

Retrait du bruit de rétrodiffusion en imagerie ultrason à l'aide de trois réseaux d'apprentissage profond

Thomas Allard¹(1950175), Félix-Antoine Johnson¹(1960086) and Grégoire Marie(2360259)²

¹Polytechnique Montréal ²ÉTS Montréal

Abstract

Cette étude explore l'application de trois modèles d'apprentissage profond au problème de la réduction du bruit dans l'imagerie ultrasonore. Le *speckle noise*, une interférence granulaire qui se produit dans les images ultrasonores, dégrade considérablement la qualité de l'image, ce qui a un impact sur le processus de diagnostic. Les techniques de filtrage traditionnelles n'ont que partiellement réussi à atténuer ce bruit. Cependant, les récentes avancées dans le domaine de l'apprentissage profond offrent de nouvelles voies prometteuses pour améliorer efficacement la clarté de l'image. Cet article se concentre sur trois modèles spécifiques : le réseau de débruitage à autoencodeur à convolution dilatée (Di-Conv-AE-Net), le réseau en U à renormalisation par lots (Br-U-Net) et le réseau en U de débruitage (D-U-Net). Chaque modèle applique des stratégies architecturales et des méthodologies d'apprentissage uniques pour traiter le *speckle noise*, caractérisé par des distributions Gaussiennes additionnées et multipliées, dans un ensemble de données normalisé. Nous évaluons et comparons leurs performances à l'aide de deux mesures principales : le rapport signal-bruit maximal (PSNR) et la mesure de l'indice de similarité structurelle (SSIM), afin de déterminer leur efficacité dans la production d'images échographiques utilisables en clinique. Les résultats démontrent des améliorations significatives par rapport aux méthodes existantes, indiquant le potentiel des techniques d'apprentissage profond dans les scénarios d'imagerie médicale.

1 Introduction

L'imagerie ultrasonore joue un rôle essentiel dans le domaine médical puisqu'elle offre une méthode non-invasive de visualisation de certaines structures internes du corps. Par le biais d'ondes acoustiques à hautes fréquences qui réfléchissent sur les tissus, il est possible d'obtenir des images d'organes internes, de vaisseaux sanguins et de nerfs. Étant non-ionisante et pouvant fournir des images en temps réel, cette technique

est souvent préférée en obstétrique ou cardiologie. Le diagnostic via cette méthode est toutefois souvent affecté par un bruit de granularité ou *speckle noise* [1].

Ce *speckle Noise* provient des interférences entre les ondes hautes fréquences sortantes et celles entrantes ayant rebondis sur les tissus. La technologie actuelle d'imagerie ultrasonore fait en sorte qu'il y a présence de ce bruit granulaire sur l'ensemble des données collectées, rendant parfois difficile le diagnostic lorsqu'il y a obscurcissement de détails fins de l'image. Ce problème est particulièrement important pour certaines pathologies subtiles où il est nécessaire d'avoir des images ayant de hautes résolutions des détails [1].

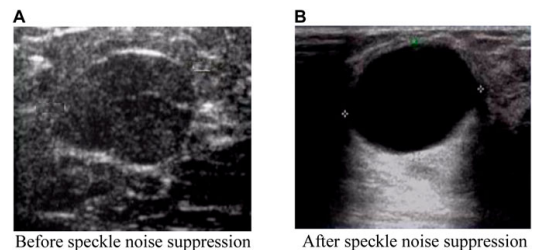


Figure 1: Exemple d'imagerie ultrason bruitée et dé-bruitée (*speckle noise*) [2].

Ces dernières années, plusieurs méthodes de traitement d'image ont été développées pour atténuer le *speckle noise*. Parmi elles, le cas du filtrage de diffusion anisotrope, introduit par Chauhan et Kaushik [3] s'est distingué. Plus récemment, les approches reposant sur l'apprentissage profond se sont révélées particulièrement prometteuses. Les réseaux auto-encodeurs débruitants, tels que ceux proposés par Perdios et al. [4], ont surpassé les performances des méthodes traditionnelles de filtrage, autrefois considérées comme la référence. Cette avancée est également observée avec l'emploi de réseaux antagonistes génératifs (GAN) qui contribuent à préserver les détails fins de l'image, tout en diminuant le bruit.

Ce projet se concentre sur l'application de cinq modèles d'apprentissage profond dédiés au débruitage d'images ultrasonores, développés initialement par Karaoglu et al. [5]. Trois de ces modèles ont été sélectionnés pour une étude approfondie: *Dilated Convolution Autoencoder Denoising Network* (Di-Conv-AE-Net), *BatchRenormalization U-Net* (Br-

U-Net) et *Denoising U-Shaped Network* (D-U-Net). Chaque modèle présente une stratégie et une architecture uniques, destinées à traiter des bruits de différentes intensités, caractérisés par une distribution de Rayleigh. L'objectif est d'évaluer et de comparer la performance de ces trois modèles en utilisant des images volontairement bruitées issues du dataset de la compétition *Kaggle* intitulée *Ultrasound Nerve Segmentation*.

2 Méthodologie

Cette section traite des manipulations effectuées sur l'ensemble de données ainsi que des modèles utilisés pour ce projet.

2.1 Pré-traitement des données

Normalisation des données

Afin de standardiser les entrées du modèle, les pixels de chaque image sont normalisés selon une distribution gaussienne avec une moyenne de 0.5 et un écart-type de 0.5. Bien que le choix de ces valeurs soit arbitraire, il a été dicté par les contraintes de temps de calcul; une optimisation plus poussée aurait pu être envisagée avec des ressources supplémentaires. De plus, la taille des images a été réduite de 580x420 pixels à 128x128 pixels afin de réduire l'empreinte mémoire et d'accélérer le traitement des données.

Ajout du bruit

Dans le prolongement des travaux antérieurs mentionnés dans l'article [5], le bruit appliqué aux images est généré suivant une distribution de Rayleigh avec un paramètre σ variant entre 0.1 et 0.75. Pour des raisons de performance computationnelle, nous avons opté pour une méthode de bruitage en deux temps : un bruit multiplicatif et un bruit additif, tous deux issus d'une distribution gaussienne centrée réduite. Cette approche permet de conserver une complexité calculatoire gérable et de maintenir la nature du bruit souhaité.

Découpage des données

La base de données est divisée en deux ensembles distincts : l'ensemble d'entraînement, qui comprend 5635 images, et l'ensemble de test, qui en compte 5508. L'ensemble d'entraînement est lui-même scindé en deux parties : 80% des données sont destinées à l'apprentissage du modèle, et les 20% restants servent à l'évaluation de la performance du modèle en validation, ceci sans rétro-propagation du gradient. Le découpage entre les données d'entraînement et de validation est effectué aléatoirement pour permettre la réalisation de multiples sessions d'entraînement et assurer la stabilité des performances du modèle sur différentes subdivisions des données.

Mélange des données

Pour introduire une variabilité et améliorer la robustesse du modèle, l'ordre de présentation des images lors de chaque époque d'entraînement est aléatoirement modifié. Cette méthode de mélange par mini-batch permet d'optimiser la descente de gradient. Il est à noter que les ensembles de validation et de test ne requièrent pas un tel mélange, puisqu'ils sont utilisés uniquement pour évaluer la capacité de généralisation du modèle en inférence.

2.2 Modèles

Dilated convolution autoencoder denoising network (Di-Conv-AE-Net)

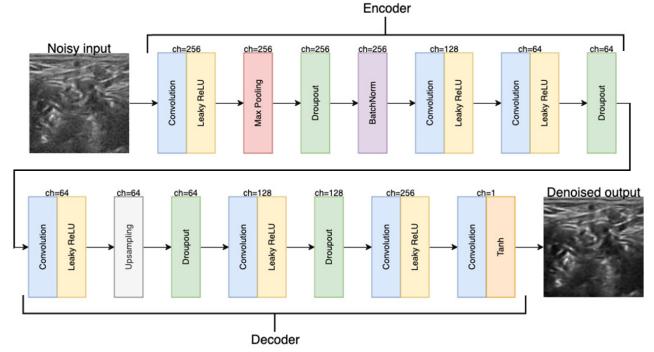


Figure 2: Architecture du modèle Di-Conv-AE-Net

Ce premier modèle exploité se base sur le concept de convolution diluée permettant une expansion de la taille de l'image en prévenant la perte de données. Cette convolution diluée peut être définie sous la formule suivante:

$$(F_l * k)(p) = \sum_{s+lt=p} F(s)k(t) \quad (1)$$

Cette couche permet ainsi d'éviter le *max pooling* qui engendre une perte de résolution de l'image malgré les bénéfices de réduction de surapprentissage et de *feature extraction*. La résolution de l'image étant fort importante dans ce contexte d'application, la convolution diluée applique un élargissement avec des 'trous' assurant qu'il n'y a pas de perte de résolutions. Malgré l'augmentation du nombre de paramètres du filtre, le nombre d'opérations effectuées en convolution demeure constant.

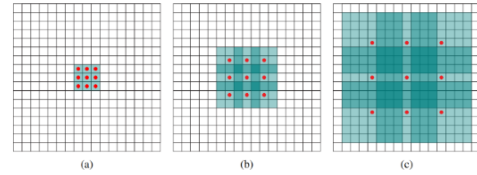


Figure 3: Exemple du fonctionnement d'un champ récepteur dilué.

Denoising U-Shaped Net (D-U-Net)

Un *U-Net* est un réseau composé de deux branches. La branche de gauche est la voie de contraction et la branche de droite est la branche d'expansion. Dans la branche de contraction, on fait à répétition un cycle de Maxpooling, de Dropout, de convolution 3x3, de BatchNorm et de ReLu. À chaque étape de la branche, on augmente par un facteur de 2 le nombre de *channels*. On attend par la suite le *bottleneck* qui applique les mêmes transformations que la branche de contraction en plus d'ajouter une convolution transposée. Dans la voie d'expansion, on prend en entrée la concaténation d'un bloc de la branche de contraction et du bloc précédent. On

fait à répétition un cycle de Dropout, de convolution 3x3, de BatchNorm, de ReLU et d'un second BatchNorm. On répète ce cycle pour une quantité qui est égale au nombre de cycles de la branche de contraction avant d'appliquer une convolution finale afin d'avoir une valeur de *channel* de 1 [6].

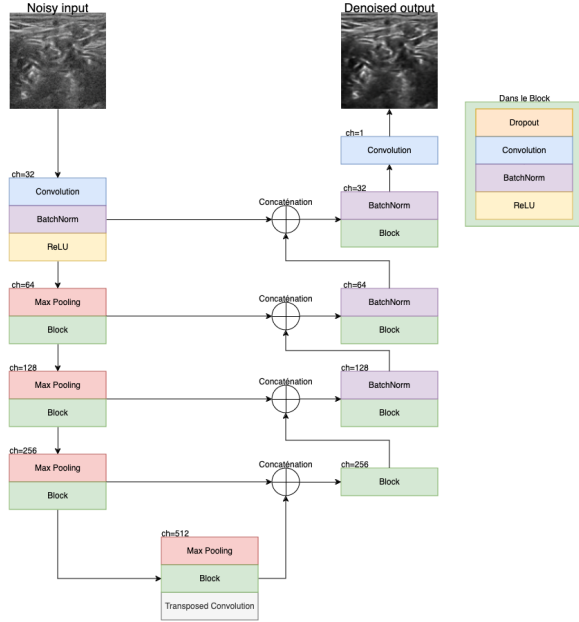


Figure 4: Modèle Denoising U-Shaped Net implémenté.

BatchRenormalization U-Net (Br-U-Net)

Le BatchRenormalization U-Net reprend l'architecture classique d'un *U-Net* présentée en D-U-Net, mais y change les couche ReLU pour du Leaky ReLU (voir pourquoi) et les couches *BatchNorm* en couches *BatchRenorm*.

Ces nouvelles couches de renormalisation diffèrent des couches généralement utilisées à cet effet par leur fonctionnement basé sur une normalisation non-centrée tel que décrit ci-dessous:

$$\frac{x_i - \mu}{\sigma} = \frac{x_i - \mu_B}{\sigma_B} \cdot \mathbf{r} + d \quad (2)$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sigma_B} \cdot \mathbf{r} + d \quad (3)$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \quad (4)$$

La Batch Renormalization est une variante avancée de la normalisation par lots, conçue pour atténuer le phénomène connu sous le nom de *Reduced Internal Covariate Shift* (ICS), qui se réfère aux changements des distributions des entrées de réseau neuronal au cours de l'entraînement. En contrôlant efficacement ce décalage interne, la Batch Renormalization permet d'utiliser des tailles de mini-lots plus petites, ce qui est particulièrement bénéfique dans des contextes où les ressources de calcul ou la quantité de données sont limitées.[7]

En plus, cette technique facilite l'emploi de taux d'apprentissage plus élevés et de taux de dropout plus bas sans compromettre la stabilité de l'apprentissage. Ces

ajustements contribuent directement à accélérer la vitesse d'entraînement du modèle tout en améliorant sa stabilité globale. Enfin, un autre avantage notable de la Batch Renormalization réside dans sa capacité à réduire la dépendance du modèle à son initialisation. L'architecture du modèle implémenté est montré à la figure 5

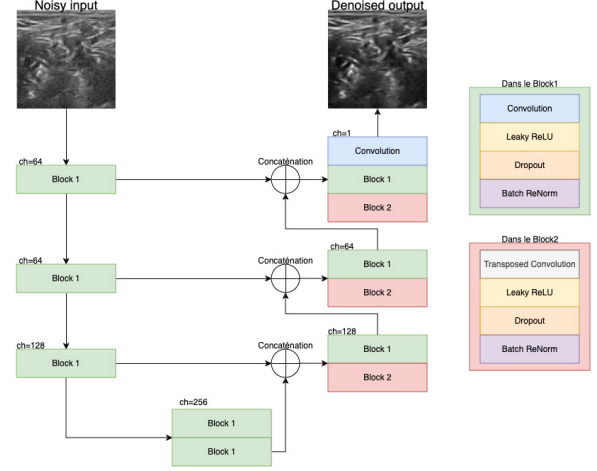


Figure 5: Modèle BatchRenormalization U-Shaped Net implémenté.

3 Résultats expérimentaux

L'ensemble des opérations décrites dans cette section ont été exécutées sur Google Colab Pro pour y exploiter un GPU de type L4. Cela a permis aussi d'accéder à un environnement de mémoire RAM élevée facilitant l'exécution de nos modèles. Les hyperparamètres utilisés pour chacun de ces modèles sont présentés dans le Tableau 1 en annexe. La nature des données utilisées a été décrite en début de section 2.

Explications des métriques

La première métrique d'évaluation d'importance exploitée est le *Peak Signal to Noise Ratio* (PSNR). Il s'agit du rapport entre la puissance maximale d'un signal et la puissance du bruit affectant. Cette métrique, que l'on tente de maximiser, témoigne de la clarté et de l'intégrité des images reproduites suite au processus de compression et de restauration.[8]

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \quad (5)$$

La seconde métrique d'évaluation, le *Structural Similarity Index Measure* (SSIM), adresse plutôt la qualité de l'image en terme de changements perçus dans sa structure plutôt que sur un niveau d'erreur pur. Cette métrique s'apparente davantage à une évaluation de la perception visuelle humaine, ce qui s'avère essentiel puisque l'objectif du débruitage est de faciliter le diagnostic par le professionnel de la santé. Une SSIM qui tend vers la valeur optimale de 1 témoigne d'une qualité d'expression des structures fines et des textures de l'imagerie ultrasonore.[5]

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (6)$$

Performance des modèles

Tout d’abord, les métriques Loss, PSNR et SSIM ont toutes trois été mesurées auprès des trois modèles neuronaux présentés à la section 2. Différents niveaux de bruits de multiplicateurs [1, 5, 10] ont été utilisés sur ces modèles. L’ensemble des expérimentations ont été effectuées pour une quantité fixe de 10 *epochs* pour chacun des modèles. L’allure des graphes de métrique de *loss* en fonction des *epochs* s’apparente à ce qui est présenté en figure 6 et ce, pour chaque modèle, peu importe l’instance. On remarque qu’à la première *epoch*, on obtient une perte sur l’ensemble d’entraînement plutôt élevée relativement aux autres valeurs. Pour les *epochs* suivantes, la perte sur l’ensemble d’entraînement diminue et se stabilise. La courbe de la perte sur l’ensemble de validation semble constante pour les 10 *epochs* et est inférieure que la courbe de la perte sur l’ensemble d’entraînement pour chaque *epoch*.

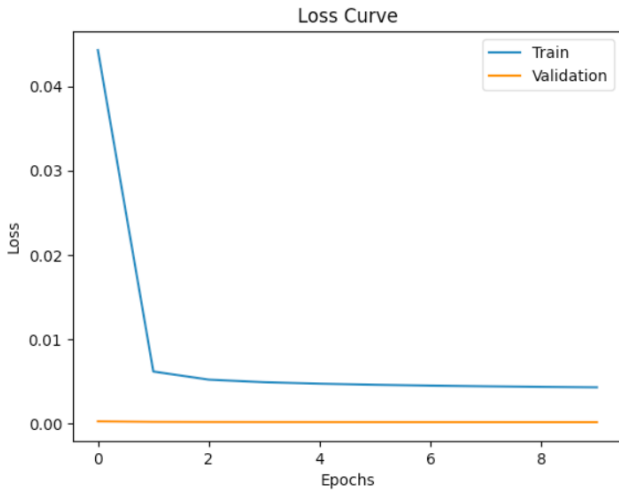


Figure 6: Métrique Loss en fonction du nombre d’epochs pour le modèle D-U-Net avec multiplicateur de bruit x1.

L’ensemble des résultats d’évaluations des trois modèles aux trois niveaux de bruits distincts est présenté au tableau 2 en Annexe. Parmi les trois modèles, on remarque que celui qui a la meilleure performance au niveau des pertes est le D-U-Net. On remarque que cette implémentation possède la plus faible perte sur l’ensemble d’entraînement et sur l’ensemble de test pour les trois niveaux de bruits. Pour l’évaluation de la performance des modèles avec le PSNR, le Br-U-Net a le meilleur ratio pour un facteur de bruit de 1, alors que le Di-Conv-AE-Net a le meilleur ratio pour un facteur de bruit de 5 et que le D-U-Net a le meilleur ratio pour un facteur de bruit de 10. Pour l’évaluation de la performance avec SSIM, les trois modèles avaient un résultat comparable pour chaque niveau de bruit, mais le modèle du D-U-Net avait un indice de similarité plus élevé que ses deux contreparties pour chacun des trois niveaux de *speckle noise*. Lorsque vient le temps de comparer le temps pris par chaque modèle pour l’entraînement et la validation sur 10 *epochs*, on remarque que D-U-Net était beaucoup plus rapide avec un temps total de 3:38, alors que les deux autres modèles ont eu besoin de

plus de 10 minutes pour effectuer les mêmes opérations. Le modèle le plus lent sur l’ensemble des instances s’avère être le Br-U-Net.

4 Discussion

–Aborder les hauts résultats de chaque modèle pour l’ensemble de bruit. Les fluctuations en performance proviennent d’hyperparamétrisation puisqu’elles sont majoritairement non significatives

–La performance des modèles n’est que légèrement altérée par l’augmentation de la magnitude du bruit de multiplication. Cette diminution est à la limite de la significativité, en comparaison avec la magnitude de l’augmentation du bruitage. Cela signifie que les modèles se montrent robustes face à des bruits proéminents, et est capable d’apprendre à reconstruire des images dans des conditions hautement dégradées.

On dénote d’ailleurs que le modèle le plus lent pour chaque *epoch* est le Br-U-Net. Tel qu’indiqué lors de la méthodologie, celui-ci est théoriquement, une amélioration du Denoising U-Shaped Net permettant un entraînement plus rapide. Les résultats contraires à ceux attendus peuvent être dûs au fait que l’implémentation des couches BatchRenorm dans le modèle ne sont pas issues de PyTorch, mais plutôt par une utilisation locale de classes trouvées sur un dépôt Git. Ainsi, les calculs à chaque implication d’un bloc de BatchRenorm ne serait pas optimal et engendrerait des temps de calculs plus grands[9]

–La performance en ensemble de validation est meilleure que celle de l’ensemble d’entraînement, dès l’epoch 0. Cela peut s’expliquer par le fait que l’adaptation des poids est réalisée par mini-batch sur l’ensemble d’entraînement, avant l’évaluation de l’ensemble de validation. En revanche, il est possible de noter que les courbes fonction de coût (*loss*) d’ensembles d’entraînement et de validation partagent la même tendance baissière, jusqu’à l’epoch 4 en moyenne. À partir de celle-ci, la fonction de coût évaluée sur l’ensemble de validation stagne, mais ne remonte pas, ce qui dénote une absence de phénomène de sur-apprentissage.

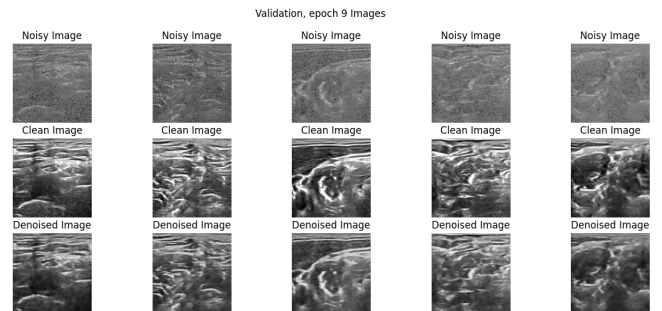


Figure 7: Ensemble d’images initiales, bruitées avec un facteur de multiplication x10 et débruitées de l’ensemble de validation pour le modèle Di-Conv-AE-Net.

–Aborder la clarté des images débruitées de façon qualitative à partir des images obtenues pour le groupe test en figure 7

–Points d’améliorations et considérations de changement des modèles

5 Analyse critique

Bien que les résultats obtenus soient positifs, comme décrit dans la section précédente du rapport, l’exécution du projet comporte certaines imperfections. L’intégration de la *BatchRenormalization* a posé des défis, notamment l’absence d’implémentation native dans PyTorch. Cela a conduit à une implémentation manuelle à partir d’un dépôt Git [9]. Cette approche manuelle, bien que fonctionnelle, n’est pas optimisée pour la vitesse de traitement, ce qui a impacté le processus d’apprentissage et l’efficacité globale du modèle.



Figure 8: Imagerie ultrasonore bruitée selon une distribution de Rayleigh $\sigma = [0, 0.25, 0.5, 0.75]$ [5].

De plus, les performances remarquables observées en termes de réduction des pertes pourraient indiquer un surapprentissage du modèle sur les niveaux de bruit spécifiques utilisés. Étant donné que le bruit n’est pas généré selon une distribution de Rayleigh, mais plutôt sous forme multiplicative et additive, il est possible que le modèle excelle uniquement sur le type de bruit testé et pourrait ne pas être aussi performant avec d’autres formes de bruit. Une visualisation du bruitage selon cette distribution est fournie en figure 8. Il serait pertinent pour les recherches futures d’intégrer divers types de bruits aux données, afin d’observer la capacité d’adaptation et de généralisation des modèles.

En conclusion, l’article qui sert de fondement à ce projet présente plusieurs incohérences qui pourraient avoir influencé négativement l’implémentation de nos réseaux neuronaux. Notamment, les hyperparamètres décrits dans le rapport ne concordent pas toujours avec les illustrations fournies, et les diagrammes des sur lesquels se basaient nos premières implémentations comportaient des erreurs. Par exemple, comme le montrent la figure 9 et la figure 10 de l’annexe, l’architecture originale inclut du max pooling dans la phase d’expansion du U-net, ce qui est atypique. Or, le tableau des hyperparamètres indique la présence de seulement quatre unités de max pooling, suggérant l’absence de pooling dans la phase d’expansion. En outre, pour ce même modèle, le tableau suggère l’utilisation de la fonction d’activation sigmoïde, alors que le texte et les figures décrivent l’utilisation de ReLU. Ces divergences ont conduit à réaliser des interprétations dans nos implémentations des modèles, qui pourraient influencer négativement la performance des réseaux développés. Cela souligne l’importance d’une validation rigoureuse et d’une adaptation prudente des modèles issus de la littérature scientifique.

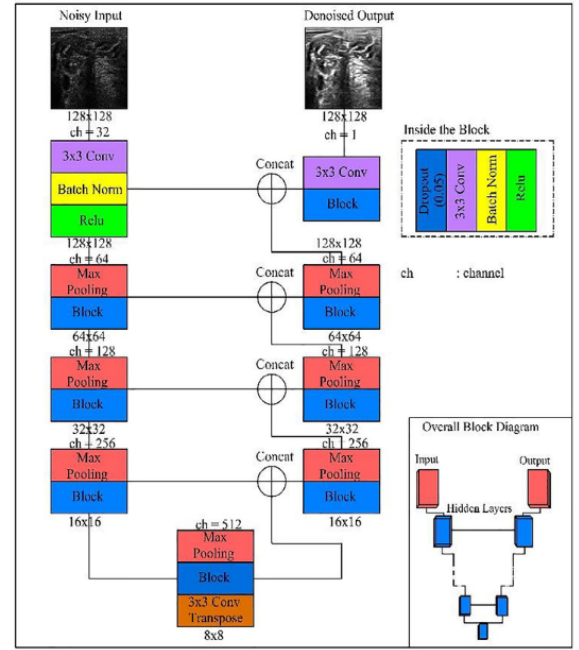


Figure 9: Modèle Denoising U-shaped Net erroné présenté dans l’article sur lequel se base le projet. [5]

6 Conclusion

En guise de conclusion, cette reproduction du travail de recherche de Karaoglu et al. a mis en lumière le potentiel de l’apprentissage profond dans l’amélioration significative du dé-bruitage de *Speckle Noise* en imagerie ultrasonore. Les trois architectures explorées (Di-Conv-AE-Net, D-U-Net et Br-U-Net) ont toutes démontré de fortes performances en réduction *speckle noise*, comme en témoignent les scores de Loss, de PSNR et de SSIM obtenus. Le modèle D-U-Net, en particulier, a offert un équilibre remarquable entre performance de dé-bruitage et rapidité d’exécution, ce qui souligne son applicabilité dans des contextes cliniques où le temps est un facteur critique.

Les expériences réalisées ont également ouvert des perspectives d’amélioration, notamment en ce qui concerne la Batch Renormalization. Une intégration plus optimisée pourrait augmenter la vitesse d’exécution tout en préservant, voire en améliorant, la performance des modèles. Cela en est de même pour l’implémentation de distributions de bruits plus complexes que ce qui a été utilisé. Il serait intéressant de voir la capacité de ces modèles de généraliser le dé-bruitage dans des applications en temps réelles tel que demandé lors des applications d’imagerie ultrasonore en hôpital.

References

- [1] O. V. Michailovich and A. Tannenbaum, “Despeckling of medical ultrasound images,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 1, pp. 64–78, 2006.
- [2] X. Li, Y. Wang, Y. Zhao, and Y. Wei, “Fast speckle noise suppression algorithm in breast ultra-

- sound image using three-dimensional deep learning,” *Frontiers in Physiology*, vol. 13, 2022. [Online]. Available: <https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2022.880966>
- [3] K. Sumathi, K. R. Anandh, V. Mahesh, and S. Ramakrishnan, “Anisotropic diffusion filter based edge enhancement for the segmentation of carotid intima-media layer in ultrasound images using variational level set method without re-initialisation,” pp. 238–241, 2014.
 - [4] D. Perdios, A. Besson, M. Arditi, and J.-P. Thiran, “A deep learning approach to ultrasound image recovery,” pp. 1–1, 2017.
 - [5] O. Karaoğlu, H. Bilge, and Uluer, “Removal of speckle noises from ultrasound images using five different deep learning networks,” *Engineering Science and Technology, an International Journal*, vol. 29, no. 101030, p. 101030, 2022.
 - [6] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.
 - [7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015.
 - [8] A. Hore and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.
 - [9] ludvb, “Batch renormalization - pytorch implementation,” <https://github.com/ludvb/batchrenorm>, 2023, available at: <https://github.com/ludvb/batchrenorm>.

A Annexe

Table 1: Comparaison des hyperparamètres des architectures de modèles.

Paramètre	Di-Conv-AE-Net	D-U-Net	Br-U-Net
Learning Rate	0.01	0.005	0.0001
Facteurs d'amplification du bruit de multiplication	[1,5,10]	[1,5,10]	[1,5,10]
Batch size	32	32	32
Max Pool stride	2	2	None
Conv2d dilatation	2	2	2
Conv2d kernel size	3	3	3
Conv2d stride	1	1	1
Conv2d padding	2	1	2
Dropout	0.2	0.1	0.05
ConvTranspose2d dilatation	None	2	2
ConvTranspose2d kernel size	4	2	2
ConvTranspose2d stride	2	1	2
ConvTranspose2d padding	1	None	None
ReLU	None	0.1	None
LeakyReLU	None	None	0.1
BatchRenorm eps	None	None	0.001
BatchRenorm momentum	None	None	0.01

Table 2: Résultats expérimentaux pour différents facteurs de bruit sur les modèles Di-Conv-AE-Net, D-U-Net, et Br-U-Net.

Noise factor	Di-Conv-AE-Net			D-U-Net			Br-U-Net		
	1	5	10	1	5	10	1	5	10
Train Loss	0.0053	0.0102	0.0164	0.0042	0.0090	0.0157	0.0043	0.0102	0.0160
Validation Loss	0.0001	0.0003	0.0005	0.0002	0.0003	0.0004	0.0001	0.0003	0.0005
PSNR	24.5382	22.1032	20.1033	24.9299	21.8953	20.5506	25.0124	21.0654	19.2527
SSIM	0.999927	0.999883	0.999797	0.999944	0.999889	0.999884	0.999944	0.999874	0.999797
Total run time	10:36	10:35	10:35	3:38	3:38	3:38	13:13	13:13	13:14

Table 1
Deep learning network hyperparameters.

Hyperparameters	Di-Conv-AE-Net	DGAN-NET		D-U-NET
		Generator	Discriminator	
Optimizer	Adamax	Adam	Adam	Adamax
Validation Rate	10%	-	-	10%
Dropout Rate	20%	50%	-	10%
Dilated Convolution	Used	-	-	-
Number of Max-Pooling Layers	1	-	-	4
Activation Function	Tanh	Tanh	Leaky ReLU	Sigmoid
Number of Layers	21	34	11	57
Learning Rate	0.0001	0.0001	0.0001	0.0001
Mini-batch size	20	20	20	32

Figure 10: Extrait des hyperparamètres contradictoires de l'article sur lesquels se base le projet