 1970] is the oldest iterative method used for tomographic reconstruction. Considering the matrix of the Radon operator as $A = (a_{ij})_{M \times N}$ and denoting each row vector i by a_i, the discrete tomographic reconstruction problem consists in solving the linear system $Af = p$ as:

$$\begin{cases} \langle a_1, f\rangle &=& a_{11}f_1 + a_{12}f_2 + \cdots + a_{1N}f_n &=& p_1 \\ \langle a_2, f\rangle &=& a_{21}f_1 + a_{22}f_2 + \cdots + a_{2N}f_n &=& p_2 \\ & & \vdots & & \\ \langle a_M, f\rangle &=& a_{M1}f_1 + a_{M2}f_2 + \cdots + a_{MN}f_n &=& p_M \end{cases} \tag{2.1}$$

Each equation corresponds to the definition of a hyperplane of $\mathbb{R}^N$. The solutions of the system are in the intersection of those hyperplanes. To find such a solution, the Kaczmarz method is used. It consists in iteratively projecting the current solution onto those hyperplanes. The orthogonal projections can be written as

$$\Pi_i(f) = f - \frac{\langle a_i, f\rangle - p_i}{\|a_i\|^2} a_i$$

so that the ART algorithm writes

Algorithm 2: ART algorithm

```
Initialize f^0
for n = 0, 1, 2, ... do
    for i = 1, ..., M do
        f^{n+1} ← Π_i(f^n)
```

where the speed of convergence for the algorithm depends on the angles formed between the different hyperplanes.

Most of the time a solution of (2.1) does not exist; rather ART will compute $\arg\min_f ||Af - p||_2^2$. A major drawback of such a method is the high sensitivity to noise.

If algebraic methods were initially an alternative to analytical reconstructions, other techniques such as SIRT (Simultaneous Iterative Reconstruction Technique) or MLEM (Maximum Likelihood Expectation Maximization) have later outperformed them.

2.1.2 SIRT

SIRT [Gilbert, 1972] aims at minimizing the cost function

$$||Af - p||_2^2. \tag{2.2}$$

From a statistical point of view, this is equivalent to maximizing the log-likelihood when considering data corrupted by Gaussian noise, i.e when $p = Af + \epsilon$, where $\epsilon = (\epsilon_1, \ldots, \epsilon_M)$ and all the ϵ_i are Gaussian random variables - see 1.1.4.

To minimize (2.2) and given an initial estimation f^0, the algorithm writes:

$$f^{n+1} = f^n - \lambda \frac{1}{A^*1} A^* \left[\frac{Af^n - p}{A1} \right] \tag{2.3}$$

with A^* the adjoint of the projection matrix A (i.e the backprojection operator), λ is the step size, 1 is the unit vector and vector-vector operations are performed element-wise.

The image is therefore updated with the backprojection of the normalized error between p and the projected estimate, considering sensitivity correction.

The two main differences with SIRT compared to ART are mentioned in [Gilbert, 1972]:

- In each iteration of SIRT, data from all the projections are applied simultaneously to update f^n, whereas for ART they are applied sequentially. This has the major advantage of increasing the stability of the algorithm.

- Sensitivity correction is performed in SIRT as shown in (2.3), which is not the case for ART.

2.1.3 MLEM

One of the most common iterative algorithm used in emission tomography is MLEM (Maximum Likelihood Expectation Maximization). It was proposed in [Shepp and Vardi, 1982] and [Lange et al., 1984] as an algorithm for PET imaging in which Poisson noise is dominant. The method consists in using a maximum-likelihood solution, for data corrupted by Poisson noise, that is found thanks to the expectation maximization (EM) algorithm [Dempster et al., 1977]. The popularity of MLEM came from its ability to produce good image quality in the presence of high noise levels.

The EM algorithm is a general method to maximize the likelihood Q of some random variable that depends on parameters θ and on data that cannot be observed. The observation of the realization of the random variable thus depends on latent variables Z, so the likelihood Q needs to be estimated. Once Q is estimated, we aim to maximize it over θ. These two steps are successively realized at each iteration until convergence.

Mathematically, the first step (E-step) consists in assuming a current value θ^k for the parameters, and computing the expectation over the latent variables Z as

$$Q(\theta|\theta^k) = E_Z(\log F(\theta, Z)|y, \theta^k) \tag{2.4}$$

knowing the observed data y, the current value of parameters θ^k and the probability density function F. The second step (M-step) is to maximize $Q(\theta|\theta^k)$ over all θ to get θ^{k+1}, and the process is then repeated.

We apply here the EM algorithm to a reconstruction problem where we are maximizing the likelihood of the data. We consider that the projections p are the observed data - corresponding to y - that follow a Poisson distribution with mean Af. The latent variable Z corresponds in this case to the contribution of pixel/voxel j to the LOR i such that we have $p_i = \sum_j z_{ij}$. The quantity of interest is here $\theta = f$, i.e the image. We also assume that for each (i, j) pair, z_{ij} follows a Poisson distribution with expected value $a_{ij} f_j$. The likelihood for measurements p, denoted as $\pi(p|f, Z)$, can be written

$$\pi(p|f, Z) = \prod_i (\sum_j z_{ij})^{p_i} \frac{e^{-\sum_j z_{ij}}}{p_i!} \tag{2.5}$$

so the function to maximize, considering the expectation over Z of the log-likelihood

denoted as $Q(f|f_1,\dots,f_K)$, is then

$$Q(f|f_1,\cdots,f_K) = \sum_i \left(p_i \log \sum_j a_{ij} f_j - \sum_j a_{ij} f_j - \log p_i! \right). \tag{2.6}$$

The EM strategy applied to this maximization problem leads to the algorithm known as MLEM:

$$f_k^{n+1} = f_k^n \frac{1}{\sum_i a_{ik}} \sum_i \frac{a_{ik} p_i}{\sum_j a_{ij} f_j^n} \tag{2.7}$$

where f_k corresponds to each pixel/voxel of the reconstruction. One can rewrite the update step in matrix form, with A^* the adjoint of A as:

$$f^{n+1} = \frac{f^n}{A^*1} A^* \left[\frac{p}{A f^n} \right]. \tag{2.8}$$

Here the update step consists in multiplying the current estimate with the backprojection of the error, with sensitivity correction.

MLEM ensures the positivity of the estimated image. This is not only the case at convergence but also at every iteration of the algorithm. We note however some of the main drawbacks one can encounter when using MLEM:

1. The MLEM algorithm is relatively slow to converge.

2. Once convergence is reached regarding the cost function decrease, the image gets more and more noisy: in practice early stopping needs to be performed to prevent the reconstruction from being too noisy. We will see later that regularization is also a solution.

3. An important issue in PET imaging is the fact that convergence is slower in "cold" areas, i.e in areas where the activity is less important. Indeed if we denote the negative log-likelihood in (2.6) as $L(f)$, then Equation (2.7) can also be written

$$f_k^{n+1} = f_k^n + \frac{f_k^n}{\sum_i a_{ik}} \frac{\partial L}{\partial f_k}(f_k^n) \tag{2.9}$$

   so that it is a gradient descent where the step is proportional to the voxel activity, which explains the slower convergence for cold areas.

4. The positivity constraint also leads to a positive bias in the reconstructed image.

OSEM and other MLEM variations In order to get a faster algorithm compared to MLEM, a widely used method in practice is the Ordered Subset Expectation Maximization (OSEM) algorithm [Hudson and Larkin, 1994]. The idea is to divide the measurements data space into m different subsets. One iteration of OSEM then consists of m sub-iterations, each of them applying the MLEM update equation (2.8) for a particular subset, considering only a fraction of all the projections p.

In practice an iteration of OSEM gives an estimate roughly similar to what could be obtained with m iterations of MLEM, which is a significant improvement in terms of

computation time. This makes OSEM more used in clinical practice compared to MLEM. However OSEM does not converge but rather produces a limit cycle.

There are other variations of MLEM in the literature and we do not cover all of them here. Examples include ABEMML [Byrne, 1998] where authors propose an algorithm which generalizes MLEM by constraining the solution between two bounds α and β, also allowing α to be negative.

2.1.4 FISTA acceleration

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [Beck and Teboulle, 2009] is an acceleration technique that increases the convergence rate of first order methods for non-smooth convex problems. As both SIRT and MLEM can be seen as gradient descent methods, FISTA acceleration can be applied to those algorithms. Specifically, a sequence of scalars is computed as $t^0 = 1$ and

$$t^{n+1} = \frac{1 + \sqrt{1 + 4(t^n)^2}}{2} \tag{2.10}$$

then the $n+1$-th estimate $f^{\hat{n+1}}$ obtained with either SIRT or MLEM is updated as

$$f^{n+1} \leftarrow f^{\hat{n+1}} + \left(\frac{t^n - 1}{t^{n+1}}\right)\left(f^{\hat{n+1}} - f^n\right) \tag{2.11}$$

It is shown in [Beck and Teboulle, 2009] that FISTA algorithm has a convergence rate of $O(1/n^2)$. In our experiments we will use such an acceleration technique.

2.1.5 Regularization

Algorithms such as ART, SIRT and MLEM have a drawback in common: they tend to produce a noisy image when measurements are noisy, especially when no early-stopping is performed. A solution is to consider *a priori* information on the solution. This consists in adding a penalty term to the functional to minimize, in addition to the data-fidelity term. Giving enough weight to this regularization term, this allows obtaining reconstructions where noise is less visible. We refer to the algorithms that aims at minimizing such a functional as regularized algorithms.

The functional that we consider can be written

$$\min_f d(Af, p) + R(f) \tag{2.12}$$

where d is a data-fidelity term - chosen depending on the assumed noise on the data - and $R(f)$ is the penalty that expresses the *a priori* knowledge on the image to reconstruct.

Several algorithms that converge to the solution of the penalized function (2.12) have been proposed. In PET, a modified EM algorithm for penalized likelihood estimation was used in [De Pierro, 1995]. In clinical practice for PET, the BSREM algorithm [De Pierro and Yamagishi, 2001, Ahn and Fessler, 2003] has also been widely used.

Here we present the particular case where the penalty term corresponds to the Total Variation (TV).

2.1.6 Total Variation

TV is a semi-norm and for images f it can be defined as:

$$TV(f) = \||\nabla f|\|_1 \tag{2.13}$$

where ∇ is the gradient operator and $\||\cdot|\|_1$ corresponds to the sum of modulus of gradients for all pixels/voxels in the image.

Total Variation can be of particular interest for image denoising [Rudin et al., 1992]. It allows solutions to preserve edges whilst smoothing flatter areas. If g is the noisy image and u is the potential solution, the TV denoising problem assuming Gaussian noise is

$$\min_u \|u - g\|_2^2 + \alpha TV(u) \tag{2.14}$$

with α a weighting parameter.

In [Chambolle, 2004] authors propose a method based on the dual formulation of the problem (2.14) and show that the solution has the form $u = g - \alpha \text{div}\varphi$, where div is the divergence and with φ that can be computed iteratively as $\varphi^0 = 0$ and

$$\varphi^{n+1} = \frac{\varphi^n + \tau\nabla\left(\text{div}\varphi^n - g/\alpha\right)}{1 + \tau|\nabla\text{div}\varphi^n - g/\alpha|}$$

with $\tau \leq 1/8$ (in 2D) the gradient step. We refer to this denoising algorithm as TVDenoise.

2.1.7 Total Variation regularization

Initially proposed for image denoising, TV has proven to be efficient in inverse problems for tomographic reconstruction, especially to restore piece-wise constant images. This has been made possible by considering TV as a regularization term in the functional to minimize [Sidky et al., 2006]. This has shown interesting results in missing angle tomography [Banjak et al., 2018] and in low-statistics emission tomography acquisitions [Sawatzky et al., 2008]. In what follows we present the regularized versions of SIRT and MLEM.

2.1.7.1 Regularized SIRT

The minimization problem when considering the TV regularized version of SIRT is

$$\min_f \|Af - p\|_2^2 + \alpha TV(f) \tag{2.15}$$

where α is the regularization parameter.

In [Banjak et al., 2018] the algorithm SIRT-FISTA-TV, a FISTA accelerated version of SIRT-TV, is used in order to reconstruct volumes where Gaussian noise is considered on the projections. Each iteration consists of a SIRT update as (2.3), followed by TV

denoising and FISTA acceleration. The corresponding SIRT-FISTA-TV algorithm can be written as

Algorithm 3: Pseudo-code of SIRT-FISTA-TV reconstruction algorithm

$\lambda, \alpha, f^0,\ Niter,\ NTViter$
$t^0 \leftarrow 1$
 # SIRT-FISTA-TV iteration
for $n \in [1, 2, \ldots, Niter]$ **do**
  # SIRT update
  $f^{n+\frac{1}{2}} = \texttt{SIRT}\,(f^n)$
  # TV minimization
  $f^{\hat{n+1}} = \texttt{TVDenoise}(f^{n+\frac{1}{2}}, \alpha, NTViter)$
  # FISTA acceleration
  $t^{n+1} = \frac{1+\sqrt{1+4(t^n)^2}}{2}$
  $f^{n+1} = \hat{f^n} + \left(\frac{t^n-1}{t^{n+1}}\right)\left(f^{\hat{n+1}} - f^n\right)$

2.1.7.2 Regularized MLEM

Similarly to SIRT-TV for Gaussian noise, it is possible to consider a TV regularized version of MLEM for Poisson noise, that we denote as EM-TV. The minimization problem here is

$$\min_{f \geq 0} KL(Af, p) + \alpha TV(f) \tag{2.16}$$

where KL is the Kullback-Leibler divergence.

The following iterative scheme is proposed in [Sawatzky et al., 2008]

$$f^{n+\frac{1}{2}} = \frac{f^n}{A^*1} A^*\left(\frac{p}{Af^n}\right) \tag{2.17}$$

$$f^{n+1} = f^{n+\frac{1}{2}} - \alpha \frac{f^n}{A^*1} h^{n+1} \tag{2.18}$$

with $h^{n+1} \in \partial TV(f^{n+1})$. One can recognize a MLEM reconstruction step followed by a TV denoising step.

It is also possible to write the denoising minimization problem at each step as

$$f^{(n+1)} = \arg\min_u \{H(u) + \alpha TV(u)\} \tag{2.19}$$

where $H(u) = \langle u - f^{n+\frac{1}{2}} \log u, s\rangle$ with $s = A^*1$.

Using the Fenchel-Rockafellar duality theorem as well as the Lagrangian for optimization we get:

$$f^{n+1} = \frac{s f^{n+\frac{1}{2}}}{s + \alpha \operatorname{div} \phi^*}, \tag{2.20}$$

with ϕ^* that is obtained iteratively with the following semi-implicit gradient descent scheme:

$$\phi^{k+1} = \frac{\phi^k - \tau z^k}{1 + \tau|z^k|} \tag{2.21}$$

with $z^k = \nabla(\frac{sf^{n+1/2}}{s+\text{div}\,\phi^k})$. The convergence of the algorithm is proved in [Maxim et al., 2018] for α and τ small enough. For instance in 2D, upper bounds are given as $\alpha \leq s_{\min}/4$ and $\tau \leq \frac{(s_{\min}-4\alpha)^2}{8\alpha||sf^{n+1/2}||_\infty}$, where $s_{\min}$ is the smallest value of s.

2.1.8 Chambolle-Pock algorithm applied to tomography

We consider here the minimization of (2.12) without isolating the reconstruction and the denoising steps, with the Chambolle-Pock (CP) algorithm.

We will focus on the problem that involves Poisson noise, with the Kullback-Leibler divergence as the data-fidelity term. For a KL distance and a TV regularization term, we rewrite the objective function as

$$\min_f \sum_i [Af - p + p\log p - p\log(pos(Af))]_i + \alpha TV(f) \tag{2.22}$$

where $pos(x) = \max(0, x)$ and log operates on the components of its argument - except when it is zero, for which case in practice one may consider an output of zero for the function log.

So we have, in order to fall into the same case as for the general CP algorithm (see 1.1.3.2, Algorithm 1 and the corresponding notations)

$$F(y,z) = F_1(y) + F_2(z)$$

$$F_1(y) = \sum_i [Af - p + p\log p - p\log(pos(y))]_i$$

$$F_2(z) = \alpha TV(z)$$

which amounts to taking $K = (A, \nabla)$ and $G(f) = 0$ in Algorithm 1.

The CP algorithm for our application is given in [Sidky et al., 2012] as:

Algorithm 4: Pseudo-code for N-steps of the KL-TV CP algorithm. $\|(A,\nabla)\|_2$ is the largest singular value of (A, ∇).

$L \leftarrow \|(A,\nabla)\|_2; \tau \leftarrow 1/L; \sigma \leftarrow 1/L; \theta \leftarrow 1; n \leftarrow 0$
initialize f^0, g^0 and q^0 to zero
$\hat{f}^0 \leftarrow f^0$
while $n \leq N$ **do**
  $g^{n+1} \leftarrow \frac{1}{2}\left(\mathbf{1}_D + g^n + \sigma A\hat{f}^n - \sqrt{(g^n + \sigma A\hat{f}^n - \mathbf{1})^2 + 4\sigma p}\right)$
  $q^{n+1} \leftarrow \alpha(q^n + \sigma\nabla\hat{f}^n)/\max(\alpha\mathbf{1}, |q^n + \sigma\nabla\hat{f}^n|)$
  $f^{n+1} \leftarrow f^n - \tau A^T g^{n+1} + \tau div\ q^{n+1}$
  $\hat{f}^{n+1} \leftarrow f^{n+1} + \theta(f^{n+1} - f^n)$
  $n \leftarrow n + 1$

The Chambolle-Pock algorithm is efficient but appears to have a quite slow rate of convergence. Several adaptations can be found in the literature, for which the choice of parameters improves the speed of convergence. These parameters can be fixed or adaptive. We present here a preconditioned version of the Chambolle-Pock algorithm (PCP) [Pock and Chambolle, 2011].

The initial parameters are replaced by vectors as follows:

$$\sigma \rightarrow \Sigma = \frac{1_Y}{|(A, \nabla)|1_X} \tag{2.23}$$

$$\tau \rightarrow T = \frac{1_X}{|(A, \nabla)|^T 1_Y} \tag{2.24}$$

where X is the image space and Y denotes here the measurements space, and $|(A, \nabla)|$ is the matrix formed by taking the absolute value of each element of (A, ∇).

The proximal mapping is then defined as

$$prox_\sigma[F](y) = \arg\min_{y'} F(y') + \frac{1}{2}(y - y')^T(\frac{y - y'}{\Sigma}). \tag{2.25}$$

With PCP, one does not need to compute $\|(A, \nabla)\|_2$. In a general case, deriving the proximal mapping can be non trivial. For the KL-TV optimization problem though, it appears that one just needs to replace σ by Σ and τ by T in Algorithm 4. In [Sidky et al., 2012] it is argued that convergence is faster with PCP compared to the ordinary CP algorithm especially for small α; in our tests, we find the convergence speed to be improved even for relatively high values of α.

There are others ways to initialize CP algorithm such as n-OCP [Qiao et al., 2019] and ACP [Chambolle and Pock, 2016] but we do not recall them here since better performance was observed with PCP in our experiments.

In this section we introduced several - more or less common - algorithms for tomographic reconstruction. Some of them are better suited to Gaussian noise, and the others for Poisson noise. Real data generally contain a mix of both type of noise, although depending on the application one can be dominant over the other. In the next section we put the emphasis on the evaluation and comparison of these algorithms depending on the mix between Poisson and Gaussian noise considered in the data.

2.2 Comparative study for a mix of Poisson-Gaussian noise

We derived algorithms that are commonly used and specifically designed for Poisson OR Gaussian noise. The Poisson-Gaussian model is often simplified with variance stabilization techniques [Murtagh et al., 1995]. Few studies actually deal with the mixed Poisson-Gaussian noise often encountered in real data [Benvenuto et al., 2008, Jezierska et al., 2012, Calatroni et al., 2017]. In [Calatroni et al., 2017] considering both Poisson and Gaussian noise leads to a cost function with two data-fidelity terms, which implies tuning an extra weighting parameter. This makes practical parameters tuning more complex.

The study that we aim to perform therefore consists in evaluating the performance of algorithms derived specifically for one type of noise, on data with mixed Poisson-Gaussian

noise. The main idea is to see how the performance of these algorithms is affected when the noise distribution is different than the noise considered when deriving the algorithms. It is of major importance for practical applications, since this amounts to finding answers to the general question: how the performance of the reconstruction method is affected if the assumption made on the noise is not accurate?

For this study we compare three algorithms: Preconditioned Chambolle-Pock with KL-TV cost function, SIRT-FISTA-TV, and a FISTA accelerated version of the algorithm from [Maxim et al., 2018] that we call EM-FISTA-TV. The cost function that is minimized in this study is thus (2.12) with either the ℓ_2 norm or Kullback-Leibler divergence as discrepancy term d and with the total variation as regularization term that is weighted with the α parameter.

We compare the different algorithms in terms of quality of reconstruction and computational efficiency on simulated and experimental data from transmission electron microscopy.

The experiments that we perform are split into two parts. The first part consists in preliminary tests. We fix a configuration for the Poisson-Gaussian noise: we show how one compare algorithms performance with quantitative results on synthetic data. We also assess whether the observations made on simulated data are consistent with real data. For the latter we consider the particular case of three-dimensional data for a transmission electron microscopy sample: tomographic reconstruction is in this case similar to CT reconstruction with parallel beams described in Chapter 1. Results from these experiments were published in [Leuliet et al., 2021a].

In the second part we extend this preliminary study to multiple Poisson-Gaussian noise configurations. We use synthetic data and we vary the ratio between the amount of Gaussian and Poisson noise. In this case the performance of the different algorithms is evaluated with respect to this ratio and we assess the extent to which such algorithms can be used when actual noise configuration does not match the assumption made on the data likelihood.

2.2.1 Fixed noise configuration: simulation details

For the synthetic data we use a 256×256 pixel Shepp-Logan phantom represented in Figure 2.1. Projections and back-projections were computed on GPU thanks to ASTRA Toolbox [van Aarle et al., 2015] in Python. TV denoising is also implemented on GPU to fasten computation. The dataset is constructed as follows. Random values are drawn from a Poisson law with mean the exact projections, followed by the addition of a zero-mean Gaussian noise with standard deviation $\sigma = 2\%$ of the maximum value in the projections. According to preliminary experiments, the number of iterations to convergence within the internal denoising loop has been fixed to 100 for SIRT-FISTA-TV and 30 for EM-FISTA-TV. We evaluate the reconstruction quality with the Mean Squared Error (MSE) defined for two images f and $g \in \mathbb{R}^N$ as

$$MSE(f,g) = \frac{1}{N}\sum_{i=1}^{N}(f_i - g_i)^2.$$

The MSE values that we present are the ones obtained when convergence is reached. The convergence speed of the different algorithms is calculated in terms of outer iterations

and time; for this we have studied the decrease of both the MSE and the cost function for each algorithm. As we will discuss in more details later, the regularization parameter has a major influence on the reconstructed solutions. In this first work we stick to a basic approach where we study a range of different parameters to find the best reconstruction results.

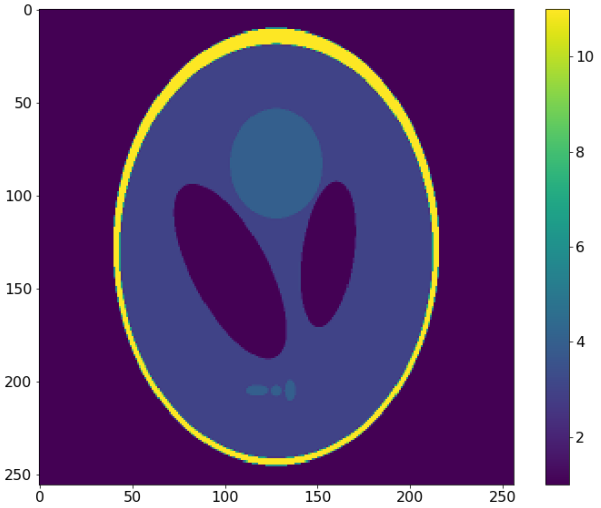


Figure 2.1: Shepp Logan phantom used for experiments.

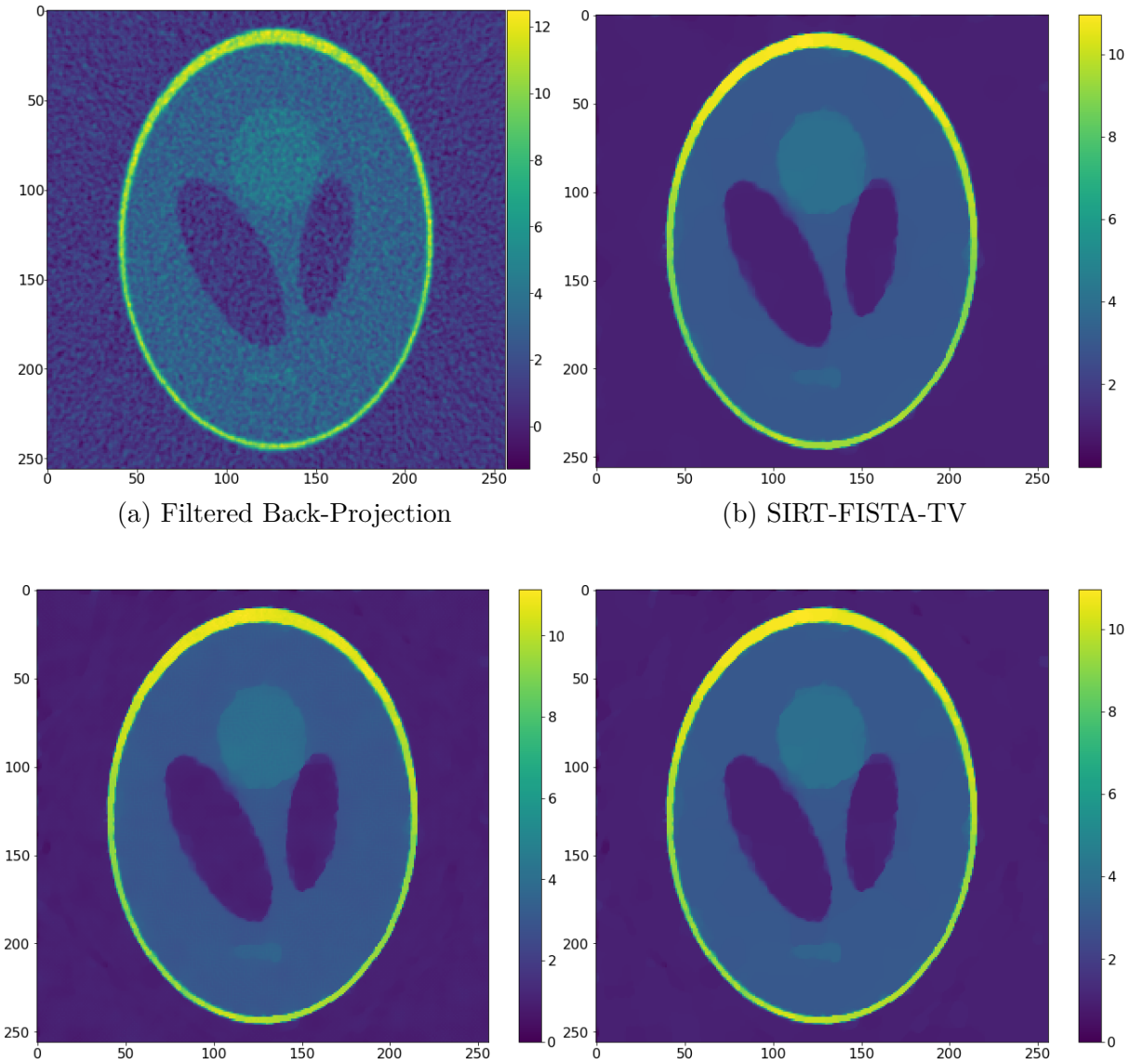
2.2.2 Computational efficiency for simulated data

Reconstructed images of the Shepp-Logan phantom are shown in Figure 2.2 for the Filtered Back-Projection (FBP) algorithm with Hamming filtering and for the three algorithms we study. One dimensional profiles of the reconstructed image (central horizontal line) are displayed in Figure 2.3.

At convergence, the values of the MSE are respectively 0.13 for EM-FISTA-TV and PCP and 0.16 for SIRT-FISTA-TV.

We see in Figure 2.4 that decrease of MSE is faster for EM-FISTA-TV in terms of iterations. However each iteration of SIRT-FISTA-TV and EM-FISTA-TV requires the computation of a projection and a back-projection in the outer loop, followed by the iterative resolution of a denoising problem, while PCP is made of a single loop. This makes each iteration shorter for PCP. The difference in the way these algorithms are built makes the comparison of the computation time difficult; indeed it depends on whether projections and back-projections are fast to compute, which depends on the implementation. Note that the comparison is also dependent on the choice of the number of inner denoising iterations which we fixed once for all here. This is illustrated in Figure 2.5 where both EM-FISTA-TV and PCP algorithms minimize the same cost function $d_{KL}(Af^n, p) + \alpha TV(f^n)$. One can observe the fact that an iteration of PCP is faster than EM-FISTA-TV by comparing the two curves. Here we find that EM-FISTA-TV is slightly faster to converge compared to PCP with our implementation.

The difference between SIRT-FISTA-TV and EM-FISTA-TV in terms of computation time is less dependent on the computational cost of forward and backward projections



(a) Filtered Back-Projection (b) SIRT-FISTA-TV

(c) EM-FISTA-TV (d) PCP

Figure 2.2: Reconstructed images of the Shepp-Logan phantom.

since they rely on the same principle. However it is highly dependent on the number of inner iterations chosen for both. The choice of this number was made so that no difference could be seen in the denoised images when increasing this number. Based on this choice, we found EM-FISTA-TV to be faster than SIRT-FISTA-TV as illustrated in Figure 2.4.

2.2.3 Application to experimental data

In order to assess the extent to which observations on convergence are transcribed for real data, we perform the reconstruction of a three-dimensional sample of CoOCNT observed in transmission electron microscopy. The data were acquired at IPCMS laboratory in Strasbourg.

The mixed Poisson-Gaussian noise model is adapted to this imaging modality [Öktem,

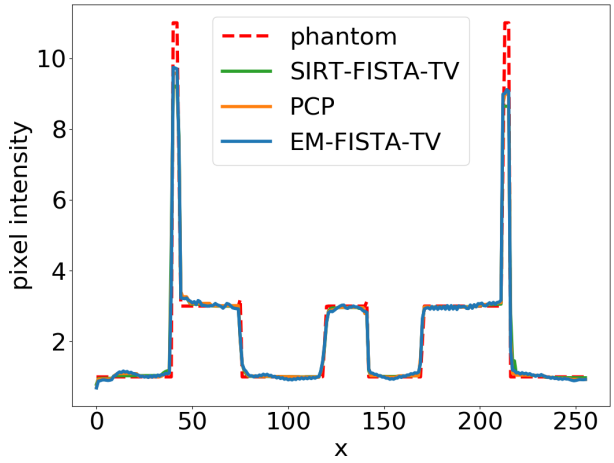


Figure 2.3: Profiles extracted from the reconstructed and phantom images.

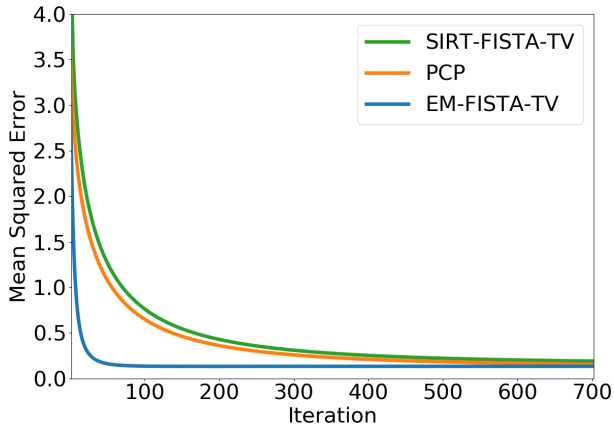


Figure 2.4: Evolution of MSE with respect to the number of reconstruction iterations for the three considered algorithms.

2008]. For these experiments, faceted cobalt–cobalt oxide based nanoparticles (NPs) with high density and narrow size distribution (50 $\pm$ 5 nm) were selectively cast inside the channels of multi-walled carbon nanotubes (CNTs) through the controlled thermal decomposition of cobalt stearate in the presence of oleic acid as surfactant. A number of 59 projections of the sample, ranging from -70 to 75 degrees were acquired. One example of projection is shown in Figure 2.6a. We reconstructed a 3D-volume with dimensions 512^3 pixels with regularization parameter $\alpha = 0.8$. Figure 2.6b shows a slice of the reconstructed volume. Results for SIRT-FISTA-TV and PCP are visually similar.

We can also observe in Figure 2.7 that convergence is obtained faster for EM-FISTA-TV on the real TEM data.

2.2.4 Conclusions for a fixed noise configuration

For the Poisson-Gaussian noise considered in our simulations as well as for TEM real data, results tend to indicate that convergence is slower for SIRT-FISTA-TV compared to PCP and EM-FISTA-TV algorithms. For the considered noise, the first approach gives inferior reconstruction results in our simulations in terms of MSE even after numerical

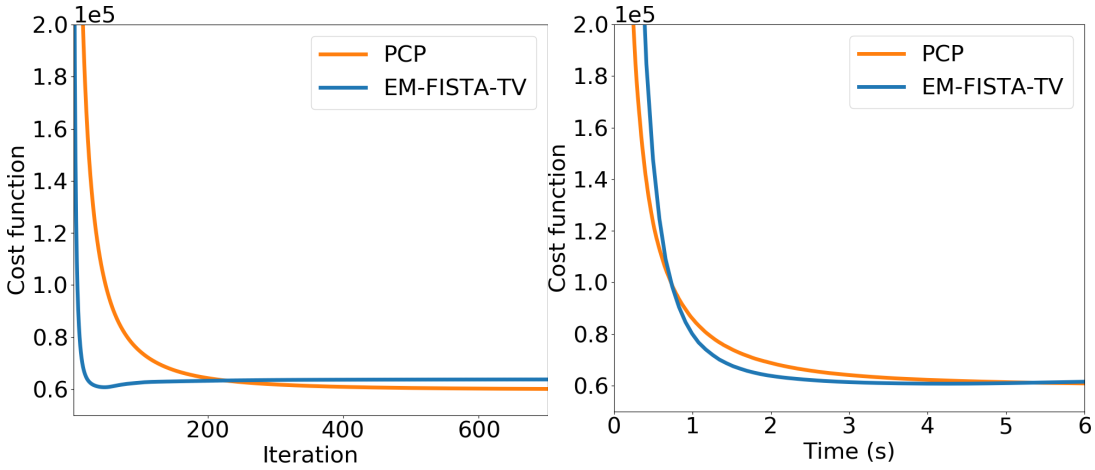


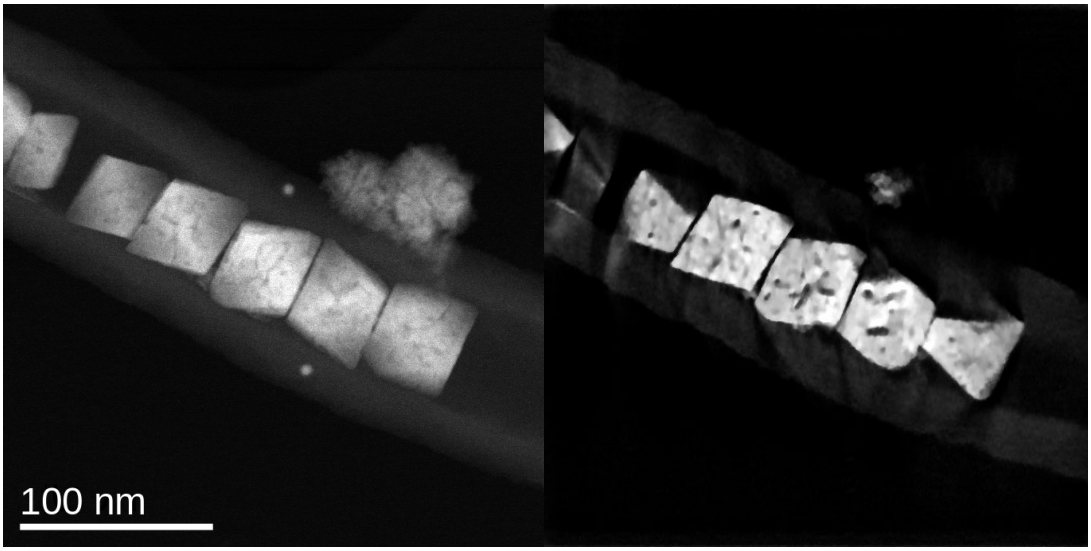
Figure 2.5: Evolution of the cost function $d_{KL}(Af^n, p) + \alpha TV(f^n)$ with (a) the number of iterations (b) time. Due to the FISTA acceleration, EM-FISTA-TV is not always decreasing. Reconstructions are performed on a Quadro P2000 GPU card.

convergence. This statement however needs to be clearly balanced since the chosen configuration corresponds to a dominant Poisson noise compared to Gaussian noise - we will see in what follows how to compare both contributions.

In our tests EM-FISTA-TV requires less iterations than PCP to reach the solution. Nevertheless for a fair comparison, the structure of these two algorithms need to be taken into account. Indeed, the first one has an internal loop that aims at denoising the MLEM-reconstructed image at each iteration. For the second one, the whole reconstruction process, TV denoising included, is performed within a single loop. In that sense, it is interesting to check convergence as a function of total computation time. In these particular tests, EM-FISTA-TV is the faster approach. However, these results are strongly dependent on several factors and may change from one application to another. First, the number of iterations in the denoising part is chosen to insure convergence in the internal loop, though there is no indication that less iterations inside this loop could lead to a similar final solution - and thus in a smaller computation time. Second, the results depend on the implementation and will not be the same if e.g CPU is used instead of GPU for denoising. Finally, the computational results depend on the ratio between the cost of one projection/back-projection pair and the cost of the denoising algorithm.

Another key point concerns memory footprint. We found that memory requirements are similar for both EM-FISTA-TV and PCP that need the equivalent of 11 copies of the initial 3D-volume in our implementation. However, if FISTA acceleration is not performed in the denoising loop of EM-FISTA-TV, it requires only 8 copies of the size of the volume, so one could choose to slow down convergence in order to reduce memory requirements.

In this preliminary work we have compared several iterative algorithms based on TV regularization for tomographic reconstruction with a mixture of Poisson and Gaussian noise. Results are largely dependent on the ratio between Poisson and Gaussian noise, so we will perform in what follows a similar study when varying this ratio.



(a) A projection obtained with TEM from the studied CoOCNTs sample

(b) Slice of the reconstructed image of CoOCNTs with EM-FISTA-TV algorithm

Figure 2.6: Experiments on real TEM acquisition data of a CoOCNTs sample[1].

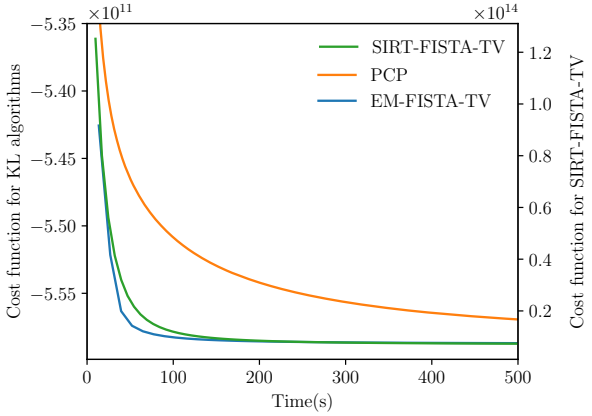


Figure 2.7: Cost function decay with respect to the algorithm running time for reconstructions on real TEM data. The cost function of PCP and EM-FISTA-TV is different than the one for SIRT-FISTA-TV, thus the latter has been represented on a different scale.

2.2.5 Experiments with varying Poisson to Gaussian noise ratio

In this part we consider simulated data that are similar to the previous experiments. The difference is that the ratio between Gaussian and Poisson noise is variable. The ratio that we consider is the following :

$$r = \frac{\sigma_p}{\sigma_g} \tag{2.26}$$

[1]Data are the courtesy of Ovidiu Ersen from ICPMS laboratory, Strasbourg.

r	Best algorithm
≤ 0.25	SIRT-FISTA-TV
$\in [0.25, 0.70]$	variable
≥ 0.70	EM-FISTA-TV (PCP close)

Table 2.1: Algorithm with the best perfomance in terms of MSE depending on the ratio between Poisson and Gaussian noise intensities. Here we assume that Gaussian noise is under a threshold value $\sigma_g/\bar{P} \leq 11.5\%$ where $\bar{P}$ is the mean value of the projections pixel intensities.

where $\sigma_p = \frac{1}{n}\sum_i \sqrt{p_i}$ is the mean value of the squared roots of each value in the projections - this corresponds to the mean absolute fluctuation as the variance for Poisson noise is equal to the mean value -, and $\sigma_g = \sigma \max(p)$ since this is the manual choice that we made for Gaussian noise's standard deviation. This is representative of the ratio between the fluctuations in the data due to Poisson noise and the ones due to Gaussian noise.

We make this ratio variable by scaling the values of the Shepp-Logan phantom. We consider a maximum pixel value between 4 and 100 in our experiments. We also add to each detector pixel p_i a realization of a random Gaussian variable with zero mean and the standard deviation $\sigma \max(p)$ where σ varies between 0.01 and 0.1.

To evaluate the performance of each algorithm we compute 2 metrics: MSE and time until convergence. The convergence is considered when the slope of the MSE curve is below 1% of its final value. Once again, computation time is dependent on the implementation and we only are able to give results for the one we performed. We consider the number of iterations until convergence as an irrelevant criteria here for the reasons mentioned before.

We compute those 2 metrics on 10 different values for σ and 10 different pixel maximum values for the Shepp Logan phantom, which gives 100 different combinations/values of r. It is also to be noted that regarding the number of data that these experiments represent, we fixed the regularization parameter for each algorithm. We took $\alpha = 0.8$ for EM-TV and PCP and $\alpha = 0.2$ for SIRT-TV, as it is a common choice for these algorithms.

2.2.6 Results with varying Poisson to Gaussian noise ratio

Table 2.1 shows the algorithm that has the best MSE depending on the ratio r, for cases where Gaussian noise is not too high, i.e $\sigma_g/\bar{P} \leq 11.5\%$ where $\bar{P}$ is the mean value of the projections pixel intensities.

We also represent a scatterplot in Figure 2.8 that shows the most efficient algorithm in terms of MSE depending on the considered Poisson and Gaussian noises. The values on the x-axis represent the Gaussian noise as $\sigma_g/\bar{P}$ and the y-axis corresponds to the Poisson noise as $\sigma_p/\bar{P}$

We observe that when the Gaussian noise is high enough, i.e $\frac{\sigma_g}{\bar{P}} \geq 11.5\%$, SIRT-FISTA-TV is the most performant algorithm regardless of the Poisson noise. For a lower amount of Gaussian noise, the algorithm with the smallest MSE depends on the ratio r. When $r \leq 0.25$, SIRT-FISTA-TV has a lower MSE compared to the two other algorithms.

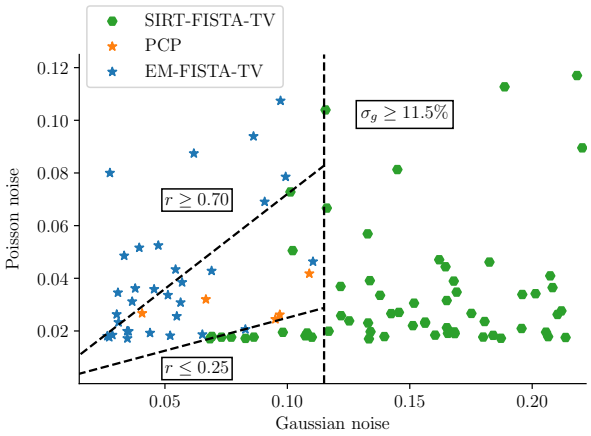


Figure 2.8: Algorithm with the lowest MSE depending on the intensity of both Poisson and Gaussian noise.

For $r \in [0.25, 0.70]$, there is no significant difference as the MSE of all three algorithms is very close. When $r \geq 0.70$, EM-TV and PCP have better MSE compared to SIRT-TV, with the best values for EM-TV though the difference with PCP is very slight.

These results are adequate to the expectations as SIRT-TV is adapted to Gaussian noise and EM-TV and PCP are adapted to Poisson noise. The intermediate zone does not allow to get one algorithm that outperforms the others; thus for $r \in [0.25, 0.70]$, we analyzed the time of convergence to see if this could be a criteria of choice for our three methods.

We observed that in this range for r, EM-TV was faster in 28% of the cases; PCP in 19% of the cases and SIRT-TV was faster in 53% of the experiments. This is a slight advantage for SIRT-TV even if it might not be significative.

For $r \geq 0.70$ we compared the convergence time for EM-TV and PCP as their MSE is very close in such a case; in 31% of the cases EM-TV was faster, against 69% for PCP.

The only clear pattern that we could observe regarding the computation time until convergence is the following: PCP converges faster than EM-TV for a high Poisson noise, and the contrary holds for lower Poisson noise. This is true no matter what the Gaussian noise is; and it holds until the Poisson noise becomes too high in which case the performance of PCP is affected. This result is illustrated in Figure 2.9 where we consider a Gaussian noise with $\sigma = 2\%$ and two different configurations for the Poisson noise.

Observations and conclusions that we are able to draw here are true for the particular case where we fixed the regularization parameter α to 0.8 or 0.2 as explained. We did so because results seemed visually satisfying; though it is clear that a more sophisticated approach to tune the parameters depending on the noise level would be beneficial.

On the TEM real data, we evaluated the ratio r as follows: for σ_p, we computed $\sigma_p = \frac{1}{n}\sum_i \sqrt{p_i}$ on an area that is affected by Poisson noise, i.e where the intensities are important. For σ_g, we took the standard deviation on a background area. We find the values $\sigma_p/\bar{P} = 5.1\%$ and $\sigma_g/\bar{P} = 7.4\%$, thus corresponding to a ratio $r = 0.69$. Note also that the standard deviation is significantly lower in an empty area compared to a high intensity area which suggests that Poisson noise indeed affects the data.

As a conclusion of this whole study, we find that as expected, SIRT-FISTA-TV is

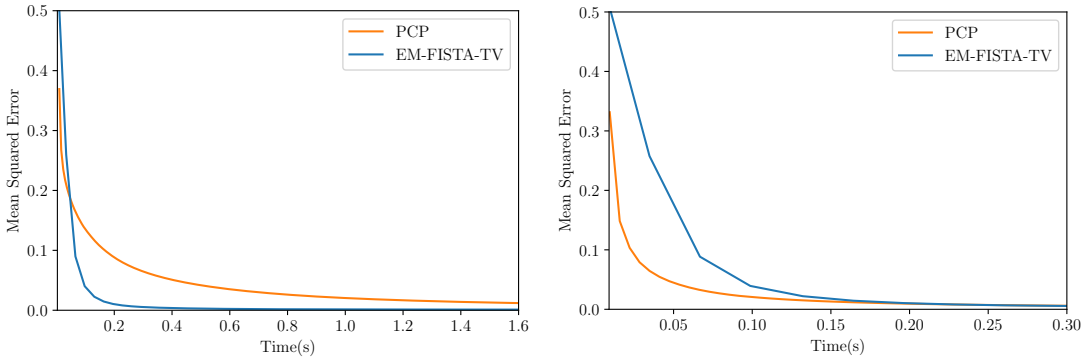


Figure 2.9: Convergence of PCP and EM-FISTA-TV algorithms for low Poisson noise (maximum intensity of 100 in the initial image, left) and high Poisson noise (maximum intensity of 10 in the initial image, right).

more adapted to cases with higher Gaussian noise and PCP and EM-FISTA-TV are more performant when the Poisson noise is dominant. For intermediate values, there is no clear trend to be observed. This is true for a fixed regularization parameter, but results could be different if one decides to give more or less importance to the regularization term. An interesting result is that PCP is faster than EM-FISTA-TV for high Poisson noise and this is the contrary for low Poisson noise. On real TEM data, our experiments show that EM-FISTA-TV is the fastest algorithm to converge, for visual results that are very similar for all of the algorithms. This can let us suppose that the Poisson noise is low in this case. Work on more advanced modelization of the noise could be of interest to be able to decide which algorithm is the most suitable for a particular task. This is even more true when considering the fact that we only evaluated three methods here though there are plenty more. Especially, comparative results might differ when tests are made on different data [Friot-Giroux et al., 2022].

2.3 Limits on iterative methods: towards deep learning

Throughout this chapter, we have evaluated the performance of iterative algorithms and compared them with each other. We found four different points that constitute major drawbacks and are inherent to such methods:

- All of the methods that we considered here assume that the forward model $\boldsymbol{A}$ is perfectly known. As soon as it is no longer the case, the performance of these algorithms can be drastically reduced. Depending on the application, the assumption of the perfect knowledge of the forward model might be valid or not.

- The cost function to use is highly dependent on the nature of the data that one is working with. Even if we performed experiments that might help for the choice of a particular algorithm for a specific task, designing the right cost function along with the most appropriate iterative algorithm among all of the choices is a tedious

task and it seems hard to find an automatic solution to such an issue for clinical practice.

- In our experiments we were able to observe the importance of the choice of the regularization parameter α. In this chapter we kept a basic approach consisting in manually testing different values. Methods for finding an optimal parameter have been proposed [Bertero et al., 2010, Ito et al., 2011]. However in practice such algorithms turned out to be very slow in our tests and convergence could not be always obtained so that results are not shown here.

- As for the practical point of view, a similarity that exists for all iterative methods is the high computational cost, whether it is in terms of memory or time. For this point, it seems that only implementation and/or hardware optimization can be useful to fasten the reconstruction process. As for the image reconstruction method itself, it seems that this represents an inevitable limitation.

All of those points represent, according to us, a relevant justification towards the investigation of deep learning methods for tomographic reconstruction. The rest of this thesis thus focuses on this particular area of research. In what follows we detail such methods so that we are able to have an overview on the extent to which deep learning might be a solution for tackling the aforementioned limitations of traditional reconstruction methods.

Chapter 3

Deep learning for tomographic reconstruction

Following the large success of artificial intelligence - and especially deep learning - that has been observed over the last ten years for computer vision tasks, the possibility to benefit from those advances has been investigated in many different research areas. This is inevitably the case for tomographic reconstruction, for which deep learning methods have shown very promising results around 2017, with FBPConvNet [Jin et al., 2017] - that will be discussed in this chapter - as one of the main reference methods.

Deep learning based algorithms represent a paradigm shift compared to iterative methods. One of the main differences is that parameter tuning is done once for all, before the actual reconstruction task. This alleviates one of the major constraints implied by traditional reconstruction techniques. Another key benefit that could be brought by such methods in clinical routine is the computation time, which is in most cases way shorter compared to the methods discussed in the previous chapter. This opens the way for breakthroughs in applications that require ultra-fast image reconstruction. As we also saw in the previous chapter that knowledge of the noise corrupting the data is of significant importance for iterative reconstruction, this is generally no longer the case for deep learning based methods since the noise statistics can be - implicitly - learned from the data on which the algorithm is trained. An interpretation that can be made for the success of these methods is that the learning process make neural networks able to learn *a priori* information on the ground-truth image to reconstruct.

Deep learning based methods look like a step forward for the performance of reconstruction techniques since they have the ability to make use of available data, which is not the case for traditional algorithms. In clinical practice however, the use of such methods has been very limited so far. Better understanding of the methods and solid proofs for robustness need to be achieved in order to become the standard of use in most clinical cases, and this is the reason why we will focus on this area of research from now on.

There is probably an infinite number of possibilities to design a neural network for the reconstruction task. In the literature, large reviews of deep learning based methods for inverse problems [Arridge et al., 2019] or PET imaging in particular [Reader et al., 2020] have been proposed. This chapter aims at splitting all of those methods into different categories. As it represents a very active research field, it seems impossible to cover all of the existing algorithms but we aim here to give an overview of the theory and applications

that are related to each of the considered category. Especially we will see that some of the existing methods are fully based on neural networks, while others try to combine analytical or iterative algorithms with deep learning.

In this chapter we start by giving an overview of what is deep learning in Section 3.1, in order to handle the main concepts that will be used in this thesis. The rest of this chapter is then divided as follows: first we cover the most common methods relying on supervised learning in 3.2. We then put the emphasis on generative adversarial networks in Section 3.3 since it has been a breakthrough in the computer vision field since 2014; we detail some of the applications they might have for image reconstruction. Section 3.4 is on the particular case of using neural networks to learn a regularization term, and Section 3.5 covers the case of self-supervised methods.

3.1 Deep learning for computer vision

Deep learning is part of the broader family of machine learning methods, that aim at solving problems by learning on data. In this section we give a brief overview on the main principles of deep learning. The aim is obviously not to cover all the aspects, but rather to explain the notions that will be used and discussed in the rest of this thesis. For a comprehensive book on deep learning theory fundamentals, one might refer to [Goodfellow et al., 2016]; for practical aspects on implementation with *Tensorflow*, the book [Géron, 2019] is a valuable asset.

3.1.1 Machine learning

As mentioned, deep learning is only a part of machine learning. Machine learning is defined in [Géron, 2019] as "the science (and art) of programming computers so they can *learn* from data". It is an effective tool for automatizing tasks, finding patterns in some data that humans could not find in a reasonable amount of time for instance, or predicting outcomes based on past experience.

Most of machine learning techniques rely on supervised learning. In this case a *training* dataset is used to e.g learn to make predictions. For instance, if a task consists in assessing whether a tumor is present in some CT image, the training dataset consists images that are paired with a binary output: 1 if a tumor is indeed present, 0 otherwise. The role of the machine learning model is to, based on the *experience* acquired from the data, be able to predict whether a CT image that has never been given to the model contains a tumor or not.

Mathematically, this corresponds to having a predictor h_θ that depends on parameters θ, that is trained to correctly evaluate a new input x as $h_\theta(x)$. This is done by giving paired data (x_i, y_i) where y_i is the true output (or *label*) corresponding to the input x_i, and finding the optimal parameters θ^* that minimize some error between the predicted outputs $h_\theta(x_i)$ and the ground-truths y_i. Considering the previous examples, $x_i \in \mathbb{R}^n$ correspond to CT images and $y_i \in \{0,1\}$ are the binary outputs assessing whether a tumor is present or not. The efficiency of a machine learning model can be assessed by evaluating the accuracy of the prediction $h_\theta(x)$ with x that has never been "seen" by the model. An other example from [Géron, 2019] is given in Figure 3.1 to illustrate how a

machine learning model can be trained for spam detection/classification.

The supervised learning methodology can be summarized as follows:

1. Training: the predictor h_θ is trained with paired (x_i, y_i) in order to minimize the error between predictions $h_\theta(x_i)$ and ground-truth y_i.

2. Validation: the predictor h_θ depends on parameters θ but also on other parameters that are related to the way h_θ is built (linear regressor, artificial neural network, ...) and to the way optimization on θ is performed. These are referred to as hyper-parameters (HP). The choice of these HP need to be made on a dataset different than the training set to prevent over-fitting the data, i.e to avoid situations where the HP are only adapted to the training set but not to new data. Different strategies for tuning the HP are detailed in [Géron, 2019].

3. Testing: once the HP are chosen on the validation set and the optimal parameters θ^* found thanks to the training, the quality of the predictor h_{θ^*} can be evaluated before being deployed to production. Once again, this needs to be performed on data that are not part of either the training or the validation set in order to be fair: the data on which the predictor will then be actually used are data that are not part of those datasets. With testing, one is able to evaluate the ability of the predictor to generalize to new cases.

Note that another major category of machine methods is based on unsupervised learning. This time, the training dataset does not contain ground-truth y_i. Main applications for unsupervised learning include clustering, anomaly or novelty detection, dimensionality reduction, etc. We will see in this chapter that generative models, used e.g to generate fake images, are based on unsupervised learning.

When only a small amount of the training data is labeled - the rest being unlabeled -, this refers to the notion of semi-supervised learning. We will also discuss in Section 3.5 the notion of self-supervised learning.

3.1.2 Deep learning vs machine learning

The major thing that makes a machine learning algorithm a "deep" learning one is the form taken by the predictor h_θ: it corresponds to a so-called *Artificial Neural Network (ANN)*. An ANN is a Machine Learning model inspired by the networks of biological

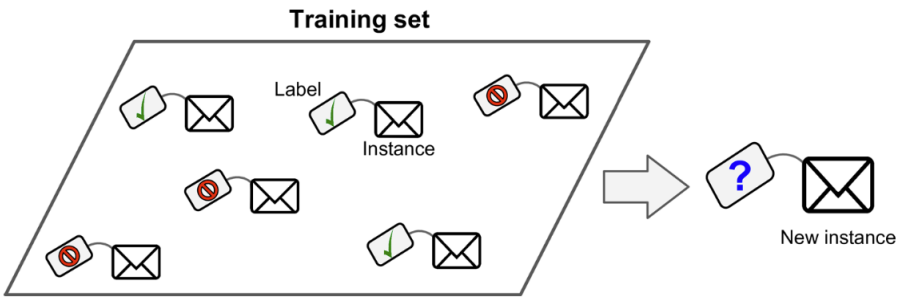


Figure 3.1: Example of supervised learning for spam detection [Géron, 2019]

.

neurons found in humans' brain. The architecture of ANN is such that they are ideal to tackle machine learning problems involving large datasets and complex tasks, such as image classification, speech recognition, or beating the world champion at the game of Go.

The element at the core basis of neural networks is the neuron, or threshold logic unit (TLU). A scheme of such a unit is given in Figure 3.2. An artificial neuron takes some numbers x_i as input, computes a weighted sum $z = x^T w$ and the output of the neuron is $a(z)$ where a is an activation function. This activation can be a sigmoid function for instance or a Rectified Linear Unit function (ReLU) defined as $ReLU(z) = \max(0, z)$.

A neural network is therefore a predictor h_θ that is a combination of neurons that are organized in different layers as represented in Figure 3.3 and 3.4. For instance the original Perceptron proposed in [Rosenblatt, 1958] was composed of a single layer of TLUs, with each TLU connected to all the inputs as shown in Figure 3.5. In this case, the parameters θ to optimize in order to find the optimal predictor correspond to the weights w. Note that a neural network is not necessarily composed of stacked fully-connected layers as in Figure 3.3, as we will see with Convolutional Neural Networks.

Now that we have an overview of how a neural network is built, a key component of

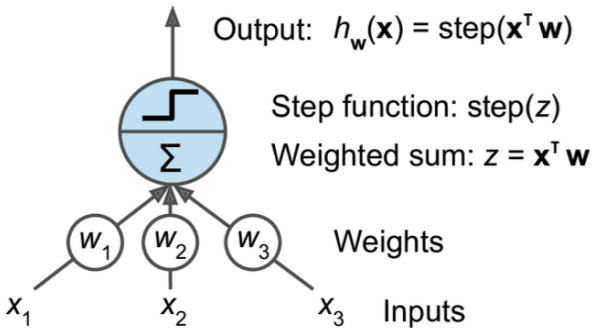


Figure 3.2: Illustration of a threshold logic unit or artificial neuron [Géron, 2019].

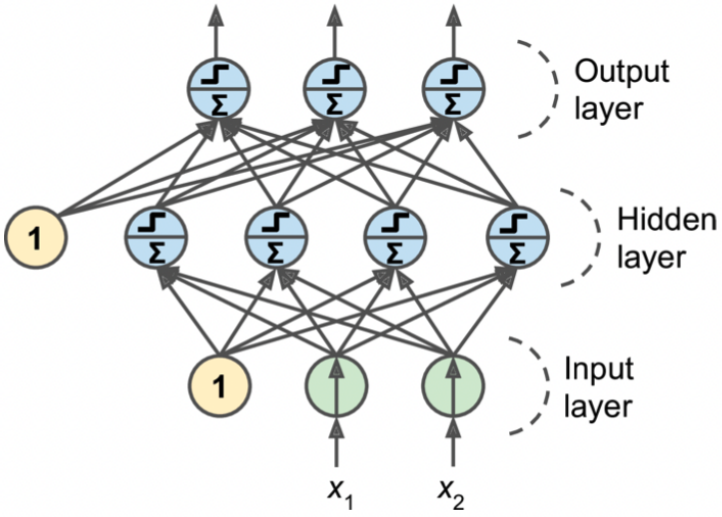


Figure 3.3: Multilayer Perceptron with two inputs, one hidden layer of four neurons, and three output neurons [Géron, 2019].

learning based methods is the method to find the optimal weights w, or more generally the optimal parameters θ of h_θ. The reference method was published in [Rumelhart et al., 1985] as the *backpropagation* algorithm. The method is based on a gradient descent to optimize the parameters θ. It makes use of the automatic differentiation technique named *reverse-mode autodiff* to compute the gradients of a *loss function* w.r.t θ. The *loss function* computes the error between the predictions $h_\theta(x_i)$ and the ground-truths y_i. The training process aims at finding out how the parameters θ - or weights w - should be tweaked in order to reduce the error until reaching its minimum. Note that we will use the terminology of weights in what follows to mention the parameters to tweak for the neural networks.

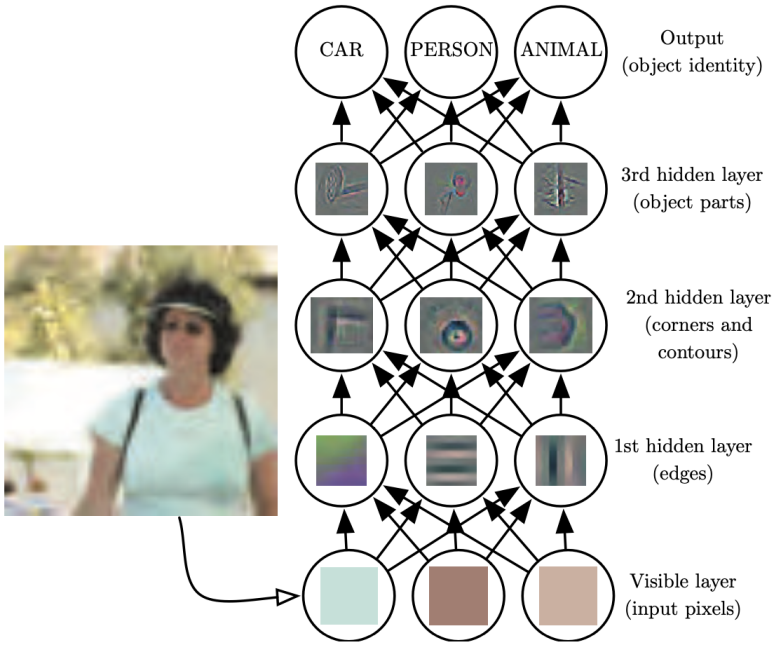


Figure 3.4: Illustration of a deep learning model [Goodfellow et al., 2016].

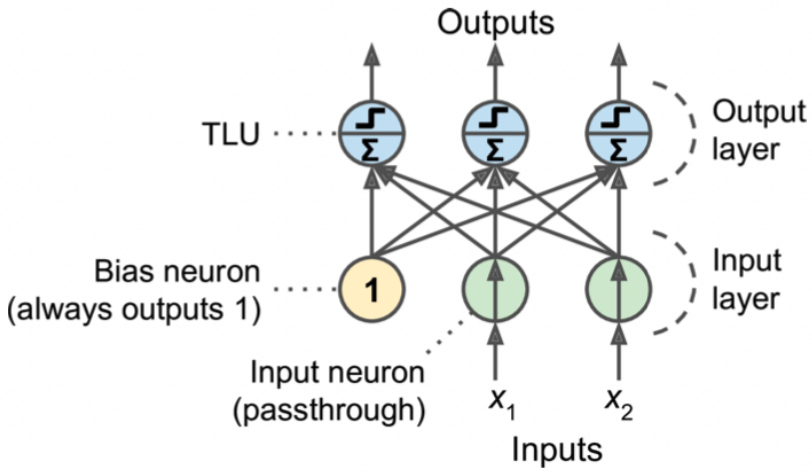


Figure 3.5: Architecture of a Perceptron with two input neurons, one bias neuron, and three output neurons [Géron, 2019].

The backpropagation algorithm is summarized in [Géron, 2019] as follows: for each training instance - e.g paired (x_i, y_i) -, the algorithm first makes a prediction (forward pass) and measures the error, then goes through each layer in reverse to measure the error contribution from each connection (reverse pass), and finally tweaks the connection weights to reduce the error (Gradient Descent step).

The rule for updating the weights w is given by the optimizer. A simple gradient descent can be a relevant choice; in this thesis we will mostly use Adam optimizer [Kingma and Ba, 2014] that is an optimized version of the gradient descend which has proven to be generally efficient to train neural networks.

3.1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a particular type of ANN that have shown excellent results for computer vision tasks and others. Just like the ANN is inspired from the structure of the brain, CNNs have been inspired by the structure of the visual cortex. In the visual cortex, biological neurons especially respond to specific patterns in small regions of the visual field called receptive fields; as the visual signal makes its way through consecutive brain modules, neurons respond to more complex patterns in larger receptive fields [Géron, 2019]. A breakthrough for CNNs was presented in [LeCun et al., 1998] with the famous *LeNet-5* architecture.

The building block of a CNN is the convolutional layer, which is represented in Figure 3.6. Contrary to the fully-connected layers presented before, neurons in one convolutional layer are not connected to every single inputs x_i. For image inputs, neurons are only connected to pixels in their receptive fields, see Figure 3.6. Stacked convolutional layers proceed in the same way: neurons in some layer l are only connected to few neurons in the layer $l-1$. This architecture allows the network to learn low-level features in the first layers, then higher-level features as the number of layers increases. The main reason for the success of CNNs in computer vision tasks is the fact that this hierarchical structure is similar for natural images.

Weights for CNNs can be represented as small images that have the size of the receptive field: they are called *filters*. A layer of neurons using the same filter outputs a

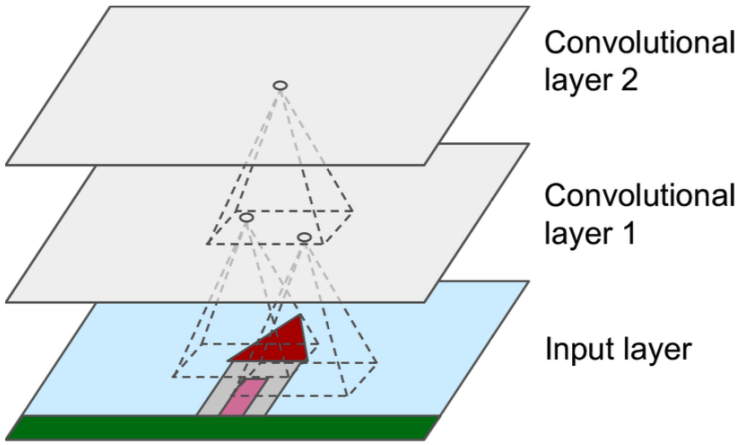


Figure 3.6: Convolutional Neural Networks and receptive fields [Géron, 2019].

so-called *feature map*. Each layer of the CNN can have several feature maps, depending on the number of filters that is considered. A scheme of the core structure of CNNs with feature maps is given in Figure 3.7.

A description of how to compute outputs of neurons is given in [Géron, 2019]: let's consider a neuron located in row i, column j of the feature map k in some convolutional layer l. It is connected to the outputs of the neurons in the previous layer $l-1$, located in rows $i \times s_h$ to $i \times s_h + f_h - 1$ and columns $j \times s_w$ to $j \times s_w + f_w - 1$ where f_h and f_w are the height and width of the receptive field. The output $z_{i,j,k}$ of this neuron can be computed as

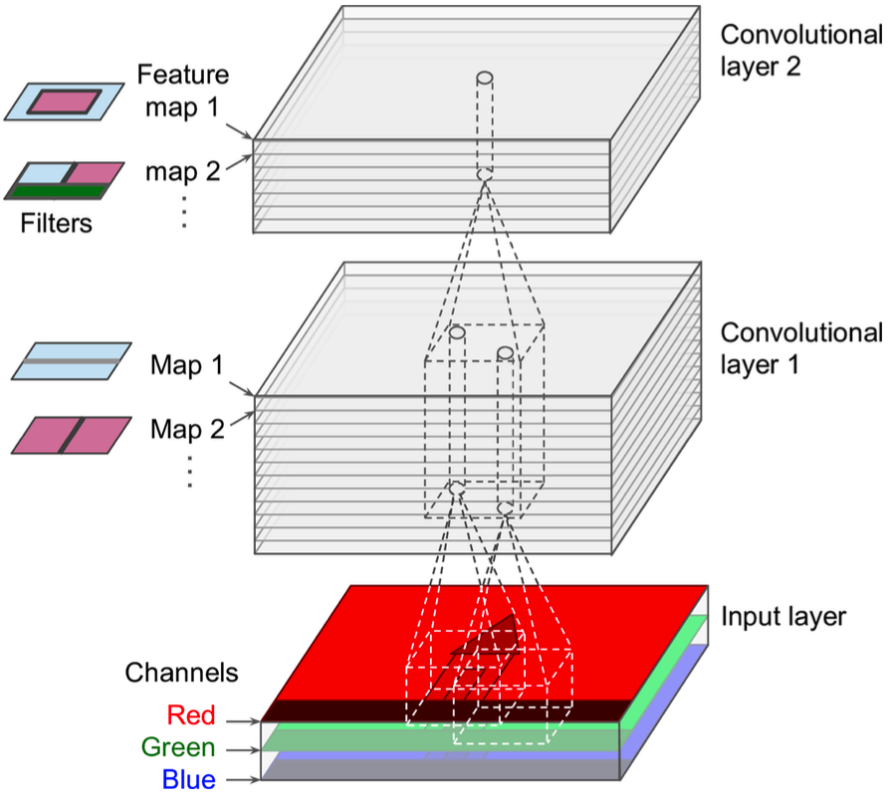


Figure 3.7: Convolutional layers with multiple feature maps. Images here are with three color channels [Géron, 2019].

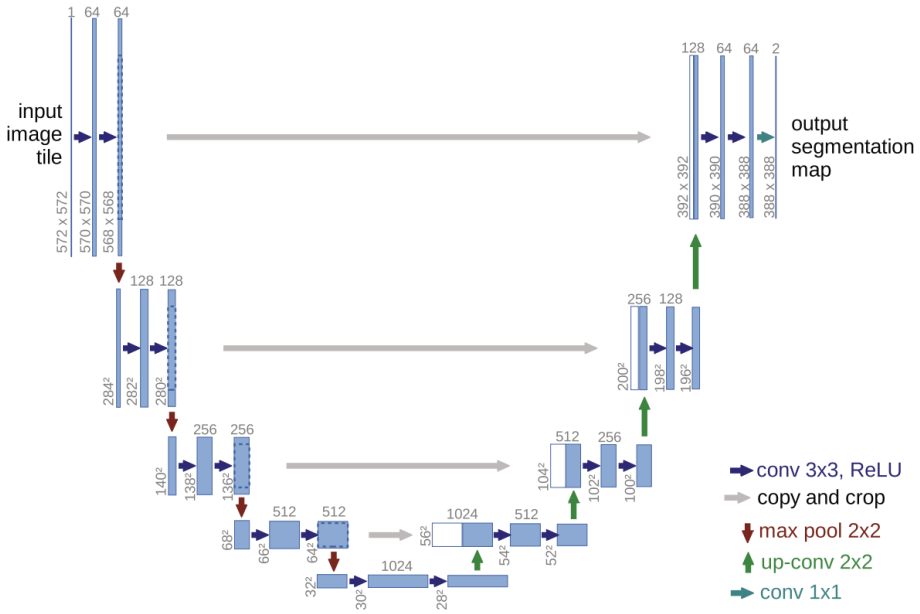


Figure 3.8: Original architecture of the UNET [Ronneberger et al., 2015].

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i+u,j+v,k'} w_{u,v,k',k} \tag{3.1}$$

where:

- $x_{i+u,j+v,k'}$ is the output of the neuron located in layer $l-1$, row $i+u$, column $j+v$ and feature map k'.
- b_k is the bias term for feature map k in layer l.
- $w_{u,v,k',k}$ is the connection weight between any neuron in feature map k of the layer l and its input located at row u column b and feature map k'.

Note that we do not take into account the notion of *stride* here for simplicity. More details on the structure of CNNs can be found in [Géron, 2019].

There has been a lot of research towards the design of efficient CNN architectures. For tomographic reconstruction, an architecture that is largely used in many methods in the U-NET architecture. It was originally designed for biomedical image segmentation in [Ronneberger et al., 2015], but the architecture was adapted to image reconstruction in many works. A scheme of the original architecture is represented in Figure 3.8. One can observe that the structure consists of a contracting path and an expanding path with skip connections. This structure allows learning low-level features while preserving high-level information from the input data very efficiently.

3.2 Supervised methods for image reconstruction

In this section we present deep learning based methods that require a training dataset consisting of paired input and ground-truth data.

There are two main possibilities to acquire such pairs in practice, the first one being to numerically simulate images, and obtain acquisition data through simulation as well. The ground truth is then the simulated image, and the input can be the raw simulated acquisition data or some transformations applied on these. This configuration offers the advantage to have full flexibility on the considered images. This advantage comes with the cost that a software for the acquisition simulation is required; another drawback is that one needs to assess the good transcription of networks trained on simulations and applied on real acquisition data. We will use this method to get a training dataset in Chapter 6.

The second possibility is to have patients "real" data. In this case the ground truth is a reconstruction - obtained with an iterative algorithm for instance - that is considered as satisfying, and the input of the network is the actual raw data, or some transformation on it once again. This allows getting networks that can directly be applied to real acquisition data once they are trained, with potentially more robustness guarantee compared to networks trained on simulations. Obtaining ground truth data can however be tedious especially because of the drawbacks of analytical and iterative methods. A common application of such methods is the low-dose to high-dose task that consists in using a neural network in order to get a reconstruction similar to what could be obtained in a high-dose setting, with low-dose acquisition data. We will detail this process in the next chapter as it corresponds to our application with micro-CT data.

3.2.1 Reconstruction post-processing

Among the advantages mentioned for the neural networks, the computation time might be one of the most meaningful from a clinical point of view. Compared to the other methods presented in this chapter, enhancing the quality of a reconstruction obtained from an analytical[1] method with neural networks is the fastest solution.

In addition to the computation time, another advantage of such methods is that they make use of convolutional neural networks in the image domain, which is specifically the scope for which CNNs have proven to be efficient in the computer vision realm.

The first methods to appear for CT reconstruction post-processing were FBPConvNet [Jin et al., 2017] and RED-CNN [Chen et al., 2017]. The scheme of the general reconstruction pipeline is given in Figure 3.9. Both of them consist in feeding a convolutional neural network with the FBP of some sinogram data. After being trained to map this FBP to a corresponding ground-truth, the neural network has the ability to enhance the quality of the initial FBP reconstruction in limited amount of time. Both methods only differ in the structure of the neural network; a U-NET for FBPConvNET and a residual encoder-decoder for RED-CNN.

The same methodology is used in [da Costa-Luis and Reader, 2017] for PET imaging, with a deep convolutional network that takes as input a pair of PET reconstruction and

[1]Note that those methods could be applied on reconstructions obtained from iterative algorithms as well but this annihilates the benefits of having a fast reconstruction.

T1 MR image; the PET reconstruction being obtained with MLEM algorithm. Results here suggest that such a network has the possibility to reduce the noise, get sharper edges, improve resolution and partially remove artefacts.

Those methods work on 2D slices; the main reason is that it raises less constraints in terms of computational resources. Working on slices might however lead to the loss of spatial resolution in the third dimension; also neural networks could potentially benefit from information of adjacent slices. This idea is exploited in [Xu et al., 2017] for PET imaging where the network has a multi-slice input that consists of the slice of interest concatenated with the adjacent slices in order to retrieve the 2D slice of interest as an output.

Better performance can be expected when working in 3D since it can help capture spatial information across slices; instead of working with 2D convolutional filters in the networks, one must then use 3D filters. In the same way, the input and ground truth data correspond to 3D volume. This has been addressed in [Yang et al., 2017, Shan et al., 2018] where enhanced quality was observed in the reconstructions. More recently the network FastPET [Whiteley et al., 2020] made use of such a 3D network for post-processing images for time of flight PET imaging.

Working with 3D data however implies a significant increase in terms of computational resources; working on patches rather than on the whole volume is one of the solutions to deal with the issue, as well with reducing the batch size during training for instance. It is to be noted that it is not a trivial task to compare the performance of 2D and 3D methods since the 3D networks have more parameters compared to 2D ones for a fixed architecture; it is hard to isolate the impact of having more training parameters vs considering spatial information to measure the improvements brought by a 3D method.

A common question that can be raised for reconstruction post-processing method is the loss of information on the raw data that is caused by the initial reconstruction. It is hard to evaluate the ability of neural networks to retrieve the solution when a certain amount of information from the measurements is lost during the first analytical/iterative reconstruction step.

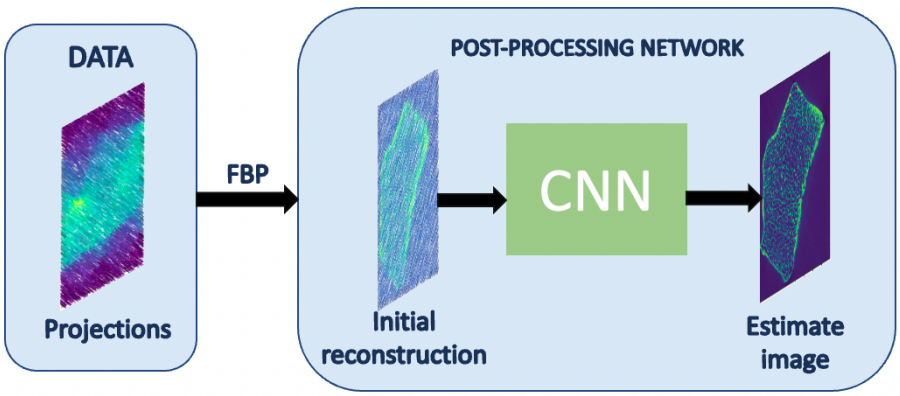


Figure 3.9: Post-processing networks reconstruction pipeline. Note that in general the initial reconstruction can be obtained with FBP but also with any other analytical or iterative method.

3.2.2 Direct reconstruction methods

Instead of getting a first approximation of the image and then reduce noise or remove artefacts from it, one can leave the entire reconstruction task to a neural network. The network can then perform sinogram correction, image domain mapping and image quality enhancement within a single neural architecture. The general principle for these methods is illustrated in 3.10.

This is the idea in [Li et al., 2019] where such a network is used to obtain a reconstruction from sinogram data under different acquisition conditions. This approach, illustrated for CT in the article, allows one to tackle different problems such as limited angle view, data undersampling or even specific artefacts linked to each modality, within a single framework. The downside here is that the architecture implies a large number of parameters for the network; to prevent overfitting, it is then highly desirable to have a sufficient amount of data for training such networks.

In [Liang et al., 2018], an interesting study is performed to compare direct reconstruction frameworks with reconstruction post-processing. The study also includes a comparison with methods that correct the projections with a neural network and take an anaytical reconstruction such as the FBP afterwards.

Results show that the projection estimation network removes streaking artefacts by estimating the missing data, though the final images are still blurry with some details that are lost. The image-domain UNET produces better outputs visually, though some artefacts might be created. The direct reconstruction network achieves more accurate reconstructions for the considered metrics e.g RMSE, with streaking artefacts removed and details well preserved.

Once again, we put the emphasis on the difficulty to compare methods that are intrinsically different; even if the direct reconstruction framework shows better performance here, it has a number of parameters significantly higher compared to the other tested methods. In this sense it is not clear whether results hold for a limited number of training data, or if a post-reconstruction UNET-like network would perform better if its number of parameters was higher.

Another point of attention to be raised for such methods is the complete absence of model knowledge in the neural network. For reconstruction post-processing networks, the forward model - which represents the acquisition settings - can be implicitly known through the reconstruction operator, even if some information might be lost as explained

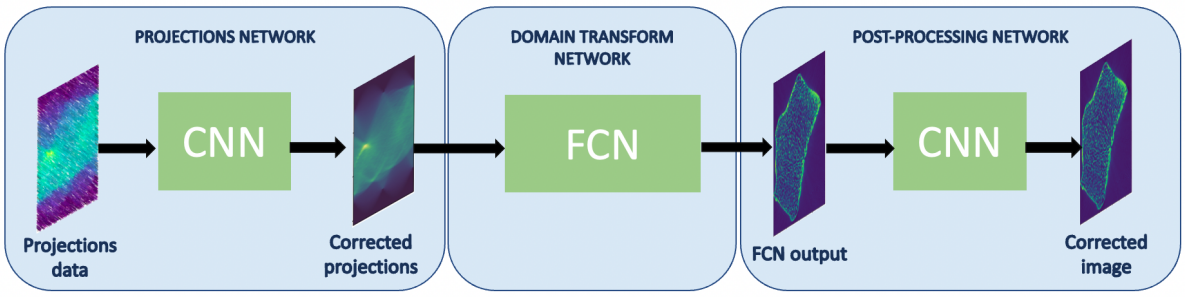


Figure 3.10: Direct methods' reconstruction pipeline. FCN is a fully-connected network as in [Li et al., 2019].

earlier. Without any information on the forward model here, the question of robustness can be raised. It seems that such methods are mostly dependent on the amount of training data available, and medical imaging is a field where data is particularly hard to obtain.

3.2.3 Unrolling iterative methods into neural networks

Unrolled iterative schemes come as a relevant answer to the issue raised by the lack of model knowledge in the design of direct reconstruction frameworks as well as potential loss of non-retrievable information for post-reconstruction neural networks. The general idea is that such methods take the structure of traditional iterative algorithms such as the ones derived in Chapter 2, but the parameters associated to these algorithms are learned beforehand thanks to paired input/ground-truth data. The common advantage of these methods over the two others derived so far is the need for fewer training data; this can be particularly relevant for medical applications.

One of the first works dealing with such methods was proposed in [Adler and Öktem, 2017], where the concept is derived for the general case of inverse problems. Let's consider the example of gradient descent. The usual update at iteration $k+1$ for the image f to retrieve is

$$f^{k+1} = f^k - w^k \nabla_f J(f^k, p) \tag{3.2}$$

with $J(f^k, p)$ a cost function depending on the current estimate f^k of the solution f^* and measurements data p, and w^k is an update parameter that is either fixed or adapted according to some rule. The idea of using unrolled iterative schemes here is to replace the previous update term by a neural network, which gives

$$f^{k+1} = f^k - \Gamma^k_{\theta_k}(\nabla_f J(f^k, p)) \tag{3.3}$$

with $\Gamma^k_{\theta_k}$ a CNN parametrized by θ_k. In this sense there are N convolutional neural networks in the overall architecture if N is the number of iterations of the iterative algorithms. In the literature the initial estimate f_0 can be either a constant image or some initial reconstruction of the data p like the FBP.

The question is: how to train those CNNs $\Gamma^k_{\theta_k}$? In practice, one has access to pairs of data $(p_i, f_i^*)_{i=1..n}$ where f_i^* is the considered ground truth at the origin of data p_i. In that sense the network takes as input data p_i, outputs an estimate f_i^N and all of the blocks $\Gamma^k_{\theta_k}$ are trained simultaneously according to a loss function which can be for instance the mean squared error between the ground truth and the predicted output as

$$\ell_\theta(f^N, f^*) = \frac{1}{n}\sum_{i=0}^{n} ||f_i^N(\theta) - f_i^*||_2^2, \tag{3.4}$$

with $\theta = (\theta_1, ..., \theta_N)$. More generally, most iterative algorithms can be replaced by the following iterative scheme using deep neural networks (DNNs)

$$\begin{cases} f^0 = f_0 \text{ chosen} \\ f^{k+1} = \Gamma^k_{\theta_k}(f^k, f^k_m, \nabla_f J(f^k, p)) \end{cases} \tag{3.5}$$

with f^k_m computed from f^k, for example the intermediate estimate $f^{k+1/2}$ in EM-TV algorithm. This idea to "optimize over optimization solvers" [Arridge et al., 2019]] can

be applied to a wide range of iterative algorithms such as gradient descent or algorithms that we presented in Chapter 2.

An application of unrolled methods to CT low-dose imaging has been proposed with the LEARN network [Chen et al., 2018] that corresponds to unrolling a gradient descent scheme for a cost function including a regularization term. An illustration of the network is given in Figure 3.11.

For CT reconstruction, the learned primal-dual algorithm [Adler and Öktem, 2018] has proven to be a reference method. It consists in adapting the Chambolle-Pock algorithm - Algorithm 1 presented in Section 1 - by replacing proximal operators with neural networks. Here again, sinogram data are used as input data and the forward operator and its adjoint are part of the network. The resulting learned iterative reconstruction scheme involves CNNs in both the image and data spaces - or primal and dual spaces.

While efficient, the method however remains heavy in terms of computational resources since the forward operator and its transpose are included in the network. This can represent a burden for 3D image reconstruction especially for areas such as cone-beam CT where the computation of projections is time and memory consuming. Another point to raise is the fact that CNNs do not have the same properties as proximal operators, so the guarantees on convergence for the learned primal-dual are weaker compared to the non data-driven version.

To alleviate the constraint of computation time and memory, a stochastic primal-dual unrolled network was recently proposed in [Tang et al., 2021] which, similarly to OSEM for MLEM, works on subsets of the forward and adjoint operators. This offers a significant improvement in terms of computational efficiency, and results on low-dose as

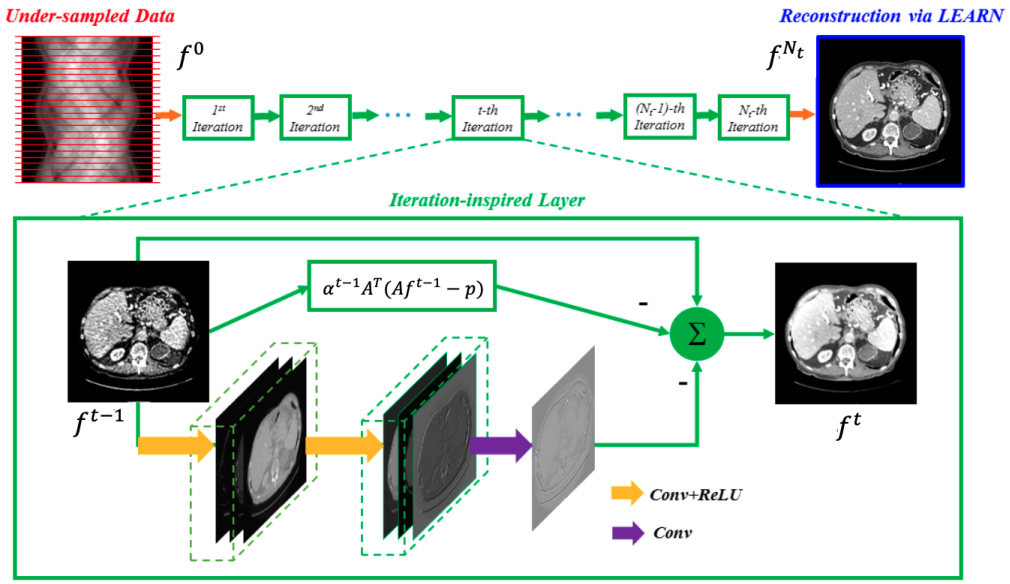


Figure 3.11: LEARN unrolling network [Chen et al., 2018]. The regularization parameter α^t can be different for each iteration block: it is learned. Estimates f^t in each iteration block are obtained via the equation $f^t = f^{t-1} - \alpha^t A^T(Af^t - p) - CNN(f^{t-1})$ where p are the projections and CNN the convolutional block represented in the scheme. The CNN and parameters α^t are optimized with supervised training.

well as sparse-view CT look very promising.

For PET imaging, unrolled iterative schemes also exist and we can mention EMNet [Gong et al., 2019a] and MAPEM-Net [Gong et al., 2019b] which, as their names suggest, are adaptation of EM and MAP-EM algorithms into neural networks.

We see that unrolling iterative schemes into neural networks is a very active research field for tomographic reconstruction and seems quite promising. It especially allows training networks with fewer amount of data compared to the methods derived so far, which is of paramount importance in the medical imaging area. Those methods seem to answer the issues of parameter tuning, though their main limitation remains the computational aspect. They might therefore not be suited for applications requiring fast reconstruction, even though algorithms such as [Tang et al., 2021] might open up for some perspectives in the future.

3.3 Generative networks for image reconstruction

In 2014 a revolution similar to the one observed for CNNs in the 1980s[1] emerged in the computer vision field with the development of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. As we showed previously that the efficiency of CNNs in the computer vision field can be translated to inverse problems and tomographic reconstruction, it seems relevant to investigate the application of GANs to this realm.

Interest in such methods for tomographic reconstrution has indeed grown very fast over the past few years [[Yi et al., 2019]. In this section we first derive the theory related to GANs, with the emphasis put on Wasserstein GANs since these are the networks that we will mostly consider in the experiments later. We then cover the various applications that GANs have for the reconstruction task, from supervised to self-supervised methods.

Note also that the realm of generative networks in deep learning also include the variational auto-encoders [Kingma and Welling, 2013]. We do not focus on such methods in this thesis so we do not discuss them in this chapter.

3.3.1 Generative adversarial networks

The initial objective of GANs, first proposed in [Goodfellow et al., 2014], is to learn a probability distribution. Instead of estimating the real probability distribution of some true images that belong to a space X, which explicit definition cannot be obtained, one can define a random variable $z \sim \pi_z$ and build a function $g_\theta : Z \to X$, parametrized by $\theta \in \Theta$ which generates samples according to a distribution P_θ. This function g_θ takes in practice the form of a neural network G_θ and its objective is to make P_θ as close to the true distribution P_r as possible.

This is made possible by a double structure consisting of a generator and a discriminator - or critic - as shown in Figure 3.12. The objective of the generator is to create realistic data - e.g face images, handwritten digits [Goodfellow et al., 2014] - while the discriminator aims at distinguishing between true and fake samples. In this setting for

[1]The actual breakthrough of convolutional neural networks in terms of performance in computer vision tasks rather appeared in the 2010s thanks to improved computational power, but the concept was developed in the 1980s.

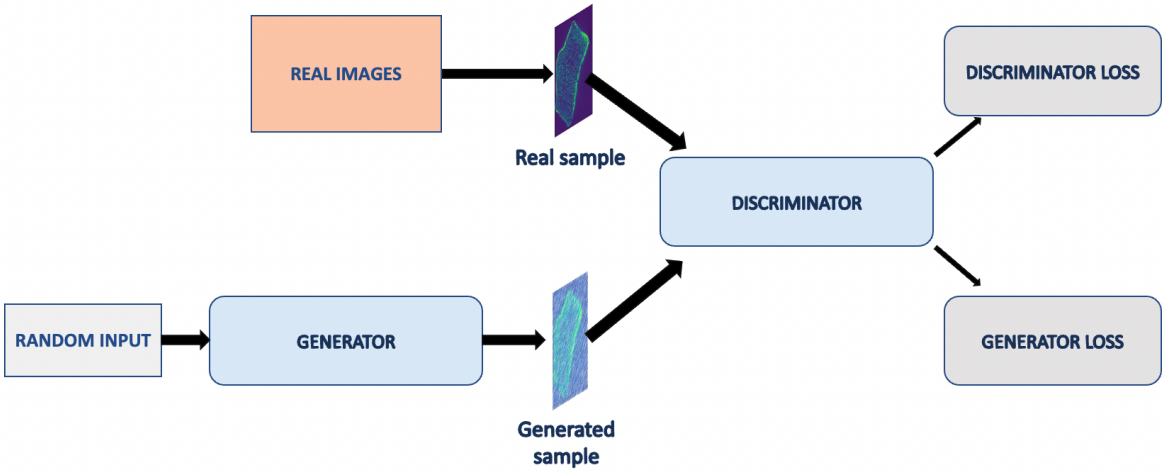


Figure 3.12: Generative Adversarial Network. Both the generator and discriminator are neural networks. The discriminator takes either a real or a generated sample. The losses are backpropagated to update the corresponding network's weights. The objective of adversarial learning is to make the generator able to create outputs in the distribution of real images.

GAN training, one thus only needs unsupervised data consisting of true samples of the distribution probability that is estimated.

The advantage of such a network is to easily generate data, noting that one does not need the expression of the density P_r. The important question to address is how to correctly measure the distance between two probability distributions P_r and P_θ.

In [Goodfellow et al., 2014], authors use a Jenson-Shannon (JS) divergence and derive a neural network able to generate such a distribution P_θ. The obtained network however turns out to be often delicate to train due to its instability. This is the reason why we focus here on the derivation of Wasserstein GANs (WGAN) [Arjovsky et al., 2017].

3.3.1.1 Wassertein 1-distance

Theory One important characteristic that one might expect when training a neural network to approximate a probability distribution is the continuity of P_θ w.r.t θ. Indeed, at each step t of the training, the network is a generator of samples that are distributed according to a distribution P_{θ_t}, where $(P_{\theta_t})_t$ is expected to converge towards the true probability distribution. Hence, continuity involves the fact that $\theta_t \to \theta \implies P_{\theta_t} \to P_\theta$. In this case only it seems reasonable to perform the optimization on parameters θ_t.

Convergence in terms of probability distributions can be defined in different ways. Indeed, a sequence of distributions $(P_{\theta_t})_{t\in\mathbb{N}}$ converges[1] if and only if there is a distribution P_θ such that the distance $d(P_{\theta_t}, P_\theta)$ tends to zero, which specifically depends on the distance d that one considers. In [Arjovsky et al., 2017], it is shown that continuity is not obtained when the convergence of probability distributions is considered w.r.t the JS distance, neither it is the case with distances like Total Variation or Kullback-Leibler divergence. The use of such distances between two probability distributions is a major

[1]Here we only consider convergence in distribution, i.e weak convergence.

explanation of the difficulty to train GANs. However, authors propose instead to use the Earth-Mover (EM) distance - or Wasserstein-1 distance - widely used for optimal transport problems. It is defined as follows :

$$W(P_r, P_\theta) = \inf_{\gamma \in \pi(P_r, P_\theta)} \mathbb{E}_{(x,v)\sim\gamma}[||x - v||_1] \tag{3.6}$$

where $\pi(P_r, P_\theta)$ is the set of all joint distributions $\gamma(x, v)$ whose marginals are P_r and P_θ respectively, and where $x \sim P_r$ and $v \sim P_\theta$ are samples of the generated one. These joint distributions can be interpreted as the mass that has to be transported from x to v in order to transform P_r into P_θ, the EM distance being the cost of the optimal transport plan.

Authors show that continuity of P_θ w.r.t θ is guaranteed with the EM distance, and that $\theta \rightarrow W(P_r, P_\theta)$ is not only continuous everywhere but also differentiable almost everywhere under mild assumptions. As a loss function to train generative networks, the EM distance between probability distributions seems therefore more appropriate compared to KL or JS distances[1].

Computation of the EM distance As defined in (3.6), it is not computationally feasible to evaluate the Wasserstein-1 distance. To address this issue, one might use the Kantorovitch-Rubinstein dual expression of this distance [Villani, 2008] :

$$W(P_r, P_\theta) = \sup_{h \in Lip(X)} \mathbb{E}_{x\sim P_r}[h(x)] - \mathbb{E}_{v\sim P_\theta}[h(v)] \tag{3.7}$$

where $Lip(X)$ denotes real-valued 1-Lipschitz maps on X, X being the image space. Once again it is not possible to cover the space of all these functions and one can parametrize them with a neural network $(D_w)_w$ where w represent the network's parameters. Using this parametrization and according to the way P_θ was built, the Wasserstein-1 distance is hence defined as

$$\max_{w \in W}(\mathbb{E}_{x\sim P_r}[D_w(x)] - \mathbb{E}_{z\sim\pi_z}[D_w(g_\theta(z)]). \tag{3.8}$$

The formulation (3.8) however does not take into account the 1-Lipschitz condition on the function D_w. To address this, authors in [Arjovsky et al., 2017] used weight clipping in the corresponding neural network. This amounts to restricting the range of potential values that can be taken by the parameters in the network,, which reduces the space of potential functions that can be represented by this network.

A solution is proposed in [Gulrajani et al., 2017] where the authors first show that the optimal solution h^* of (3.7) has, under mild assumptions, a gradient norm value of 1 almost everywhere. The resulting distance is hence proposed as follows:

$$\max_{w}(\mathbb{E}_{x\sim P_r}[D_w(x)] - \mathbb{E}_{z\sim\pi_z}[D_w(g_\theta(z)] + \lambda\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D_w(\hat{x})||_2 - 1)^2]) \tag{3.9}$$

where $\hat{x} \sim P_{\hat{x}}$ are straight lines between samples from P_r and the generated distribution P_θ, and λ is a weighting parameter for which a standard value of 10 is generally considered. The added term is the gradient penalty that aims at ensuring the 1-Lipschitz condition on

[1]EM distance is said to be weaker than other distances as for convergence of probability distributions.

the discriminator. It is two-sided since it appeared to be more efficient and the optimal solution has a 1-gradient norm almost everywhere.

This WGAN with a gradient penalty term (WGAN-GP) is able to learn much more complex probability distribution compared to the previous one that makes use of weight clipping.

In practice the training of such a network consists in alternately updating the generator's parameters by minimizing

$$\mathbb{E}_{x\sim P_r}[D_w(x)] - \mathbb{E}_{z\sim\pi_z}[D_w(g_\theta(z)] + \lambda\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}}D_w(\hat{x})||_2 - 1)^2] \tag{3.10}$$

w.r.t θ and the discriminator's parameters through maximization of (3.10), both with optimization methods common to neural networks.

3.3.1.2 Conditional GAN

Before delving into the various applications of GANs for tomographic reconstruction, we detail here the notion of Conditional Generative Adversarial Networks (CGAN) as they represent an important part of the research towards using for GANs medical imaging.

CGANs for computer vision The first time conditional GANs were introduced was in 2014 in [Mirza and Osindero, 2014], shortly after the release of the paper from Goodfellow et al. The idea underlying CGAN is actually really close to traditional GANs; the generator/discriminator structure is kept, the only difference is that images are generated conditionally to some information. For instance instead of generating images of handwritten digits, one can generate them conditionally to the information "1", "2", etc. If the condition put in the generator is "1", the produced images will - assuming the network has been correctly trained - still be handwritten digits with different handwriting styles, but this time only images representing the digit "1" will be generated. From an implementation point of view, this amounts to concatenating the information to the latent variable z as input to the generator network, and also concatenating the same information to the input of the discriminator, namely the generated image. The principle of CGAN is illustrated in Figure 3.13.

Formally, the minimax game that both the generator G_θ and the discriminator D_w are playing can be represented by the following optimization process :

$$\min_\theta \max_w L_{cGAN}(D_w, G_\theta, y) = \mathbb{E}_{x\sim P_r(y)}[\log D_w(x|y)] + \mathbb{E}_{z\sim\pi_z(y)}[\log(1 - D(G_\theta(z|y)|y)] \tag{3.11}$$

with x the real data, z the latent variables and y the condition on which the image generation is based. For π_z a simple distribution - e.g Gaussian - is generally considered in practice. Note that here the formulation is based on the Jenson-Shannon distance. Just like traditional GANs, θ and w are updated alternately with some optimizer and backpropagation of the gradient of the loss function.

The adaptation of CGANs to image-to-image translation was proposed three years later in [Isola et al., 2017]. Here the condition given to the generator corresponds to another image. An example can be the generation of a landscape image given the image of all of its edges; or the generation of a colored image given a gray-scale version of it. One needs to keep in mind here than the idea is still to generate a distribution of

images given a single condition; for instance given a gray-scale image, one can generate an entire probability distribution of corresponding colored images. Indeed the purpose is to learn a mapping from a given image y and random noise vector z to the output v, $G : (y, z) \rightarrow v = G_\theta(z|y)$. In practice authors actually add a L_1 loss to the objective function of the generator so that it is trained with

$$L_G(\theta) = L_{cGAN}(D_w, G_\theta) + \eta L_1(G_\theta). \tag{3.12}$$

with η a weighting parameter. The reason is that this allows ensuring that the generated output matches the given conditional image in a pixel-wise manner. Authors mention that L_1 allows capturing the low frequencies, and the GAN discriminator is restricted to model high-frequency structures. We will go back to more thorough analysis of such loss functions in the next chapter.

Authors also point out the fact that the input random noise z tends to be ignored by the network in their experiments which thus gives almost deterministic outputs.

Conditional Wasserstein GAN In [Adler and Öktem, 2018], the concept of Wasserstein GANs is extended to conditional generation of images: the so-called conditional Wasserstein GAN (CWGAN). We will see later how such a method can be applied to tomographic reconstruction and we derive here the theory. Given an input $y \in Y$, one wants to approximate the posterior $\pi(x|y)$ with $G(y)$ that maps each $y \in Y$ to a probability measure on X where X is the set of potential images for example. An optimal generator G is then

$$G^* = \arg \inf_{G \in \Gamma} \mathbb{E}_{y \sim \sigma}[W(G(y), \pi(.|y))] \tag{3.13}$$

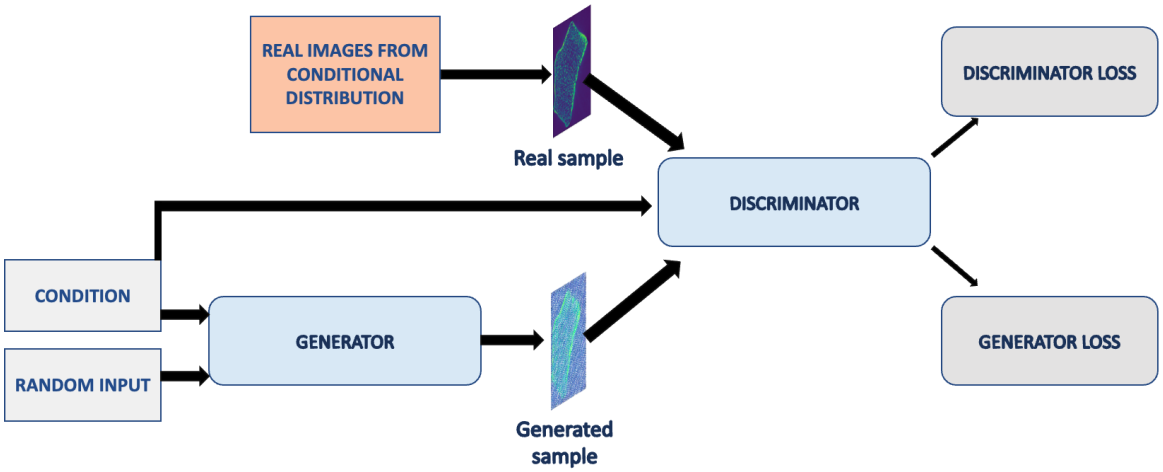


Figure 3.13: Conditional Generative Adversarial Network. Both the generator and discriminator are neural networks. The generator has both a random tensor and a condition as input. The discriminator takes the conditional data as input along with either a real or a generated sample. The losses are backpropagated to update the corresponding network's weights.

where σ is the unknown distribution that generates the input data; in practice its empirical counterpart can be taken. Here we measured the distance between the two distribution probabilities with the Wasserstein-1 distance for all the reasons we mentioned before. The Kantorovitch-Rubinstein duality theorem gives in this case

$$W(G(y), \pi(.|y)) = \sup_{h_y \in Lip(X)} \mathbb{E}_{\substack{x \sim \pi(.|y) \\ v \sim G(y)}}[h_y(x) - h_y(v)]. \tag{3.14}$$

Moreover, authors prove that

$$\inf_{G \in \Gamma} \mathbb{E}_{y \sim \sigma}[\sup_{h_y \in Lip(X)} \mathbb{E}_{\substack{x \sim \pi(x|y) \\ v \sim G(y)}}[h_y(x) - h_y(v)]] = \inf_{G \in \Gamma} \sup_{D \in D(X \times Y)} [\mathbb{E}_{\substack{(x,y) \sim \mu \\ v \sim G(y)}}[D(x,y) - D(v,y)]] \tag{3.15}$$

where $D(X \times Y)$ represents mappings on $X \times Y$ that are 1-Lipschitz in the X-variable for every $y \in Y$, and μ the joint distribution of data (x, y) which is unknown but which can be estimated with empirical data.

For some Z-valued realization $z \sim \eta$ we have $v = G_\theta(z, y)$ and the discriminator D is also parametrized so that we have

$$G_\theta : Z \times Y \to X$$
$$D_w : X \times Y \to \mathbb{R}$$

The parameters for the optimal generator are obtained as

$$\theta^* \in \arg\min_{\theta \in \Theta} \sup_{w \in W} \mathbb{E}_{\substack{(x,y) \sim \mu \\ z \sim \eta}}[D_w(x, y) - D_w(G_\theta(z, y), y)] \tag{3.16}$$

where in practice parameters w are trained jointly with θ according to (3.16). For training the joint distribution μ is replaced with training paired data (x_i, y_i). Note that here also a gradient penalty term must be added to ensure the 1-Lipschitz condition on D_w.

Such a CWGAN can then be used to generate a probability distribution given an input data - an image for instance - and we show hereafter that this can have applications for medical imaging.

3.3.2 Application to image reconstruction

Now that we have derived the theoretical basis underlying (conditional) generative adversarial networks, we present different ways to use such networks for the image reconstruction task. We will cover the case where a Wasserstein distance is used to train a network for a reconstruction post-processing task; we also derive an application of CWGAN to generate a posterior sampling of images given an input data and discuss potential applications that it can have in the medical imaging realm; we also mention the interest of GANs when supervised data is not available. Note also that we do not cover in this section the possibility to use adversarial networks as regularization functionals in iterative schemes, nor we will detail the applications for self-supervised learning, as it will be part of the related specific categories in the next sections.

3.3.2.1 GANs for reconstruction post-processing

Here the pipeline for reconstruction is the same as for 3.2.1 : some initial reconstruction - usually the FBP - is fed into a neural network that aims at enhancing the quality of the obtained image. The difference is that the training of such a network involves an adversarial loss: the Wasserstein distance for instance. For the reasons explained earlier for the image-to-image translation network, a content loss - e.g pixel-based losses such as the mean squared error or the mean absolute error - is almost always added to the adversarial loss, the objective being not only to retrieve high-frequency features with the adversarial loss but also to retrieve low-frequency information and similarities with the input data. In extreme cases without any content loss, one could obtain networks that generate images from the true image distribution, but that do not correspond to the FBP given as an input of the network.

In [Yang et al., 2018], an adversarial loss based on the Wasserstein distance is used to denoise a FBP reconstruction for CT imaging. The objective of the network is to match this FBP obtained from low-dose acquisition (LDCT) with the FBP that could be retrieved with normal-dose acquisition (NDCT). The network is thus trained with paired low-dose/normal-dose FBPs. The content loss used in addition to the Wasserstein distance is a perceptual loss and we will delve into this type of loss functions in the next chapter.

The structure of the corresponding network can be represented as in Figure 3.14. In this case the discriminator takes either a generated image G_θ(LDCT) or a ground-truth NDCT image and the value of its output is as high as this input "is believed" to be a generated/false image.

The same idea is considered in DPIR-Net [Hu et al., 2020] for PET imaging. The only difference here is that the network is directly fed with sinogram data instead of FBP or some initial reconstruction; so this cannot be considered as a post-processing

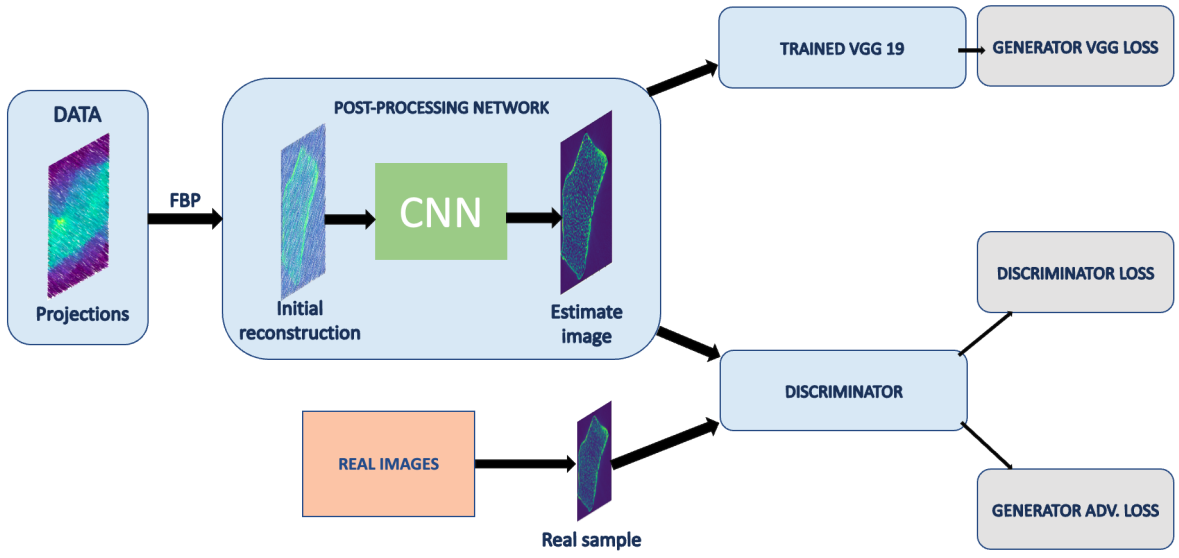


Figure 3.14: Scheme of WGAN-VGG. Both the generator and discriminator are updated during training, while the weights of VGG are fixed. The generator is trained with both VGG and adversarial (Adv.) losses.

method but it rather corresponds to an direct reconstruction framework. In this work both perceptual and pixel-based losses are added to the adversarial loss to generate high-quality PET images from the raw data.

Application to 3D data was also performed in [Wang et al., 2018] for the enhancement of PET images reconstructed from low-dose data. Here authors refer to conditional generative adversarial networks but the generated image is unique; this theoretically corresponds to a conditional GAN where the sampling distribution is a Dirac distribution. Their 3D network allows obtaining better performance compared to 2D and the adversarial loss turns out to be particularly efficient at retrieving specific characteristics from region of interest.

From a general point of view, we see here that there are two ways to interpret such adversarial networks for post-processing a reconstruction regarding the theory derived so far. On the one hand, they can be considered as similar to image-to-image translation networks [Isola et al., 2017] - with a different content loss but this is not our interest here -, so that they belong to the category of CWGANs. However the output generated by such a network is deterministic, so the corresponding latent variables follow a Dirac distribution: in practice there is no need to consider such latent variables and no sampling is needed here.

On the other hand, one can consider these networks as traditional WGANs, with the difference that the input distribution π_z corresponds to the LDCT images distribution. Instead of sampling data from a classic Gaussian distribution for instance, the network maps the distribution of LDCT to the one of NDCT. The content loss is then here to make sure that both LDCT and NDCT samples correspond to the same data.

In practice the difference between the two interpretations lies on the conditioning of the discriminator or not; if the input data is concatenated to the discriminator, then it refers to a CWGAN. Otherwise it is simply a WGAN where the latent distribution is the distribution of LDCT images.

In the literature for tomographic reconstruction the notion of GAN (or conditional GAN) is generally used though what really matters is the notion of adversarial loss - Wasserstein 1-distance in most recent works - that is used to train a network that predicts the NDCT image. In those methods a deterministic output is generated, which is different from the initial objective of GANs. We will see in what follows how GAN's stochasticity can be used for medical imaging.

3.3.2.2 Conditional GANs to estimate a posterior distribution

An application of conditional Wasserstein GANs for medical imaging was given in [Adler and Öktem, 2018], for which the theory has been derived earlier. The decribed method is a conditional GAN as originally defined in [Mirza and Osindero, 2014] and [Isola et al., 2017] since images are generated in a stochastic manner, conditionally to some FBP data.

Given a LDCT image y, the stochastic network does not reconstruct directly the NDCT version of y; rather it estimates the posterior distribution $x \to \pi(x|y)$ of the NDCT images that can correspond to the low-dose version y. In practice for a given y the generator $z \to G_{\theta^*}(z, y)$ thus takes as input both y and random noise z drawn from a distribution that can be easily computed. It generates a sample $G_{\theta^*}(z, y)$ from $\pi(x|y)$. It is possible to consider the mean image as the most likely NDCT version of y, or also to

compute the variance image from the estimated distribution.

This represents a completely different approach compared to traditional image reconstruction methods, for which the information brought by the estimation of the posterior distribution can be of particular interest. One can indeed obtain statistical features by sampling the posterior; for instance the probability of presence or not of a nodule corresponding to a tumor can be estimated with such a network. Contrary to deterministic methods, the posterior distribution allows one to assess whether the presence of such elements on some reconstructed image is due to a statistical fluke or if it is indeed relevant to consider. More generally, hypothesis tests can be performed with such a method.

From a computational point of view, authors show that the formulation (3.16) might lead to mode collapse, i.e a case where the network does not take noise into account and the variability of generated images is very low.

Therefore a modification on the discriminator is proposed to address this issue if one is interested in obtaining variability on the samples generated. The WGAN loss with the conditional WGAN discriminator is given by the following, the expectation being replaced by summation over training samples that consist of paired (x, y) data :

$$L_W(\theta, w) = \mathbb{E}_{\substack{(x,y)\sim\mu \\ (z_1,z_2)\sim\eta}} [\frac{1}{2}(D_w((x, G_\theta(z_1, y)), y) + D_w((G_\theta(z_2, y), x), y)) - D_w((G_\theta(z_1, y), G_\theta(z_2, y)), y)], \tag{3.17}$$

the loss function for the generator being in [Adler and Öktem, 2018]

$$L_G(\theta) = L_W(\theta, w) + 10^{-4}||\theta||^2 \tag{3.18}$$

and the loss function for the discriminator is

$$L_D(w) = -L_W(\theta, w) + 10L_{\text{grad}}(\theta, w) + 10^{-3}L_{\text{drift}}(\theta, w) + 10^{-4}||w||^2 \tag{3.19}$$

with L_{drift} a regularization on the discriminator that prevents training from being unstable because the loss in invariant w.r.t a constant in D_w, and L_{grad} is to impose the 1-Lipschitz constraint on the discriminator as in (3.9). We gave here the values given in [Adler and Öktem, 2018] for the weighting parameters in the loss functions.

It is shown in [Adler and Öktem, 2018] that θ^* is a minimizer of $L_W(\theta, w^*)$ in (3.17) - where w^* is obtained by taking the supremum on w - if and only if it is a minimizer in (3.16). In practice however, training a WGAN with (3.17) has the advantage to obtain more variability in the generated outputs, which corrects the tendency of conditional GANs to ignore the input noise as described first in [Isola et al., 2017]. The reason for such an outcome is that in (3.17) the discriminator distinguishes between unordered pairs in the image space containing either true images or random samples generated by the generative model G_θ.

3.3.2.3 Case with no paired training data

In the previous section we assumed that we had access to paired data; the most common case being a sinogram - or its FBP - obtained with low-dose settings, paired with a desirable reconstruction that could be performed with normal dose. For multiple reasons, medical imaging is known to be a field where such datasets are hard to obtain. Especially, there might be cases where low-dose and normal-dose reconstructions are available, but

they are not paired. In practice indeed, only one acquisition is performed so pairs of such data rarely exist without any simulation involved.

Here again, advances in the computer vision field can be transcribed to tomographic reconstruction problems. CycleGANs [Zhu et al., 2017] correspond to generative adversarial networks that are specifically designed for applications as the ones described here when no paired data are available. The initial idea was to map similar images with low-level features that are different: a paint and a photograph, a winter landscape and its summer version, two images with different colors, etc.

This amounts to learning to translate an image from a source domain Y to a target domain X in the absence of paired examples. For this, the aim is to find a mapping G such that $G(y) \in X$ is indistinguishable from true images $x \in X$. This task could be achieved using a simple adversarial loss; however we want in addition to have an image that matches its corresponding version in the source domain.

This is the reason why authors suggest that the translation should be cycle-consistent. If we have a translator $G : Y \to X$ and another translator $H : X \to Y$, we would expect that $H(G(y)) \approx y$ and $G(H(x)) \approx x$. CycleGAN thus combines a cycle consistency loss with the adversarial losses on domains X and Y.

For a generator G that maps some $y \in Y$ into the domain X and a discriminator in this domain D_X, considering the JS distance we have the following adversarial loss:

$$L_{GAN}(G, D_X) = \mathbb{E}_{x\sim P_r}[\log D_X(x)] + \mathbb{E}_{y\sim \pi_{data}}[\log(1 - D_X(G(y)))] \tag{3.20}$$

where π_{data} (resp. P_r) is the distribution of data in the domain Y (resp. X). The consistency loss

$$L_{cyc}(G, H) = \mathbb{E}_{x\sim P_r)}[||G(H(x)) - x||_1] + \mathbb{E}_{y\sim \pi_{data}}[||H(G(y)) - y||_1] \tag{3.21}$$

is then added to the generator loss function for the reasons explained before. The final objective function for a CycleGAN can be written

$$L(G, H, D_Y, D_X) = L_{GAN}(G, D_X) + L_{GAN}(H, D_Y) + \lambda L_{cyc}(G, H) \tag{3.22}$$

with λ a weighting parameter; the aim is then to solve

$$G^*, H^* = \arg\min_{G,H} \max_{D_X, D_Y} L(G, H, D_X, D_Y) \tag{3.23}$$

The use of such a CycleGAN was performed in [Lim and Ye, 2019] for microscopy applications, but this is also relevant for the cases of CT or PET modalities considering unpaired distributions of low-dose and normal-dose images for instance.

If this method is particularly interesting for the case of unpaired data, one needs to remember that whenever paired training data are available, image-to-image GANs as derived before are more efficient compared to a CycleGAN.

3.3.3 Improvements in GAN structures

We saw that GANs can be a method of choice for image reconstruction; an essential element in the way a GAN is built is the choice of the loss function to consider for the adversarial training of both the generator and discriminator networks. We mentioned the

Jenson-Shannon distance which was originally proposed in [Goodfellow et al., 2014] and the Wasserstein-1 distance as it showed improvements for training the networks. The latter is derived from optimal transport theory; in [Salimans et al., 2018] an other way to approximate the Wasserstein 1-distance is proposed with the aim to make training more stable. In practice however the memory requirements for such a method can be a limiting factor.

More generally, the choice of the loss function to use for generative adversarial networks has represented an active field of research [Genevay et al., 2017, Wu et al., 2019].

It is also to be noted that the performance of GANs has improved fastly over the past few years, and this might be beneficial in the future for tomographic reconstruction. Work on the structure of those networks has especially allowed to generate images that are more and more realistic, and one can especially mention in the chronological order ProgressiveGAN [Karras et al., 2017], StyleGAN [Karras et al., 2019][1] and the most recent network at the time of the writing being StyleGAN3 [Karras et al., 2020].

3.4 Learning a regularization term

All of the previous methods consist of a reconstruction operator that involves a neural network; in all of the cases so far, the network produces an estimated solution. In this section we consider a different paradigm where a neural network is only used as a regularization term for solving the optimization problem discussed in Chapter 2 with traditional iterative methods. Learning a regularization term allows getting rid of the choice of a handcrafted regularization function. The computation time remains however roughly the same as the reconstruction is performed with the same algorithms in the end. We derive here two different methods for such a purpose.

3.4.1 Adversarial regularizer

We recall the objective function in (2.12)

$$\min_f \mathrm{d}(A(f), p) + \alpha R(f) \tag{3.24}$$

with d some data-fidelity distance - l_2 or Kullback-Leibler for example -, and $R(f)$ a regularizing term for the image f, with weighting parameter α. The focus here is put on the choice of the term $R(f)$: it needs to take small values for samples in the distribution of "true" images (e.g NDCT), and larger values when samples are further from this distribution. So far we have set fixed the regularization function, considering it especially as a TV norm. In [Lunz et al., 2018], neural networks are used in order to represent this function

$$R(f) = S_\theta(f) \tag{3.25}$$

with S_θ a deep neural network parametrized by weights θ. Denoting the distribution of correct samples as π_{true} and the distribution of other images as π_{false}, the network S is

[1] An interesting explanation of the method can be found in https://www.youtube.com/watch?v=kSLJriaOumA .

trained according to the loss function

$$L(\theta) = \mathbb{E}_{f\sim\pi_{true}}[G_1(S_\theta(f))] - \mathbb{E}_{h\sim\pi_{false}}[G_2(S_\theta(h))] \tag{3.26}$$

with G_1 and G_2 some monotone functions. In practice the distribution π_{false} can correspond to the FBP of n measurements data p while we have access to m ground truth data f so that we have

$$\hat{\theta} \in \arg\min_\theta \left\{ \frac{1}{m}\sum_{i=1}^{m} G_1(S_\theta(f_i)) - \frac{1}{n}\sum_{i=1}^{n} G_2(S_\theta(FBP(p_i))) \right\}. \tag{3.27}$$

In [Lunz et al., 2018] G_1 and G_2 both correspond to the identity function. Also a gradient penalty term is added to the network S so that the function to minimize becomes

$$L(\theta) = \mathbb{E}_{f\sim\pi_{true}}[S_\theta(f)] - \mathbb{E}_{f\sim\pi false}[S_\theta(f)] + \lambda\mathbb{E}_{\hat{f}\sim\pi_{mix}}[||(\nabla S_\theta(\hat{f})|| - 1)^2] \tag{3.28}$$

where λ is a weighting parameter and π_{mix} is the distribution of samples drawn on straight lines between samples drawn from both π_{true} and π_{false} : we recognize the cost function of a discriminator in a WGAN-GP scheme. Here instead of having a generator that produces samples in π_{false}, the distribution correspond to known poor-quality FBPs.

The notion of adversarial regularizer thus refers to the training of a neural network with an adversarial loss - here the Wasserstein 1-distance - so that it acts as a discriminator which purpose is to distinguish between true and deteriorated images.

3.4.2 NETT approach

In [Li et al., 2020], the aim is also to find a deep architecture to represent the regularization functional, but the method differs a little bit from the previous one. This time, the regularization is written as

$$R(f) = S(\Phi_\theta(f)) \tag{3.29}$$

with $\Phi_\theta : X \to \Theta$ a DNN. The regularization function $S : \Theta \to [0, \infty[$ is fixed this time, but instead of applying it to f directly, it takes values in the space Θ which corresponds to the output of a network. Once again, S is to take small values for desired images and penalize those with artefacts or other unwanted structures.

For Φ, authors consider the encoder part of a UNET-like encoder-decoder network; S is a norm on the output of Φ.

We denote the entire network as $\Psi \circ \Phi$ where Ψ is the decoder with parameters θ'. The training of $\Psi \circ \Phi$ proceeds as follows: some training phantoms x_n are used, as well as their back-projection images denoted as z_n. The difference between these images is then stored as $r_n = x_n - z_n$ for $1 \leq n \leq N$. High value of the components of r_n should imply a high value for $S(\Phi_\theta(z_n))$ so that the regularizer is efficient. On the other hand, one can set $z_n = x_n$ ($r_n = 0$) for $N \leq n \leq 2N$, which corresponds to perfect reconstruction of ground-truth images x_n. The way the overall network is trained is by minimizing the following energy function

$$\hat{\theta} = \min_\theta \mathbb{E} = \sum_{n=1}^{2N} \ell(\Phi_\theta \circ \Psi_{\theta'}(z_n), r_n) \tag{3.30}$$

considering some distance ℓ that is generally a mean squared error or mean absolute error. This makes the network producing outputs that are close to r_n; in other words given some image z_n, the whole network is trained to estimate the artifact part of z_n. With this remark, it is to be expected that some norm taken from the encoder part should be low when the image z_n is artifact-free, and high otherwise.

Once the network is trained and the parameters have been optimized, the reconstruction operator is hence taken as

$$\Gamma_{\hat{\theta}} = \arg\min \mathrm{d}(A(f), p) + \alpha S(\Phi_{\hat{\theta}}(f)). \tag{3.31}$$

NETT and adversarial regularizers thus represent alternatives to deep reconstruction operators, but the computation time implied by the iterative minimization process is way higher. It would be interesting to evaluate the performance of such methods against faster methods such as e.g reconstruction post-processing, where training data can be obtained in the same way as for the adversarial regularizer. We also note that there are other methods in the literature for learning a regularization term such as [Zhang et al., 2017] and [Obmann et al., 2020].

3.5 Self-supervised methods

The methods discussed so far in this chapter require having access to a training dataset, whether it is supervised - i.e with paired input/ground-truth images - or not. Neural networks can also be trained in a self-supervised setting. In this case one no longer needs to have access to a training dataset, but rather learning is performed *on the fly* with only the input data to learn from.

With self-supervised methods, the task is to reconstruct an image from data as input: this is a setting similar to methods discussed in Chapter 2. As the methods that we are discussing involve neural networks and related notions such as CNN, activation functions, backpropagation of the gradient to update weights, these methods are referred to as self-supervised learning methods.

There are several ways to learn without any training dataset; we focus here on a specific method, namely the Deep Image Prior (DIP) that was proposed for inverse problems in [Ulyanov et al., 2017]. The reason is that this is a method at the basis of many reconstruction techniques, and we will propose a DIP-based method in Chapter 6. We will briefly mention other typical methods based on self-supervised learning for tomographic reconstruction at the end of this section.

3.5.1 Deep Image Prior

Deep Image Prior is a concept developed in [Ulyanov et al., 2017] for general inverse problems with applications in the computer vision field. It showed excellent results for various tasks such as denoising super-resolution and inpainting for instance.

The concept is based on a paradigm shift compared to the learning-based methods that we presented so far. The core of the idea is that priors on images are captured by the structure of a generator such as a convolutional neural network. This is quite opposite to the common thought that image priors *must* be learned from some training data. It

is argued that the structure of a network "resonates" with the structure of the data that one aims at generating. For instance, convolutional operations in CNNs naturally impose self-similarity on the generated images because the filters are applied across the entire visual field.

In DIP principle, the network weights serve as a parametrization of the image to be retrieved. An image f can therefore be represented by a neural network Γ_θ with parameters θ as

$$f = \Gamma_\theta(z) \tag{3.32}$$

with z that is some fixed input tensor for the network.

In practice, the solution f^* is found by finding optimal parameters θ^*. The only information used by Γ_θ to do so is the input data - e.g noisy/blurry image, projections - and the structure of the network that is considered.

If we denote the deteriorated input data as f_0, the optimal solution is found by minimizing an energy function such as

$$f^* = \arg\min_f J(f, f_0) + R(f) \tag{3.33}$$

where $R(f)$ is a regularization term considered on the image f. The form of J is entirely dependent on the task, whether it is inpainting, denoising or tomographic reconstruction. The minimizer θ^* - such that $f^* = \Gamma_{\theta^*}(z)$ - is found with typical optimizers such as gradient descent or Adam, after being randomly initialized. A scheme of DIP principle is shown in Figure 3.15.

The regularizer $R(f)$ aims at capturing the regularity on natural images, with Total Variation for instance, whereas it is shown that a significant amount of information on the image distribution is contained in the structure of the network Γ.

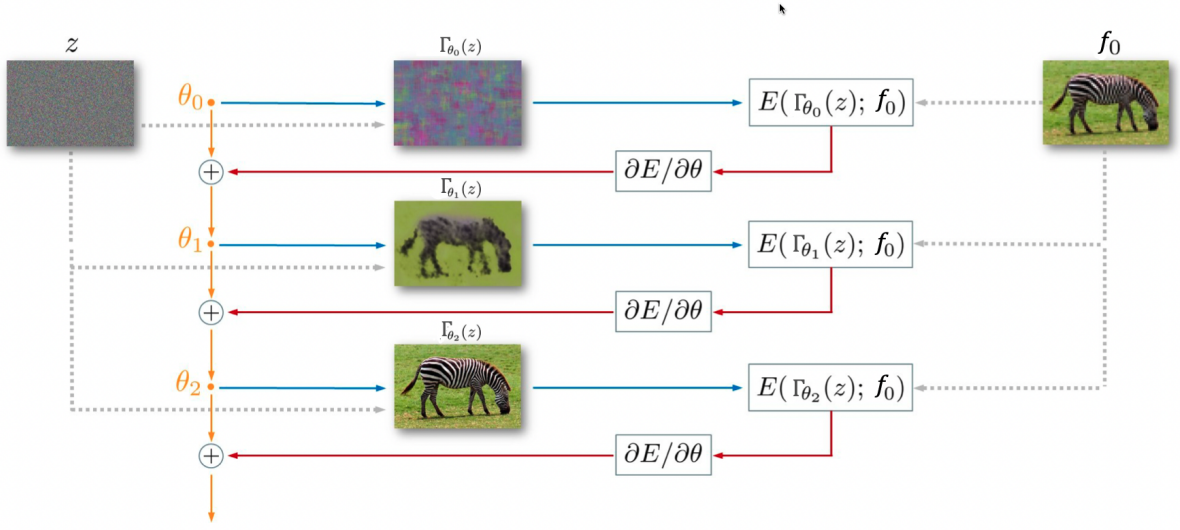


Figure 3.15: Deep Image Prior scheme [Ulyanov et al., 2017]. At every iteration the weights θ are mapped to an image $f = \Gamma_\theta(z)$ where Γ is a neural network with parameters θ. We consider here an energy function $E(f, f_0)$ that depends on the estimate image f and the input data f_0. An example of such an energy function is (3.33).

3.5.2 Application to tomographic reconstruction

In what follows we show an example on how to use DIP for tomographic reconstruction.

As we mentioned the framework of DIP is general and the difference between tasks lies on the choice of the energy function J. Also it is shown in [Ulyanov et al., 2017] that the choice of the architecture of Γ does have an impact on the prior considered for the image distribution[1].

In [Gong et al., 2019] the DIP framework is adapted to the task of PET image reconstruction. The corresponding framework is named DIPRecon.

Considering PET measurements data y, a natural choice for the energy function is the log-likelihood considering Poisson noise on the data. Therefore in DIPRecon the objective function is

$$J(y, f) = KL(y, A\Gamma_\theta(z)) \tag{3.34}$$

with KL the Kullback-Leibler divergence and A the forward operator for the PET model geometry that is considered. The network's architecture is similar to a 3D UNET, and it is also observed that the performance is enhanced when considering the input tensor z as an image obtained from an other modality, e.g the T1-weighted MR image in their case. Overall better performance is observed with DIPRecon compared to traditional iterative methods such as MLEM.

3.5.3 Other methods

Other self-supervised methods have been successful in the recent years; they mostly correspond to denoising methods and the paradigm is different compared to DIP. One can mention Noise2Noise [Lehtinen et al., 2018], Noise2Void [Krull et al., 2019], Noise2Self [Batson and Royer, 2019], Noisier2Noise [Moran et al., 2020], etc. The latter for instance proceeds as follows:

1. Degrade the input image y assuming a deterioration process. For instance, add Poisson noise to an image that is supposed to be corrupted by Poisson noise. Let's call this synthetic image y'.

2. Train a neural network to retrieve y from y'.

3. Apply the trained network to y in order to retrieve the clean image.

This process could be potentially adapted to image reconstruction.

Finally we mention a method that has been implemented for cryo-electron microscopy (cryo-EM) in [Gupta et al., 2021], which pipeline is as follows:

1. Start with an initial estimation of the image.

2. Feed this image into a physics simulator; this corresponds to a forward model adapted to cryo-EM in the original paper.

[1] It is shown that different network architectures lead to different priors. However it is shown that any architecture can reach the optimal solution; the difference is that some architectures allow faster convergence compared to others depending on the task.

3. Compare the generated output with the real measurements data using a discriminative network.

4. Update the estimate image with backpropagation of the gradient of the adversarial loss.

If this scheme seems very promising and able to adapt to tomographic reconstruction in CT or PET for instance, it is to be noted that the number of projections data in cryo-EM is way higher, e.g $41,000$ in [Gupta et al., 2021]. There is therefore no guarantee that such a method would be suited for CT or PET.

3.6 Conclusion on deep learning methods

This chapter illustrates how wide the spectrum of deep learning methods for tomographic reconstruction is. There is probably not one category that is strictly more adequate to one other regardless of the application. That is to say, one should consider one of the different categories that we derived here depending on the application requirements: data availability, requirements for reconstruction time, training time, diversity of the images of interest, etc. We sum up in Table 3.1 the main - and current - pros and cons for each type of methods, even if this is of course more complex in reality as within each category there might be solutions to overcome some of the limitations. The stochastic primal-dual unrolled network is an example of potential improvements that can be obtained within the realm of unrolled iterative schemes. Furthermore, the flexibility of deep learning based methods implies that combinations of different methods can be considered depending on the applications; this idea will be delved into in Chapter 6.

In what follows we will perform two different studies, the first one on CT data and the other for PET modality. In the next chapter we will study the impact of the loss function of deep neural networks on the quality of the reconstructed images; the last chapters are dealing with blind deconvolution and especially an application on time-of-flight (TOF) intraoperative PET imaging. As the impact of the loss function can be studied regardless of the structure of the network, and TOFPET imaging is a modality that requires very low reconstruction time, from now on we will mainly work with the first category mentioned in this chapter which are post-processing methods.

Method	Pros	Cons
Post-processing	Fast reconstruction	Large supervised dataset required
Direct method	Knowledge of forward operator not required + more flexibility	Large dataset required + high number of parameters. Potentially highest reconstruction time
Unrolling	Forward operator included in the model + less training data required	Reconstruction time and memory footprint
GAN : deterministic output	High-frequency features in the reconstructions	Harder to train (+ need for a content loss)
Posterior sampling (i.e. GAN)	Statistical information	Reconstruction time
Regularizer learning	No paired data + better suited prior compared to iterative methods	Reconstruction time
Self-supervised	No training data required	Slow convergence so high reconstruction time

Table 3.1: Summary of the different categories of deep learning methods for tomographic reconstruction.

Chapter 4

Impact of the loss function on deep learning based bone microarchitecture reconstruction

We saw in Chapter 3 that the range of possibilities to design a neural network for tomographic reconstruction in the medical context is wide, and still many more methods are likely to appear in the future. Modifications of the architecture of networks or training strategies are among the numerous elements that are tweaked to improve the efficiency of deep learning methods for the reconstruction task. One thing that is however common to almost all of those methods is the fact that neural networks rely on the backpropagation of the gradient of a loss function in order to adjust their parameters. This loss function has a role of evaluation and answers the question: how good the prediction is compared to some reference image? This is obviously the case for supervised post-processing methods and unrolled iterative schemes, but even for self-supervised learning one needs to compare the output of the network to some data.

The choice of a loss function for training neural networks is a topic of interest for both the computer vision and the medical imaging fields. This choice can have a significant impact on the output of a network during inference (i.e for the prediction once the network is trained). The role of the loss function is key since it is the element given to the network for it to assess its own performance; for instance, if the mean squared error between predictions and reference images is decreasing, this suggests that the performance of the network is improving (on the training set at least). The efficiency of the learning process is thus based on the capacity of the loss function to be in adequacy with the role of the network: does minimizing the loss function correspond to producing outputs that are satisfying with respect to the actual purpose of the network?

In this chapter we put the emphasis on the choice of the loss function used during training. As the main requirement in practice for a reconstructed image is that it is reliable and exploitable for diagnosis, we drive our study towards the impact of the loss function on qualitative and quantitative information that are relevant to this purpose. As most of the loss functions that we consider have proven to be efficient when evaluated in the field of computer vision, we aim at extending or not these results when the quality of a network's predictions is assessed w.r.t a specific medical aspect.

In order to perform such a study, we need to focus on a particular application. Our

first objective is to address the case of image reconstruction for low-dose imaging, as it is currently an important topic for deep learning based methods, and the deterioration of the images due to corrupted data might have a significant impact on both quantitative and qualitative information available for the practitioners. We also need to deal with data that allow to perform a study that is comprehensive enough and that can be useful to the application domain. For this we consider micro-CT (μ-CT) data for bone microarchitecture imaging; the complexity of the corresponding images' structure is such that comprehensive analysis can be performed so that significant conclusions can be drawn as for the comparative study of loss functions. Moreover, a key point is that diagnosis of bone pathologies is directly linked to the computation of quantitative parameters; hence it is of major importance to assess whether neural networks allow to retrieve such parameters accurately, and whether some loss functions are more appropriate than others towards this purpose.

The chapter is organized as follows: in section 4.1 we explain the need for low-dose imaging and we give elements on the quantification of bone pathologies, with a focus on the context of μ-CT imaging. In section 4.2 we perform a study on the impact of different combinations of loss functions on different metrics and parameters relevant to the medical context. For this we use a post-processing method as it offers practical advantages from a computational point of view. Finally in section 4.3 we study the possibility to use a stochastic CWGAN to get statistical information on quantitative parameters from a bone sample given a FBP obtained from a low-dose acquisition.

4.1 Low-dose bone microarchitecture imaging

4.1.1 Principle and interest of low-dose CT imaging

Here we derive some medical considerations that justify the need for dose reduction in CT imaging in general. Two points need to be clarified: first, acquisition techniques of dose reduction are not the topic of our work. The methods that we discuss rather consist in enhancing the quality of images that are, in the particular application that we choose, obtained after a low-dose acquisition. This distinction needs to be done as dose reduction techniques represent a field of research itself that is out of our scope. The second point is that the elements given for dose reduction here are associated to CT imaging in general; our study then focuses on bone microarchitecture μ-CT imaging which represents a tiny part of the entire field. We will cover this particular case afterwards.

Undergoing a CT scan is linked to an increased risk of developing a cancer [Hall and Brenner, 2008]. As the use of CT scans for diagnosis became the standard of care, the exposure of patients to radiation increased drastically. In the US, population exposure to radiation due to X-ray imaging was multiplied by six between 1982 and 2006 [Mettler Jr et al., 2008]. This was mainly explained by the increase in the number of scans performed - normalization of the CT scan as a diagnosis tool - and the increase of the dose used in the scanning procedure.

The exposure to the radiation from X-rays has serious consequences on the patient's health. A whole body scan gives tissues doses in the range of 10-30 mSv [Mettler et al., 2000], which is equivalent to the maximum radiation dose recommended to nuclear plants

employees. In 2010, more than 400 patients in the US have been exposed to abnormally high level of radiation in brain CT scans [Bogdanich, 2010]. This risk is all the more so high as the patient is young; in [Mathews et al., 2013], it is shown that there was an increased incidence of cancer after CT scan exposure during childhood or adolescence, mainly due to irradiation.

Even if those long term studies were performed with data from older scanners, it clearly shows the importance of reducing the radiation dose for CT imaging, and it has been a major research interest in the last twenty years.

In practice, dose reduction might be obtained with sparse sampling - i.e obtaining less projections -, though it is not really the current standard. Indeed in most of the commercial CT scanners, the X-ray source delivers X-rays constantly, even though accurate and fast grid-switching are getting introduced to monitor the X-ray source in modern scanners [Mei et al., 2017]. The second possibility to lower the radiation dose during a CT scan is to reduce the tube current and it has been largely studied [Coursey and Frush, 2008].

As detailed in [Kubo, 2019], the current stake is mainly to find methods that are hardware-independent in order to get CT images while satisfying both the radiation dose requirements and the quality necessary for diagnosis purposes. This is a very active research field in deep learning and a lot of work has arisen to make neural networks a tool to produce quality images from a low-dose CT protocol.

4.1.2 Basics of X-ray bone imaging

As mentioned in the introduction, one of the main concerns in this chapter is to study how results on loss functions from computer vision translate to medical imaging; hence we need to delve into the relevant parameters that are used to help establish a diagnosis. We detail here elements related to bone pathologies as this corresponds to the application of our study in this chapter.

One of the main pathologies related to bones is osteoporosis, for which it is estimated that more than 200 million people are suffering from worldwide [Sözen et al., 2017]. It is a skeletal disorder that reduces the bone strength so that the risk of fracture is increased. Bone strength depends on two main features:

- Bone mineral density (BMD) which is a projected density value expressed in g/cm^2.
- Bone quality which includes many properties of bone, like trabecular bone microarchitecture, the presence of micro-fractures or other damage, etc.

The BMD is generally computed within a specific area, e.g the femoral neck, and is most commonly given as the T-score, which expressed the measured BMD (in g/cm^2) as the number of standard deviations above or below the mean BMD value - measured within the same area - for a healthy 30-year-old adult of the same sex and ethnicity as the patient. The choice to compare to 30-year-old adults is due to the fact that this is the age corresponding to the peak for bone mass. Considering the BMD criteria, osteoporosis can be defined as a T-score under -2.5. Note also that the Z-score can be used, in which case the BMD is compared with people with the same age as the patient.

BMD measurement is currently the most common method for osteoporosis diagnosis and evaluation of the risk of fracture [Oei et al., 2016]. It was shown in [Ammann and

Rizzoli, 2003] that up to 70% of the variation in bone strength can be attributed to the measured BMD. Also, each standard deviation decrease of BMD corresponds to a two-fold increase in fracture risk [Dawson-Hughes et al., 2008]. In clinical use, dual-energy X-ray absorptiometry (DXA) is the most commonly used and studied bone density measurement technology[1]. It is however also possible to estimate BMD from CT images since bone density can be retrieved for each voxel in this case.

There are limits to the BMD, as for instance a lot of fractures due to bone weakness happen while the T-score does not indicate osteoporosis. If low bone density as well as a historic of fractures are important risk factors, they cannot provide a comprehensive diagnosis tool for the prediction of future bone failure.

Especially, when BMD does not allow distinguishing between population with high-risk of fracture or not, spatial arrangement of the trabecular bone structure can be a criterion for distinction [Homminga et al., 2002]. A lot of research has also been conducted towards the link between bone microarchitecture properties and the risk of failure. For instance it has been observed that the bone volume to total volume ratio (BV/TV) is considerably reduced from osteoporotic patients compared to healthy ones [Nazarian et al., 2006]. The authors conclude that low BV/TV values is an efficient predictor for failure of cancellous bone, especially when it is computed within subregions of the bone since bone failure can occur in specific parts of the microstructure. Authors also show that microstructural indices improves the efficiency of a predictive model based on measured parameters. BV/TV ratio can also be a tool for assessing the efficiency of some drugs in placebo-controlled trials [Genant et al., 2008], which illustrates its importance.

Studying connectivity in the bone volume also allows one to get insight on the bone microarchitecture [Kabel et al., 1999]. Here again this can be a parameter to monitor for drugs trials; it is the case for the study of paired biopsies taken before and after treatment with human parathyroid hormone (PTH). It was especially shown in [Fox et al., 2005] that after 19 months of PTH treatment compared with placebo, BV/TV increased by 44% and connectivity density by 25% demonstrating the usefulness of the drug treatment to improve bone quality.

BV/TV, BMD, 3D connectivity and other parameters obtained from trabecular bone images are thus of paramount importance as illustrated by applications such as osteoporosis or risk of fracture diagnosis, but also for monitoring the response of patients in clinical trials. It is hence crucial to ensure that algorithms used for reconstructing bone microstructure correctly retrieve those different parameters.

Performing low-dose acquisitions can be a factor of degradation of the accuracy of the parameters retrieved in the reconstructed images. Significant changes in absolute BMD values were observed in [Mei et al., 2017] when dose reduction was performed by lowering tube current. Indeed, noise causes in this case an important bias in quantitative values. This study also shows that a better estimation of BMD was obtained when low-dose acquisition is performed through sparse-sampling, especially because this reduces photon noise; even if this may open up brighter perspectives for the future, it is to be noted that those results are obtained when reconstructions are performed with iterative algorithms, which are not used in clinical routine because of the computational burden mentioned in Chapter 2. This highlights the need to perform such a study for metrics obtained on

[1]DXA is a spectral imaging technique where two X-ray beams with different energy levels are aimed at the patient's bones.

deep learning based reconstructions.

4.1.3 μ-CT imaging

Before delving into our experimental study, we give here the the principle and the application range of μ-CT imaging since the data that we will use are obtained from such a modality.

μ-CT is a CT technique that provides images with spatial resolution at micrometer scale [Peyrin and Engelke, 2012]. It is an imaging system mostly used for bone samples as it is non destructive and for in vivo imaging of small animals such as mice and rats model as it is non invasive. It is however not a modality that can be used for in-vivo scanning of human tissues.

Early μ-CT used synchrotron radiation [Ebashi et al., 1991, Salome-Pateyron et al., 1997] but more recent tube-based commercial scanners have been developed, some of them being specialized for analyzing bone structure thanks to integrated softwares [Hildebrand and Rüegsegger, 1997]. In [Genant et al., 2008], it is shown that μ-CT is a relevant tool for studying bones' features of interest as it can provide accurate parameters in topological and morphological analysis.

The question on the relevance of considering low-dose imaging for μ-CT in our study needs to be addressed. First we mentioned the fact that this is an imaging technique that can be used for in vivo animal studies, so in this case lowering the radiation dose makes sense. Diagnosis-related parameters are easily evaluated in vitro currently, but non-destructive/invasive techniques for in vivo use would be even more useful and it is a field of research for the radiology of osteoporosis [Genant et al., 2008]. Therefore, we perform our study on available μ-CT, simulating low-dose acquisitions so that it can be transcribed to further in vivo acquisitions for which reducing the radiation dose will be inevitable. The fact to use ex vivo bone samples allows one to have quality ground truth data for the study's objectives, a quality that potentially could not be obtained from real patient data with reasonable radiation dose. The results that we obtain here are useful for the field of μ-CT data, but we stress the fact that the scope of this study is expected to be larger than this particular modality: we aim at getting insights on the impact that different loss functions used for training a neural network have on the parameters of interest in the case of bone related pathologies such as osteoporosis.

To sum up, here are listed the motivations of the study:

1. We focus on deep learning methods for low-dose image reconstruction since it is a need for the CT imaging field in general.

2. We want to assess whether neural networks are able to reconstruct images from which one can accurately retrieve quantitative parameters of interest for diagnosis, with the focus put on the impact of the training loss function on those relevant parameters. Indeed the loss function is the basis of the learning process so it must be in adequacy with the final purpose of the neural network which is to produce reliable images from a medical point of view.

3. We consider μ-CT bone microstructure data since the ground truth real images that are available have high quality, and the complexity of bone microstructure is

such that we expect the analysis to clearly emphasize limitations and/or benefits of different loss functions. Also diagnosis of diseases such as osteoporosis would benefit from accurate quantitative and structural parameters retrieved in the reconstructed images, which makes the scope of the study particularly useful for this application.

4.2 Study of the loss function for a post-processing method

Here we want to assess whether different loss functions, which have proven to be efficient in the computer vision field, are able or not to produce good quality reconstructions, the quality being evaluated regarding parameters that were discussed in the previous section. Indeed, two networks with identical architectures can produce very different reconstructions depending on the way they have been trained.

It is to be noted that not every loss function can be used to train a neural network. Indeed the gradient of such a function needs to be computed and backpropagated the networks' parameters. For instance, it is not feasible to consider BV/TV as a loss function. We therefore need to find loss functions that can be differentiated and, in the same time, be compatible with the desired purpose of the network.

Note also that the effect of loss functions was studied in [Kim et al., 2019] for the denoising of low dose CT images. There is however, as explained in the previous section, a need to perform a comparative study with task-specific metrics.

In this section we study combinations of pixelwise, structural and adversarial losses and evaluate the benefits and drawbacks from each of these. To do so, we consider metrics that are common to computer vision, but also ones that are relevant for the diagnosis of bone diseases. To conduct the study, we consider the simplest task that consists in enhancing the quality of a FBP image obtained from low-dose projections with a deep convolutional neural network (CNN) trained on high-dose/low-dose paired images. This work is expected to give some insight on the impact of the loss function in the context of tomographic reconstruction and provide a guide in selecting the appropriate loss function when using neural networks to reconstruct bone microarchitecture.

Most of the methods, results and analysis presented here have been published in [Leuliet et al., 2022a]. We present here all of those elements as it represents one of the major contributions of this thesis.

4.2.1 Model

Here we recall the model considered in Chapter 3. We consider $y \in Y$ the image reconstructed with FBP from low-dose projections, Y being the space of these low-dose FBP reconstructions. Let $x \in X$ be the corresponding high-dose image, where X is the target space of images obtained in the high-dose setting. The aim is to find the reconstruction operator G_θ such that

$$x = G_\theta(y) \tag{4.1}$$

where G_θ is a deep CNN parameterized by θ. Note that we talk about a reconstruction operator for simplicity here even though y does not correspond to a projection. In what

follows we consider paired high-dose and low-dose images (x, y) and a loss function L such that

$$\theta^* \in \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mu}[L(G_\theta(y), x)] \tag{4.2}$$

where μ is the joint distribution of (x, y) and parameters of G_θ are trained according to the loss function L with backpropagation. In practice, empirical expectations obtained with training data are considered.

4.2.2 Training losses

As mentioned above, we distinguish three types of loss functions. Pixelwise losses compare each pixel of the predicted image with the corresponding pixel in the ground truth and the average error is then considered. Structural losses compare statistics or features from the prediction and the ground truth in order to match the way human eye evaluates similarities between images. Adversarial losses allow to assess whether the predicted image belongs to the distribution of ground truths or not, i.e if the network is producing an image that could be reconstructed from a high-dose acquisition in our case.

4.2.2.1 Pixelwise losses

We study here two types of piwelwise losses. The mean squared error (MSE) or L_2 loss is widely used to train a neural network and is often the default loss in deep learning libraries; it writes

$$L_{\text{MSE}}(G_\theta(y), x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - [G_\theta(y)]_i)^2 \tag{4.3}$$

where n is the total number of pixels in the image and subscript i denotes pixel values of x. Another widely used loss that performs operations between pixels is the mean absolute error or L_1 loss

$$L_{\text{MAE}}(G_\theta(y), x) = \frac{1}{n}\sum_{i=1}^{n}|x_i - [G_\theta(y)]_i|. \tag{4.4}$$

In both cases, pixels are considered independently and outliers - for instance one pixel value $[G_\theta(y)]_i$ that is very far from x_i - are largely penalized. MSE might lead to oversmoothing in the reconstructions, but it is generally efficient to retrieve flat areas. For sharp objects, MAE is often preferred since less oversmoothing is observed in the solutions. This can be explained by the fact that MSE corresponds to a Gaussian statistic of the noise in the likelihood in a Baysesian framework, while MAE corresponds to a more sparse Laplace prior as mentioned in 1.1.4. Note that in both cases structural features are not taken into account so that there is no guarantee that the bone microstructure is well retrieved or even that the retrieved image is consistent with the structure of a real sample.

4.2.2.2 Structural losses

To ensure the correctness of the reconstruction in terms of anatomical structure, a solution can be to train networks with loss functions that compare images with respect

to aggregated statistics or features within each of them. SSIM was developed in [Wang et al., 2004] to measure the similarity of two images with respect to the structure rather than operating pixel by pixel. Its value ranges between -1 and 1, with a SSIM of 1 corresponding to identical images. A loss function that can be considered is thus the negative SSIM that writes

$$L_{SSIM}(G_\theta(y), x) = -\frac{(2\mu_g\mu_x + c_1)(2\sigma_{gx} + c_2)}{(\mu_g^2 + \mu_x^2 + c_1)(\sigma_g^2 + \sigma_x^2 + c_2)} \tag{4.5}$$

where μ_g is the average of $G_\theta(y)$, μ_x is the average of x, σ_g^2 and σ_x^2 are their corresponding variance, σ_{gx} is the covariance of $G_\theta(y)$ and x, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ with L the dynamic range of the pixel values that is 1 in our case - due to rescaling - and $k_1 = 0.01$ and $k_2 = 0.03$ as we considered standard values. In practice, SSIM index is computed on sliding Gaussian windows of size 11×11 with standard deviation $\sigma = 1.5$ and the actual SSIM value is the average of the local similarities.

Perceptual losses can also be computed by comparing features within the two images. Those features can be obtained by feeding a trained neural network with the images of interest and considering some layer output: this is the idea of VGG loss in [Johnson et al., 2016] that writes

$$L_{VGG}(G_\theta(x), y) = \frac{1}{n}||VGG(G_\theta(y)) - VGG(x)||_2^2 \tag{4.6}$$

where VGG is the 16^{th} output of the VGG-19 model [Simonyan and Zisserman, 2014] that performs classification of natural images. Note that the weights that we will use for computing the VGG loss are the same as those obtained from training VGG-19 on natural images in [Simonyan and Zisserman, 2014] - we do not retrain this network with different images here. It is shown in [Zhang et al., 2018] that such a loss better suits human perception compared to pixelwise losses; their study focuses on natural images and is not driven towards medical purpose.

The intuition behind both SSIM and VGG losses is that the human eye does not compare every pixel independently. Rather, it operates on different areas and assembles the information to draw a conclusion on the similarities between images. If the efficiency of such metrics is now acknowledged in the computer vision field, it is not proven for medical imaging.

Whether it is SSIM or VGG, one can still expect that using such losses should allow to retrieve relevant structures in the bone reconstruction since the network specifically learns to minimize the difference in terms of structural features during the training stage. However in SSIM the pixel values are only considered with aggregated statistics and for VGG loss there is no consideration at all given to pixel values. Those losses could thus lead to networks that correctly transcribe the bone microstructure but where the BMD correspondence is missing.

4.2.2.3 Adversarial loss

In Chapter 3 we detailed the concept of generative adversarial networks and especially adversarial losses such as the approximated Wasserstein distance in WGAN. We recall the corresponding loss function:

$$
\begin{aligned}
L_{\text{WGAN}}(D_w, G_\theta) =& \mathbb{E}_{x\sim P_x}[D_w(x)] - \mathbb{E}_{y\sim P_y}[D_w(G_\theta(y))] \\
&- \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D_w(\hat{x})||_2 - 1)^2]
\end{aligned}
\tag{4.7}
$$

where D_w is a neural network that is - jointly with G_θ - trained to maximize L_{WGAN}, P_y and P_x are the empirical distributions of respectively low-dose FBP data and high-dose images, $\hat{x} \sim P_{\hat{x}}$ are sampled along straight lines between real high-dose images and generated ones, λ is the weighting term for the gradient penalty, that we fix to 10 in this study as it is a standard value. Here the expectation is in practice obtained with samples within a single batch; contrary to previous loss functions it makes less sense to give a definition for a single pair $(G_\theta(y), x)$ - unless the batch size is 1 which will not be the case for our experiments.

Here we only consider WGAN-like networks; the aim of the discriminator is to evaluate whether the generated image belongs to the high-dose images distribution or not, but the network is not stochastic. Rather, G_θ aims to be a mapping from the distribution of low-dose images onto the one of high-dose images. Here the training loss evaluates whether the generated image belongs to X, but it does not indicate whether the content corresponds to the input low-dose FBP. As a consequence, a content loss should be added to ensure that x matches its low dose version y.

As the WGAN loss evaluates the quality of the generated image thanks to a probability distribution model, it is reasonable to think that both BMD and bone microstructure are taken into account in that case, and the impact of the adversarial loss should be studied accordingly.

4.2.3 Comparative study

We propose to combine different loss functions with weighting parameters to form a more complex cost function, with the hope to benefit from the strengths of each part. There are 31 possible combinations from the 5 losses that we presented (and we could have considered even more possibilities).

Our aim here is to assess the impact of each category of loss functions, and potentially find the most relevant one from each category. Combining losses from the same category - for instance L_1 and L_2 - is thus not interesting for us. This leaves us with 17 potential combinations, and even only 16 if we do not consider WGAN alone as we mentioned that it should be used with a content loss. We will report in this work results for 12 of these combinations - shown in Table 4.1 - as they allow to answer the questions raised in this study, the 4 other combinations adding no further insights for our problem.

Table 4.1 highlights the potential drawback of using a complex loss function: adding weighting parameters that need to be tuned during the training stage increases the computation time for a fixed hyper-parameter optimization strategy. For instance, a grid search strategy consisting in testing n different values for each hyper-parameter requires n^2 times more trainings for WGAN-SSIM-L_1 (2 weighting parameters) compared to CNN-SSIM (no weighting parameter).

Network	Loss function
CNN-L_1	$L_{\text{MAE}}(G)$
CNN-L_2	$L_{\text{MSE}}(G)$
CNN-SSIM	$L_{\text{SSIM}}(G)$
CNN-VGG	$L_{\text{VGG}}(G)$
CNN-SSIM-L_1	$L_{\text{SSIM}}(G) + \lambda_1 L_{\text{MAE}}(G)$
CNN-VGG-L_1	$L_{\text{VGG}}(G) + \lambda_1 L_{\text{MAE}}(G)$
CNN-VGG-L_2	$L_{\text{VGG}}(G) + \lambda_1 L_{\text{MSE}}(G)$
WGAN-L_1	$L_{\text{WGAN}}(D,G) + \lambda_1 L_{\text{MAE}}(G)$
WGAN-VGG	$L_{\text{WGAN}}(D,G) + \lambda_1 L_{\text{VGG}}(G)$
WGAN-SSIM-L_1	$L_{\text{WGAN}}(D,G) + \lambda_1 L_{\text{SSIM}}(G) + \lambda_2 L_{\text{MAE}}(G)$
WGAN-VGG-L_1	$L_{\text{WGAN}}(D,G) + \lambda_1 L_{\text{VGG}}(G) + \lambda_2 L_{\text{MAE}}(G)$
WGAN-VGG-L_2	$L_{\text{WGAN}}(D,G) + \lambda_1 L_{\text{VGG}}(G) + \lambda_2 L_{\text{MSE}}(G)$

Table 4.1: Tested networks and their training loss function. λ_1 and λ_2 are weighting parameters. (G) implies that the loss function is used to update the generator's weights, while (D, G) refers to loss functions that update both the generator and the discriminator.

4.2.4 Metrics

Here we detail the metrics that we use for evaluating the reconstruction performance. Once again, note that loss functions and metrics are different notions; loss functions allow the weights of the network to get updated thanks to differentiation, while a metric is a tool for assessing the quality of the final reconstruction, once the network is trained. Most of the metrics that we will use are indeed not suited to train a neural network, though they are relevant for assessing its performance.

4.2.4.1 Metrics from the computer vision field

Even if the main interest of this study is to perform the evaluation on diagnosis oriented metrics, we still present results for common metrics used in computer vision.

The Peak Signal to Noise Ratio (PSNR) is defined as

$$PSNR(f,g) = 10\log\left(\frac{L^2}{MSE(f,g)^2}\right)$$

with L the dynamic range of the pixel values and $MSE(f,g)$ is the mean squared error between images f and g. The PSNR is a measure of signal transcription in the reconstructed image compared to the ground truth; it can be a good indicator of the noise removal capacity of the reconstruction method. However the overall visual quality of the reconstruction is not evaluated with PSNR.

It is also possible to assess the quality of the reconstruction with SSIM:

$$SSIM(f,g) = -L_{SSIM}(f,g).$$

Here the structure of the image is indeed taken into account, but this is a metric that has been specifically designed for traditional computer vision applications and originally it was also designed for the evaluation of contrast and luminance of images.

4.2.4.2 Resolution assessment

In our study we found relevant to assess the ability of neural networks to reconstruct thin details in the images; indeed it does not correspond to any parameter analyzed for diagnosis but it is intuitive to think that images with better resolution are of higher interest for medical experts, so it might be a criteria in this comparative study.

The ability to reconstruct thin details can be assessed with the resolution of the obtained image. In [Banterle et al., 2013],] the authors introduced the Fourier Ring Correlation (FRC) as a metric to estimate the resolution of a reconstruction. The idea is to compute the correlation between an estimated 2D image f with respect to some ground truth g in the Fourier domain as

$$FRC_{f,g}(R_i) = \frac{\sum_{r \in C(R_i)} |\Re(\hat{f}^*(r)\hat{g}(r))|}{\sqrt{\sum_{r \in C(R_i)} |\hat{f}(r)|^2 \sum_{r \in C(R_i)} |\hat{g}(r)|^2}} \tag{4.8}$$

where R_i is the radius of the ring $C(R_i)$ in the Fourier domain within which the correlation is computed, $\hat{f}$ is the Fourier transform of f, $\hat{f}^*$ denotes the conjugate of $\hat{f}$, and $\Re$ denotes the real part. The metric aims at measuring the ability of the reconstruction to recover information at a certain frequency level. The resolution ρ of the reconstruction can then be determined as

$$\rho = \frac{1}{R_{FRC(R) \leq \tau(R)}} \tag{4.9}$$

where $R_{FRC(R) \leq \tau(R)}$ is the radius for which the FRC is lower than a threshold τ. This threshold may depend on the radius and in [Banterle et al., 2013] it is computed as

$$\tau(R) = \frac{2}{\sqrt{\frac{N_p(R)}{2}}} \tag{4.10}$$

with R the radius in the Fourier domain and $N_p(R)$ the number of pixels contained within the corresponding ring.

4.2.4.3 Diagnosis oriented metrics

Bone mineral density transcription The BMD, which is one important element to assess bones weakness in the diagnosis of osteoporosis, can be estimated from the Hounsfield Units (HU) values of the CT image. An accurate reconstruction should then correctly retrieve the values of density per voxel in HU; indeed if the BMD is computed as the overall density within a precise area, one needs to ensure that the density value for all voxels from this area is reliable. To this purpose, we study the flattened HU distribution of the voxels that are reconstructed for each method. Quantitative analysis of the differences in terms of voxel values can be performed by computing the Wasserstein-1 distance - see [Ramdas et al., 2017] - between the 1D distributions obtained when considering each

voxel of the volume as one realization of a random variable. Indeed we find this approach more thorough compared to directly computing the BMD which is the overall density within specific areas; in that case we could get accurate mean values without having the right density distribution across voxels, therefore a good BMD correspondence could be obtained "by chance". The Wasserstein-1 distance is more appropriated to this purpose and it writes in one dimension

$$W_1(\phi_1, \phi_2) = \inf_{\pi \in \Gamma(\phi_1, \phi_2)} \int_{\mathbb{R}\times\mathbb{R}} |u - v| d\pi(u, v) \tag{4.11}$$

where ϕ_1 and ϕ_2 are the two considered 1D distributions and Γ is the set of joint distributions (ϕ_1, ϕ_2). Note here that this distance does not correspond to the one that is approximated with neural networks. The latter considers distributions over n-dimensional vectors, n being the number of pixels in an image. Rather in (4.11), u and v are the distributions obtained when taking n realizations of a 1-dimensional random variable, which are drawn from the distributions of the voxels taken in either the ground truth or the estimated volume. The role of such a metric is here to assess that the density per voxel distribution across the volume is accurate; a low distance between reconstructions and ground truths data make the BMD measurement reliable. Also, using the Wasserstein 1-distance instead of MAE allows not penalizing reconstructions with accurate density per voxel - thus accurate BMD - but with a structure that is slighlty shifted compared to the ground truth for instance. We compute this Wasserstein-1 distance between HU distributions with the *Stats* module of *Scipy* library in Python.

BV/TV The ratio between the bone volume and the total volume (BV/TV) is a key information for mechanical failure prediction. Computation of this metric is performed on images that have been segmented to distinguish between areas corresponding to bones and the rest of the image. For this, we post-process the reconstructions with Otsu segmentation [Otsu, 1979] and simply compute the fraction of bone volume on this segmentation.

Connectivity Studying connectivity in the bone volume allows one to get insight on the bone microarchitecture. Connectivity can be determined in an unbiased manner by the Euler number. We evaluate it to assess the fidelity of the reconstruction in terms of structure. In actual medical settings, this is performed considering the 3D volume but since in our study the networks are built for 2D slices, we focus on the comparison for the 2D Euler number. Indeed there might be inconsistencies in the third dimension as it is not considered in the training process; those inconsistencies would largely affect the compuation of the 3D connectivity, which is why we stick to 2D Euler number for the sake of the study. In the 2D case, computation of this number amounts to counting the difference between the number of objects and the number of holes that are perceived in the image obtained after segmentation. We show results considering 4 neighboring pixels for the objects counts (4-Connectivity), but similar results are observed with 8 neighboring pixels (8-Connectivity). Computation of the Euler number is performed with the *measure* module of *scikit-image* library in Python.

Note that we also compute the DICE value on the obtained segmentations; it might help further assess the accuracy of the reconstructions in terms of bone volume fraction, though this is a metric that is computed at the pixel level. It writes

Metrics	Interest
Computer vision based	
PSNR	Signal retrieval and noise removal
SSIM	Structure of the image
Diagnosis oriented	
1D Wasserstein distance	BMD accuracy
Euler number	Bone connectivity
BV/TV	Bone fraction
Others	
Resolution (FRC)	Thin details restitution
DICE	Pixelwise accuracy of the segmentation

Table 4.2: Metrics used during the study and their interest.

$$DICE(f_s, g_s) = 2\frac{|I(f_s, g_s)|}{|f_s| + |g_s|}$$

where $I(f_s, g_s)$ is the intersection of segmentations f_s and g_s, an $|f_s|$ - resp. $|g_s|$, $|I(f_s, g_s)|$ - is the number of 1 in the 0/1 image f_s - resp. g_s, $I(f_s, g_s)$.

Finally all of the metrics that we consider for evaluating the quality of the reconstructions depending on the loss function combinations are represented in Table 4.2.

4.2.5 Data

The ground truth data[1] consist of 3D volumes of human radius and tibia structures obtained on a SCANCO μ-CT 100 with a 24-μm voxel size. The training dataset is composed of ten volumes from different human donors. Two volumes from two other subjects are considered for evaluating the methods; the networks are not trained with those two volumes and the hyperparameters are not tuned according to these data. These two evaluation volumes have respectively a number of slices, height and width of $164 \times 882 \times 752$ and $194 \times 466 \times 372$ voxels. The ground truth training data are illustrated in Figure 4.1 and the volumes for evaluation are illustrated in Figure 4.2.

Denoting by ρ the ground truth volume, projections $p(\rho)$ were computed with the parallel Radon transform from these volumes. This was performed with ASTRA Toolbox [van Aarle et al., 2015] in Python. To simulate low dose data, we first consider 400 projections corresponding to approximately 50 % of the total number of projections in the high-dose setting. We consider a source intensity I_0 of 10000 photons per detector pixel, and simulate the received intensity I at each detector pixel as $I = \text{Poisson}(\frac{I_0}{K} e^{-p(\rho)})$, with K a parameter that we vary to simulate different amounts of dose, similarly to [Leuschner et al., 2020]. For instance, $K = 10$ corresponds to 5% of the dose, since

[1]Data used in this study are the courtesy of Andrew Burghard from University of California, San Francisco, USA.

we already consider half of the projections. Then, the noisy projections are taken as $\tilde{p} = \ln \frac{I_0}{KI} + n$ with n an additive zero-mean Gaussian noise with standard deviation 1% of the first term mean value. Finally, we compute the FBP of $\tilde{p}$ with a Hann filter (cutoff 0.4) and consider it as the noisy input data of the network. In the training data, we varied K to simulate between 5% and 50% of the upper limit of the radiation dose. Note that we do not clip the value below 0 here to preserve information in the input of the neural network. When evaluating the FBP we do not clip the values either, though it could be done if required. Then data are normalized when fed into neural networks. The normalization is simply performed by dividing the images by a factor $\rho_{\max}$ which was chosen so that all data lie between -1 and 1 for the input FBP, and between 0 and 1 for the ground truth.

4.2.6 Experiments

In all models, the generator is a 16-layer Convolutional Neural Network (CNN) with 128 filters in each layer, except for the last layer which has only one filter since the output is the generated image. Worse performance was observed with fewer layers in preliminary tests. The considered deep CNN is similar to the one used in [Yang et al., 2018]. We could have used more sophisticated methods here such as unrolled iterative schemes, or even UNET for the post-processing method but this does not bring any additional value for the purpose of the study; as we had a simple 16-layer CNN working efficiently, we kept this simple architecture for the whole study.

For WGAN based networks, we use the same discriminator structure as in [Yang et al., 2018]. For both the discriminator and the generator, Leaky ReLU activations are used with parameter 0.3 and He initialization [He et al., 2015], except for the output of the discriminator that has no activation function. Zero padding is applied for every layer. Optimization is performed with Adam algorithm [Kingma and Ba, 2014] with $\beta_1 = 0.9$, $\beta_2 = 0.999$. Training is ran on 7,000 steps, which approximately corresponds to 3 epochs. The gradient weighting parameter λ is fixed to a default value of 10 as in [Gulrajani et al.,

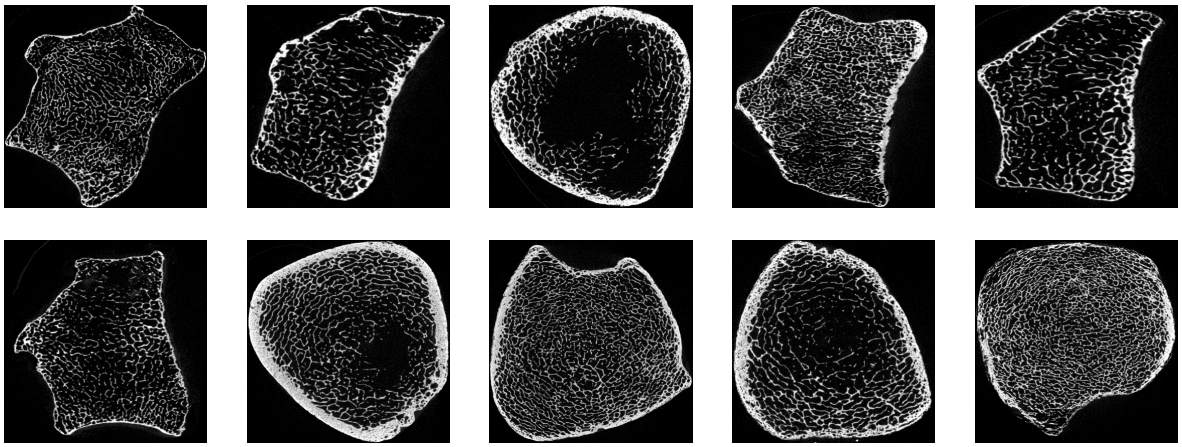


Figure 4.1: Slices of 3D volumes used for training. Each of the 10 volumes has between 152 and 248 slices, whose size ranges from 628×508 to 1068×928 voxels. Window size is $[-1000, 3000]$ HU.

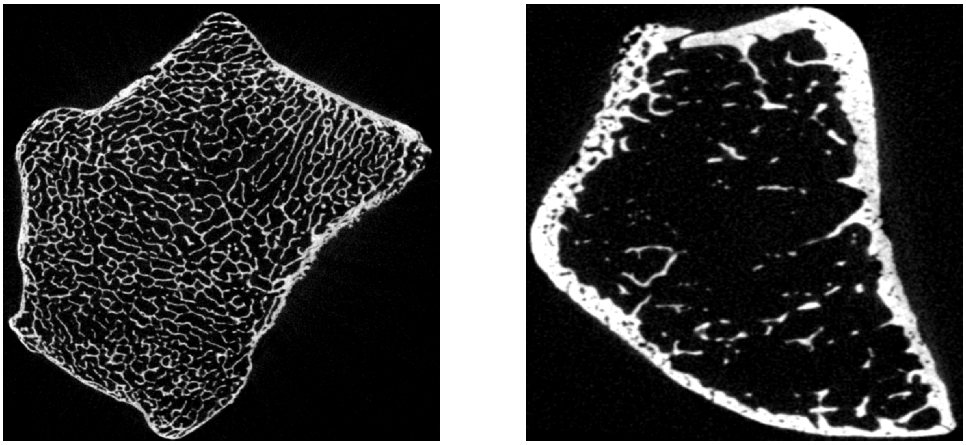


Figure 4.2: Volume 1 (left) and Volume 2 (right) used for the evaluation. Their number of slices, height and width are respectively $164 \times 882 \times 796$ and $194 \times 466 \times 372$ voxels. Window size is $[-1000, 3000]$ HU.

2017].

For a fair comparison, hyperparameters (HP) namely the kernel size, batch size, learning rate - initial and final since we use exponential decay -, number of generator updates, λ_1 and λ_2 are all optimized for every single network, on a validation set that is obtained by taking 20% of the slices from the 10 training volumes. Two stages of HP optimization are performed, the second stage allowing to zoom in the range of HP that gives the best validation PSNR. The same strategy is used for each network. Results show that for all networks, the optimal kernel size is 3×3, compared to 5×5 and 7×7. We find that 4 generator updates for 1 discriminator update is the best choice for WGAN based networks, as we tested ratios between 0.2 and 5 between both number of updates. The best batch size is 128, i.e the maximum size that could fit in the memory of the resources that were used for the study. The reason for these HP to be similar for every loss function is that those hyperparameters mostly depend on both the training data and the general structure of the networks which is considered as fixed. We only find differences in the optimal HPs for the learning rate, λ_1 and λ_2 since these HPs specifically depend on the loss function. The optimal values for those HPs are represented in Table 4.3. We tested learning rates between 10^{-8}, for which we observed very slow decrease of the loss function, and 10^{-2}, for which we observed divergence of the loss function. As for the weighting parameters we tested values between 10^{-3} and 10^{3}.

Note that we assessed over-fitting by monitoring the PSNR on the validation set with the considered HP. Figure 4.3 shows the example of CNN-L_2 and WGAN-VGG, where one can observe that with the number of steps considered, the validation curve does not decrease and it stabilizes, so not over-fitting has been observed with the corresponding training configuration. It has been checked for every algorithm and conclusions remain the same. Note that we tracked PSNR only here since other metrics such as the Euler number would be way too long to compute to make it affordable during training.

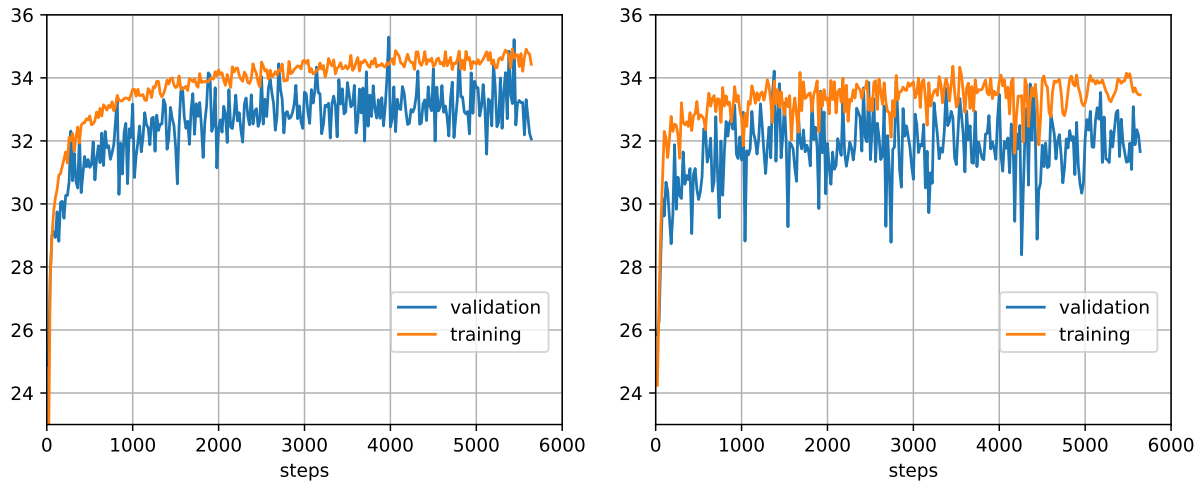
Once the hyperparameters optimal values have been found, final training is performed on 64x64 patches from all 1,992 different 2D slices for a total of 297,976 patches. Figure

Network	lr_i	lr_f	λ_1	λ_2
CNN-L_1	6×10^{-4}	8×10^{-6}	-	-
CNN-L_2	$1,5 \times 10^{-4}$	8×10^{-6}	-	-
CNN-SSIM	$1,5 \times 10^{-4}$	8×10^{-6}	-	-
CNN-VGG	10^{-3}	3×10^{-5}	-	-
CNN-SSIM-L_1	$1,5 \times 10^{-4}$	$1,5 \times 10^{-5}$	100	-
CNN-VGG-L_1	$1,5 \times 10^{-4}$	$1,5 \times 10^{-5}$	50	-
CNN-VGG-L_2	$1,5 \times 10^{-4}$	$1,5 \times 10^{-5}$	100	-
WGAN-L_1	$1,5 \times 10^{-4}$	$1,5 \times 10^{-5}$	1000	-
WGAN-VGG	$1,5 \times 10^{-4}$	8×10^{-6}	20	-
WGAN-SSIM-L_1	$1,5 \times 10^{-4}$	$1,5 \times 10^{-5}$	1	500
WGAN-VGG-L_1	$1,5 \times 10^{-4}$	8×10^{-6}	10	50
WGAN-VGG-L_2	$1,5 \times 10^{-4}$	8×10^{-6}	20	50

Table 4.3: Optimal hyperparameters for each method. These hyperparameters have been optimized on a validation set consisting of 20% of the slices obtained from the 10 training volumes. The learning rate decreases exponentially from lr_i to lr_f during training.

4.4 shows the evolution of the PSNR on the training set during this final training step for some of the tested algorithms (we do not represent all for clarity).

Computations are performed on a NVIDIA Tesla V100 GPU. The generator has slightly more than 2×10^6 trainable parameters, the discriminator has around 18×10^6 trainable parameters, and VGG loss implies 20×10^6 extra parameters that are not train-



(a) CNN-L_2 (b) WGAN-VGG

Figure 4.3: Over-fitting assessment. Training and validation PSNR are represented w.r.t the number of steps when training on 80 % of the training set and validation on the other 20 %. Results are shown for (a) CNN-L_2 and (b) WGAN-VGG

able but that still need to fit into the memory. Training of a CNN takes approximately 2 hours per epoch, and around 10 hours for WGAN-based networks since one training step consists of 5 updates: 4 for the generator and one for the discriminator. Tests show that this difference in terms of computation time cannot be avoided since convergence of WGAN based networks still require the same number of epochs as CNNs.

Note also that we could use different learning rates for the discriminator and the generator, choose other architectures, different values of α_{LEAKY} for different activations and so on, so at the end there is a multitude of hyperparameters combinations that could be tested.

Here we test only a few hyperparameters, most of the others are standard. For others (number of layers generator etc), our choice was based on preliminary tests and we took values that gave satisfying results. The most important thing is that we considered the same strategy for the different algorithms we want to compare.

As for the architecture we could have taken another generator such as UNET or even unrolled iterative networks; in this case the performance might be better for every algorithm, but our aim is to study the impact of the loss functions, so we need to fix the networks' architecture.

4.2.7 Evaluation

For evaluation, we simulate 2 configurations: 10% and 20% of the maximum dose. Note that we control the dose amount, which is not equivalent to controlling the amount of Poisson noise since the latter depends on the density of the volume: there is more attenuation and thus more Poisson noise for more dense volumes.

In what follows, we study PSNR, SSIM, resolution (Resol), Wasserstein distance for the 1D distribution within the whole volume (WV) and within the bone area (WB). In the segmented reconstructions we study DICE, BV/TV, mean absolute Euler number

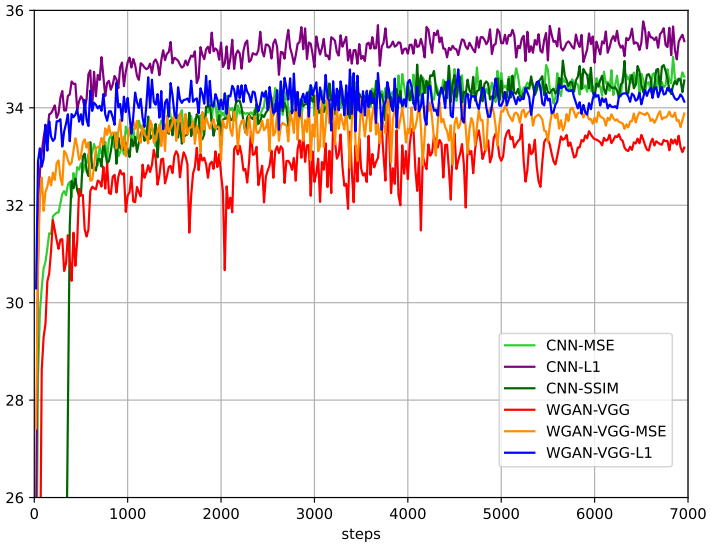


Figure 4.4: Evolution of the PSNR for final training (100% of the training data are used here) for some of the tested algorithms. CNN-MSE refers to CNN-L_2.

difference compared to the ground truth (E-N) as well as the relative absolute difference of object counts (O-C).

For the evaluation we do not post-process the outputs of the network. This could be done otherwise - e.g clipping the values below 0-, but the point is that we want to evaluate the performance of the algorithms and not hide some of their components by post-processing data.

4.2.8 Results

4.2.8.1 Pixelwise loss function study

Table 4.4 reports all tested metrics for different configurations, with the emphasis put on comparing the presence of L_1, L_2 or no pixelwise loss function in the overall cost function. Metrics are given for the evaluation volume 1 for 10% dose but results are similar for 20% and for the other volume. By comparing each row from every block, one can observe that for resolution, L_1 loss improves the performance compared to using no pixelwise loss or using L_2 loss. Also, using no pixelwise loss function with CNN-VGG leads to a significant performance drop for most of the metrics. We notice that the BV/TV ratio is higher for L_2 loss. The differences between each method are slight when looking at PSNR, SSIM or DICE. This observation can also be made on the other test volumes and for different dose configurations, thus we cannot use those metrics to distinguish between the performance of each loss function. As for connectivity and metrics involving the Wasserstein-1 distance, different performance can be observed depending on the loss function, but it is not related to the pixelwise loss according to this table.

Table 4.5 highlights the enhancement of resolution with L_1 loss. Indeed, the most performing networks are those who have the mean absolute error as part of the loss function, with the exception of WGAN-VGG-L_1 that ranks behind CNN-SSIM, but this

WGAN-VGG	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
$\emptyset$	28.91	0.811	0.848	0.140	29 ± 20	0.07 ± 0.05	98.6 ± 6.7	21.62	43.81
L_1	**29.94**	**0.842**	**0.864**	**0.140**	24 ± 17	0.07 ± 0.05	$\mathbf{86.6 \pm 6.0}$	13.61	**19.93**
L_2	29.63	0.829	0.859	0.141	$\mathbf{23 \pm 18}$	$\mathbf{0.06 \pm 0.04}$	93.2 ± 5.9	**10.53**	31.20
CNN-VGG	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
$\emptyset$	26.87	0.128	0.858	0.140	40 ± 28	$\mathbf{0.20 \pm 0.11}$	98.3 ± 4.4	332.57	581.71
L_1	**30.43**	0.846	**0.866**	**0.140**	$\mathbf{37 \pm 27}$	0.21 ± 0.05	$\mathbf{77.9 \pm 4.4}$	**27.75**	**106.36**
L_2	30.19	**0.851**	0.858	0.147	79 ± 40	0.29 ± 0.05	94.1 ± 4.6	41.55	209.64
CNN-SSIM	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
$\emptyset$	**30.36**	**0.871**	0.865	0.141	44 ± 29	0.23 ± 0.05	83.1 ± 4.1	**29.45**	**103.6**
L_1	30.35	0.859	**0.865**	**0.139**	$\mathbf{26 \pm 20}$	$\mathbf{0.16 \pm 0.05}$	$\mathbf{77.3 \pm 5.3}$	30.47	109.56
L_2	30.27	0.863	0.859	0.148	63 ± 37	0.28 ± 0.05	91.1 ± 5.1	39.22	215.16
CNN	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
L_1	**30.43**	0.848	**0.866**	**0.140**	$\mathbf{33 \pm 26}$	$\mathbf{0.19 \pm 0.05}$	$\mathbf{78.7 \pm 5.0}$	**26.24**	**107.08**
L_2	30.17	**0.852**	0.856	0.148	37 ± 31	0.20 ± 0.05	95.4 ± 5.1	44.48	240.55

Table 4.4: Metrics for volume 1 and 10% dose. Here we study the influence of the pixelwise loss. Bold entries in the first column indicate the part of the loss function that is fixed. BV/TV ratio for ground truth is 0.138.

will be discussed in what follows with the impact of the structural loss.

Figure 4.5 illustrates the increased performance of L_1 loss on CNN-SSIM and WGAN-VGG examples. The figure shows the evolution of FRC value with respect to the maximum frequency value for which the correlation is considered - the ring radius. Since the correlation is computed on 2D slices, we selected a slice on volume 1 to display the curves for the different methods on 10% dose. One can observe that for high frequencies, the L_1 curve is above the other curves. For CNN-SSIM, improvement with L_1 is observed between 9 mm^{-1} and 11 mm^{-1} (high frequencies), whereas for WGAN-VGG it is already observed with lower frequencies, around 6 mm^{-1}. In both cases, this indicates that the L_1 loss better transcribes high frequencies which allows reconstructing finer details.

4.2.8.2 Structural loss function study

By performing a similar ablation study to investigate the impact of the structural part of the loss - SSIM vs VGG - we find that VGG is more efficient when associated to WGAN and SSIM with CNN, i.e with no adversarial loss. In the same way as for pixelwise loss functions, not all of the metrics allow to clearly distinguish between WGAN-VGG and CNN-SSIM based networks; this is however the case for resolution and connectivity related metrics. Table 4.6 shows the mean and standard deviation of the difference between the Euler number of both the predicted slices and the ground truth ones. As the Euler number computes the difference between the number of objects and the number of holes estimated in the image, the other column represents the relative difference for the object count only. This allows one to ensure that the observed performance for the Euler number metric is not biased by the fact that both the count of holes and objects are not correct. Results clearly show that WGAN-VGG outperforms CNN-SSIM, independently

	Volume 1		Volume 2	
Method	10%	20%	10%	20%
WGAN-L_1	**75.1 ± 5.1**	72.4 ± 4.3	**74.8 ± 6.2**	**73.5 ± 5.1**
WGAN-SSIM-L_1	75.4 ± 4.6	**72.3 ± 4.0**	75.3 ± 5.9	74.0 ± 5.5
CNN-SSIM-L_1	77.3 ± 5.3	73.9 ± 4.3	76.4 ± 5.3	75.3 ± 5.1
CNN-L_1	78.7 ± 5.0	75.5 ± 3.9	76.2 ± 6.7	74.7 ± 5.7
CNN-VGG-L_1	77.9 ± 4.4	75.4 ± 4.3	76.9 ± 6.4	75.6 ± 6.0
CNN-SSIM	83.1 ± 4.1	80.1 ± 4.0	79.3 ± 5.6	78.6 ± 6.2
WGAN-VGG-L_1	86.6 ± 6.0	83.1 ± 5.3	82.9 ± 7.2	79.7 ± 7.1
WGAN-VGG-L_2	93.2 ± 5.9	87.8 ± 4.9	85.2 ± 7.8	82.3 ± 7.0
CNN-VGG-L_2	94.1 ± 4.7	90.9 ± 4.7	90.4 ± 6.8	90.5 ± 6.7
CNN-L_2	95.4 ± 5.1	91.9 ± 5.2	88.8 ± 8.1	87.2 ± 7.8
WGAN-VGG	98.6 ± 6.7	93.6 ± 6.7	94.3 ± 10.8	93.1 ± 10.8
CNN-VGG	98.3 ± 4.4	95.3 ± 4.1	91.2 ± 6.7	90.3 ± 6.7

Table 4.5: Values of resolution in μm for each method and for the test volumes 1 and 2 considering 10% and 20% dose for both.

from the pixelwise loss function that is potentially associated. Connectivity is thus better represented with WGAN-VGG according to our results. We also notice that using no structural loss decreases the performance in terms of connectivity : on volume 1, the error in terms of objects count is more than 10 % higher for CNN-L_1 compared to WGAN-VGG networks, and on volume 2 the Euler number mean absolute difference is between 2 and 3 times larger for CNN-L_1. Nevertheless, it can be observed in Table 4.4 or 4.5 that VGG loss - especially when associated with WGAN - leads to a higher value for the resolution which means a reduced ability to transcribe high frequencies.

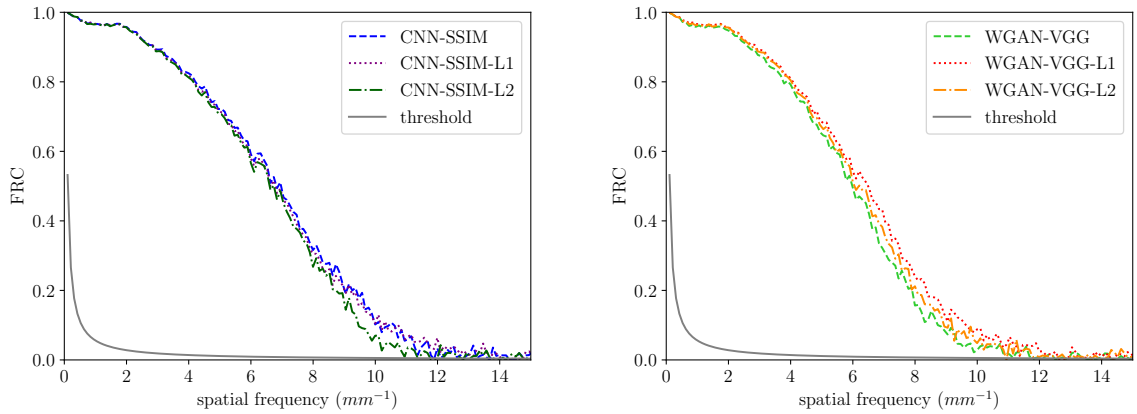


Figure 4.5: FRC curve on a selected slice on volume 1 for different reconstruction methods, for 10% dose. The y-axis represents the Fourier Ring Correlation value between 0 and 1, the x-axis is the radius of the ring in the Fourier domain within which the correlation is computed. The threshold to compute the resolution according to (4.10) is also represented.

	Volume 1				Volume 2			
	10%		20%		10%		20%	
Method	E-N	Obj. c	E-N	Obj. c	E-N	Obj. c	E-N	Obj. c
WGAN-VGG-L_2	**23 ± 18**	**0.06 ± 0.04**	29 ± 23	0.08 ± 0.05	**6 ± 5**	0.08 ± 0.06	**6 ± 4**	0.07 ± 0.06
WGAN-VGG-L_1	24 ± 17	0.07 ± 0.05	**25 ± 17**	**0.05 ± 0.04**	6 ± 5	0.12 ± 0.09	6 ± 5	0.09 ± 0.07
WGAN-VGG	29 ± 20	0.07 ± 0.05	34 ± 31	0.11 ± 0.06	7 ± 5	0.13 ± 0.08	6 ± 4	0.13 ± 0.08
CNN-SSIM-L_1	26 ± 20	0.16 ± 0.05	35 ± 25	0.18 ± 0.04	16 ± 8	0.08 ± 0.06	16 ± 7	0.07 ± 0.06
CNN-L_1	33 ± 26	0.19 ± 0.05	39 ± 28	0.20 ± 0.04	15 ± 7	0.08 ± 0.06	16 ± 7	0.07 ± 0.05
WGAN-L_1	39 ± 25	0.20 ± 0.05	43 ± 26	0.19 ± 0.04	10 ± 7	0.08 ± 0.06	9 ± 6	0.07 ± 0.05
CNN-VGG-L_1	37 ± 27	0.21 ± 0.05	42 ± 28	0.21 ± 0.04	15 ± 8	0.09 ± 0.06	12 ± 7	0.23 ± 0.08
CNN-L_2	37 ± 31	0.20 ± 0.05	58 ± 37	0.26 ± 0.05	11 ± 7	0.32 ± 0.09	12 ± 8	0.30 ± 0.09
CNN-SSIM	44 ± 29	0.23 ± 0.05	56 ± 31	0.24 ± 0.04	9 ± 6	0.10 ± 0.07	8 ± 6	0.09 ± 0.06
WGAN-SSIM-L_1	55 ± 26	0.22 ± 0.05	67 ± 28	0.23 ± 0.04	8 ± 6	**0.07 ± 0.05**	7 ± 5	**0.06 ± 0.05**
CNN-VGG-L_2	79 ± 40	0.29 ± 0.05	91 ± 43	0.32 ± 0.05	10 ± 7	0.25 ± 0.09	12 ± 7	0.23 ± 0.08

Table 4.6: Connectivity metrics : Euler number absolute difference and objects count relative difference compared to ground truth for each method and for test volumes 1 and 2 with 10% and 20% dose.

4.2.8.3 Adversarial loss function study

Finally, we isolate the impact of the presence or not of an adversarial loss in the cost function. Table 4.7 shows improved performance of WGAN based network when focusing on the W_1 distance between the 1D distributions, especially when focusing on the distribution within the bone area, which is represented in the second column. Note that this is not straightforward since the Wasserstein distance that is used for training the networks is not the same as the one used as a metric (n-dimensional vs 1-dimensional). This results suggests that the adversarial loss helps retrieve the correct BMD in the volumes, which is of significant importance for assessing the bone strength.

In order to illustrate those results, Figure 4.6a shows the HU histograms for each method, including the ground truth, for the volume 1 on 10% dose. Figure 4.6b allows a better visualization since the graph only represents the 1D distributions when considering the areas that are segmented as bones with the Otsu threshold.

Figure 4.7 shows reconstructions with the methods that gave the best performance when taking all metrics into account, and especially resolution, connectivity and W_1 distance.

4.2.9 Analysis and conclusion of the study

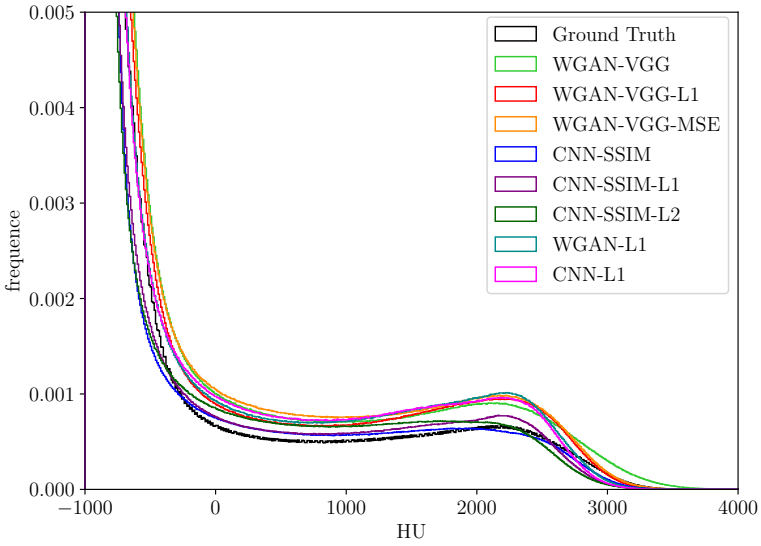
In our study PSNR, SSIM and DICE did not allow to distinguish between pixelwise, structural and adversarial losses. It is an argument to encourage future studies to evaluate methods regarding task-specific metrics since they allow to do so according to our

	Volume 1				Volume 2			
	10%		20%		10%		20%	
VGG	WV	WB	WV	WB	WV	WB	WV	WB
CNN	332.57	581.71	319.16	499.60	350.40	703.88	347.73	653.04
WGAN	21.62	43.81	28.97	98.68	35.27	85.67	40.59	78.73
L_1	WV	WB	WV	WB	WV	WB	WV	WB
CNN	26.24	107.08	22.04	41.43	44.20	161.46	34.55	117.62
WGAN	22.75	59.95	22.11	22.13	39.49	142.54	30.40	88.64
VGG-L_1	WV	WB	WV	WB	WV	WB	WV	WB
CNN	27.75	106.36	20.72	30.38	43.02	148.86	33.42	103.89
WGAN	13.61	19.93	23.22	65.17	43.68	126.14	39.45	75.97
SSIM-L_1	WV	WB	WV	WB	WV	WB	WV	WB
CNN	30.47	109.56	21.95	29.49	46.01	125.38	38.30	73.50
WGAN	17.55	32.62	24.50	43.50	35.18	106.09	27.17	54.11

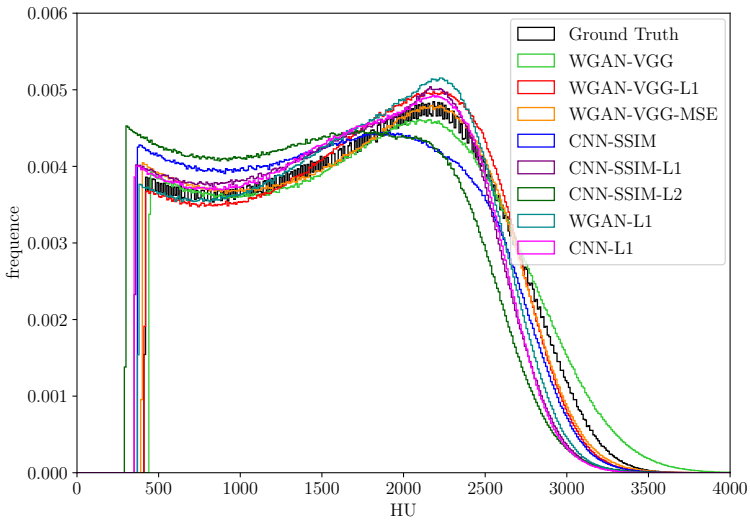
Table 4.7: Wasserstein 1 distance for the 1D distributions in the entire volume (WV) and in areas considered as bone (WB) by the segmentation algorithm. Here we study the influence of the adversarial loss on those metrics. Bold entries in the first column indicate the part of the loss function that is fixed.

experiments.

Our results clearly suggest that pixelwise loss functions have a major role in the obtained resolution in the reconstructions. We showed that the L_1 loss is the most suited one for the task of reconstructing images with the best resolution. Moreover, the choice of



(a) Entire volume distribution

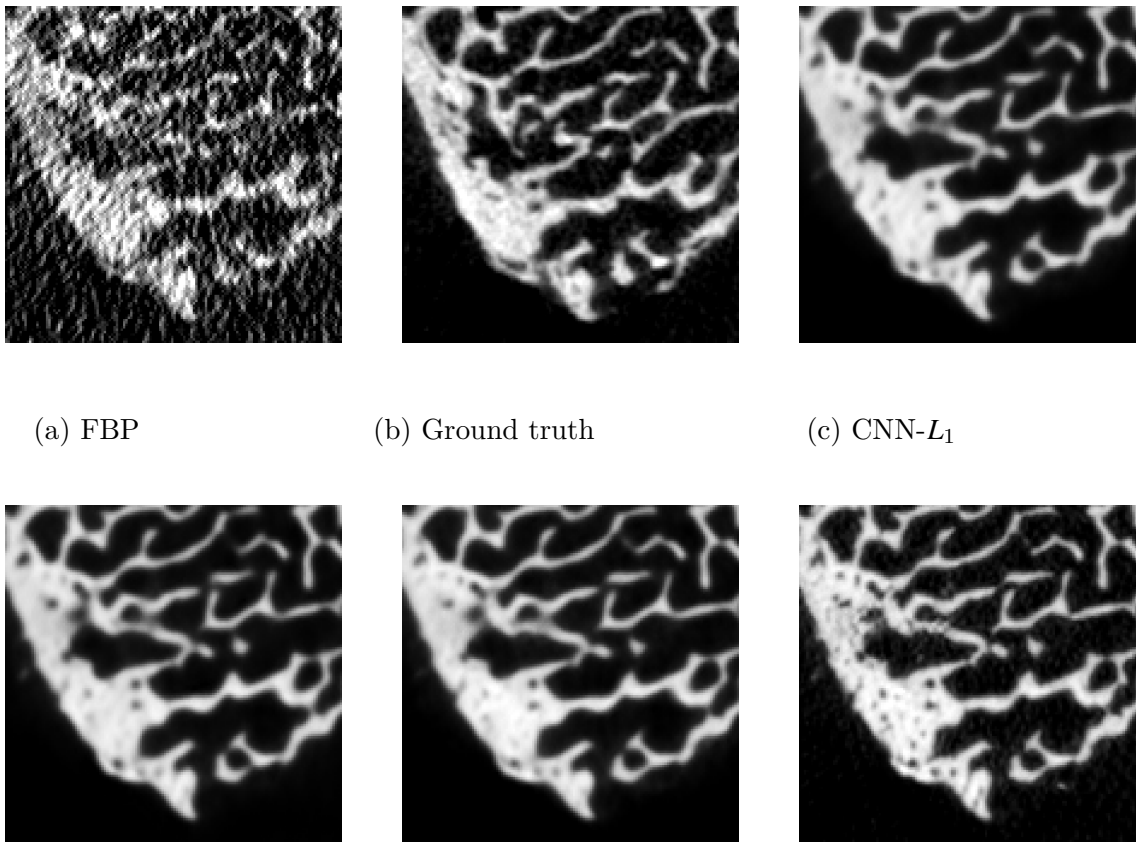


(b) Bone area distribution

Figure 4.6: Histogram of the HU distribution for each method on the volume 1 for 10% dose. Frequencies for lower densities are hidden in 4.6a to put the emphasis on the more dense areas, knowing that the bone ratio in the entire volume is around 15 %. In 4.6b, only densities above the segmentation threshold are considered in the distributions to focus on the bone area.

L_1 loss has no negative impact on all the other metrics that were tested. We also showed that L_2 loss, besides deteriorating the resolution, tends to increase the BV/TV ratio in the segmented reconstruction. This can be explained by the common tendency of such a loss to oversmooth images, which encourages the segmentation algorithm to consider bone areas wider than they really are. Our results therefore strongly suggest L_1 to be considered as part of the loss function for its ability to improve resolution performance without decreasing the quality of the reconstruction considering other metrics.

As for the structural loss, experiments show that using VGG loss alone implies a significant drop in performance for quantitative metrics. This is due to the fact that VGG network was trained to perform classification on natural images; only the structures are helpful for VGG to perform this task. The pixel intensities are not considered as relevant features for this purpose. The satisfying performance of CNN-VGG in terms of connectivity metrics, DICE and BV/TV compared to its poor performance in terms of quantitative metrics such as PSNR or Wasserstein distance is a perfect example that highlights the



(a) FBP (b) Ground truth (c) CNN-L_1

(d) CNN-SSIM-L_1 (e) WGAN-L_1 (f) WGAN-VGG-L_1

Figure 4.7: ROI from volume 1 of size 140×140 voxels obtained with different methods. Window size is $[-1000, 3000]$ HU. FBP is obtained after simulation of 10% of the normal dose to obtain the projections from the ground truth. Networks are fed with this FBP as input.

need for a loss function to contain elements that take both structure and pixel values into account. Following this observation, we find enhanced overall performance when VGG is associated to an adversarial loss. WGAN-VGG networks showed to significantly improve performance in terms of connectivity in the reconstructions. This however results in a decreased performance in terms of resolution, even with L_1 loss. SSIM does not present this drawback in terms of resolution, but on the other side it only shows limited improvement in terms of connectivity. The positive impact of the structural loss on connectivity metrics is thus more significant for WGAN-VGG than for CNN-SSIM, but it appears to induce a less optimal resolution. We suggest that depending on the application, the trade-off could be dealt with by tuning λ_1 and λ_2 accordingly while using a network like WGAN-VGG-L_1.

The last point of interest is the presence or not of an adversarial loss in the cost function. We showed in the previous point that when associated with VGG, the adversarial loss has a positive impact on the connectivity metrics. This can be understood by the fact that learning the probability distribution of high-dose reconstructions helps capture the anatomically correct shapes in the bone microstructure. We also observe better accuracy in terms of density per voxel distribution. As WGAN-based networks try to learn the n-dimensional distribution of high-dose images, n being the total number of pixels, it is reasonable to think that such networks are more likely to retrieve the 1-dimensional distribution of density values since it can be induced by the knowledge of the former. Since it is required to have reliable BMD in order to diagnose various bone-related diseases, our results strongly suggest using such a loss for training neural networks for bones imaging.

Another aspect to consider when making the choice of a loss function is the computation time and memory requirement for training the networks. In our case, the reconstruction time during inference is the same for all methods since we use a similar generator for all of them. However, we mentioned that using VGG loss increases the memory consumption. This can for instance reduce the maximum batch size to use for training compared to other methods and potentially decrease performance, even if this is not an issue that we experienced in our tests. Also, using the adversarial loss increases the training time by a factor of 5 in our experiments, which also needs to be taken into account when considering the improvement brought by such a loss for BMD accuracy. Finally, when considering a loss function composed of different parts, this adds extra hyperparameters to tune during the training phase.

The fact of increasing the training time or the number of hyperparameters might have a negative impact on the final performance of the network. Indeed, for a fixed computational budget, the number of hyperparameters that can be tested to validate the performance of the networks can be significantly reduced for complex loss functions and particularly those which require an adversarial loss. Our study does not put the emphasis on such constraints, and it is not clear whether the focus should rather be put on spending time finding optimal hyperparameters with a simpler loss function or not. Analyzing the results considering the computational cost of the loss function could therefore be the subject of another study.

Also, our hyperparameters selection was performed by choosing PSNR as the validation metric since it is common practice and we did not want results to be biased towards a stronger importance given to a specific metric. Our results however suggest that the choice of those hyperparameters should be driven by task-dependent metrics

since we noticed that they do not necessarily match common evaluation metrics such as PSNR or SSIM. For practical purpose, the metrics used for optimizing the network's hyperparameters need to be carefully chosen.

Our work mainly focuses on the study of the reconstruction depending on the loss used for training the network. One needs to keep in mind that the method still relies on applying a CNN on the low-dose FBP as input; there is no guarantee that such a method is optimal. It can be expected that better results should be observed when considering a different architecture like it has been detailed in Chapter 3. 3D networks and use of a U-NET generator could for instance be solutions to improve performance. We make the assumption that these architectures would benefit to every training scheme without modifying the comparative results that we obtained: the loss function indeed aims at assessing the quality of a prediction, independently from the way this prediction has been generated. In any case, our results allow to understand the impact of different types of loss functions when reconstructing bone microarchitecture with deep learning based methods and remain of interest even if more complex networks are used for practical application.

Comparisons between algorithms hold as long as networks are able to correctly reconstruct images, which is no longer the case when the initial FBP is too deteriorated; in that case all networks fail to retrieve an accurate reconstruction, which is due to the limits of the reconstruction method itself and not to the choice of the loss functions.

Another point is that this study only focuses on bone microarchitecture reconstruction. Finding in terms of comparative results are therefore specific to this particular application. It is to be noted that the better performance that is observed with L_1 loss compared to L_2 loss for instance is dependent on the structures one wants to reconstruct; these negative findings for L_2 may well not carry through other tasks for images with i.e. ground truths of smooth edges.

As a conclusion, the assessment of the quality of the reconstruction of bone microstructures seems to be insufficient when only considering PSNR and SSIM. Instead, relevant features such as the BV/TV ratio, Euler number to study connectivity, Wasserstein distance between the HU distributions of bones densities to study BMD transcription, are among metrics that allow the evaluation of the retrieval of key parameters used for post-reconstruction diagnosis. We showed that the loss function used to train a neural network has a major influence on those metrics; hence this should be a major concern when designing a neural network for medical imaging tasks. Pixelwise loss functions were found to improve the resolution observed in the reconstructions, with L_1 loss being the most effective in our tests. Structural loss functions play a role on the ability of networks to retrieve bone structures as shown by connectivity metrics, and VGG loss improves performance in that sense, at the cost of a deteriorated resolution. Adding an adversarial loss leads to reconstructions with more accuracy in terms of BMD. When choosing the most suited loss function for the particular task of reconstructing bone microstructure with accurate BMD values, one needs to keep in mind the trade-off between the computational cost of complex losses and the improved performance that they bring.

Finally, improvements of this study could include a larger database for evaluation to get results that are statistically more significant; availability of such a database for bone microstructure is however not guaranteed. Also, more combinations and more loss functions exist and could nourish the analysis; our main point here was however to show

that there is indeed a major impact of the loss function on the reliability of relevant parameters in the reconstructed images so that it should be a main criteria when designing a neural network for medical purpose.

4.3 Stochastic Conditional Wasserstein GAN to reconstruct bone microarchitecture

In this section we mention tests that we performed in order to make use of conditional generative adversarial networks towards a medical purpose. Our aim here is to learn a posterior distribution given a deteriorated bone sample; instead of having a single high quality image as the output of the network, we want to be able to perform posterior sampling and thus approximate the distribution of true bone images that could be at the origin of the retrieved reconstruction. The interest of such a method is to be able to get statistical information on the different parameters of interest for bone diagnosis that we derived throughout this chapter. For instance, here is a scenario that could have a medical interest:

1. Learn a conditional probability distribution given some low-dose acquisition (or the corresponding FBP).
2. Sample n images from this learned distribution.
3. Compute the T-score for each of these samples.
4. Get the mean T-score as well as intervals of confidence for its true value.

With such a method one could have more or less confidence on the retrieved metric, which could therefore help for the related diagnosis.

Our experimental protocol was split into two parts:

1. Evaluate the performance of a CWGAN as a reconstruction algorithm. For this we consider a reconstruction as the mean image from the learned posterior distribution.
2. Check the reliability of statistical information retrieved from posterior sampling.

Main results of the first part were published in [Leuliet et al., 2021b]; in this work we combined both CWGAN with a VGG perceptual loss for image content matching. We present in this section a summary of the work performed for this aim. Unfortunately the second part did not give exploitable results and we present here some explanations.

4.3.1 Principle

In [Adler and Öktem, 2018], authors propose a conditional Wasserstein GAN (CWGAN) to capture the probability distribution of some volume conditionally to the FBP obtained from low-dose projections. We detailed such a method in Chapter 3.

In this work we combine this CWGAN with VGG perceptual loss; indeed the aim is to provide the network with a content loss so that it should preserve bone microstructure

information. The resulting CWGAN-VGG network also learns a probability distribution conditionally to the FBP obtained from low-dose projections. Our objective is to use this distribution in order to exploit statistical information from quantitative parameters useful for diagnosis.

The preliminary study [Leuliet et al., 2021b] has two objectives; first we want to evaluate the performance of a reconstruction obtained by averaging the outputs of the stochastic generator. We also evaluate two main features of such a network: the impact of conditioning the network and the interest of stochasticity, i.e the ability of the network to produce outputs with enough variance so that statistical information can be exploited.

Both [Isola et al., 2017] and [Adler and Öktem, 2018] pointed out the difficulties of CWGAN to generate stochasticity, as the network tends to ignore the input noise. In our tests, we also implement a deterministic CWGAN-VGG (Det-CWGAN-VGG) that only learns a Dirac distribution, for comparison. This is equivalent to fixing the latent variable z that the generator takes as an input. In practice we simply ignore z so that the input of the network is the FBP alone.

We also implement a second version of the CWGAN's discrimnator as in (3.17), which should allow to produce variability in the generated samples as explained in [Adler and Öktem, 2018].

Therefore the different algorithms of comparison were in this study:

- CNN-L_2
- WGAN-VGG
- CWGAN
- Det-CWGAN
- CWGAN D_2
- CWGAN-VGG
- Det-CWGAN-VGG
- CWGAN-VGG D_2

where Det-CWGAN is the deterministic version of CWGAN, and CWGAN D_2 and CWGAN-VGG D_2 refer to networks trained with (3.17).

4.3.2 Materials and methods

In this study we use the same dataset as in the previous section, with slightly different simulations performed for training the networks as shown in [Leuliet et al., 2021b]. The trained networks are here evaluated with PSNR, SSIM, DICE and BV/TV. For comparing the different algorithms we need to consider a single reconstruction. Since CWGAN and CWGAN-VGG (and their second version) produce stochastic outputs, we average each voxel of 10 generated outputs to produce the volume for evaluation. In our tests, increasing this number does not improve the performance. Study of the statistical information obtained from posterior sampling will be discussed afterwards. We use a strategy to optimize hyperparameters that is the same as in the previous study.

4.3.3 Results

In our preliminary results, we find - as in the previous section - that it is efficient to use a perceptual loss for training the generator; indeed CWGAN-VGG outperforms CWGAN for the different metrics, whether it is on the deterministic or stochastic version.

We find that conditioning the discriminator enhances the performance of the network. This is the case in our tests, where CWGAN-VGG (both versions) produces better results than WGAN-VGG for 3 out of the 4 tested metrics, and the DICE index for both methods is very close.

Deterministic networks give similar results compared to CWGAN (and CWGAN-VGG) trained with (3.16). In the contrary we found that the strategy of averaging several stochastic outputs from CWGAN D_2 and CWGAN-VGG D_2 gives an improvement compared to using a deterministic network. The apparent reason is that CWGAN and CWGAN-VGG do not manage to generate high variance. In Figure 4.8 that represents the variance images obtained by sampling 100 outputs of each stochastic generator, one can observe that the second formulation (3.17) of the discriminator gives predictions with higher variance compared to the original formulation.

This, at first, may lead to think that the conditional distribution is well retrieved since uncertainties on the reconstruction are highlighted.

However in Figure 4.9 we observe the hidden drawback behind both the enhanced performance (for averaged outputs) and higher variability in the outputs of CWGAN-VGG D_2. The curves represent PSNR w.r.t the number of outputs that are averaged from the stochastic generator. For D_2 networks, we observe that when a certain number of outputs are averaged - around 5 - the performance significantly increases and it outperforms the original CWGAN or CWGAN-VGG. For the latter, there is no significant variation in the metrics especially because we showed that there is only limited variability in the outputs that are generated. These curves allow to understand why the performance of CWGAN-VGG D_2 in the results presented in [Leuliet et al., 2021b] is indeed interesting, since averaging 10 outputs voxel-wise allows enhancing the quality of the mean image.

The problem though is that the quality of a single generated image is very low compared to other algorithms as seen in Figure 4.9. Note that we have repeated this evaluation several times so that we ensure that poor performance from a single generated image is not a statistical fluke. To illustrate, we show in Figure 4.10 examples of generated samples from CWGAN-VGG D_2. Compared to the ground-truth in Figure 4.7, one can observe that the averaged image is visually accurate, though inconsistencies can be observed in each of the generated samples.

With the discriminator D_2, the information given by a single generated output can not be reliable; the learned probability distribution is very likely to be incorrect, even if averaging the voxels from each sample from this distribution gives an image with satisfying quality. If we consider our example scenario that aims at getting statistical information on the T-score, none of the stochastic networks would provide a reliable information. Indeed, with the first formulation of CWGAN or CWGAN-VGG, only limited variability is produced in the outputs, so that very few statistical information can be obtained compared to a deterministic network. On the other hand, computing metrics on a single generated output of CWGAN D_2 or CWGAN-VGG D_2 is irrelevant due to the poor-quality of the generated sample.

To illustrate this, we cannot use BMD as this is a metric that is linear w.r.t values of voxels so averaged BMD values would likely be accurate. We do not use the T-score as we do not have the reference BMD required for its computation. Therefore we illustrate the poor accuracy of the statistical information given by CWGAN-VGG D_2 with the BV/TV ratio which is not linear w.r.t voxels values. In Figure 4.11 we show the histogram representing the BV/TV ratio computed on 100 different images sampled from the posterior distribution of CWGAN-VGG D_2. Compared to the ground-truth, all BV/TV ratios are underestimated. Using the learned posterior distribution therefore does not allow to get reliable statistical information on a non-linear metric such as BV/TV.

As a consequence, the only conclusion that we were able to draw in this study is the fact that we managed to improve the quality of a reconstruction when we considered an average image taken from 10 different outputs of CWGAN-VGG D_2. This results has limited benefits since this implies increasing the computation time by a factor of 10. This

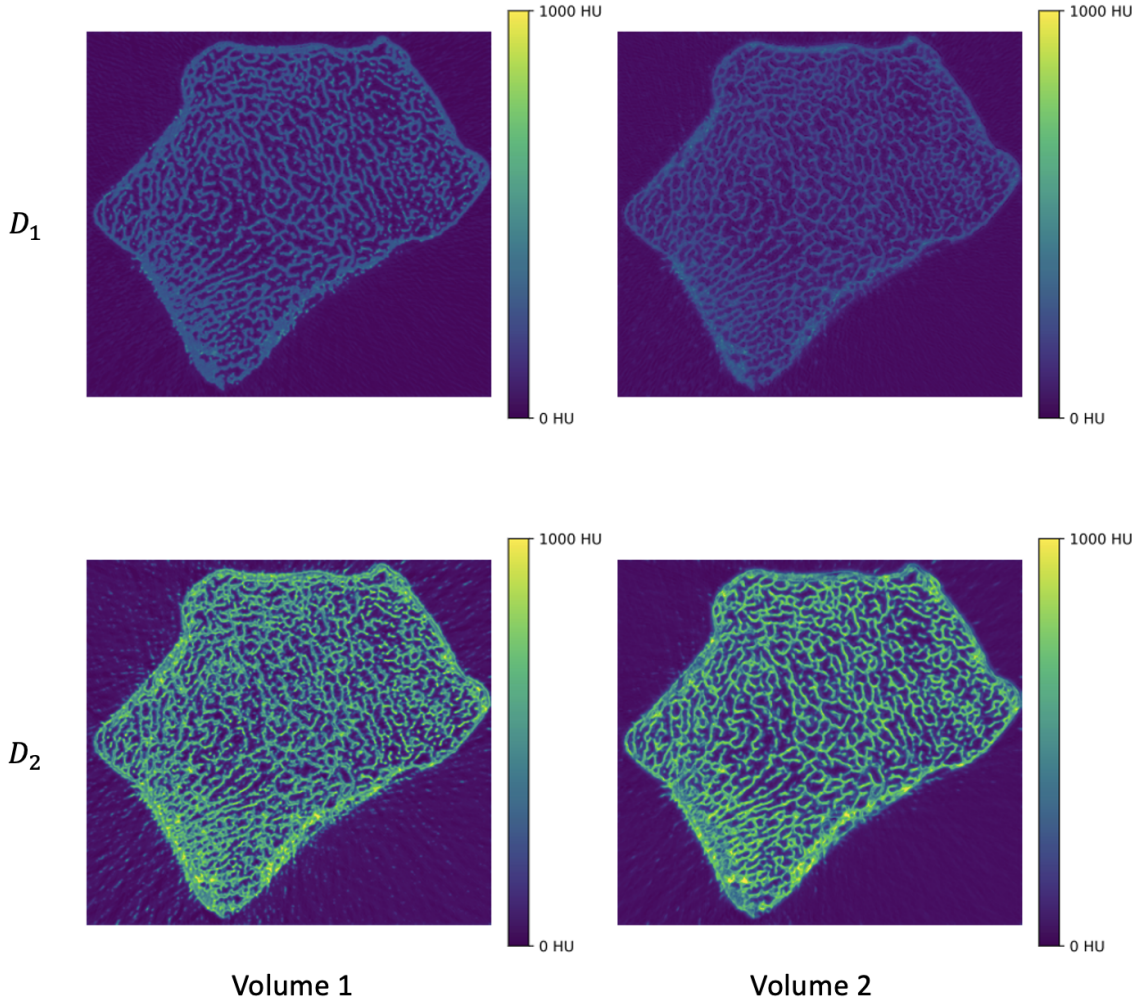


Figure 4.8: Variance in the posterior distribution in Hounsfield Units. Computation of the variance is performed by sampling 100 outputs of the stochastic generator. Results are shown for networks trained with (3.16) (CWGAN and CWGAN-VGG, top row) and with (3.17)(CWGAN D_2 and CWGAN-VGG D_2, bottom row).

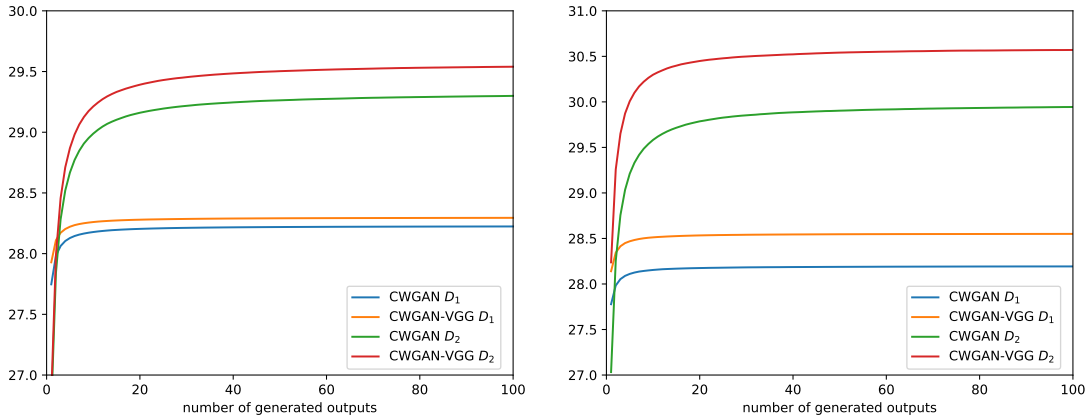


Figure 4.9: Evolution of PSNR depending on the number of outputs that are averaged from the stochastic generator on volume 1 (left) and 2 (right) for 10% dose. CWGAN D_1 refers to the initial CWGAN.

might be a drawback in practice especially when dealing with large 3D volumes. The initial objective to get statistical guarantees on parameters computed from the different

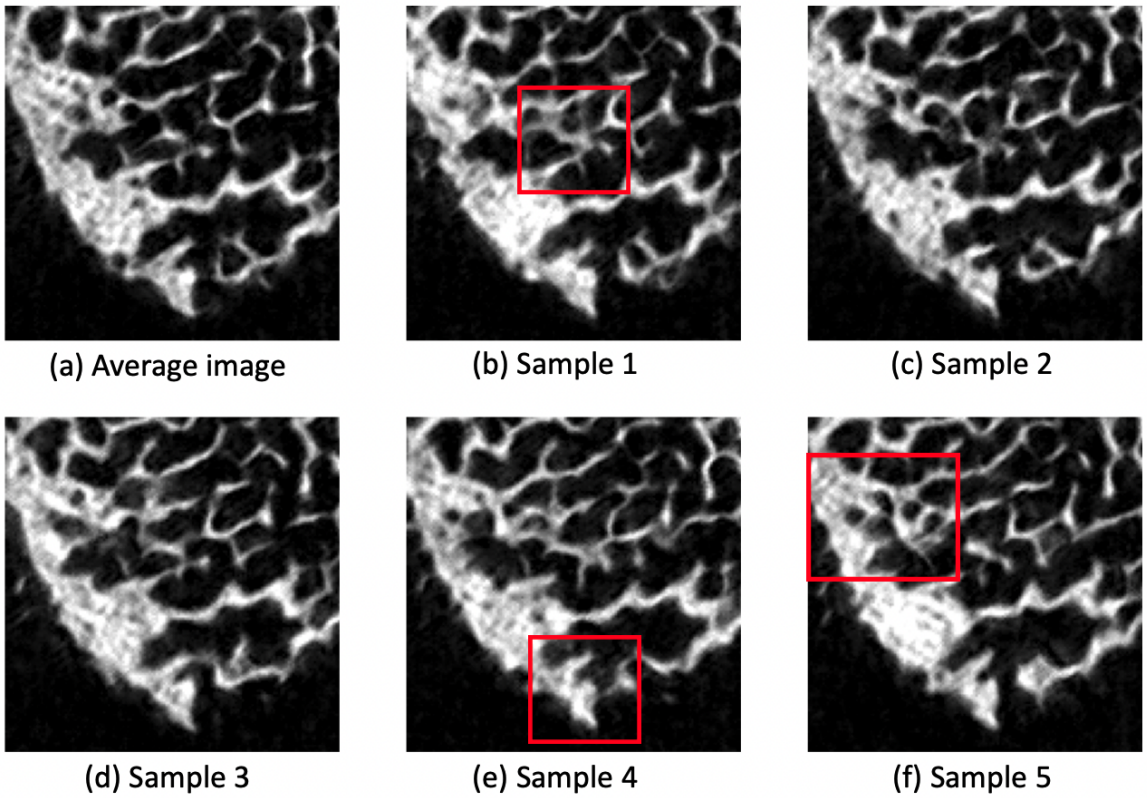


Figure 4.10: Examples of ROI from generated outputs by CWGAN-VGG D_2 in (b), (c), (d), (e) and (f). In (a) we represent the voxel-wise averaged image. In red we show areas where inconsistencies can be observed in the generated samples.

reconstructions can therefore not be validated with the experiments that we performed here.

It is still unclear why exactly do D_2 networks perform so well on the computed metrics when we average a number of outputs which, when considered separately, are of poor-quality. We conjecture that smoothing due to averaging might be a part of the explanation, though we have no evidence for that.

4.3.4 Conclusion

Even if the initial objectives of this study were not fulfilled, we can draw several conclusions:

- Averaging a number $n \geq 5$ of samples from a CWGAN-VGG trained with (3.17) (CWGAN-VGG D_2) improves the tested metrics; though it is hard to fully understand the reason for such a positive result since the quality of each of the produced output is poor.
- On the tested metrics, both conditioning the discriminator and using the VGG loss in the generator gave benefits.

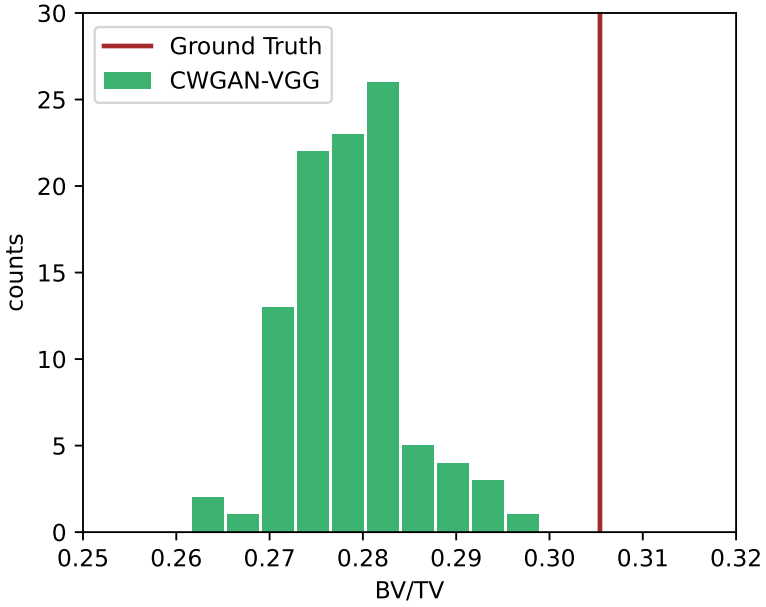


Figure 4.11: Histogram of BV/TV ratios for different samples of CWGAN-VGG D_2. The BV/TV ratio is computed on the ROI of the volume 1 for reconstruction from 10% dose. The ground-truth value is 0.306 (the value is different than in Table 4.4 because the zone is restricted to the ROI. One can observe that every single generated sample under-estimates the BV/TV ratio. However we showed in [Leuliet et al., 2021b] that the voxel-wise averaged image correctly estimated the BV/TV ratio. This illustrates the nuance between good averaging performance and poor-quality estimation for every single sample.

- It is clear that, as suggested in [Adler and Öktem, 2018], more variability in the outputs can be obtained when training the network with (3.17) compared to (3.16).
- The main issue found in this study is the poor quality of each of the generated outputs from CWGAN(-VGG) D_2 which does not allow to obtain reliable statistical information on parameters such as T-scores, BV/TV etc.

Chapter 5

PET imaging and blind deconvolution

In this chapter and in the next one, the focus is put on PET imaging. Especially, experiments in the next chapter are performed on time-of-flight (TOF) PET data. This particular application leads us to consider a more specific forward model in the inverse problem to solve. In this chapter we detail the elements from physics that justify the need for a more accurate - yet more complex to solve - formulation of the inverse problem for TOF PET imaging. The notion of point spread function is explained and we discuss several ways to consider the forward model. As this leads to a forward operator that is partly unknown, we will see that solving the related inverse problem includes a blind deconvolution step. Similarly to Chapters 2 and 3, we describe methods for solving such an inverse problem with analytical and iterative methods, as well as with data-driven methods in the last section of this chapter.

5.1 Partial volume effect and point spread function in PET systems

The purpose of this section is to lay the basis for the inverse problem formulation that we consider from now on. Physics of PET imaging was described in Chapter 1 but we focus here on a particular aspect: the partial volume effect. We will see that this effect implies that when imaging a point source in PET systems, the reconstructed image does not show a point also. Rather, a spread can be observed, which justifies the term of point spread function (PSF). We will also see in this section how the shape of the PSF can be partly determined by TOF reconstruction characteristics.

5.1.1 Partial volume effect in PET imaging

We mentioned in Chapter 1 the need for correcting physical factors in PET imaging such as photon attenuation, Compton scattering or random coincidences. As we are interested in an accurate transcription of the imaged tissues' activity, the limited spatial resolution of PET imagers should also be taken into consideration. The effects of such limited spatial resolution are referred to as partial volume effects (PVE); those can depend on

both the imaging system and the imaged object, as it is described in [Erlandsson et al., 2012].

There are two main types of PVE, the main one being the spill-over of counts, also known as cross-contamination, between different image regions. This refers to the fact that the activity retrieved in one particular voxel is contaminated by the true activity of adjacent voxels because of the limited resolution of the system. In the same time the true activity of the same voxel affects the retrieved activity in adjacent voxels. Also, the location of the radioactive decay at the origin of a detected event might not be exactly in the LOR. Reasons for such spill-over of counts include:

- Positron range: the positron covers a small distance before being annihilated by an electron.
- Photon acollinearity: photons from the annihilation may not be emitted in exactly opposite directions.
- Detector resolution which is limited for non position-sensitive electronics due to finite crystal size.
- Volume discretization since a voxel size larger than the crystal size can enhance the effect of limited resolution.
- Inter-crystal scatter.
- Parallax or depth of interaction (DOI) effect which corresponds to photons that are detected at a certain depth in the detector, thus creating a distance between the true annihilation position and the LOR.
- Limited angle effect for some PET imagers.

Figure 5.1 illustrates the notions of positron range and photon accolinearity. PVE therefore depends on both the system and the imaged object. The fact that it depends on the object will be at the core of the methods derived in the next chapter. PVE results in quantitative bias especially when imaging small objects, so partial volume correction

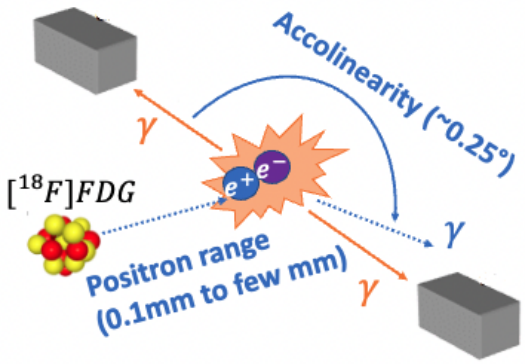


Figure 5.1: Positron range and photon accolinearity in PET imaging.

(PVC) is all the more so needed in emission tomography involving tracer uptakes in small structures. For instance, this effect can be a major issue when the activity concentration needs to be quantified in one specific target region, e.g. grey matter tissue in neurology, or a tumour in oncology. In this case adjacent voxels are considered as background regions and their potential activity should not affect the target region retrieved activity.

The second type of PVE corresponds to the fact that because of image discretization, a single voxel might contain two or more different tissue types (grey matter and white matter, tumourous and normal tissue...). This is referred to as the tissue-fraction effect.

Depending on the type of PVE that is considered, methods for correction might differ. In [[Tohka and Reilhac, 2008], it is argued that cross-contamination can be corrected based on the PET image whereas tissue-fraction correction requires anatomical information. When the image reconstruction process includes PVE correction, a common consequence is the so-called ringing (or ring) artefacts that can be visible at the edges of some structures, see Figure 5.2. Study of such ring artefacts due to PVE correction was performed in [Tong et al., 2011] and [Thielemans et al., 2010], as it represents a major issue in PET imaging.

Note also that we focus here on PVE linked to spatial factors but it can also be due to temporal factors in the case of motion blurring.

5.1.2 Point spread function

We saw that the positron emission at the origin of the detected count might not be located exactly in the LOR between the two coincidence detection crystals. If this phenomenon

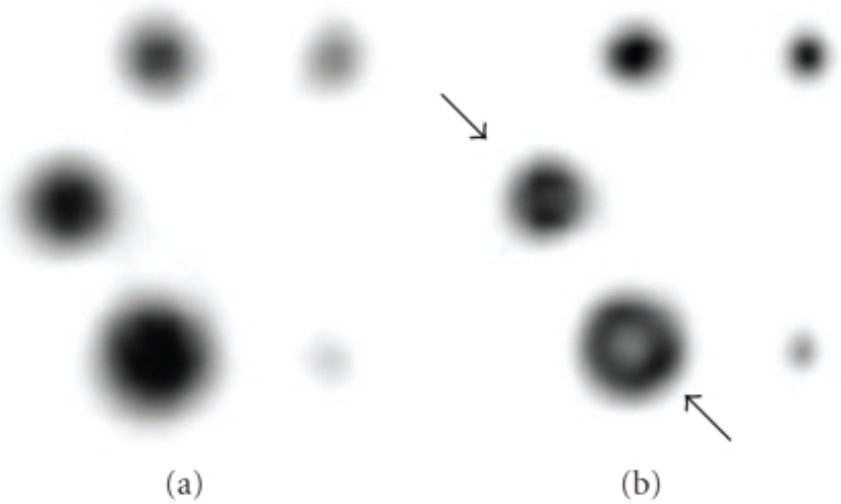


Figure 5.2: Example of Gibbs-like ringing artefacts [Kangasmaa et al., 2011], i.e ring artefacts. Image (a) represents a reconstructed slice of a phantom with active spheres without PVE correction. Image (b) shows the same slice reconstructed with PVE correction. In image (b) the resolution is improved but typical artefacts can be seen as pointed by the arrows.

is not taken into account during the reconstruction process, a blur can be observed in the reconstructed image when a point source is considered as shown for example in Figure 5.2a. This blur corresponds to the point spread function (PSF), and it characterizes the spatial resolution of the imaging system. The notion of blur kernel can also be used since the reconstruction of an image $\hat{f}$ can in this case be modeled as the convolution of the true image f and this blur kernel k such as

$$\hat{f} = k * f \tag{5.1}$$

where k is the convolution kernel of the PSF.

The PSF can also be expressed in the Fourier domain as the modulation transfer function (MTF). The MTF contains the same information as the PSF; it can be of particular interest from a computational point of view for instance, as the convolution with a PSF in the spatial domain is equivalent to the multiplication with the MTF in the frequency domain. The fact that convolution between the true image with the PSF is equivalent to multiplying the Fourier transform of the image with the MTF helps understand the loss of high-frequency information in reconstruction: indeed the MTF at high frequencies is essentially zero. Restoring these image components might lead to image artefacts and/or noise-amplification for those high frequencies.

The shape and width of the PSF are directly linked to the elements that have been listed before (positron range, DOI effect, etc). Estimation of the PSF might be non-trivial in some cases, yet mismatch between the estimated and the true PSF can be a major reason for ringing artefacts as shown in [Tong et al., 2011].

Especially, accurate estimation of the PSF should take into account the position of the source. Response of the detector - i.e the distribution of values around the position of the point source - might indeed vary depending on the position of the source in the field of view, which leads to a spatially-variant PSF. In many situations however, it can be reasonable to assume that the PSF is position-invariant [Erlandsson et al., 2012, Panin et al., 2006, Sureau et al., 2008, Alessio et al., 2010]. In this case, the reconstructed PET image can be described as a convolution of the true activity distribution with the PSF as in (5.1). It is also common to model the PSF as a Gaussian function, characterized by its full width at half maximum (FWHM), which can be different in different spatial directions. Alternative models describing the PSF can also be considered, e.g. [Taschereau et al., 2011]. Note that the fact to approximate the PSF as being position-invariant has computational advantages as we will describe in the section dedicated to solving the inverse problem.

There are cases where the geometry of the PET imaging system is such that it is no longer relevant to approximate the PSF as position-invariant. This is the case in [Gravel et al., 2019] where experiments are performed on a dual-panel breast PET imager. This type of imaging system has the advantage to offer increased sensitivity, in addition to being cost-effective, but limited angle geometry as well as the depth of interaction effects that are too important make the PSF highly deformed and spatially variable. Especially the PSF is strongly anisotropic and one can observe asymmetric kernel deformations in the direction that is orthogonal to the detectors.

In what follows we will clearly indicate when the PSF is considered as spatially-variant; otherwise it is considered - or at least approximated - as spatially-invariant.

5.1.3 Forward model and blind deconvolution

So far we saw that partial volume effect implies the consideration of the PSF in the inverse problem formulation. Indeed the model considered in Chapter 1 as $p = Af$ - with p the projection data, A the Radon operator and f the image - is unsufficient to explain the fact that a point source is not reconstructed accurately without taking the PSF into account. Equation (5.1) gives a first approach to model the underlying problem in the image space. This implies that some algorithm is used to transform measurements into an initial reconstruction. The question on how to model the whole tomographic reconstruction problem is largely studied in the literature and we detail here different ways to model the whole PET image reconstruction problem.

In [Gravel et al., 2019], the general model is expressed as

$$p = NK_{\text{data}}BAK_{\text{img}}f \tag{5.2}$$

with K_{img} an operator that contains the potentially spatially-variant degradation effects in image space (positron range, photon acollinearity), K_{data} contains the degradation effects in data space (DOI, inter-crystal scatter), B contains the attenuation factors and N the normalization factors.

In [Sureau et al., 2008] the considered inverse problem rather writes

$$p = BNAK_{\text{img}}f + s + r \tag{5.3}$$

where s and r are respectively the scatter and random coincidences.

In our work we will not focus on scatter, random, attenuation and normalization corrections, even though in practice this needs to be considered. We rather put the focus on methods to correct for the PSF so in the model that we derive we will no longer use N, B, s or r.

Considering all this, we can write the general forward model that we are interested in as:

$$p = K_{\text{data}}AK_{\text{img}}f \tag{5.4}$$

for which (5.3) is a particular case where K_{data} is the identity.

The question of considering resolution modeling (i.e consideration of the PSF in the model) in the data space, in the image space or both therefore needs to be addressed. A major criteria mentioned in [Gravel et al., 2019] is the fact that resolution modeling in the data space is complex and computationally demanding. This is one of the reasons why in their work they end up considering image-based resolution modeling only. Also, the choice of the model might be based on the fact that K_{img} and K_{data} do not account for the same physicals factors as explained earlier.

In [Tohka and Reilhac, 2008], the PSF is a 3D Gaussian kernel in the image domain; resolution modeling in the image domain is also considered in [Sureau et al., 2008], as well as in [Erlandsson et al., 2012] where the model is reduced to a simple convolution such as in Equation (5.1). Especially in [Sureau et al., 2008] an other mentioned advantage of including the PSF in the image space is to facilitate list-mode reconstruction in dynamic studies. Nevertheless in [Thielemans et al., 2010] and [Tong et al., 2011] the PSF is considered in the projection domain.

More generally in the literature one can find different modelizations of the problem where resolution modeling is considered either in the image domain or in the projection domain, depending on the application. In addition to computational and acquisition physics elements (e.g DOI vs positron range), the fact to consider a spatially-variant PSF might be a criteria for the choice of resolution modeling. This is one of the reasons why the PSF is modeled in the image domain in [Gravel et al., 2019], as it is easier to consider spatially-variant kernels in this case. Authors consider for K_{img} several PSF kernels that are parametrized by fitting each point source using a Gaussian mixture model. For this, they simulate separate point sources equally spaced and reconstruct each of them separately; then each PSF kernel is parametrized in image space by fitting the reconstructed point source with the gaussian mixture model, so that the PSF parameters depend on the source location.

We see that solving the inverse problem in the case of PET imaging is not only a tomographic reconstruction problem, but also a deconvolution problem. As the PSF is dependent on the imaged object and on physical elements that are hard to estimate accurately, the convolution operators K_{data} and K_{img} are unknown, contrary to the operator A. The problem is therefore referred to as blind deconvolution. For some applications K_{data} and K_{img} can be parametrized so that only parameters are unknown; in this case the term semi-blind deconvolution is more appropriate.

In what follows, we will discuss the impact of TOF in the formulation of the PSF in the general model (5.4).

5.1.4 Time of flight and impact on PSF

Time-of-Flight (TOF) PET systems[1] refer to imaging systems that are able to measure the time difference between the detection of two photons that form a coincidence event. The information on the annihilation position is therefore more accurate; instead of only knowing the LOR where the photon annihilation took place, such systems allow to know the location of such an event in the LOR, with an accuracy that depends on the timing resolution. In the particular case of limited-angle data, TOF can for instance provide new information by reducing the missing cone region in frequency domain [Li et al., 2016]. A scheme of TOF imaging is given in Figure 5.3.

Usually the timing resolution of the imaging system can be estimated with experimental measurements, see e.g [Sajedi et al., 2022]. For backprojecting the data, timing resolution is used to match the reconstruction TOF kernel. In the image space, data is backprojected as a Gaussian function with a kernel that has a width equal to the measured coincidence timing resolution: if t_{FWHM} is the measured timing resolution FWHM, then the FWHM considered for the Gaussian function is $d_{FWHM} = ct_{FWHM}$ with c the speed of light in vacuum. The center of the Gaussian kernel is located at the distance measured from the timing difference between both detected photons.

If we consider the case of reconstructing a point source f_{PS}, the TOF backprojection f_{TOF} can be expressed as

$$f_{\text{TOF}} = k_{\text{TOF}} * f_{\text{PS}} \tag{5.5}$$

[1] All the main manufactures (GE, Siemens, Philips) currently offer TOF information for image reconstruction. The Gemini TF scanners were the first to have the TOF option in 2006.

where k_{TOF} is the TOF kernel associated to the convolution operator K_{TOF}. This is valid when omitting the spatial resolution modeling represented by K_{img} previously. In practice we saw in (5.1) that a blur is observed when reconstructing a point source even without considering TOF information. In what follows we keep the unique notation of the operator K_{img} and the kernel k_{img}. Therefore when data are obtained from TOF PET imagers, K_{img} accounts for both the physical fluctuations related to PET imaging (e.g positron range) and TOF blurring due to imperfect timing resolution in such systems [Matej et al., 2009].

We can therefore write the TOF reconstruction problem as $f_{\text{TOF}} = k_{img} * f$, or when considering the more general operator K_{img}:

$$f_{\text{TOF}} = K_{img} f \tag{5.6}$$

where f_{TOF} is the reconstruction obtained with TOF backprojection and f is the imaged object. It is important to note that the contribution of both factors in K_{img} (partial volume effect vs blurring from TOF reconstruction) is not equivalent. Indeed in the first case, the FWHM of the PSF kernel (if it is considered alone without TOF) is generally considered as ranging from approximately 1 to 10 mm [Stute and Comtat, 2013]. Recent PET scanners can on the other hand offer a coincidence timing resolution (CTR) of less than 300 ps [Conti and Bendriem, 2019], as shown by the recently commercialized PET system from Siemens that has a measured timing resolution of 210 ps [Reddin et al., 2018]. Even more recent work has shown the possibility of having a timing resolution of less than 100 ps [Cates and Levin, 2018]. Considering a CTR between 100 ps and 300 ps, the corresponding FWHM of the TOF kernel ranges between 30 mm and 90 mm, which is considerably larger than the width due to positron range, photon accolinearity etc.

The model that we assume for TOF reconstruction is (5.6), which is the considered model in e.g [Matej et al., 2009]. It especially allows keeping the same formulation for analytical and iterative methods. With the model (5.6), the projections and backprojections within an iterative scheme are performed in the image domain since they are related to the operator K_{img}. Also, this model is particularly adapted to list-mode data and allows one to work only in the image space, which has computational benefits. As K_{img}

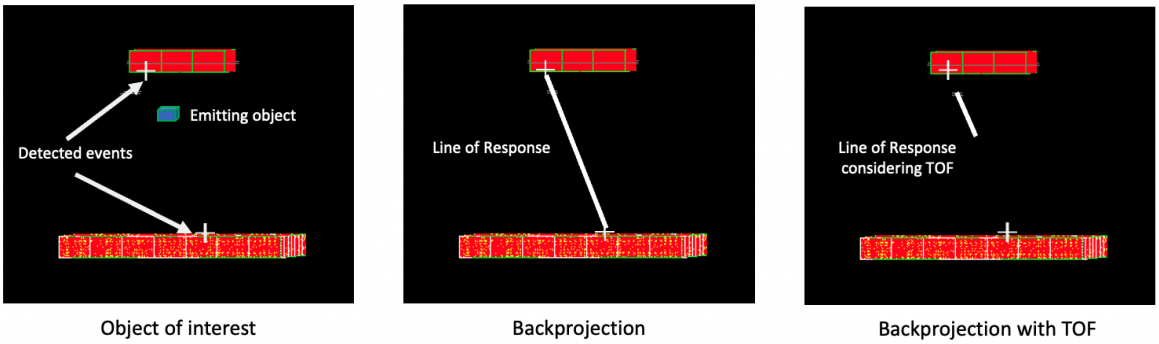


Figure 5.3: Illustration of reconstruction taking time of flight into account. In the right image, the length of the LOR depends on the coincidence timing resolution(CTR). A better timing resolution (lower CTR) implies a smaller LOR: the position of the back-projected event is more accurate.

is a convolution operator, retrieving f in (5.6) therefore corresponds to a deconvolution problem.

When considering list-mode data, f_{TOF} is also referred to as the histo-image corresponding to all detected events. An histo-image $f_{\text{HI}}(S)$ for a subset of events S corresponds to the backprojection of those events only. In this case the group of events is referred to as a view. This is useful for reconstruction algorithms that consider data subsets such as OSEM described in Chapter 2. It is possible to form those groups of events by considering geometrically ordered subsets [Popescu et al., 2004] - gathering a certain number of LOR for instance - or time ordered subsets [Reader et al., 2002].

5.2 Methods for resolution

In Section 5.1 we detailed the specific forward model for PET imaging, depending on whether we consider TOF or not. It was important to study how physics of PET imaging as well as consideration of TOF impact the formulation on the inverse problem; it is now important to review the methods that can be used to solve such a problem. We remind that in addition to the forward model derived in Chapter 1, we have here a convolution operator which might be considered in both the image domain and the projection domain. In most of the reconstruction methods however, it is considered in only one out of the two domains, in which case it accounts for the PSF of the system. We also remind that the PSF can be spatially-variant and it can depend on the object f to reconstruct; in the general case it is then unknown.

A first solution is to use some inverse filtering technique such as Wiener filter, however this usually leads to noise amplification or image artefacts similarly to analytical methods for tomographic reconstruction derived in Chapter 1. Some methods may also rely on inter-modality, making use of high-frequency information obtained from a high-resolution anatomical image (CT/MRI) for instance [Erlandsson et al., 2012]. The structural information from other imaging modalities can in this case be used as a priori information. This allows for instance segmenting an image into a number of compartments where the distribution is considered as uniform; in that case correction is performed between voxels in different regions but not between voxels within the same region. There are also methods that do not specifically require retrieving the image f, but only mean values in some ROI for instance; in such cases correction can be done in the projection domain without reconstructing the image [Huesman, 1984, Carson, 1986]. We do not elaborate on such methods here since multi-modality is out of our scope and we aim at retrieving the entire image f.

Instead, in a first part we focus on classic iterative methods to retrieve the image while retrieving the PSF in the same time, or methods that require prior PSF estimation. We then study data-driven methods that are specific to blind deconvolution.

5.2.1 Model-based methods

The objective of this section is to provide an overview of methods for solving (5.4) - or (5.6) for TOF image reconstruction - that do not rely on neural networks. In this section we split these methods for tomographic reconstruction and PSF correction for

PET imaging into two main categories:

- Estimation of the unknown PSF beforehand; then resolution modeling is either used for post-reconstruction deblurring of the image with model (5.6), or incorporated within an iterative reconstruction algorithm.
- Blind (or semi-blind) deconvolution in which both the PSF and the image are unknown and should be retrieved by the reconstruction algorithm.

5.2.1.1 Decoupled PSF estimation and image reconstruction

In the case of prior PSF estimation, the approximation is made that the PSF is not object dependent. The PSF estimation can be obtained through Monte Carlo simulations or with experimental measurements. For instance in [Sureau et al., 2008] where resolution modeling is considered in the image domain, estimation of the convolution kernel k_{img} for the operator K_{img} is performed with two different methods, each of them considering two different isotropic and stationary resolution kernels. In the first case the considered kernel writes

$$k_{\text{img}}(r) = \frac{1}{N}(\mathrm{e}^{-\alpha r} + \beta) \tag{5.7}$$

and in the second test it is written as

$$k_{\text{img}}(r) = \frac{1}{N}(\beta\,\mathrm{e}^{-\alpha_1 r} + (1-\beta)\,\mathrm{e}^{-\alpha_2 r}) \tag{5.8}$$

where N is such that the kernel is normalized to 1, and α, α_1, α_2, β are parameters that are estimated by reconstructing a 1 mm point source of known activity.

In the experiments authors find improved performance in the reconstruction with the formulation (5.8) which might be considered as more complex/comprehensive. This particularly encourages work towards obtaining more complex - understand with less constraints in the formulation - convolution kernels, with deep learning methods for instance; we will discuss such possibilities in this chapter and in the next one. Especially authors argue that more complex kernels should be used in cases where there is an important degradation of the resolution and if the PSF is spatially variant. However one needs to remember that spatially-invariant kernels have the advantage to speed up the reconstruction, thanks to computation in the Fourier domain. In addition to these considerations, the number of voxels used to model the PSF is also found to be important to limit ringing artefacts.

When the PSF is estimated beforehand, the problem then amounts to perform deconvolution and/or reconstruction with a known forward operator. We first mention methods that perform the reconstruction without resolution modeling then deconvolve the obtained image with the estimated PSF; then we derive methods that include the PSF in the forward operator within tomographic reconstruction.

Reconstruct then deconvolve with estimated PSF As mentioned, a straightforward method is to perform reconstruction with methods similar to those derived in Chapter 2 with the forward operator A. If we consider that K_{data} is the identity - resolution modeling in the image space -, then this amounts to retrieving $K_{\text{img}} f$. Post-reconstruction

deblurring is then needed to retrieve the image f, and this can be performed with classic deconvolution methods.

In [Tohka and Reilhac, 2008] for instance, two methods are compared. The first case is the Richardson-Lucy (RL) algorithm [Richardson, 1972, Lucy, 1974]. It is an algorithm similar to the MLEM algorithm derived in Chapter 2, but it is generally mentioned with this name in the field of astronomy for instance, and also for deconvolution tasks. The only difference with the MLEM mentioned before is that the forward operator here is K_{img} instead of A since reconstruction from projection data has already been performed. The algorithm writes in this case

$$f^{i+1} = f^i(K_{\text{img}}^T \frac{f_{\text{noRM}}}{K_{\text{img}} f^i}) \tag{5.9}$$

where f^0 is a non-zero initialization and f_{noRM} is the initial estimation of f obtained after reconstruction, i.e it is supposed to correspond to $K_{\text{img}} f$. We also recall that multiplications and divisions between vectors are performed element-wise.

Another iterative algorithm presented in [Tohka and Reilhac, 2008] makes use of the reblurred Van Cittert iteration [Carasso, 1999] which writes

$$f^{i+1} = f^k + \lambda K_{\text{img}}^T (f_{\text{noRM}} - K_{\text{img}} f^i). \tag{5.10}$$

with λ a weighting parameter.

Resolution modeling within tomographic reconstruction An other possibility to use the estimated PSF is to incorporate the corresponding operator into the forward model so that reconstruction can be performed as a single step using iterative algorithms [Reader et al., 2003, Sureau et al., 2008], with AK_{img} or $K_{\text{data}}A$ for forward and back-projections. An argument given in [Erlandsson et al., 2012] is that this helps reduce noise.

In [Sureau et al., 2008], resolution modeling is considered in the image domain and OSEM (see 2.1.3) algorithm is used with AK_{img} as the forward operator so that the algorithm simultaneously performs reconstruction and deconvolution. By running experiments on phantoms and on brain real data authors show reduced PVE compared to no resolution modeling. It is however observed that ringing artefacts can appear in the reconstructed image, especially when the resolution effects are overestimated, but also when the PSF perfectly matches the true one. In the study slower convergence is also observed when taking resolution modeling into account.

Reconstruction with MLEM algorithm with resolution modeling in the projection domain was performed in [Tong et al., 2011] and [Thielemans et al., 2010]. In this case the EM update equation writes

$$f^{i+1} = f^i \frac{1}{A^T K_{\text{data}}^T 1} A^T K_{\text{data}}^T [\frac{p}{K_{\text{data}} A f^i}] \tag{5.11}$$

Results in [Thielemans et al., 2010] show that at first iterations, the reconstructed image has a lower resolution than without including the PSF. At later iterations this is no longer the case but the algorithm generates too high contrast. For large sources, ringing artefacts are also observed, even though overshooting decreases slowly over iterations. In

[Tong et al., 2011] also, artefacts in the reconstructed image are observed especially at the edges of structures. Especially, mismatch between the kernel used in the reconstruction and the actual system PSF can strengthen the presence of such artefacts since the mismatch is somehow amplified throughout the reconstruction process. Specifically, a study of the object-specific modulation transfer function (OMTF) is performed. The OMTF is defined as the spectrum of the reconstruction divided by the spectrum of the true object (division in terms of amplitude in the frequency domain). When PSF is not taken into account in the reconstruction, OMTF shows the loss of high-frequency information, i.e a value of zero for high frequencies. This results in a blurred reconstructed object. For the case where PSF is taken into account, high-frequency can be preserved, but there is an amplified band in the OMTF, i.e a band of frequencies where the amplitude for the reconstructed object is higher compared to the true one. This amplified band appears to be correlated with the presence of ringing in the PSF-reconstructed images.

Iterative reconstruction with resolution modeling is also performed in [Gravel et al., 2019]; the PSF is considered in the image domain but more important, it is considered as spatially-variant. In this case reconstruction is performed with the row-action maximum likelihood algorithm (RAMLA) [Browne and De Pierro, 1996], an accelerated maximum-likelihood algorithm similar to OSEM and adapted to emission tomography in particular. Instead of performing image reconstruction using separate reconstructions that consider spatially invariant image-based resolution for each of the different PSF kernels (cf [Matej et al., 2009]), the spatially-variant image-based resolution model is included within a single reconstruction. Results show that such a method allows enhanced quality in the reconstructed image, however convergence is slower and computation time is increased compared to e.g a post-reconstruction deconvolution approach. Indeed 3D convolution operations have to be applied to the image at each forward and backward projection operations for each view in the case where PSF modeling is included in the reconstruction algorithm.

Overall we see that whether resolution modeling is considered in the image domain or in the projection domain, artefacts are often visible in the reconstructed image, even if performance is enhanced compared to no resolution modeling. Also, poor-quality estimation of the PSF leads to amplification of the artefacts.

Reconstruction with estimated kernel for TOF The study of mismatch between the estimated and true kernel of $\boldsymbol{K}_{\text{img}}$ was performed in [Daube-Witherspoon et al., 2006], but in this case considering TOF and a corresponding estimated kernel k. Performing the reconstruction with OSEM algorithm, the study shows that when the considered kernel is narrower compared to the true timing resolution of the PET system, this results in a decreased contrast in the reconstructed image. Using a larger kernel does not affect the outcomes in this study which focuses on complete data. However when dealing with limited-angle data, using larger kernels compared to the true timing resolution leads to ringing artefacts that are similar to the edge effects observed with LOR resolution modeling, as shown in [Gravel et al., 2020] with the dual-panel breast-PET scanner mentioned in 5.1.2. Those artefacts are even wider compared to the ones mentioned before since the spatial extent of the TOF uncertainty is larger compared to the width of LOR resolution blur, see 5.1.4.

The conclusion that we can draw from this review of methods is that when the decon-

volution is performed in the image domain as a post-reconstruction method, this generally results in noise amplification. If the PSF is incorporated within the iterative reconstruction - in the image or projection domain -, the performance in terms of noise is enhanced. Both methods however suffer from artefacts that correspond to ringing in the vicinity of sharps boundaries (Gibbs artefacts), due to missing high frequency information because of limitations of the detector system and/or too large voxels. Also we note that mismatch between the true kernel and the estimated kernel leads to amplified artefacts, whether the kernel accounts for spatial or timing resolution. Therefore it seems of paramount importance to correctly estimate such a kernel and/or correcting for the artefacts caused by the spatial/timing resolution of the PET system.

5.2.1.2 Blind deconvolution

Estimating the PSF beforehand limits the extent to which it can be formulated; for instance, considering object dependent PSF is hardly possible with such a method. As we saw the importance of correctly estimating the convolution kernel, we derive here methods referred to as blind deconvolution; the operator K_{img} or K_{data} is considered as unknown prior to the reconstruction process. The reconstruction task then amounts to retrieving both the image and the convolution operator. In this case the estimated PSF can be different from a reconstruction to another. The difficulty here lies on the non-uniqueness of the solution; when imaging an object with soft edges for instance, one needs to identify the kernel from the smooth-edged object. There are infinitely many solutions to the problem which makes blind deconvolution a particularly hard reconstruction task.

The methods that we mention here are specific to the realm of blind deconvolution where the problem is limited to retrieving K_{img} (or semi-blind if retrieving only parameters of the kernel k_{img}) and f where the data y is such that

$$y = K_{\text{img}} f \tag{5.12}$$

so that in our case this can be viewed as a post-reconstruction method. When considering TOF data, Equation (5.12) is equivalent to (5.6) with $y = f_{\text{TOF}}$. Otherwise one can assume that $y = f_{\text{noRM}}$, i.e a reconstruction obtained by solving (5.4) with no resolution modeling: a FBP for instance. Note that methods discussed here are performed in the image domain so the convolution operator and kernel will be written K and k for simplicity.

To solve (5.12) considering a kernel k for the convolution operator, one can consider a functional such as $||y - k * f||_2^2$ to minimize. The functional is convex w.r.t f or k, but it is not jointly convex so that it is possible to reach a local minima. The usual strategy adopted for (semi-)blind deconvolution is then to perform alternate minimization on f (with k fixed) and on k (with f fixed). Such alternating strategies were used for minimization of a (regularized) least-squares functional [Ayers and Dainty, 1988, You and Kaveh, 1996].

Alternating minimization on both the kernel and the image was also performed using the Richardson-Lucy algorithm in [Fish et al., 1995], considering the Kullback-Leibler divergence $KL(y, k * f)$. The different updates are in this case:

$$k^{i+1} = \frac{k^i}{(f^i)^T * 1} . (f^i)^T * \frac{y}{k^i * f^i} \tag{5.13}$$

$$f^{i+1} = \frac{f^i}{(k^{i+1})^T * 1}.(k^{i+1})^T * \frac{y}{k^{i+1} * f^i}. \tag{5.14}$$

In practice a different number of iterations can be used for the kernel and the image updates; one can do m iterations on k before doing n iterations on f, with m potentially not equal to n.

In [Lecharlier and De Mol, 2013] an algorithm is proposed for minimizing over k and f a cost function involving the Kullback-Leibler divergence - thus adapted to Poisson data - and regularization on both the image and the convolution kernel. Especially, regularization is applied on the PSF to avoid converging towards a trivial solution with an impulse kernel [Levin et al., 2009]. The function to minimize can be written

$$J(k, f) = KL(y, k * f) + \frac{\mu}{2}||k||_2^2 + \lambda||f||_1 + \frac{\nu}{2}||f||_2^2 \tag{5.15}$$

under the constraint that the sum of k equals 1, with μ, λ and ν weighting parameters. Algorithm 5 represents the different steps for updating both the kernel and the image in this case, where one can recognize the alternating Richardson-Lucy algorithm when regularization parameters are set to zero. Note that TV regularization can also be included in such a framework; in this case one may perform TV denoising after the update of f^{i+1} with e.g Chambolle Pock algorithm for Poisson noise.

Algorithm 5: Alternating minimization of the regularized Kullback-Leibler divergence. In practice the α parameter is found with Newton method. However it might not converge sometimes, in this case a solution is to set α to 0 and force the sum of the PSF to be 1 after the update of k^{i+1}. Here f and k are n-dimensional vectors of component f_s and k_s.

initialize $f^0, k^0, \sum_s k_s^0 = 1$
$U = k^i.(\frac{y}{k^i * f^i} * ((f^i)^T))$
Find α Lagrange parameter s.t
if $\mu > 0$, $\mu + n^2(\alpha + \sum_s^n f_s^i) - \sum_s(\sqrt{(\alpha + \sum_s f_s^i)^2 + 4\mu U})_s = 0$
else $\alpha = \sum_s y_s - \sum_s k_s$
$B = \alpha + \sum_s f_s$
$k^{i+1} = \frac{2U}{B + \sqrt{B^2 + 4\mu U}}$
$C = f^i.(k^{i+1})^T * \frac{y}{k^{i+1} * f^i}$
$D = \lambda + \sum_s k_s$
$f^{i+1} = \frac{2C}{D + \sqrt{D^2 + 4\nu U}}$

These two algorithms - regularized or not - offer the possibility to estimate the PSF in the same time as the image. No prior experimental estimation is needed, and the PSF estimation entirely depends on data y - that is obtained from the projections p initially -, thus it is object-dependent. The main drawback of such methods however is that regularization is much needed so that the kernel estimation does not converge towards a trivial solution with k being the identity. Similarly to algorithms in Chapter 2, it is hard to find a once-for-all strategy for the regularization choice. In what follows we will, for these reasons, focus on deep learning based methods that aim to perform blind deconvolution.

5.2.2 Data-driven methods for blind deconvolution

For data-driven methods, resolution modeling with initial PSF estimation will not be discussed. Indeed this amounts to modifying the forward operator within the methods already described in Chapter 3.

Methods that we describe here are therefore mainly related to blind deconvolution, where both the image and the convolution operator are unknown, corresponding to model (5.12). In what follows we split the data-driven methods for blind deconvolution into supervised and self-supervised methods.

5.2.2.1 Supervised methods

Here we focus on methods that require a training dataset with paired blurred/true image, as well as true kernel potentially.

Direct methods Wwith deep learning methods, it is possible to ignore the forward model and to map a deteriorated image to an estimate of the true image. This is done for PET imaging in e.g [da Costa-Luis and Reader, 2017] where a deep convolutional network acts as a post-reconstruction image processing method to reduce artefacts.

For blind deconvolution it is similar; it is not essential to have an estimate of the true kernel at the origin of the blurred data y. Solution for such a direct method include using a UNET, taking y as an input and some ground-truth f as the output. Some variations of the network might be used for such a task, as we will see in the experiments in the next chapter:

- 2D UNET in the coronal, axial or sagittal plane.
- 2.5D UNET similarly to [Perslev et al., 2019] where outputs from all of the three UNETs in each plane are merged - averaged for instance.
- Multi-slice UNET or residual network as in [Xu et al., 2017].
- 3D UNET as for FastPET in [Whiteley et al., 2020].

We will comment more on those variations in the next chapter but we mention here that using multi-slice input or 3D networks in the case of PVE or TOF correction particularly makes sense since it is likely that information in adjacent slices can be useful for the estimation of each slice, since the PSF spreads the activity across different slices.

Model aware methods There are also methods that perform blind deconvolution with deep neural networks while taking into account the fact that both the PSF and the image are unknown. In [Rego et al., 2021] for instance, both the image and the kernel are retrieved by a single network. The corresponding network is applied to image reconstruction for lensless camera[1] for which the model (5.12) is totally adapted. Especially, it is not possible in this application to have an estimation of the PSF done beforehand as for these cameras it can be very different for every acquisition.

[1]The purpose of lensless camera is to reduce size, weight and cost of a camera. Withdrawing the lens however reduces performance, thus there is a need to improve image reconstruction.

The architecture of such a network is represented in Figure 5.4. Two different UNETs are used for either retrieving the image or the PSF from the blurred image. For image comparison, the loss function L_{im} is a combination of adversarial, perceptual and L_2 losses. For the PSF, the loss L_{PSF} is a combination of adversarial and L_1 losses. Training of the two UNETs is however not independent; in order to assess if the convolution of the obtained image and PSF corresponds to the input data, a MSE consistency loss is added to each loss function used for training the two UNets. Denoting the estimated kernel as $\hat{k}$ and estimated image as $\hat{f}$, the loss function then writes

$$L(\hat{k}, \hat{f}) = L_{im}(\hat{f}, f_{GT}) + \lambda_1 L_{PSF}(\hat{k}, k_{GT}) + \lambda_2 ||\hat{k} * \hat{f} - y||_2^2 \tag{5.16}$$

where subscript GT stands for ground truth.

Weights from both networks are therefore updated in an alternate manner with training data consisting of simulated blurred images and corresponding true kernels and true images. It is shown in the study that this decoupled architecture performs better than a simple UNET that does not account for the PSF model. Also, authors note that the PSF network does not retrieve the identity PSF because it has been trained to also recover the right PSF.

It is to be mentioned here that such methods are indeed efficient, however they rely on accurate ground-truth PSF for training the neural network. Depending on the application, the accuracy of this PSF, especially when considered as object-dependent, might be discussed.

5.2.2.2 Self-supervised methods

A lot of methods for blind deconvolution using neural networks involve self-supervised learning. Here methods only require the input blurred image y, and no training dataset is

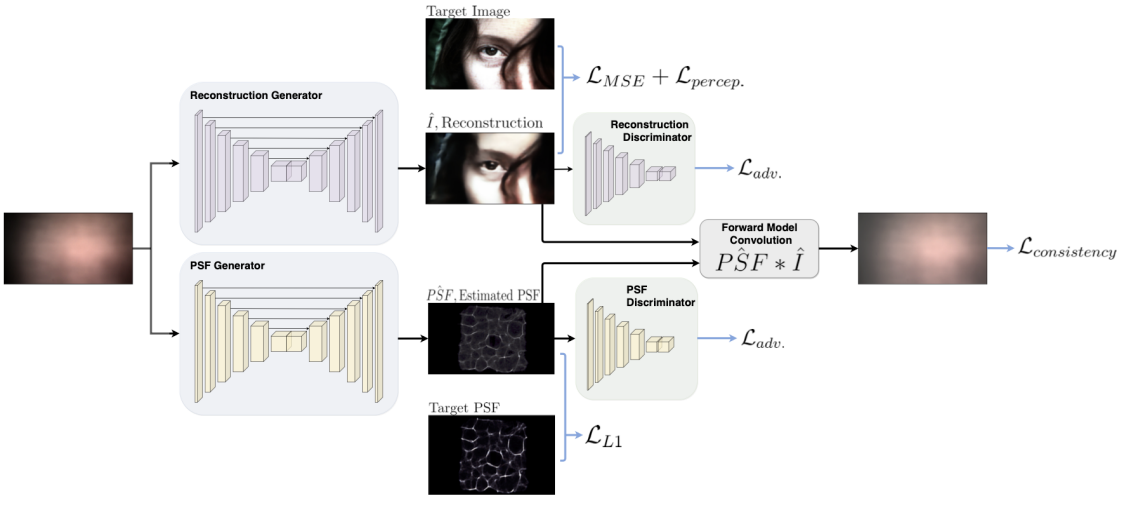


Figure 5.4: Double UNET architecture for image and PSF retrieval in [Rego et al., 2021]. Both PSF and image generators are trained with the mentioned loss: adversarial, perceptual and L_2 for the image, adversarial and L_1 for the PSF. A self-supervised consistency loss with respect to the forward imaging model is incorporated.

needed: update of the weights and thus reconstruction are performed on the fly, similarly to traditional iterative methods.

These methods are mostly based on the Deep Image Prior (DIP) principle mentioned in Chapter 3. We recall that the general idea of DIP networks is to assume that a natural image can be represented by a convolutional neural network. In practice the solution is characterized by a neural network that takes as input a fixed random tensor z, the k-th estimation of the solution being

$$f_k = \Gamma_{\theta_k}(z) \tag{5.17}$$

where Γ is some neural network with parameters θ_k. Here training is only performed with one reference data, i.e the blurred image y. At each epoch - or iteration k -, the weights θ_k are updated by backpropagation of the gradient of a loss function that involves $\Gamma_{\theta_k}(z)$ and y. In what follows we will mention different loss functions and/or architectures that are used for blind deconvolution based on self-supervised learning. We remind that in [Ulyanov et al., 2017] it is shown that the the prior on the data implied by the DIP structure depends on the architecture of the network.

Model-aware methods In [Ren et al., 2020], the architecture of the network is similar to the one used in [Rego et al., 2021] with supervised learning, in the sense that it is a double architecture in order to retrieve both the image and the convolution kernel at the origin of data y. The main difference for the architecture is that the sub-network that retrieves the PSF is a fully-connected network. As mentioned before, it relies on the DIP principle, so the input of the image generator G_ϕ^f is a fixed random image z_f and the PSF generator G_θ^k also takes some random z_k vector as an input. Here note that these inputs never change through the training process. The weights ϕ - resp. θ of G_ϕ^f - resp. G_θ^k - are initialized and then updated with backpropagation with:

$$\min_{\theta,\phi} ||G_\theta^k(z_k) * G_\phi^f(z_f) - y||_2^2 + \lambda TV(G_\phi^f(z_f)) \tag{5.18}$$

where one can notice that TV regularization is considered on the generated image. With this method, no prior training is required since weights are updated only according to the given data at test-time. Also it is mentioned that prior on both the image and the kernel are included within the structure of the neural networks. For instance, they are designed such that positivity constraints are considered in the image - with activation functions -, and a layer in the fully-connected network forces the sum of the PSF pixels to be equal to one. Experiments also show that the structure of the networks allow not to reach the trivial solution of an impulse kernel.

Here the model is taken into account mainly thanks to the double architecture of the network; also the loss function is clearly chosen so that it is adapted to the problem (5.12). It is also possible to consider unrolled iterative scheme for blind deconvolution. We mentioned in this chapter iterative methods, and similarly to what was presented in Chapter 3, those methods can be adapted to neural networks in order to learn the parameters involved in their formulation. Following this idea, the blind Richardson-Lucy algorithm [Fish et al., 1995] was unrolled into a neural network in [Agarwal et al., 2020]. The network, named Deep-URL for Deep Unfolded Richardson Lucy, has a two-fold structure that consists of an image and a kernel generator. Both sub-networks' weights

are updated with backpropagation of a loss function similar to (5.18), with the negative SSIM instead of the L_2 norm. The difference however lies in the structure of G_ϕ^f and G_θ^k networks. Instead of mapping z_k or z_f to the estimated kernel or image with a simple UNET or fully-connected network, rather the overall architecture is derived from the blind RL algorithm. We recall the classic blind RL updates for both the image and the kernel at iteration $i+1$:

$$k^{i+1} = ([\frac{y}{f^i * k^i} * f^{i^\dagger}).k^i \tag{5.19}$$

$$f^{i+1} = ([\frac{y}{f^i * k^{i+1}} * k^{i+1^\dagger}).f^i \tag{5.20}$$

where $(.)^\dagger$ is the flipped version of the vector/matrix argument. The unrolled version of the blind RL algorithm then consists of updating weights W_f^i and W_k^i for both the image and the kernel. The number of simulated iterations is fixed to a number L, the output of the image generator being $G_\phi^f(z_f) = f^L$ and the output of the kernel generator is $G_\theta^k(z_k) = k^L$ with

$$k^{i+1} = \sigma(ReLU([\frac{y}{ReLU(f^i * W_k^i)} * f^{i^\dagger}).W_k^i) \tag{5.21}$$

$$f^{i+1} = \sigma(ReLU([\frac{y}{ReLU(W_f^i * k^{i+1})} * k^{i+1^\dagger}).W_f^i) \tag{5.22}$$

where $f^0 = z_f$, $\sigma(.)$ is the sigmoid activation function in order to ensure some range constraints for both the image and the kernel, and ReLU activation is used in order to ensure the constraint of non-negativity.

One element that is particularly interesting with this method is the fact that once the self-supervised model is optimized for a given blur kernel, it is argued that the learned weights W_f^i and W_k^i can be directly used for performing deconvolution of any image blurred with the same kernel. Overall, results show that such a network outperforms the blind RL iterative method and in the experiments, it does not converge to trivial solution of an impulse kernel.

Other methods An other possibility of using self-supervised learning for blind deconvolution is presented in [Asim et al., 2020]. Here the paradigm is different compared to what was presented with the DIP-like frameworks. Indeed, unsupervised training is used to learn priors on both the images and the kernels with generative networks (GAN or VAE). Once these priors - i.e the generative networks - are learned, the optimization is performed on the latent spaces by backpropagating the gradients of the loss function computed from the blurred image y. Only the latent space values are modified throughout the self-supervised learning so that one ensures that the generated image/kernel belongs to the range of the generative networks. Therefore the difference with previous algorithms is that the input random noise is no longer fixed; instead the weights of the image/kernel generators are fixed. We denote these generators with fixed weights as G_I and G_K.

Such a network is represented in Figure 5.5 and the optimization process corresponds to the following minimization:

$$\min_{z_k, z_i} ||y - G_K(z_k) * G_I(z_i)||_2^2 \tag{5.23}$$

Experiments also contain modification of the loss function, including TV regularization but also methods that give the possibility for the generated image or kernel to be slightly out of the range of the learned generative models. We will not discuss details here but this is to be noted since the method presented as in (5.23) highly relies on the quality of the priors G_I and G_K.

We also note the possibility to use Cycle GAN-like architectures in the case of unpaired blurred/clean images, though this does not apply to our application in the next chapter. We refer to works from [Pan et al., 2020] and [Lim and Ye, 2019] for the reader interested in such methods.

5.3 Conclusion

Overall we saw that the inverse problem that includes some unknown convolution kernel - whether it comes from resolution modeling or TOF blurring - in the forward model leads to increased difficulties in retrieving a clean image. For non data-driven methods, a solution can be to estimate this convolution kernel beforehand, whether the estimated PSF is included in the forward operator within reconstruction methods from projection data, or for deconvolving an image once an initial reconstruction is obtained. This does not allow considering object-dependent PSF, but this is a method that is commonly used in practice. Mismatch between the true and estimated kernel can however lead to amplified artefacts in the reconstruction. It is also possible to perform blind deconvolution from an initially reconstructed image, in which case the PSF does not need to be estimated beforehand. We saw that in this case though, regularization is key in order to avoid trivial solutions. In any case with all of those methods, we observe that ringing artefacts are the main cause of decreased performance for the reconstruction task.

Deep learning based methods allow to get rid of the need for handcrafted regularization. They also have the possibility to reduce artefacts in the reconstructed images. We saw in this chapter that some of these methods rely on supervised training; results seem encouraging but one needs to remember that these methods require ground-truth PSF, which can be particularly hard to obtain especially when the PSF is considered as

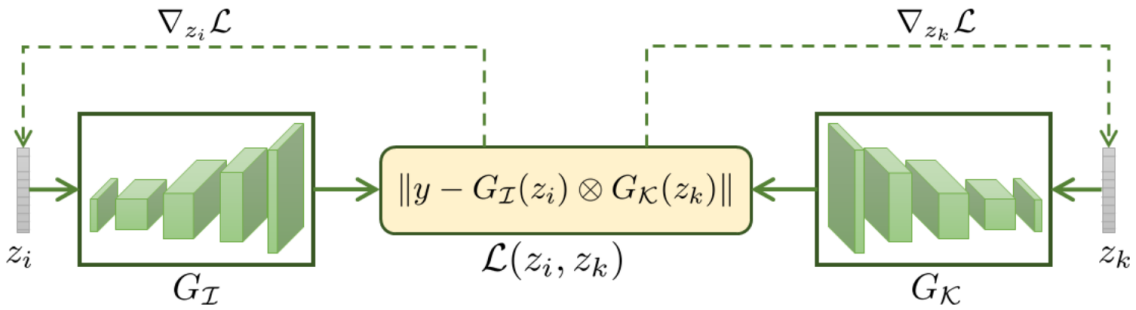


Figure 5.5: Self-supervised scheme for blind deconvolution in [Asim et al., 2020]. Weights from both G_I and G_K are fixed here. They have been trained before hand to learn the distribution of images and kernels. The only variables that are updated in the learning process here are z_i and z_k with $L(z_i, z_k)$ corresponding to (5.23).

object-dependent. This is one of the reasons why many methods in the literature rely on self-supervised learning; they are less constrained compared to traditional iterative methods since they require less parameter tuning. Nevertheless such methods are generally slow to converge. For the medical application that we are focusing on in the next chapter, the PSF is object-dependent and computation time is a key parameter. This lead us to propose a method that enables taking the benefits from the deep learning based techniques discussed in this chapter, without suffering from some of their main drawbacks, especially the computation time associated to self-supervised learning.

Chapter 6

A neural network for image reconstruction and blind deconvolution: application on TOF PET intraoperative imaging

This last chapter is dedicated to the application of deep learning based methods for image reconstruction on a TOF PET system that is designed for intraoperative imaging. This work results from a collaboration between the CREATIS laboratory in Lyon and the Radiation Physics Instrumentation Laboratory (RPIL), which is part of Department of Radiology, Massachusetts General Hospital, Harvard Medical School, in Boston. Work in RPIL has been done to design an innovative PET imaging system, optimize the geometry and the overall configuration of the device regarding the objectives that will be detailed in this chapter. Also, experimental data have been acquired to establish a proof of concept for such a system. On the other hand the expertise of the CREATIS laboratory in image reconstruction is expected to improve the current performance of the detector regarding the image quality and metrics of interest. Especially, existing deep learning methods derived in the previous chapter as well as a newly proposed architecture/training method have been studied for the specific application. This chapter focuses on the corresponding experiments.

The underlying reconstruction problem is consistent with the modelization described in Chapter 5. Constraints related to the requirements for the expected clinical use also lead us to design a deep learning based method that is particularly adapted to a blind deconvolution problem that involves TOF information.

In Section 6.1 we detail the imaging system that is at the core of this study, as well as the corresponding medical applications it can have. Section 6.2 aims at exploring the potential solutions for image reconstruction regarding the constraints of the application, and we also detail the concept of our solution. In Section 6.3 we perform a proof of concept for our solution on a simplified simulated dataset. Tests are pushed further in Section 6.4 where we perform our tests on GATE [Jan et al., 2004] simulations of acquisitions corresponding to the actual imaging system. Finally in Section 6.5 we mention the on-going tests that aim at further validating the solution on realistic simulations and experimental data, as well as potential improvements that can be added to the current

method for reconstruction.

6.1 TOF PET intraoperative imaging

As experiments in this chapter are performed on data from a TOF PET intraoperative imaging system, we detail here the clinical use for such a device and the context for which it can be particularly interesting. Also we describe here the specifics of the imager studied in RPIL that will be used for the experiments.

6.1.1 Presentation of the studied imaging system

The ultimate goal of the studied imaging system is to be a tool for determining the metastatic status of the regional nodes draining a primary tumor, as it is a key prognostic factor in patients with early stage cancer [Sajedi et al., 2019]. This metastatic status can indeed be a valuable asset for evaluating the tumor staging as well as choosing the right therapy for the patient.

The first nodes draining from the tumor are called the sentinel lymph nodes (SLN). SLN identification is used as standard of care in breast and skin cancer patients. A common procedure consists in injecting the patient intratumorally with e.g ^{99}Tc colloid. The first node that is draining the tumor is then identified as SLN, and sent out for histological evaluation regardless of being cancerous or not. If it is evaluated as cancerous, all the secondary nodes connected to the SNL are removed, here also regardless of their cancerous status. Nodes dissection is not without consequences for the patient, and the current standard of care usually does not allow for the evaluation of more than few nodes.

There are guidelines from the American Society of Clinical Oncology (ASCO) regarding SLN identification: the aim is to have more than 85% of detected lymph nodes actually corresponding to lymph nodes (identification rate), and less than 5% of the true nodes that are not detected (false negative rate) [Lyman et al., 2005]. The objective concerning the false negative rate (FNR) is not reached in breast cancer with histological examination as shown in [Pesek et al., 2012]. Tools that enable examination of these nodes accurately, and especially offering the possibility to examine a large enough number of nodes are still required.

In this sense, the argument in [Sajedi et al., 2019] is that neither SLN or secondary nodes need to be removed if one can determine their status during surgery post intravenous (as opposed to intratumoral) injection of radiotracer.

If standard whole-body PET (WB-PET) images can be used to detect lymph nodes larger than 1 cm [Yamamoto et al., 2007], they are not reliable when it comes to detecting nodes smaller than 5 mm [Kalinyak et al., 2014]. Reasons for such results from WB-PET images include low spatial resolution (4-5 mm FWHM) [Sajedi et al., 2022], low uptake value due to the use of 18F-FDG, and low sensitivity of conventional WB-PET scanners. Such limitations are part of the reasons why methods for imaging in an intraoperative context have raised interest.

The current standard of care for SLN identification in a breast cancer patient is to use a two-fold method:

1. Direct visual inspection during surgery, using injection with blue-dye.

2. Use of a gamma-probe thanks to peri-tumoral injection of radiopharmaceuticals.

However, gamma probes as well as near infrared (NIR) probes currently fail to obtain a FNR lower than 5%. Detectability of nodes with such systems is dependent on the tumor location: performance is especially poor near the injection site and for tumors that are deeply located.

Regarding the current limitations of existing methods, the potential of TOF PET for such applications looks promising; it has indeed shown great potential to increase the reconstructed image signal to noise [Cates and Levin, 2018]. As mentioned in Chapter 5, the Coincidence Timing Resolution (CTR) of such detectors have been reduced thanks to recent developments in PET detectors. These developments have encouraged the RPIL group to investigate the potential of TOF PET intraoperative imaging as an alternative to intraoperative gamma probes, 3D gamma cameras [Bluemel et al., 2013] and preoperative PET [Jiang et al., 2019a, Jiang et al., 2019b]. In [Sajedi et al., 2019] and [Sajedi et al., 2022] in particular, a proof of concept for intraoperative TOF PET imaging was performed on simulations and experimental data; the studied imaging system therefore corresponds to the basis for our study.

The overall configuration of the conceptual imaging system is presented in Figure 6.1. The idea is to place a high-resolution PET detector - the probe - atop the patient, as well as a detector underneath the patient's bed, the objective being to incorporate detectors with excellent CTR to compensate for the lacking data in such limited angle geometry. An illustration of the two detectors is given in Figure 6.2. As one can observe, the detectors are placed in a configuration such that solid angle coverage is limited. It was shown in [Sajedi et al., 2019] that bringing the detector modules close to the patient increases the detector solid angle and thus the geometrical sensitivity. Experiments in [Sajedi et al., 2022] - see Figure 6.3 - then consisted in considering flat panel detector geometries placed in parallel planes.

One can observe in Figure 6.1 that the configuration includes the possibility for the top detector to be moved vertically in order to indeed get closer to the patient. The X-Y coordinates of the two detector panels are however considered as fixed with respect to each other. Both top and bottom detector buckets are comprised of arrays of TOF PET detector modules with coincidence timing resolution (CTR) lower than 300 ps.

6.1.2 Current reconstruction method and results

The ultimate objective of this imaging system is to obtain images that are able to meet the requirements listed before, e.g having a FNR lower than 5% for SLNs. In addition to the optimization of the geometry of the detectors, the method to reconstruct the image from acquisition data is of paramount importance as we saw throughout this thesis. The current method used for image reconstruction is based on non-iterative 3D backprojection of the detected coincidence events, taking into account the TOF information. Note also that the current method involves oversampling of the LORs since detector pixels are large (around 4 mm) compared to the image voxel size (1 mm) in this proof of concept detector unit; oversampling then allows obtaining smoother images and to reduce the speckles artefacts that can be observed in the reconstruction. A major advantage of using a non-iterative method is the short reconstruction time which is key for intraoperative use.

Tests shown in [Sajedi et al., 2022] for proof of concept include experimental data. The experimental details are listed below:

- use of two Hamamatsu TOF PET detector modules (C13500-4075YC-12) featuring 12×12 array of 4.14×4.14×20 mm^3 lutetium fine silicate (LFS) crystal pixels with 4.2 mm pitch, one-to-one coupled to silicon photomultiplier (SiPM) pixels. The overall active area of the detector module is 51×51 mm^2.
- Detector coincidence timing resolution (CTR) was measured at 271 ps FWHM for the whole detector.
- 3D phantom was used, containing spheres ranging from 2 to 10 mm diameter, representing lymph nodes and placed inside a 10-liter warm background water phantom of 25 cm thickness.

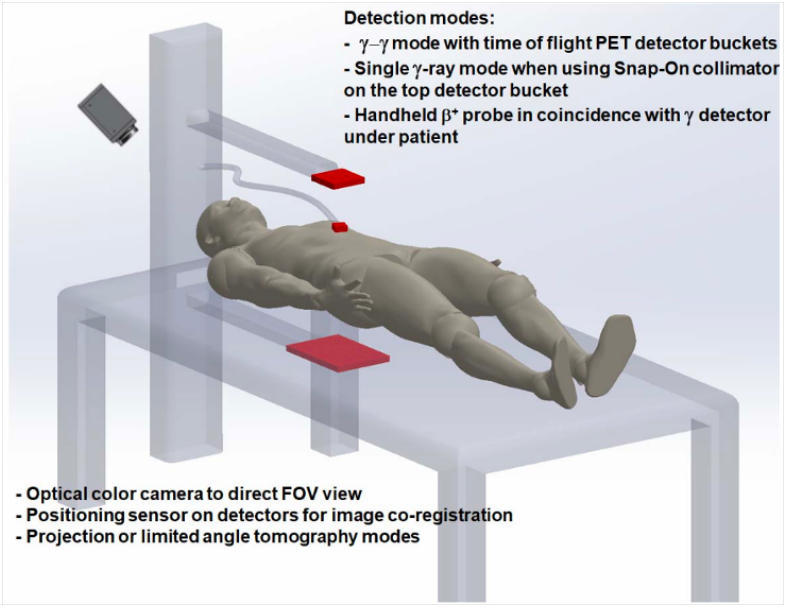


Figure 6.1: Illustration of the conceptual design of the studied intraoperative imaging platform [Sajedi et al., 2019].

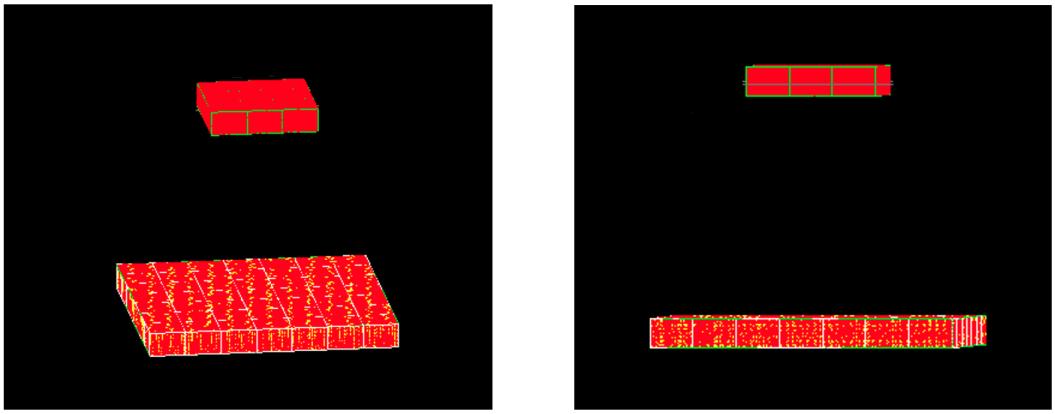


Figure 6.2: Simplified scheme (GATE) of two flat panel detectors placed in parallel planes.

With simple backprojection as the reconstruction method, experiments showed that 6 mm diameter spheres can be identified when the activity ratio between the sources and the background is 10:1, with sub-minute data acquisition. Note that the spheres were placed at 2 cm depth; this distance was chosen since it corresponds to a distance for which no existing optical based intraoperative camera can detect lymph nodes as penetration of visible and NIR photons is limited to only a few millimeters in tissue [Sajedi et al., 2022]. Also, GATE simulations showed that the image quality improves as the CTR improves and with smaller water phantom depth or higher lesion to background activity ratio. An example of reconstruction obtained from both GATE simulations and experimental data can be observed in Figure 6.4.

6.1.3 Modelization

Because it involves PET limited angle data and TOF information, the context is similar to what has been derived in Chapter 5. The model that we therefore assume is the one in Equation (5.12) that we recall here as

$$y = Kf \tag{6.1}$$

where the blurred data y is the TOF reconstruction obtained with backprojection as in Figure 6.4 - however here we do not consider oversampling of the data-, and K is the

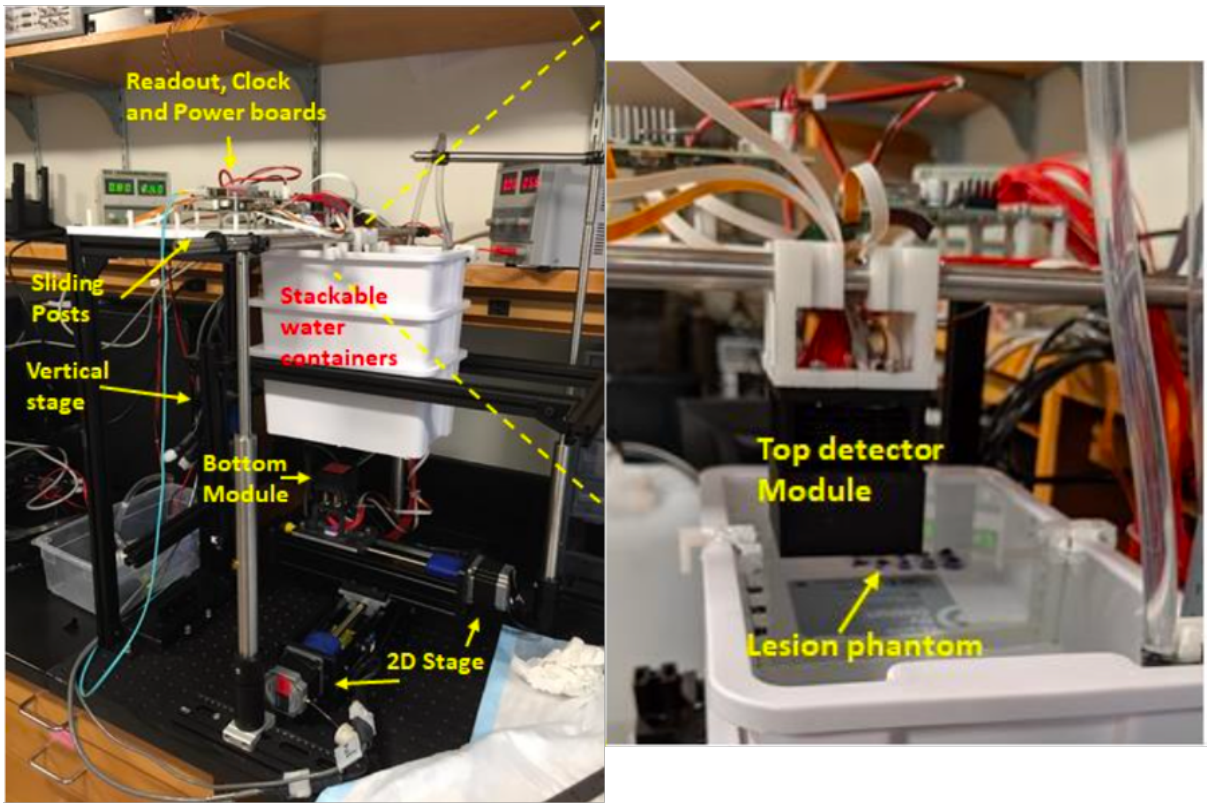


Figure 6.3: Experiments to obtain real acquisitions data in RPIL [Sajedi et al., 2022]. The images represent (a) the overall experimental setup and (b) a zoom on the top detector module right above the hot spheres.

convolution operator. Note that due to the Poisson noise on the data - photon counting noise -, noise also affects y.

Coincidence timing resolution will mostly be considered as ranging between 200 ps and 300 ps in this chapter; this corresponds to 30 to 45 mm localization along the LOR. Compared to the 1-10 mm FWHM uncertainty related to resolution modeling, the effect of TOF uncertainty is predominant in the operator K.

In Section 6.4 the accuracy of the model (6.1) will be assessed with GATE simulated data. Experiments that will be presented in the next sections are performed considering a spatially-invariant PSF. The validity of such an approximation will also be evaluated with GATE experiments. Considering spatially-variant PSFs will be the object of further experiments in the future.

6.2 PAVENET: a hybrid learning method

6.2.1 Requirements for the reconstruction method

Now that the application has been described and the reconstruction problem modeled, the natural next step is to have a reconstruction method that enables reaching the objectives such as detecting spheres/nodes smaller than 6mm and having a FNR lower than 5%, with data acquired from the TOF PET configuration that we have detailed.

Considering the model (6.1), one can draw up a list of groups of methods as mentioned in Chapter 5:

- Model-based methods with no learning process: difficulties to find optimal parameters and computation time are reasons why we do not investigate such methods.

- Decoupled PSF estimation and image reconstruction: we consider here object-dependent PSF, which makes even more sense when covering the case of spatially-variant PSF in the future, so these methods are not adapted.

- Self-supervised learning based methods: here the computation time is once again an issue because of slow convergence for these methods.

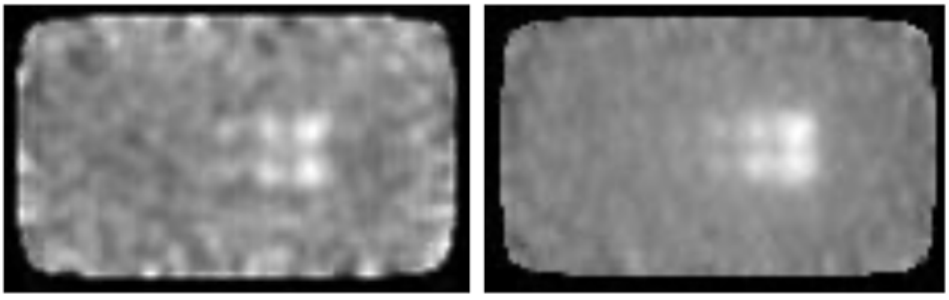


Figure 6.4: Reconstructions obtained by the RPIL group on experimental data (left) and GATE simulated acquisitions (right).

- Supervised learning methods such as UNET from a reconstructed image or a network for retrieving both the image and the PSF considering a consistency loss as in [Rego et al., 2021].

We see that most of the methods presented in the previous chapter are not adapted to our specific task here. It is however possible to consider supervised learning based methods. For instance, a UNET with a TOF backprojection (TOFREC) as input might be a solution. Variations of such a method, especially taking into account spatial information, will be studied in Section 6.4.

Another possibility is to adapt the double architecture from [Rego et al., 2021] - see 5.2.2.1 - to the problem (6.1) by considering the TOF backprojection as the input of the network and the reference image for the consistency loss.

In practice however, the data on which one trains a neural network might belong to a distribution that is shifted compared to the testing data, i.e the data on which the reconstruction actually matters. This is referred to as "distribution shift" in [Darestani et al., 2022]. It can correspond to change of image class, noise level or forward operator at test time. In our case, training will be performed via GATE simulations of realistic acquisitions, but the objective then is to apply the trained networks on data obtained with the real experimental setup. Difference between reconstructions from GATE simulated data and experimental data can be observed in Figure 6.4. Another major concern is the fact that because the position of the top detector can vary continuously in the considered setting, the shape of the PSF can be very different depending on this position, and it is not feasible to include every potential position of the top detector into a training database. Finally, the memory footprint of creating a database from GATE simulations is such that the amount of volumes used for the training process is limited - around 100 volumes is an upper bound in our experiments -, so that the diversity of the considered structures within these volumes is necessarily limited. All of those considerations tend to indicate that our reconstruction method would benefit from a self-supervised scheme, since its very principle is to adapt a network's weights to any input data. As self-supervised networks with random initialization of the weights are slow to converge, they are not suited to intraoperative imaging.

6.2.2 Motivations of PAVENET

Following the multiple requirements listed above, our idea is to consider a self-supervised learning scheme where the weights are carefully initialized. This initialization is, in our method, performed with a pre-training that is supervised, using paired ground truths and TOF reconstructions from GATE simulated acquisitions. Such a method allows one to get rid of the drawbacks mentioned in the previous paragraph since self-supervised learning theoretically allows adapting to any kind of data, and we expect the pre-training to significantly hasten convergence time.

Note that this idea of updating the weights of a network initially trained with supervised data has recently been studied in [Darestani et al., 2022] and in [Barbano et al., 2021]. In the former, this is referred to as test-time training, while in the latter this is interpreted as a DIP-like network where pre-training is performed to "warm-start" the network. Yet, this is a method that has not received a lot of attention for reconstruction tasks or even in the computer vision field more generally. In what follows we will use the

notion of pre-training when referring to the initial supervised training of our network, and test-time training for the self-supervised part. The overall training method is considered as a hybrid learning method.

The combination of pre-training and test-time training represents the main aspect of our proposed method for reconstruction TOF PET images. As for the architecture to use, one needs to remember the DIP principle; with this self-supervised method, the idea is to represent the estimated image $\hat{f}$ as a neural network, and compare the input data y to $k * \hat{f}$ in order to update the weights of the network. Knowledge of k is required to perform self-supervised training. As the PSF is unknown, dependent of the object, etc, a simple UNET architecture would hardly allow to perform hybrid learning. Inspired by the double architecture from [Rego et al., 2021], we thus consider two sub-networks: one aims at retrieving the image, the other one being responsible for retrieving the PSF.

To conclude on this part, the reasons that led us to consider this approach are:

1. Self-supervised learning only or non-data driven methods are not suitable for our task because of the high reconstruction time.

2. Any method requiring an initial accurate estimation of the PSF is not suitable since we consider it as unknown and object-dependent.

3. Supervised learning methods are the only existing viable option but they may be not robust to distribution shifts between training and testing data.

4. Hybrid learning appears to be a solution to obtain fast reconstruction while being robust to these distribution shifts.

5. As self-supervised learning based on DIP is involved, the architecture of the network needs to be two-fold for PSF and image retrieval, similarly to the supervised method in [Rego et al., 2021].

6.2.3 Description of the proposed method

The proposed network, PAVENET - the name was originally given for PArtial Volume Effect NETwork - consists of two blocks as mentioned previously. We are considering convolutional neural networks - UNET especially - so we mostly work in the image space. From a general point of view, the input data corresponds to an analytical reconstruction denoted as y. It can be the TOF backprojection for TOF-PET imaging, or e.g the FBP for cases where no TOF information is available. The first block H_w then maps y to the estimated image; the second block G_θ maps y to the estimated kernel of the PSF. We will detail the exact architecture of those blocks in the experiments later; the main difference however between both blocks is that the last layer of G_θ is a sigmoid layer followed by a normalizing layer in order to put the sum of the kernel pixels to 1. The scheme of the resulting network is given in Figure 6.5.

For pre-training, we assume we have access to ground-truth images and PSF, so that classical methods for training can be used to train each sub-network. The method for estimating ground-truth PSF in practice will be explained in Section 6.4. Both sub-networks are not trained independently. Similarly to [Rego et al., 2021], we consider a

consistency loss that ensures that the estimations are consistent with the input data y. The loss function during pre-training thus takes the general form:

$$L_{w,\theta}(\tilde{k},\tilde{f}) = L_{im}(f,\tilde{f}) + L_{PSF}(k,\tilde{k}) + L_{cons}(\tilde{k},\tilde{f},y) \tag{6.2}$$

where $\tilde{f} = H_w(y)$ and $\tilde{k} = G_\theta(y)$ are respectively the estimated image and PSF, and f and k are the corresponding ground-truths. The loss functions L_{im} and L_{PSF} are typically MSE or MAE in our case but one could also use adversarial or perceptual losses as well.

The consistency loss used in (6.2) is used afterwards for the test-time training in the self-supervised learning step. Our experiments show that including the consistency loss during pre-training is required for correctly initializing the weights; if both sub-networks are trained independently - i.e only with L_{im} and L_{PSF}-, pre-training does not bring any value to the self-supervised learning at test-time.

The actual loss function that we thus consider for pre-training H_w and G_θ is the following:

$$L_{w,\theta}(\tilde{f},\tilde{k}) = ||f-\tilde{f}||_1 + \lambda_1||k-\tilde{k}||_1 + \lambda_2||y-\tilde{k}*\tilde{f}||_2^2, \tag{6.3}$$

where w and θ are alternately updated with backpropagation of the gradient, λ_1 and λ_2 are weighting parameters. During test-time training (TTT), only the consistency loss is used to update the weights of PAVENET, the loss function being

$$L^{TTT}_{w,\theta}(\tilde{f},\tilde{k}) = ||y-\tilde{k}*\tilde{f}||_2^2. \tag{6.4}$$

The loss function during TTT suggests that noise and/or artefacts might appear after some iterations. We recall that in the DIP paradigm, regularization comes from the

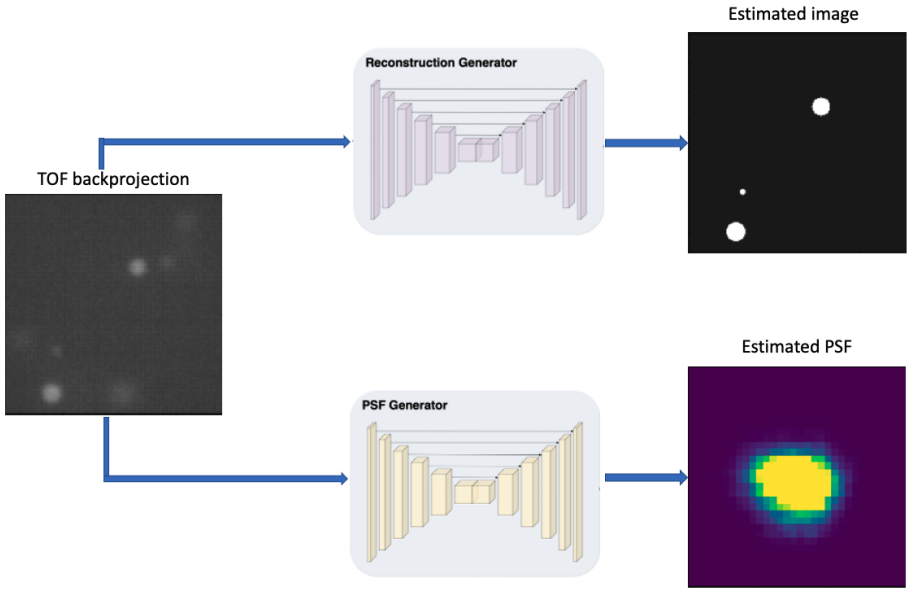


Figure 6.5: Architecture of PAVENET for TOF backprojection input. The network is first pre-trained with supervised data. Weights are then updated at test-time given some TOF backprojection new input. The architecture is similar to [Rego et al., 2021] but the training method is different.

architecture of the network, so that clean images can be predicted by the network even with noisy data. In our experiments, we also consider early-stopping and we test explicit regularization terms to prevent the network from generating noisy images.

For the regularization during test-time training, one can for instance consider TV regularization as it is done in most DIP-based methods. In preliminary experiments we also considered a regularization term that forces the estimated image - or PSF - to be close to the initial prediction after supervised pre-training. As neural networks after supervised training are able to remove noise from the image and generate smooth background for instance, we expected such a regularization to prevent the image from getting too noisy after many TTT iterations. We found that such a regularization term is equivalent to reducing the learning rate during TTT for a fixed number of iterations, so we do not elaborate on such a method.

We put the emphasis on the fact that the proposed method needs to be considered as a general framework that relies on two pillars: hybrid learning and double architecture to retrieve both the image and the PSF. We will consider in the experiments spatially-invariant PSF as this alleviates a constraint for the experiments, especially regarding the architecture of the network and the computation time for hyperparameter tuning. Considering spatially-variant PSF is however part of perspectives for future work that we will detail in Section 6.5. We also remind that in Section 6.4, the validity of the model (6.1) along with the fact to estimate spatially-invariant ground-truth PSF will be evaluated.

Now that we have derived the steps that lead to the design of PAVENET, the validity of such a method needs to be assessed. Our experimental protocol for this task is the following:

1. As a proof of concept, we evaluate the solution on *Python* simulations of simple objects, considering a known model with known PSFs for generating the data. The objective of evaluating the method on a simplistic dataset is to assess whether PAVENET is indeed efficient when the considered model is exact; if it is not the case, there is no reason to go further with experiments where the model is only assumed.

2. The second step is to consider the actual geometry of the TOF PET system detailed in Section 6.1; we consider volumes with hot spheres inside and perform GATE simulated acquisitions from these volumes with a realistic configuration of the imaging system. The PSF for pre-training thus needs to be estimated, and we have no guarantee that the model (6.1) is coherent with the obtained data. PAVENET is then evaluated on such volumes and compared with supervised learning methods (UNET).

3. The final step - which will not be presented in this thesis as it is an on-going work - will be to validate the method on more complex GATE simulations, i.e with lower activity to background ratio, larger phantom depth, etc. Tests will also be performed on experimental data to evaluate the robustness of the method when applied to non-simulated data.

6.3 Proof of concept on simulated data with known forward model

In this section we show the first step of the experimental protocol, i.e evaluation of the method on simulations based on the explicit model. Part of the results have been presented in [Leuliet et al., 2022b].

6.3.1 Experiments

The different steps of our experiments can be listed as follows:

1. Generate a database for pre-training PAVENET, and also for training the supervised method for comparison.

2. Generate 4 different testing sets for evaluating the performance of the networks, and especially evaluate the robustness regarding distribution shift.

3. Perform supervised pre-training on the training set, with optimization of the hyperparameters (HPs) on a separate validation set. Once all HPs are optimized, training on the full training+validation dataset is performed.

4. Tune the HPs for test-time training on a validation set, the weights being initialized thanks to pre-training. We will detail the nature of the HPs related to test-time training in this section.

5. Once HPs are optimized, we use them to perform test-time training - i.e reconstruction - on the different testing sets with the values of HPs obtained in the previous step.

In our simulations in *Python* with Astra toolbox, no TOF information can be considered. Therefore we generate projections data $p = AKf$, where A is the Radon operator. This is still consistent with the model (6.1) since we will consider an initial noisy reconstruction y corresponding to the FBP, instead of the TOF backprojection. We consider here that the convolution operator K can be represented by a convolution kernel k so that the PSF is spatially-invariant. Also we consider a simple Gaussian kernel for k that depends on σ_x and σ_y which are the standard deviations with respect to both 2D-axis. Poisson noise is considered for each pixel of the projections p, the initial pixel intensity being the mean value for the Poisson noise.

Our objective is first to evaluate the benefits of including the model knowledge within the supervised training part; then we evaluate the effect of test-time training on the quality of the reconstructed images. As mentioned, a key element for evaluation is the robustness of the tested methods regarding distribution shifts between training and testing data. For this, the testing datasets are generated with different characteristics compared to the training dataset.

In order to check that the potential performance improvement due to test-time training is allowed thanks to the architecture of PAVENET, we also consider a hybrid learning version of the UNET in which the weights are updated at test-time with (6.4) considering $\tilde{k}$ as the identity since no estimation can be obtained with this network.

6.3.2 Data

The supervised training database is simulated with Python. It consists of 20,000 images of size 128×128 in which disks of different sizes are randomly placed. The size of the disks is different from a disk to another within a single image. The considered activity ratio is different from one image to another, however it is kept constant for all disks within a single image. For the PSF, we consider Gaussian kernels of size 16×16, with varying σ_x and σ_y. For each image-kernel pair, projections data p is generated according to the pipeline mentioned above, the parallel projections being obtained with Astra library [van Aarle et al., 2015]. Ramp filter is considered for the FBP that is given as input of the networks and as reference for the consistency loss.

We consider 20% of the database, i.e 4,000 images, as a validation set in order to tune the hyperparameters and check that no overfitting could be observed during training. Indeed, poor performance on a testing set can be explained either by overfitting during training, or by the lack of robustness when testing a network on different data. As no overfitting was observed in our case, we ensure that the potential drop in performance on testing set is likely related to lack of robustness w.r.t distribution shift.

Four different test sets are generated, each of them consisting of 100 images. These are the images on which supervised methods are tested, and also on which test-time training is performed for hybrid learning based methods. The parameters that vary across all the datasets are the radius of the disks, the lesion to background activity ratio, and the width of the PSF. The advantage of using a simplified database is to be able to manipulate these parameters for evaluating the robustness to distribution shifts. The parameters used for each of the testing sets are represented in Table 6.1. One can observe that distribution shift is modeled as smaller or larger disks (tests B and C) or larger standard deviation for the PSF (test D). Also note that the choice of the width for the PSF is done in accordance to the typical values for uncertainties due to spatial resolution; we mentioned in the previous chapter that FWHM for PSF accounting for the physical fluctuations inherent to PET imaging is inferior to 10 mm. For a Gaussian kernel, we have the formula of the FWHM depending on the standard deviation σ:

$$FWHM = 2\sqrt{2\ln 2}\sigma \tag{6.5}$$

so that the chosen standard deviations in the training set (and tests A, B and C) correspond to a FWHM between 2.5 and 10 mm.

Finally, for each method, supervised training is performed 10 times. This allows having 10 different random initializations of the weights, thus 10 different predictions for each method. Therefore we have 10 different values for the metrics. The results that we present account for the statistical fluctuations due to this random initialization, so that we expect to have metrics values that are statistically significant.

6.3.3 Networks

As mentioned, we consider the FBP as input to the networks instead of f_{TOF}, only for those experiments. The UNET that we use for comparison is a slightly modified version of [Ronneberger et al., 2015], where we consider LeakyReLU activation [Maas et al., 2013] instead of ReLU activations, with parameters $\alpha = 0.3$ for the encoder and $\alpha = 0.01$ for

	Disks radius (mm)	σ_{PSF} (mm)
Training	$[6, 10]$	$[1, 4]$
Test A	$[6, 10]$	$[1, 4]$
Test B	$[4, 6]$	$[1, 4]$
Test C	$[10, 15]$	$[1, 4]$
Test D	$[6, 10]$	$[4, 6]$

Table 6.1: Parameters for all datasets. On each image, between 6 and 12 disks are placed at a random position, with an activity between 6 and 16 Bq/mm^2, the background having an activity of 1 Bq/mm^2. The activity is the same for all disks within a single image; it only varies across different images.

the decoder. There is no fully-connected layer. We do not use any activation function at the end of the network, and we use 32 filters for the first UNET layer instead of 64 in the original paper; this allows reducing the number of parameters while the performance is unchanged as the complexity of the studied structures is simple compared to actual medical images.

The two PAVENET blocks H_w and G_θ are similar to the described UNET; for the PSF block, sigmoid and normalizing layers are added so that the sum of the kernel pixels is equal to 1. Note that using a UNET architecture for H_w seems natural as its role is to perform an image-to-image mapping. For G_θ however, this might be less straightforward. We first tried a network with the same architecture as the encoder part of the UNET that maps the 128×128 image to the 16×16 PSF, but results were not convincing. The performance of G_θ turned out to be satisfying enough with the UNET architecture as is so that we did not look further for a - maybe - more appropriate architecture. Removing skip connections from the UNET could for instance be a solution, though we have not tried it in our experiments. Note also that the output of G_θ is of size 128×128 in order to suit the UNET architecture; however, convolution operation in the consistency loss is performed by only considering the 16×16 center of the output since the ground-truth that we consider is of size 16×16. This also allows hastening computation.

Supervised training of the neural networks is performed with Adam optimization, on 200 epochs for UNET and 250 for PAVENET, the first 50 epochs in the latter being made withouth the consistency loss because intial predictions $\tilde{f}$ and $\tilde{k}$ are of poor quality during the first epochs. For the 200 epochs including the consistency loss, optimization of H_w and G_θ is performed alternately. Simultaneous optimization and different ratios between optimization steps for both sub-networks - e.g 3 steps for one and 1 for the other - do not improve performance. The batch size that we have considered for training is 8. Supervised training of the UNET takes around 4h and for PAVENET it takes around 14h on a Nvidia Tesla V100 GPU. Prediction of an image - without test-time training - takes less than a second even on CPU.

For test-time training, PAVENET is optimized according to (6.4) with Adam optimizer also. We show results after $1,000$ iterations - or epochs - since this is the value that

we fixed for optimizing the TTT hyperparameters on the validation set, as it corresponds to a reasonable 40 seconds computation time on the mentioned GPU. We will also show some results for the TTT performed on UNET only.

Hyperparameters such as batch size, number of filters and standard values for parameters related to activation functions or Adam optimizer were selected manually as an initial step in our experiments. As for the learning rate for both H_w and G_θ, λ_1, λ_2, we performed a grid-search optimization on the validation set. Following this optimization step, the simple UNET learning rate was set to 10^{-3}. H_w is pre-trained on 50 epochs with a learning rate of 10^{-3}, then 10^{-5} for 200 epochs and finally 10^{-6} for test-time training. G_θ is pre-trained on 50 epochs with a learning rate of 10^{-4} then 10^{-5} for 200 epochs and 10^{-6} for test-time training.

The metrics that we show (mean and standard deviation) correspond to mean values across the 10 different predictions obtained from different weights initializations. Each of them is a mean over the 100 predictions corresponding to the 100 test images. The standard deviation over the 100 predictions, for a single weights initialization, is relatively low compared to the standard deviation we show.

6.3.4 Results

Figure 6.6 shows images reconstructed with FBP, UNET and PAVENET on test A that has similar characteristics compared to the training dataset. One can observe that in this case, the UNET seems to accurately reconstruct the image; retrieved activities are however slightly under-estimated compared to the ground-truth, which is not the case for PAVENET.

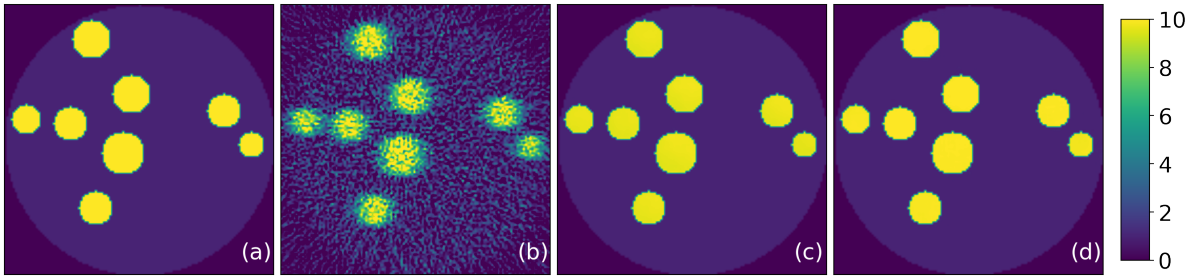


Figure 6.6: Example of reconstruction for an image from the test A dataset. (a) Ground truth, (b) FBP, (c) UNET, (d) PAVENET, after pre-training only.

On test B that contains disks of smaller size, limits of a supervised method like UNET can be observed in Figure 6.7, as the radius of the reconstructed disks is clearly larger compared to the ground-truth disks. Visually, it seems that the radius of the size of those disks correspond to the minimum size for which the network has been trained. The fact to include model knowledge during supervised training helps retrieve smaller disks as it can be observed in Figure 6.7 where the initial prediction of PAVENET after pre-training is shown in the image (c). Test-time training seems particularly efficient as shown by the PAVENET prediction after 1,000 epochs of self-supervised learning. Especially, we observe the ability of the network to retrieve disks with radius significantly smaller than the ones used for pre-training. In Figure 6.7 the PSF prediction of the network is similar

to the ground-truth; this allows the consistency loss to be particularly efficient during test-time training.

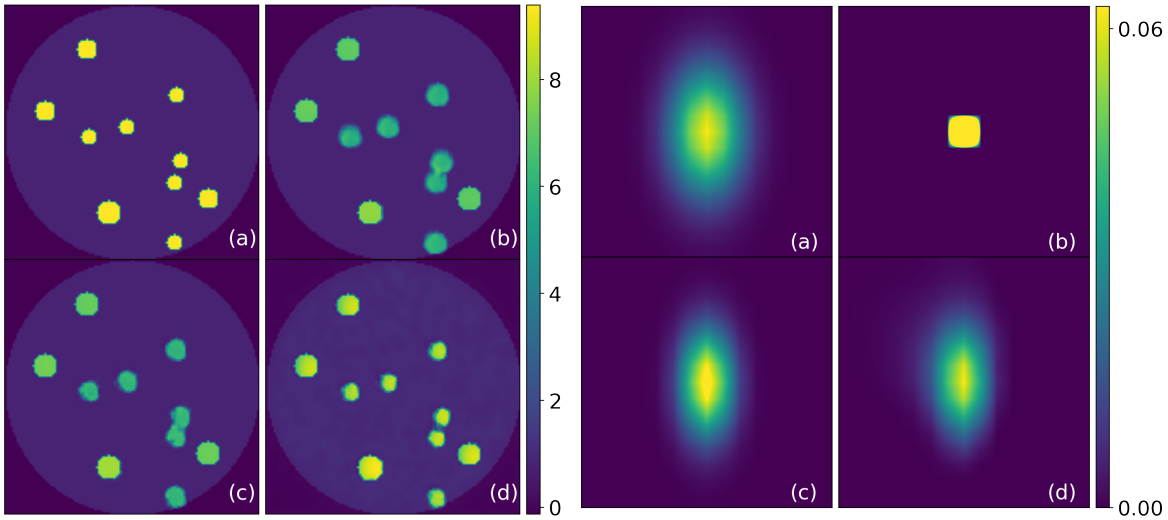


Figure 6.7: Test B : Image (left) and PSF (right) for (a) the ground-truth, (b) UNET, (c) PAVENET after pre-training only, (d) PAVENET after test-time training. Note that no actual explicit kernel is retrieved by the UNET which amounts to consider a identity kernel as shown in the Figure.

In addition to this visual assessment, we compute metrics for the reconstructions on the different testing sets. We mentioned in Section 6.1 the importance of the FNR for which we aim at "missing" less than 5% of the spheres in the experimental setting. An other point is the objective of 85% identification rate, i.e the fact to have a reconstruction method that achieves less than 15% of false positives, corresponding to spheres that should not have been identified as hot spheres. Both of these metrics are not adapted to this preliminary study; indeed the task being simpler compared to reconstructing actual experimental data, all methods including FBP manage to have a 0 % FNR or 100 % identification rate. Another aspect to evaluate is the activity that is retrieved in the reconstructed image; indeed the value of activity - or uptake value - within a hot sphere might provide important information for the image analysis. We therefore compute the Concentration Recovery Coefficient (CRC) which is defined as the ratio $CRC = \tilde{f}_{\text{act}}/f_{\text{act}}$ where subscript act refers to the mean activity inside disks in the image . For the reconstructed images we use Otsu thresholding [Otsu, 1979] to delineate disks and background so that we can compute the mean activity value within the disks. As the ratio should be close to 1, we present results for $CRC_{err} = |CRC - 1|$.

Also, we observe that test-time training with the consistency loss leads to noisy areas in the reconstructed images after some iterations, because of the fact that the data-fidelity term is related to the noisy FBP. This noise does not deteriorate the accuracy of the reconstructed image in terms of activity and shape of the retrieved disks because it essentially corresponds to background zones. This however leads to decrease of the SSIM metric in our tests; we therefore propose a corrected SSIM (SSIM Corr.) that computes SSIM between images where all the pixels that are not part of a disk - w.r.t Otsu thresholding - are set to 0. This allows comparing the structure of images focusing

	Test A				Test B			
Methods	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.
FBP	12.86 ± 0.00	0.093 ± 0.000	34.31 ± 0.00	0.284 ± 0.000	14.65 ± 0.00	0.058 ± 0.000	61.90 ± 0.00	0.109 ± 0.000
UNET	44.66 ± 3.29	0.950 ± 0.031	1.60 ± 0.78	**0.999 ± 0.001**	21.89 ± 0.14	0.883 ± 0.029	22.70 ± 0.95	0.931 ± 0.003
PAVENET-0	**50.29 ± 0.24**	**0.993 ± 0.001**	**0.54 ± 0.12**	**0.999 ± 0.001**	21.79 ± 0.26	**0.924 ± 0.004**	21.50 ± 1.08	0.930 ± 0.005
PAVENET-1000	43.31 ± 0.32	0.932 ± 0.017	1.06 ± 0.42	0.997 ± 0.004	**22.58 ± 0.28**	0.899 ± 0.015	**19.25 ± 1.13**	**0.940 ± 0.004**
	Test C				Test D			
	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.
FBP	10.60 ± 0.00	0.116 ± 0.000	17.37 ± 0.00	0.255 ± 0.000	12.24 ± 0.00	0.052 ± 0.000	59.21 ± 0.00	0.066 ± 0.000
UNET	18.64 ± 0.39	0.775 ± 0.034	4.23 ± 1.28	0.895 ± 0.008	25.91 ± 0.71	0.903 ± 0.042	11.11 ± 1.11	0.977 ± 0.007
PAVENET-0	18.50 ± 0.25	**0.818 ± 0.008**	7.81 ± 0.78	0.903 ± 0.009	26.90 ± 0.38	**0.947 ± 0.007**	**8.57 ± 0.87**	0.971 ± 0.005
PAVENET-1000	**20.96 ± 0.42**	0.783 ± 0.016	**3.27 ± 0.78**	**0.935 ± 0.006**	**27.64 ± 0.20**	0.891 ± 0.015	9.01 ± 0.79	**0.980 ± 0.003**

Table 6.2: Metrics for all tested methods. PAVENET-0 refers to the reconstructed images from PAVENET after pre-training only, while PAVENET-1000 corresponds to predictions after 1000 epochs of self-supervised learing.

only on areas of interest. Note that PSNR might be affected by the presence of residual noise also, but as the objective of the metric is specifically to measure the signal to noise ratio, no corrected PSNR is proposed.

In Table 6.2 we show the results for these metrics. We present results for PAVENET before test-time training, i.e after supervised pre-training, as PAVENET-0. The results for 1000 epochs of test-time training are given in the row PAVENET-1000. On test A, quantitative evaluation confirm the good performance observed for the UNET in terms of noise removal and accuracy of the structure in reconstructed images, even if using PAVENET for supervised training allows obtaining even better results for PSNR and SSIM. Small decrease of PSNR can be observed for PAVENET-1000, probably because of the presence of residual noise in the reconstructed images, but overall the performance is satisfying for all networks in this testing set corresponding to similar configuration compared to the training database.

When testing on disks with smaller radius, self-supervised learning with PAVENET shows significant improvement compared to UNET in terms of PSNR and CRC, with an error on the retrieved activity ratio reduced by almost 4%. This improvement is also visible on PSNR and CRC for test sets C and D with larger sphere size or PSF. As for the SSIM, the improvement compared to UNET is only visible when considering the corrected version for the reasons that we mentioned earlier.

Figure 6.8 shows the evolution of PSNR and CRC error with respect to the number of epochs during self-supervised learning. We include results for PAVENET, and also for UNET where we use self-supervised learning with the identity for the estimation of the PSF as explained earlier. Value of metrics at epoch 0 correspond to predictions obtained after pre-training on the supervised dataset. One can observe in these graphs that self-supervised learning for UNET is not efficient despite fine-tuning of the learning rate. Therefore, it is the fact of including model knowledge, in addition to an accurate estimation of the PSF, that allows PAVENET to benefit from the self-supervised learning scheme, as we observe a clear improvement on PSNR and CRC for the considered test. This statement is valid when considering test B, C and D; when it comes to test A, performance of PAVENET might slightly decrease as mentioned before.

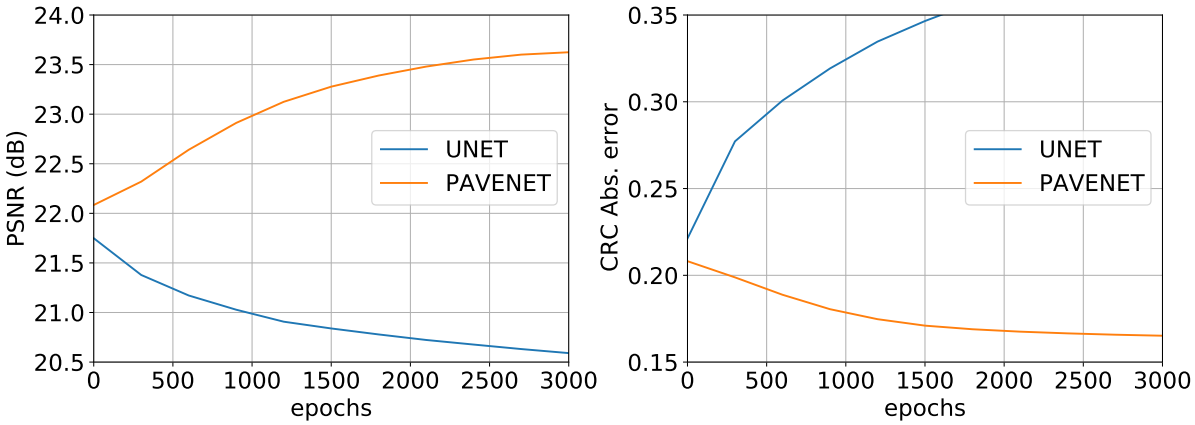


Figure 6.8: Metrics during test-time training on test B with smaller disks.

6.3.5 Analysis and first conclusion

We showed in this simplified context of PET acquisitions data that model knowledge included in the neural network enhances the performance of the reconstructed images for the considered metrics. Indeed, comparing only the supervised methods, PAVENET-0 outperforms UNET on images from the test A dataset. The limitations of UNET become clearly visible when a distribution shift is simulated in the testing dataset. The hybrid learning method that we propose for PAVENET allows overcoming those limits as shown by the ability of the network to retrieve disks with a radius considerably smaller compared to the disks used in the training set. This is mainly due to the self-supervised learning scheme that allows adapting to new characteristics of the data. When considering self-supervised learning, we saw that it is crucial to use a network such as PAVENET that takes into account - and estimates correctly - the convolution operator when computing the consistency loss, as shown by the drop of performance when using test-time training on the UNET only. Note also that results that we showed are obtained after 40 seconds of test-time training, knowing that implementation can probably be optimized. When no pre-training is performed however, we found no way to get decent results with PAVENET in a reasonable amount of time; we tried up to $10,000$ epochs. This is an argument that is consistent with results shown on other applications as in [Darestani et al., 2022] and [Barbano et al., 2021].

The increased performance of PAVENET on smaller disks is particularly encouraging at this point, since the application for TOF PET intraoperative imaging especially requires identifying small spheres for getting a FNR inferior to 5%. The current TOF reconstruction method indeed only achieves identification of 6 mm diameter spheres.

Overall, the results that are shown in this section correspond to tests on generated data where the forward model is known. These experiments allow to validate PAVENET when the model which is considered for coming up with the method itself is correct on the training and testing data. In the next section we will assess whether this holds when we do not entirely control the model at the origin of the data, so that we can only assume that it corresponds to (6.1).

6.4 Experiments on GATE simulated data

Since tests in Section 6.3 suggest that PAVENET is a viable alternative to supervised post-processing methods, it is relevant to perform experiments on data corresponding to the studied TOF-PET imaging system. Here the data are acquired via Monte-Carlo simulations, so that we can only assume that the forward model corresponds to (6.1).

Our first objective is to evaluate the performance of PAVENET compared to a post-processing UNET. The choice to compare PAVENET with such a method is motivated by the fact that this is the most adapted state-of-the art deep learning based method regarding the constraints of TOF-PET intraoperative imaging, as justified in 6.2.1.

In our experiments we have considered a more general second objective, which is to assess the ability of deep learning based methods to enhance the quality of TOF reconstructions for the particular studied device, since it has not been done before. UNET and PAVENET are therefore the two algorithms that we consider towards this objective. Variations of UNET will be tested in the experiments for this purpose: use of adjacent slices as input, training in different directions (i.e on slices parallel and/or orthogonal to the detectors), etc. Study of these variations is mostly decoupled from the PAVENET vs UNET study since for PAVENET we only tested training on coronal or sagittal slices so comparison would not be entirely fair.

Our experimental procedure is similar to the one considered in 6.3.1. There are only two differences:

- The initial reconstructions are obtained by backprojection using TOF information, from list-mode data obtained with GATE simulations.
- As the ground-truths that we manually generate only consist of images - instead of kernels and images before -, the ground-truth kernels need to be estimated for pre-training PAVENET. We will detail the method for PSF estimation in 6.4.2.

6.4.1 Data

For the training set we were able to simulate acquisitions for 100 different volumes. For each of the testing sets, 5 volumes are used. For training the networks we need paired ground-truth volumes and TOF reconstruction. The method to generate ground-truth volumes is presented in 6.4.1.1. We detail in 6.4.1.2 the detectors geometry, in 6.4.1.3 we explain how acquisition is performed with GATE and in 6.4.1.4 we describe the TOF reconstruction method that we use to consider the input of the networks. TOF reconstruction also corresponds to the reference for which we compare the tested algorithms since it is the currently used method. For PAVENET, ground-truth PSF are required and we explain the method to obtain them in 6.4.2.

6.4.1.1 Phantom and hot spheres

The phantom for all volumes is the same. It consists of a rectangular container measuring $350 \times 50 \times 350$ ($L \times H \times W$) mm^3, corresponding to a simulation of a slice of patient body placed in the scanner field of view. Similarly to [Sajedi et al., 2022], the container is filled with 5.3 kBq/cc back-to-back gamma source, i.e a source where two annihilation photons

	Sphere Radius (mm)
Training	$[1.5, 5]$
Test A	$[1.5, 5]$
Test B	1

Table 6.3: Parameters for all datasets. In each volume, between 8 and 16 spheres are placed at a random position, with a 10:1 sphere-to-background activity ratio.

are emitted in opposite directions. The background tracer concentration is based on a 10 mCi injection of 18F-FDG into a 70 kg human.

Between 8 and 16 hot spheres are randomly placed inside each volume. Spheres can be placed at any depth inside the phantom; we restrict the range of the position of spheres center for the X and Z axis to $[-40, 40]$ mm so that they are placed under the top detector (see 6.4.1.2). Similarly to our first experiments, we consider two testing sets, one of which consisting in smaller spheres compared to the training database as shown in Table 6.3. We consider a 10:1 sphere-to-background activity ratio for every sphere in the training and testing sets in order to simulate tumor and lymph nodes. In our experiments the spheres are filled with back-to-back gamma sources, for two main reasons. First, we wanted to take a similar configuration compared to [Sajedi et al., 2022]. Second, running GATE acquisitions for many volumes in order to create training and testing data, is time and memory consuming. The fact to use back-to-back gamma sources hastens computation compared to positron sources. The consequence of such a choice is that positron range is not taken into account in these experiments. For the second phase of experiments (item 4 in the experimental protocol), this can be included.

6.4.1.2 Geometry

For the geometry of the detectors we simulated a configuration that is illustrated in Figure 6.9. There are two detector panels with different numbers of detector modules per panel. The top detector refers to the detector placed above the patient during intraoperative imaging. The bottom detector is the detector located underneath the patient's bed.

In our simulations the top detector consists of a 3×3 array of same size modules. Each module is comprised of $24 \times 7 \times 24$ array (7 DOI levels) of $2.1 \times 5.0 \times 2.1$ mm^3 lutetium fine silicate (LFS) crystal pixels with 2.2 mm pitch one-to-one coupled with silicone photomultiplier (SiPM) pixels. The overall dimensions of the top detector are $159 \times 35 \times 159$ mm^3.

The bottom detector consists of a 7×7 array of modules. Each module is comprised of $12 \times 7 \times 12$ array (7 DOI levels) of $4.1 \times 5.0 \times 4.1$ mm^3 similar crystal pixels with 4.2 mm pitch. The overall dimensions of the nottom detector are $371 \times 35 \times 371$ mm^3.

For a given number of modules and corresponding crystal size, this configuration was shown to be optimal by the RPIL.

Also, the face to face distance between the two detector panels is 287.5 mm. Note that in our experiments the top of the phantom is placed at 2.5mm from the top detector, similarly to what could be done with intraoperative imaging.

6.4.1.3 Acquisition

Simulation was performed across 20 CPU threads with 3 seconds acquisition time per thread using random engine with automatic seed. Based on [Sajedi et al., 2022], detectors were modeled with 300 ps CTR and 16.7 % energy blurring with 511 keV as the energy of reference.

Simulations were performed with version 8.2 of GATE. The physical processes[1] that we considered are:

- Photoelectric effect
- Compton scattering
- Rayleigh scattering
- Ionization
- Bremsstrahlung
- Positron annihilation
- Multiple scattering.

6.4.1.4 Method for reconstruction

For reconstruction we used a *C++* code developed by the RPIL that takes list-mode data as input. It consists of a simple 3D back-projection method with TOF information [Sajedi et al., 2019]. In this method, a Gaussian curve is back-projected with 300 ps FWHM into the image matrix, and sensitivity correction is performed.

[1]See GATE documentation.

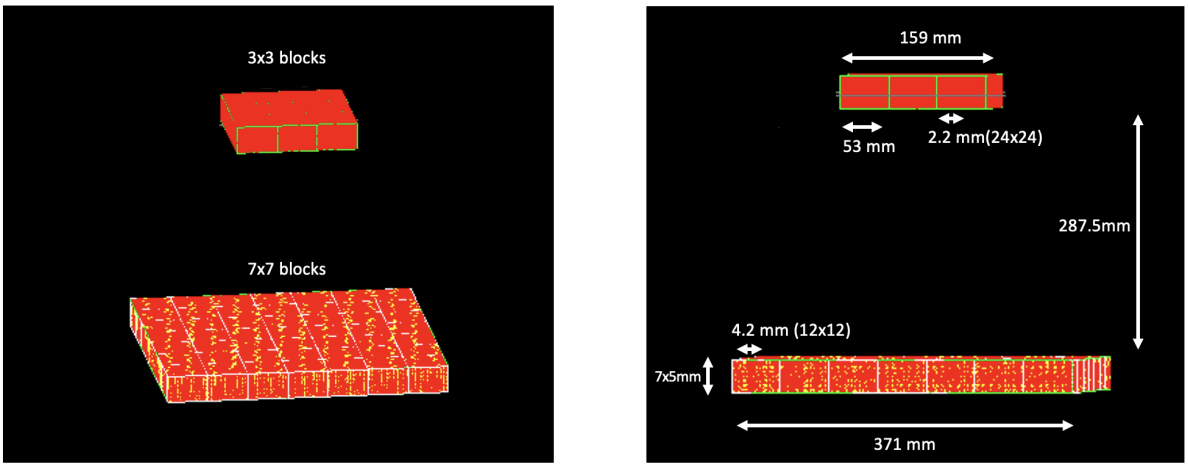


Figure 6.9: GATE render of the intraoperative PET system geometry considered in our simulations. There are 49 (7×7) and 9 (3×3) detector modules under and above the patient. All modules have a 53 mm width and length, with 35 mm depth. The crystal pitch is different for top and bottom detectors (2.2 mm vs 4.2 mm).

In our experiments we consider a $192 \times 100 \times 192$ image matrix with 0.5 mm voxel size. We do not reconstruct the entire volume since we know that spheres are located inside the $96 \times 50 \times 96$ mm^3 area corresponding to this image matrix.

In [Sajedi et al., 2022] the small image voxel when used with 2.2 mm or 4.2 mm detector pixels leads to speckle artefact in the reconstructed image. To minimize this artefact, authors used oversampling of the interaction position in X-Z directions across the detector. A 10x oversampling factor in each dimension was considered, creating 100 LORs from one LOR in the list-mode data. However in our case we do not perform oversampling. Indeed, we aim at having fast reconstruction, whether it is for practical use but also because database creation is time-consuming. Also, we argue that post-processing methods can remove the speckle artefact. Considering this, the TOF reconstruction that we will consider for reference when showing results is the one obtained without oversampling.

6.4.2 Methods

The deep neural networks that we use for post-processing the TOF reconstruction are 2D. We list several reasons for this choice:

- Time and memory are limiting factors to build the training database. Here we were able to create 100 volumes; for a deeper phantom - 25 cm vs 5 cm, that we will consider in the next phase of experiments - computational cost is higher for GATE acquisitions. In this case we were able to generate only 40 volumes. The relatively low number of volumes can be limiting when using 3D networks, whereas working with 2D slices allows increasing the number of data points and use less networks' parameters.

- Current implementation of PAVENET is only adapted to 2D. For a fair comparison all the networks are 2D. Note that the comparison between 3D and 2D networks might be biased since number of data points and networks parameters are inevitably different.

- 2D networks allow to easily select a number of slices at test-time, e.g reconstruct only slices at a certain depth. Conditionally to the possibility of parallelizing the computation, this allows getting faster reconstructions.

6.4.2.1 Model

We consider the forward model (6.1) where y is the TOF reconstruction, and k and f are respectively the unknown PSF kernel and image. For pre-training PAVENET we need to have access to ground-truth PSF. Contrary to our previous experiments, we do not generate these PSF, so we need to estimate them.

Method for estimating the ground-truth PSF For each ground-truth f in the training set with a 10:1 sphere-to-background activity ratio, we simulate the corresponding high-dose reconstruction f_{HD} with 200:1 activity ratio. The TOF reconstruction of f_{HD}, denoted as y_{HD}, is therefore much less noisy compared to y. This allows one to use an iterative algorithm without regularization to find an estimate kernel k^i from each 2D

pairs (y^i_{HD}, f^i_{HD}) in the training dataset. Tuning a regularization parameter for each data pairs in the training set would indeed be impractical.

Note that we are considering 2D convolution with PAVENET; therefore the pairs that we consider correspond to 2D slices. This means that there is a ground-truth kernel for every slice in the training set. As PAVENET can be trained either on coronal or sagittal slices, the kernel is obtained from the deconvolution performed in the corresponding direction.

To retrieve k^i from (y^i_{HD}, f^i_{HD}), we solve the following minimization problem:

$$\min_{k^i} KL(y^i_{HD}, k * f^i_{HD}). \tag{6.6}$$

This corresponds to a non-blind deconvolution problem that we solve with 100 iterations of Richardson-Lucy algorithm. Once the kernel is obtained, the training data points consist of corresponding (y^i, f^i, k^i) triplets.

We consider 32×32 PSF with 0.5 mm pixel size. This is relatively small compared to the standard deviation of the Gaussian function used for 3D backprojection, which is around 19 mm considering 300 ps FWHM CTR. This is especially the case considering kernels on sagittal slices, as TOF blurring is mostly in the Y direction, i.e the direction orthogonal to both detectors. However in our tests we found that increasing the size of the PSF array did not reduce $KL(y^i_{HD}, k * f^i_{HD})$, with the drawback that larger PSF implies higher computation time for PAVENET.

Examples of training data In Figures 6.10 and 6.11 we show some examples of PSF in the training sets, for sagittal and coronal slices. This allows one to have a visual assessment on the quality of the PSF estimation. Especially we show in Figure 6.12 the convolution between the estimated ground-truth kernel k and the ground-truth f for several examples. One can notice that this is visually similar to a non-noisy version of the TOF reconstruction y. This is consistent with the fact to use a data-fidelity term between y and the convolution of the estimates of k and f in the consistency loss for PAVENET.

Also, one can observe that the kernels considered as ground-truths are different depending on the object/slice. This supports the fact that we consider an object-dependent PSF in the model (6.1).

6.4.2.2 Networks and training

In what follows we give details on architectures and training parameters that are used to train networks for which we show the results afterwards. Once again, 20 % of the training data are used as validation to tweak the different hyperparameters. Once these HP are found, training is performed on the entire training set.

UNET and PAVENET In the preliminary study we considered 2D images; here the images in the database are 3D. As mentioned, we use 2D networks. Therefore, one can train the networks either with axial, coronal or sagittal slices. The symmetry of the geometry described in 6.4.1.2 is such that it is equivalent to consider either axial or sagittal slices. All networks are thus trained in two different configurations: coronal and sagittal.

3D images are $192 \times 100 \times 192$; the size of the images considered for the networks trained with coronal slices is therefore 192×192, and for sagittal slices it is 100×192. The architecture of PAVENET is such that the output kernel size is the same as the input image; however we only consider a 32×32 area for computing the convolution and the loss function with the ground-truth kernel, so that most of pixels in the actual output of G_θ are not significant.

Configuration for both UNET and PAVENET is mostly similar to our initial experiments in 6.3.3. We have however tested different number of layers depth for the UNET block; on the validation set, a 4-layer deep UNET gave similar performance compared to the original 5-layer. We explain this by the low complexity of images compared to CT images used in the original paper [Ronneberger et al., 2015]. We therefore use a depth

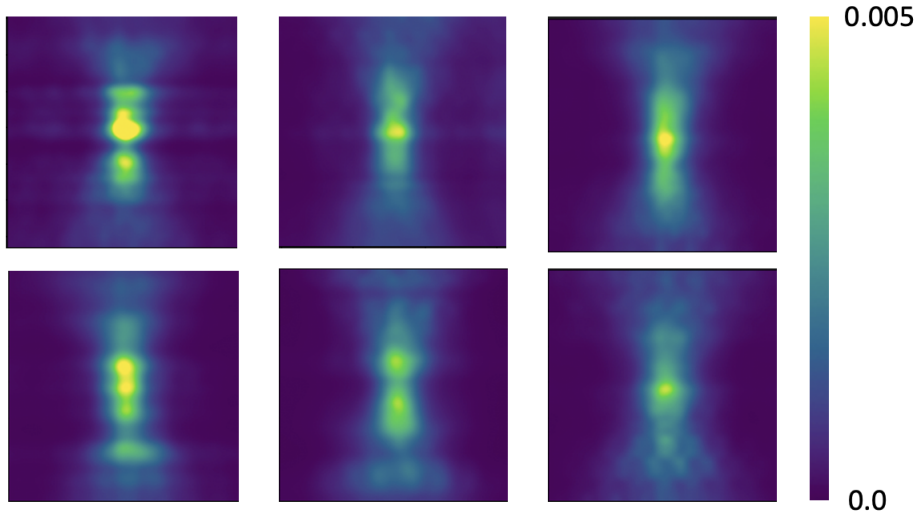


Figure 6.10: Examples of PSF in the database for training network on sagital slices.

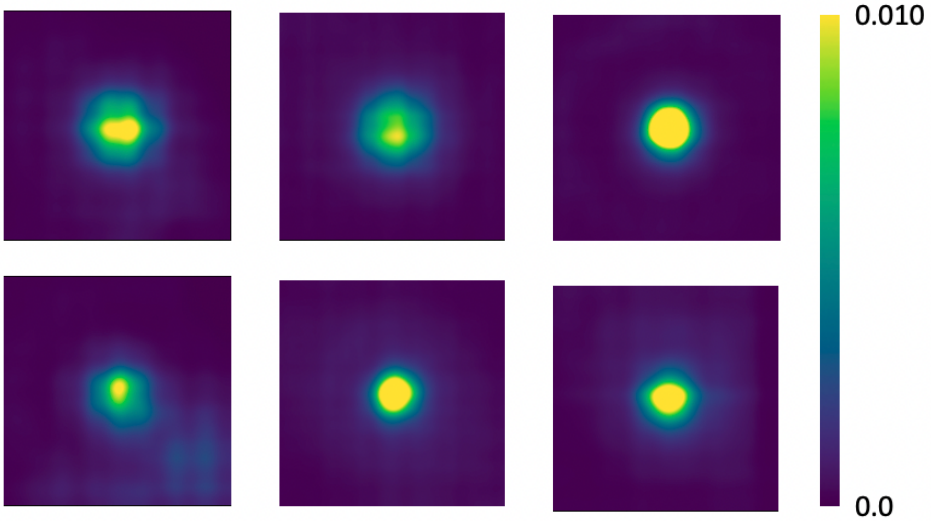


Figure 6.11: Examples of PSF in the database for training network on coronal slices.

of 4 for the UNET and for the two PAVENET blocks, as it does not change performance though it reduces the number of parameters. This can help avoid overfitting especially.

Supervised training of the neural networks is performed with Adam optimization, on 60 epochs for both UNET and PAVENET. For the latter the first 30 epochs are done without the consistency loss for the reasons mentioned in the previous study. The number of epochs is lower compared to the previous study in order to avoid overfitting, see Figure 6.13.

PAVENET is optimized according to (6.4) with Adam optimizer and, except when we mention otherwise, results are shown for 1,000 test-time training epochs corresponding to approximately 40 seconds computation. Parallelization of the test-time training for multiple slices in order to get the 3D volume is possible, since this amounts to considering a batch size of 100 for coronal slices or 192 for sagittal ones. Note also that we performed experiments with TV regularization included in (6.4), but performance was unchanged so we do not present the corresponding results.

A grid-search strategy was used for hyperparameter tuning. The HPs that we considered were the batch size, learning rates for both $\boldsymbol{H}_w$ and $\boldsymbol{G}_\theta$, λ_1, λ_2. The specific values are given in Table 6.4, where we have denoted the UNET trained on coronal (resp. sagittal) by UNET-C (resp. UNET-S), with similar signification for PAVENET-C and PAVENET-S. HP tuning is performed on a validation set consisting of slices from 20 out

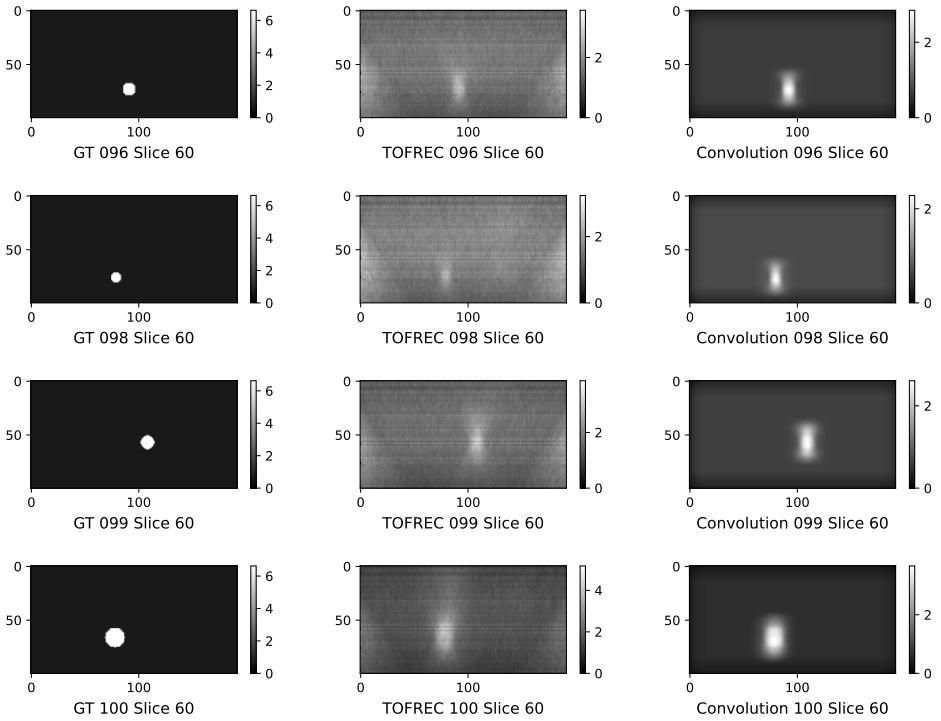


Figure 6.12: Evaluation of the method for creating ground-truth PSF. We show different slices of ground-truth images (left) and the corresponding TOF reconstruction (center). We show the convolution of the ground-truth with the estimated ground-truth PSF in the database (right). The data-fidelity term used to train PAVENET compares the TOF reconstruction with the convolution of the estimated image and PSF.

of the 100 volumes in the training set.

UNET and spatial information One of the objectives of this study is to evaluate the performance of deep learning based methods, thus not only PAVENET. Among existing methods, we saw that we could train UNET in different directions. TOF backprojection is such that the PSF has an large width in the Y direction, i.e across coronal slices. We will see that on 2D coronal slices, the TOF reconstruction includes areas with significant activity though no sphere is located in the same slice for the ground truth. This is due to TOF blurring that is important the Y direction. The contribution of TOF blurring in the overall blur within a coronal slice in the X and Z directions is way less important. This suggests that UNET-S and UNET-C might have a significant difference in performance since the input/ground-truth pairs are not comparable. This also suggests that both could benefit from spatial information in the third dimension. As we do not consider 3D networks in this study, a first solution is to consider a network such as [Xu et al., 2017]. The corresponding architecture is given in Figure 6.14. The major difference with networks described before is the fact that several slices are given as inputs of the network: the slice of interest along with adjacent slices. This allows the 2D network to get spatial information in the third dimension. The architecture is a residual network (ResNet), which means that the slice of interest is added to the output of the encoder-decoder. The latter therefore learns the residual between the ground-truth and the TOF reconstruction. In practice, this especially allows the network to "know" which of the input slices is the one of interest, since ordering of the slices is meaningless in the implementation.

We name such a network trained on coronal slices with spatial information in the Y direction MS-RESNET-C, MS referring to multi-slice. It is also possible to use the same architecture for sagittal slices; we refer to this network as MS-RESNET-S. In this case, the network gets spatial information in the X direction.

In our experiments for the encoder-decoder we have considered a similar architecture compared to the UNET described before. The number of adjacent slices was however considered as a hyperparameter to tune. We observed that performance was improved for 2 adjacent slices (1 on each side) as compared to the classic UNET, and even more with 4 adjacent slices. When increasing this number, no improvement was then observed. This

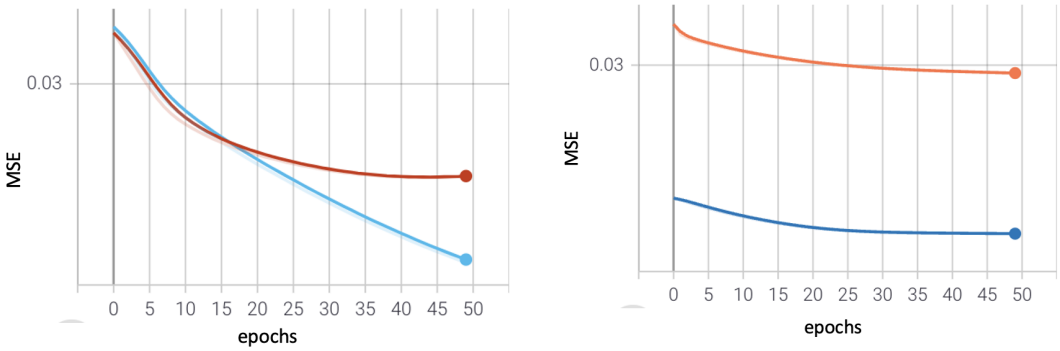


Figure 6.13: Training (blue) and validation (orange) MSE computed between estimated and GT images for PAVENET trained on coronal (left) and sagittal (right) slices. One can observe that MSE is not decreasing after 45-50 epochs in both cases.

Network	LR H_w	LR G_θ	λ_1	λ_2	Batch size	Channels
UNET-C	10^{-3}	-	-	-	8	1
UNET-S	10^{-4}	-	-	-	8	1
PAVENET-C	10^{-7}	10^{-6}	1000	0.01	8	1
PAVENET-S	10^{-7}	10^{-7}	1000	0.01	8	1
MS-RESNET-C	10^{-3}	-	-	-	64	5
MS-RESNET-S	10^{-3}	-	-	-	64	5

Table 6.4: Optimal hyperparameters for each method. These HPs for PAVENET are only related to supervised training. For test-time training, both H_w and G_θ are trained with the cost function given (6.4) and a 10^{-7} learning rate.

is valid for both MS-RESNET-C and MS-RESNET-S, so we consider 4 adjacent slices in the results, which corresponds to 5 input channels (cf Table 6.4). It might seem not totally fair to compare a UNET with a residual network. We therefore tested RESNET with a single input slice; the corresponding performance was decreased in our experiments compared to UNET, so we do not report the results.

An other possibility to make use of the spatial information is to train 3 different networks: one UNET on coronal slices, one on axial and the other on sagittal slices. The considered output is the weighted average of the 3 generated volumes, the average being considered voxel-wise. These weights were optimized on the validation set, and we show results for the optimal observed configuration: 0.45 for axial and sagittal UNETs, and 0.10 for the coronal UNET. This network is referred to as 2.5D UNET in what follows.

MS-RESNET and 2.5D networks could be considered for PAVENET as well but in this study we separate the impact of using PAVENET and test-time training, with the impact of using spatial information by using MS-RESNET or 2.5D. We will discuss in the analysis whether such methods could benefit PAVENET or not.

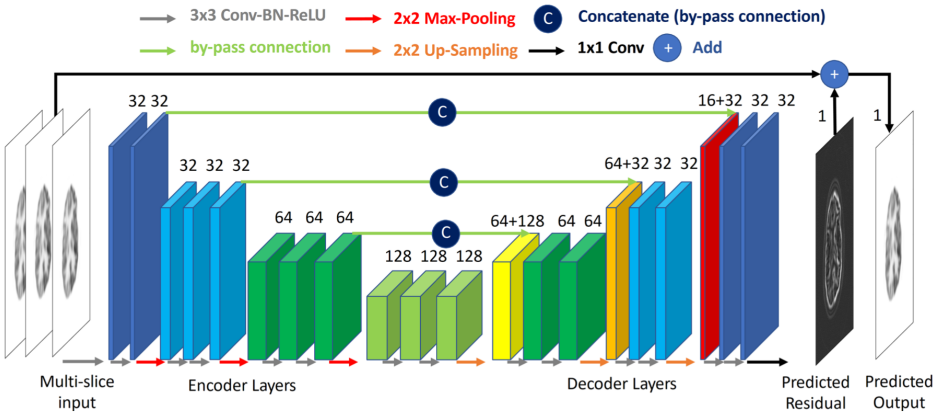


Figure 6.14: Multi-slice ResNet [Xu et al., 2017].

6.4.2.3 Metrics

There are five different metrics that we consider for evaluating the networks. The first one is PSNR; similarly to Chapter 4, this might not be the most appropriate. Indeed, the major interest here is to focus on the detectability of the spheres and the retrieved activity in the areas where spheres are located. Also, there is a high proportion of background in the volumes. PSNR however gives the same importance for every voxel, similarly to MSE. Therefore the areas of interest - the spheres - only account for a tiny part of the overall metric. For instance, we found in test B with 1 mm radius spheres that a reconstruction which retrieves zero sphere still has a PSNR above 40dB as the ground-truth volume is almost empty.

The four other metrics focus on areas corresponding to spheres. First, in the same way as the previous study, we computed the error in terms of CRC (CRC Err.).

In Section 6.1 we mentioned the major criteria of having less than 5% of False Negative Rate (FNR) for detection of SLNs. A way to assess whether a sphere is detected or not is to compute the Contrast to Noise Ratio (CNR). It is defined as

$$CNR = \frac{|act_S - act_{BG}|}{std_{BG}} \tag{6.7}$$

where act_S is the mean activity in the reconstruction within an area where a sphere is located in the ground-truth; act_{BG} and std_{BG} are the mean and standard deviation for the activity in the background of the reconstruction. It corresponds to the ratio of the image contrast (in the ROI) to the noise in the background. A common criteria for a sphere to be considered as detected is to have a CNR higher than 3. Therefore we compute the FNR (%) as the ratio of the spheres not detected to the total spheres, considering the detectability criteria as $CNR > 3$. When computing this ratio, we do not distinguish between volumes from a test set. Instead we consider all spheres within the five test volumes.

For TOF reconstructions, it can be relevant to show the mean and standard deviation of the CNR across all spheres. However for reconstructions obtained with neural networks, we find these values not to be relevant. Indeed neural networks have the ability to remove noise correctly, which makes std_{BG} close to zero in some areas. Value of CNR for some spheres are therefore very high so taking the mean value across all spheres gives biased information. Rather, we show results considering only the numerator of the CNR, i.e the contrast retrieved in an area of interest. As the ground-truth contrast value is known (difference between hot spheres and background activity in the simulations), we show the contrast error as the relative absolute difference with the ground-truth contrast as

$$CNR_{corr} = \frac{|(act_S - act_{BG}) - cst_{GT}|}{cst_{GT}} \tag{6.8}$$

where cst_{GT} is the known ground-truth contrast. This allows further evaluating the ability of neural networks to detect spheres, with more precision compared to only using FNR which relies on a threshold.

Finally, we also show results for the False Positive Rate (FPR), i.e the ratio of spheres retrieved in the reconstruction that are not in the ground truth. We recall the guideline from ASCO which is to have an identification rate of more than 85% i.e less than 15%

FPR. We use the Otsu thresholding method [Otsu, 1979] for the segmentation of the volumes so that we can assess whether a sphere is generated by the network (or TOF reconstruction) or not.

Each test set is composed of 5 test volumes. For each of these tests, PSNR is computed for each volumes, so results are shown with inter-volumes standard deviation (SD). CNR_{corr} and CRC_{err} are computed on each sphere in the 5 test volumes. Results are shown for inter-spheres SD in these cases. For the FNR and the FPR, we only represent the ratio, computed considering all spheres in the 5 test volumes.

6.4.3 Results

6.4.3.1 PAVENET vs UNET

Here we compare the performance of PAVENET with UNET and the TOF reconstruction reference. Figure 6.15 shows a slice reconstructed with different methods, on test A with spheres with radius between 1.5 mm and 5 mm. We also represent the estimated PSF from PAVENET-C, after 1000 test-time training iterations. One can observe that TOF blurring leads to areas in the TOF reconstruction with high activity compared to the background, though no ground-truth sphere is located at the corresponding depth. At the considered depth, 2 out of 3 ground-truth spheres are retrieved in UNET-C reconstruction. All of the spheres are visible in the slice reconstructed with PAVENET-C.

Figure 6.16 shows two different sagittal slices (top and bottom row) obtained with differents methods including UNET-S and PAVENET-S. The top row shows a particular case where two spheres are almost touching in the ground-truth. In the TOF reconstruction one can observe that blurring in the Y direction makes those spheres indistinguishable. None of the neural networks are able to distinguish between those spheres either. The bottom row shows a case where reconstruction seems accurate for both UNET-S and PAVENET-S. The corresponding PSF retrieved by PAVENET-S are shown in Figure 6.17.

Table 6.5 shows metrics computed on all spheres from test A. The performance of both deep learning based methods is significantly better compared to the TOF reconstruction. We observe only slight difference for CRC values. Decreased performance for PSNR can be observed for PAVENET-S compared to UNET-S.

There is a 4.8 % improvement with PAVENET compared to UNET for the FNR, when networks are trained on coronal slices. This allows PAVENET-C to have a FNR

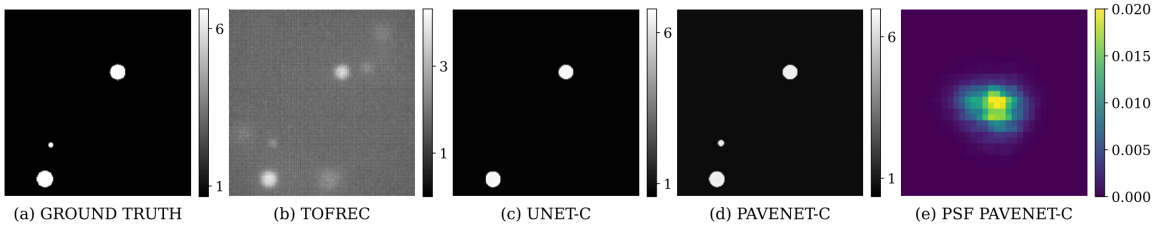


Figure 6.15: Example of a coronal slice obtained with different methods. Both UNET and PAVENET are trained on coronal slices here. Image and PSF for PAVENET-C are obtained 1000 iterations of test-time training.

below 5%, considering the criteria $CNR > 3$. For comparison, the TOF reconstruction has a 22.5% False Negative Rate. Note also that we found in our experiments that all spheres of radius equal or above 2 mm are detected by the deep learning based methods. For the TOF reconstruction, all spheres with radius 3 mm or larger are detected. All spheres are detected for networks trained on sagittal slices.

The amelioration of sphere detectability for PAVENET-C vs UNET-C can also be seen with CNR Corr. Indeed the retrieved contrast is 10 % closer to the ground truth in average. Note that there is a high standard deviation for this metric since it is computed on spheres with radius ranging from 1.5 mm, for which CNR Corr. might be high, to 5 mm for which CNR Corr. is close to 0 %. We also notice that even though UNET-S allows the detection of all spheres w.r.t the CNR criteria, the retrieved contrast is more accurate for PAVENET-S, with a mean relative contrast difference (CNR Corr.) of 7.5% compared to 16.1 % for UNET-S.

The results of a statistical study that focuses on spheres of 1.5 mm are shown in Figure 6.18. We do not represent the FNR for UNET-S and PAVENET-S since it is 0 %. One can notice that whether networks are trained on coronal or sagittal slices, the performance is improved when using PAVENET. Especially the mean relative contrast difference is 84.7

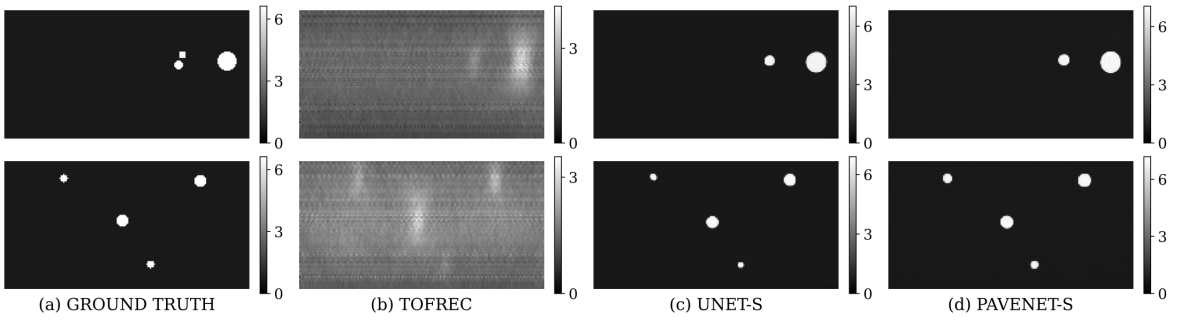


Figure 6.16: Example of sagittal slices obtained with different methods. Both UNET and PAVENET are trained on sagittal slices here. Images for PAVENET-C are obtained 1000 iterations of test-time training. Each row corresponds to a similar slice.

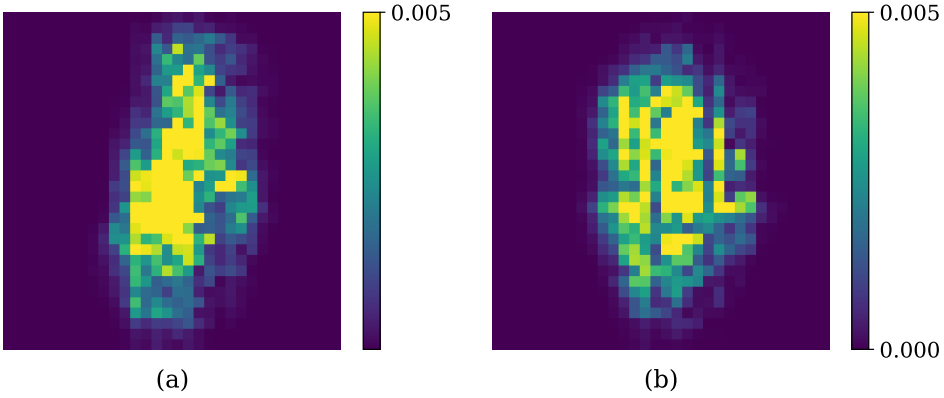


Figure 6.17: Example of PSF obtained with PAVENET-S after 1000 iterations. (a) corresponds to the slice in top row of Figure 6.16 and (b) to the slice in bottom row.

	PSNR	CRC Err. (%)	CNR Corr. (%)	FNR (%)	FPR (%)
TOFREC	14.98 ± 0.11	71.7 ± 0.3	84.2 ± 6.7	22.5	100.0
UNET-C	29.66 ± 0.61	2.3 ± 0.1	34.7 ± 29.9	9.6	7.6
PAVENET-C	30.49 ± 0.22	3.4 ± 0.3	24.7 ± 26.8	4.8	6.0
UNET-S	**32.43 ± 0.39**	1.1 ± 0.1	16.1 ± 20.6	**0.0**	**1.2**
PAVENET-S	30.48 ± 0.43	**0.9 ± 0.3**	**7.5 ± 14.1**	**0.0**	1.5

Table 6.5: Metrics on Test A for spheres of radius between 1.5 mm and 5 mm. UNET-C and PAVENET-C have been trained on coronal slices, while UNET-S and PAVENET-S have been trained on sagittal slices.

% for UNET-C and 70.8 % for PAVENET-C. It is 57.0 % for UNET-S and 31.4 % for PAVENET-S which is a significant improvement for the sphere detectability.

Finally, the False Positive rate is below 8% for all UNET and PAVENET, which is satisfying considering the 85% identification rate criteria. Note that we find a 100% FPR for TOF reconstruction because our computation relies on a segmentation which quality is quite poor on those volumes.

Overall, better performance can be observed for networks trained on sagittal slices compared to those trained on coronal slices, for spheres between 1.5 mm and 5 mm radius in Test A.

Test B consists of spheres with 1 mm radius. We observe in our experiments that neither UNET-C nor PAVENET-C is able to detect such small spheres.

PAVENET-S is however able to retrieve 10.4 % of the spheres, vs only 3.5 % for UNET-S. Especially, it is interesting to note that the FNR decreases with the number of test-time training iterations as shown in Figure 6.19. The CRC value is also better for PAVENET in this case with 46.2% error after 1000 iterations, compared to 69.4% for

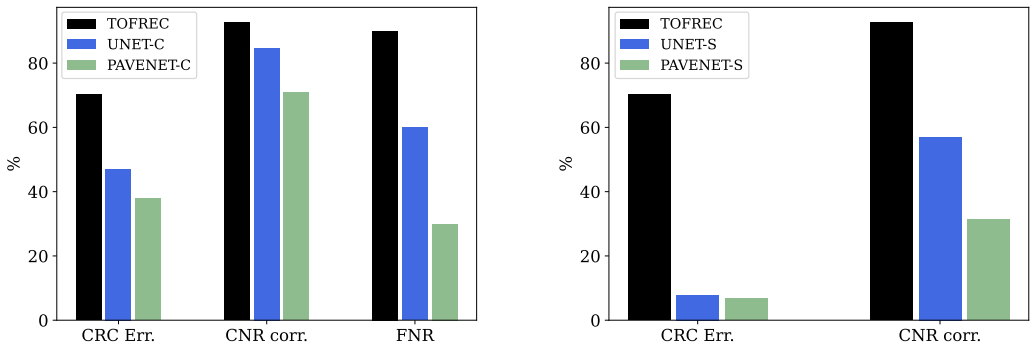


Figure 6.18: Comparison of CRC Err., CNR Corr. and FNR for TOFREC, UNET and PAVENET. Metrics are computed on Test A, only for 1.5 mm radius spheres. All metrics are in percentage. FNR is not represented fornetworks trained on sagittal slices since it is 0 % for both UNET-S and PAVENET-S.

UNET-S. We have not tested more iterations so far so we are currently not able to tell if the improvement in FNR continues with more iterations. Note that too many iterations is not necessarily desirable as we aim for fast reconstruction methods for intraoperative imaging.

6.4.3.2 UNET and variations

Here we assess the performance of networks trained with spatial information in the third dimension, whether it is for MS-RESNET or 2.5D.

We show metrics on Test A in Figure 6.6. No significant improvement from MS-RESNET-S compared to UNET-S can be observed. Multi-slice input strategy however improves metrics when considering networks trained on coronal slices. There is a 4.8 % decrease in the FNR for MS-RESNET-C compared to UNET-C. Here also all spheres larger than 2 mm radius are detected. The improvement of 4.8 % FNR overall therefore corresponds to a FNR reduced from 60 % to 30 % on 1.5 mm spheres. These values are however higher compared to UNET-S and MS-RESNET-S for which all spheres are detected. More generally, there is no metric for which MS-RESNET-C has a better performance compared to networks trained on sagittal slices.

As for the 2.5D network, the only observed improvement is on PSNR. Note that all spheres are also detected with 2.5D.

For intra-operative imaging, the operator is placed above the patient, so slices of inter-

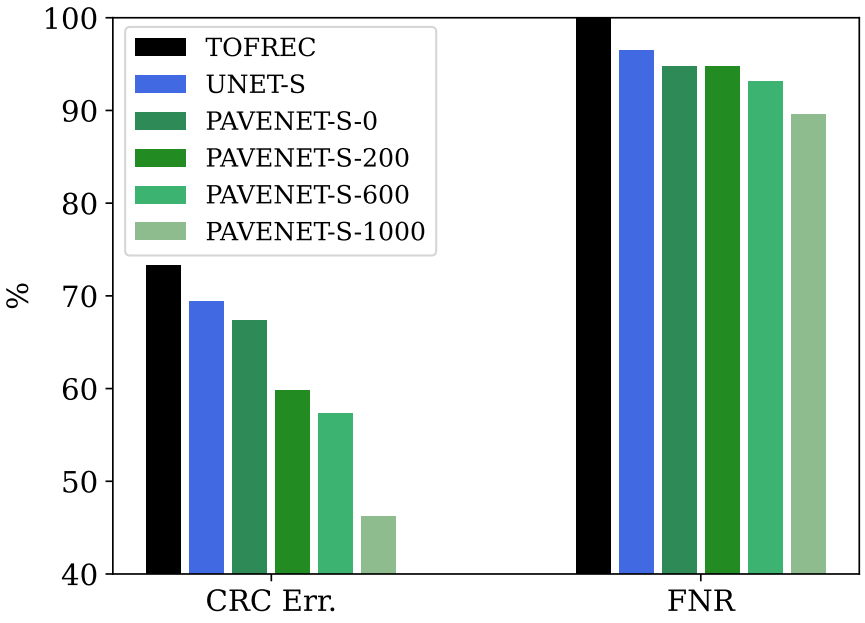


Figure 6.19: Comparison of CRC Err. and FNR for TOFREC, UNET and PAVENET. The networks are trained on sagittal slices. Results for PAVENET are given depending on the number of test-time training iterations. Metrics are computed on Test B for 1 mm radius spheres. All metrics are in percentage.

	PSNR	CRC Err. (%)	CNR Corr.	FNR (%)	FPR (%)
TOFREC	14.98 ± 0.11	71.7 ± 0.3	84.2 ± 6.7	22.5	100
UNET-C	29.66 ± 0.61	2.3 ± 0.1	34.7 ± 29.9	9.6	7.6
MS-RESNET-C	31.43 ± 0.37	3.4 ± 0.7	19.4 ± 27.7	4.8	2.8
UNET-S	32.43 ± 0.39	1.1 ± 0.1	**16.1 ± 20.6**	**0.0**	**1.2**
MS-RESNET-S	32.91 ± 0.64	**0.7 ± 0.3**	17.4 ± 22.8	**0.0**	2.8
UNET 2.5D	**34.02 ± 0.20**	6.1 ± 0.8	17.9 ± 18.4	**0.0**	2.6

Table 6.6: Metrics on Test A for spheres of radius between 1.5 mm and 5 mm. Results are shown for UNET trained on coronal slices (UNET-C), sagittal slices (UNET-S) and on different strategies to include the third spatial dimension.

est might in this case be the coronal ones. We show in Figure 6.20 an example of coronal slice reconstructed with all methods, PAVENET included. We note a clear improvement in the visual aspect of hot spheres for networks trained on coronal slices compared to those trained on sagittal slices. For the latter, all present spheres are detected, and the False Positive observed in UNET-C is less visible, but the retrieved spheres are less sharp. This is an element to balance the improved performance in terms of metrics for networks trained on sagittal slices.

Note that we can also observe similar results when visualizing sagittal slices as in Figure 6.21: spheres in reconstructions from networks trained on sagittal slices are sharper.

6.4.4 Analysis and conclusion

6.4.4.1 Performance of PAVENET

Regarding metrics, the only observed drawback of PAVENET is the drop of performance in terms of PSNR for PAVENET-S. As mentioned, PSNR is not the most relevant metric for the type of data that we study here. This metric is indeed largely impacted by background voxels, which are of limited interest. An explanation of the drop of performance for PAVENET is that the data-fidelity term used for test-time training leads to appearance of noise in the background in PAVENET reconstruction, due to noise visible in the TOF reconstruction. This reduces the PSNR, but we are more interested on metrics specifically related to hot spheres, e.g evaluation of the performance in terms of nodes detection and activity quantification.

For networks trained on coronal slices, improved performance for PAVENET over UNET is clear. It seems especially hard for the UNET to distinguish between slices containing actual spheres or slices where areas with high activity are due to spheres located in adjacent slices, and visible in the TOF reconstruction due to blurring in the Y direction. Test-time training allows PAVENET to correct for such errors in the predictions. TTT can only be efficient if the estimated PSF is accurate, which seems to be the case in our experiments. When spheres are too small - 1 mm radius -, PAVENET-C is however not able to detect them.

Improvement of the performance linked to PAVENET is also visible when networks are

trained on sagittal slices even if results from UNET-S are significantly better compared to UNET-C. For 1.5 mm radius or larger spheres, UNET-S has a 0 % FNR; this is particularly due to the ability of the network to smooth background so the CNR is higher than the considered threshold for detectability. We however observed that the contrast retrieved for spheres with 1.5 mm radius is more important for PAVENET-S as shown by CNR Corr.

An interesting point is the FNR decrease w.r.t test-time training iterations for PAVENET. This emphasizes the fact that test-time (or self-supervised) training allows overcoming limits of supervised methods. We recall the fact that the efficiency of test-time training is only enabled when using an adequate architecture. Including a correctly estimated kernel in the consistency loss is necessary as shown in experiments in 6.3.1. This is the reason why we did not try a self-supervised strategy for a simple UNET architecture here.

Overall, whether it is trained on coronal or sagittal slices, we have found PAVENET able to improve the performance of UNET, thanks to a combination of test-time training with an architecture adapted to the task.

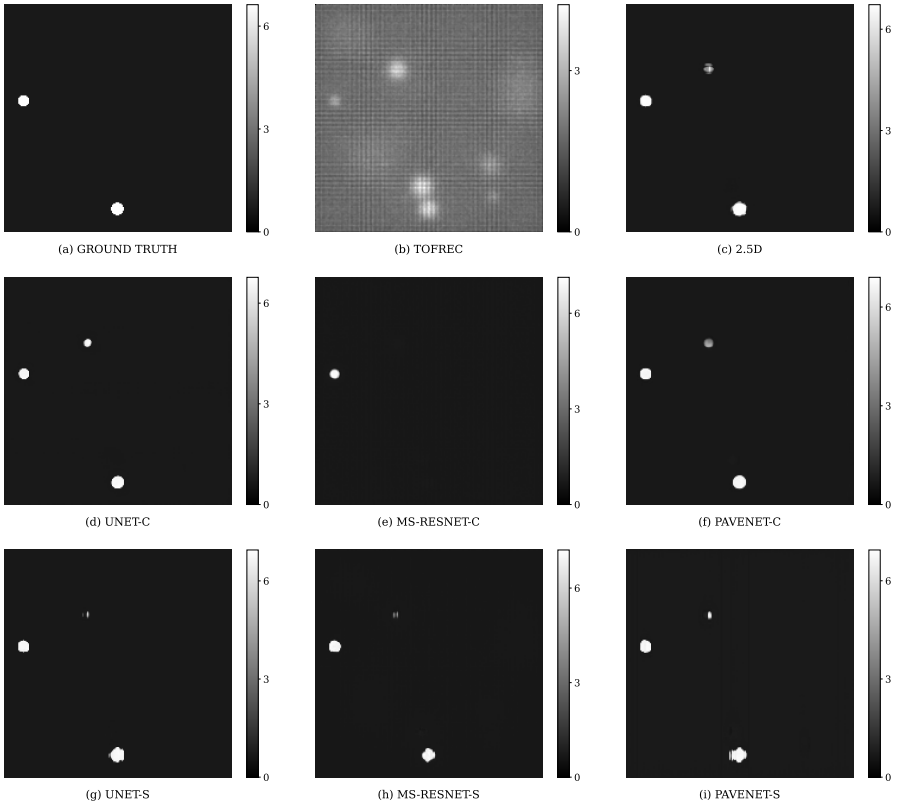


Figure 6.20: Example of a coronal slice obtained with all methods. Images for PAVENET are obtained with 1000 iterations of test-time training.

6.4.4.2 Impact of the training strategy for UNET

On metrics, no improvement is observed for the multi-slice input strategy when networks are trained on sagittal slices, whereas it is the case when training is performed on coronal slices. This is mostly due to the fact that the PSF is essentially oriented in the Y direction due to TOF blurring. We especially considered 300 ps CTR in these experiments which is relatively high and makes TOF blurring significant compared to resolution blurring. Also, positron range is not considered in GATE simulations which reduces the blur in X and Z directions even more. Therefore, the spatial information in the Y direction given to MS-RESNET-C is particularly important, whereas we observe that it has almost no impact for MS-RESNET-S.

The 2.5D network only improves the PSNR. The main reason is that averaging voxels from 3 different predictions has a smoothing effect in the reconstructions. Otherwise, we do not observe improvements for metrics of interest related to hot spheres.

6.4.4.3 Overall analysis

A first and major result is that deep learning based methods significantly improve the tested metrics compared to the initial TOF reconstruction. An improvement of at least 2 mm in the diameter for which all spheres are detected is observed with these methods. Indeed TOF reconstruction does not allow to retrieve all spheres between 4 mm and 6 mm diameter, while it is the case for all of the networks. We recall that no over-sampling is performed on the TOF reconstructions in our experiments; over-sampling could potentially improve the metrics, though it significantly increases the computation

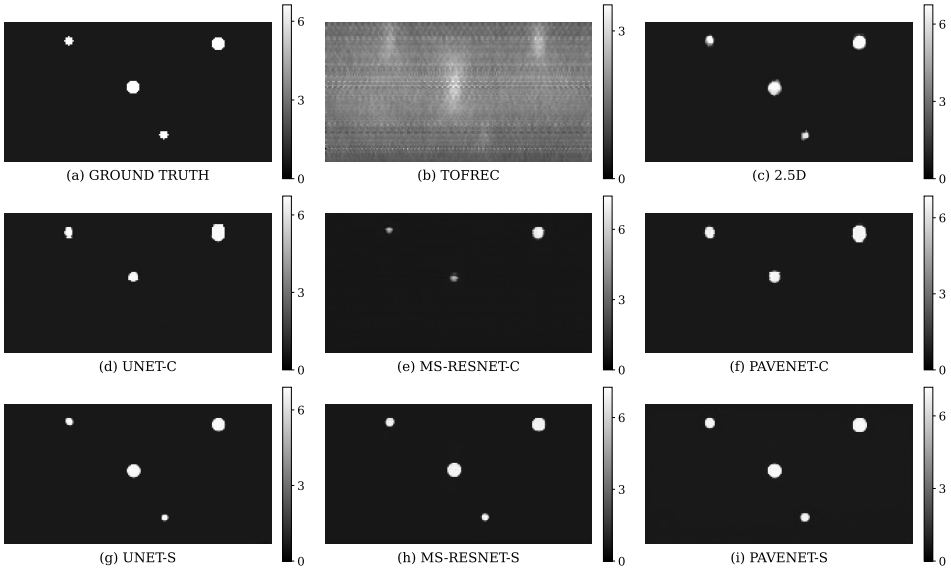


Figure 6.21: Example of a sagittal slice obtained with all methods. Images for PAVENET are obtained 1000 iterations of test-time training.

time on CPU (no GPU implementation is currently available for the reconstruction code).

Even if MS-RESNET-C has a better performance compared to UNET-C, networks trained on coronal slices are overall less performant than those trained on sagittal slices regarding all metrics. This can be explained by the improved quality of the input given to the network, since 2D coronal slices from TOF reconstruction are of poor-quality due to TOF blurring in the Y direction. In practice however it is more likely that coronal slices are the ones of interest. Visual assessment of coronal slices shows that hot spheres look sharper for networks trained in the corresponding direction. This might be a criteria to balance the overall better performance for networks trained on sagittal slices.

If visual aspect is an important criteria, a trade-off could be to use either MS-RESNET-C or PAVENET-C since visual quality is satisfying on coronal slices and overall performance regarding metrics is enhanced compared to UNET-C. In our experiments both networks have the same FNR, and CNR Corr. is 5 % better for MS-RESNET-C. It would therefore be interesting to test a multi-slice input strategy for PAVENET-C. Such a strategy is less likely to benefit PAVENET-S; indeed we observed no significant difference between UNET-S and MS-RESNET-S so multi-slice inputs in this direction does not seem to improve the reconstruction.

Finally, the performance of neural networks need to be balanced. Indeed, TOF reconstruction does not rely on prior training with some database. Supervised methods (UNET, MS-RESNET, 2.5D) show good performance on the testing sets but they have been trained on similar images. For different applications, it is likely that the training strategy should be adapted (e.g other training database) to reach similar performance, whereas TOF reconstruction is not affected by such a change in the testing data. PAVENET is built so that it can adapt to new data, thanks to self-supervised learning. For now we have only assessed the robustness of the network regarding distribution shift by simply testing it with different spheres sizes. This currently does not ensure that it can adapt to any data, e.g more or less noise in the images, different structures, etc. The influence of pre-training is also not entirely evaluated. We know that it clearly accelerates convergence for self-supervised training; however it is not clear whether self-supervised training would also be efficient when the data is of different nature compared to supervised data used for pre-training. This is an aspect to elaborate as we will mention in the perspectives.

6.4.4.4 Conclusion

These experiments on GATE simulations show that deep learning based methods have the potential to meet the objectives for nodes identification with the studied TOF intra-operative imaging system. Robustness of such methods however still need to be assessed on experimental data.

The consideration of spatial information in the third dimension for 2D networks was shown to be useful only when working on coronal slices. However regarding the metrics of interest used for this study, it seems preferable to train neural networks on sagittal slices (if 3D is not an option).

We also showed that PAVENET is able to overcome the limits of a post-processing UNET, especially thanks to test-time training. The ability of the network to retrieve smaller spheres is particularly promising. The second step of the experimental proce-

dure to validate such a method being satisfied, tests can now be performed on more complex/realistic data and it will be discussed in the perspectives.

PAVENET is still a network that needs to be studied further; especially we will mention in the next section methods that could improve the network's performance and robustness, as well as ways to evaluate the network in a more comprehensive manner.

6.5 Perspectives

The elements that we mention in this section either concern methods that can be included for comparison or that can be incorporated to PAVENET to improve its performance. We also mention tests that can be performed to further assess the efficiency of such a network. Some of the mentioned tests and methods are work in progress, others are perspectives for the future.

6.5.1 Current and planned tests

Our first priority is to finalize the experimental protocol that we proposed in order to validate PAVENET. For this we will train neural networks on more realistic and complex acquisition data from GATE simulations. This includes e.g lower activity to background ratio, larger phantom depth and lower crystal thickness. We will also test the performance of the networks on experimental data obtained in [Sajedi et al., 2022].

Tests in order to further evaluate the ability of the networks - especially PAVENET - to adapt to new configuration could also be performed. This includes tests on already obtained experimental data, but also tests with:

- Different activity ratio compared to the supervised training configuration. Also varying activity across spheres might be considered.
- Different phantom depth compared to supervised (pre-)training.
- Top detector moved vertically, which modifies the true PSF similarly to Test D in experiments 6.3.1.

These tests seem to be indeed more complete compared to only modifying the size of the hot spheres in order to evaluate networks' performance w.r.t distribution shift.

6.5.2 Spatially-variant PSF

In these experiments we computed a spatially fixed PSF. We gave a visual assessment of the accuracy of this model in Section 6.4, but the system's geometry is such that spatially variant PSF might be more accurate for the forward model.

If so, we note two things that need to be performed so that PAVENET can be adapted to such PSF:

1. Estimate the ground-truth PSFs considering sub-regions in the field of view.
2. Adapt the architecture of PAVENET so that it is able to retrieve a different PSF depending on the image sub-region.

However before going into this direction, we need to assess whether significant PSF deformations are observed across the FOV.

6.5.3 Others

There are several methods that might offer benefits to reconstructions based on deep learning for the studied imaging system. The first one is the fact to use 3D networks. Indeed our experiments showed the limits of missing one spatial dimension for training the networks. With similar architecture, 3D networks have more parameters so the risk of overfitting is high for supervised methods. This is even more true considering the limited size for the database, which is a limit that is hard to overcome due to high memory and time related to the database creation. Considering a 3D architecture for PAVENET can on the other side become a computation burden if one wants to keep a fast reconstruction method.

Another aspect that could be studied further is the impact of the number of test-time training iterations. The "right" choice for early stopping is actually dependent on the learning rate used during self-supervised learning. It can also be dependent on some regularization considered on the predicted image and kernel. The upper limit for the number of iterations can be determined by the desired reconstruction time. As for the rest, it is likely that the optimal number of TTT iterations and learning rates are dependent on the input data.

We saw that test-time training increases the performance of neural networks. It can only be true if the PSF prediction is accurate: otherwise the consistency loss is irrelevant. Study on the optimal architecture for the PSF generator could therefore be useful, along with methods to better estimate the ground-truth PSF before supervised training. For instance, the estimated ground-truth PSF for slices where no sphere is present might not be relevant; giving such groud-truth to PAVENET during pre-training can in this case reduce its performance. Another strategy to overcome the limit of having to estimate ground-truth PSF is to perform pre-training with known simulated PSF, and then perform test-time training with actual acquisitions data. Currently we have not performed experiments that could confirm that pre-training accelerates convergence of self-supervised learning in this case.

Overall, PAVENET seems to offer the possibility to better adapt to new data, but robustness on different acquisition configurations still needs to be studied further.

Conclusion

In this thesis we have studied deep learning based methods for tomographic reconstruction in CT and PET imaging. We have explored their potential to overcome some of the limits encountered with analytical and iterative methods. The analysis of their performance was, when possible, driven towards medical purposes.

In the first part we have compared the performance of three selected iterative algorithms on simulated and experimental data. This simple study showed that the choice of a particular algorithm and its related hyper-parameters depending on the data characteristics, e.g the nature of the noise, is a tedious task in practice. This choice has a high impact on the reconstruction performance. The computational cost is an other major drawback of iterative algorithms.

In the second part we used the example of bone microarchitecture imaging to see how the transcription of quantitative parameters in the reconstructed image can be affected by the learning method, when considering supervised data. Supervised methods rely on a feedback mechanism that involves comparing an estimated image with the corresponding ground-truth: we have shown that the way to assess the quality of the estimated image during training has a strong impact on the accuracy of the reconstructed image at test-time. Especially, the study of pixel-wise, perceptual and adversarial losses in our experiments emphasized the need to evaluate the performance of neural networks with respect to metrics related to diagnosis and quantification of pathologies.

Finally, we saw that the structure of a neural network along with the training method should be adapted to the application. Study on simulations of TOF PET data showed the need for designing innovative solutions to overcome limits of existing neural networks for the considered intraoperative imaging system. We proposed PAVENET to satisfy the main criteria imposed by the application: having a fast reconstruction that is robust to different data and acquisition settings. On Monte-Carlo simulations, we showed that the network has a better ability to retrieve small spheres from TOF PET acquisitions compared to UNET. We also showed enhanced performance in terms of robustness regarding distributions shifts for the data. Further experiments on more realistic settings for the acquisitions along with tests on experimental data need to be performed to validate the solution. These experiments also highlighted the potential of deep learning based methods for TOF PET imaging, since they allow significantly reducing the False Negative Rate for nodes detection compared to TOF backprojection. These results have huge interest in the context of breast cancer staging but the robustness of such methods still needs to be further assessed.

Overall, we saw how advances in the computer vision field with neural networks can translate into advances in the tomographic reconstruction field. One can therefore expect that the continuous improvements in deep learning can largely benefit the medical imaging

research community. Especially, methods such as generative adversarial networks or self-supervised learning could be a major step forward for tomographic reconstruction in the future.

Synthèse

Dans cette thèse nous nous intéressons à la reconstruction tomographique. Dans un contexte médical, la tomographie est ce qui permet d'observer l'intérieur d'un patient à partir de données d'acquisition. Pour le diagnostic, le traitement ou le suivi de l'évolution d'une pathologie, il est indispensable que l'algorithme de reconstruction soit précis afin d'assister le radiologue du mieux que possible dans sa prise de décision. D'un point de vue mathématique, la reconstruction tomographique revient à résoudre un problème inverse.

L'objectif de cette thèse est d'étuder les méthodes d'apprentissage profond pour la reconstruction tomographique en CT (Computed Tomography, ou tomodensitométrie TDM) et en TEP (Tomographie par Emission de Positons). Nous mettons notamment l'accent sur l'évaluation de la performance des réseaux de neurones par rapport à des critères propres à l'application médicale.

Nous proposons également une méthode d'apprentissage profond pour résoudre des problèmes de déconvolution aveugle. La fonction d'étalement du point (PSF) étant difficile à estimer pour l'imagerie d'émission, notre méthode présente un intérêt particulier pour la TEP. Nous montrons des résultats préliminaires sur un dispositif d'imagerie péropératoire TEP qui a pour objectif de déterminer le statut métastatique des ganglions lymphatiques pour le cancer du sein.

Reconstruction tomographique en TEP et TDM: résolution d'un problème inverse

Avant d'expliquer les méthodes de résolution pour le problème de reconstruction tomographique, nous donnons ici les éléments nécessaires à sa formulation pour la TEP et la TDM. Dans les deux cas, il s'agit de résoudre un problème inverse.

Problème inverse

La formulation générale du problème inverse que nous considérons peut s'écrire de la façon suivante:

$$p = A(f) \tag{1}$$

où A est un opérateur qui transforme un volume d'intérêt f en ses projections p. Nous considérons A comme étant linéaire dans cette thèse.

Reconstruction tomographique pour les différentes modalités d'imagerie

En tomographie, un détecteur est utilisé pour mesurer l'intensité d'un rayonnement après qu'il ait intéragi avec un objet d'intérêt et son environnement. La source du rayonnement

peut être placée autour de l'objet, le détecteur mesurant ainsi l'atténuation de ce rayonnement par la matière. Pour la tomographie d'émission en revanche, la source du rayonnement est interne à l'objet d'intérêt.

TDM Les scanners CT utilisent un tube à rayons X tournant ainsi qu'un ensemble de détecteurs permettant la mesure de l'atténuation par les tissues du corps. Les mesures d'atténuation, obtenues à partir de positions angulaires différentes, sont traitées par ordinateur à l'aide d'algorithmes de reconstruction.

Si l'on considère un faisceau source en position $r = 0$ et une intensité mesurée I_1 pour un détecteur situé en $r = L$, nous pouvons introduire la notion de projection P_ℓ, qui dépend de l'atténuation linéaire $u(r)$, telle que:

$$P_\ell = \log(\frac{I_0}{I_1}) = \int_0^L u(r)dr. \tag{2}$$

Ainsi, en calculant le logarithme du ratio entre l'intensité initiale et l'intensité reçue, nous pouvons mesurer l'intégrale de l'atténuation inconnue $u(r)$ à travers la ligne de longueur L pour chaque paire source-détécteur.

TEP La TEP est une technique d'imagerie médicale qui permet d'obtenir des informations fonctionnelles sur les tissues imagés. Le principe est d'injecter une paire vecteur-radionucléide au patient. Le rôle du vecteur est de se fixer sur des régions d'intérêt (une tumeur par exemple); la désintégration radioactive du radionucléide permet ensuite de localiser l'objet d'intérêt et de quantifier son activité.

Le principe de la TEP repose sur la détection simultanée de deux photons obtenus à la suite de l'annihilation du positon émis avec un électron du milieu. Ces deux photons étant détectés en coïncidence, les données d'acquisition en TEP correspondent à un ensemble de lignes de réponse (LOR). A partir de ces lignes de réponse, un algorithme de reconstruction permet de retrouver la concentration du traceur dans l'espace, notée $\lambda(r)$.

On note t la distance entre une certaine LOR et le centre de l'anneau de détection, et θ l'angle formée par la direction orthogonale à la LOR par rapport à l'origine. Nous pouvons dans ce cas exprimer l'activité totale détectée sur cette même LOR comme:

$$R\lambda(\theta,t) = \int_{\ell(\theta,t)} \lambda(r)dr \tag{3}$$

où $\ell(\theta,t)$ correspond à la LOR à laquelle nous nous intéressons. Le problème est donc sensiblement similaire à celui de la tomodensitométrie. Les détecteurs ici mesurent l'activité totale sur une ligne et non pas l'atténuation totale sur cette même ligne comme en TDM.

Reconstruction analytique

La transformée de Radon transforme une fonction en l'ensemble de ses intégrales sur des hyperplans de $\mathbb{R}^n$. Pour tout vecteur unité $\vec{\alpha} \in \mathbb{R}^n$ et pour tout $t \in \mathbb{R}$, la transformée de Radon d'un objet f s'écrit

$$Rf(\vec{\alpha},t) = \int_{\vec{v}.\vec{\alpha}=t} f(\vec{v})d\vec{v}. \tag{4}$$

Ainsi, $Rf(\vec{\alpha},t)$ est l'intégrale de f sur un hyperplan orthogonal au vecteur $\vec{\alpha}$. Dans ce que nous avons présenté précédemment, le vecteur $\vec{\alpha}$ correspond au vecteur orthogonal à la ligne entre deux détecteurs.

Puisque nous pouvons décrire l'ensemble des mesures obtenues à partir de f à l'aide la transformée de Radon Rf, la question est de savoir s'il est possible de résoudre le problème inverse visant à retrouver f à partir de Rf.

Rétroprojection La rétroprojection consiste à reporter les projections $Rf(\theta,t)$ dans l'espace image. Pour chaque angle θ, on ajoute ainsi la valeur $Rf(\theta,t)$ à chaque point de la ligne $l(\theta,t)$, c.a.d à chaque $(x,y) \in \mathbb{R}^2$ tels que $t = x\cos\theta + y\sin\theta$.

La rétroprojection correspond à l'adjoint R^* de la transformée de Radon mais pas à son inverse R^{-1}. Elle permet de retrouver la fonction inconnue f convoluée avec un noyau de lissage. En comparaison avec les hautes fréquences, un grand nombre de basses fréquences sont reportées dans la rétroprojection, ce qui donne un aspect lisse à l'image reconstruite.

Rétroprojection filtrée La rétroprojection filtrée (FBP) permet de ne pas recourir à l'interpolation des valeurs dans le domaine fréquentiel, nécessaire en raison du nombre fini de projections en pratique. La FBP procède en deux étapes: filtration des projections, puis rétroprojection.

La FBP correspond à l'inverse de la transformée de Radon. Si nous avons accès à l'ensemble des projections dans toutes les directions, il est alors possible de reconstruire exactement f.

Objectifs

Dans cette thèse nous considérons plusieurs problèmes inverses pour la reconstruction tomographique, notamment pour la TDM et la TEP. L'objectif commun correspond à la résolution d'un problème inverse en considérant un opérateur linéaire, à savoir la transformée de Radon. Les reconstructions analytiques telles que la rétroprojection et la FBP représentent une première solution à ce problème. Néanmoins ce dernier est en pratique mal posé, ce qui rend la qualité de ce type de reconstruction limitée, notamment en présence de bruit et/ou de données sous-échantillonnées. Des méthodes plus robustes au caractère mal posé du problème ont été étudiées au fil des années: les méthodes itératives et, plus récemment, les méthodes basées sur l'apprentissage profond.

Méthodes itératives et limites

Les méthodes itératives correspondent à des algorithmes qui ont pour but d'estimer et de mettre à jour, de façon itérative, une solution $\hat{f}$ afin de minimiser une fonction coût. Ces méthodes sont généralement plus robustes face aux perturbations des données. Elles permettent notamment de prendre en compte la physique de l'acquisition, le modèle pour le bruit et également un potentiel *a priori* sur la solution recherchée dans le cas des méthodes régularisées.

De nombreux algorithmes itératifs existent pour la reconstruction tomographique, et le choix de l'un d'entre eux dépend en partie de la fonction coût considérée, mais

aussi du temps de calcul, ses propriétés de convergence, et de la stabilité numérique de l'algorithme.

La fonction à minimiser peut s'écrire de la façon suivante:

$$\min_f d(Af, p) + \alpha R(f) \tag{5}$$

où d est un terme d'attache aux données, $R(f)$ est un terme de pénalité permettant de retranscrire la connaissance *a priori* des caractéristiques de l'image à reconstruire, et α est un paramètre de pondération.

Nous avons réalisé une étude comparative de la performance de différents algorithmes itératifs lorsque les données sont entâchées à la fois d'un bruit Gaussien et également d'un bruit de type Poisson. Les algorithmes que nous comparons sont: SIRT-FISTA-TV et EM-FISTA-TV qui sont des versions régularisées (TV pour variation totale) et accélérées de SIRT et MLEM respectivement, ainsi qu'une version préconditonnée de l'algorithme de Chambolle-Pock (PCP) pour un terme d'attache aux données type Kullback-Leibler et une régularisation TV.

Dans cette étude nous observons que, comme attendu, SIRT-FISTA-TV est plus adapté dans les cas où le bruit Gaussien est élevé, alors que PCP et EM-FISTA-TV sont plus performants quand le bruit Poisson domine. Pour des valeurs intermédiaires de ce ratio, nous n'observons pas de tendance particulière. Nous observons également que pour converger, PCP est plus rapide que EM-FISTA-TV pour un bruit Poisson important, alors que c'est le contraire pour un ratio Gaussien/Poisson élevé. Ces résultats sont obtenus pour un paramètre de pondération donné, et il n'est pas garanti que ces résultats soient les mêmes pour différents paramètres α.

Ces tests nous ont surtout permis de voir les limites inhérentes aux méthodes itératives pour la reconstruction d'image. Elles peuvent être résumées comme suit:

- Toutes les méthodes considérées reposent sur l'hypothèse que l'opérateur A est parfaitement connu, ce qui n'est pas toujours le cas selon l'application.

- Le choix de la fonction coût et de l'algorithme adéquat représente une limite pour l'utilisation clinique, sachant qu'une solution pour effectuer ce choix de manière automatique semble difficilement envisageable.

- Nous avons pu, au cours de nos tests, constaté l'importance du choix du paramètre de pondération α. Ce choix a un impact non négligeable sur la qualité de l'image reconstruite et il est peu pratique de l'effectuer pour chaque acquisition.

- La complexité en temps et en mémoire de ces algorithmes représente un frein indéniable pour l'utilisation clinique.

Tous ces éléments représentent selon nous une raison suffisante pour la recherche de solutions basées sur l'apprentissage. Les travaux effectués dans la suite sont ainsi orientés sur ces méthodes plus récentes. Nous cherchons ainsi à voir un aperçu de la capacité de ces méthodes à surmonter les limites des méthodes traditionnelles.

Méthodes d'apprentissage profond

Les algorithmes basés sur l'apprentissage profond reposent sur un paradigme différent par rapport aux méthodes itératives. Une différence notable est que l'ajustement des hyper-paramètres est réalisé une fois pour toutes, indépendamment - en théorie - de l'image à reconstruire. En routine clinique, le temps de reconstruction étant souvent largement réduit, il s'agit également d'un critère majeur pour lequel ces méthodes sont une solution intéressante par rapport aux méthodes itératives. Aussi, la connaissance précise de la nature du bruit sur les données n'est plus aussi importante pour les reconstructions obtenues avec des réseaux de neurones. En effet, ce bruit peut être implicitement appris grâce aux données sur lesquelles l'algorithme est entraîné. Une interprétation de ces méthodes est que le processus d'apprentissage permet aux réseaux d'apprendre un *a priori* sur les images à reconstruire en s'appuyant sur des vérités-terrain.

Il y a plusieurs catégories de méthodes pour la reconstruction d'image avec des méthodes d'apprentissage profond. Les méthodes supervisées utilisent des données appareillées constituées de paires "données/image de référence" (ou "reconstruction détériorée/image de référence"). Il est possible par exemple d'améliorer une image reconstruite avec un algorithme classique: en général la rétroprojection, potentiellement filtrée. Ces méthodes sont des méthodes dites de *post-processing*. Un réseau de neurones peut également faire la tâche de reconstruction dans son ensemble, en prenant des projections en entrée et en les transformant en une image reconstruite: ce sont les architectures *end-to-end*. Enfin certains réseaux de neurones reproduisent le schéma des méthodes itératives, à la différence près que la stratégie de mise à jour de l'image estimée ainsi que les paramètres de mise à jour correspondants sont obtenus après un processus d'apprentissage grâce à des données appareillées.

L'utilisation de réseaux antagonistes génératifs (GAN) permet quant à elle de produire des images appartenant à la distribution - apprise et donc estimée - des images de référence. Les sorties de ce type de réseaux peuvent être déterministes, auquel cas ils ne se distinguent des autres réseaux que par l'utilisation d'une fonction coût antagoniste: la distance de Wasserstein par exemple. Il est également possible d'apprendre la distribution d'images conditionnellement à la donnée d'entrée, qui peut être la FBP de faible qualité par exemple. Dans ce cas, il est possible d'estimer une image reconstruite de façon stochastique, et ainsi d'avoir de potentielles informations statistiques (moyenne et variance par voxel, intervalles de confiance sur des paramètres quantitatifs à partir de l'image, etc).

D'autres méthodes existent et permettent d'apprendre uniquement un terme de régularisation adéquat pour l'inclure dans une fonction coût de type (5). Un algorithme itératif peut ensuite être utilisé pour minimiser cette fonction coût.

Enfin, les méthodes d'apprentissage auto-supervisées permettent d'utiliser des réseaux de neurones pour estimer une image reconstruite sans avoir besoin d'une base de données d'apprentissage. Les poids des réseaux sont mis à jour directement à partir des données d'entrée.

Un résumé des avantages et inconvénients de ces différentes méthodes est donné dans le Tableau 6.7.

Etude de l'impact de la fonction coût de réseaux de neurones pour la reconstruction de la micro-architecture osseuse

Le champ des possibilités afin de créer un réseau de neurones pour la reconstruction tomographique dans un contexte médical est très vaste. Un élément cependant commun à toutes les méthodes est le fait que les réseaux de neurones dépendent de la rétropropagation du gardient d'une fonction coût afin d'ajuster ses paramètres. Cette fonction coût a un rôle d'évaluation et permet de répondre à la question: à quel point la prédiction est-elle similaire à une image référence (vérité-terrain)?

Le choix de la fonction coût pour entraîner les réseaux de neurones est un sujet d'intérêt majeur notamment dans le domaine de la vision par ordinateur, mais aussi pour l'imagerie médicale. Ce choix peut avoir un impact non négigeable sur la sortie d'un réseau à l'inférence (au moment de la prédiction une fois que le réseau est entraîné). L'efficacité du processus d'apprentissage dépend donc de la capacité de la fonction coût à être en accord avec le rôle du réseau: est-ce que minimiser cette fonction coût permet

Méthode	Avantages	Inconvénients
Post-processing	Reconstruction rapide	Nécessité d'une base de données supervisée de taille importante
End-to-end	Pas la nécessité de connaître l'opérateur direct et + de flexibilité	Beaucoup de données requises pour l'apprentissage et plus grand nombre de paramètres pour les réseaux. Potentiellement temps de reconstruction plus long.
Schéma itératif déroulé	Opérateur direct inclus dans le modèle + moins de données d'apprentissage	Temps de reconstruction et impact mémoire
GAN : sortie déterministe	Informations hautes-fréquence retranscrites	Difficulté accrue pour l'entraînement
GAN : sortie stochastique	Informations statistiques sur différents paramètres	Temps de reconstruction
Apprentissage de régularisation	Pas de données appareillées et régularisation adaptée au type de données	Temps de reconstruction
Auto-supervisé	Pas d'apprentissage initial donc pas de base de données	Convergence lente donc temps de reconstruction important

Table 6.7: Différentes catégories de méthodes d'apprentissage profond pour la reconstruction tomographique

en effet de produire des sorties qui sont satisfaisantes par rapport au but final du réseau?

Dans ce travail nous nous focalisons donc sur ce choix de fonction coût pendant l'apprentissage. Etant donné que le critère principal en pratique est que l'image reconstruite soit précise et exploitable dans l'optique de réaliser un diagnostic (ou une décision médicale de manière générale), nous effectuons notre étude sur l'impact de la fonction coût par rapport à des critères quantitatifs et qualitatifs qui sont pertinents en ce sens.

Notre objectif principal est de travailler sur le cas de la reconstruction d'images à partir de données acquises avec une faible dose de radiation; il s'agit en effet d'un axe de recherche particulièrement important pour les méthodes d'apprentissage profond. La dégradation des images liée à la faible qualité des données acquises dans de telles conditions peuvent en effet avoir des conséquences importantes sur la précision des informations quantitatives et qualitatives obtenues. Nous devons également nous assurer d'avoir à disposition des images permettant une étude suffisamment complète, qui puisse aussi être utile pour le domaine d'application. Pour cela nous considérons des données micro-CT (μ-CT) pour l'imagerie de la micro-architecture osseuse. La complexité de la structure des images correspondantes est telle qu'elle permet d'évaluer l'impact de la fonction coût de façon détaillée. Aussi, le diagnostic des pathologies osseuses est directement lié au calcul de certains paramètres quantitatifs; il est donc nécessaire de s'assurer que les réseaux de neurones permettent de retrouver des valeurs fiables pour ces paramètres. Nous pouvons ainsi évaluer la capacité ou non de certaines fonctions coût à être plus fiables que d'autres en ce sens.

Une des pathologies principales en lien avec les os est l'ostéoporose, pour laquelle on estime que plus de 200 millions de personnes souffrent dans le monde [Sözen et al., 2017]. Il s'agit d'une maladie diffuse du squelette qui fragilise l'os et donc augmente le risque de fracture. La fragilité de l'os dépend notamment de deux facteurs principaux:

- La densité minérale osseuse (DMO) qui est une mesure en g/cm^2 de la densitée projetée sur une zone d'intérêt.

- La "qualité" osseuse qui peut être caractérisée par différentes propriétés comme la micro-architecture osseuse de l'os trabéculaire, la prèsence de micro-fractures, etc.

Expériences

Méthodes Nous étudions différentes combinaisons de fonction coût: par pixel, structurelles et antagonistes notamment. Nous cherchons à évaluer les avantages et inconvénients de chacune de ces catégories, et nous voulons également déterminer si certaines fonctions coût sont plus adaptées que d'autres au sein même de ces catégories. Afin de réaliser cette étude nous considérons des métriques qui sont propres au domaine de la vision par ordinateur, mais également des métriques pertinentes vis à vis du diagnostic post-reconstrution de pathologies osseuses. Nous considérons une tâche simple qui consiste à améliorer la qualité d'une FBP obtenue à partir de projections en faible dose avec un réseau de neurones convolutionnel (CNN) entraîné sur des paires d'images "dose normale/faible dose". Ce travail à pour objectif de donner des indications sur l'impact de la fonction coût dans le contexte de la reconstruction tomographique et également de donner des recommendations sur la stratégie à adopter pour le choix de cette fonction coût

Métrique	Intérêt
Vision par ordinateur	
PSNR	Récupération du signal et débruitage
SSIM	Structure de l'image
Orienté diagnostic	
Distance de Wasserstein 1D	Fiabilité de la DMO estimée
Nombre d'Euler	Connectivité de l'os
BV/TV	Fraction volumique de l'os
Autres	
Résolution (FRC)	Restitution des détails de l'image
DICE	Précision de la segmenation par pixel

Table 6.8: Métriques utilisées pour l'étude et leur intérêt respectif.

lorsque la reconstruction de la micro-architecture osseuse est effectuée par apprentissage profond.

Pour les fonctions coût par pixel, nous considérons les distances l_1 et l_2. Pour les fonctions coût structurelles, nous utilisons soit SSIM, soit une fonction coût de type VGG, qui est une distance entre les caractéristiques (*features*) des images, basée sur le réseau de neurones VGG-19 pré-entraîné sur des millions d'images pour des tâches de classificiation. Enfin, nous considérons la distance de Wasserstein estimée avec des réseaux de neurones (de type Wasserstein GAN) pour la catégorie des fonctions coût antagonistes.

Les métriques que nous incluons dans l'étude afin d'évaluer la qualité des reconstructions sont résumées dans le Tableau 6.8.

Données Les données de référence sont des volumes correspondants à des radius et tibias humains obtenus sur un SCANCO μ-CT 100 avec une taille de voxel de 24-μm. La base de données d'entraînement est composée de dix volumes de différents donneurs. Deux volumes provenant de deux autres sujets sont utilisés uniquement pour l'évaluation des méthodes.

Nous calculons la transformée de Radon (parallèle) à partir de ces volumes. Afin de simuler des données à faible dose, nous considérons d'abord 400 projections, ce qui correspond à environ 50 % du nombre total de projections pour une dose normale. Nous considérons ensuite un bruit Poisson sur le nombre de photons reçus par le détecteur. En faisant varier l'intensité du rayonnement I_0, c'est à dire le nombre de photons envoyés par la source, nous pouvons simuler différents niveaux de dose. Dans la base de données d'entraînement, nous considérons ainsi entre 5% et 50% de la dose normale pour les données d'acquisition.

Les hyper-paramètres des réseaux de neurones sont optimisés sur des données de validation correspondant à 20 % des données d'entraînement, la stratégie étant la même pour tous les réseaux. Pour l'évaluation, nous simulons deux configurations: 10 % et 20

% de la dose, ce pour chacun des deux volumes.

Résultats Le Tableau 6.9 montre les valeurs des métriques pour tous les réseaux considérés, pour le volume test 1 avec 10 % de la dose. Les résultats comparatifs sont similaires pour l'autre volume et 20 % de la dose.

Analyses et conclusion Dans notre étude, PSNR, SSIM et le DICE ne permettent pas de faire la distinction entre les fonctions coût par pixel, structurelles et antagonistes. L'évaluation de la qualité de la reconstruction des micro-structures osseuses semble ainsi insuffisante en ne considérant que ces métriques.

Au contraire, le ratio BV/TV, le nombre d'Euler afin d'étudier la connectivité de l'os, la distance de Wasserstein entre les distributions de densité par voxel pour évaluer la fiabilité de l'estimation de la DMO, sont autant de métriques permettant de juger la capacité des réseaux de neurones à retrouver des paramètres clés pour le diagnostic post-reconstruction. Nous avons montré que la fonction coût utilisée pour l'entraînement a une influence majeure sur ces métriques. Par conséquent, le choix de la fonction coût doit être un élément important dans la construction d'une méthode basée sur l'apprentissage profond adaptée à l'application médicale. L'utilisation d'une fonction coût par pixel, notamment l_1, permet d'améliorer la résolution de l'image reconstruite. Les fonctions coûts structurelles ont un rôle important pour permettre au réseau de retrouver des structures osseuses précises comme le montrent les métriques de connectivité. La fonction coût VGG améliore les performances du réseau dans ce sens, au détriment d'une diminution observée de la résolution. L'ajout d'une fonction coût antagoniste permet d'obtenir des reconstructions plus précises en terme de DMO. Lorsque le choix de la fonction coût est réalisé, il est en revanche important de garder à l'esprit qu'un compromis doit être fait entre coût en ressources de calcul, du à la complexité de la fonction coût, et performance

WGAN-VGG	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
$\emptyset$	28.91	0.811	0.848	0.140	29 ± 20	0.07 ± 0.05	98.6 ± 6.7	21.62	43.81
L_1	**29.94**	**0.842**	**0.864**	**0.140**	24 ± 17	0.07 ± 0.05	**86.6 ± 6.0**	13.61	**19.93**
L_2	29.63	0.829	0.859	0.141	**23 ± 18**	**0.06 ± 0.04**	93.2 ± 5.9	**10.53**	31.20
CNN-VGG	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
$\emptyset$	26.87	0.128	0.858	0.140	40 ± 28	**0.20 ± 0.11**	98.3 ± 4.4	332.57	581.71
L_1	**30.43**	0.846	**0.866**	**0.140**	**37 ± 27**	0.21 ± 0.05	**77.9 ± 4.4**	**27.75**	**106.36**
L_2	30.19	**0.851**	0.858	0.147	79 ± 40	0.29 ± 0.05	94.1 ± 4.6	41.55	209.64
CNN-SSIM	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
$\emptyset$	**30.36**	**0.871**	0.865	0.141	44 ± 29	0.23 ± 0.05	83.1 ± 4.1	**29.45**	**103.6**
L_1	30.35	0.859	**0.865**	**0.139**	**26 ± 20**	**0.16 ± 0.05**	**77.3 ± 5.3**	30.47	109.56
L_2	30.27	0.863	0.859	0.148	63 ± 37	0.28 ± 0.05	91.1 ± 5.1	39.22	215.16
CNN	PSNR	SSIM	DICE	BV/TV	E-N	O-C	Resol (μm)	WV	WB
L_1	**30.43**	0.848	**0.866**	**0.140**	**33 ± 26**	**0.19 ± 0.05**	**78.7 ± 5.0**	**26.24**	**107.08**
L_2	30.17	**0.852**	0.856	0.148	37 ± 31	0.20 ± 0.05	95.4 ± 5.1	44.48	240.55

Table 6.9: Métriques pour le volume 1 obtenu avec 10 % de la dose. Les entrées en gras dans la première colonne indiquent la partie fixe de la fonction coût. Le ratio BV/TV de référence est 0.138.

finale.

La plupart des méthodes, résultats et analyses liés à cette étude ont été publiés dans [Leuliet et al., 2022a].

Déconvolution aveugle pour la TEP

Nous nous intéressons à présent à l'imagerie TEP. Les tests que nous effectuons par la suite sont réalisés sur des données TEP avec temps de vol (TOF). Cette application nous amène à considérer un modèle direct plus spécifique pour le problème inverse à résoudre.

Effet de volume partiel L'effet de volume partiel fait référence au fait que l'activité retrouvée dans un voxel en particulier est liée à la véritable activité des voxels adjacents en raison de la résolution limitée du système d'imagerie. De la même façon, la véritable activité de ce voxel affecte l'activité retrouvée dans les voxels adjacents. Aussi, la position de la désintégration radioactive à l'origine de l'événement détecté peut être légérement en dehors de la ligne de réponse. Plusieurs raisons peuvent expliquer ce phénomène: le *positron range*, l'accolinéarité des photons émis, la résolution du détecteur limitée, la discrétisation du volume, la diffusion inter-crystal, l'effet de la profondeur d'interaction, etc.

Si le fait que la position de l'émission du positon à l'origine de l'événement détecté n'est pas pris en compte dans le processus de reconstruction, un étalement peut être observé sur l'image reconstruite, même lorsque que l'on considère une source ponctuelle. Ceci fait référence à la notion de fonction d'étalement du point (PSF), qui caractérise ainsi la résolution spatiale du système d'imagerie.

Reconstruction avec temps de vol Les systèmes TEP avec temps de vol (TOF) correspondent à des systèmes d'imagerie capables de mesurer le temps entre deux détections de photons formant un même événement de coïncidence. L'information obtenue sur la position de l'annihilation est par conséquent plus précise; au lieu de connaître la ligne de réponse sur laquelle l'annihilation a eu lieu, ces systèmes permettent de connaître directement cette position, avec une précision qui dépend de la résolution temporelle.

Dans l'espace image, les données sont rétroprojetées avec une fonction Gaussienne dont le noyau a une largeur correspondante à la résolution temporelle; si t_{FWHM} est le FWHM (full width at half maximum) mesuré pour la résolution temporelle, alors le FWHM considéré pour la fonction Gaussienne est $d_{FWHM} = ct_{FWHM}$ avec c la vitesse de la lumière dans le vide. L'intensité maximale de la Gaussienne rétroprojetée est elle située au niveau de la distance mesurée à partir de la différence temporelle entre les deux détections.

Déconvolution aveugle Le modèle que nous considérons pour la reconstruction avec temps de vol est le suivant:

$$y = k * f \tag{6}$$

où y est la reconstruction TOF, k est le noyau de convolution de la PSF, qui correspond donc à la fois à l'étalement du à la rétroprojection TOF et à la résolution spatiale du

système liée à la PSF. Nous considérons ce noyau k comme inconnu et dépendant de l'objet. Le problème auquel nous nous intéressons dans (6) est donc un problème de déconvolution aveugle. Du bruit est également présent sur y. Ce modèle permet notamment de conserver une formulation unique pour les méthodes de reconstruction analytiques et itératives. Il est également particulièrement adapté aux données obtenues en *list-mode* et permet de travailler uniquement dans le domaine image, ce qui présente des avantages du point de vue de l'implémentation.

Méthodes de résolution Parmi les méthodes classiques de résolution de ce problème de déconvolution aveugle, il est possible de découpler l'estimation de la PSF et la reconstruction de l'image. Une possibilité est de reconstruire l'image sans prendre en compte la PSF, et de résoudre ensuite un problème de déconvolution avec une PSF estimée grâce à des mesures effectuées expérimentalement. L'estimation de la PSF permet également d'utiliser des algorithmes de reconstruction en considérant un opérateur direct incluant l'opérateur de convolution représentant la PSF.

Une autre possibilité, plus adapté au problème que nous considérons étant donné que la PSF n'est pas supposée connue, est de résoudre le problème de déconvolution aveugle avec des méthodes itératives. Dans ce cas, l'estimation alternée de la PSF et de l'image peut être effectuée avec l'algorithme de Richardson-Lucy par exemple.

Les méthodes d'apprentissage sont également adaptées au problème de déconvolution aveugle. Il est possible d'utiliser des méthodes directes qui permettent d'améliorer la qualité d'une reconstruction de faible qualité (voire de faire la reconstruction à partir des projections) à l'aide d'un réseau de neurones, par exemple un U-NET. Ce réseau peut être un réseau 2D ou 3D. Lorsqu'il n'est pas concevable en pratique de considérer un réseau 3D, il est possible de prendre en compte l'information spatiale manquante avec des réseaux prenant plusieurs coupes adjacentes en entrée par exemple.

D'autres méthodes basées sur l'apprentissage supervisée existent, en prenant cette fois en compte le modèle direct considéré. Une solution est par exemple d'estimer à la fois l'image et la PSF à partir des données d'entrée; ainsi, une fonction coût de cohérence peut être utilisé pour mettre à jour les poids du réseau. Elle permet de s'assurer que la convolution de l'image et de la PSF est en adéquation avec les données d'entrée, selon (6) [Rego et al., 2021].

De nombreuses méthodes pour la déconvolution aveugle reposent sur l'apprentissage auto-supervisé. Certaines sont notamment inspirées du principe DIP (Deep Image Prior). L'idée des réseaux DIP est de considérer qu'une image peut être représentée comme un réseau de neurones convolutionnel. En pratique, la solution est caractérisée par un réseau de neurones qui prend un tenseur aléatoire fixe z en entrée, la k-ième estimation de la solution correspondant ainsi à $f_k = \Gamma_{\theta_k}(z)$ où Γ est un réseau de neurones avec paramètres θ_k. Ici l'entraînement est uniquement réalisé avec une seule donnée de référence, à savoir l'image dégradée y. A chaque *epoch* - ou itération - k, les poids θ_k sont mis à jour par rétropropagation du gradient d'une fonction coût incluant à la fois $\Gamma_{\theta_k}(z)$ et y.

Une première solution basée sur le modèle DIP est d'estimer à la fois l'image et la PSF en sortie du réseau, et d'utiliser une fonction coût de cohérence comparant uniquement la convolution de ces deux estimations avec y [Ren et al., 2020]. Il est également possible de considérer un réseau itératif déroulé comme c'est le cas pour le réseau Deep-URL [Agarwal et al., 2020] par exemple.

Application à des données TEP avec temps de vol pour l'imagerie péropératoire

L'objectif final du système d'imagerie que nous étudions est d'être un outil permettant de déterminer le statut métastatique des ganglions drainant une tumeur primaire: les ganglions sentinelles. Il s'agit en effet d'un facteur clé pour le pronostic des patients ayant un cancer [Sajedi et al., 2019]. Ce statut métastatique permet notamment d'évaluer le stade du cancer, et il permet d'orienter le choix de la thérapie adéquat pour le patient.

Notre étude porte sur un système d'imagerie péopératoire dont le dispositif est illustré sur la Figure 6.22.

En utilisant une rétroprojection simple pour reconstruire l'image à partir des données d'acquisition, des expériences ont montré que des sphères de diamètre 6 mm correspondant à des sources radioactives peuvent être identifiées grâce à ce dispositif d'imagerie lorsque le ratio de l'activité entre les sources et le fond est de 10:1, avec un temps d'acquisition d'au plus une minute.

Motivations pour la méthode proposée

L'ensemble des méthodes pour le problème que l'on considère peut être résumé de la façon suivante:

- Les méthodes itératives ne sont pas adaptés à notre tâche en raison du temps de reconstruction trop important pour l'imagerie péropératoire.
- Les méthodes nécessitant une estimation préalable de la PSF ne sont également pas adaptées: la PSF est en effet considérée comme inconnue et dépendante de l'objet imagé. Aussi, la possibilité de déplacer le détecteur au-dessus du patient fait que cette PSF est nécessairement variable.

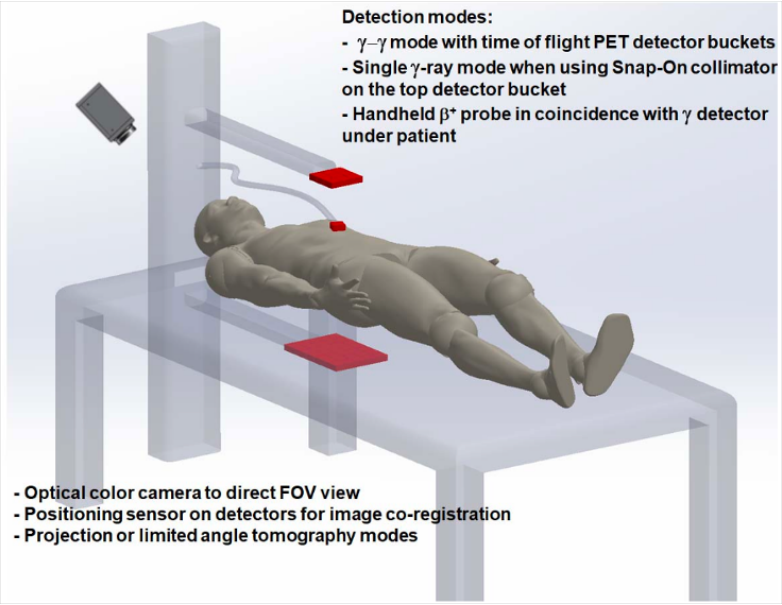


Figure 6.22: Illustration du concept du dispositif étudié pour l'imagerie péropératoire [Sajedi et al., 2019].

- Les méthodes d'apprentissage supervisé peuvent être adéquats mais elles sont potentiellement non robustes face au décalage existant entre les données disponibles pour l'entraînement et les données tests. Ce décalage peut être expliqué par une difficulté de création d'une base de données suffisamment grande en raison des simulations Monte-Carlo qui sont requises. Aussi, le lien entre simulations Monte-Carlo et données réelles, ainsi que l'impossibilité de représenter l'ensemble des configurations pour la position du détecteur et l'objet d'intérêt font que ces méthodes peuvent être peu robustes.

- Les méthodes d'apprentissage auto-supervisé doivent nécessairement faire l'estimation de la PSF afin d'avoir une fonction coût de cohérence pertinente. Un problème majeur est que les méthodes existantes présentent un temps de reconstruction trop long pour l'imagerie péropératoire.

Tous ces élements nous amènent à proposer une méthode permettant de bénéficier des avantages des méthodes d'apprentissage profond sans pâtir des conséquences telles que le temps de reconstruction non adéquat lié à l'apprentissage auto-supervisé.

PAVENET

Le réseau que nous proposons, appelé PAVENET, est composé de deux blocs distincts. Nous considérons un UNET pour ces deux blocs, et nous travaillons uniquement dans le domaine image. L'entrée de ces deux réseaux correspond à une reconstruction analytique y; dans notre cas, il s'agit de la reconstruction TOF. Le premier bloc H_w estime l'image à partir de y; le second bloc G_θ estime le noyau de convolution de la PSF à partir de y. Le schéma du réseau de neurones est représenté sur la Figure 6.23.

La méthode est basée sur l'apprentissage auto-supervisé; les poids de H_w et G_θ sont en effet mis à jour à l'inférence, pour toute nouvelle donnée d'entrée y. Afin de pallier au problème de convergence lente lié à l'apprentissage auto-supervisé, nous effectuons un pré-entraînement afin d'initialiser les poids des deux blocs de façon optimisée. Nous utilisons pour cela une base de données d'entraînement en considérant des images et PSF de référence connues. Nous pré-entraînons ainsi les deux blocs avec une fonction coût telle que:

$$L_{w,\theta}(\tilde{k}, \tilde{f}) = L_{im}(f, \tilde{f}) + L_{PSF}(k, \tilde{k}) + L_{cons}(\tilde{k}, \tilde{f}, y) \tag{7}$$

où $\tilde{f} = H_w(y)$ et $\tilde{k} = G_\theta(y)$ sont respectivement l'image et la PSF estimées, et f et k sont les vérités-terrain correspondantes. Les fonctions coût L_{im} et L_{PSF} sont typiquement les distances l_2 et l_1 dans notre cas.

La fonction coût de cohérence L_{cons} est par la suite utilisée pour l'entraînement auto-supervisé afin de mettre à jour w et θ.

Il est important de noter que la méthode proposée est une méthode générale qui repose sur deux principes majeurs: un apprentissage hybride (auto-supervisé avec initialisation des poids grâce à un pré-entraînement supervisé), et une double architecture permettant d'estimer à la fois la PSF et l'image. Cette méthode peut ainsi être adaptée à tout problème de reconstruction faisant intervenir une étape de déconvolution aveugle.

Protocole expérimental

Nous cherchons à valider l'efficacité de PAVENET. Le protocole expérimental que nous considérons est le suivant:

1. Pour preuve de concept, nous évaluons la performance de PAVENET sur des simulations *Python* avec des objets simples (des disques, en 2D), en considérant le modèle direct comme connu avec des PSF également connues afin de générer les données. L'objectif de cette étape est de s'assurer que PAVENET est en effet performant lorsque le modèle considéré est exact; si ce n'est pas le cas, il n'y a pas de raison d'aller plus loin dans les tests lorsque le modèle est simplement estimé.

2. Pour la deuxième étape nous considérons la véritable géométrie du système TEP d'intérêt. Nous considérons des volumes 3D en plaçant des sphères radioactives et nous réalisons des simulations GATE d'acquisitions à partir de ces volumes. La PSF pour le pré-entraînement doit ainsi être estimée, et nous n'avons pas de garantie que le modèle (6) est en adéquation avec les données obtenues. PAVENET est ensuite évalué sur de tels volumes et comparé avec une méthode d'apprentissage supervisé: UNET.

3. La dernière étape, non présentée dans cette thèse car il s'agit de tests en cours, sera de valider la méthode sur des simulations GATE plus complexes et réalistes, en considérant une activité plus faible pour les sources chaudes, un phantom plus profond, etc. Des tests seront également réalisés sur des données expérimentales pour évaluer la robustesse de la méthode vis à vis de données non simulées.

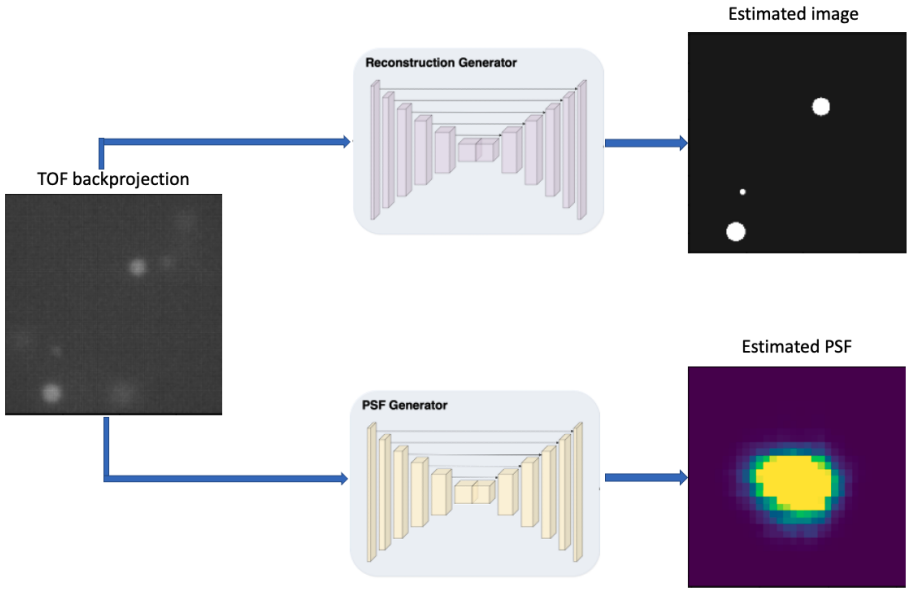


Figure 6.23: Architecture de PAVENET pour une entrée correspondant à une reconstruction TOF. Le réseau est pré-entraîné avec des données supervisées. Les poids sont ensuite mis à jour à l'inférence étant donné une reconstruction TOF en entrée. L'architecture est similaire à [Rego et al., 2021] mais la méthode d'entraînement est différente.

	Test A				Test B			
Methods	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.
FBP	12.86 ± 0.00	0.093 ± 0.000	34.31 ± 0.00	0.284 ± 0.000	14.65 ± 0.00	0.058 ± 0.000	61.90 ± 0.00	0.109 ± 0.000
UNET	44.66 ± 3.29	0.950 ± 0.031	1.60 ± 0.78	**0.999 ± 0.001**	21.89 ± 0.14	0.883 ± 0.029	22.70 ± 0.95	0.931 ± 0.003
PAVENET-0	**50.29 ± 0.24**	**0.993 ± 0.001**	**0.54 ± 0.12**	**0.999 ± 0.001**	21.79 ± 0.26	**0.924 ± 0.004**	21.50 ± 1.08	0.930 ± 0.005
PAVENET-1000	43.31 ± 0.32	0.932 ± 0.017	1.06 ± 0.42	0.997 ± 0.004	**22.58 ± 0.28**	0.899 ± 0.015	**19.25 ± 1.13**	**0.940 ± 0.004**
	Test C				Test D			
	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.	PSNR (dB)	SSIM	CRC Err. (%)	SSIM Corr.
FBP	10.60 ± 0.00	0.116 ± 0.000	17.37 ± 0.00	0.255 ± 0.000	12.24 ± 0.00	0.052 ± 0.000	59.21 ± 0.00	0.066 ± 0.000
UNET	18.64 ± 0.39	0.775 ± 0.034	4.23 ± 1.28	0.895 ± 0.008	25.91 ± 0.71	0.903 ± 0.042	11.11 ± 1.11	0.977 ± 0.007
PAVENET-0	18.50 ± 0.25	**0.818 ± 0.008**	7.81 ± 0.78	0.903 ± 0.009	26.90 ± 0.38	**0.947 ± 0.007**	**8.57 ± 0.87**	0.971 ± 0.005
PAVENET-1000	**20.96 ± 0.42**	0.783 ± 0.016	**3.27 ± 0.78**	**0.935 ± 0.006**	**27.64 ± 0.20**	0.891 ± 0.015	9.01 ± 0.79	**0.980 ± 0.003**

Table 6.10: Métriques pour les différentes méthodes testées. PAVENET-0 correspond aux images reconstruites par PAVENET uniquement après le pré-entraînement, et PAVENET-1000 correspond aux prédictions faites après 1000 itérations d'apprentissage auto-supervisé.

Validation avec images et PSF simulées

Nous simulons 20 000 images/PSF pour l'entraînement du UNET, qui sont également utilisées pour pré-entraîner PAVENET. Des disques de taille et activité variable sont placés dans ces images. L'étalement de la PSF est également variable à travers la base de données. Les données d'entrée correspondent à la FBP des projections bruitées à partir de la convolution de ces paires image/PSF.

Pour évaluer la performance de PAVENET en comparaison avec le UNET supervisé, nous créons 4 jeux de données test. Le jeu de données test A possède les même caractéristiques que l'entraînement (disques de taille similaire, étalement de la PSF similaire). Le test B contient uniquement des disques de rayon inférieur aux rayons considérés dans l'entraînement; le test C contient des disques de rayon supérieur. Enfin, le test D est obtenu avec des PSF dont l'étalement est plus large que celui considéré pendant l'entraînement.

Les résultats en terme de métriques sont représentés sur le Tableau 6.10. Notamment, nous représentons le Concentration Recovery Coefficient (CRC), défini comme le ratio $CRC = \tilde{f}_{\text{act}}/f_{\text{act}}$ où l'indice act fait référence à l'activité moyenne retrouvée à l'intérieur des disques de l'image. Comme ce ratio doit dans l'idéal être proche de 1, nous présentons les résultats pour $CRC_{err} = |CRC - 1|$.

Dans ce contexte très simplifié d'acquisitions de données TEP, nous montrons que la connaissance du modèle dans le réseau de neurones permet d'améliorer la qualité de l'image reconstruite pour les métriques considérées.

Les limites du UNET sont clairement visibles quand un décalage entre les données tests est observé par rapport aux données d'entraînement. La méthode d'apprentissage hybride proposée pour PAVENET permet de surmonter ces limites comme le montre la capacité du réseau à reconstruire des disques de rayon considérablement plus petit en comparaison avec les disques utilisés pour l'entraînement: une illustration est donnée sur la Figure 6.24.

Cette différence de performance est essentiellement liée à l'apprentissage supervisé qui permet de s'adapter à de nouvelles caractéristiques sur les données. Lorsque nous

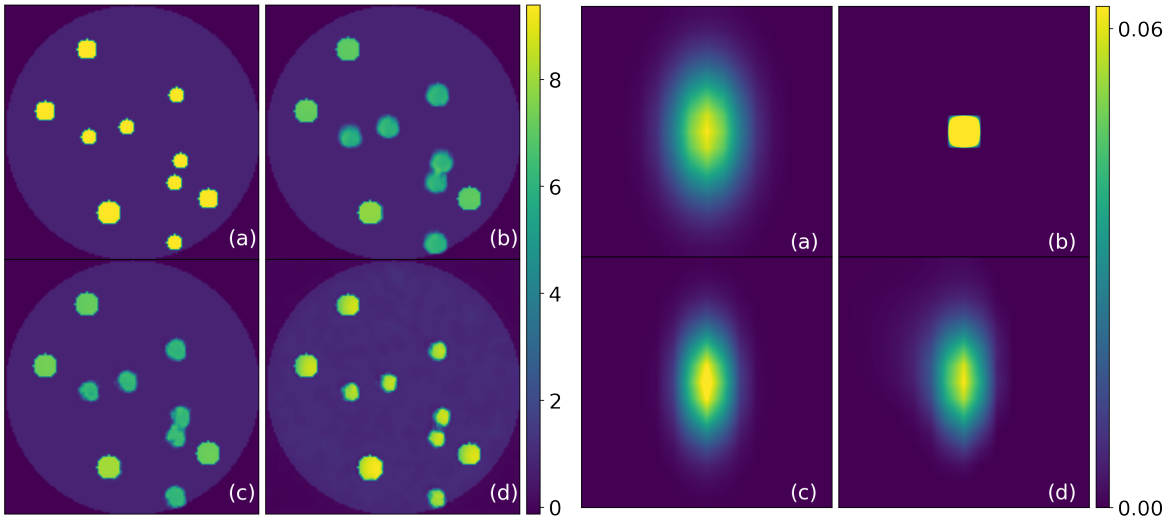


Figure 6.24: Test B: image (gauche) et PSF (droite) pour (a) la vérité-terrain, (b) UNET, (c) PAVENET initialisé après pré-entraînement (d) PAVENET après apprentissage auto-supervisé.

considérons les méthodes d'apprentissage auto-supervisé, nous voyons également qu'il est primordial d'utiliser une structure du type PAVENET qui prend en compte - et estime correctement - l'opérateur de convolution pour calculer la fonction coût de cohérence. En effet dans nos tests nous montrer une baisse de performance lorsque l'apprentissage auto-supervisé est utilisé sur un seul bloc image UNET, correspondant à une estimation d'un opérateur de convolution égal à l'identité. Les résultats que nous montrons sont obtenus après 40 secondes d'apprentissage auto-supervisé, en sachant que l'implémentation peut encore être améliorée. Lorsqu'aucun pré-entraînement n'est effectué, nous n'avons trouvé aucun moyen d'obtenir des résultats satisfaisants dans un temps raisonnable, correspondant dans notre cas à 10 000 itérations.

Validation sur simulations Monte Carlo

Méthode La méthode pour simuler les jeux de données ici est similaire à la précédente, avec deux différences notables:

- Les reconstructions initiales sont obtenues par rétroprojection TOF à partir des données *list-mode* obtenues sur les simulations GATE.
- Les vérités-terrain que nous simulons sont ici uniquement des volumes. Nous n'avons donc pas de PSF vérité-terrain pour le pré-entraînement. Nous donnons la méthode pour estimer ces PSF ci-après.

Nous simulons des volumes en trois dimensions, mais les réseaux que nous considérons sont des réseaux 2D. Par conséquent, la base de données d'entraînement est constituée de coupes. Nous allons séparer les résultats obtenus lorsque l'entraînement est effectué selon les coupes coronales, et lorsqu'il est effectué selon les coupes sagittales.

Pour chaque image de référence f dans le jeu de données d'entraînement créé avec un ratio de 10:1 pour l'activité des sources chaudes, nous simulons un volume correspondant

à haute dose f_{HD} avec un ratio d'activité de 200:1. La reconstruction TOF de f_{HD}, notée y_{HD}, est par conséquent moins bruitée que y. Cela nous permet d'utiliser un algorithme itératif sans régularisation afin d'estimer le noyau k^i à partir de chaque paire 2D (y^i_{HD}, f^i_{HD}) dans la base de données d'entraînement. Régler un paramètre de régularisation ne serait en effet pas envisageable au vu du nombre de données dans la base d'entraînement.

PAVENET est un réseau qui est implémenté en 2D; les paires sur lesquelles il est entraîné sont des coupes 2D. Ainsi, nous considérer un noyau de convolution référence pour chaque coupe dans la base de données d'entraînement. Ces noyaux de convolutions sont donc différents dépendamment du fait que le réseau soit entraîné sur les coupes coronales ou sagittales.

Pour retrouver k^i à partir de (y^i_{HD}, f^i_{HD}), nous résolvons le problème de minimisation suivant:

$$\min_{k^i} KL(y^i_{HD}, k * f^i_{HD}). \tag{8}$$

Cela correspond à un problème de déconvolution (non aveugle) que nous résolvons avec 100 itérations de l'algorithme de Richardson-Lucy. Une fois le noyau de convolution estimé, les données pour le pré-entraînement de PAVENET correspondent aux triplets (y^i, f^i, k^i).

Un critère majeur pour l'application médicale ici est un taux de faux négatifs (FNR) pour la détection de ganglions sentinelles de moins de 5 %. Un moyen d'évaluer la détection ou non d'une sphère est d'utiliser le Contrast to Noise Ratio (CNR), défini comme:

$$CNR = \frac{|act_S - act_{BG}|}{std_{BG}}. \tag{9}$$

où act_S est l'activité moyenne de la sphère sur la reconstruction, act_{BG} est l'activité moyenne du fond et std_{BG} son écart-type.

Nous calculons le FNR (%) comme le ratio de sphères non détectées sur le nombre total de sphères, en considérant un critère de détectabilité $CNR > 3$.

La valeur du contraste de référence cst_{GT} est connue: il s'agit de la différence d'activité entre source chaude et fond sur les simulations. Nous présentons ainsi l'erreur sur le contraste retrouvée comme la différence absolue relative par rapport à cette valeur de référence, notée CNR_{corr} et telle que:

$$CNR_{corr} = \frac{|(act_S - act_{BG}) - cst_{GT}|}{cst_{GT}}. \tag{10}$$

Enfin, nous montrons les résultats pour le taux de faux positifs (FPR), c'est à dire le ratio de sphères dans la reconstruction qui ne sont pas prèsentes sur l'image de référence. Un critère de l'ASCO est d'avoir un FPR inférieur à 15% .

En plus de tester UNET et PAVENET dans deux directions différentes, nous testons également un réseau similaire à [Xu et al., 2017], qui prend en entrée plusieurs coupes adjacentes à la coupe d'intérêt. Il s'agit d'un réseau de neurones résiduel ici (ResNet). Nous testons également un UNET 2.5D, où chaque voxel de l'image prédite est la moyenne pondérée de 3 UNET entraînés selon les coupes axiales, coronales et sagittales.

	PSNR	CRC Err. (%)	CNR Corr. (%)	FNR (%)	FPR (%)
TOFREC	14.98 ± 0.11	71.7 ± 0.3	84.2 ± 6.7	22.5	100.0
UNET-C	29.66 ± 0.61	2.3 ± 0.1	34.7 ± 29.9	9.6	7.6
PAVENET-C	30.49 ± 0.22	3.4 ± 0.3	24.7 ± 26.8	4.8	6.0
UNET-S	**32.43 ± 0.39**	1.1 ± 0.1	16.1 ± 20.6	**0.0**	**1.2**
PAVENET-S	30.48 ± 0.43	**0.9 ± 0.3**	**7.5 ± 14.1**	**0.0**	1.5

Table 6.11: Métriques sur le test A pour des sphères de rayon entre 1.5 mm et 5 mm. UNET-C et PAVENET-C sont entraînés sur les coupes coronales, UNET-S et PAVENET-S sur les coupes sagittales.

Résultats principaux Nous montrons sur la Figure 6.25 un exemple de reconstruction d'une coupe coronale obtenue avec différents réseaux, entraînés sur les coupes coronales. Les réseaux entraînés sur de telles coupes sont mentionnées avec "-C", et la mention "-S" est ajoutée pour les réseaux entraînés sur les coupes sagittales.

Les métriques sur le test A, avec des sphères de diamètre similaire à la configuration d'entraînement, sont restituées sur la Table 6.11.

Lorsque nous testons les différents réseaux sur des volumes contenant des sphères de rayon 1 mm (plus petites donc que la base de données d'entraînement), nous observons que UNET-C et PAVENET-C ne sont pas capables de détecter ces sphères.

PAVENET-S est en revanche capable de retrouver 10.4 % de ces sphères, contre seulement 3.5 % pour UNET-S. Aussi, nous notons que le FNR décroît avec le nombre d'itérations pour l'apprentissage auto-supervisé de PAVENET-S. La valeur du CRC est également meilleure pour ce réseau dans ce cas, avec 46.2% après 1000 itérations contre 69.4% pour UNET-S.

Synthèse des résultats Un premier constat majeur est que les méthodes d'apprentissage profond permettent d'améliorer significativement les résultats en considérant les métriques présentées, lorsque nous comparons la performance avec la reconstruction TOF initiale. Une amélioration d'au moins 2 mm sur le diamètre pour lequel toutes les sphères sont détectées est en effet observée grâce à ces méthodes.

Les réseaux entraînés sur des coupes coronales sont de façon générale moins perfor-

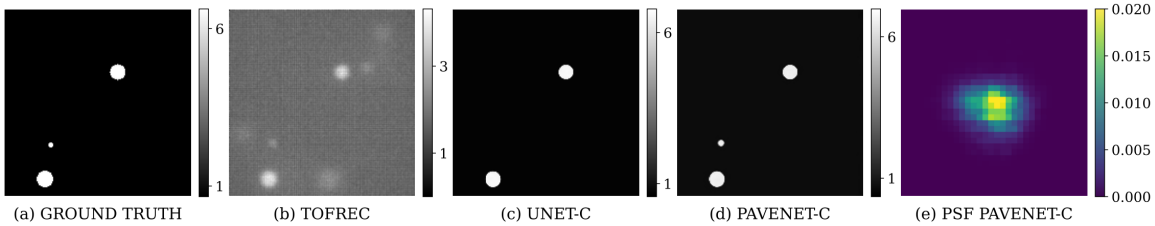


Figure 6.25: Exemple de coupe coronale reconstruite par différentes méhtodes. UNET et PAVENET sont entraînés sur des coupes coronales ici. L'image et la PSF obtenue par PAVENET-C sont obtenues après 1000 itérations d'apprentissage auto-supervisé.

mants comparés à ceux entraînés sur les coupes sagittales. En revanche visuellement, les sphères sont plus nettes lorsque nous observons les coupes coronales reconstruites avec UNET-C et PAVENET-C.

Pour les réseaux entraînés sur les coupes coronales, une nette amélioration est observée pour PAVENET en comparaison avec UNET. L'apprentissage auto-supervisé permet à PAVENET de corriger les erreurs de prédiction après apprentissage uniquement supervisé. Cet apprentissage auto-supervisé ne peut être pertinent et efficace que si la PSF est estimée de façon précise, ce qui semble être le cas dans nos tests. Lorsque les sphères sont trop petites en revanche - rayon d'1 mm -, PAVENET-C n'est pas capable de les détecter.

L'amélioration de la performance liée à PAVENET est également visible lorsque les réseaux sont entraînés sur les coupes sagittales, même si les performances de UNET-S sont meilleures que UNET-C. Pour des sphères de rayon 1.5 mm ou plus, UNET-S a un taux de faux négatifs de 0 %. Cela est du à la capacité du réseau à lisser le fond, permettant ainsi au CNR d'avoir une valeur supérieure au seuil fixé à 3. Nous observons cependant que le contraste obtenu pour les sphères de rayon 1.5 mm est plus important pour PAVENET-S.

Un dernier point d'intérêt est la diminution du FNR avec les itérations de PAVENET. Cela montre notamment que l'apprentissage auto-supervisé permet de surmonter les limites des méthodes supervisées dans notre étude.

Conclusion Ces tests sur simulations GATE montrent que les méthodes basées sur l'apprentissage profond ont le potentiel de répondre aux objectifs sur l'identification des nodules pour le système d'imagerie TEP péropératoire étudié. La robustesse de ces méthodes doit cependant être validée sur données expérimentales.

Le fait de considérer l'information spatiale dans la troisième dimension pour les réseaux 2D ne semble être efficace que dans le cas où les coupes considérées pour l'entraînement sont les coupes coronales. Néanmoins, selon les métriques considérées pour l'étude, il semble préférable d'entraîner les réseaux de neurones sur les coupes sagittales si les réseaux 3D ne sont pas une option.

Nous avons également montré que PAVENET, grâce à l'apprentissage auto-supervisé, permet de surmonter les limites de l'apprentissage supervisé pour un réseau de type UNET. La capacité de PAVENET à reconstruire des sphères de petite taille est particulièrement encourageante. La deuxième étape du protocole expérimental afin de valider la méthode étant satisfaisante, des tests peuvent être réalisés sur des données plus réalistes et il s'agit d'un travail en cours.

De façon générale, PAVENET reste un réseau qui mérite d'être étudié plus en profondeur. Certaines méthodes peuvent notamment améliorer sa performance; aussi, d'autres tests permettraient d'évaluer sa performance de façon plus globale.

Bibliography

[Adler and Öktem, 2018] Adler, J. and Öktem, O. (2018). Deep bayesian inversion. *arXiv preprint arXiv:1811.05910*.

[Adler and Öktem, 2017] Adler, J. and Öktem, O. (2017). Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007.

[Adler and Öktem, 2018] Adler, J. and Öktem, O. (2018). Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332.

[Agarwal et al., 2020] Agarwal, C., Khobahi, S., Bose, A., Soltanalian, M., and Schonfeld, D. (2020). DEEP-URL: A Model-Aware Approach to Blind Deconvolution Based on Deep Unfolded Richardson-Lucy Network. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3299–3303.

[Ahn and Fessler, 2003] Ahn, S. and Fessler, J. A. (2003). Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms. *IEEE transactions on medical imaging*, 22(5):613–626.

[Alessio et al., 2010] Alessio, A. M., Stearns, C. W., Tong, S., Ross, S. G., Kohlmyer, S., Ganin, A., and Kinahan, P. E. (2010). Application and evaluation of a measured spatially variant system model for PET image reconstruction. *IEEE transactions on medical imaging*, 29(3):938–949.

[Ammann and Rizzoli, 2003] Ammann, P. and Rizzoli, R. (2003). Bone strength and its determinants. *Osteoporosis international*, 14(3):13–18.

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

[Arridge et al., 2019] Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. (2019). Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174.

[Asim et al., 2020] Asim, M., Shamshad, F., and Ahmed, A. (2020). Blind image deconvolution using deep generative priors. *IEEE Transactions on Computational Imaging*, 6:1493–1506.

[Ayers and Dainty, 1988] Ayers, G. and Dainty, J. C. (1988). Iterative blind deconvolution method and its applications. *Optics letters*, 13(7):547–549.

[Bailey et al., 2005] Bailey, D. L., Maisey, M. N., Townsend, D. W., and Valk, P. E. (2005). *Positron emission tomography*, volume 2. Springer.

[Banjak et al., 2018] Banjak, H., Grenier, T., Epicier, T., Koneti, S., Roiban, L., Gay, A.-S., Magnin, I., Peyrin, F., and Maxim, V. (2018). Evaluation of noise and blur effects with SIRT-FISTA-TV reconstruction algorithm: Application to fast environmental transmission electron tomography. *Ultramicroscopy*, 189:109–123.

[Banterle et al., 2013] Banterle, N., Bui, K., Lemke, E., and Beck, M. (2013). Fourier ring correlation as a resolution criterion for super-resolution microscopy. *Journal of Structural Biology*, 183(3):363–367.

[Barbano et al., 2021] Barbano, R., Leuschner, J., Schmidt, M., Denker, A., Hauptmann, A., Maaß, P., and Jin, B. (2021). Is deep image prior in need of a good education? *arXiv preprint arXiv:2111.11926*.

[Batson and Royer, 2019] Batson, J. and Royer, L. (2019). Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR.

[Beck, 2017] Beck, A. (2017). *First-order methods in optimization*. SIAM.

[Beck and Teboulle, 2009] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

[Benvenuto et al., 2008] Benvenuto, F., La Camera, A., Theys, C., Ferrari, A., Lanteri, H., and Bertero, M. (2008). The study of an iterative method for the reconstruction of images corrupted by Poisson and Gaussian noise. *Inverse Problems*, 24(3):1085–1088.

[Bertero et al., 2010] Bertero, M., Boccacci, P., Talenti, G., Zanella, R., and Zanni, L. (2010). A discrepancy principle for poisson data. *Inverse Problems*.

[Bluemel et al., 2013] Bluemel, C., Schnelzer, A., Okur, A., Ehlerding, A., Paepke, S., Scheidhauer, K., and Kiechle, M. (2013). Freehand SPECT for image-guided sentinel lymph node biopsy in breast cancer. *European journal of nuclear medicine and molecular imaging*, 40(11):1656–1661.

[Bogdanich, 2010] Bogdanich, W. (2010). After stroke scans, patients face serious health risks. *The New York Times*, 31.

[Browne and De Pierro, 1996] Browne, J. and De Pierro, A. (1996). A row-action alternative to the EM algorithm for maximizing likelihood in emission tomography. *IEEE transactions on medical imaging*, 15(5):687–699.

[Byrne, 1998] Byrne, C. (1998). Iterative algorithms for deblurring and deconvolution with constraints. *Inverse Problems*, 14(6):1455.

[Calatroni et al., 2017] Calatroni, L., De Los Reyes, J. C., and Schönlieb, C.-B. (2017). Infimal convolution of data discrepancies for mixed noise removal. *SIAM Journal on Imaging Sciences*, 10(3):1196–1233.

[Carasso, 1999] Carasso, A. S. (1999). Linear and nonlinear image deblurring: A documented study. *SIAM journal on numerical analysis*, 36(6):1659–1689.

[Carson, 1986] Carson, R. E. (1986). A maximum likelihood method for region-of-interest evaluation in emission tomography. *Journal of computer assisted tomography*, 10(4):654–663.

[Cates and Levin, 2018] Cates, J. W. and Levin, C. S. (2018). Evaluation of a clinical TOF-PET detector design that achieves ≤ 100 ps coincidence time resolution. *Physics in Medicine & Biology*, 63(11):115011.

[Chambolle, 2004] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97.

[Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145.

[Chambolle and Pock, 2016] Chambolle, A. and Pock, T. (2016). An introduction to continuous optimization for imaging. *Acta Numerica, Cambridge University Press (CUP)*. Acta Numerica, 25, pp.161-319. 10.1017/S096249291600009X . hal-01346507.

[Chen et al., 2018] Chen, H., Zhang, Y., Chen, Y., Zhang, J., Zhang, W., Sun, H., Lv, Y., Liao, P., Zhou, J., and Wang, G. (2018). LEARN: Learned Experts' Assessment-Based Reconstruction Network for Sparse-Data CT. *IEEE transactions on medical imaging*, 37(6):1333—1347.

[Chen et al., 2017] Chen, H., Zhang, Y., Kalra, M. K., Lin, F., Chen, Y., Liao, P., Zhou, J., and Wang, G. (2017). Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging*, 36(12):2524–2535.

[Coleman, 1999] Coleman, R. (1999). PET in lung cancer. *Journal of Nuclear Medicine*, 40(5):814–820.

[Conti and Bendriem, 2019] Conti, M. and Bendriem, B. (2019). The new opportunities for high time resolution clinical TOF PET. *Clinical and Translational Imaging*, 7(2):139–147.

[Coursey and Frush, 2008] Coursey, C. A. and Frush, D. P. (2008). CT and radiation: What radiologists should know. *Applied radiology*, 37(3):22.

[da Costa-Luis and Reader, 2017] da Costa-Luis, C. O. and Reader, A. J. (2017). Deep learning for suppression of resolution-recovery artefacts in MLEM PET image reconstruction. In *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–3.

[Darestani et al., 2022] Darestani, M. Z., Liu, J., and Heckel, R. (2022). Test-time training can close the natural distribution shift performance gap in deep learning based compressed sensing. *arXiv preprint arXiv:2204.07204*.

[Dashti and Stuart, 2017] Dashti, M. and Stuart, A. M. (2017). The bayesian approach to inverse problems. In *Handbook of uncertainty quantification*, pages 311–428. Springer.

[Daube-Witherspoon et al., 2006] Daube-Witherspoon, M. E., Surti, S., Matej, S., Werner, M., Jayanthi, S., and Karp, J. S. (2006). Influence of time-of-flight kernel accuracy in TOF-PET reconstruction. In *2006 IEEE Nuclear Science Symposium Conference Record*, volume 3, pages 1723–1727. IEEE.

[Dawson-Hughes et al., 2008] Dawson-Hughes, B., Tosteson, A., Melton, L., Baim, S., Favus, M., Khosla, S., and Lindsay, R. (2008). Implications of absolute fracture risk assessment for osteoporosis practice guidelines in the USA. *Osteoporosis International*, 19(4):449–458.

[De Pierro, 1995] De Pierro, A. R. (1995). A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE transactions on medical imaging*, 14(1):132–137.

[De Pierro and Yamagishi, 2001] De Pierro, A. R. and Yamagishi, M. B. (2001). Fast EM-like methods for maximum "a posteriori" estimates in emission tomography. *IEEE transactions on medical imaging*, 20(4):280–288.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[Ebashi et al., 1991] Ebashi, S., Koch, M., and Rubenstein, E. (1991). Handbook on synchrotron radiation.

[Erlandsson et al., 2012] Erlandsson, K., Buvat, I., Pretorius, P., Thomas, B., and Hutton, B. (2012). A review of partial volume correction techniques for emission tomography and their applications in neurology, cardiology and oncology. *Physics in Medicine and Biology*, 57(21):R119–R159.

[Fass, 2008] Fass, L. (2008). Imaging and cancer: a review. *Molecular oncology*, 2(2):115–152.

[Fish et al., 1995] Fish, D., Brinicombe, A., Pike, E., and Walker, J. (1995). Blind deconvolution by means of the Richardson–Lucy algorithm. *JOSA A*, 12(1):58–65.

[Fox et al., 2005] Fox, J., Miller, M., Recker, R., Bare, S., Smith, S., and Moreau, I. (2005). Treatment of postmenopausal osteoporotic women with parathyroid hormone 1-84 for 18 months increases cancellous bone formation and improves cancellous architecture: a study of iliac crest biopsies using histomorphometry and micro computed tomography. *Journal of Musculoskeletal and Neuronal Interactions*, 5(4):356.

[Friot-Giroux et al., 2022] Friot-Giroux, L., Peyrin, F., and Maxim, V. (2022). Iterative tomographic reconstruction with tv prior for low-dose cbct dental imaging. *Physics in Medicine & Biology*, 67(20):205010.

[Genant et al., 2008] Genant, H. K., Engelke, K., and Prevrhal, S. (2008). Advanced CT bone imaging in osteoporosis. *Rheumatology*, 47:iv9–iv16.

[Genevay et al., 2017] Genevay, A., Peyré, G., and Cuturi, M. (2017). GAN and VAE from an optimal transport point of view. *arXiv preprint arXiv:1706.01807*.

[Géron, 2019] Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".

[Gilbert, 1972] Gilbert, P. (1972). Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of theoretical biology*, 36(1):105–117.

[Gong et al., 2019] Gong, K., Catana, C., Qi, J., and Li, Q. (2019). PET Image Reconstruction Using Deep Image Prior. *IEEE Transactions on Medical Imaging*, 38(7):1655–1665.

[Gong et al., 2019a] Gong, K., Wu, D., Kim, K., Yang, J., El Fakhri, G., Seo, Y., and Li, Q. (2019a). EMnet: an unrolled deep neural network for PET image reconstruction. In *Medical Imaging 2019: Physics of Medical Imaging*, volume 10948, pages 1203–1208. SPIE.

[Gong et al., 2019b] Gong, K., Wu, D., Kim, K., Yang, J., Sun, T., El Fakhri, G., Seo, Y., and Li, Q. (2019b). MAPEM-Net: an unrolled neural network for Fully 3D PET image reconstruction. In *15th International meeting on fully three-dimensional image reconstruction in radiology and nuclear medicine*, volume 11072, pages 109–113. SPIE.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[Gordon et al., 1970] Gordon, R., Bender, R., and Herman, G. T. (1970). Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of theoretical Biology*, 29(3):471–481.

[Gravel et al., 2020] Gravel, P., Li, Y., and Matej, S. (2020). Effects of TOF resolution models on edge artifacts in PET reconstruction from limited-angle data. *IEEE transactions on radiation and plasma medical sciences*, 4(5):603–612.

[Gravel et al., 2019] Gravel, P., Surti, S., Krishnamoorthy, S., Karp, J., and Matej, S. (2019). Spatially-variant image-based modeling of PSF deformations with application to a limited angle geometry from a dual-panel breast-PET imager. *Physics in Medicine & Biology*, 64(22):225015.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved Training of Wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5769–5779. Curran Associates Inc.

[Gupta et al., 2021] Gupta, H., McCann, M. T., Donati, L., and Unser, M. (2021). CryoGAN: a new reconstruction paradigm for single-particle cryo-EM via deep adversarial learning. *IEEE Transactions on Computational Imaging*, 7:759–774.

[Hall and Brenner, 2008] Hall, E. and Brenner, D. (2008). Cancer risks from diagnostic radiology. *The British journal of radiology*, 81(965):362–378.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv e-prints*.

[Hildebrand and Rüegsegger, 1997] Hildebrand, T. and Rüegsegger, P. (1997). Quantification of bone microarchitecture with the structure model index. *Computer Methods in Biomechanics and Bio Medical Engineering*, 1(1):15–23.

[Homminga et al., 2002] Homminga, J., McCreadie, B., Ciarelli, T., Weinans, H., Goldstein, S., and Huiskes, R. (2002). Cancellous bone mechanical properties from normals and patients with hip fractures differ on the structure level, not on the bone hard tissue level. *Bone*, 30(5):759–764.

[Hu et al., 2020] Hu, Z., Xue, H., Zhang, Q., Gao, J., Zhang, N., Zou, S., Teng, Y., Liu, X., Yang, Y., Liang, D., et al. (2020). DPIR-Net: Direct PET image reconstruction based on the Wasserstein generative adversarial network. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(1):35–43.

[Hudson and Larkin, 1994] Hudson, H. M. and Larkin, R. S. (1994). Accelerated image reconstruction using ordered subsets of projection data. *IEEE transactions on medical imaging*, 13(4):601–609.

[Huesman, 1984] Huesman, R. H. (1984). A new fast algorithm for the evaluation of regions of interest and statistical uncertainty in computed tomography. *Physics in Medicine & Biology*, 29(5):543.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

[Ito et al., 2011] Ito, K., Jin, B., and Takeuchi, T. (2011). A regularization parameter for nonsmooth tikhonov regularization. *SIAM Journal on Scientific Computing*, 33(3):1415–1438.

[Ito et al., 2005] Ito, M., Ikeda, K., Nishiguchi, M., Shindo, H., Uetani, M., Hosoi, T., and Orimo, H. (2005). Multi-detector row CT imaging of vertebral microstructure for evaluation of fracture risk. *Journal of Bone and Mineral Research*, 20(10):1828–1836.

[Jacobs et al., 2003] Jacobs, A. H., Li, H., Winkeler, A., Hilker, R., Knoess, C., Rueger, A., Galldiks, N., Schaller, B., Sobesky, J., Kracht, L., et al. (2003). PET-based molecular imaging in neuroscience. *European journal of nuclear medicine and molecular imaging*, 30(7):1051–1065.

[Jan et al., 2004] Jan, S., Santin, G., Strul, D., Staelens, S., Assié, K., Autret, D., Avner, S., Barbier, R., Bardies, M., Bloomfield, P., et al. (2004). GATE: a simulation toolkit for PET and SPECT. *Physics in Medicine & Biology*, 49(19):4543.

[Jezierska et al., 2012] Jezierska, A., Chouzenoux, E., Pesquet, J.-C., and Talbot, H. (2012). A primal-dual proximal splitting approach for restoring data corrupted with Poisson-Gaussian noise. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1085–1088. IEEE.

[Jiang et al., 2019a] Jiang, J., Li, K., Wang, Q., Puterbaugh, K., Young, J. W., Siegel, S. B., O'Sullivan, J. A., and Tai, Y.-C. (2019a). A second-generation virtual-pinhole PET device for enhancing contrast recovery and improving lesion detectability of a whole-body PET/CT scanner. *Medical physics*, 46(9):4165–4176.

[Jiang et al., 2019b] Jiang, J., Samanta, S., Li, K., Siegel, S. B., Mintzer, R. A., Cho, S., Conti, M., Schmand, M., O'Sullivan, J., and Tai, Y.-C. (2019b). Augmented whole-body scanning via magnifying PET. *IEEE transactions on medical imaging*, 39(11):3268–3277.

[Jin et al., 2017] Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. (2017). Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522.

[Johnson et al., 2016] Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision – ECCV 2016*, pages 694–711, Cham. Springer International Publishing.

[Kabel et al., 1999] Kabel, J., Odgaard, A., van Rietbergen, B., and Huiskes, R. (1999). Connectivity and the elastic properties of cancellous bone. *Bone*, 24(2):115–120.

[Kalinyak et al., 2014] Kalinyak, J. E., Berg, W. A., Schilling, K., Madsen, K. S., Narayanan, D., and Tartar, M. (2014). Breast cancer detection using high-resolution breast PET compared to whole-body PET or PET/CT. *European journal of nuclear medicine and molecular imaging*, 41(2):260–275.

[Kangasmaa et al., 2011] Kangasmaa, T., Sohlberg, A., and Kuikka, J. T. (2011). Reduction of collimator correction artefacts with bayesian reconstruction in SPECT. *International journal of molecular imaging*, 2011.

[Karras et al., 2017] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

[Karras et al., 2020] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.

[Kim et al., 2019] Kim, B., Han, M., Shim, H., and Baek, J. (2019). A performance comparison of convolutional neural network-based image denoising methods: The effect of loss functions on low-dose CT images. *Medical Physics*, 46(9):3906–3923.

[Kingma and Ba, 2014] Kingma, D. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv e-prints*.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Krull et al., 2019] Krull, A., Buchholz, T.-O., and Jug, F. (2019). Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2129–2137.

[Kubo, 2019] Kubo, T. (2019). Vendor free basics of radiation dose reduction techniques for CT. *European Journal of Radiology*, 110:14–21.

[Lange et al., 1984] Lange, K., Carson, R., et al. (1984). EM reconstruction algorithms for emission and transmission tomography. *J Comput Assist Tomogr*, 8(2):306–16.

[Lecharlier and De Mol, 2013] Lecharlier, L. and De Mol, C. (2013). Regularized blind deconvolution with Poisson data. In *Journal of Physics: Conference Series*, volume 464, page 012003. IOP Publishing.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lehtinen et al., 2018] Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. (2018). Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*.

[Leuliet et al., 2021a] Leuliet, T., Friot--Giroux, L., Baaziz, W., Bretin, E., Ersen, O., Peyrin, F., Sixou, B., and Maxim, V. (2021a). Efficiency of TV-regularized algorithms in computed tomography with Poisson-Gaussian noise. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1294–1298.

[Leuliet et al., 2021b] Leuliet, T., Maxim, V., Peyrin, F., and Sixou, B. (2021b). Combining conditional GAN with VGG perceptual loss for bones CT image reconstruction. In *16th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine (Fully3D)*, number 281-284.

[Leuliet et al., 2022a] Leuliet, T., Maxim, V., Peyrin, F., and Sixou, B. (2022a). Impact of the training loss in deep learning based CT reconstruction of bone microarchitecture. *Medical Physics*.

[Leuliet et al., 2022b] Leuliet, T., Maxim, V., and Sixou, B. (2022b). Reconstruction tomographique et déconvolution aveugle en TEP: une méthode d'apprentissage profond hybride. In *XXVIIIème Colloque Francophone de Traitement du Signal et des Images (Gretsi 2022)*.

[Leuschner et al., 2020] Leuschner, J., Schmidt, M., Baguer, D., and Maaß, P. (2020). The LoDoPaB-CT dataset: A benchmark dataset for low-dose CT reconstruction methods.

[Levin et al., 2009] Levin, A., Weiss, Y., Durand, F., and Freeman, W. T. (2009). Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971. IEEE.

[Levin and Hoffman, 1999] Levin, C. S. and Hoffman, E. J. (1999). Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution. *Physics in Medicine & Biology*, 44(3):781.

[Li et al., 2020] Li, H., Schwab, J., Antholzer, S., and Haltmeier, M. (2020). Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005.

[Li et al., 2016] Li, Y., Defrise, M., Matej, S., and Metzler, S. D. (2016). Fourier rebinning and consistency equations for time-of-flight PET planograms. *Inverse Problems*, 32(9):095004.

[Li et al., 2019] Li, Y., Li, K., Zhang, C., Montoya, J., and Chen, G. (2019). Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions. *IEEE Transactions on Medical Imaging*, 38(10):2469–2481.

[Liang et al., 2018] Liang, K., Yang, H., and Xing, Y. (2018). Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction. *arXiv preprint arXiv:1804.04289*.

[Lim and Ye, 2019] Lim, S. and Ye, J. C. (2019). Blind Deconvolution Microscopy Using Cycle Consistent CNN with Explicit PSF Layer. In *Machine Learning for Medical Image Reconstruction*, pages 173–180, Cham. Springer International Publishing.

[Lucy, 1974] Lucy, L. B. (1974). An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745.

[Lunz et al., 2018] Lunz, S., Öktem, O., and Schönlieb, C.-B. (2018). Adversarial regularizers in inverse problems. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8516–8525, Red Hook, NY, USA. Curran Associates Inc.

[Lyman et al., 2005] Lyman, G. H., Giuliano, A. E., Somerfield, M. R., Benson III, A. B., Bodurka, D. C., Burstein, H. J., Cochran, A. J., Cody III, H. S., Edge, S. B., Galper, S., et al. (2005). American society of clinical oncology guideline recommendations for sentinel lymph node biopsy in early-stage breast cancer. *Journal of clinical oncology*, 23(30):7703–7720.

[Maas et al., 2013] Maas, A. et al. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer.

[Macmanamus et al., 2009] Macmanamus, M. et al. (2009). Use of PET and PET/CT for Radiation Therapy Planning: IAEA expert report 2006–2007. *Radiotherapy and Oncology*, 91(1):85–94.

[Markhardt et al., 2009] Markhardt, B. K., Gross, J. M., and Monu, J. (2009). Schatzker classification of tibial plateau fractures: use of CT and MR imaging improves assessment. *Radiographics*, 29(2):585–597.

[Matej et al., 2009] Matej, S., Surti, S., Jayanthi, S., Daube-Witherspoon, M. E., Lewitt, R. M., and Karp, J. S. (2009). Efficient 3-D TOF PET reconstruction using view-grouped histo-images: DIRECT—Direct image reconstruction for TOF. *IEEE Transactions on medical imaging*, 28(5):739–751.

[Mathews et al., 2013] Mathews, J. D., Forsythe, A. V., Brady, Z., Butler, M. W., Goergen, S. K., Byrnes, G. B., Giles, G. G., Wallace, A. B., Anderson, P. R., Guiver, T. A., et al. (2013). Cancer risk in 680 000 people exposed to computed tomography scans in childhood or adolescence: data linkage study of 11 million australians. *Bmj*, 346.

[Maxim et al., 2018] Maxim, V., Feng, Y., Banjak, H., and Bretin, E. (2018). Tomographic reconstruction from Poisson distributed data: a fast and convergent EM-TV dual approach. working paper or preprint.

[Mei et al., 2017] Mei et al., K. (2017). Is multidetector CT-based bone mineral density and quantitative bone microstructure assessment at the spine still feasible using ultra-low tube current and sparse sampling? *Eur Radiol*, 27:5261–5271.

[Mettler et al., 2000] Mettler, F. A., Wiest, P. W., Locken, J. A., and Kelsey, C. A. (2000). CT scanning: patterns of use and dose. *Journal of radiological Protection*, 20(4):353.

[Mettler Jr et al., 2008] Mettler Jr, F. A., Thomadsen, B. R., Bhargavan, M., Gilley, D. B., Gray, J. E., Lipoti, J. A., McCrohan, J., Yoshizumi, T. T., and Mahesh, M. (2008). Medical radiation exposure in the US in 2006: preliminary results. *Health physics*, 95(5):502–507.

[Mirza and Osindero, 2014] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[Moran et al., 2020] Moran, N., Schmidt, D., Zhong, Y., and Coady, P. (2020). Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12064–12072.

[Murtagh et al., 1995] Murtagh, F., Starck, J., and Bijauoui, A. (1995). Image restoration with noise suppression using a multiresolution support. *Astronomy and Astrophysics supplement*, 112:179–189.

[Nazarian et al., 2006] Nazarian, A., Stauber, M., Zurakowski, D., Snyder, B., and Müller, R. (2006). The interaction of microstructure and volume fraction in predicting failure in cancellous bone. *Bone*, 39(6):1196–1202.

[Nocedal and Wright, 1999] Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer.

[Obmann et al., 2020] Obmann, D., Schwab, J., and Haltmeier, M. (2020). Deep synthesis network for regularizing inverse problems. *Inverse Problems*, 37(1):015005.

[Oei et al., 2016] Oei, L., Koromani, F., Rivadeneira, F., Zillikens, M., and Oei, E. (2016). Quantitative imaging methods in osteoporosis. *Quantitative Imaging in Medicine and Surgery*, 6(6).

[Öktem, 2008] Öktem, O. (2008). Reconstruction methods in electron tomography.

[Otsu, 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern. Syst.*, 9(1):62–66.

[Pan et al., 2020] Pan, J., Dong, J., Liu, Y., Zhang, J., Ren, J., Tang, J., Tai, Y. W., and Yang, M. H. (2020). Physics-based generative adversarial models for image restoration and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

[Panin et al., 2006] Panin, V., Kehren, F., Rothfuss, H., Hu, D., Michel, C., and Casey, M. (2006). PET reconstruction with system matrix derived from point source measurements. *IEEE transactions on nuclear science*, 53(1):152–159.

[Perslev et al., 2019] Perslev, M., Dam, E. B., Pai, A., and Igel, C. (2019). One network to segment them all: A general, lightweight system for accurate 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 30–38. Springer.

[Pesek et al., 2012] Pesek, S., Ashikaga, T., Krag, L. E., and Krag, D. (2012). The false-negative rate of sentinel node biopsy in patients with breast cancer: a meta-analysis. *World journal of surgery*, 36(9):2239–2251.

[Peyrin and Engelke, 2012] Peyrin, F. and Engelke, K. (2012). CT imaging: Basics and new trends. *Handbook of Particle Detection and Imaging*, page 883.

[Pock and Chambolle, 2011] Pock, T. and Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *2011 International Conference on Computer Vision*, pages 1762–1769. IEEE.

[Popescu et al., 2004] Popescu, L. M., Matej, S., and Lewitt, R. M. (2004). Iterative image reconstruction using geometrically ordered subsets with list-mode data. In *IEEE Symposium Conference Record Nuclear Science 2004.*, volume 6, pages 3536–3540. IEEE.

[Qiao et al., 2019] Qiao, Z., Zhu, Y., Redler, G., and Tang, S. (2019). The integrated acceleration of the Chambolle-Pock algorithm applied to constrained TV minimization in CT image reconstruction. *Inverse Problems in Science and Engineering*.

[Ramdas et al., 2017] Ramdas, A., García Trillos, N., and Cuturi, M. (2017). On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47.

[Reader et al., 2020] Reader, A. J., Corda, G., Mehranian, A., da Costa-Luis, C., Ellis, S., and Schnabel, J. A. (2020). Deep learning for PET image reconstruction. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(1):1–25.

[Reader et al., 2003] Reader, A. J., Julyan, P. J., Williams, H., Hastings, D. L., and Zweit, J. (2003). EM algorithm system modeling by image-space techniques for PET reconstruction. *IEEE Transactions on Nuclear Science*, 50(5):1392–1397.

[Reader et al., 2002] Reader, A. J., Manavaki, R., Zhao, S., Julyan, P. J., Hastings, D. L., and Zweit, J. (2002). Accelerated list-mode EM algorithm. *IEEE Transactions on Nuclear Science*, 49(1):42–49.

[Reddin et al., 2018] Reddin, J. S., Scheuermann, J. S., Bharkhada, D., Smith, A. M., Casey, M. E., Conti, M., and Karp, J. S. (2018). Performance evaluation of the SiPM-based Siemens Biograph Vision PET/CT system. In *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, pages 1–5. IEEE.

[Rego et al., 2021] Rego, J., Kulkarni, K., and Jayasuriya, S. (2021). Robust lensless image reconstruction via PSF estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 403–412.

[Rehani and Berry, 2000] Rehani, M. M. and Berry, M. (2000). Radiation doses in computed tomography: the increasing doses of radiation need to be controlled. *Bmj*, 320(7235):593–594.

[Ren et al., 2020] Ren, D., Zhang, K., Wang, Q., Hu, Q., and Zuo, W. (2020). Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Reske and Kotzerke, 2001] Reske, S. N. and Kotzerke, J. (2001). FDG-PET for clinical use. *European journal of nuclear medicine*, 28(11):1707–1723.

[Richardson, 1972] Richardson, W. H. (1972). Bayesian-based iterative method of image restoration. *JoSA*, 62(1):55–59.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Rudin et al., 1992] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268.

[Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

[Sajedi et al., 2022] Sajedi, S., Bläckberg, L., Majewski, S., and Sabet, H. (2022). Limited-angle TOF-PET for intraoperative surgical applications: proof of concept and first experimental data. *Journal of Instrumentation*, 17(01):T01002.

[Sajedi et al., 2019] Sajedi, S., Bläckberg, L., Vittum, B., Devabhaktuni, A., Nejad, M. M., El Fakhri, G., Choi, H., and Sabet, H. (2019). Limited-angle TOF-PET for intraoperative surgical application. In *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–4. IEEE.

[Salimans et al., 2018] Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). Improving GANs using optimal transport. In *International Conference on Learning Representations*.

[Salome-Pateyron et al., 1997] Salome-Pateyron, M., Peyrin, F., Borras, G., Cloetens, P., and Laval-Jeantet, A. (1997). Quantification de l'architecture osseuse par microtomographie 3D utilisant le rayonnement synchrotron. In *SEIZIÈME COLLOQUE GRETSI*.

[Sawatzky et al., 2008] Sawatzky, A., Brune, C., Wubbeling, F., Kosters, T., Schafers, K., and Burger, M. (2008). Accurate EM-TV algorithm in PET with low SNR. In *2008 IEEE nuclear science symposium conference record*, pages 5133–5137. IEEE.

[Schlifske and Medeiros, 2016] Schlifske, D. and Medeiros, H. (2016). A fast GPU-based approach to branchless distance-driven projection and back-projection in cone beam CT. In *Medical Imaging 2016: Physics of Medical Imaging*, volume 9783, pages 742–749. SPIE.

[Shan et al., 2018] Shan, H., Zhang, Y., Yang, Q., Kruger, U., Kalra, M. K., Sun, L., Cong, W., and Wang, G. (2018). 3-D convolutional encoder-decoder network for low-dose CT via transfer learning from a 2-D trained network. *IEEE Transactions on Medical Imaging*, 37(6):1522–1534.

[Shepp and Vardi, 1982] Shepp, L. A. and Vardi, Y. (1982). Maximum likelihood reconstruction for emission tomography. *IEEE transactions on medical imaging*, 1(2):113–122.

[Sidky et al., 2012] Sidky, E. Y., Jørgensen, J. H., and Pan, X. (2012). Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm. *Physics in Medicine & Biology*, 57(10):3065.

[Sidky et al., 2006] Sidky, E. Y., Kao, C.-M., and Pan, X. (2006). Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT. *Journal of X-ray Science and Technology*, 14(2):119–139.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints*.

[Sözen et al., 2017] Sözen, T., Özışık, L., and Başaran, N. Ç. (2017). An overview and management of osteoporosis. *European journal of rheumatology*, 4(1):46.

[Staniszewska, 2002] Staniszewska, M. (2002). Evaluation of patient exposure in computerised tomogram in Poland. *Radiation protection dosimetry*, 98(4):437–440.

[Stute and Comtat, 2013] Stute, S. and Comtat, C. (2013). Practical considerations for image-based PSF and blobs reconstruction in PET. *Physics in Medicine & Biology*, 58(11):3849.

[Sureau et al., 2008] Sureau, F., Reader, A., Comtat, C., Leroy, C., Ribeiro, M., Buvat, I., and Trébossen, R. (2008). Impact of image-space resolution modeling for studies with the high-resolution research tomograph. *Journal of Nuclear Medicine*, 49(6):1000–1008.

[Tang et al., 2021] Tang, J., Mukherjee, S., and Schönlieb, C.-B. (2021). Stochastic primal-dual deep unrolling. *arXiv preprint arXiv:2110.10093*.

[Taschereau et al., 2011] Taschereau, R., Rannou, F. R., and Chatziioannou, A. F. (2011). A modeled point spread function for a noise-free system matrix. In *2011 IEEE Nuclear Science Symposium Conference Record*, pages 4102–4105. IEEE.

[Thielemans et al., 2010] Thielemans, K., Asma, E., Ahn, S., Manjeshwar, R. M., Deller, T., Ross, S. G., Stearns, C. W., and Ganin, A. (2010). Impact of PSF modelling on the convergence rate and edge behaviour of EM images in PET. In *IEEE Nuclear Science Symposuim Medical Imaging Conference*, pages 3267–3272.

[Tohka and Reilhac, 2008] Tohka, J. and Reilhac, A. (2008). Deconvolution-based partial volume correction in Raclopride-PET and Monte Carlo comparison to MR-based method. *NeuroImage*, 39(4):1570–1584.

[Tong et al., 2011] Tong, S., Alessio, A. M., Thielemans, K., Stearns, C., Ross, S., and Kinahan, P. E. (2011). Properties and mitigation of edge artifacts in PSF-based PET reconstruction. *IEEE Transactions on Nuclear Science*, 58(5):2264–2275.

[Ulyanov et al., 2017] Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. (2017). Deep Image Prior. *CoRR*, abs/1711.10925.

[United Nations Scientific Committee on the Effects of Atomic Radiation et al., 1996] United Nations Scientific Committee on the Effects of Atomic Radiation et al. (1996). Sources and effects of ionizing radiation. *UNSCEAR 1996 report to the General Assembly, with scientific annex*.

[van Aarle et al., 2015] van Aarle, W., Palenstijn, W. J., De Beenhouwer, J., Altantzis, T., Bals, S., Batenburg, K. J., and Sijbers, J. (2015). The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157:35–47.

[Villani, 2008] Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

[Wang et al., 2018] Wang, Y., Yu, B., Wang, L., Zu, C., Lalush, D. S., Lin, W., Wu, X., Zhou, J., Shen, D., and Zhou, L. (2018). 3d conditional generative adversarial networks for high-quality pet image estimation at low dose. *NeuroImage*, 174:550 – 562.

[Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

[Whiteley et al., 2020] Whiteley, W., Panin, V., Zhou, C., Cabello, J., Bharkhada, D., and Gregor, J. (2020). FastPET: near real-time reconstruction of PET histo-image data using a neural network. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(1):65–77.

[Wu et al., 2019] Wu, J., Huang, Z., Acharya, D., Li, W., Thoma, J., Paudel, D. P., and Van Gool, L. (2019). Sliced Wasserstein generative models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3708–3717.

[Xu et al., 2017] Xu, J., Gong, E., Pauly, J., and Zaharchuk, G. (2017). 200x low-dose PET reconstruction using deep learning. *arXiv preprint arXiv:1712.04119*.

[Yamamoto et al., 2007] Yamamoto, Y., Wong, T. Z., Turkington, T. G., Hawk, T. C., and Coleman, R. E. (2007). Head and neck cancer: dedicated FDG PET/CT protocol for detection—phantom and initial clinical studies. *Radiology*, 244(1):263–272.

[Yang et al., 2018] Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M. K., Zhang, Y., Sun, L., and Wang, G. (2018). Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE Transactions on Medical Imaging*, 37(6):1348–1357.

[Yang et al., 2017] Yang, W., Zhang, H., Yang, J., Wu, J., Yin, X., Chen, Y., Shu, H., Luo, L., Coatrieux, G., Gui, Z., and Feng, Q. (2017). Improving low-dose CT image using residual convolutional network. *IEEE Access*, 5:24698–24705.

[Yi et al., 2019] Yi, X., Walia, E., and Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552.

[You and Kaveh, 1996] You, Y.-L. and Kaveh, M. (1996). A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing*, 5(3):416–428.

[Zhang et al., 2017] Zhang, K., Zuo, W., Gu, S., and Zhang, L. (2017). Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938.

[Zhang et al., 2018] Zhang, R., Isola, P., Efros, A., Shechtman, E., and Wang, . (2018). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *arXiv e-prints*.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

FOLIO ADMINISTRATIF

THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

NOM : LEULIET DATE de SOUTENANCE : 04/11/2022

Prénom : Théo

TITRE : Deep Learning for tomographic reconstruction: study and application to computed tomography and positron emission imaging

NATURE : Doctorat Numéro d'ordre : 2022ISAL0093

Ecole doctorale : Électronique, Électrotechnique, Automatique (EEA)

Spécialité : Traitement du Signal et de l'Image

RESUME :

The purpose of tomography is to reconstruct a volume from its projections. In Computed Tomography (CT), X-rays are transmitted to a patient and attenuated by their tissues: the projections are obtained from the measured attenuation. For Positron Emission Tomography (PET), a radionuclide injected inside a patient emits a positron that generates two gamma photons in opposite directions. The projections correspond to the set of lines of response between each pair of simultaneously detected photons.
Tomographic reconstruction for PET or CT amounts to solving an inverse problem. Analytical methods are fast but their efficiency is limited when data are under-sampled or noisy. Iterative methods are efficient for noise and artefacts removal, but the computation time represents a major drawback for practical use.
Deep learning based methods have the potential to overcome those limits. The first objective of this thesis is to study the impact of the training loss on medical diagnosis-oriented evaluation metrics. We perform this study on bone microarchitecture CT imaging and show that in this case L1 loss should be used regarding all the considered metrics. Networks trained with perceptual losses show better transcription of structural features, at the cost of a deteriorated resolution. Adversarial losses improve the accuracy of the reconstruction in terms of density distribution.
We then focus on Time of Flight (TOF) PET data for intraoperative surgical applications; our aim is to design a reconstruction method to improve the detectability of small tumors in the context of breast cancer. We propose a neural network called PAVENET that simultaneously retrieves the image and the image-dependent point-spread function (PSF) from a poor-quality initial reconstruction. We present in this thesis the proof of concept for PAVENET with experiments on Monte-Carlo simulations reproducing acquisitions from an innovative detector studied in the Radiation Physics Instrumentation Laboratory (RPIL) in Boston.

MOTS-CLÉS : Deep learning, inverse problems, tomographic reconstruction, Computed Tomography (CT), Positron Emission Tomography (PET)

Laboratoire de recherche : Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé (CREATIS)

Directeur de thèse: Voichita MAXIM

Président de jury : Carole LARTIZIEN

Composition du jury :

Reader, Andrew	Professor	King's College London	Rapporteur
Talbot, Hugues	Professeur des Universités	CentraleSupélec	Rapporteur
Lartizien, Carole	Directrice de recherche	CNRS	Examinatrice
Sabet, Hamid	Assistant Professor	Harvard Medical School	Examinateur
Maxim, Voichita	Maître de Conférences HdR	INSA Lyon	Directrice de thèse
Sixou, Bruno	Maître de Conférences HdR	INSA Lyon	Co-directeur de thèse