



Institut Laue-Langevin



ENSIMAG

Rapport de stage Assistant Ingénieur (AI)

Effectué chez Institut Laue-Langevin (ILL)

Débruitage des images par Deep Learning

BADAOUI Abdessamad
2e année – Option MMIS

22 mai 2023 – 18 août 2023

Institut Laue-Langevin
71 Av. des Martyrs,
CS 20156
38042 GRENOBLE Cedex 9

Responsable de stage
LE GOC Yannick
Tuteur de l'école
HAHMANN Stefanie

Contents

1	Introduction	3
2	Contexte	3
3	Sujet de stage	4
3.1	Problématique	4
3.1.1	Débruitage par les méthodes classiques	4
3.1.2	Débruitage par Deep Learning	5
3.1.3	Le format des données	6
3.2	Les objectifs précis attendus	6
3.3	L'état de l'art des solutions existantes	7
3.3.1	Noise2Noise	7
3.3.2	Noise2Void	8
3.3.3	Noise2Self	9
3.3.4	Noisier2Noise	9
3.4	Solutions choisies	10
4	Implémentation de la solution	10
4.1	Architecture haut niveau de la solution	10
4.2	Détails techniques de l'implémentation	11
4.2.1	Les coupes par Horace Matlab	11
4.2.2	Le processus de débruitage	12
4.2.3	Préparation des données pour visualisation	13
4.3	Test et évaluation	15
5	Résultats obtenus	16
5.1	Comparaison des résultats des 4 méthodes	16
5.2	Regard critique sur les résultats obtenus	18
6	Bilan personnel	19
7	Conclusoin	19
8	Références	21
9	Annexes	22
9.1	Le dataset bnfs10K	22
9.2	Le dataset TbTiO1p5K :	24

1 Introduction

Dans le domaine des expériences de diffusion de neutrons, les détecteurs utilisés jouent un rôle crucial dans la collecte des données. Cependant, ces détecteurs ne sont pas exempts de bruit, ce qui peut rendre les images obtenues moins interprétables. Le bruit peut être causé par divers facteurs, tels que les conditions expérimentales et le faible flux intrinsèque des faisceaux de neutrons. La présence de bruit dans les images des détecteurs peut compromettre la qualité des données et entraver l'analyse précise des phénomènes étudiés.

Afin de résoudre ce problème, des outils de débruitage sont nécessaires. Ces outils visent à réduire le bruit indésirable tout en préservant les caractéristiques importantes des images. Au fil des années, les méthodes de débruitage ont connu des avancées significatives, notamment avec l'émergence des techniques de Deep Learning. Ces méthodes utilisent des réseaux neuronaux profonds pour apprendre à débruiter les images de manière automatique et efficace. Elles permettent donc de surmonter les limitations des méthodes traditionnelles de débruitage et offrent des opportunités passionnantes pour améliorer la compréhension de la physique des matériaux à travers les expériences de diffusion de neutrons. Bien que ces projets soient plutôt destinés à l'imagerie médicale, la similitude avec nos images détecteurs nous invite à vouloir expérimenter ces techniques sur nos données afin d'améliorer notre processus de débruitage.

L'objectif du stage est de faire une comparaison de ces techniques avec les techniques classiques sur des données détecteurs de l'ILL.

2 Contexte

L'institut Laue-Langevin (ILL), fruit d'une collaboration entre la France, le Royaume-Uni et l'Allemagne, est mondialement reconnu en tant que leader dans le domaine de la recherche neutronique. Son objectif principal est l'étude de la matière, englobant ainsi un large éventail de disciplines fondamentales telles que la physique nucléaire, la biologie, la chimie verte, la cristallographie, la spectroscopie, la physique des matériaux et de la matière condensée, tout en s'intéressant également aux défis sociaux tels que les énergies, la santé et l'environnement.

L'institut dispose de plusieurs instruments et appareils [2] qui permettent de manipuler et faire des expériences sur la matière. Ces instruments sont alimentés par le réacteur à haut flux le plus puissant au monde, qui est exclusivement dédié à la recherche [1]. Cependant, ces instruments sont relativement complexes, composés de différentes composantes mécaniques et électroniques gérées par de multiples paramètres. **Le Service de Contrôle des Instruments (SCI)** joue un rôle essentiel en fournissant un support informatique et électronique pour les instruments. Ce service est également chargé de veiller au bon déroulement des manipulations expérimentales, en évitant tout problème technique. Le chef de service est responsable de la construction de nouveaux instruments, de la gestion des plans de construction et du budget associé.

Le Service de Contrôle des Instruments (SCI) de l'ILL est composé d'une équipe d'environ quinze employés, regroupant des informaticiens et des électroniciens. Ils jouent un rôle essentiel dans l'acquisition des données expérimentales, en assurant la programmation des cartes d'acquisition, la fabrication des circuits imprimés, l'implémentation des drivers et le contrôle des moteurs et des capteurs. Pour gérer efficacement tous

ces éléments, le service a développé un logiciel nommé **NOMAD** [3], doté de modules spécifiques pour chaque instrument. NOMAD permet de conserver un historique détaillé des expériences et fournit des informations en temps réel pendant les manipulations, telles que la température des moteurs et la position des capteurs. Les informaticiens du service SCI sont responsables de la maintenance et de ce logiciel.

Parmis les grandes instruments à l'ILL est le **IN5B**. Ce dernier est un spectromètre de géométrie directe à temps de vol (ToF or Time-of-Flight) de haute précision. Il est utilisé pour étudier les processus de transfert d'énergie à faible énergie en fonction du transfert de moment. Typiquement, cet instrument est utilisé pour des mesures dans la région de faible transfert d'énergie et de moment, avec une résolution d'énergie d'environ 1 pour cent (par exemple, diffusion quasi-élastique dans les solides, les liquides, les cristaux moléculaires et diffusion inélastique avec de petits transferts d'énergie dans la plage de 10 μeV à 100 meV). L'instrument IN5B offre une multitude d'applications passionnantes dans divers domaines de recherche. Il est largement utilisé pour étudier la diffusion à la fois locale et à longue distance dans des systèmes désordonnés tels que les liquides, les cristaux moléculaires, les solides amorphes, les polymères et les systèmes hydrogène-métal, ainsi que les conducteurs ioniques. De plus, il permet d'explorer la dynamique de la "matière molle", y compris les gels, les protéines et les membranes biologiques. L'instrument IN5B joue également un rôle essentiel dans l'étude de la dynamique des liquides quantiques, le tunnelling rotationnel dans les cristaux moléculaires, la séparation du champ cristallin, les dynamiques de spin dans les supraconducteurs à haute température critique, ainsi que les phénomènes de diffusion critique dans les gaz denses et les solides. Ces différentes applications témoignent de la polyvalence et de la portée scientifique de l'instrument IN5B dans de nombreux domaines de recherche.

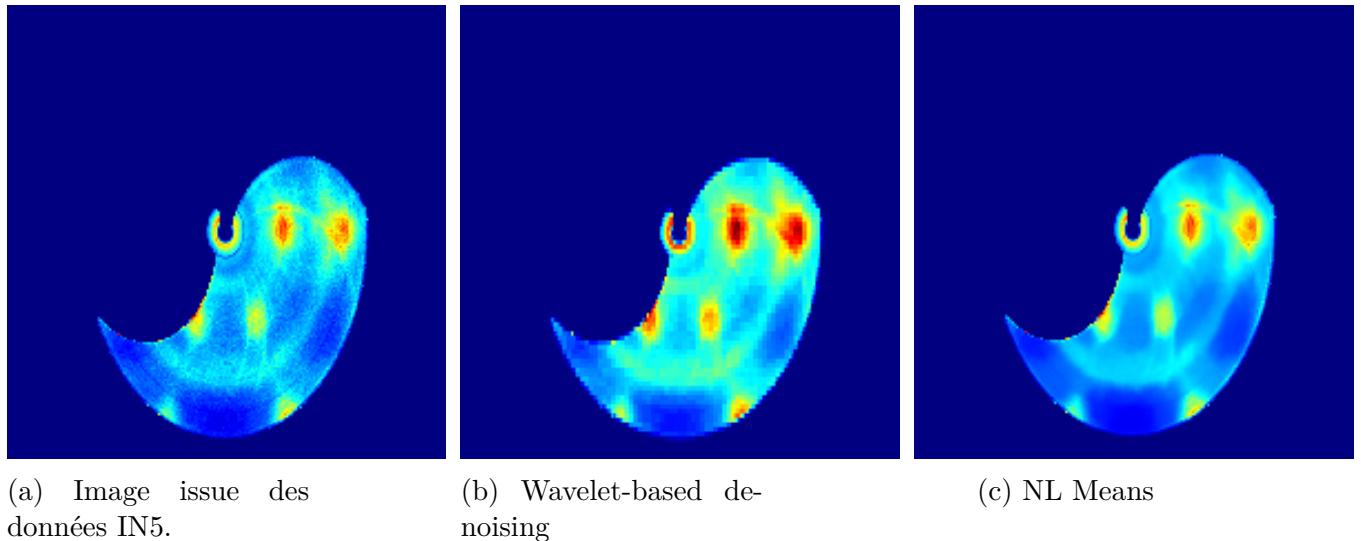
Ce sont les données provenant de cet instrument qui seront soumises à un processus de débruitage, qui intervient après différentes étapes de préparation des données pour faciliter leur exploitation.

3 Sujet de stage

3.1 Problématique

3.1.1 Débruitage par les méthodes classiques

Le méthodes classiques comme **Non Local Means** et **Wavelet-based denoising** ont été largement utilisées pour faire le débruitage des images. Ces méthodes présentent certaines caractéristiques intéressantes, notamment une rapidité d'exécution, l'absence d'entraînement requis, moins de paramètres à régler et une facilité d'interprétation. Cependant, malgré ces avantages, les résultats obtenus avec ces méthodes ne sont pas toujours satisfaisants. Les images débruitées peuvent encore présenter des artefacts, une perte de détails ou une altération des propriétés physiques des données (Voir exemple dans la figure 1). On remarque bien sur cet exemple que les taches rouges ne sont pas très bien débruitées par la méthode Wavelet, car il y a l'apparition de blocs de pixels, également connus sous le nom de pixellisation. Quant à Nl means, il s'en sort plutôt bien, mais il n'a pas débruité de manière optimale les zones rouges. Ces méthodes donc ne donnent pas vraiment des résultats satisfaisantes pour pouvoir les utiliser et les interpréter.



(a) Image issue des données IN5.

(b) Wavelet-based de-noising

(c) NL Means

Figure 1: Débruitage d'une image issue des données de l'instrument IN5 par les méthodes de débruitage classiques

3.1.2 Débruitage par Deep Learning

Les méthodes classiques présentent plusieurs limitations car elles effectuent le débruitage sans entraînement. En termes de quantité d'information, les méthodes de Deep Learning, basées sur l'entraînement, sont attendues pour fournir de meilleurs résultats.

Tout d'abord, considérons un vaste ensemble de données (X_i, Y_i) comprenant des paires d'images, associant une image bruitée X_i à son homologue débruité Y_i . Le principe de notre approche repose sur l'utilisation d'un réseau de neurones profond, où chaque pixel de l'image bruitée est traité en tant qu'entrée du réseau (Voir Figure 2). Ce réseau est composé de multiples couches, et chaque connexion entre deux nœuds est représentée par un poids spécifique. Ainsi, l'information subit de multiples transformations en passant par les couches jusqu'à atteindre la couche finale.

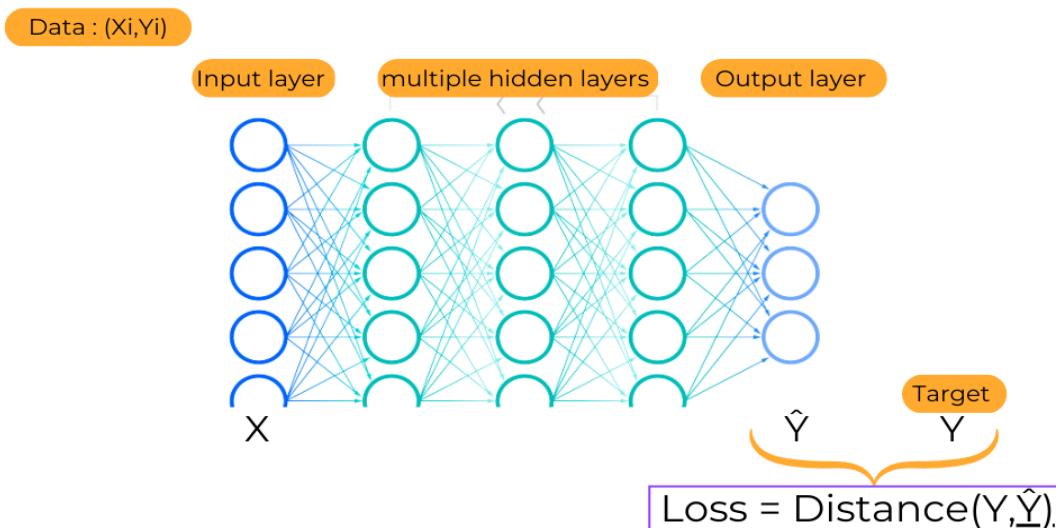


Figure 2: Représentation haut niveau d'un réseau de neurones pour le débruitage d'images

Au départ, lorsque les poids du réseau sont initialisés de manière arbitraire, l'image résultante \hat{Y} n'a généralement aucun similitude avec l'image d'origine. Cependant, ces

poids sont mis à jour à chaque passage d'une image dans le réseau. Cela se produit grâce à la comparaison avec l'image débruitée associée, représentée par une fonction de loss donnée qui représente la distance entre l'image de prédition et l'image propre. En utilisant une méthode numérique comme une descente de gradient stochastique, nous pouvons minimiser cette fonction et améliorer progressivement les performances du réseau. Une fois que le réseau est correctement entraîné, il peut être utilisé pour prédire le débruitage d'autres images.

Cependant, deux contraintes se présentent à nous :

Premièrement, le réseau nécessite un entraînement sur une très grande base de données pour généraliser efficacement. Malheureusement, à l'ILL, nous ne disposons pas d'une telle base de données, ce qui signifie que nous devons effectuer l'entraînement à chaque fois que nous souhaitons débruiter des données spécifiques.

Deuxièmement, nous ne disposons pas toujours des images débruitées pour effectuer cet entraînement, ce qui est également notre cas. Cependant, nous pouvons surmonter cette limitation en utilisant des méthodes de Deep Learning qui ne nécessitent pas d'images propres pour l'entraînement. Ces méthodes remplacent l'image débruitée cible par une autre entité, ce qui sera notre approche dans cette étude

Le but de mon stage est donc de découvrir ces nouvelles méthodes, d'effectuer un ajustement fin des paramètres (fine-tuning), de les tester et de les comparer aux méthodes traditionnelles.

3.1.3 Le format des données

L'instrument IN5 est utilisé pour mener des expériences de diffusion, et les données collectées sont enregistrées dans des fichiers au format .nxs (Nexus). Ces fichiers sont ensuite traités par le logiciel Mantid, qui permet leur réduction, puis rassemblés à l'aide de la bibliothèque Horace de Matlab. Les données finales sont représentées sous la forme de fichiers .sqw Horace, qui contiennent des données à cinq dimensions (**I, h, k, l, E**). Les trois dimensions (h, k, l) sont exprimées en angstrom (\AA) et caractérisent l'espace réciproque de l'échantillon placé à l'intérieur de l'instrument. La quatrième dimension, l'énergie du neutron (E), est exprimée en milliélectron-volts (meV). Lorsque le neutron interagit avec l'échantillon, une intensité sans dimension (I) est mesurée par les capteurs de l'instrument.

Ce format de données n'est pas courant, car les outils que nous avons à disposition sont destinés au débruitage d'images en 2 dimensions, alors que nous disposons des données à 4 dimensions. Par conséquent, nous sommes confrontés à une problématique supplémentaire consistant à définir un workflow approprié pour réaliser le débruitage de ces données.

3.2 Les objectifs précis attendus

Les objectifs précis de mon stage sont les suivants :

Tester et trouver les algorithmes les plus appropriées avec les paramètres adéquats pour obtenir les meilleurs résultats tout en respectant un temps d'exécution acceptable : Il est important de noter que l'outil développé sera spécifiquement conçu pour traiter les données propres à l'ILL (Institut Laue-Langevin). Cela signifie que l'entraînement doit être effectué sur les données spécifiques à l'ILL.

La définition d'un workflow pour résoudre la problématique : Comme il a été mentionné précédemment, les données sont en 4 dimensions, et il est donc nécessaire de trouver une solution pour pouvoir utiliser les méthodes de débruitage d'images sur ce type de données.

Assurer que l'outil de débruitage conserve autant que possible les propriétés physiques des données : Les matériaux étudiés à l'ILL présentent plusieurs caractéristiques physiques géométriques et énergétiques. Ces caractéristiques, parfois très fines dans les images, doivent être débruitées sans être totalement éliminées par le processus de débruitage. Il est essentiel de préserver ces propriétés physiques lors du débruitage.

En résumé, mon objectif principal est de trouver les meilleures méthodes et paramètres, en termes de temps d'exécution et de précision, pour débruiter les données spécifiques à l'ILL, tout en préservant les caractéristiques physiques importantes présentes dans ces données.

3.3 L'état de l'art des solutions existantes

Définition

Le débruitage d'image est la tâche consistante à examiner une image bruitée $\mathbf{x} = \mathbf{s} + \mathbf{n}$ afin de la séparer en deux composants : son signal \mathbf{s} et le bruit de dégradation du signal \mathbf{n} que nous souhaitons éliminer.

Les méthodes de débruitage reposent généralement sur l'hypothèse que les valeurs des pixels dans \mathbf{s} ne sont pas statistiquement indépendantes. Autrement dit, observer le contexte de l'image d'un pixel non observé peut nous permettre de faire des prédictions sur l'intensité du pixel. Dans cette partie, nous allons présenter quatre méthodes de Deep Learning qui permettent de surmonter la contrainte d'avoir des images propres pour l'entraînement :

3.3.1 Noise2Noise

Au lieu d'entraîner un CNN (un réseau de neurones convolutionnel) pour faire correspondre des entrées bruitées à des images nettes de référence, la méthode **Noise2Noise (N2N)** tente d'apprendre une correspondance entre des paires de versions indépendamment dégradées de la même image d'entraînement, c'est-à-dire $(\mathbf{s} + \mathbf{n}, \mathbf{s} + \mathbf{n}')$, qui intègrent le même signal \mathbf{s} , mais avec des bruits indépendants \mathbf{n} et \mathbf{n}' . Naturellement, un réseau neuronal ne peut pas apprendre à prédire parfaitement une image bruitée à partir d'une autre. Cependant, les réseaux formés sur cette tâche d'entraînement impossible peuvent produire des résultats qui convergent vers les mêmes prédictions que les réseaux traditionnellement formés qui ont accès aux images nettes de référence. Dans les cas où les données de référence sont physiquement impossibles à obtenir, N2N peut toujours permettre l'entraînement de réseaux de débruitage. Cependant, cela nécessite que deux images capturant le même contenu \mathbf{s} avec des bruits indépendants (\mathbf{n}, \mathbf{n}') puissent être acquises.

Malgré ces avantages de l'entraînement N2N, il existe au moins deux inconvénients à cette approche : (i) l'entraînement N2N nécessite la disponibilité de paires d'images bruitées, et (ii) l'acquisition de telles paires avec un signal \mathbf{s} (quasi) constant n'est possible que pour des scènes (quasi) statiques.

3.3.2 Noise2Void

En plus de l'hypothèse de régularité du signal, une autre hypothèse est formulée pour cette méthode : le bruit est conditionnellement indépendant d'un pixel à l'autre, étant donné le signal. C'est une hypothèse statistique qui signifie que le bruit n'a pas une forme géométrique spécifique dépendant du signal.

Le principe fondamental de **Noise2Void (N2V)** repose sur le fait que les images bruitées contiennent généralement des informations redondantes et répétitives. En exploitant ces motifs, N2V entraîne un réseau de neurones pour prédire les pixels manquants dans une image bruitée en se basant sur les informations des pixels voisins. Cela permet au réseau de débruitage de restaurer les détails de l'image (**par régularité du signal**) tout en supprimant efficacement le bruit indésirable (**par indépendance entre le bruit et le signal**). Le processus d'entraînement de N2V est conçu pour exploiter au mieux les caractéristiques locales de l'image, ce qui le rend particulièrement adapté aux images contenant du bruit complexe et variable.

En termes d'équations, nous avons les trois hypothèses suivantes :

Hypothèse 1 : On peut toujours gagner de l'information sur un pixel donné en regardant ses pixels voisins :

$$p(s_i|s_j) \neq p(s_i) \quad (1)$$

Hypothèse 2 : Le bruit est indépendant du signal :

$$p(n|s) = \prod_i p(n_i|s_i) \quad (2)$$

Hypothèse 3 : On suppose que le bruit a une moyenne nulle :

$$E[n_i] = 0 \quad (3)$$

Puisque nous supposons que le bruit est indépendant pixel par pixel étant donné le signal, les pixels voisins n'apportent aucune information sur la valeur n_i (la valeur du bruit au pixel masqué dans le patch autour de cette valeur), et donc il est impossible pour le réseau de produire une estimation meilleure que la valeur moyenne a priori. Cependant, le signal est supposé contenir des dépendances statistiques, et donc le réseau peut estimer la valeur s_i d'un pixel en regardant son voisinage. Ceci est faite en minimisant la fonction de loss suivante :

$$\arg \min_{\theta} \sum_i \sum_j L \left(f \left(\tilde{x}_{\text{RF}(i)}^j; \theta \right), x_i^j \right) \quad (4)$$

avec f est la fonction du modèle du réseau, $\tilde{x}_{\text{RF}(i)}^j$ est le champ récepteur of x_i^j avec une valeur dans le centre du champ choisi aléatoirement. Nous pouvons également le voir comme un patch entourant le pixel i , extrait de l'image d'entraînement x^j . La valeur de pixel cible correspondante, s_i^j , est extraite de l'image de vérité terrain s^j à la même position. Et par conséquent $f(\tilde{x}_{\text{RF}(i)}^j)$ est la prédition pour le target x_i^j .

Il y a deux limitations principales à cette méthode. L'une de ces limitations réside dans les images contenant de nombreuses irrégularités. Autrement dit, il est difficile de prédire la valeur d'un pixel en se basant uniquement sur son voisinage. Le problème découle du fait que nous omettons la valeur du pixel dans le champ réceptif pendant l'entraînement, rendant difficile d'avoir une bonne estimation de ce pixel. L'autre limitation est que N2V ne peut pas faire la distinction entre le signal et le bruit structuré qui viole l'hypothèse d'indépendance pixel par pixel.

3.3.3 Noise2Self

La méthode **Noise2Self (N2S)** repose sur l'idée que chaque pixel d'une image bruitée contient des informations sur le bruit présent dans les autres pixels. Cette idée est exploitée pour former un réseau de neurones à prédire la valeur de chaque pixel à partir des valeurs des autres pixels, sans utiliser sa propre valeur bruitée. Mathématiquement, cela peut être formulé comme un problème d'optimisation. Soit x une image bruitée et y une image propre. Le but est de trouver une fonction f , paramétrisée par un réseau de neurones, qui minimise l'erreur quadratique moyenne MSE entre $f(x)$ et y . Cependant, comme y n'est pas disponible, l'approche Noise2Self modifie ce problème en minimisant l'erreur entre $f(x)$ et x pour chaque pixel, mais en excluant la valeur du pixel cible lors de la prédiction. La fonction à optimisé est donc la suivante :

$$L(f) = \mathbf{E}\|f(x) - x\|^2 \quad (5)$$

Si x est un estimateur non biaisé de y , c'est à dire $\mathbf{E}[\mathbf{x}|y] = \mathbf{y}$, on peut montrer que :

$$L(f) = \mathbf{E}\|f(x) - x\|^2 = \mathbf{E}\|f(x) - y\|^2 + \mathbf{E}\|x - y\|^2 \quad (6)$$

Ce qui est équivalent à :

$L(f) = \text{ordinary supervised loss} + \text{variance of the loss}$

(7)

La décomposition 7 montre que la fonction de perte que Noise2Self minimise est équivalente à la perte supervisée ordinaire plus la variance du bruit. Cela signifie que Noise2Self est capable de gérer efficacement le bruit qui varie d'un pixel à l'autre, car il minimise non seulement l'erreur de prédiction, mais aussi la variabilité du bruit.

Techniquement, cela est réalisé en créant un masque pour chaque pixel qui exclut ce pixel mais inclut tous les autres pixels. Ce masque est appliqué à l'image bruitée pour créer une image masquée, qui est ensuite utilisée comme entrée pour le réseau de neurones. Le réseau est formé pour minimiser la différence entre sa prédiction et la valeur réelle du pixel, en utilisant une fonction de perte comme l'erreur quadratique moyenne. Une fois le réseau formé, il peut être utilisé pour restaurer l'ensemble de l'image. Pour chaque pixel, le réseau prédit sa valeur à partir des valeurs des autres pixels, produisant ainsi une image restaurée. Cette approche est capable de gérer efficacement le bruit qui varie d'un pixel à l'autre, car elle traite chaque pixel indépendamment. Il est important de noter que cette méthode suppose que le bruit est indépendant d'un pixel à l'autre, ce qui n'est pas toujours le cas dans les images réelles. Cependant, dans de nombreux cas, cette hypothèse est suffisamment précise pour permettre une restauration d'image efficace.

3.3.4 Noisier2Noise

La méthode **Noisier2Noise** consiste à générer un échantillon de bruit synthétique à partir de notre modèle statistique (ce qui suppose implicitement qu'on connaît le type du bruit), à l'ajouter à l'image déjà bruitée, puis à demander au réseau de prédire l'image bruitée d'origine à partir de cette image doublement bruitée. Intuitivement, le réseau ne pourra pas faire la distinction entre le bruit original et le bruit synthétique supplémentaire, et il pourra minimiser sa perte en prédisant que la moitié du bruit observé à chaque pixel était présente dans l'image source, tandis que l'autre moitié a été ajoutée de manière synthétique. Ainsi, le réseau apprendra à produire une image située à mi-chemin entre

l'entrée doublement bruitée et la véritable image nette inconnue. Nous pouvons ensuite obtenir une estimation de l'image nette par une simple manipulation mathématique.

Soit X l'image propre et $X + N$ l'image bruitée à laquelle on ajoute le bruit M du même type que le bruit N pour obtenir $X + N + M$, l'image doublement bruitée. Le but de l'apprentissage est donc d'essayer de prédire $X + N$: $\mathbf{E}[X + N|X + N + M]$. Mathématiquement, on peut résoudre cela comme suit :

On pose : $Y_1 = X + N$ et $Y_2 = X + N + M$

et sachant que : $\mathbf{E}[N|Y_2] = \mathbf{E}[N|Y_1]$ car N et M suivent le même loi, alors on a :

$$2\mathbf{E}[Y_1|Y_2] = \mathbf{E}[X|Y_2] + \mathbf{E}[N|Y_2] + \mathbf{E}[X|Y_2] + \mathbf{E}[M|Y_2] \quad (8)$$

$$= \mathbf{E}[X|Y_2] + \mathbf{E}[X + N + M|Y_2] \quad (9)$$

$$= \mathbf{E}[X|Y_2] + Y_2 \quad (10)$$

Finalement :

$$\boxed{\mathbf{E}[X|Y_2] = 2\mathbf{E}[Y_1|Y_2] - Y_2} \quad (11)$$

L'équation 11 indique que la prédiction de l'image propre $\mathbf{E}[X|Y_2]$ peut être obtenue en doublant la prédiction du réseau $\mathbf{E}[Y_1|Y_2]$ et en soustrayant la valeur de l'image bruitée plus le bruit supplémentaire M .

3.4 Solutions choisies

Les quatre méthodes ont été testées sur des données que j'ai générées, afin d'évaluer la performance de chacune d'elles, mais aussi pour ajuster les différents hyperparamètres des modèles, tels que le nombre d'itérations, la taille du réseau, la taille des lots, etc. Les méthodes que j'ai choisies sont **N2V** et **N2S**. Ces outils ont été sélectionnés car ils n'utilisent que des images bruitées pour l'entraînement, contrairement aux deux autres qui utilisent des images propres pour l'entraînement. En effet, pour les deux autres méthodes, c'est l'outil lui-même qui prépare les données bruitées avec lesquelles il va travailler, rendant ces outils inutilisables sauf si on a la possibilité de générer des simulations propres pour les données ILL.

4 Implémentation de la solution

4.1 Architecture haut niveau de la solution

Comme nous l'avons déjà mentionné, les données issues de l'instrument IN5 de l'ILL sont en quatre dimensions, mais à ce stade, nous ne savons débruiter que des données en deux dimensions, c'est-à-dire des images. Pour résoudre ce problème, nous allons fixer deux dimensions et réaliser des coupes selon les deux autres dimensions à l'aide de la bibliothèque Horace de Matlab. Ainsi, nous obtenons une série d'images après la conversion de ces coupes, et ce sont ces images qui seront utilisées pour le débruitage. Une première comparaison est effectuée entre les images bruitées et débruitées, puis ces images sont rassemblées et reconverties à leur état initial pour permettre une comparaison des deux fichiers en quatre dimensions. Cette comparaison est rendue possible grâce au module **hypervolume** de l'outil **Inviwo** [4], qui permet une visualisation plus fluide de ces données volumétriques. Le flux de travail qui sera donc suivi pour répondre à cette problématique est le suivant (Figure 3) :

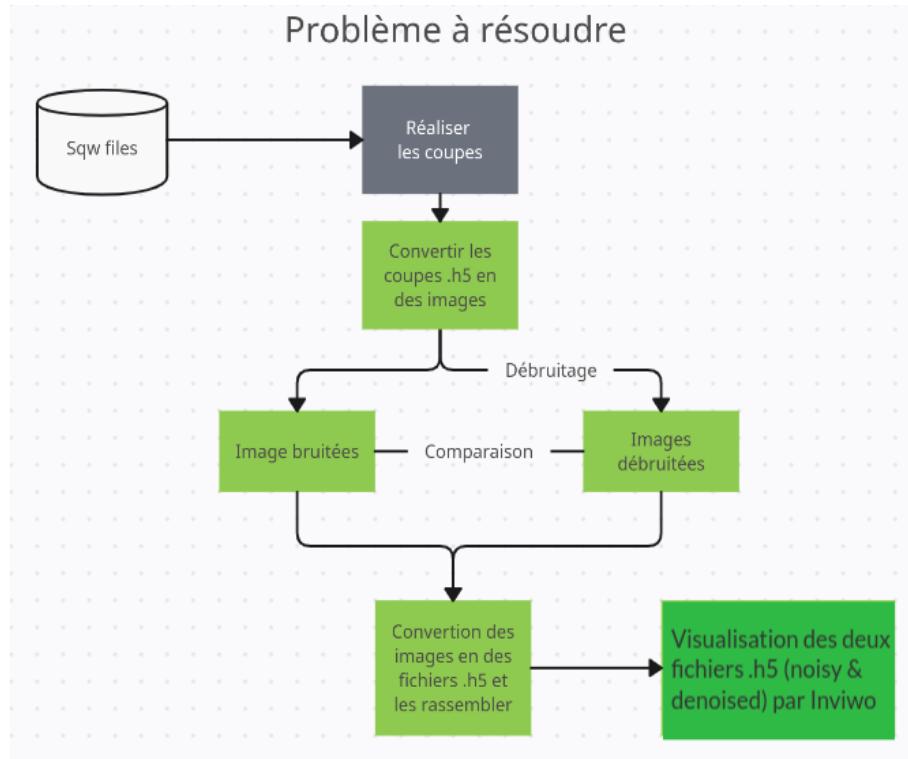


Figure 3: Le Workflow suivie pour résoudre la problématique

4.2 Détails techniques de l'implémentation

4.2.1 Les coupes par Horace Matlab

Les coupes sont principalement réalisées à l'aide de la fonction `cut_sqw` de la bibliothèque **Horace** de Matlab [5]. En effet, cette fonction est utilisée pour manipuler et extraire des données à partir de fichiers sqw. Elle nécessite un fichier sqw en entrée, ainsi que des axes de projection et des arguments de binning. Les axes de projection déterminent le système de coordonnées utilisé pour visualiser les données. On peut spécifier les vecteurs de projection `proj.u` et `proj.v` pour définir les deux premiers axes de visualisation, tandis que le troisième axe est calculé automatiquement comme le produit vectoriel des deux premiers. Le quatrième axe correspond toujours à l'énergie. Les arguments de binning, tels que `p1_bin`, `p2_bin` et `p3_bin`, permettent de découper ou d'intégrer les données le long des axes Q. On peut spécifier un nombre unique pour définir la largeur de bin, et les limites inférieure et supérieure seront déterminées en fonction de l'étendue des données. Alternativement, on peut spécifier un vecteur avec trois composantes pour définir les limites inférieure et supérieure de découpe, ainsi que la largeur de bin. De plus, il est possible de spécifier un vecteur avec deux composantes pour intégrer le signal entre les limites spécifiées. Les coupes réalisées sont des fichiers .h5 qui peuvent être visualisé par l'outil **Silx View**. L'opération de faire un grand nombre de coupes est très couteuse en terme de temps d'exécution, par exemple sur les deux datasets avec lesquelles on va travailler `bnfs_10K.sqw` et `TbTiO_1p5K.sqw`:

	Nombre de coupes	Dimensions	Temps d'exécution
bnfs10K.sqw	1407	$21 \times 67 \times 175 \times 366$	3-4h
TbTiO1p5K.sqw	550	$25 \times 22 \times 101 \times 101$	2-3h

Figure 4: Quelques informations sur les coupes effectuées sur les deux datasets.

4.2.2 Le processus de débruitage

Après la conversion des fichiers .h5 en des images, on passe à l'étape la plus importante de ce processus à savoir le débruitage de ces images. Durant cette étape on essaye de trouver les bons paramètres du réseau pour qu'on obtient les meilleurs résultats possibles (comment on définit les meilleurs résultats sera expliqué dans la partie consacrée aux métriques d'évaluation de l'efficacité de la solution). Tout d'abord on prépare les données d'entraînement qui servent à faire entraîner le modèle et les données de validation aussi qui sont utilisées pour évaluer la performance du réseau au fur et à mesure de l'entraînement et de faire quelques changements sur le pas d'entraînement si il est nécessaire. Une première décision à prendre c'est de diviser donc ce dataset. En effet, lorsqu'il s'agit de diviser un dataset en ensembles d'entraînement et de validation, il n'y a pas de règle stricte, car cela dépend du contexte et des besoins spécifiques du projet. En général, une division courante est de 80% pour cent des données pour l'entraînement et 20% pour la validation. Cependant, si le dataset est petit, il peut être plus judicieux d'opter pour une validation croisée, où les données sont divisées en k-folds, permettant d'effectuer k entraînements et évaluations pour obtenir une estimation plus fiable des performances du modèle. Pour les datasets plus grands, une proportion de validation plus petite peut suffire à évaluer le modèle de manière adéquate. Par exemple dans le cas de la méthode **N2V** généralement un dataset plus ou moins grande puisque on travaille avec des patches d'images, et donc on fait plus de 90% pour l'entraînement et le reste pour la validation.

En plus, il existe d'autres paramètres importants à prendre en compte pour obtenir de meilleurs résultats. Pour **N2V** les paramètres clés comprennent le nombre d'epochs, le nombre et la taille des batches, le nombre de pas d'entraînement par epoch, ainsi que le pourcentage des pixels à masquer :

Le nombre d'epochs détermine le nombre de fois que l'algorithme parcourt l'ensemble des données lors de l'entraînement. Un nombre plus élevé d'epochs permet généralement au modèle d'apprendre des caractéristiques plus complexes et d'améliorer sa performance. Cependant, un nombre excessivement élevé d'epochs peut entraîner un surapprentissage, où le modèle s'adapte trop spécifiquement aux données d'entraînement et ne généralise pas bien sur de nouvelles données.

Le nombre et la taille des batches font référence à la division de l'ensemble de données en sous-ensembles plus petits, appelés batches. Un batch est utilisé pour calculer les gradients et mettre à jour les poids du modèle. Un nombre plus élevé de batches peut accélérer l'entraînement, mais cela nécessite également plus de mémoire.

Le nombre de pas d'entraînement par epoch indique combien de fois le modèle est mis à jour à chaque epoch. Un nombre plus élevé de pas d'entraînement peut permettre une convergence plus rapide, mais cela peut également augmenter le risque de surajustement (Overfitting).

Le pourcentage des pixels à masquer est une étape cruciale de Noise2Void. Il

	Taille des batchs	Nbr de pas d'entraînement	Pourcentage de pixels
bnfs10k.sqw	128	860	0.198
TbTiO1p5K.sqw	128	500	0.198

Figure 5: Les paramètres choisies pour l'entraînement sur les deux datatsets, avec un nombre d'epochs égale à 200

s'agit de masquer une partie des pixels de l'image d'entrée afin de forcer le modèle à apprendre à reconstruire les informations manquantes. Un pourcentage plus élevé de pixels masqués peut améliorer la capacité du modèle à généraliser, mais cela peut également entraîner une perte d'informations.

Quant à la méthode N2S, Les paramètres clés, en plus du nombre d'epochs, qui jouent un rôle crucial dans la performance de cette méthode sont :

La taille du réseau fait référence à l'architecture du réseau de neurones utilisé dans Noise2Self. Un réseau plus grand avec plus de couches et de neurones peut potentiellement capturer des informations plus complexes et améliorer la performance du modèle. Cependant, un réseau trop grand peut également entraîner un surapprentissage et nécessiter plus de ressources de calcul.

La taille du masque est un paramètre important dans Noise2Self. Il détermine la taille des régions masquées lors de l'entraînement du modèle. Un masque plus grand peut permettre au modèle d'apprendre à reconstruire des informations manquantes sur de plus grandes régions de l'image, tandis qu'un masque plus petit peut se concentrer sur des détails plus fins. Le choix de la taille du masque dépend de la nature du bruit et des caractéristiques de l'image.

D'après les tests et les analyses qui ont été réalisés, il a été observé que des valeurs de masque comprises entre 2 et 4, ainsi qu'une taille de réseau entre 12 et 18, donnent les meilleurs résultats. Ces valeurs ont été identifiées comme étant les plus appropriées pour obtenir une performance optimale avec la méthode Noise2Self. Cependant, il est important de noter que ces valeurs peuvent varier en fonction du jeu de données spécifique et du problème à résoudre. Des expérimentations supplémentaires sont nécessaires pour affiner ces paramètres en fonction du contexte d'application.

Après un mois d'expérimentation, j'ai identifié les paramètres que je considère comme optimaux pour les deux ensembles de données. Il est important de noter que le temps d'entraînement dans notre cas varie entre 3 et 6 heures, ce qui explique la durée de cette étape. Une fois l'entraînement terminé, nous utilisons un script simple pour appliquer le processus de débruitage à toutes nos images, ce qui nous permet d'obtenir nos images débruitées.

4.2.3 Préparation des données pour visualisation

Pour faciliter la visualisation des images débruitées et bruitées, nous avons tout d'abord effectué une transformation de l'espace de couleur de niveaux de gris à l'espace de couleur "jet". Cette transformation permet de mieux distinguer les différentes intensités de pixels et de mettre en évidence les détails importants dans les images.

Ensuite on reconvertie les images en des fichies .h5 pour ensuite les rassembler séparément, c'est à dire les images bruitées tous en une seule fichier .h5 et les images débruitées tous en un autre ficheir .h5. Cette étape de rassemblage des images est fait par le script que j'ai fait appellé `merge_h5_files.py`. Ce script est plus ou moins compliquée puisque il prend 10 arguments en entrée stdandard. Les 10 paramètres sont les suivantes :

```

1 argv[1] = input_directory
2 argv[2] = output_h5_file
3 argv[3] = Number of cuts with respect to first dimension
4 argv[4] = Number of cuts with respect to second dimension
5 argv[5] = cutting step along the first dimension
6 argv[6] = cutting step along the second dimension
7 argv[7] = Min with respect to the first dimension
8 argv[8] = Min with respect to the second dimension
9 argv[9] = the third dimension
10 argv[10] = the fourth dimension

```

La plupart des paramètres sont simples à obtenir, certains nécessitent des calculs supplémentaires. Dans l'exemple du jeu de données `bnfs_10K.sqw`, nous avons réalisé des coupes en fonction de l'énergie et du troisième axe. Pour calculer le nombre de coupes réalisées par rapport au premier et au deuxième dimension, nous utilisons les formules suivantes :

$$\begin{aligned} \text{argv}[3] &= \text{round}((E_{\max} - E_{\min})/\text{CuttingStep}_E)) \\ \text{argv}[4] &= \text{round}((w_{\max} - w_{\min})/\text{CuttingStep}_w)) \end{aligned}$$

avec CuttingStep_E et CuttingStep_w sont les pas des coupes par rapport à l'énergie et le troisième axe respectivement.

Une fois que tous les arguments sont connus, nous pouvons procéder à la réalisation de la fusion des fichiers .h5. Cette opération sera effectuée pour les deux sous-ensembles, c'est-à-dire les fichiers .h5 correspondant aux images bruitées et les fichiers .h5 correspondant aux images non bruitées. Pour l'exemple `bnfs_10K.sqw`, la commande suivante sera utilisée :

```

1 python3 merge_h5_files.py resultats/noisy_h5/ resultats/
    noisy_merged.h5 21 67 0.24 0.05 -0.9825 -3.0342 175 366

```

La commande créera un fichier .h5 résultant de la fusion de tous les fichiers .h5 bruités. Pour vérifier rapidement l'intégrité des données, on peut utiliser l'outil **Slix View** [6], qui permet une visualisation initiale des fichiers .h5 et offre une première indication sur d'éventuels problèmes. On peut voir déjà un exemple de coupe bruitée à l'aide de cette outil sur la figure 6 :

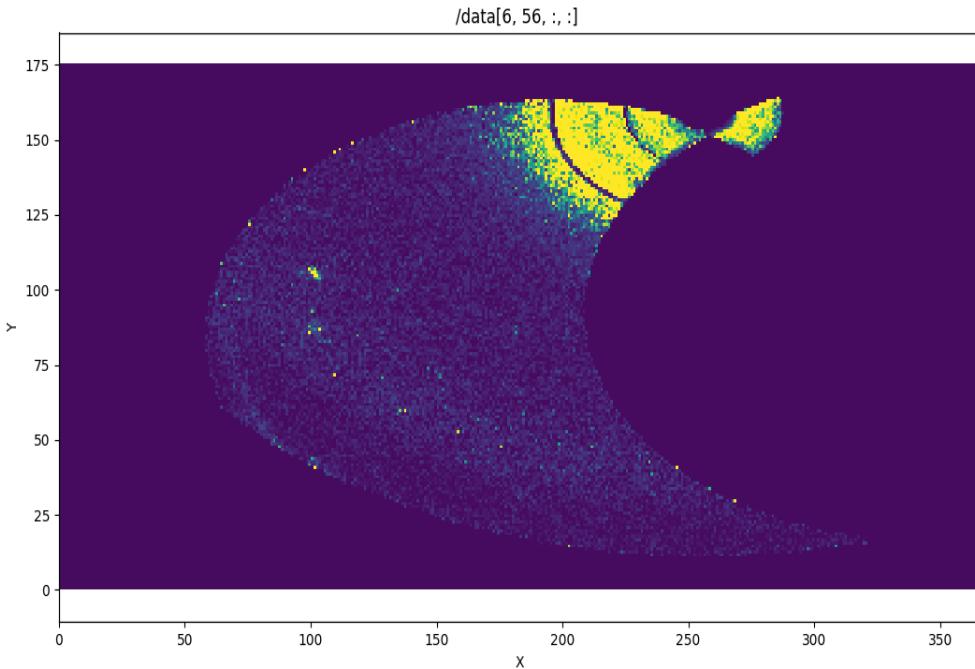


Figure 6: Coupe d'indice 6 par rapport à l'énergie et 56 par rapport au troisième axe sur Silx View.

4.3 Test et évaluation

Le PSNR, ou Peak Signal-to-Noise Ratio, est une mesure couramment utilisée pour évaluer la qualité de débruitage d'une image ou d'une vidéo. Il représente le rapport entre la puissance maximale du signal (l'image originale non bruitée) et la puissance du bruit qui a été ajouté à l'image débruitée :

$$PSNR = 10 \cdot \log_{10} \left(\frac{\max^2}{MSE} \right) \quad (12)$$

où :

- max représente la valeur maximale possible du pixel (par exemple, pour une image RVB de 8 bits, max serait égal à 255).
- MSE correspond à l'erreur quadratique moyenne entre l'image originale et l'image débruitée.

Plus le PSNR est élevé, plus l'image débruitée est considérée comme de haute qualité, car cela signifie que le bruit a été réduit de manière significative. Cependant, le PSNR présente des limites dans certaines situations, notamment lorsqu'il s'agit d'évaluer des propriétés spécifiques liées à la physique des neutrons dans les images. Le PSNR mesure la qualité globale de l'image en termes de bruit perceptible, mais il ne peut pas détecter les variations subtiles ou les détails importants liés à des phénomènes physiques spécifiques.

C'est pourquoi, dans le cas où l'on s'intéresse particulièrement aux caractéristiques liées aux neutrons dans les images, il est important de prendre en compte d'autres facteurs, comme la perception visuelle. La perception visuelle implique l'évaluation subjective de l'image par des humains, prenant en compte des aspects tels que la clarté des détails, la

fidélité des couleurs et la préservation des informations cruciales. Cette approche permet de mieux comprendre comment les détails importants sont affectés par le débruitage, en se rapprochant davantage de ce que l'œil humain perçoit.

De plus, dans certains scénarios tels que celui à l'ILL, où seules les données bruitées sont disponibles sans accès aux images propres, l'utilisation du PSNR devient impossible. Sans cette référence d'image non bruitée, on ne peut pas calculer le PSNR de manière significative, car il nécessite une comparaison directe entre l'image débruitée et l'image d'origine pour évaluer l'efficacité du débruitage. Afin d'évaluer donc la performance du débruitage sur les données de l'ILL, nous opterons pour l'utilisation de la perception visuelle comme métrique essentielle. Dans cette démarche, les experts du domaine joueront un rôle clé dans l'évaluation. Leur expertise et leur connaissance approfondie des propriétés physiques des neutrons dans les images leur permettront d'analyser de manière précise la qualité du débruitage. Ils seront en mesure de déterminer si les détails critiques liés aux neutrons ont été préservés après le traitement de débruitage et s'ils sont encore visibles et identifiables dans les données débruitées.

Cette approche globale nous permettra donc de garantir que le débruitage est adapté aux besoins spécifiques de l'expérimentation neutronique et qu'il n'affecte pas de manière indésirable les informations essentielles contenues dans les données de l'ILL.

5 Résultats obtenus

5.1 Comparaison des résultats des 4 méthodes

Voici les résultats obtenus sur quelques exemples des données de l'ILL :

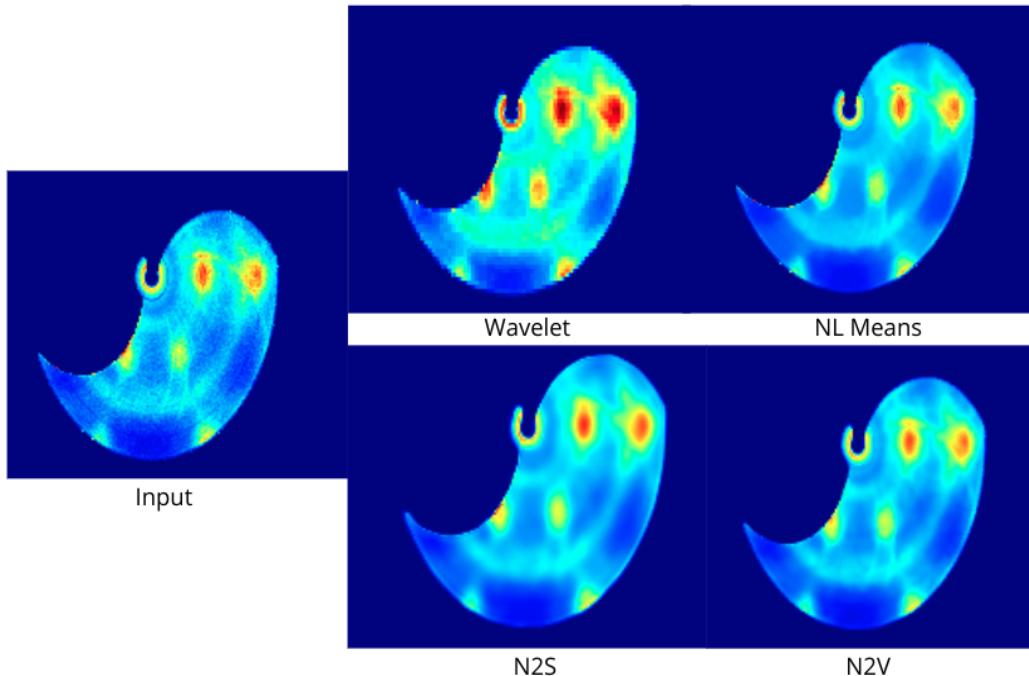


Figure 7: Résultats des 4 méthodes de débruitage sur une image du dataset TbTi01p5K1

Sur la figure 7, on remarque très clairement que la méthode Wavelet introduit quelques artefacts et provoque des pertes au niveau de certains détails importants. Le NL Means

est relativement similaire sur cet exemple, tout comme les deux autres méthodes de Deep Learning, où l'on peut observer un certain niveau de débruitage réalisé. Cependant, il est primordial de garder à l'esprit que notre objectif principal est de préserver les propriétés physiques des neutrons dans les images. C'est la raison pour laquelle nous allons maintenant zoomer sur la grande tache rouge afin d'observer plus attentivement les différences et de comparer les performances de ces quatre méthodes :

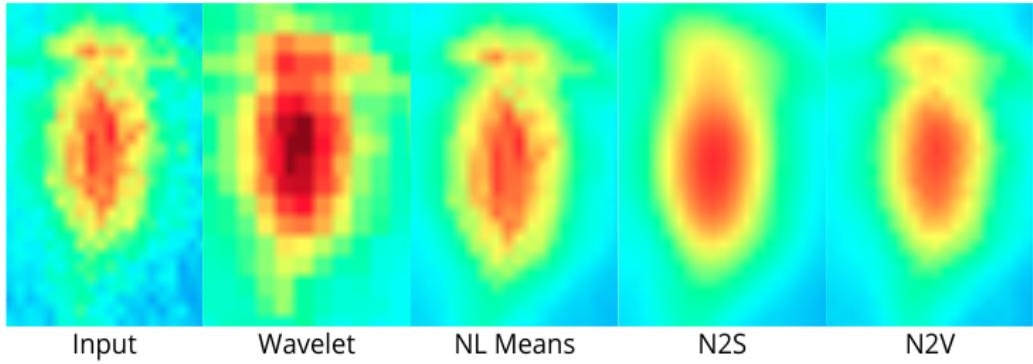


Figure 8: Le débruitage réalisé par les quatre méthodes au niveau de la tache rouge.

On constate dans la figure 8 l'apparition de plusieurs blocs de pixels de la même couleur avec la méthode Wavelet, ce qui la rend moins appropriée pour ce type d'application de débruitage. La méthode qui réalise le meilleur débruitage (visuellement) est la méthode N2S, mais le problème avec cette méthode est qu'elle a combiné la petite tache rouge en haut avec la grande tache au milieu, contrairement aux méthodes N2V et NL Means. Cette dernière, bien qu'elle conserve les deux taches séparées, ne réalise pas un très bon débruitage au niveau de la tache rouge, tandis que N2V le fait très bien. On peut en déduire que N2V donne le meilleur résultat sur cet exemple.

Voyons maintenant un autre exemple sur l'autre dataset `bnfs_10k.sqw` :

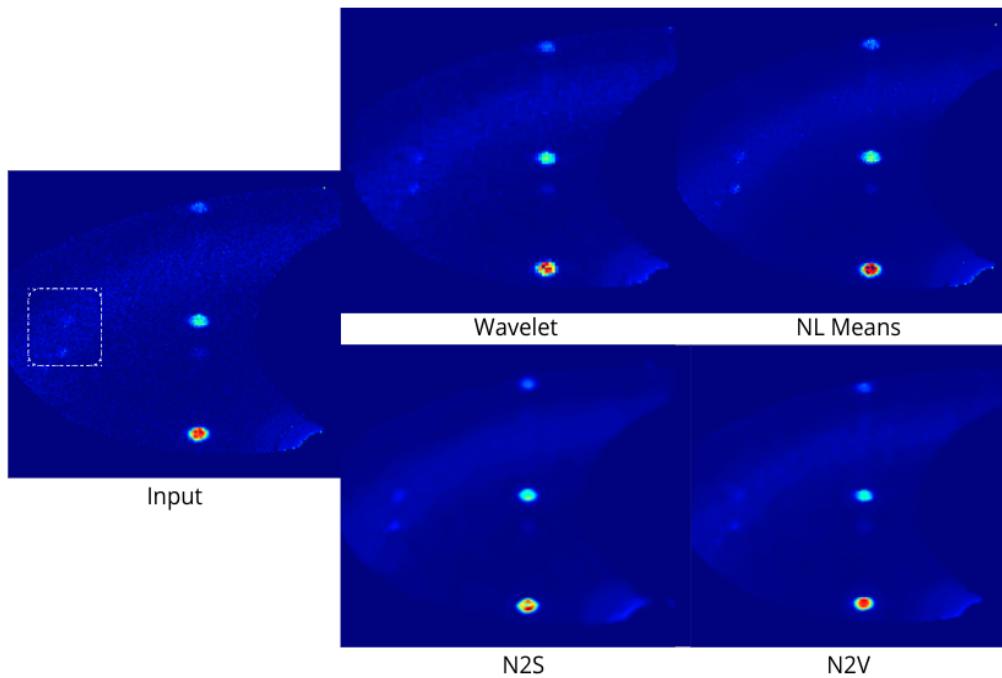


Figure 9: Le débruitage réalisé par les quatre méthodes sur une image du dataset `bnfs10K`

On peut remarquer une fois de plus que généralement le fond de la forme géométrique dans l'image est plus ou moins bien débruité par toutes les méthodes. C'est pourquoi il est nécessaire de se concentrer sur des points précis afin de pouvoir distinguer les performances de chacune des méthodes. Nous allons donc zoomer une fois de plus sur le point rouge en bas :

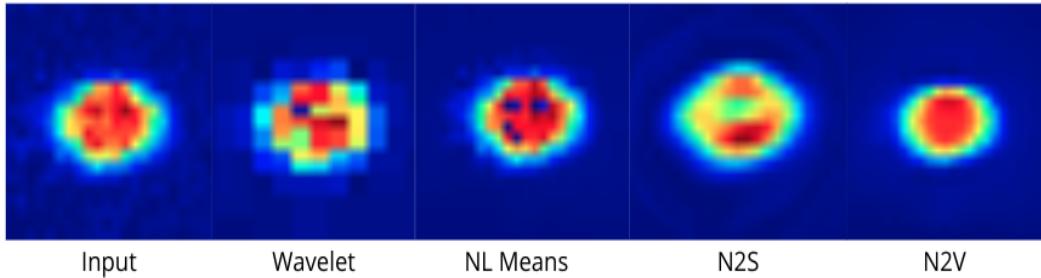


Figure 10: Le débruitage réalisé par les quatre méthodes au niveau de la tache rouge.

Une fois de plus, on peut constater que les méthodes de débruitage classiques ne donnent pas des résultats satisfaisants. En revanche, la méthode N2V offre le meilleur résultat visuel, qui reste à confirmer par les experts du domaine. Bien que la méthode N2S donne un résultat étrange au niveau de la tache rouge, elle permet de faire apparaître les deux points cachés derrière le bruit (voir le carré pointé en blanc dans la Figure 9) mieux que N2V.

Le choix entre les méthodes N2S et N2V dépend des propriétés physiques que l'on souhaite préserver dans les données débruitées. N2S est efficace pour révéler des détails cachés derrière le bruit, tandis que N2V offre des résultats visuellement convaincants. Tester les deux méthodes permet de déterminer celle qui correspond le mieux aux résultats souhaités pour l'application de débruitage sur les données de l'ILL.

Une fois cette comparaison effectuée, il est également intéressant d'examiner les données dans leur état d'origine, c'est-à-dire sous forme volumétrique, comme nous l'avons discuté précédemment, en utilisant l'outil Inviwo. Des exemples ont été insérés dans la section Annexes à la fin du document.

5.2 Regard critique sur les résultats obtenus

Les résultats obtenus lors de mon stage sont satisfaisants, mais cela ne signifie pas que nous ne pouvons pas obtenir de meilleurs résultats. Malheureusement, la durée limitée de mon stage ne m'a pas permis de réaliser une fine-tuning exhaustive sur tous les paramètres du modèle, car cela nécessiterait des ressources importantes. De plus, la quantité de données disponibles à l'ILL est limitée, ce qui rend l'outil utilisable uniquement pour la dataset concernée.

D'autre part, nous n'avons pas pu évaluer la performance d'autres méthodes, telles que Noise2Noise et Noisier2Noise, car nous n'avions pas les données appropriées pour l'entraînement. Une piste envisageable aurait pu être de générer des images propres correspondant aux données de l'ILL, mais cela soulève une autre problématique liée à la simulation, qui nécessiterait potentiellement l'utilisation de lois et d'équations physiques pour les créer.

6 Bilan personnel

Au cours de ce projet, j'ai eu ma première expérience dans un environnement professionnel au sein d'une entreprise très diversifiée, comptant plus de 32 nationalités. Cela m'a permis de développer de nombreuses compétences interpersonnelles, telles que la ponctualité, la gestion du temps et une communication efficace avec les différents membres de l'équipe.

La visite au réacteur de l'ILL a été très intéressante et enrichissante. J'ai pu découvrir le fonctionnement des différents instruments et appareils utilisés dans les recherches nucléaires et neutroniques. De plus, j'ai appris de nouvelles informations sur l'instrument IN5, dont les données ont constitué la base principale de mon travail.

Tout au long du projet, j'ai également rencontré pas mal de difficultés. Notamment, le temps d'attente pour obtenir les résultats des expériences a constitué un défi important, en particulier lors des longs entraînements des réseaux de neurones. Pour surmonter cette difficulté, j'ai dû développer des compétences en organisation et en planification, et j'ai appris à gérer efficacement mon temps en effectuant d'autres tâches en parallèle pour maximiser ma productivité.

Ce que j'ai vraiment apprécié, c'est le côté pratique de ce projet. J'ai pu mettre en œuvre les connaissances théoriques que j'ai acquises en utilisant des outils et des bibliothèques de deep learning comme TensorFlow et PyTorch. J'ai manipulé des modèles, réalisé des entraînements, ajusté les hyperparamètres et évalué les performances des réseaux de neurones. Tout cela a renforcé ma confiance en mes compétences et m'a donné envie de relever des défis encore plus complexes dans ce domaine.

Au-delà de l'aspect technique, cette expérience a également été une véritable source de motivation et de détermination pour mon avenir professionnel. Elle a renforcé mon désir de poursuivre mes études dans le domaine du machine learning, et j'envisage désormais de me spécialiser davantage dans cette discipline en suivant des formations avancées et en réalisant des projets de recherche innovants.

En résumé, ce projet m'a offert une opportunité unique d'apprentissage et de croissance personnelle. J'ai découvert le monde professionnel, renforcé mes compétences interpersonnelles et développé une expertise concrète en machine et deep learning. Ces apprentissages précieux continueront de me guider dans mes futures études et carrière, où je suis déterminé à exploiter pleinement mon intérêt pour le machine learning pour contribuer à des avancées significatives dans le domaine scientifique et technologique.

7 Conclusoion

Les résultats obtenus constituent une base solide pour les futures améliorations et recherches dans le domaine du débruitage des données de l'ILL. La compréhension des contraintes liées à la disponibilité des données et aux ressources de calcul nous incite à adopter des approches créatives afin de maximiser l'efficacité du débruitage, tout en tenant compte

des spécificités de l’expérimentation neutronique. Cette approche d’investigation continue nous permettra potentiellement d’élargir l’applicabilité des méthodes de débruitage et de parvenir à des résultats encore plus performants, répondant ainsi aux besoins spécifiques de l’ILL et d’autres expérimentations similaires.

En poursuivant nos efforts dans ce domaine, nous pourrons exploiter pleinement les possibilités offertes par le Deep Learning pour optimiser les méthodes de débruitage et les adapter à des ensembles de données à 4 dimensions. Ces résultats encourageants justifient pleinement l’investissement en temps et en ressources consenti lors de cette étude. Ils ouvrent la voie à de nouvelles opportunités pour améliorer la qualité des données neutroniques et faciliter leur analyse, ce qui pourrait avoir un impact significatif sur les futures recherches dans le domaine scientifique et technologique.

FIN

8 Références

References

- [1] ILL. *High-flux Reactor*. URL: www.ill.eu/reactor-and-safety/high-flux-reactor.
- [2] ILL. *Instruments list*. URL: www.ill.eu/users/instruments/instruments-list.
- [3] ILL. *Remote Instrument Control*. URL: <https://www.ill.eu/users/support-labs-infrastructure/instrument-control-service/remote-instrument-control/>.
- [4] *Inviwo Documentation*. URL: https://inviwo.org/manual_index.html.
- [5] PACE-Neutrons. *Horace Documentation v3.6.3*. URL: <https://pace-neutrons.github.io/Horace/v3.6.3/>.
- [6] *Silx Documentation - Applications - View*. URL: <http://www.silx.org/doc/silx/0.7.0/applications/view.html>.

9 Annexes

On présente ici quelques exemples de visualisation des données à l'aide de l'outil Inviwo.

9.1 Le dataset bnfs10K

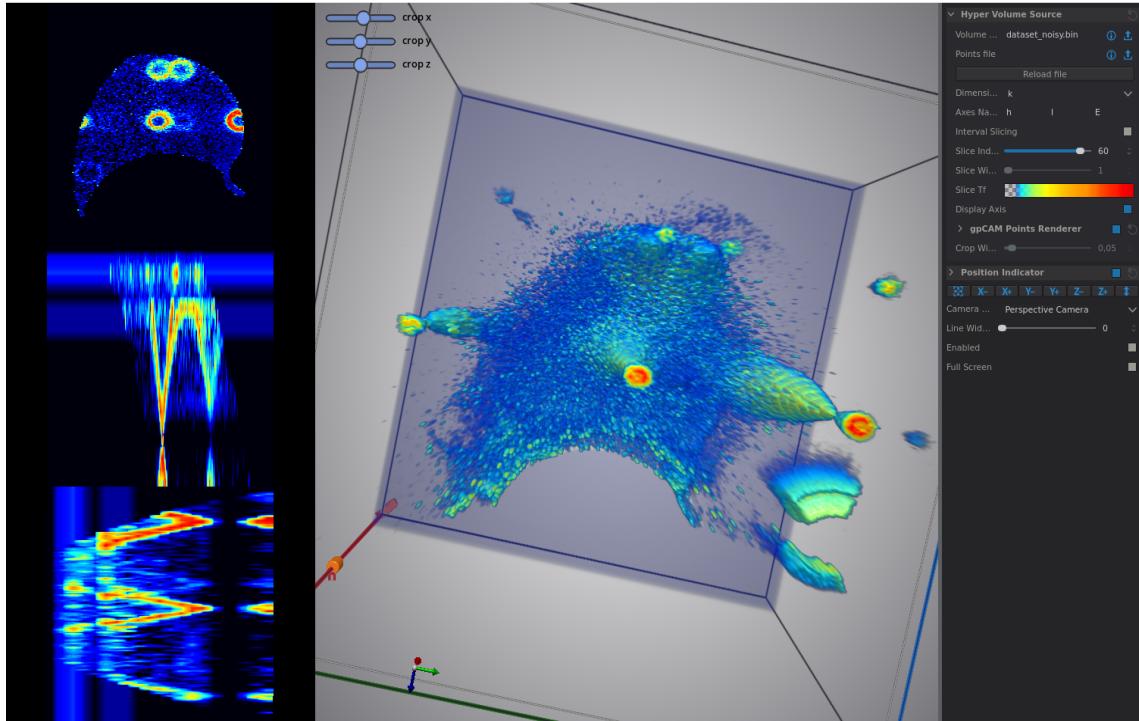


Figure 11: Au milieu, une coupe 3D bruitée du dataset **bnfs10K**, à gauche, les coupes 2D usuelles selon différents axes.

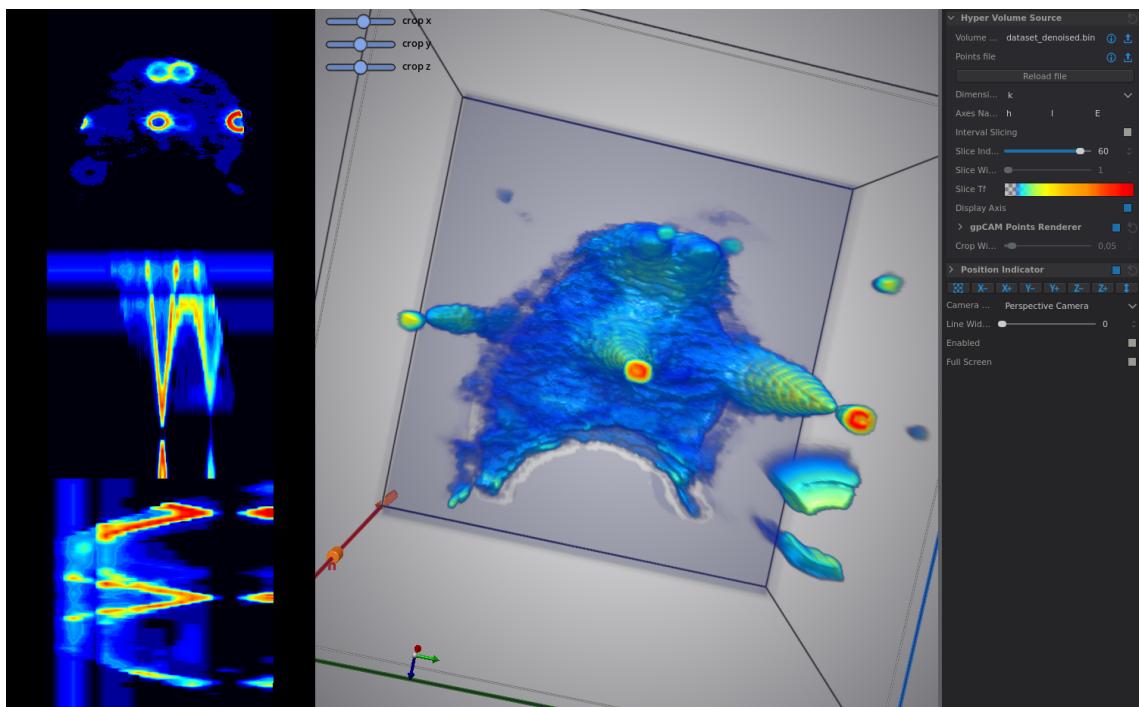


Figure 12: La même coupe dans la figure 11 débruitée.

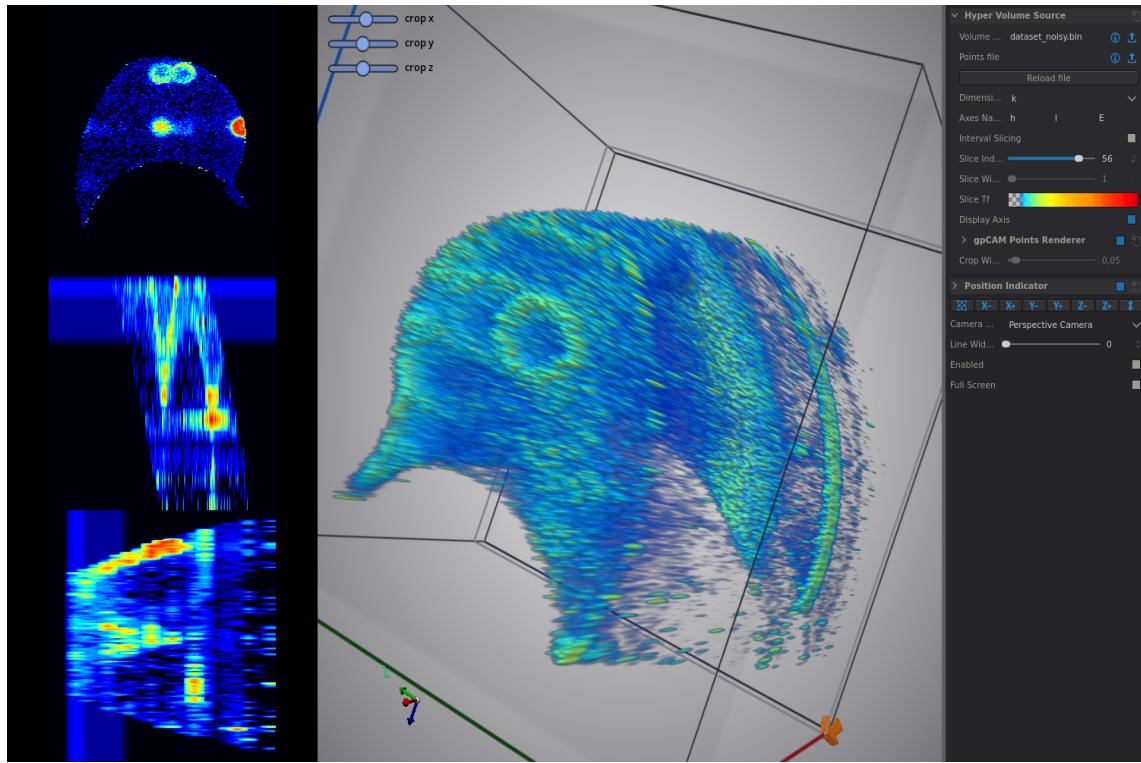


Figure 13: Au milieu, une autre coupe 3D bruitée du dataset **bnfs10K** , à gauche, les coupes 2D usuelles selon différents axes.

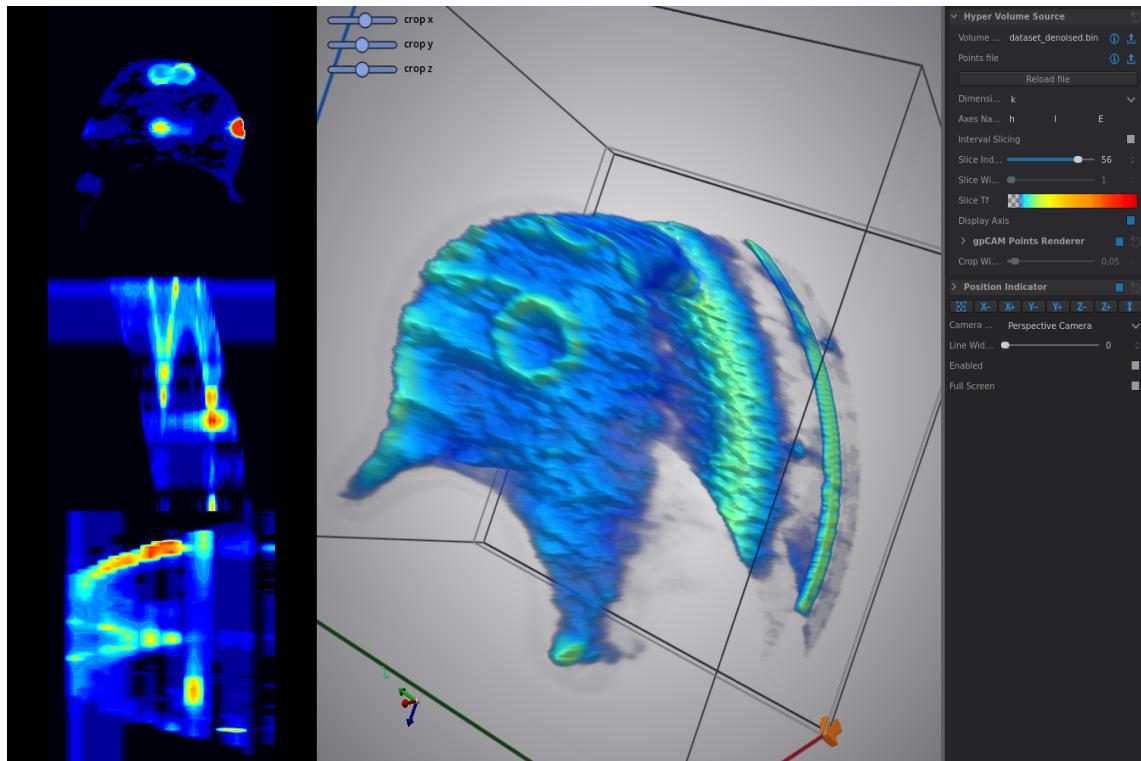


Figure 14: La même coupe dans la figure 13 débruitée.

9.2 Le dataset TbTiO1p5K :

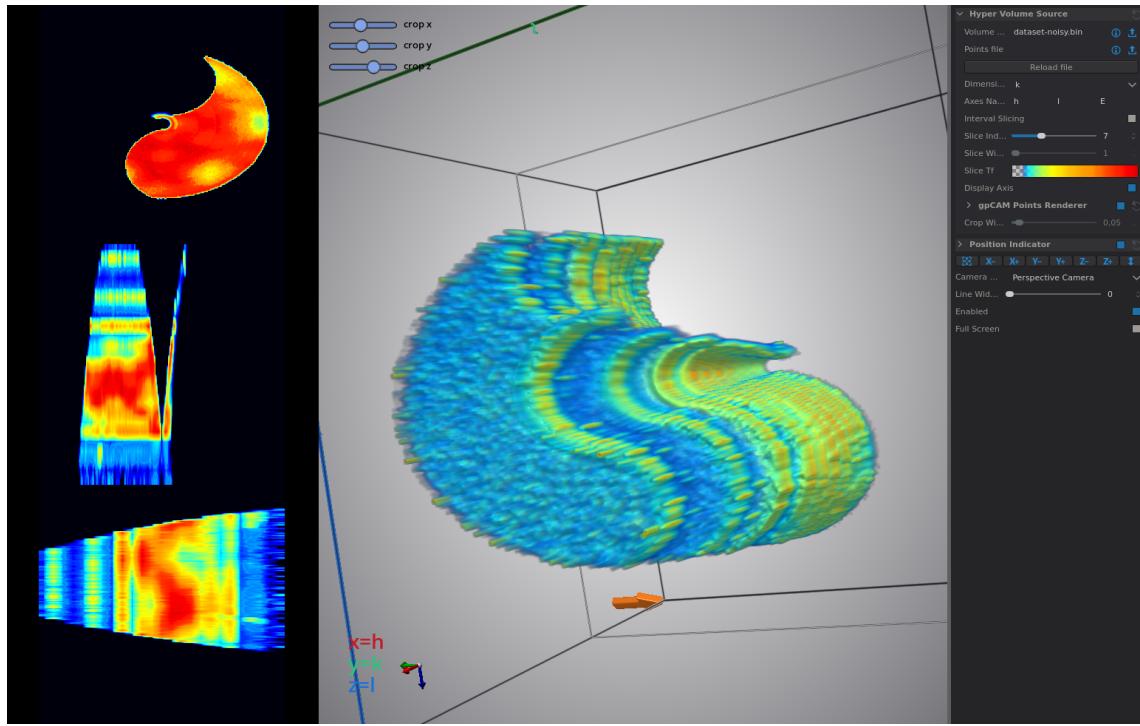


Figure 15: Au milieu, une coupe 3D bruitée du dataset **TbTiO1p5K** , à gauche, les coupes 2D usuelles selon différents axes.

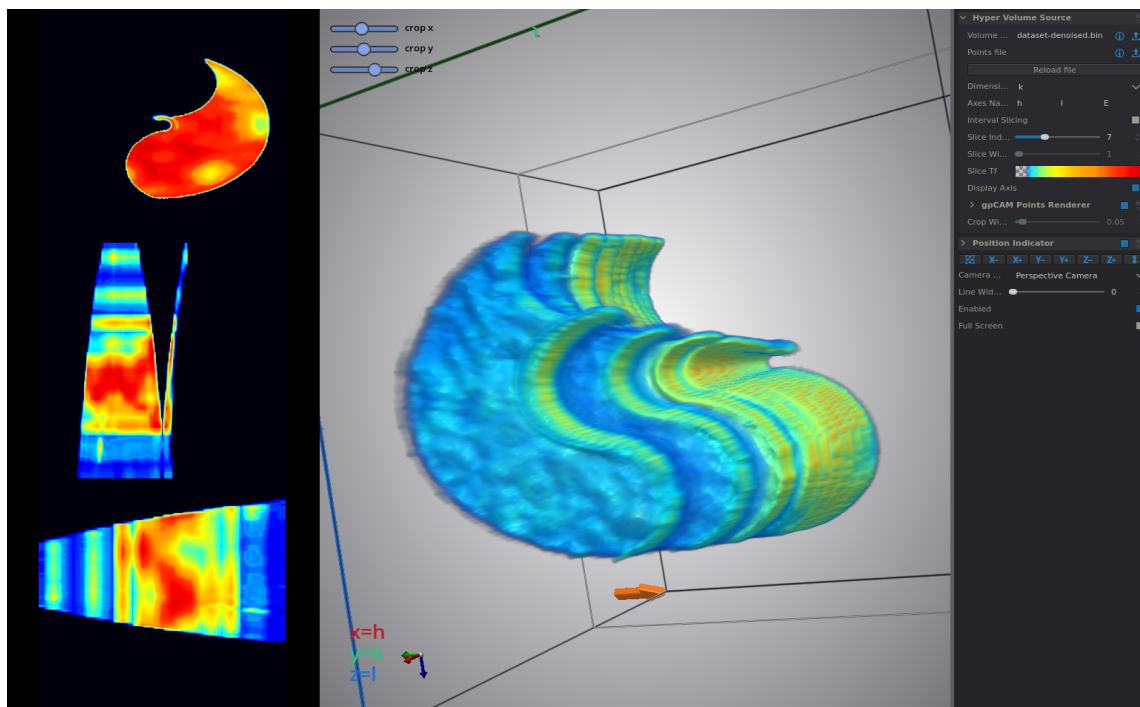


Figure 16: La même coupe dans la figure 15 débruitée.

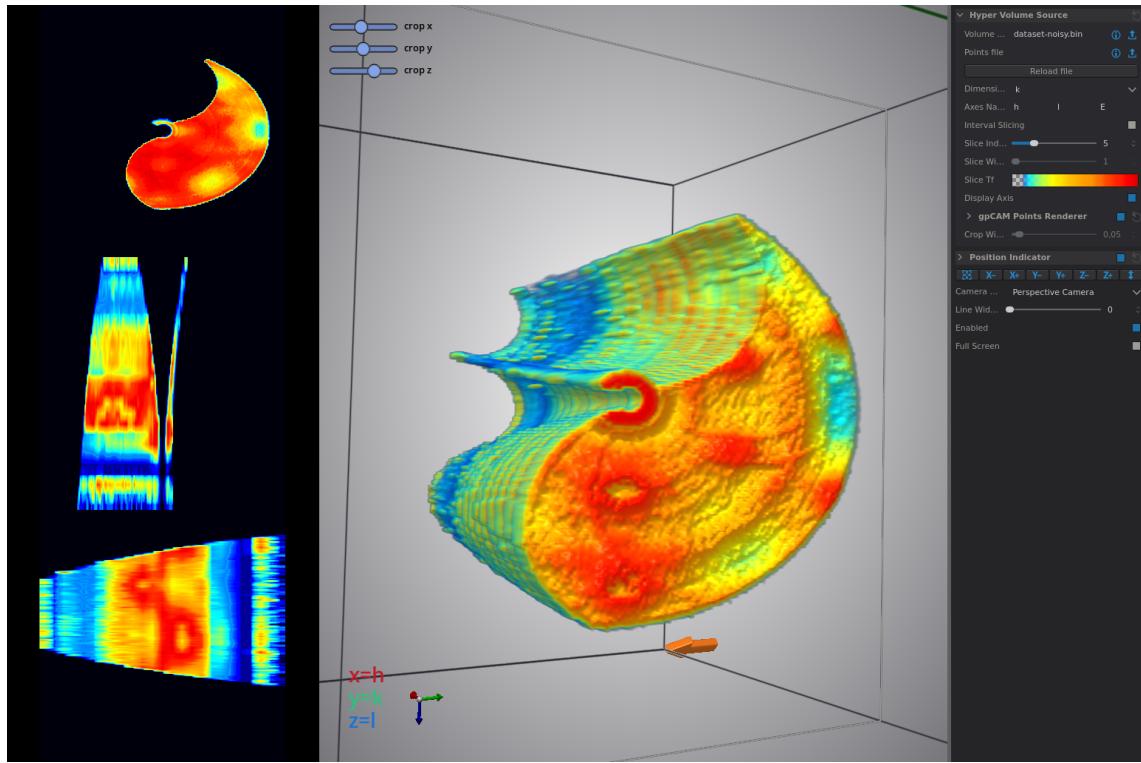


Figure 17: Au milieu, une autre coupe 3D bruitée du dataset **TbTiO1p5K**, à gauche, les coupes 2D usuelles selon différents axes.

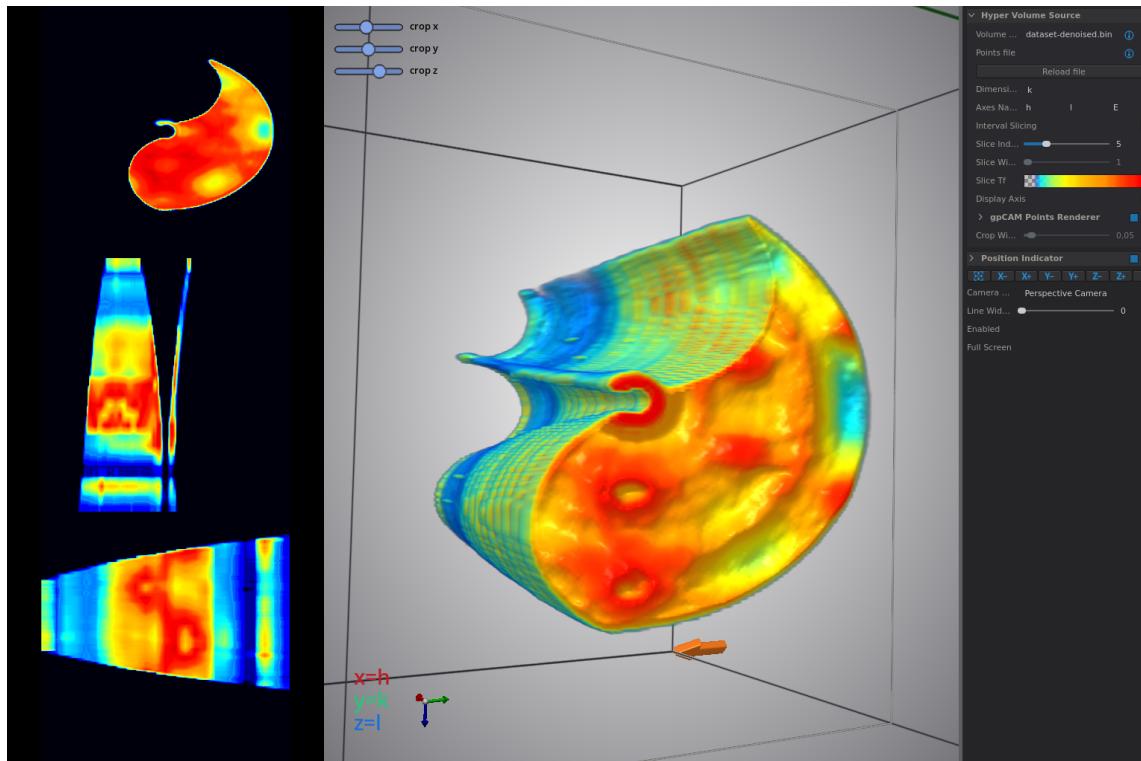


Figure 18: La même coupe dans la figure 17 débruitée.