



MASTER INFORMATIQUE

# Débruitage d'IRM pour la Sténose du Canal Lombaire

*Hocine LOUBAR et Athena CHRYSANTHOU*

*Encadrant : Clara BRÉMOND*

*Semestre de printemps 2025*

# Table des matières

<b>1 Résumé</b>	<b>3</b>
<b>2 Introduction</b>	<b>3</b>
<b>3 Etat de l'art sur le débruitage</b>	<b>4</b>
3.1 Les méthodes classiques de débruitage . . . . .	4
3.2 Suport Vector Machines . . . . .	6
3.2.1 Principes théoriques du SVM . . . . .	6
3.2.2 SVM et débruitage d'images . . . . .	8
3.2.3 Application au projet . . . . .	8
3.3 Algorithmes génératifs . . . . .	9
3.3.1 DaNet . . . . .	9
<b>4 Planification des tâches</b>	<b>11</b>
<b>5 Méthodes</b>	<b>12</b>
5.1 Base de donnée d'IRM de sténose du canal lombaire . . . . .	12
5.1.1 Format des données . . . . .	12
5.1.2 Modalités d'imagerie : T1 et T2 . . . . .	12
5.2 Vérité terrain . . . . .	13
5.2.1 Choix de vérité terrain . . . . .	13
5.2.2 Impact sur l'apprentissage . . . . .	13
5.3 Pré-traitement des données . . . . .	14
5.4 Ajout du bruit . . . . .	14
5.4.1 Distribution Rician . . . . .	14
5.4.2 Distribution Gaussienne . . . . .	14
5.5 Métriques d'évaluation . . . . .	17
5.6 Comparaison de modèles . . . . .	19
5.6.1 DANET . . . . .	19
5.6.2 SVR . . . . .	20
<b>6 Résultats</b>	<b>20</b>
6.1 Résultats Qualitatifs . . . . .	20
6.1.1 Débruitage par SVR . . . . .	20
6.1.2 Débruitage par GAN . . . . .	25
6.2 Résultats Quantitatifs . . . . .	27
6.2.1 SVR . . . . .	27
6.2.2 DANET . . . . .	29
<b>7 Discussion</b>	<b>31</b>

<b>8</b>	<b>Lexique</b>	<b>32</b>
<b>9</b>	<b>Anexe</b>	<b>32</b>

# 1 Résumé

L'imagerie par résonance magnétique est un outil essentiel dans le diagnostic de pathologies telles que la sténose du canal lombaire. Une bonne qualité d'image est indispensable pour un diagnostic fiable. Cependant, les IRMs sont souvent affectées par divers types de bruit, nuisant à leur lisibilité.

Ce projet vise à améliorer la qualité des IRMs en comparant deux approches de débruitage : une méthode classique de régression par Support Vector Regressor (SVR), et une méthode avancée de génération d'images via des GANs (Generative Adversarial Networks).

Les résultats montrent que le SVR fournit de bonnes performances quantitatives et une meilleure stabilité, notamment pour les petits patchs (3x3). En revanche, le GAN, bien que prometteur théoriquement, s'est révélé difficile à entraîner et peu efficace dans notre contexte.

Ce travail ouvre la voie à des approches hybrides combinant précision locale (SVR) et capacité de généralisation (GAN), dans l'objectif de développer un système de débruitage robuste pour l'imagerie médicale.

## 2 Introduction

La sténose du canal rachidien désigne le rétrécissement du canal de la colonne vertébrale. Un rétrécissement du canal rachidien provoque une pression douloureuse sur le système nerveux central et tous les nerfs qui y circulent. Outre les fortes douleurs, la sténose du canal rachidien touche principalement les personnes âgées et fait partie des maladies dégénératives typiques [12].

L'imagerie par résonance magnétique (IRM) est l'une des méthodes d'examen radio les plus avancées aujourd'hui. La création d'images multi-dimensionnelles des organes humains est possible car l'IRM utilise de puissants aimants et des ondes radio générées par ordinateur. Ce type d'imagerie n'endommage pas le corps humain et est donc largement plus sûre que le CT scan, ce qui justifie sa préférence au sein du domaine mais surtout pour les examens d'imagerie du cerveau et de la moelle épinière [20].

Cependant, l'IRM présente une résolution spatiale plus faible, et un temps d'acquisition plus long ce qui induit plus d'artefacts. Plus le temps de balayage est long plus le bruit thermique est important. De plus les images médicales sont souvent contaminées par divers types de bruit lors de l'acquisition.

L'intensité des données IRM en présence de bruit suit généralement une distribution de Rician lorsqu'elles sont acquises avec des systèmes à bobine unique[20]. Par ailleurs la distribution Gaussienne peut approximer le bruit Rician dans les régions à fort SNR[20]. Très souvent, le bruit affectant les pixels d'une image est de nature gaussienne et détériore uniformément les pixels d'information de l'image. Il est donc nécessaire de trouver des moyens pour atténuer le bruit dans les images de type IRM, notre problème devient alors un problème de débruitage, dans le but de retrouver des images aussi proches des images "idéales" (Vérité terrain).

Le problème de débruitage est l'une des bases du traitement des images classique, où l'image est lissée à l'aide d'un filtre de lissage. Or ces approches lissent les images, mais, en se débarrassant du bruit les détails fins nécessaires à un diagnostic médical fiable sont omis.

Pour pallier ces limitations, deux approches sont proposées. Une première approche classique de machine learning repose sur un **SVM (Support Vector Machine)**. La deuxième approche se base sur les **GANs (Generative Adversarial Networks)**. Cette dernière est plus avancée car elle tire profit des capacités des réseaux de neurones adversariaux à apprendre des "patterns" complexes et à générer des images de haute qualité.

Ce projet vise à évaluer l'efficacité de ces méthodes (classique Machine learning Vs Deep learning) dans le contexte d'IRM bruitées, en explorant une pipeline améliorant la qualité des images médicales.

### 3 Etat de l'art sur le débruitage

Les méthodes de débruitage classiques sont explicités dans le cadre du traitement d'images et du filtrage dans cette section. Les algorithmes avancés notamment les SVM et les méthodes génératives seront abordés ensuite, en donnant une vue générale sur les principes et formulations mathématiques qui leurs sont associés.

#### 3.1 Les méthodes classiques de débruitage

Généralement l'étude de signaux (et d'images) nécessitent l'atténuation au maximum du bruit dû à d'autres facteurs. Le filtrage est alors utilisé pour débruiter le signal en préservant le maximum d'information. Le débruitage dans le domaine spatial est l'opération de filtrage qui élimine les fréquences élevées[4] (c'est à dire les détails fins et les variations brusques) par le biais de filtres qui agissent sélectivement sur les fréquences pour les sélectionner. Le bruit est souvent aléatoire et entache les images aléatoirement, il en résulte des variations d'intensité brusques et nécessite donc un filtrage passe-bas pour débruiter.

Un filtre **passe-bas** va éliminer ou atténuer fortement l'amplitude des hautes fréquences dans le domaine spectral et ne laisse passer que les basses fréquences [4].

Le filtrage linéaire spatial est une opération de convolution 2D, tel que l'image est **convoluée** avec la réponse impulsionnelle du filtre :

Soit  $i$  et  $j$  appartenant respectivement à  $[-\frac{m}{2}, \frac{m}{2}]$  et  $[-\frac{n}{2}, \frac{n}{2}]$ .

$$I'(x, y) = \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} h(i, j) \cdot I(x - i, y - j) \quad (1)$$

Tel que  $I(x, y)$  est l'intensité du pixel à la position  $(x, y)$  dans l'image originale et  $h(i, j)$  est le masque de convolution.  $I'(x, y)$  est la valeur filtrée du pixel  $(x, y)$ .  $m, n$  sont les dimensions de la fenêtre de convolution.

Si le noyau de convolution est carré alors  $m = n$  et est généralement impair et symétrique. Les noyaux de convolution sont à support beaucoup plus petit que celui de l'image, pour que la convolution soit appliquée localement autour du pixel. La valeur du pixel est alors la moyenne pondérée des intensités des pixels voisins, le processus est montré sur la figure 1.

La somme est normalisée pour que la somme des poids du filtre soit égale à l'unité, pour que la luminosité de l'image soit conservée :

soit  $k \in [1, n * m]$  :

$$\sum_k W_k = 1 \quad (2)$$

Un type de padding est utilisé pour ajouter des bords soit par des zéros ou par réflexion, ainsi tous les pixels de l'image seront traités.

Parmi les filtres passe-bas usuels pour le débruitage et le lissage, existent :

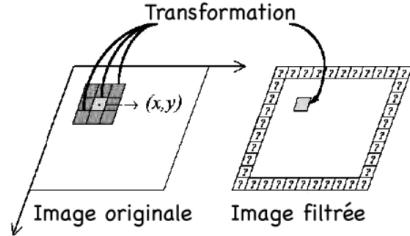


FIGURE 1 – Filtrage local[4]

- **Filtre moyenneur** : il affecte à chaque pixel la somme moyennée de toutes les intensités de son voisinage[4], son noyau de convolution est représenté sur la figure 2.

$$\frac{1}{9} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{25} \cdot \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

FIGURE 2 – Filtres de moyenne de tailles 3 x 3 et 5 x 5 respectivement[4]

- **Filtre gaussien** : Un filtre gaussien est la discréétisation de la fonction gaussienne bi-dimensionnelle[4] :

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2 + y^2}{2\sigma^2} \quad (3)$$

la moyenne est dans ce cas nulle et l'écart-type  $\sigma$ .

$x, y$

sont les indices de position dans la fenêtre gaussienne.

La taille du filtre est gouvernée par la valeur de l'écart type, si ce dernier est égal à 0.8 le filtre est alors de taille 3x3 comme montré dans la figure 3.

Souvent un écart type inférieur à 1 est utilisé car plus celui-ci est grand plus l'image est floue, moins il est grand plus le filtre est localisé et le lissage en devient plus fin.

D'autres filtres existent pour du débruitage classique, tels que le filtre médian (non-linéaire), le filtre de butter-worth passe-bas pour un filtrage fréquentiel.

Le lissage par filtres de convolution permet d'atténuer efficacement le bruit, cependant, la finesse des détails des IRMs en est également atténuée. Or le compromis entre la **réduction de bruit** et la **préservation des détails** est important pour les interprétations.

$$\begin{array}{|c|c|c|} \hline G(-1, -1) & G(0, -1) & G(1, -1) \\ \hline G(-1, 0) & G(0, 0) & G(1, 0) \\ \hline G(-1, 1) & G(0, 1) & G(1, 1) \\ \hline \end{array} \simeq \frac{1}{16} \cdot \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

FIGURE 3 – Noyau fini gaussien de taille 3x3

Les nouvelles approches de débruitage dans la littérature[10] intègrent le machine learning, notamment les SVM[11] afin d'aller au delà des inconvénients des méthodes classiques.

## 3.2 Suport Vector Machines

### 3.2.1 Principes théoriques du SVM

Le SVM est un algorithme supervisé qui peut être pour la classification mais également dans le cadre de la régression [17]. L'objectif de cette méthode est de trouver un hyperplan (si les données sont séparables linéairement) qui sépare les données, avec une contrainte sur la marge. Le SVM cherche à séparer les classes avec une marge maximale entre l'hyperplan et les vecteurs à support (les points de données les plus proches de l'hyperplan).

Pour chaque point  $x_i$  l'estimation de  $y_i$  est :

$$f(x_i) = w^T x_i + b \quad (4)$$

Tels que :

- $w^T$  est le vecteur des poids de dimension égale à celle de  $x_i$ .
- $b$  est le biais.

La contrainte est que tous les points doivent vérifier la contrainte (SVM stricte) :

$$y_i f(x_i) \geq 1 \quad (5)$$

La solution est alors le résultat du problème d'optimisation suivant (formulation primale) :

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (6)$$

Sous la contrainte :

$$\forall i, (w^T x_i + b) y_i \geq 1 \quad (7)$$

Pour résoudre ce problème, les multiplicateurs de Lagrange sont introduits :

$\forall i \in [1, n]$ ,  $\alpha_i$  est le multiplicateur de Lagrange associée à la contrainte  $(w^T x_i + b) y_i \geq 1$ , avec  $\alpha_i \geq 0$ .

Le Lagrangien à minimiser est :

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i (w^T x_i + b) - 1] \quad (8)$$

Les dérivées par rapport au vecteur poids et le biais nous donnent :

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \quad (9)$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (10)$$

$w$  est alors la combinaison linéaire des données d'apprentissage  $x_i$ , leurs labels associés  $y_i$  et des multiplicateurs de Lagrange. La résolution est un problème d'optimisation, consistant à trouver les  $\alpha_i$  qui maximisent le problème duale, et permettent d'obtenir  $w$ .

$$\frac{\partial \mathcal{L}}{\partial b} = - \sum_{i=1}^n \alpha_i y_i = 0 \quad (11)$$

En remplaçant  $w$  dans  $L$  on obtient la formulation duale du problème qui devient :

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i^T x_j) \quad (12)$$

Le problème à optimiser est fonction du produit scalaire  $(x_i^T x_j)$ , sous les contraintes :

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \forall i \quad (13)$$

Le problème est devenu un problème de maximisation des  $\alpha_i$ . Nous introduisons un **Kernel Trick** (astuce du noyau), qui sert à séparer les données dans un hyperplan non linéaire en conservant un classifieur linéaire. Le principe est de simplifier le produit scalaire des espaces de données afin d'étendre le SVM à un espace de plus grande dimension. Nous remplaçons alors le produit scalaire  $(x_i^T x_j)$  par le noyau  $K(x_i, x_j)$ , tel que

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

avec  $\phi(x)$  est une fonction pour augmenter l'expressivité de la donnée  $x$ .

Le problème devient alors :

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (14)$$

Sous les contraintes :

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \forall i \quad (15)$$

Ceci conclut l'idée générale et la théorie derrière la classification par SVM.

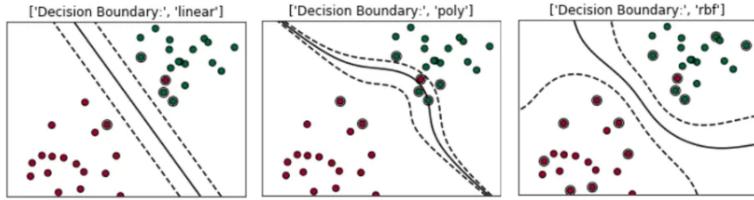


FIGURE 4 – Classification par SVM dépendant du kernel utilisé. de gauche à droite : une séparatrice linéaire, une séparatrice polynomiale, une séparatrice gaussienne[17]

### 3.2.2 SVM et débruitage d’images

Dans le cadre du machine learning, les **SVM** ont été détournés de leur usage classique de classification, pour être utilisés pour débruiter les images. L’article de McCACKIN et SHIRANI[11] porte sur une méthode où le SVM est utilisé pour sélectionner dynamiquement la meilleure méthode de débruitage (parmi plusieurs) à appliquer pour chaque pixel en se basant sur des caractéristiques locales comme l’énergie des coutures et la saillance visuelle. Cette approche a donné de meilleurs résultats que d’utiliser chaque méthodes isolément, les images sont débruitées tout en gardant les détails importants des images.

L’article de ZHANG et al.[21] propose d’utiliser le **SVM**, comme modèle de régression (car il prédit une valeur continue) pour prédire directement la version débruitee de chaque pixel. Il apprend à reconstruire une image propre à partir d’exemples (en paires) d’images bruités et non bruités. Les résultats sont satisfaisants comparés aux méthodes classiques de filtrage, cependant, il est coûteux en calcul.

Le **SVM** a pour avantages de modéliser les relations complexes entre les pixels, de s’adapter aux caractéristiques locales de l’image et de préserver les détails fins des images selon ces deux articles.

Néanmoins chaque pixel de l’image devient une données, ce qui accroît la taille de la base de données, et le temps d’entraînement du modèle qui en dépend est alors augmenté. Un autre inconvénient est qu’un mauvais choix de features peut diminuer l’efficacité si l’on considère l’article de McCACKIN et SHIRANI[11].

### 3.2.3 Application au projet

Pour le débruitage d’IRMs, le SVM va classer les pixels (ou patchs d’image) en deux catégories. Le SVM va apprendre à détecter les motifs liés au bruit, et lesquels sont des structures normales d’IRMs. Après classification il sera possible d’appliquer un traitement spécifique seulement pour les pixels (ou régions) bruités, ce qui permettra de préserver les détails dans les régions non-bruitées. **Le SVM réduit la perte excessive d’information**, mais dans le cas où un lissage sera appliqué aux régions bruitées, une perte pourrait être inévitable.

La deuxième approche, porte sur une approche basée sur les réseaux adverses génératifs GANs pour reconstruire directement une version propre (et de haute qualité) de l’image, en apprenant à générer des IRMs débruitées à partir des IRMs bruitées.

### 3.3 Algorithmes génératifs

Une approche largement utilisée dans le domaine de l'IA actuelle est celle des algorithmes génératifs. Ces algorithmes ont pour objectif de créer de nouvelles données en apprenant des motifs et des structures à partir des données sur lesquelles ils ont été entraînés. Les trois familles principales de systèmes génératifs sont les Réseaux Antagonistes Génératifs (GANs), les Autoencodeurs et les Modèles Autoregressifs SAKIRIN et KUSUMA [14]. Les GANs fonctionnent de manière antagoniste et sont composés de deux réseaux de neurones : un générateur et un discriminateur, qui s'affrontent. Les autoencodeurs, aussi utilisent des réseaux de neurones, apprennent à encoder les données d'entrée dans un espace de dimension inférieure puis à les décoder pour retrouver l'espace original. Les modèles autoregressifs, en revanche, tentent de prédire une sortie basée sur des prédictions précédentes.

Les GANs, en détail, se composent de deux parties principales comme mentionné précédemment : le générateur, qui tente de trouver la meilleure distribution  $p$  qui caractérise les données, et le discriminateur, qui tente d'identifier si un échantillon est réel ou faux. Le générateur  $G$  est défini par une distribution  $p(z)$  sur un vecteur  $z$ , qui sert d'entrée à la fonction génératrice  $G(z, \theta_G)$ . Le générateur cherche à trouver le meilleur  $\theta_G$ , où  $\theta_G$  représente les paramètres qui décrivent notre générateur  $G$ . Les distributions couramment utilisées sont une distribution gaussienne de haute dimension ou une distribution uniforme sur un hypercube GOODFELLOW et al. [7].

Les ensembles de données utilisés dans ces systèmes proviennent généralement d'une distribution inconnue  $d$  GOODFELLOW et al.[7], et l'objectif est que le modèle apprenne une distribution  $\hat{d}$  qui approche la véritable distribution  $d$ . Ces méthodes sont utilisés à la fois pour générer de nouvelles instances de données et pour entraîner des classificateurs robustes.

Le discriminateur  $D$ , quant à lui, examine les échantillons  $x$  en utilisant ses paramètres  $\theta_D$  dans la fonction  $D(x, \theta_D)$ . Il cherche également à trouver le meilleur  $\theta_D$ .

#### 3.3.1 DaNet

Pour notre projet, nous nous sommes inspirés de l'article *DaNet* YUE et al. [19]. L'approche utilisée repose sur les algorithmes génératifs et, plus précisément, les GANs. L'adaptation des GANs nécessaire a été l'introduction de l'entraînement avec une paire d'images, ainsi que l'ajout d'un débruiteur. Cette approche est elle même inspirée par Triple-GAN. Ce GAN est divisé en trois parties principales (**Débruiteur, Générateur, Discriminateur**) et utilise des paires d'images, une image considérée comme une image claire et une image bruitée. Ces trois réseaux travaillent ensemble afin de s'améliorer et apprennent de l'ensemble d'erreurs (voir diagramme 5). Pour ce faire, nous utilisons l'approximation de la probabilité conjointe  $p(x, y)$ .

**Le Débruiteur :** Le débruiteur est responsable, à partir d'une image bruitée en entrée, de produire une image claire en sortie. Il utilise la distribution conditionnelle bayésienne  $p(\hat{x}|y)$ .  $\hat{x} = R(y)$  est l'image débruitée que nous obtenons en sortie, et  $y$  est l'image issue de la vérité terrain. Puisque nous supposons que  $y \sim p(y)$ , nous obtenons :

$$p_R(x, y) \approx p_R(x|y)p(y)$$

**Le Générateur :** La deuxième partie est le Générateur, qui a un rôle similaire mais inverse. À partir d'une image claire  $y$  et d'une distribution de bruit  $z$ , il génère une image bruitée. De plus, le générateur est capable d'apprendre la distribution correcte du bruit. Les équations sont similaires à celles du débruiteur, mais cette fois-ci nous avons deux variables :  $y$  et  $z$ . Ainsi, nous avons  $G(x, z) \sim p_G(y|x, z)$ . Supposons que  $z \sim p(z)$  et  $x \sim p(x)$ , nous obtenons alors :

$$\hat{y} = G(x, z)$$

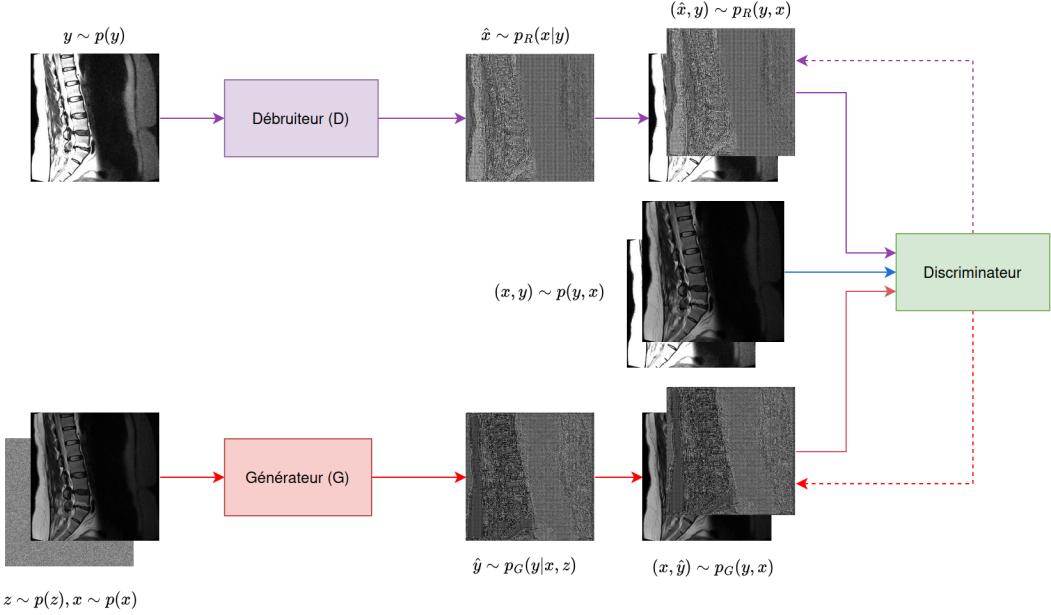


FIGURE 5 – Triple-GAN avec utilisation de paire d’images. Les lignes non continues représentent le retour d’erreur.

**Le Discriminateur :** Enfin, le dernier composant est le Discriminateur  $D$ , qui détermine si l’image est réelle ou fausse, c’est-à-dire si elle est bruitée ou claire. Il prend en entrée une paire d’images. Le discriminateur permet au générateur et au débruiteur d’améliorer leur paramètre et donc leur résultat. La fonction de perte a la forme suivante :

$$\min_{R,G} \max_D \mathcal{L}_{\text{gan}}(R, G, D) + \tau_1 \|\hat{x} - x\|_1 + \tau_2 \|GF(\hat{y} - x) - GF(y - x)\|_1 \quad (16)$$

avec  $\tau_1$  et  $\tau_2$  les deux hyper-paramètres de notre modèle  $GF$  le filtre Gaussien. Ce Gaussien est utilisé simplement pour la pénalité de régularisation. Pour le filtre  $GF$  la moyenne est nulle et l’écart-type  $\sigma$  dépend du noyaux choisi. La fonction de perte est dérivée de façon suivante :

$$\min_{R,G} \max_D \mathcal{L}_{\text{gan}}(R, G, D) = \mathbb{E}_{(x,y)}[D(x,y)] - \alpha \mathbb{E}_{(\hat{x},y)}[D(\hat{x},y)] - (1 - \alpha) \mathbb{E}_{(x,\hat{y})}[D(x,\hat{y})] \quad (17)$$

Le but de ce modèle est de rapprocher le plus possible  $p_R(x,y)$  (probabilité jointe du débruiteur) et  $p_G(x,y)$  (probabilité jointe du générateur) à  $p(x,y)$  qui est la probabilité jointe de notre véritable terrain.

Le DaNet représente une approche pertinente pour ce projet, car l’objectif final est de reconstruire des images IRM à partir d’images bruitées. Ce schéma nous permet non seulement de disposer d’un débruiteur performant pour traiter nos images, mais également d’un générateur capable de produire des images.

L’implémentation du modèle YUE et al. [19] est disponible en ligne : <https://github.com/zsy0AOA/DANet>.

## 4 Planification des tâches

Dans le cadre de ce projet, il était question d'améliorer la qualité des images de type IRM, qui dans certains cas comme celui de la **sténose du canal lombaire**, sont touchées par un bruit. Le but était alors de débruiter ces images (ou bien de pouvoir reconnaître le bruit dans ces images) pour ne pas fausser les diagnostiques cliniques. Ce travail a exploré alors différentes approches pour pallier ce problème, en touchant aux notions du traitement d'images et en passant par l'apprentissage automatique (**SVM**) et aller vers une approche générative (**GAN/DaNet**) de façon à pouvoir conclure sur la meilleure approche de débruitage des images sans perte d'information. Les grandes tâches effectuées sont représentées sur la figure 6 et l'organisation est illustrée par la figure 7.

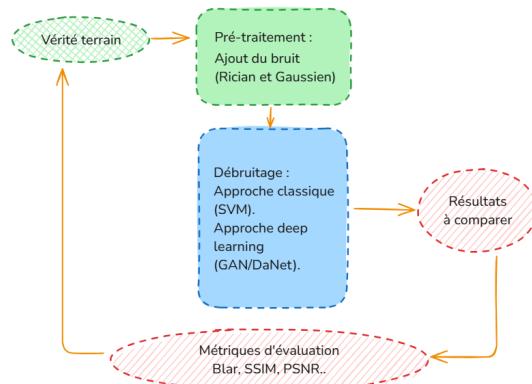


FIGURE 6 – Pipeline générale du projet. **Vert** : tâches effectuées au début. **Bleu** : tâches effectuées au mi-rapport. **Rouge** : tâches terminé en fin de projet

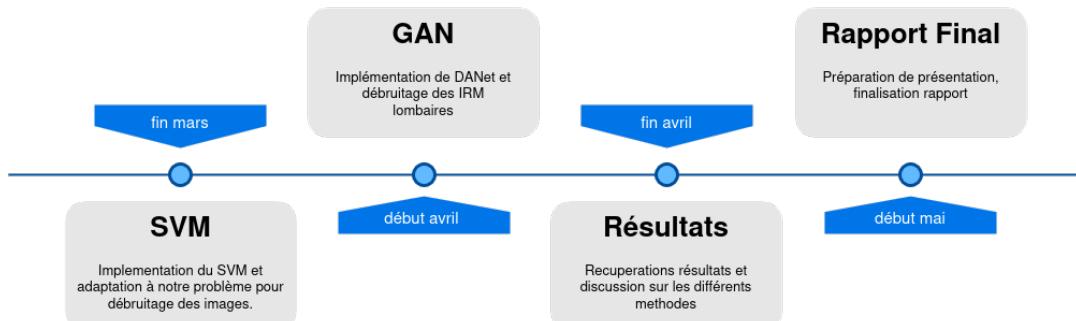


FIGURE 7 – Chronologie des tâches

## 5 Méthodes

### 5.1 Base de donnée d'IRM de sténose du canal lombaire

La base de données utilisée pour ce projet, est une étude clinique rendue anonyme, qui est constituée d'un ensemble de scans sur 515 patients avec des douleurs du dos symptomatiques [15]. Un patient peut avoir plusieurs études qui lui sont associées, et chaque étude contient des coupes (fichier ".ima", qui est soit d'une vue axiale ou bien sagittale), principalement sur les trois vertèbres les plus basses et les trois dernières IVDs. Un total de 48345 coupes IRM sont présentes dans cette base de donnée, dont la majorité est de résolution 360x360 pixels et, chaque pixel est codé sur 12 bits pour plus de précision et de clarté. Chaque acquisition a duré entre 15 à 45 minutes[15].

#### 5.1.1 Format des données

Les fichiers ".ima" sont des fichiers d'images médicales brutes, et ne contiennent pas directement de métadonnées, ils sont souvent accompagnés d'un fichier ".hdr" (header) contenant les informations nécessaires sur l'image tels que les dimensions, la résolution, le type de données, etc. Un fichier ".ima" est une image binaire brute et nécessite d'être accompagnée d'informations complémentaires pour pouvoir être interprétée.

Sur ce projet nous travaillerons en Python, pour pouvoir traiter les images de notre base de données qui pour un seul patient contient plusieurs coupes (environ 15 à 20 fichiers .ima). Afin de traiter ces données, une conversion du format ".ima" vers le format ".nii" est nécessaire. Le format ".nii" (NIFTI - Neuroimaging Informatics Technology Initiative) est un standard utilisé en neuro-imagerie et en imagerie médicale qui contient toutes les informations en un seul fichier. Il peut représenter une donnée image 3D ou 4D telle qu'une IRM volumétrique avec plusieurs coupes. Nous avons transformé un ensemble de fichiers ".ima" (correspondant aux différentes "coupes" d'une IRM d'un patient) à un fichier unique .nii (qui représente un volume 3D).

#### 5.1.2 Modalités d'imagerie : T1 et T2

L'imagerie par résonance magnétique (IRM) est une technique d'imagerie médicale qui utilise un champ magnétique puissant et des ondes radio pour produire des images détaillées des structures internes du corps[18]. Contrairement au scanner (qui utilise des rayons X), l'IRM est non ionisante[9][18] et permet une excellente différenciation des tissus mous, ce qui la rend particulièrement utile pour l'analyse du système nerveux et des pathologies rachidiennes. L'IRM repose sur la résonance des protons contenus dans l'eau du corps, les étapes pour avoir une IRM sont :

- Alignement des protons dans une direction précise, sous l'effet du champs magnétique[18].
- Une impulsion radiofréquence est utilisée pour exciter les protons, et qui changent ensuite d'orientation en l'absorbant.
- Lorsque l'impulsion est arrêtée, les protons reviennent à leur équilibre, et émettent un signal détecté par l'appareil IRM (Relaxation).
- Le signal mesuré est ensuite transformé en image en niveaux de gris, où l'intensité dépend de la concentration en eau et des propriétés des tissus.

Deux types (voir plus) de séquences IRM se distinguent dépendantes de leur temps de relaxation :

- T1 (Temps de relaxation longitudinal) est le temps pour lequel les protons reviennent à leur alignement initial après excitation. La séquence T1 est excellente pour visualiser l'anatomie générale du rachis, caractérisée par le liquide sombre et la graisse blanche[18].

- T2 (Temps de relaxation transverse), est le temps pendant lequel le signal diminue après excitation, la séquence T2 est utile pour détecter les pathologies comme la sténose. Caractérisée par un liquide clair (blanc lumineux) et aussi par la moelle osseuse et tissus étant plus foncés, mettant ainsi en valeur les zones remplies de liquide[18].
- Donc si l'on veut voir le rétrécissement du canal rachidien du à la sténose, la T2 est plus adaptée, car met en évidence l'effet sur la moelle épinière et les nerfs. la T1, est plus pertinente quant à elle pour une vision claire de l'anatomie osseuse et des disques.

L'IRM est sujette au bruit, venant de différentes sources. La T2 est particulièrement affectée par le bruit car son temps de relaxation est plus long, induisant donc plus de bruit aux respirations et aux mouvements. Un bruit élevé causera une perte de détails critiques sur la sténose du canal lombaire située en zone basse de la colonne vertébrale et peut par conséquent fausser ou altérer l'interprétation. De plus la taille du canal lombaire se rétrécissant avec le niveau de vertèbre observé, un bruit obtenu sur une zone de canal rétrécit, comme au niveau lombaire, engendre une difficulté de plus pour identifier des lésions sur cette zone.

Dans le cadre de l'analyse de la sténose du canal lombaire, l'IRM T2 est privilégiée car elle met en évidence les compressions nerveuses et l'impact sur la moelle épinière. Toutefois, l'IRM T2 est plus bruitée que la T1, rendant le dé-bruitage crucial pour améliorer l'interprétation des images.

## 5.2 Vérité terrain

Dans le cadre de ce projet, la définition d'une vérité terrain (Ground Truth) pertinente est essentielle pour l'entraînement et l'évaluation des modèles. En imagerie médicale, la vérité terrain est souvent constituée d'annotations manuelles réalisées par des experts, ou d'images de référence de haute qualité, exemptes de bruit et d'artefacts.

Toutefois, dans la base de données[15] sur laquelle porte ce travail, les images fournies ne sont pas aussi bruitées que celles rencontrées en pratique clinique[12]. Afin de rendre le modèle plus robuste et capable de généraliser, nous avons choisi d'introduire artificiellement du bruit dans les images d'entraînement. Cette approche permet de simuler des conditions réalistes et d'éviter le sous-apprentissage sur des données trop propres.

### 5.2.1 Choix de vérité terrain

Pour notre étude nous avons choisi de définir notre vérité terrain comme suit :

- Les images originales (non bruitées) sont alors considérées comme la référence.
- Des versions bruitées des images avec des modèles de bruitage (Rician ou Gaussien) sont générées.
- Le modèle d'apprentissage va apprendre à partir de ces données bruitées (ou non) pour pouvoir faire la distinction sur de nouvelles données. Cette étape est importante pour la généralisation.

### 5.2.2 Impact sur l'apprentissage

Une estimation du bruit réalisé précédemment en laboratoire sur ces images a montré qu'environ 6 à 11% du contenu des images de la base de données était bruité. Nous avons considéré que la base de données (open source) n'est pas très bruitée, et ne représente pas en général les IRM cliniques rencontrées, d'où la nécessité de brouter les images (et à différents type et niveaux de bruit, et ceci aura un impact sur l'apprentissage) :

- **Généralisation :** L'exposition du modèle à différents niveaux de bruit est une manière de rendre le modèle plus robuste aux variations que l'on rencontre en clinique.

- **Effet sur la performance :** Le modèle apprends à débruiter des images de qualité variable, pour éviter le sous apprentissage (à condition de trouver le compromis pour éviter le sur-apprentissage)
- Ce choix de vérité terrain est donc cohérent avec l'objectif du projet : développer un modèle robuste, capable de gérer des images bruitées et d'améliorer la qualité des données exploitables.

### 5.3 Pré-traitement des données

Afin de traiter les données la première étape a consisté à convertir les images du format `.ima` au format NIfTI (`.nii`) en utilisant la bibliothèque `dicom2nifti` (ou bien `SimpleITK`). Le format DICOM est une version plus généralisée du format `.ima`, c'est pourquoi la conversion a été réussie avec cet outil. De plus, il était nécessaire d'utiliser une autre bibliothèque pour convertir les données en un tableau NumPy afin de faciliter leur manipulation. La bibliothèque `nibabel` a été utilisée pour cette conversion. Il faut noter que nous allons travailler uniquement avec des images sagitales (SAG, qui ont été prises de droite à gauche) afin de ne pas manquer d'informations. Pour chaque type d'image, le nombre total d'images correspond au nombre de coupes prises.

De plus, nous avons écrit un petit script pour pouvoir afficher les différentes coupes. Le script est disponible sur le [GitHub](#) du projet, et plus spécifiquement, il s'agit du fichier `pre_affichage.py`. L'image 8 représente le résultat de ce programme avec les données du premier patient et 15 coupes.

### 5.4 Ajout du bruit

Pour créer notre vérité terrain, nous avons décidé d'utiliser deux distributions de bruit : la distribution gaussienne et la distribution Rician.

Avant de commencer, voici une définition importante : Le rapport signal sur bruit (SNR) est le rapport entre le signal désiré et le bruit présent dans l'image. Plus le SNR est bas, plus la qualité de l'image est dégradée.

#### 5.4.1 Distribution Rician

En raison de différents facteurs lors de l'acquisition des images, tels que le temps d'acquisition ou les contraintes du système, il a été observé que le SNR est assez faible [3]. Cela entraîne, dans les images de résonance magnétique (RM) en magnitude, une intensité moyenne des pixels souvent supérieure à l'intensité réelle de l'image à cause de la présence du bruit. Après plusieurs recherches [8][6], il a été remarqué que la distribution de Rician décrit bien ce phénomène. Sa densité de probabilité est la suivante :

$$f(x, \sigma) = \frac{x}{\sigma^2} \exp\left(\frac{-x^2}{2\sigma^2}\right) I_0\left(\frac{xy}{\sigma^2}\right) \quad (18)$$

Où  $I_0$  est la fonction de Bessel modifiée de première espèce et d'ordre 0. Un exemple d'image résultante l'application du bruit sur la première coupe d'un patient est donné figure 9.

#### 5.4.2 Distribution Gaussienne

Les images MRI sont souvent affectées par le bruit Gaussien. Le bruit Gaussien est un bruit électronique qui est causé par l'agitation thermique au sein des composants d'acquisition[20]. Ce bruit réduit parfois significativement

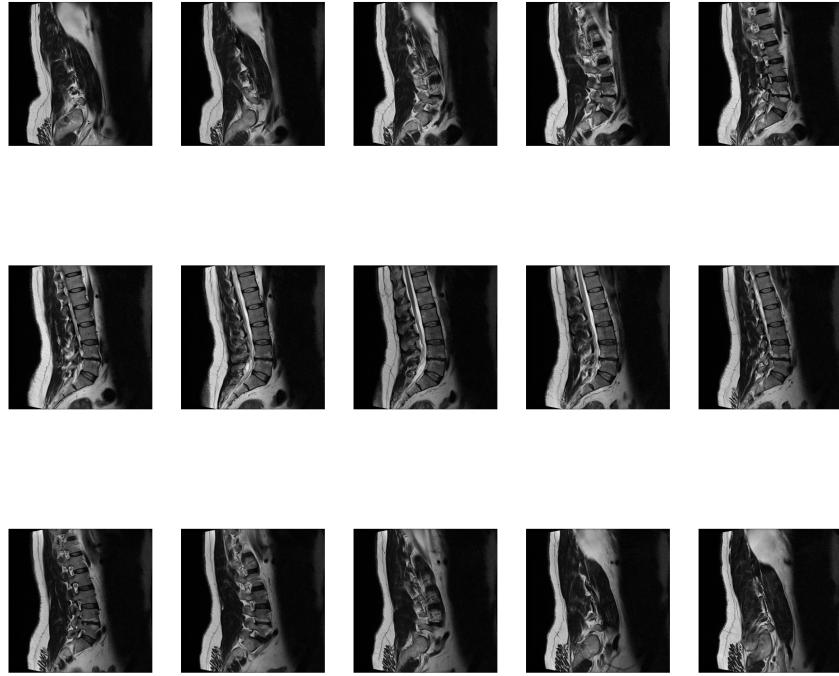


FIGURE 8 – Visualisation des coupes constituant le volume du patient n°1 provenant du dataset [15].

la qualité des images, ce qui rend délicat son utilisation dans le cadre des images médicales. D'où le choix du bruit gaussien pour le bruitage d'IRMs dans le cadre de ce projet. La densité de probabilité du bruit gaussien est :

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (19)$$

L'effet d'un tel bruit peut être visualisé dans la figure 10.

La quantité de bruit dans de vraies IRM cliniques est sensé être entre 6 à 11% [12]. La différence entre l'image bruitée et l'originale peut être visualisée dans la figure 10 à droite.

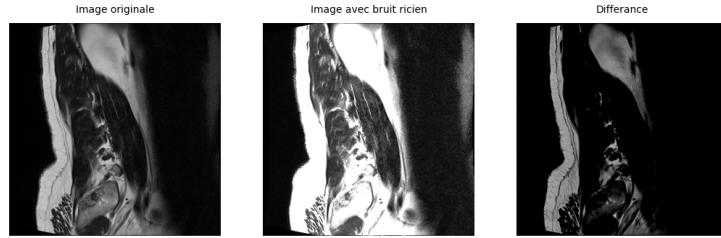


FIGURE 9 – Première coupe du volume d'origine du patient n°1 et, bruitée par bruit ricien du dataset [15]

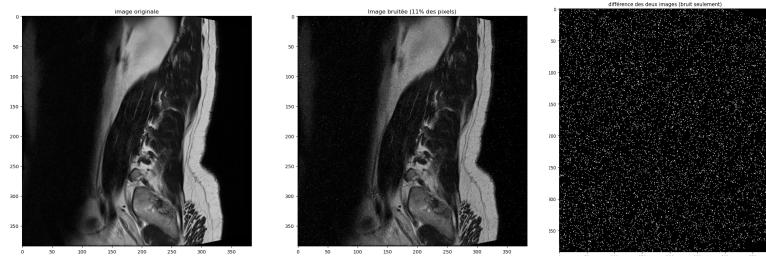


FIGURE 10 – Première coupe du volume d'origine du patient N°1 [15], au centre bruitée par du bruit gaussien, et à droite la quantité de bruit.

Ce dernier bruit a été généré en utilisant un bruit gaussien centré et appliqué seulement sur une sélection de pixels (11% de pixels de l'image). Ce bruit semblable à du "Salt And Pepper noise" est généré avec une distribution gaussienne avec le code **python** suivant :

```
def add_gaussian_noise(image, mean=0, stddev=0.35, ratio = 0.11):
    noisy_image = image.copy().astype(np.float32)
    m, n = image.shape
    nbrpx = int(m*n*ratio)
    indices = np.random.choice(m*n, nbrpx, replace=False)
    coords = np.unravel_index(indices, (m, n))

    bruit = np.random.normal(mean, stddev, nbrpx)
    noisy_image[coords] += bruit

    return np.clip(noisy_image, 0, 65535).astype(np.uint16)
```

On peut visualiser l'effet d'un bruit gaussien (sur tous les pixels de l'image) sur la figure 11.

**Remarque :** le bruit résultant des mouvements et respirations des patients touche généralement tous les pixels d'une image (comme celui sur la figure 11). D'autre part le bruit du matériel électronique en est un de type **Salt**

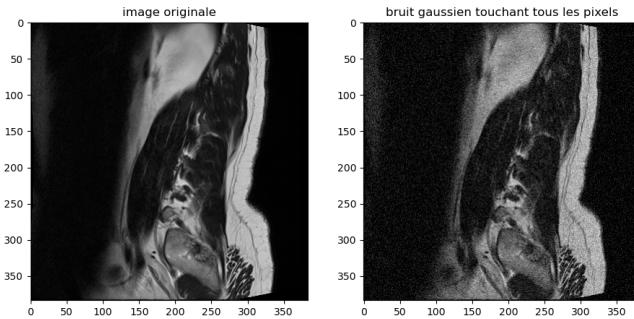


FIGURE 11 – Visualisation du bruit gaussien appliqué à une coupe du patient n°1 [15]

**And Pepper**, alors il est aussi intéressant d'envisager de combiner ces deux types de bruit pour en faire un encore plus réaliste ou bien d'utiliser les deux à but d'enrichir la base de données.

## 5.5 Métriques d'évaluation

L'évaluation des méthodes sur lesquelles porte cette étude, se base sur l'utilisation de plusieurs métriques d'évaluation. Plusieurs méthodes existent pour évaluer un modèle dans le contexte du traitement d'images (notamment le débruitage), les plus courantes sont la précision (accuracy) et l'erreur quadratique moyenne (MSE - Mean Squared Error), qui visent à évaluer l'apprentissage et la phase de test du modèle. Cependant, il existe des méthodes plus spécifiques pour les GANs. Les plus populaires pour ce type de modèle sont le FID (Fréchet Inception Distance) et l'IS (Inception Score) [5].

Ces deux métriques reposent sur l'utilisation d'un modèle pré-entraîné. Le FID intègre un ensemble d'images générées par le GAN dans un espace de caractéristiques défini par une couche spécifique de ce modèle pré-entraîné. Ensuite la moyenne et l'écart type des caractéristiques des vraies images par rapport à celles des images générées sont calculées. L'IS fonctionne de manière similaire, mais dans son cas, la moyenne de la divergence KL (Kullback-Leibler) est calculée [1].

Étant donné que ces deux méthodes utilisent un modèle pré-entraîné, elles peuvent nous donner des résultats biaisés, car les données de l'entraînement du modèle pré-entraîné peuvent fortement influencer les résultats. Par conséquent, nous allons nous concentrer sur les métriques qui évaluent la qualité de nos images générées (PSNR, MSE, BLUR) mais aussi des métriques qui évaluent la conservation de leur structure (SSIM, IM).

- Le SSIM calcule la similarité entre deux images données (générée et réelle) à différents niveaux : luminosité (L), contraste (C) et saturation (S). La mesure suit la formule suivante :

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

où  $x, y$  sont les deux images à comparer,  $\mu_x, \mu_y$  sont les moyennes,  $\sigma_x^2, \sigma_y^2$  les variances,  $\sigma_{xy}$  la covariance,  $C_1$  et  $C_2$  les constantes de stabilisation pour éviter la division par zéro, souvent définies comme : -  $C_1 = (K_1 L)^2$

-  $C_2 = (K_2 L)^2$  Où  $L$  est la dynamique des pixels et  $K_1$  et  $K_2$  sont de petites constantes. Il est également important de noter que la comparaison des deux images se fait par des morceaux des images. Le SSIM produit une valeur entre -1 et 1, où 1 indique que les deux images sont identiques, 0 indique aucune similarité, et des valeurs négatives indiquent des différences significatives.

- Le PSNR mesure la quantité de bruit présente dans l'image générée par rapport à l'image originale. Sa formule est :

$$\text{PSNR} = 10 \log_{10} \left( \frac{R^2}{\text{MSE}} \right)$$

Où  $R$  est la valeur maximale possible des pixels de l'image. Le PSNR est souvent conseillé d'être combiné avec le SSIM, car le PSNR ne prend pas en compte certains aspects perceptuels de la qualité d'image. Les valeur  $> 30db$  sont considérée comme des résultats satisfaisants.

- Le BLUR, [16] dans le cadre de notre projet, es utiliser afin d'identifier si les images débruitées sont altérées en terme de qualité. C'est une métrique basée sur la variance du Laplacien de l'image, qui donne une information sur la quantité de contours brusques dans l'image, moins la variance est grande, plus l'image est floue. La métrique de flou est définie comme la variance de l'image Laplacienne :

$$\text{Blur}(I) = \text{Var}(\Delta I) \quad (20)$$

Où :

- $I$  est l'image en niveaux de gris.
- $\Delta I$  représente l'image obtenue en appliquant un filtre Laplacien a  $I$ .
- $\text{Var}(\cdot)$  désigne la variance des intensités sur l'image filtrée.

- L'AKLD, calcule la moyenne de la divergence KL. La divergence KL est celle qui nous indique le taux de dissimilarité entre nos deux images. L'AKLD est calculé par la formule suivante :

$$AKLD = \frac{1}{L} \sum_{j=1}^L KL(P, Q)$$

Où :

$$P = \mathcal{N}(x^r, \text{diag}(V^f))$$

Et :

$$Q = \mathcal{N}(x^r, \text{diag}(V^r)).$$

De plus, nous avons :

$$V^c = \text{GF}((y^c - x^r)^2), \quad c \in \{f, r\}.$$

Dans cette formule, GF représente le filtre Gaussien avec  $\mu = 0$  et  $\sigma = 1$ . Ici,  $x^r$  est l'image réelle,  $y^r$  est l'image attendue de notre vérité terrain, et  $y^f$  est le résultat de notre générateur lorsque nous lui fournissons  $x^r$  en argument, c'est-à-dire :

$$y^f = G(x^r).$$

L'AKLD peut prendre des valeurs allant de 0 à l'infini. Des valeurs faibles de l'AKLD indiquent une grande similarité entre l'image générée et l'image originale.

- L'Information Mutuelle est utile pour mesurer combien d'information d'une variable existe dans l'autre. C'est un moyen efficace de déterminer à quel point nos images sont similaires. La formule qui la décrit est la suivante :

$$IM = H(x^r) + H(x^f) - H(x^r, x^f)$$

où  $H$  représente l'entropie,  $x^r$  est l'image de notre vérité terrain, et  $x^f$  est l'image fournie par le modèle. Les valeurs de l'Information Mutuelle peuvent varier de 0 à l'infini, avec zéro signifiant qu'il n'y a rien en commun entre les deux images, et l'infini dans des cas théoriques où les variables sont parfaitement dépendantes. RUSSAKOFF et al. [13]

## 5.6 Comparaison de modèles

Pour le débruitage des IRMs, deux approches sont comparées en termes de performances selon des métriques appropriées.

### 5.6.1 DANET

Pour l'implémentation de la première approche, nous nous sommes inspirés des recherches de YUE et al. Une fois la structure principale établie, nous avons créé différents modèles afin d'expérimenter et de déterminer sous quelles conditions le modèle de débruiteur fonctionne le mieux, ainsi que quelles modifications pourraient potentiellement améliorer les résultats.

**Modification de l'architecture du modèle** Nous avons tout d'abord réduit la taille des filtres utilisés dans les couches de convolutions afin de pouvoir exécuter les codes localement (16, 32, 64) comparé à l'architecture décrite dans l'article initial.

**Mise en place des hyperparamètres** Lors de l'entraînement de l'une des premières architectures, nous avons fixé les mêmes taux d'apprentissage pour les trois modèles. Nous avons constaté qu'il y avait un déséquilibre entre l'apprentissage du discriminateur (1.0 de précision atteint dès la deuxième epoch), du générateur et du débruiteur (0.6). Par conséquent, nous avons réalisé qu'il était nécessaire d'utiliser des valeurs différentes pour chacun d'eux. La recherche des meilleurs hyperparamètres a conduit à garder un taux d'apprentissage à 0.0002 pour le débruiteur et le générateur et, à 0.00005 pour le discriminateur, un alpha de 0.5, et enfin, un nombre critique pour le moment à 1. Ce nombre critique détermine le nombre de fois où le discriminateur est entraîné pendant une epoch. Malgré cela, nous avons observé lors de l'entraînement que le discriminateur avait une plus grande force de discrimination que les deux autres réseaux (1.00 de précision dès quelques epochs sans laisser le temps au débruiteur de s'entraîner).

**Régularisation** Nous avons tenté alors de régler le déséquilibre par des méthodes dites de régularisation comme faire varier le dropout (c'est à dire d'éliminer un certain nombre de neurone différents pour chaque batch). De 0.2 initialement, ce paramètre de régularisation a finalement été fixé à 0.4. La deuxième méthode de régularisation utilisée a été du label smoothing qui est utilisé lors de l'entraînement où un facteur plus important est accordé aux données générées vis à vis des réelles afin de rétablir là encore le déséquilibre. En effet, en multipliant par 0.3 les données générées et 0.7 aux données réelles, le rôle du discriminateur qui est censé identifier les données réelles des générées

en voit sa tâche plus complexifiée et ralentit de fait son apprentissage (au lieu de commencer à 0.8 de précision, les premières epochs d'apprentissage ont indiqué plutôt 0.2 de précision pour le discriminateur) ce qui a permis de rétablir le dééquilibre.

Nos résultats que nous allons présenter concernent les architectures les plus stables et celles présentant les plus grandes différences.

La première architecture a trois modèles, le discriminateur, le débruiteur et le générateur ayant les même tailles des filtres de couches. Étant donné que le discriminateur est dans une boucle supplémentaire par rapport au débruiteur et au générateur, nous avons décidé d'attribuer des valeurs différentes pour le taux d'apprentissage. Ainsi, pour le discriminateur, nous avons assigné une valeur de  $ld = 0.00005$ , tandis que pour le débruiteur et le générateur, nous avons utilisé un taux d'apprentissage de  $lr = 0.0002$ .

La deuxième architecture consiste en trois modèles, comme précédemment, avec la différence que notre débruiteur et générateur ont une taille de filtres de couches plus petite, des neurones en dropout (0.4), et également une incertitude pendant l'apprentissage. Cela signifie qu'au lieu que nos labels soient 0 et 1, nous avons utilisé 0.3 et 0.7, ce qui affecte le calcul de la perte.

### 5.6.2 SVR

La deuxième approche met en œuvre un Support Vector Regressor (SVR). Une méthode qui repose sur les fondements théoriques du SVM, initialement conçu pour de la classification, mais peuvent être utilisés pour prédire des valeurs continues.

L'idée d'utiliser un SVR pour la reconstruction d'IRMs bruitées, a été inspirée du travail de ZHANG et al.[21], qui a proposé un modèle qui utilise un SVM apte à prédire la valeur propre de chaque pixel à partir de sa région locale (patch). Leur approche montre que le SVM est capable de capturer les relations complexes dans l'environnement local, induisant une reconstruction capable de préserver les structures fines des IRMs tout en atténuant le bruit.

Dans les expérimentations qui suivront, le modèle apprend à prédire la valeur du pixel central d'un patch de l'image bruitée, en utilisant comme vérité terrain le pixel qui lui correspond dans l'image propre. L'entraînement sur un grand nombre de taille de patchs extraits d'images bruitées avec différents types de bruits (et parfois un seul dans des cas spéciaux). Le modèle est ensuite utilisé pour reconstruire des images entières, patch par patch.

Ces modèles seront comparés qualitativement (visuellement) et quantitativement (par métriques) dans la section **Résultats** suivante.

## 6 Résultats

### 6.1 Résultats Qualitatifs

#### 6.1.1 Débruitage par SVR

Cette partie présente les résultats visuels (images débruitées) de l'ensemble des expérimentations menées à l'aide du SVR selon les trois axes : l'influence de la taille du patch et du niveau de bruit sur un modèle entraîné sur des données hétérogènes, l'impact du chevauchement (apprentissage sans chevauchement entre les patchs) et la robustesse du modèle à plusieurs patients et plusieurs types de bruit.

##### 1. Étude multi-niveaux de bruit et multi-tailles de patchs :

- **Bruit faible** : pour cette étude, le choix des paramètres des bruits est débité ici : l'écart-type (std) du bruit gaussien est de 30, l'échelle du rician est de 30, pour le mixe un écart-type du gaussien valant 25 et une échelle du rician de 30 ont été utilisés. Les résultats sont illustrés sur la figure 12. Les images ont été débruitées, avec un lissage qui s'intensifie avec l'augmentation de la taille des patchs.
- **Bruit moyen** : les paramètres pris pour cette étude sont :  
Un std valant 40, une échelle de 53 et le mixe a un std de 33 et une échelle de 40. Les résultats peuvent être visualisées sur la figure 13. Les résultats sont analogues à ceux du bruitage faible, notamment un lissage qui s'accentue avec l'augmentation de la taille des patchs, une conclusion plus discriminante est à voir dans la section des résultats quantitatifs.
- **Bruit fort** : les paramètres sont :  
Un std de 80 pour le gaussien, échelle de 50 pour le rician, pour le mixe un écart-type de 60 et une échelle de 40 ont été utilisés. Les résultats peuvent être visualisés dans la figure 14. Il n'y a eu de débruitage interprétable que pour la taille de patch 3, pour les tailles 5, 7 et 9, le modèle procure des images très sombres, ou il n'est pas possible de voir ou de distinguer les structures présentes dans l'image originale.

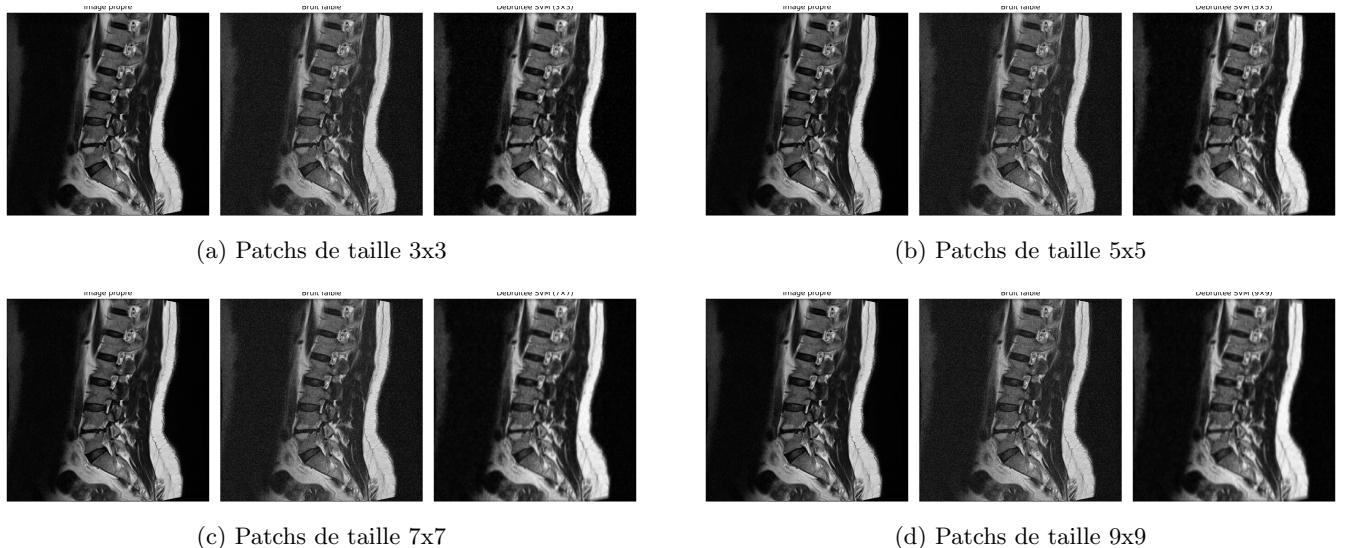


FIGURE 12 – Comparaison des résultats de débruitage par SVR sur une image bruitée avec du Rician (faible). Chaque sous-image présente, de gauche à droite : la vérité terrain, l'image bruitée et l'image débruitée. De haut en bas, la taille du patch utilisée pour l'entraînement augmente (3x3, 5x5, 7x7, 9x9).



(a) Patchs de taille 3x3



(b) Patchs de taille 5x5



(c) Patchs de taille 7x7

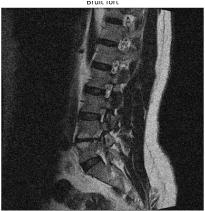


(d) Patchs de taille 9x9

FIGURE 13 – Débruitage d'une image bruitée avec du bruit mixte (niveau moyen), à l'aide d'un SVR entraîné avec différentes tailles de patchs. Chaque sous-image montre (de gauche à droite) la vérité terrain, l'image bruitée et l'image débruitée. De haut en bas, la taille du patch augmente.



(a) Patchs de taille 3x3



(b) Patchs de taille 5x5



(c) Patchs de taille 7x7



(d) Patchs de taille 9x9

FIGURE 14 – Débruitage d'une image bruitée avec un bruit mixte (niveau fort), à l'aide d'un SVR entraîné avec différentes tailles de patchs. Chaque image montre (de gauche à droite) la vérité terrain, l'image bruitée, puis l'image débruitée. De haut en bas, la taille des patchs augmente.

**2. Étude sans chevauchement :** ici le pas du glissement des patchs est égal à la taille du patch, ce qui signifie qu'il n'y aura pas de chevauchement. Cette étude se fait seulement sur du bruit moyen et vise à voir l'effet du chevauchement (ou non) sur les différentes tailles, pour vérifier si ce non-chevauchement améliore la qualité du débruitage.

Les résultats peuvent être visualisées sur la figure ???. De la même manière que l'étude précédente, les images deviennent de plus en plus floues en augmentant la taille des patchs. Ceci dit les résultats quantitatifs seront plus expressifs quand à la différence que pourrait apporter le chevauchement.

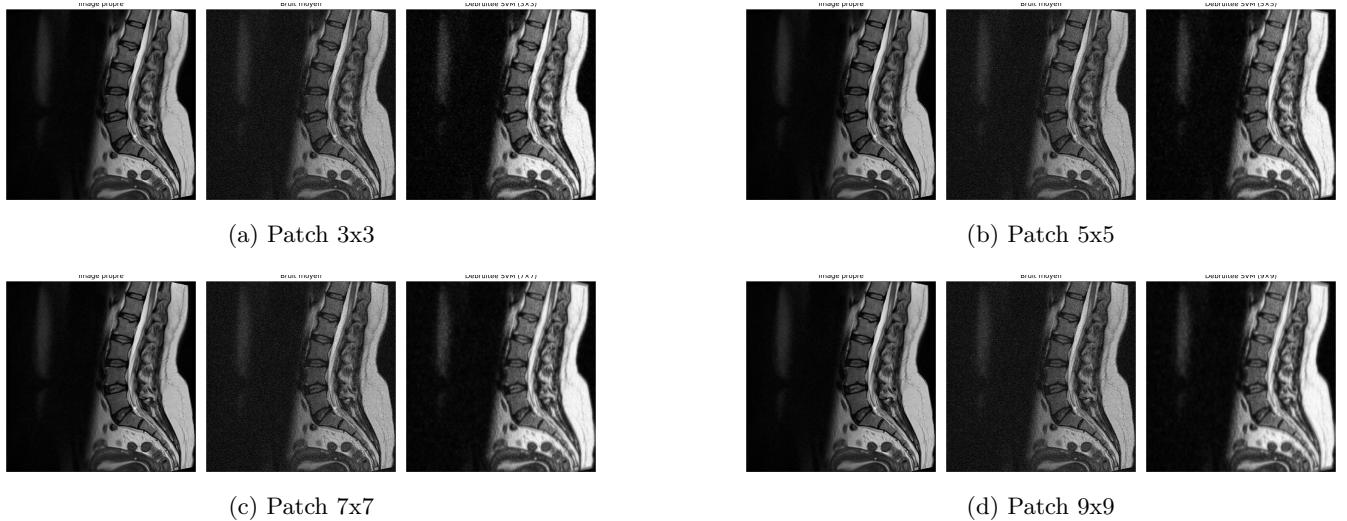


FIGURE 15 – Débruitage par SVR sans chevauchement pour différentes tailles de patchs. De gauche à droite : vérité terrain, image bruitée, image débruitée.

**3. Étude multi-patient/multi-bruit :** nous étudions la robustesse du modèle vis-à-vis des différents patients et la différence du type de bruit. Les images utilisées pour l’entraînement sont bruitées avec du bruit moyen. L’étude inclue aussi la variation de la taille des patchs.

Nous observons visuellement que les images des 3 patients pour un même bruit, ont préservé la structure anatomique (contours vertébraux et tissus, le bruit a bien été atténué par le SVR, surtout dans les zones homogènes. Le bruit visible sur l’image bruitée est fortement lissé dans la version débruitée.

Vis-à-vis du bruit, le résultat peut être plus ou moins différent, mais ceci est du probablement aux différences naturelles entre les régions et structures anatomiques prises figurantes dans les images (certaines sont plus sombres que d’autres de nature, d’autres contiennent plus de détails que d’autres..).

Les patients sur lesquels l’étude quantitative est menée sont les patients 103, 104 et 105.

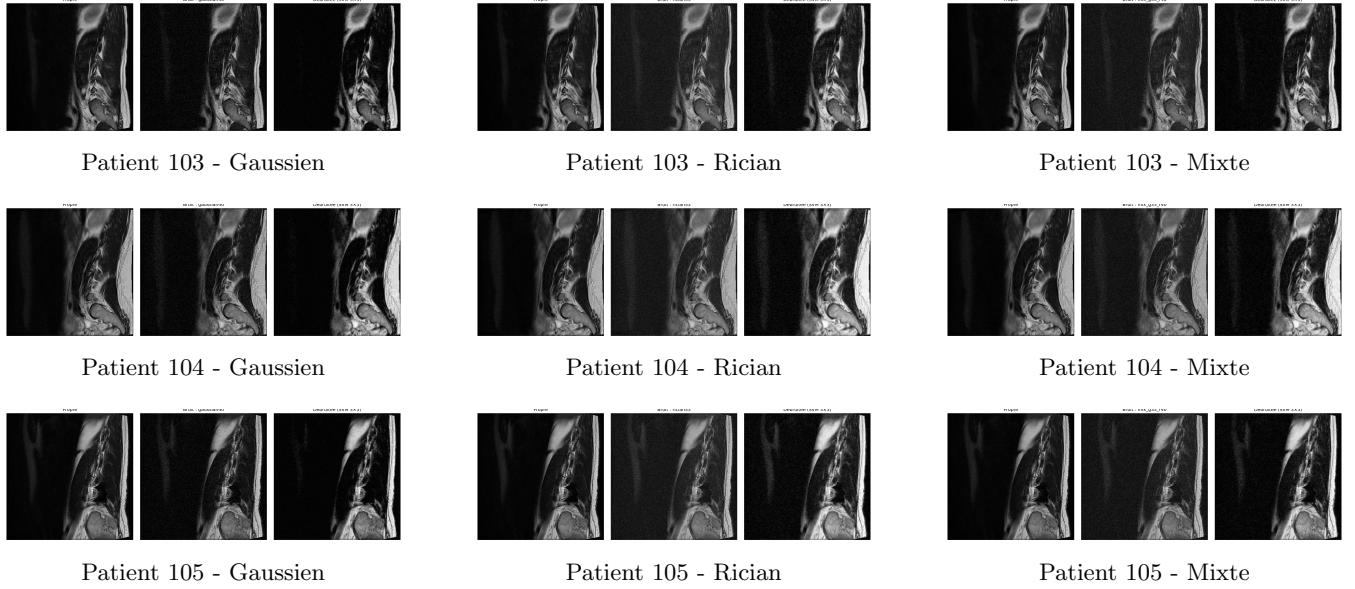


FIGURE 16 – Résultats de débruitage pour trois patients (103, 104, 105), sous trois types de bruit (Gaussien  $\sigma = 40$ , Rician  $\alpha = 53$ , Mixte  $\sigma = 33$ ,  $\alpha = 40$ ). Chaque image présente de gauche à droite : la vérité terrain, l'image bruitée, puis l'image débruitée.

### 6.1.2 Débruitage par GAN

Pour mieux visualiser les résultats de nos deux architectures du Triple-GAN, les Figures (17, 18) illustrent quatre images pour chaque patient, en présentant trois patients différents (patient n°102, n°103, n°104). La première linge présente les images originales, la deuxième linge montre les images originales avec un bruit de ricien, la troisième affiche les images générées par notre débrouteur lorsqu'il reçoit l'image bruitée comme argument, et la quatrième linge présente les images générées par le générateur en utilisant les images débruitées comme argument.

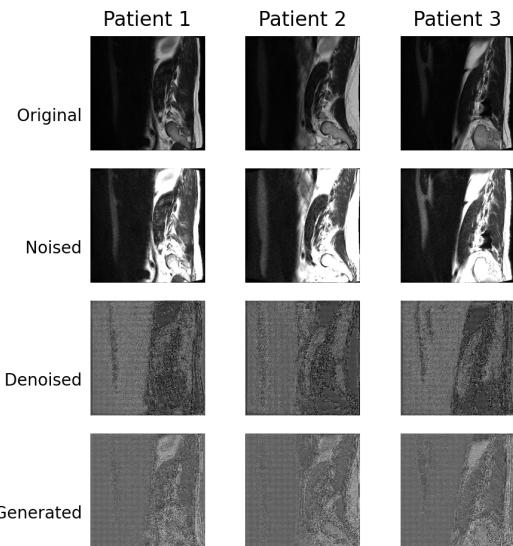


FIGURE 17 – Résultat des modèles de l'architecture 1

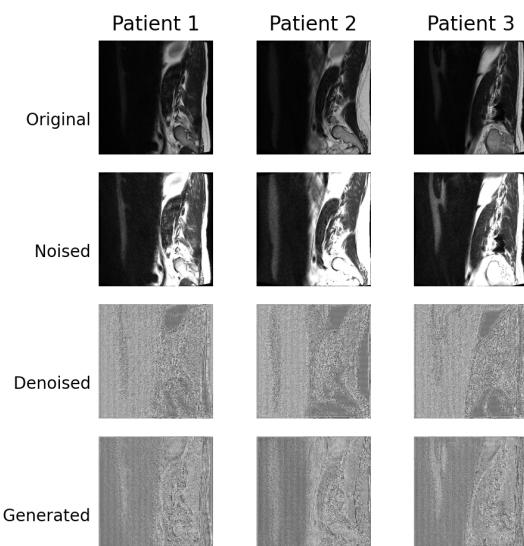


FIGURE 18 – Résultat des modèles de l'architecture 2

Ce que nous pouvons conclure de ces résultats, c'est que même en l'absence d'images nettes, il est possible

d'observer des détails tant dans les images générées par le générateur que dans celles produites par le débruiteur. La première architecture démontre une capacité d'entraînement supérieure, car les images présentées dans la Figure 17 révèlent davantage de détails par rapport à celles de la Figure 18.

## 6.2 Résultats Quantitatifs

### 6.2.1 SVR

Dans cette partie, nous abordons les résultats en termes de métriques d'évaluations, a but de voir l'impacte des différences dans les paramètres de notre modèle sur la qualité du débruitage.

**1. Étude multi-niveaux de bruit et multi-tailles de patchs :** Le tableau 1 présente les performances du modèle SVR en fonction de la taille des patchs utilisés pour le débruitage. Quatre métriques ont été analysées : le PSNR, le SSIM, le BLUR (flou basé sur la variance du Laplacien) et l'AKLD (divergence moyenne de Kullback-Leibler).

Les résultats montrent que l'utilisation de patchs de taille **3x3** fournit globalement les meilleures performances. En effet, cette configuration atteint le **meilleur PSNR moyen** (45.15 dB), un SSIM élevé (0.965), un **flou minimal** (7.69), et une **divergence KL très faible** (0.917), ce qui indique une bonne similarité statistique avec la vérité terrain. Ces valeurs traduisent une excellente capacité du modèle à atténuer le bruit tout en conservant les détails structurels de l'image.

En comparaison, les patchs de taille **5x5** induisent une **dégradation importante de la qualité**, avec une baisse drastique du PSNR et une explosion du flou (209 en moyenne avec un écart-type supérieur à 350). L'AKLD s'envole également, ce qui suggère une perte significative d'information utile. Les patchs **7x7** et **9x9** offrent des résultats intermédiaires : bien qu'un certain lissage soit constaté, ils demeurent moins stables et moins précis que les petits patchs.

On observe également que l'**augmentation de la taille des patchs augmente la variance des scores**, traduisant une instabilité du modèle selon les cas. Cette variabilité suggère que les grands patchs, bien qu'intuitivement porteurs de plus de contexte, **introduisent trop de lissage** et nuisent à la précision du débruitage.

En résumé, ces résultats confirment l'hypothèse selon laquelle les patchs plus petits (3x3) sont mieux adaptés au débruitage local des IRMs, car ils permettent au modèle de capturer les structures fines tout en atténuant le bruit.

Il est important de noter que dans nos expériences, chaque pixel est prédict une seule fois (avec chevauchement partiel entre les patchs), sans moyenne entre patchs chevauchants. Les effets de flou observés sont donc directement liés aux caractéristiques des patchs et non à la phase de reconstruction.

Taille patch	PSNR	var(PSNR)	SSIM	var(SSIM)	Blur	var(Blur)	AKLD	var(AKLD)
3x3	45.15	1.43	0.965	0.017	7.69	1.26	0.917	0.063
5x5	32.43	22.66	0.962	0.024	209.10	353.87	202.45	348.87
7x7	34.17	18.36	0.965	0.016	49.96	80.56	41.36	69.82
9x9	35.09	15.51	0.963	0.015	16.96	24.62	13.44	21.45

TABLE 1 – Résultats quantitatifs moyens selon la taille du patch utilisé pour le SVR (la moyenne sur tous les niveaux de bruit).

### 2. Étude sans chevauchement :

Taille patch	PSNR	SSIM	Blur	AKLD
3x3	44.29	0.975	8.42	0.961
5x5	43.07	0.972	5.33	1.054
7x7	41.93	0.967	3.84	1.087
9x9	41.01	0.962	2.98	1.102

TABLE 2 – Résultats quantitatifs moyens pour le débruitage par SVR sans chevauchement de patchs.

Pour évaluer l'effet du chevauchement entre les patchs sur la qualité de débruitage du SVR, dans le contexte du bruit moyen. Le tableau 2 présente les résultats quantitatifs obtenus sans chevauchement, et ils ont été comparés aux résultats précédents avec chevauchement (tableau 1).

D'un point de vue global, les performances en l'absence de chevauchement sont plus stables et moins sujettes à l'over-smoothing que dans le cas avec chevauchement. Pour les patchs de grande taille (5x5 à 9x9), le PSNR est nettement supérieur sans chevauchement : par exemple, en 5x5, le PSNR passe de 32.43 (avec chevauchement) à 43.07 (sans), et la métrique de flou chute de 209 à 5.33, révélant une réduction importante de l'effet de lissage excessif observé précédemment.

Le SSIM reste globalement similaire entre les deux approches, avec une légère supériorité pour la version sans chevauchement, notamment pour les petits patchs (0.975 en 3x3 contre 0.965 avec chevauchement). Cela indique que la structure de l'image est bien conservée dans les deux cas.

En revanche, la divergence moyenne (AKLD) est légèrement inférieure dans le cas avec chevauchement, ce qui traduit une meilleure correspondance statistique entre les images reconstruites et la vérité terrain. Ce compromis met en évidence que le chevauchement peut améliorer la cohérence globale des prédictions, mais au prix d'un flou important, surtout lorsque la taille du patch augmente.

Pour les petits patchs (3x3), la version avec chevauchement reste légèrement plus performante dans l'ensemble : meilleure AKLD (0.917 vs 0.961), PSNR supérieur (+0.8 dB), et flou plus faible. Cela confirme que le chevauchement peut être utile à petite échelle, mais devient contre-productif dès que la taille du patch augmente.

En résumé, l'absence de chevauchement entre patchs offre un meilleur équilibre entre précision, netteté et stabilité pour les tailles de patch  $\geq 5 \times 5$ . Le chevauchement n'est bénéfique que dans le cas de petits patchs très locaux (3x3), où il permet d'améliorer encore la fidélité sans générer d'effets de flou importants.

**3. Étude multi-patient/multi-bruit :** Cette étude compare les résultats du modèle SVR pour trois patients (103, 104 et 105) soumis à trois types de bruit moyen : gaussien, rician et mixte. Pour chaque cas, les métriques PSNR, SSIM, flou (BLUR) et divergence AKLD ont été évaluées.

Les résultats montrent que le bruit mixte donne en général les meilleurs scores de PSNR et SSIM pour tous les patients, ce qui signifie que le modèle arrive à reconstruire des images très proches de la vérité terrain en termes de structure et de qualité visuelle. Cependant, l'AKLD est un peu plus élevé dans ce cas, indiquant une légère différence statistique.

Le bruit gaussien est celui qui dégrade le plus les performances, surtout au niveau du SSIM, ce qui signifie que les détails fins sont plus difficiles à récupérer. Le bruit rician donne des résultats plus équilibrés, avec de bonnes performances sur toutes les métriques.

Ces résultats montrent que le modèle fonctionne de manière robuste sur plusieurs patients, mais que l'efficacité peut varier en fonction du type de bruit présent dans l'image.

Patient	Bruit	PSNR	SSIM	Blur	AKLD
102	Gaussian40	40.89	0.906	5.82	0.541
102	Rician53	45.39	0.971	6.80	1.067
102	Mix g33/r40	46.14	0.975	7.02	1.036
103	Gaussian40	39.31	0.888	7.57	0.538
103	Rician53	43.24	0.973	8.48	0.987
103	Mix g33/r40	43.74	0.972	8.92	0.924
104	Gaussian40	44.27	0.934	4.84	0.530
104	Rician53	44.27	0.965	5.99	0.864
104	Mix g33/r40	44.79	0.972	6.57	0.870

TABLE 3 – Résultats quantitatifs du modèle SVR pour trois patients sous différents types de bruit.

### 6.2.2 DANET

Pendant l'apprentissage de nos modèles, nous avons conservé les informations sur les fonctions de perte, et les figures (19, 20) montrent la variation de la perte de chaque modèle (générateur, débruiteur, discriminateur).

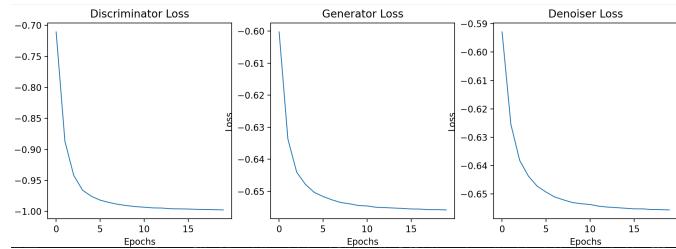


FIGURE 19 – Perte pendant l'apprentissage de l'architecture 1

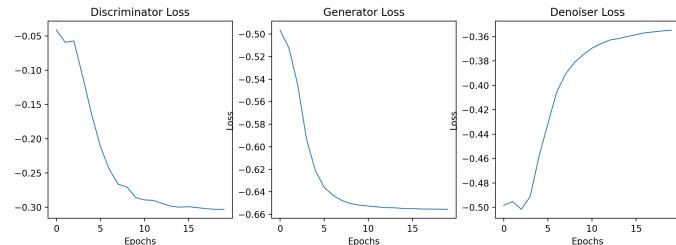


FIGURE 20 – Perte pendant l'apprentissage de l'architecture 2

À travers ces graphiques (19, 20), nous pouvons constater que le débruiteur de la deuxième architecture se comporte très différemment des autres modèles. Cela est dû au discriminateur, étant donné qu'en modifiant la

n\_critic) était égal à 1 et il avait un taux d'apprentissage (learning rate) plus faible.

Par contre, nous constatons que pour les six graphiques, nous avons commencé à observer une stabilisation au-delà de 15 epochs.

Les tableaux (4,5) présentent les différents résultats de chaque métrique pour chaque architecture de GAN calculée sous la forme de moyennes et écart-types calculées pour 88 images qui faisaient parties du test dataset. Le tableau 4 concerne la première architecture, comme mentionné précédemment, et le tableau 5 concerne la deuxième.

TABLE 4 – Résumé des Métriques pour le Débruiteur (R) et Générateur (G) de l'architecture 1

Métrique	Moyenne (R)	Écart-type (R)	Moyenne (G)	Écart-type (G)
MSE	<b>0.2184</b>	0.0064	<b>0.2659</b>	0.0165
Information Mutuelle	0.2546	0.0361	<b>0.3066</b>	0.0192
SSIM	0.0002	0.0008	-0.0008	0.0016
PSNR	<b>6.6093</b>	0.1261	5.7612	0.2700
AKLD	<b>228.4235</b>	20.7934	-	-
BLUR	<b>2.5325</b>	0.1173	4.8947	0.1497

TABLE 5 – Résumé des Métriques pour le Débruiteur (R) et Générateur (G) de l'architecture 2

Métrique	Moyenne (R)	Écart-type (R)	Moyenne (G)	Écart-type (G)
MSE	0.3919	0.0142	0.2827	0.0110
Information Mutuelle	<b>0.2799</b>	0.0393	<b>0.3328</b>	0.0241
SSIM	<b>0.0013</b>	0.0005	<b>0.0043</b>	0.0016
PSNR	4.0712	0.1571	<b>5.4903</b>	0.1688
AKLD	299.8864	22.1245	-	-
BLUR	2.4012	0.1262	2.7731	0.1620

Comme le montrent les tableaux (4,5), toutes les métriques s'accordent à dire que les images ne sont pas similaires. Nous remarquons que le débruiteur de la première architecture obtient de meilleurs résultats sur toutes les métriques qui ne prennent pas en compte l'information spatiale. Plus précisément, la première architecture affiche de meilleures performances en termes de MSE, qui mesure la différence entre les deux images, de PSNR, qui évalue la quantité de bruit dans l'image générée, d'AKLD, qui indique la dissimilarité entre les deux images, et enfin de Blur, qui nous renseigne sur le flou des images.

Visuellement, les images générées ne sont pas de très bonne qualité et présentent beaucoup de bruit. Dans notre cas, où la précision des images est cruciale pour éviter les erreurs, il est essentiel de trouver un bon équilibre entre les différentes métriques. Bien que les différences entre les deux architectures soient minimales pour les métriques IM, PSNR, AKLD et Blur, la divergence est beaucoup plus marquée pour le SSIM et le MSE. Par conséquent, nous préférions la première architecture.

## 7 Discussion

Comme le montrent les résultats quantitatifs et qualitatifs, les performances de notre Triple-GAN ne sont pas très satisfaisantes. Il est possible d'améliorer ces résultats en modifiant certains paramètres. La recherche de DaNet YUE et al. [19] a démontré qu'il est possible d'obtenir de bons résultats. Nos résultats actuels indiquent qu'avoir les mêmes tailles des filtres entre les différents modèles est bénéfique et constitue un bon point de départ.

L'utilisation des GANs est une méthode très utile, car elle permet d'obtenir des résultats satisfaisants. Malheureusement, il existe également certaines limites. Les modèles doivent être entraînés avec notamment de grands ensembles de données, et l'entraînement demande un temps considérable ainsi qu'une machine performante. Cela a représenté une contrainte significative que nous avons dû surmonter au cours de la réalisation du projet. Également, dans l'article d'origine, les images traitées n'étaient pas de type médical, en 3D, ou comportant de petites structures, ce qui a accru les contraintes et le niveau de difficulté de ce projet.

De plus, étant donné que le GAN est entraîné sur un type de données spécifique, il ne peut débruiter que des images déjà vues ou suffisamment similaires. La question se pose alors : le débruiteur entraîné par le GAN sera-t-il capable de débruiter une image qu'il n'a jamais vue, surtout si celle-ci présente un comportement atypique ?

D'un autre côté l'approche par Support Vector Regressor (SVR) a permis d'étudier les effets de plusieurs paramètres (taille de patch, chevauchement, type de bruit, variabilité patient) sur la qualité du débruitage. Les résultats quantitatifs ont montré que l'utilisation de petits patchs (notamment 3x3) maximise les performances (PSNR, SSIM) tout en limitant le flou, en effet moins le patch est petit, le moins de bruit il capture. Ce qui corrobore l'idée que la régression locale permet de préserver les détails fins des structures anatomiques. À l'inverse, l'augmentation de la taille des patchs provoque un lissage excessif et une dégradation visible de la qualité des images. Ces tendances se vérifient à travers toutes les configurations testées (multi-bruit, multi-patient).

Un autre paramètre influent est le chevauchement des patchs. Contrairement à certaines intuitions, les résultats indiquent que le retrait du chevauchement améliore le PSNR et diminue significativement la métrique de flou (BLUR), en particulier pour les grandes tailles de patchs. Cela peut s'expliquer par la réduction de la redondance entre les patchs et l'évitement d'un sur-lissage causé par des prédictions trop proches. Il est intéressant de noter que, dans notre configuration, aucun lissage de type moyenne n'a été appliqué entre les prédictions chevauchantes, ce qui suggère que le flou est directement lié à la capacité du modèle à traiter les variations locales.

L'analyse multi-patient a mis en évidence la robustesse du SVR face à la variabilité anatomique : les performances restent cohérentes d'un patient à l'autre. En revanche, la nature du bruit a un impact plus significatif. Le bruit gaussien s'est avéré plus difficile à corriger, avec une baisse notable du SSIM, tandis que le bruit rician a offert un bon compromis entre qualité perçue et fidélité statistique. Le bruit mixte a conduit aux meilleurs PSNR, mais au prix d'un flou légèrement plus élevé. Ces résultats suggèrent que le SVR s'adapte bien à des cas réels, tout en restant sensible aux propriétés du bruit.

Malgré ses bons résultats, le SVR présente plusieurs limites. Il reste sensible au bruit fort, avec une perte marquée de lisibilité sur certaines images. Sa performance dépend fortement de la taille des patchs, ce qui nécessite un ajustement manuel selon les cas. Le modèle fonctionne localement, sans exploiter le contexte global de l'image, et son apprentissage est coûteux en temps et en données. Ces éléments limitent son applicabilité directe à grande échelle en pratique clinique.

En comparaison, si le SVR offre de meilleures performances quantitatives et une stabilité d'entraînement remarquable, les GANs présentent un potentiel supérieur pour modéliser des variations globales complexes, à condition d'un entraînement suffisamment robuste. Le choix entre les deux dépend donc du compromis entre précision locale

et capacité de généralisation.

Dans le domaine médical, il est crucial d'avoir des images précises pour détecter des lésions sur ces régions et, dans le cadre de la recherche, d'obtenir des résultats fiables. Ainsi, la prochaine étape de ce projet pourrait consister à combiner les deux méthodes, GAN et SVR. Plusieurs combinaisons sont possibles, par exemple, en fournissant les résultats d'un modèle à l'autre ou en mélangeant complètement les deux approches. Des recherches existent déjà sur le mélange des deux approches, comme le montre l'étude de ASSAF et al. [2].

## 8 Lexique

- GANs : Generative Adversarial Networks.
- IVDs : Intervertebral Discs, soit Disques Intervertébraux, et sont des structures situées entre les vertèbres de la colonne vertébrale.
- FID : Frechet Inception Distance
- IS : Inception Score
- SSIM : Structural Similarity Index Measure
- PSNR : Peak Signal-to-Noise Ratio

## 9 Anexe

Lien GitHub : <https://github.com/athelc/PIMA>

## Références

- [1] Hamed ALQAHTANI, Manolya KAVAKLI-THORNE et Gulshan KUMAR. *An analysis of evaluation metrics of GANs*. Preprint. Postgraduate Student, Macquarie University; Associate Professor, Macquarie University; Assistant Professor, SBSSTC, Ferozepur. Déc. 2019. URL : <https://www.researchgate.net>.
- [2] A. M. ASSAF et al. “Improving Solar Radiation Forecasting Utilizing Data Augmentation Model Generative Adversarial Networks with Convolutional Support Vector Machine (GAN-CSVR)”. In : *Applied Sciences* 13.12768 (2023). DOI : 10.3390/app132312768. URL : <https://doi.org/10.3390/app132312768>.
- [3] Saurav BASU, Thomas FLETCHER et Ross WHITAKER. “Rician Noise Removal in Diffusion Tensor MRI”. In : (2012).
- [4] Maïtine BERGOUNIOUX. *Quelques méthodes de filtrage en Traitement d’Image*. Accessed: March 10, 2025. 2011.
- [5] Ali BORJI. “Pros and cons of GAN evaluation measures: New developments”. In : *Computer Vision and Image Understanding* 215 (2022), p. 103329. URL : <https://doi.org/10.1016/j.cviu.2022.103329>.
- [6] Pierrick COUPÉ et al. “Robust Rician noise estimation for MR images”. In : *Medical Image Analysis* 14.4 (2010), p. 483-493.
- [7] Ian GOODFELLOW et al. “Generative Adversarial Networks”. In : *Communications of the ACM* 63.11 (nov. 2020). Vol. 63, No. 11, November 2020, p. 139.

- [8] Hákon GUDBJARTSSON et Samuel PATZ. “The Rician Distribution of Noisy MRI Data”. In : *Magnetic Resonance in Medicine* (1995).
- [9] Sungwon LEE et al. “Artificial Intelligence in Spinal Imaging and Patient Care: A Review of Recent Advances”. In : *Neurospine* 21.2 (2024), p. 474-486. DOI : 10.14245/ns.2448388.194.
- [10] Jose V. MANJON et Pierrick COUPÉ. “MRI Denoising Using Deep Learning”. In : *International Workshop on Patch-based Techniques in Medical Imaging (MICCAI)*. Granada, Spain, sept. 2018, p. 12-19. DOI : 10.1007/978-3-030-00500-9\_2.
- [11] Laura MCCRACKIN et Shahram SHIRANI. “Strategic Image Denoising Using a Support Vector Machine with Seam Energy and Saliency Features”. In : *Proceedings of the International Conference on Image Processing*. McMaster University, Department of Electrical et Computer Engineering. 2015.
- [12] NEURO-MED. *Sténose du canal lombaire*. Accessed: March 10, 2025. 2025.
- [13] Daniel B. RUSSAKOFF et al. “Image Similarity Using Mutual Information of Regions”. In : *Department of Computer Science, Stanford University* (2020). Email: daniel.russakoff@cs.stanford.edu.
- [14] Tam SAKIRIN et Siddartha KUSUMA. “A Survey of Generative Artificial Intelligence Techniques”. In : *Babylonian Journal of Artificial Intelligence* 2023 (2023). Vol. 2023, 2023, p. 10-1.
- [15] Sud SUDIRMAN et al. *Lumbar Spine MRI Dataset*. Version 2. 2019. DOI : 10.17632/k57fr854j2.2.
- [16] Elena TSOMKO et Hyoung Joong KIM. “Efficient method of detecting globally blurry or sharp images”. In : *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*. IEEE. 2008, p. 171-174. DOI : 10.1109/WIAMIS.2008.28.
- [17] Analytics VIDHYA. *Image Classification using Machine Learning - Support Vector Machine (SVM)*. Accessed: March 10, 2025. 2020.
- [18] WIKIPÉDIA CONTRIBUTORS. *Imagerie par résonance magnétique*. Consulté le 10 mars 2025. 2024.
- [19] Zongsheng YUE et al. “Dual Adversarial Network: Toward Real-world Noise Removal and Noise Generation”. In : (2020). Xi'an Jiaotong University, Shaanxi, China; Hong Kong Polytechnic University, Hong Kong, China; DAMO Academy, Alibaba Group, Shenzhen, China; The Macau University of Science and Technology, Macau, China. Email: {zsyam, timmy.zhaoqian}@gmail.com, dymeng@mail.xjtu.edu.cn, cslzhang@comp.polyu.edu.hk.
- [20] Yanqiu ZENG et al. “Magnetic Resonance Image Denoising Algorithm Based on Cartoon, Texture, and Residual Parts”. In : *Computational Intelligence and Neuroscience* 2020 (avr. 2020), p. 1405647. DOI : 10.1155/2020/1405647.
- [21] G. D. ZHANG et al. “Image denoising based on support vector machine”. In : *2012 Spring Congress on Engineering and Technology*. IEEE, 2012, p. 1-4.