



Projet débruitage - Compte rendu semaine du 24/11

Viguier Killian

Wang Xihao

M2 IMAGINE
Faculté des Sciences
Université de Montpellier

30 Novembre 2025



Résumé

Cette semaine, nous avons implémenté un GAN utilisant notre VAE comme générateur et un discriminateur permettant d'améliorer nos résultats. De plus, nous avons continué la piste du bruit par expansion dynamique en recréant le même type de bruit et en entraînant notre modèle dessus.

1 Le GAN et ses différentes approches

Un GAN est une architecture de réseau de neurone particulière qui combine un générateur et un discriminateur. Le but du générateur est de générer une image débruitée et celui du discriminateur est de deviner si l'image résultante est l'image originale ou une image débruitée. Si le discriminateur devine bien la classe de l'image, le générateur modifie ses paramètres via une rétropropagation.

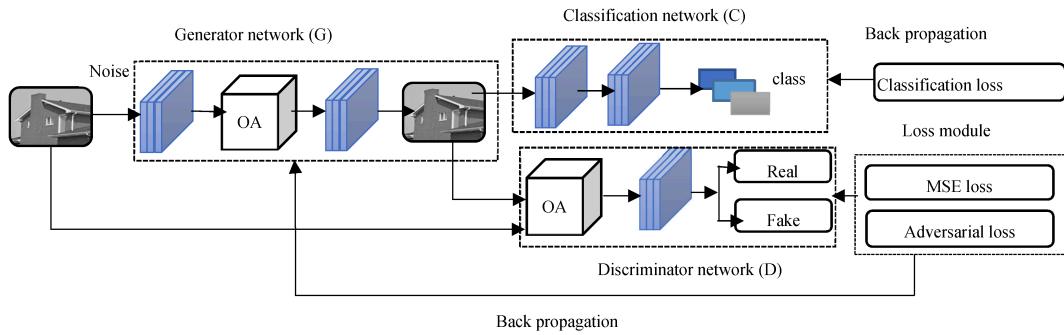
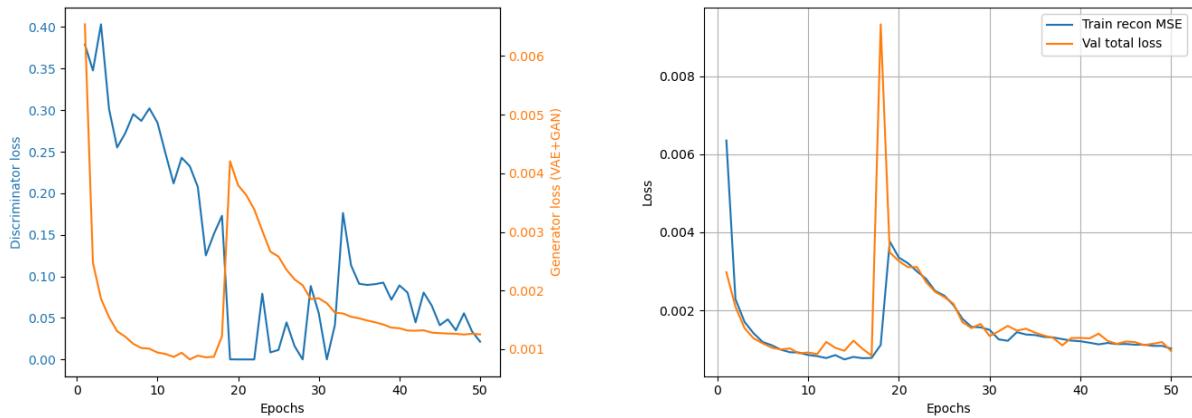


Fig. 1. – Schéma d'une architecture de GAN

Dans notre cas, le VAE utilisé à l'aide d'U-Net correspond au générateur et le discriminateur est implémenté par un classifieur qui détermine si une image est débruitée ou originale.

1.1 Entraînement du GAN et du VAE simultanément

Dans cette approche, le GAN essaie de distinguer les images *clean* et les images *recon*. Concrètement, on utilise d'abord les premières images générées par le VAE, ainsi que leurs images propres correspondantes, pour entraîner le GAN sur les *recon/clean* images. Ensuite, la loss_adv renvoyée par le discriminateur est ajoutée dans la loss_total du VAE.



(a) Loss_D (bleu) contre Loss_G (orange)

(b) Loss total : GAN + VAE

Fig. 2. – La confrontation entre le discriminateur et le générateur, l'époque 20 marquant le début du collapse.

Le problème est que le GAN apprend trop vite, ce qui casse complètement l'équilibre entre les deux réseaux: on finit par avoir un *collapse*. Les images de sorties sont alors de nouveau bruitées et de mauvaise qualité.

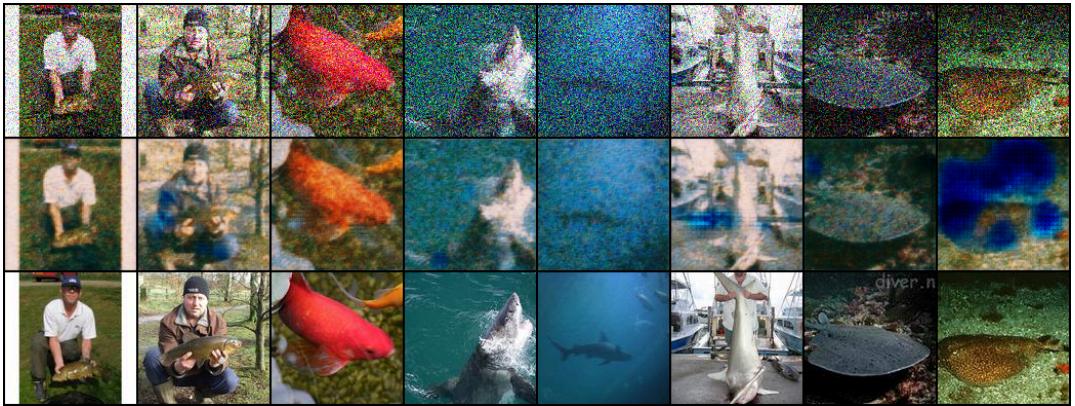
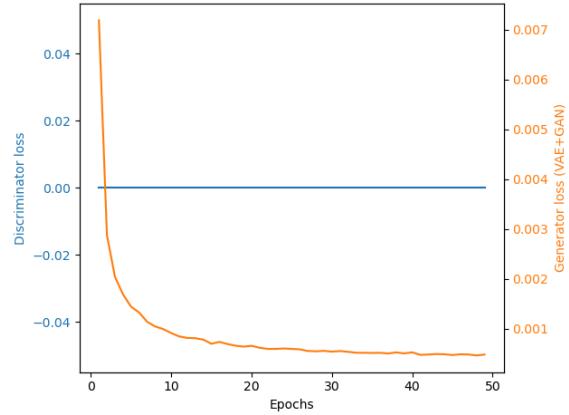


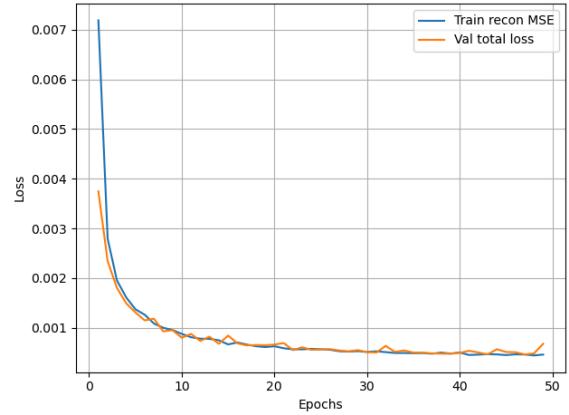
Fig. 3. – Résultat après le collapse

1.2 Entraînement conjoint, mais en changeant le rythme GAN/VAE

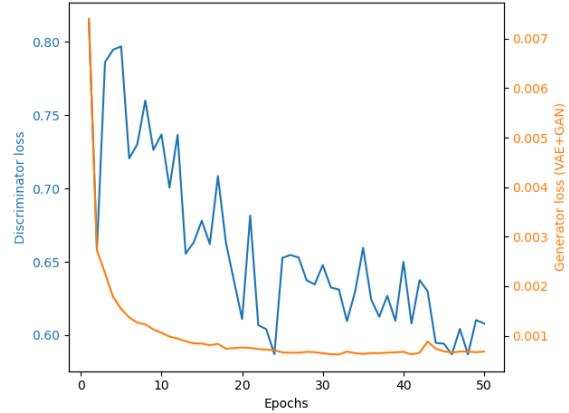
L'idée ici est la même, sauf qu'au lieu d'alterner Générateur et Discriminateur (paterne G -> D -> G -> D ...) à chaque batch, on entraîne le générateur plusieurs fois, puis seulement après on entraîne le discriminateur, et ainsi de suite (paterne G -> G -> ... -> D -> G -> G -> ...).



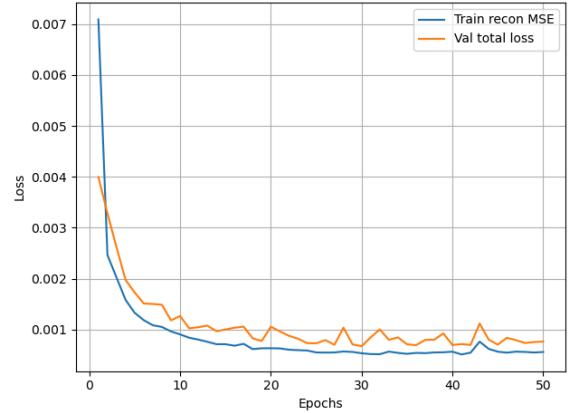
(a) Loss_D (désactivé) contre Loss_G (orange)



(b) Loss total : VAE



(c) Loss_D (bleu) contre Loss_G (orange)



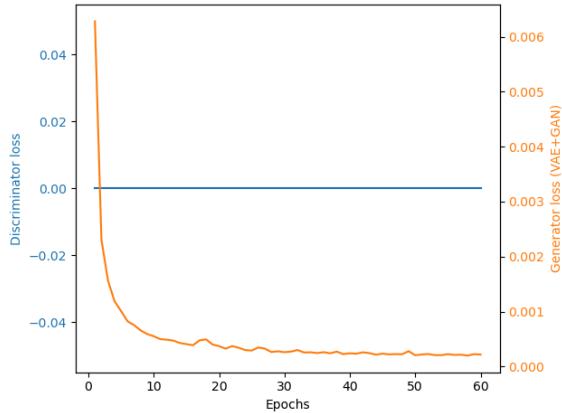
(d) Loss total : VAE + GAN

Fig. 4. – La comparaison de courbe entre VAE pur et VAE + GAN.

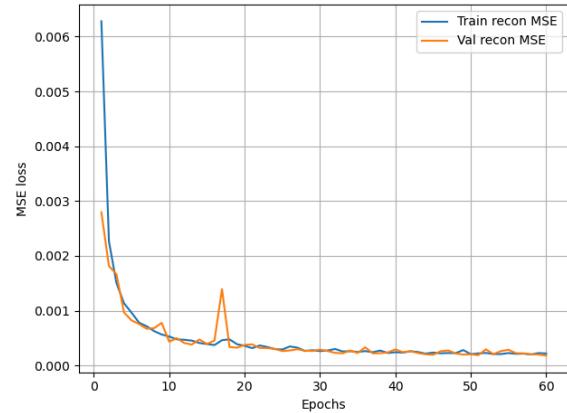
Résultat: on réussit à garder un bon équilibre entre D et G, il n'y a plus d'effondrement. Mais la qualité des images n'augmente pas du tout. Au contraire, elle devient même moins bonne que celle d'un VAE unique. Autrement dit, ce schéma VAE+GAN ne sert toujours à rien dans notre cas.

1.3 Préentraînement du VAE, puis ajouter le GAN ensuite

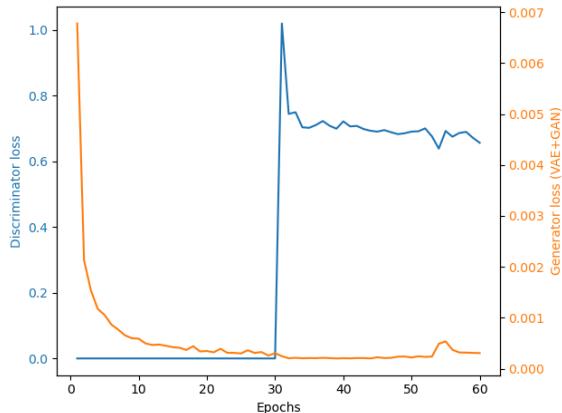
Ici, on laisse d'abord le VAE s'entraîner seul pendant plusieurs epochs, histoire d'obtenir une qualité d'image de base relativement stable. Ensuite, on ajoute le discriminateur pour tenter d'améliorer les détails (*fine-tuning*).



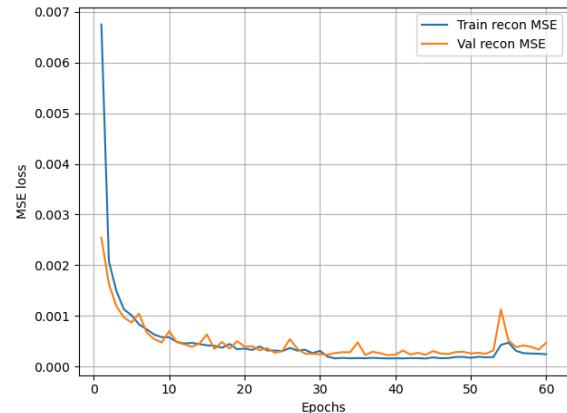
(a) Loss_D (désactivé) contre Loss_G (orange)



(b) Loss total : VAE

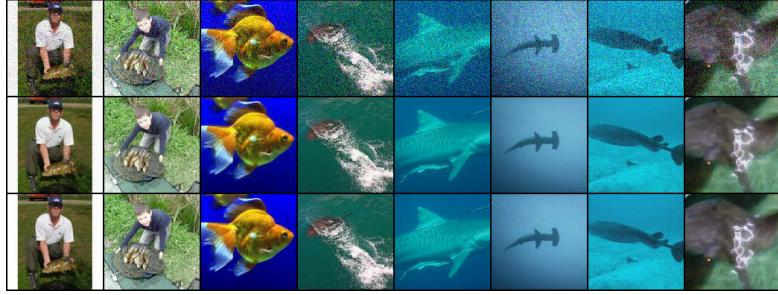


(c) Loss_D (bleu) contre Loss_G (orange)

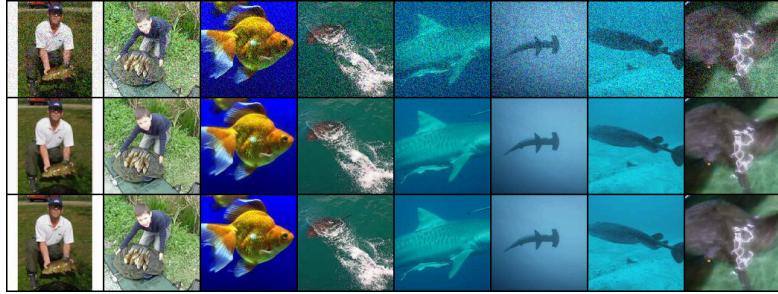


(d) Loss total : VAE + GAN

Fig. 5. – La comparaison entre VAE pur et VAE + GAN.



(a) Le meilleur résultats du VEA pur : PSNR ~37dB



(b) Le meilleur résultats du VEA + GAN : PSNR~37dB

Fig. 6. – La comparaison de résultats entre VAE pur et VAE + GAN.

Bien que les résultats aient été meilleurs que ceux obtenus précédemment, la valeur loss_val s'est également rapprochée de la valeur loss_train. Mais en comparant avec la courbe de loss de VAE pur, on voit que cette méthode dégrade aussi la qualité finale: l'ajout du GAN tire systématiquement les résultats vers le bas.

1.4 Pair-based GAN

Pour conclure ces trois échecs, le problème ne vient peut-être ni du rythme d'entraînement du GAN, ni du moment où on l'introduit, mais probablement du type de discriminateur utilisé.

En approfondissant nos recherches, nous sommes tombés sur le « *Dual-Path Adversarial Denoising Network Based on UNet* ». Dans cet article, le discriminateur ne compare pas clean vs recon, mais (noisy, clean) contre (noisy, recon).

Les auteurs expliquent que, pour la tâche de débruitage, le modèle doit apprendre la différence entre bruit et image propre, pas simplement reconstruire l'image. C'est ce qui lui permet, pour un bruit donné, de produire une image débruitée plus fidèle.

Autre point important : la plupart des GAN appliqués au débruitage ne sont pas utilisés avec une architecture U-Net. Or, un U-Net produit déjà des reconstructions très proches du clean. Si on utilise un GAN « classique » par-dessus, on risque au contraire de dégrader l'image (artefacts, couleurs incohérentes, etc.).

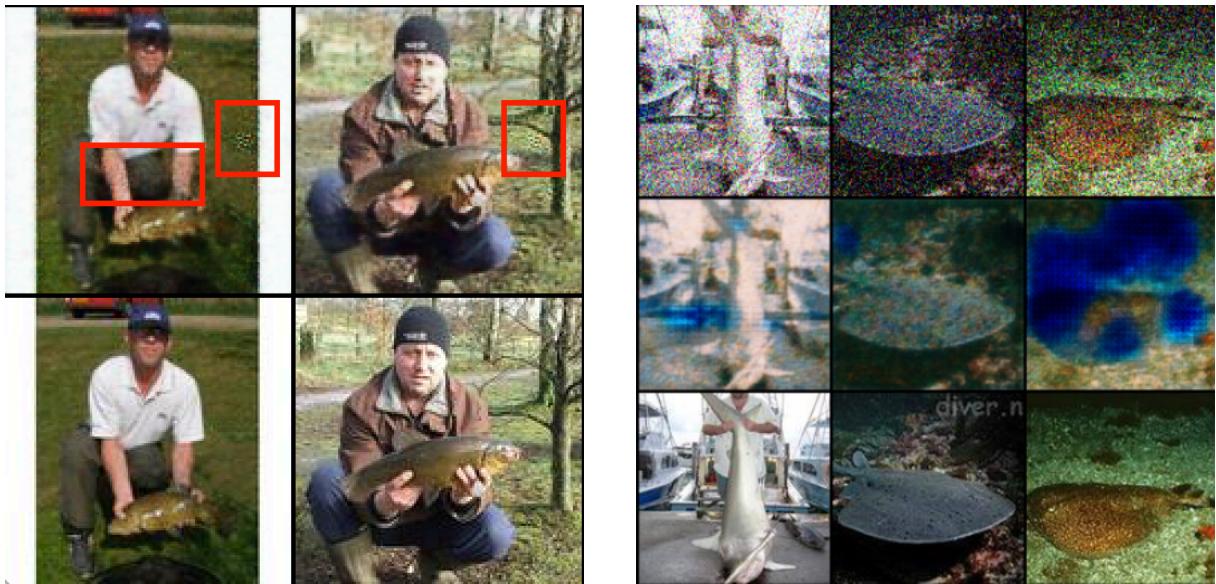


Fig. 7. – Artefacts avec des couleurs incohérentes et du flou.

Nous testons donc actuellement ce modèle pair-based (*noisy, clean*) / (*noisy, recon*), en ajustant les paramètres, en espérant obtenir un vrai gain de qualité.

1.5 PSNR

Le PSNR des VAE purs utilisant U-Net fluctue entre 34 et 37 dB pour les images débruitées, en fonction de l'intensité et du type de bruit. Ici, nous adoptons 37 dB comme référence.

- La méthode 2.1 a vu le PSNR tomber en dessous de 30 après le collapse, pour atteindre ensuite un pic à 29 dB.
- La méthode 2.2 a atteint un PSNR optimal de 33 dB.
- La méthode 2.3 a le PSNR optimal qui était presque le même que celui du VAE pur, mais ça n'était pas vraiment une amélioration.
- La méthode 2.4 est encore en test...

2 Bruit d'expansion dynamique

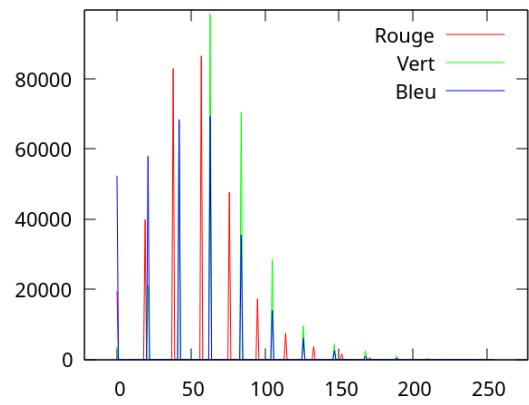
Pour aller plus loin du coté des bases de données d'images bruitées, nous avons fait en sorte de pouvoir reproduire le bruit obtenu après une expansion dynamique à partir d'une photo sombre.

2.1 Reproduction du type de bruit

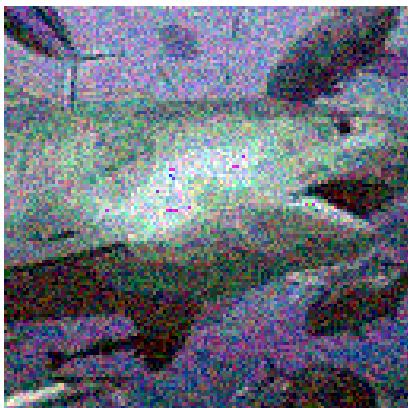
Pour cela, un algorithme simple est de diminuer la luminosité de l'image (pour chaque valeur de pixel, on la divise par une constante L) puis on applique un léger bruit sur ce résultat, qu'il soit gaussien ou aléatoire. Enfin, il suffit d'appliquer l'expansion dynamique et nous retrouvons l'effet désiré.



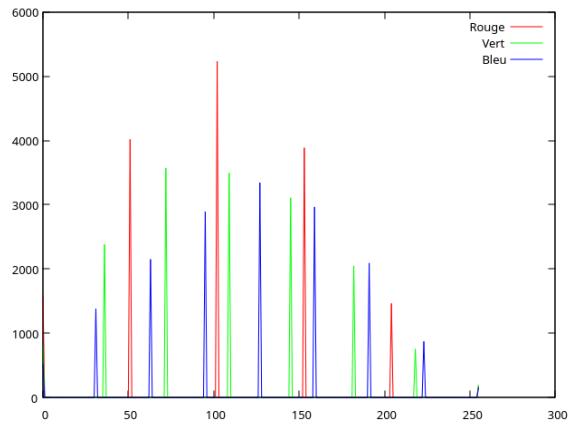
(a) Photo sombre avec expansion dynamique



(b) Histogramme correspondant à la prise



(c) Image avec bruitage d'ex. dyn. simulé



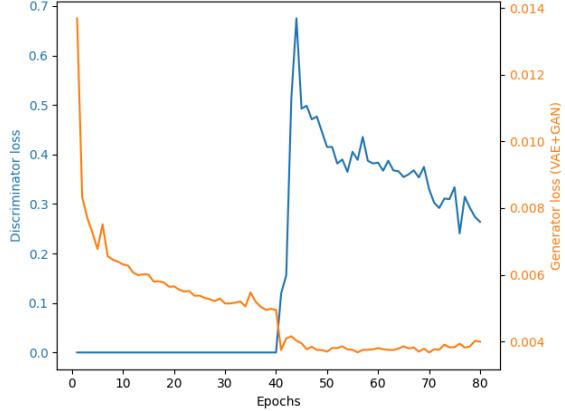
(d) Histogramme correspondant au requin

Fig. 8. – Reproduction d'un bruit par expansion dynamique

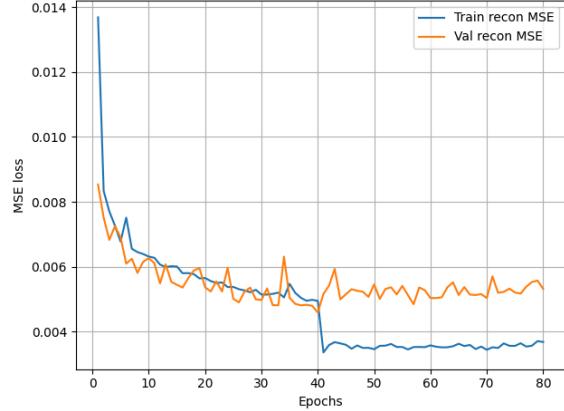
Grâce à cela, nous pouvons désormais faire en sorte que notre modèle puisse apprendre ce nouveau type de bruit et peut-être arriver à débruiter la photo de la multiprise.

2.2 Apprentissage du bruit

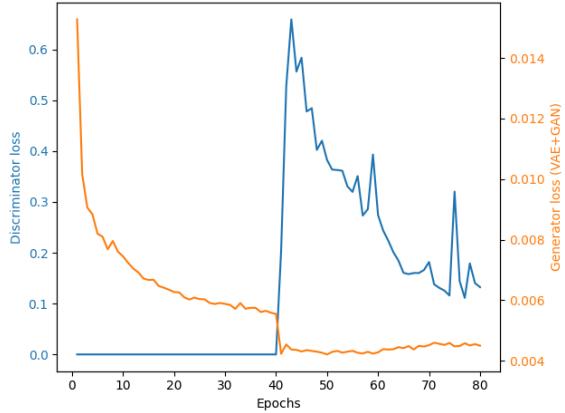
Pour l'apprentissage, nous utilisons le GAN évoqué dans la partie 1.3.



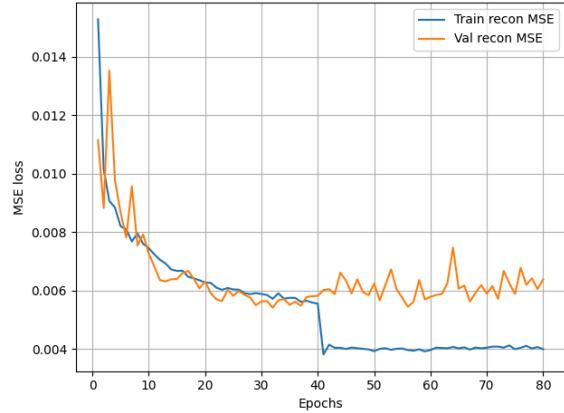
(a) loss_D (bleu) contre loss_G (orange)



(b) loss total : VAE



(c) loss_D (bleu) contre loss_G (orange)

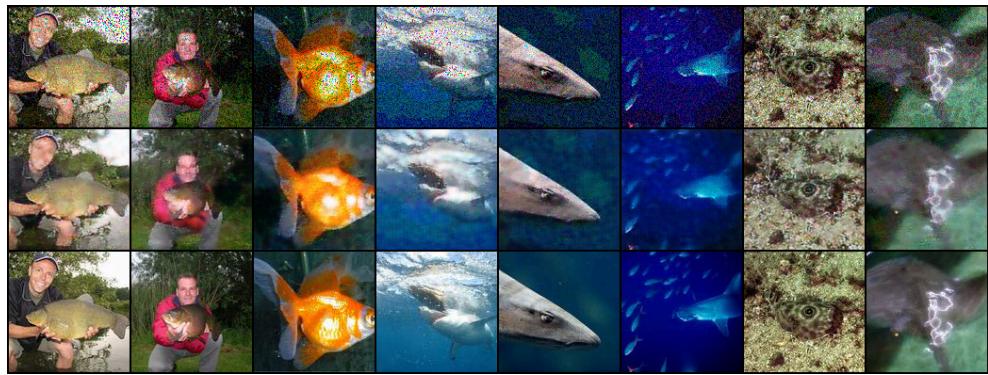


(d) loss total : VAE

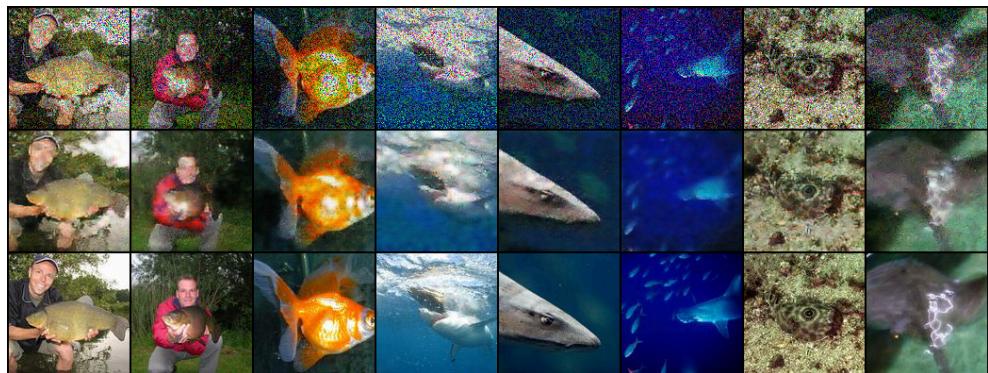
Fig. 9. – En haut : Résultats avec bruit faible par expansion dyn.

En bas : bruit fort par expansion dyn.

Les résultats tournent autour de 23-24 dB pour le PSNR du bruit faible, et aux alentours de 21-22 dB pour le bruit fort. Les PSNR sont faibles car un bruit par expansion dynamique ne conserve quasiment plus les similarités de couleurs par rapport à l'image originale, ce qui cause un problème pour la reconnaissance des formes de l'image et les résultats sont moins bons.



(a) Bruit faible



(b) Bruit fort

Fig. 10. – Résultats en image pour bruit d’expansion dynamique



(a) Bruit faible



(b) Bruit fort

Fig. 11. – Débruitage de la photo de prise avec un modèle entraîné pour le bruit par expansion dynamique

Le résultat reste très flou pour le bruit faible et encore très bruité pour le bruit fort donc notre approche ne marche que très peu. Nous aurons cependant produit un nouveau type de bruit pour entraîner notre modèle.