

# Reconstruction tomographique et déconvolution aveugle en TEP : une méthode d'apprentissage profond hybride

Théo LEULIET, Voichita MAXIM, Bruno SIXOU

Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, CNRS, Inserm,  
CREATIS UMR 5220, U1294, F-69100, LYON, France

theo.leuliet@creatis.insa-lyon.fr

**Résumé** – La reconstruction d’image dans le cadre de la Tomographie par Emission de Positons (TEP) peut être modélisée par un problème inverse incluant un opérateur de convolution inconnu en plus de la transformée de Radon. Les méthodes d’apprentissage profond ont récemment montré des résultats probants pour la reconstruction tomographique ou pour la déconvolution aveugle ; cependant à notre connaissance, il n’existe pas d’approche combinant ces deux problématiques au sein d’une même structure. Nous proposons un réseau de neurones effectuant cette tâche jointe grâce à une méthode hybride entre apprentissage supervisé et auto-supervisé. Nous montrons sur des simulations que les performances d’un tel réseau sont significativement supérieures à celles d’une approche directe avec un UNET.

**Abstract** – Image reconstruction for Positron Emission Tomography (PET) can be modeled as an inverse problem that involves a convolution operator in addition to the Radon transform. Deep learning based methods recently showed convincing results for tomographic reconstruction or for blind deconvolution; however to the best of our knowledge, there is no approach that combines both of these tasks into a unique framework. We propose a neural network that solves this joint task thanks to a hybrid method that includes both supervised and self-supervised learning. We show on simulations that our network significantly outperforms a direct approach using a UNET.

## 1 Introduction

La Tomographie par Emission de Positons (TEP) est une technique d’imagerie médicale fonctionnelle de choix pour le diagnostic du cancer ou des troubles cérébraux, et pour la planification de la radiothérapie. Elle consiste à injecter un traceur radioactif à un patient afin de visualiser l’activité des cellules sur lesquelles ce traceur se fixe.

En s’annihilant avec des électrons du milieu, les positons libérés par ce traceur produisent deux photons en coïncidence, de directions opposées. La ligne définie par la détection des deux photons par le système est appelée ligne de réponse. La reconstruction tomographique vise à retrouver la position d’émission des positons à partir de l’ensemble des lignes de réponse. D’un point de vue mathématique, il s’agit d’un problème inverse linéaire qui peut être modélisé de façon simple par  $p = Af$ , où  $p$  correspond aux projections du volume d’intérêt  $f$ ,  $A$  étant l’opérateur direct. Ces projections sont en pratique entachées de bruit, considéré principalement comme étant de type Poisson en raison de la statistique de comptage utilisée.

Ces dernières années, les méthodes d’apprentissage profond ont montré des résultats probants pour la résolution de tels problèmes inverses [1] et pour la reconstruction TEP en particulier [2]. Des méthodes directes peuvent être utilisées pour améliorer la qualité d’une reconstruction analytique, en utilisant par exemple un encodeur-décodeur à partir de la rétro-projection filtrée (FBP) [3]. Les réseaux itératifs déroulés permettent eux de reproduire les schémas de méthodes itératives avec des réseaux de neurones [4] ; ils présentent l’avantage de s’appuyer sur le modèle direct à l’origine des données  $p$ . Il est cependant commun de considérer un opérateur de convolution dans le domaine image [5] afin d’intégrer au modèle les phénomènes physiques tels que la distance parcourue par le positon avant annihilation par un électron ou la non-colinéarité de la paire de photons émis.

Le modèle devient alors

$$p = AKf, \quad (1)$$

l’opérateur  $K$  correspondant ainsi à la PSF (fonction d’étalement du point) du système.

Lorsque  $K$  est inconnu et que  $A$  est l’opérateur identité, le problème (1) est appelé déconvolution aveugle. Une méthode de choix pour le résoudre est celle de Richardson-Lucy [6], dont la version déroulée a été publiée dans [7]. D’autres réseaux par apprentissage auto-supervisé [8] ou supervisé [9] ont également été proposés. Ces méthodes ont pour objectif de récupérer à la fois une estimation de l’image et de la PSF.

Lorsque  $A$  ne correspond pas à l’identité, il est possible d’appliquer l’algorithme de Richardson-Lucy en considérant de façon alternée une estimation de la PSF et de l’image [10]. En termes d’apprentissage profond, il est possible d’utiliser une méthode directe ne prenant pas en compte le modèle. En revanche, les méthodes déroulées existantes ne peuvent être utilisées sans la connaissance de la PSF. Inspirés des travaux de [9], nous proposons donc un réseau de neurones permettant de récupérer à la fois l’image et la PSF. La différence notable est que nous prenons la transformée de Radon pour l’opérateur  $A$  et non l’identité ; nous effectuons ainsi simultanément les tâches de reconstruction tomographique et de déconvolution aveugle par apprentissage profond, ce qui constitue une première à notre connaissance. Notre réseau, baptisé PAVENET, présente également la possibilité de s’actualiser lors de l’inférence grâce à un apprentissage auto-supervisé similaire à [8].

L’objectif de notre étude est de valider la méthode sur des données simulées en considérant le modèle (1). Nous comparons les performances de notre réseau avec une méthode directe dérivée de [3]. Nous évaluons l’apport de la connaissance du modèle dans le réseau de neurones, ainsi que la robustesse de la méthode lorsque les données de test présentent des caractéristiques

différentes de celles du jeu de données d’entraînement.

## 2 Méthodes et expériences numériques

### 2.1 Modèle

Dans (1), nous modélisons l’opérateur de convolution  $K$  par une gaussienne  $k$  qui dépend de l’objet  $f$ . Nous le considérons comme fixe spatialement afin de réduire le coût en ressources de calcul et de simplifier l’architecture du réseau utilisé. Nous retenons une distribution de type Poisson sur les données  $p$  afin de reproduire les fluctuations statistiques de l’émission radioactive.

### 2.2 UNET

Comme dans [3], nous cherchons à reconstruire l’image  $f$  avec un réseau utilisant la rétro-projection filtrée  $f_{\text{FBP}}$  en entrée. Nous utilisons une version légèrement modifiée du UNET [11], en remplaçant les fonctions d’activations ReLU (Rectified Linear Unit) par des Leaky ReLU de paramètre  $\alpha = 0.3$  pour l’encodeur et  $\alpha = 0.01$  pour le décodeur, la performance du réseau étant optimale dans cette configuration. Il n’y a pas de couches entièrement connectées - le UNET est initialement conçu pour des tâches de segmentation - et nous n’utilisons pas de fonction d’activation en sortie du réseau. Nous n’utilisons que 32 noyaux convolutifs pour la première couche du UNET au lieu de 64 afin de réduire le nombre de paramètres étant donné la complexité plus faible des données utilisées en comparaison à des images médicales réelles.

### 2.3 PAVENET

Le réseau que nous proposons est composé de deux blocs parallèles. Le premier bloc  $H_w$  est similaire à celui décrit précédemment et fait l’estimation de l’image  $\tilde{f}_w = H_w(f_{\text{FBP}})$ , à partir de la FBP. Le deuxième bloc  $G_\theta$  diffère simplement par la présence d’une sigmoïde en sortie du réseau ayant pour but de mettre la somme des pixels à 1. L’objectif de  $G_\theta$  est d’estimer le noyau de convolution de l’opérateur  $K$  par  $\tilde{k}_\theta = G_\theta(f_{\text{FBP}})$ . Ces deux blocs sont liés par une fonction coût de cohérence  $L_{\text{co}} = \|f_{\text{FBP}} - \tilde{k}_\theta * \tilde{f}_w\|_2^2$  où  $*$  est l’opérateur de convolution. Nous avons également effectué des essais où nous comparions  $A(\tilde{k}_\theta * \tilde{f}_w)$  aux projections en entrée  $p$ . Nous pouvions ainsi utiliser la distance de Kullback-Leibler (KL) pour prendre en compte la nature du bruit sur les projections. L’entraînement du réseau étant environ 10 fois plus long dans cette configuration, il était difficile de tester suffisamment d’hyper-paramètres pour obtenir une performance satisfaisante dans un délai raisonnable pour cette étude. Le réseau est par conséquent similaire à celui utilisé dans [9] à la différence près que l’entrée du réseau correspond à  $f_{\text{FBP}}$  et non à  $k * f$ . La FBP représente en effet une approximation de  $k * f$  qui a l’avantage d’être calculée rapidement.

Pour entraîner à la fois  $H_w$  et  $G_\theta$ , les meilleurs résultats numériques ont été obtenus avec une norme  $L_1$  sur la PSF et des normes  $L_2$  pour les autres termes de la fonction coût qui s’écrit ainsi

$$L_{f,k,\tilde{f}_w,\tilde{k}_\theta}(w,\theta) = \|f - \tilde{f}_w\|_2^2 + \lambda_1 \|k - \tilde{k}_\theta\|_1 + \lambda_2 \|f_{\text{FBP}} - \tilde{k}_\theta * \tilde{f}_w\|_2^2, \quad (2)$$

TABLE 1 – Paramètres sur les jeux de données d’entraînement et de test. Entre 6 et 12 disques d’activité comprise entre 6 et 16 Bq/mm<sup>3</sup> sont présents sur chaque image.

	Rayon disques (mm)	$\sigma_{\text{PSF}}$ (mm)
Entraînement	[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]

où les paramètres  $w$  et  $\theta$  sont mis à jour de façon alternée par rétro-propagation du gradient,  $\lambda_1$  et  $\lambda_2$  sont des paramètres de pondération.

Le calcul de  $L_{\text{co}}$  ne nécessitant aucune vérité-terrain, nous pouvons adopter une configuration du type auto-supervisé à l’inférence. De façon similaire à [8], les poids de notre réseau peuvent être mis à jour au moment de l’inférence en utilisant l’image en entrée comme référence. Nous effectuons donc l’apprentissage auto-supervisé avec

$$L_{\tilde{f}_w,\tilde{k}_\theta}^{\text{inf}}(w,\theta) = \|f_{\text{FBP}} - \tilde{k}_\theta * \tilde{f}_w\|_2^2. \quad (3)$$

L’originalité de notre méthode repose donc sur la possibilité de mettre à jour les poids appris par apprentissage supervisé au moment de l’inférence grâce à la définition de la fonction coût utilisée. Elle repose également sur l’utilisation du modèle (1) dans la définition de l’architecture puisque nous considérons la FBP en entrée du réseau.

Enfin, nous testons ce principe de mise à jour des poids à l’inférence pour le UNET simple. Cela revient à considérer l’identité pour  $\tilde{k}$  dans (3). Ce test permet de vérifier si la performance de cette méthode est uniquement due à la mise à jour des poids du réseau post-entraînement supervisé, ou si elle est liée à l’architecture de PAVENET.

### 2.4 Base de données et entraînement

Pour entraîner les réseaux, nous simulons 20 000 images de taille  $128 \times 128$  représentant des disques répartis aléatoirement, d’activité et taille variables. Pour les PSFs, nous considérons des noyaux gaussiens  $16 \times 16$  d’écart-types  $\sigma_x$  et  $\sigma_y$  variables. Après convolution entre l’image et le noyau, nous calculons les projections parallèles grâce à la transformée de Radon calculée avec la bibliothèque Astra [12]. Du bruit Poisson est ajouté à ces projections, et nous considérons la rétro-projection filtrée avec le filtre rampe en entrée des réseaux. Nous utilisons 20% de la base de données d’entraînement pour la validation.

Pour l’évaluation des méthodes, nous simulons quatre jeux de données différents, composés chacun de 100 images. Ils sont décrits sur la Table 1. Les tests B, C et D permettent d’évaluer la robustesse des méthodes sur des données qui ont été générées avec des caractéristiques différentes de celles du jeu d’entraînement.

L’entraînement des réseaux de neurones s’effectue avec l’algorithme d’optimisation Adam, sur 200 époques pour UNET et 250 pour PAVENET. Les 50 premières époques de ce dernier sont effectuées sans la fonction coût de cohérence  $L_{\text{co}}$  - pour la partie supervisée - car les estimations de  $\tilde{f}$  et  $\tilde{k}$  ne sont pas suffisamment précises au début de l’entraînement. Nous optimisons ensuite sur  $H_w$  et  $G_\theta$  de façon alternée. L’optimisation simultanée ainsi qu’un ratio différent entre les pas d’optimisation n’améliorent pas la performance. Nous utilisons une taille

de *batch* de 8. Tous les poids des réseaux sont initialisés avec la méthode de Kaiming He [13]. L’entraînement du UNET dure environ 4h et celui de PAVENET dure 14h sur un GPU Nvidia Tesla V100. La prédiction d’une image dure moins d’une seconde.

Pour la partie auto-supervisée, le réseau est optimisé selon (3) avec l’algorithme Adam, avec un temps de calcul d’environ 40 secondes pour 1000 époques sur le GPU mentionné. D’après nos tests, l’apprentissage auto-supervisé ne fonctionne pas lorsque  $H_w$  et  $G_\theta$  ont été entraînés de façon indépendante, i.e sans  $L_{co}$ .

Les taux d’apprentissage ainsi que  $\lambda_1$  et  $\lambda_2$  sont obtenus via l’optimisation par grille de valeurs sur le jeu de données de validation. Le UNET est ainsi entraîné avec un taux d’apprentissage de  $10^{-3}$ .  $H_w$  est pré-entraîné sur 50 époques avec un taux de  $10^{-3}$  puis  $10^{-5}$  pour les suivantes.  $G_\theta$  est pré-entraîné sur 50 époques avec un taux de  $10^{-4}$  puis  $10^{-5}$ . Nous présentons les résultats pour  $\lambda_1 = 10^5$ ,  $\lambda_2 = 10^{-2}$ .

Afin d’obtenir des résultats statistiquement significatifs, nous effectuons 10 initialisations différentes des poids des réseaux. Nous présentons les résultats moyennés ainsi que les fluctuations statistiques liées à l’initialisation aléatoire des réseaux.

### 3 Résultats

La Figure 1 représente un exemple des images reconstruites sur le jeu de données A. Le UNET reconstruit correctement l’image dans ce cas, mais l’activité des disques semble sous-estimée par rapport à la référence, contrairement à PAVENET qui semble plus précis.

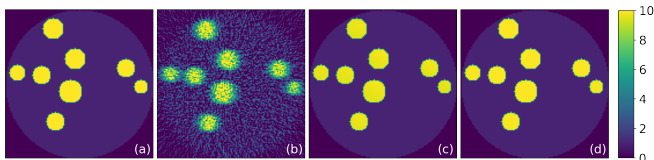


FIGURE 1 – Test A : (a) Référence, (b) FBP, (c) UNET, (d) PAVENET.

Sur le jeu de données B, nous observons les limites du UNET qui reconstruit des disques plus larges que ceux de l’image référence, comme illustré sur la Figure 2. Le rayon des disques reconstruits semble être le rayon minimal pour lequel le réseau a été entraîné. L’activité par pixel se retrouve ainsi diminuée. Lorsque le modèle direct est pris en compte dans l’entraînement, ce défaut est en partie corrigé comme nous pouvons le voir sur l’image (c) représentant la prédiction  $\hat{f}^0$  de PAVENET dans laquelle la taille des rayons est inférieure à ceux du UNET. L’adaptation des poids à l’inférence est particulièrement efficace comme le montre la prédiction de PAVENET après 3 000 époques d’entraînement auto-supervisé sur la Figure 2 (d). Nous observons sa capacité à retrouver des disques de rayon inférieur aux rayons utilisés pendant l’entraînement initial. Ceci est rendu possible par la capacité du réseau à prédire correctement la PSF comme le montre la Figure 2, ce qui rend la fonction coût de cohérence efficace.

Pour évaluer et comparer les méthodes quantitativement, nous utilisons le PSNR (Peak Signal to Noise Ratio), le SSIM (Structural SIMilarity index) et le CRC (Concentration Recovery Coefficient). Nous définissons ce dernier comme  $CRC = \hat{f}_{act}/f_{act}$  où nous considérons uniquement l’activité moyenne à l’intérieur des disques de l’image. Ce ratio doit se rapprocher de 1 et nous

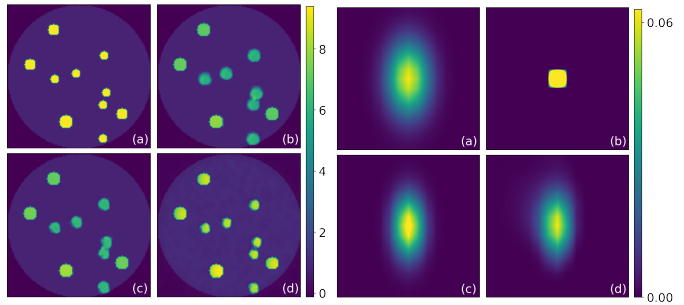


FIGURE 2 – Test B : Image (à gauche) et PSF (à droite) pour (a) la référence, (b) UNET, (c) PAVENET, (d) PAVENET après 3000 époques d’entraînement auto-supervisé.

présentons donc les résultats pour  $CRC_{err} = |CRC - 1|$ . La mise à jour de PAVENET à l’inférence avec la fonction coût de cohérence entraîne l’introduction de zones légèrement bruitées liées à l’utilisation de la FBP comme référence. Ce phénomène n’affecte que les zones où les disques ne sont pas présents mais tend à faire diminuer la valeur du SSIM. Nous proposons donc une métrique SSIM corrigée (SSIM Corr.), qui calcule le SSIM entre les images où tous les pixels n’appartenant pas à un disque sont mis à 0. Cela permet de comparer la structure des images en se focalisant sur les zones d’intérêt.

Sur la Table 2 nous représentons, pour chacun des 4 jeux de données test, la moyenne de ces métriques. Les résultats pour l’apprentissage auto-supervisé sont considérés après 1 000 époques (noté Inf. 1 000). La variation indiquée correspond à l’écart-type de cette valeur sur les dix réseaux entraînés. Notons également que nous n’observons pas de différence entre les méthodes quant à la variabilité des métriques sur les 100 images de test. Cette variabilité représente un écart-type de moins de 10% pour chaque métrique et chaque réseau testé.

Sur le test A, nous retrouvons la bonne performance du UNET en terme de suppression du bruit et de précision sur la structure des images reconstruites, bien que PAVENET soit encore plus performant pour le PSNR et le SSIM. Nous observons également une performance supérieure de PAVENET en terme de CRC avec moins de 1% d’erreur. Dans cette configuration, l’apprentissage auto-supervisé de PAVENET semble peu efficace, ce qui peut se comprendre par l’absence de régularisation ici.

Sur des disques de rayons inférieurs, l’apprentissage auto-supervisé avec PAVENET améliore significativement le PSNR et le CRC, avec notamment une erreur sur l’activité qui est réduite de près de 4% en comparaison au UNET. Cette amélioration s’observe aussi pour les jeux de données C et D. L’amélioration par rapport au UNET en termes de SSIM n’est visible que lorsque nous utilisons la version corrigée de la métrique, pour les raisons expliquées précédemment.

Enfin, la Figure 3 montre l’évolution des métriques en fonction du nombre d’époques lorsque nous utilisons l’apprentissage auto-supervisé pour mettre à jour les prédictions effectuées par UNET et PAVENET, sur le test B. Nous pouvons voir que l’apprentissage auto-supervisé est contre-productif pour le UNET. L’apprentissage auto-supervisé ne fonctionne donc qu’à la condition que la connaissance du modèle soit incluse et que la prédiction de la PSF soit précise, comme c’est le cas ici pour PAVENET.

TABLE 2 – Moyennes des métriques sur chaque test. Les écarts représentés sont liés à l’initialisation aléatoire des réseaux.

Méthodes	Test A				Test B			
	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	<b>0.999 ± 0.001</b>	21.89 ± 0.14	0.883 ± 0.029	22.70 ± 0.95	0.931 ± 0.003
PAVENET	<b>50.29 ± 0.24</b>	<b>0.993 ± 0.001</b>	<b>0.54 ± 0.12</b>	<b>0.999 ± 0.001</b>	21.79 ± 0.26	<b>0.924 ± 0.004</b>	21.50 ± 1.08	0.930 ± 0.005
Inf. 1000	43.31 ± 0.32	0.932 ± 0.017	1.06 ± 0.42	0.997 ± 0.004	<b>22.58 ± 0.28</b>	0.899 ± 0.015	<b>19.25 ± 1.13</b>	<b>0.940 ± 0.004</b>
Méthodes	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	18.50 ± 0.25	<b>0.818 ± 0.008</b>	7.81 ± 0.78	0.903 ± 0.009	26.90 ± 0.38	<b>0.947 ± 0.007</b>	<b>8.57 ± 0.87</b>	0.971 ± 0.005
Inf. 1000	<b>20.96 ± 0.42</b>	0.783 ± 0.016	<b>3.27 ± 0.78</b>	<b>0.935 ± 0.006</b>	<b>27.64 ± 0.20</b>	0.891 ± 0.015	9.01 ± 0.79	<b>0.980 ± 0.003</b>

## 4 Discussion et conclusion

Sur des simulations TEP simplifiées, nous avons montré que le réseau proposé améliore la performance pour les métriques considérées. Notre méthode permet notamment de surpasser les limites du UNET lorsqu’il est testé sur des données différentes de celles avec lesquelles il a été entraîné. La capacité de PAVENET à retrouver des disques de rayon réduit après apprentissage auto-supervisé est particulièrement intéressante étant donné qu’il s’agit d’une tâche primordiale en pratique, lorsque l’on cherche à détecter des objets pouvant par exemple correspondre à des tumeurs.

La performance de l’apprentissage auto-supervisé de PAVENET est fortement influencée par la régularisation faite ou non sur l’image et sur la PSF. Nous avons effectué des tests en ajoutant des termes de régularisation sur (3) permettant de forcer les sorties du réseau à rester proches des prédictions initiales ; nous avons cependant observé que réduire le nombre d’époques pour la partie auto-supervisée et diminuer le taux d’apprentissage représentent des solutions équivalentes en terme de régularisation. Il est cependant probable que la recherche d’une combinaison plus optimale de ces paramètres - terme de régularisation, taux d’apprentissage et nombre d’époques - améliorerait la performance de PAVENET. De la même façon, nous observons une différence de performance de l’apprentissage auto-supervisé sur le test A et sur les tests B, C et D. Notre solution gagnerait par conséquent à utiliser une méthode optimisée pour le choix de la régularisation en fonction des images qui sont données en entrée du réseau.

L’adaptation de notre réseau à des PSFs non constantes augmente les temps de calcul et complexifie l’optimisation des hyper-paramètres, ce qui nous pousse à ne pas considérer cette configuration dans cette étude. L’utilisation de la fonction coût dans le domaine des projections pourrait améliorer les performances du réseau, notamment parce qu’elle permettrait la prise en compte du modèle de bruit Poisson avec la distance de Kullback-Leibler, mais cette méthode augmente également les

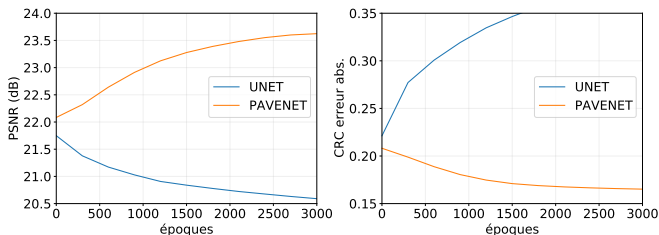


FIGURE 3 – Evolution des métriques pendant l’apprentissage auto-supervisé.

temps de calcul.

Enfin, les tests présentés dans cette étude sont effectués sur des données simplifiées ; ces simulations nous permettent d’obtenir des résultats statistiquement significatifs et d’avoir une certaine liberté sur les paramètres testés. Pour valider la performance du réseau, la prochaine étape consiste à l’appliquer sur des données réalistes pour la TEP. Il est particulièrement important d’évaluer la performance de la méthode lorsque les PSFs références doivent être estimées pour construire la base de données. Des tests sur données TEP avec prise en compte du temps de vol pour l’imagerie peropératoire sont ainsi en cours.

## Références

- [1] S. Arridge et al. Solving inverse problems using data-driven models. *Acta Numerica*, 28 :1–174, 2019.
- [2] A.J. Reader et al. Deep learning for PET image reconstruction. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(1) :1–25, 2020.
- [3] J. Xu et al. 200x low-dose PET reconstruction using deep learning. *arXiv preprint arXiv :1712.04119*, 2017.
- [4] K. Gong et al. EMnet : an unrolled deep neural network for PET image reconstruction. In *Medical Imaging 2019 : Physics of Medical Imaging*, volume 10948, pages 1203–1208. SPIE, 2019.
- [5] F.C. Sureau et al. Impact of image-space resolution modeling for studies with the high-resolution research tomograph. *Journal of Nuclear Medicine*, 49(6) :1000–1008, 2008.
- [6] D.A. Fish et al. Blind deconvolution by means of the Richardson–Lucy algorithm. *JOSA A*, 12(1) :58–65, 1995.
- [7] C. Agarwal et al. 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, 2020.
- [8] D. Ren et al. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [9] J.D. Rego et al. Robust lensless image Reconstruction via PSF estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 403–412, January 2021.
- [10] L. Lecharlier and C. De Mol. Regularized blind deconvolution with Poisson data. In *Journal of Physics : Conference Series*, volume 464, page 012003. IOP Publishing, 2013.
- [11] Ronneberger et al. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [12] Van Aarle et al. Fast and flexible X-ray tomography using the ASTRA toolbox. *Opt. Express*, 24(22) :25129–25147, 2016.
- [13] Kaiming He et al. Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Ce travail a été réalisé dans le cadre du projet LABEX PRIMES (ANR-11-IDEX-0007) de l’Université de Lyon.