

Restauration d'images sous-marines

Cours d'Imagerie Numérique

(Master MVA, 2020-2021)

Antoine Benady

École Normale Supérieure Paris-Saclay

4 Avenue des Sciences, 91190 Gif-sur-Yvette, France

antoine.benady@ens-rennes.fr

Dorian Desblancs

École Normale Supérieure Paris-Saclay

4 Avenue des Sciences, 91190 Gif-sur-Yvette, France

dorian.desblancs@mail.mcgill.ca

Abstract

Dans ce projet, nous explorons le domaine de la restauration d'images sous-marines. Ces images sont souvent bleutées. On y observe donc moins bien les couleurs, ce qui peut avoir un impact sur des tâches comme la reconnaissance d'objets et la segmentation d'images. Nous avons implémenté trois algorithmes. Le premier a été publié par Bianco et al. en 2015 [3]. Il se base sur la correction de la couleur dans l'espace de couleur lab et l'hypothèse d'un monde gris. Les deux autres algorithmes sont des réseaux de neurones, tout deux populaires sur d'autres tâches de correction de couleur en imagerie numérique. Nous observons que la méthode de Bianco et al. est bien plus performante que nos réseaux de neurones. Nos résultats ont été comparés entre-eux. Ils ont aussi été comparés à ceux de l'algorithme Sea-thru [1], une référence dans le domaine.

1. Introduction

La restauration d'images sous-marines est un domaine de recherche assez peu connu. Pourtant, son impact sur la communauté scientifique ne peut être sous-estimé. En effet, au cours des dernières années, les progrès dans les domaines de la reconnaissance et la segmentation d'objets ont été extrêmement utiles pour étudier la biodiversité. En 2019, par exemple, Tresson et al. utilisèrent un réseau de neurone convolutif afin d'étudier les interactions d'insectes [15]. La performance d'algorithmes de classification modernes se prête à ce type d'étude. Après tout, les propriétés physiques des milieux naturels que nous fréquentons permettent de facilement détecter les objets qui nous entourent sous différents angles et à différentes distances.

Ce n'est malheureusement pas le cas pour le milieu sous-marin. En effet, la vision sous-marine est teinte d'un brouillard bleuté. Ce brouillard est dû à la réfraction et à l'atténuation de la lumière visible par l'eau [7]. La lumière qui tente de pénétrer la mer est dans un premier temps réfracté par l'eau. Notons que le taux de réfraction grandit avec la salinité de l'eau. La lumière qui parvient à pénétrer l'eau est ensuite absorbée. La lumière rouge est absorbée le plus rapidement, d'où le teint bleuté de la vision sous-marine. Veuillez consulter la figure 1, qui schématisé la vision sous-marine à différentes profondeurs.

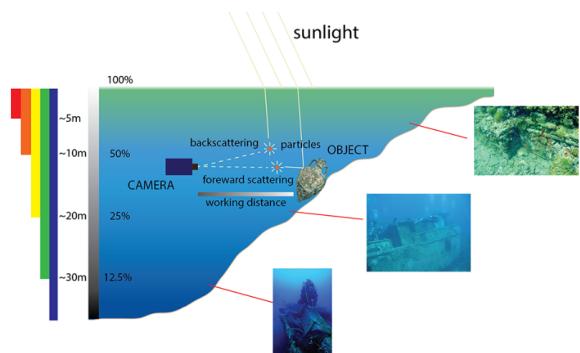


Figure 1: Schéma de l'absorption de la lumière en milieu marin [3]

Récemment, certains algorithmes de reconnaissance et de segmentation à base de réseaux de neurones ont été extrêmement performants sur des jeux de données comme ImageNet [6]. Les problèmes de vision en milieu aquatique n'ont pas permis à ces algorithmes d'avoir un impact sur la recherche sous-marine. Les différentes couleurs présentes

dans le milieu sont très souvent difficiles à distinguer. Il est donc quasiment impossible de différencier une espèce de poisson par rapport à une autre ou de différencier les éléments présents dans une barrière de corail.

Pourtant, ce milieu regorge de jeux de données intéressants tel que celui Beijbom et al. [2], qui regroupe des images de barrière de corail, et celui de Zhuang et al. [16], qui regroupe des images d'une variété d'espèces de poisson. Étudier ces jeux de données automatiquement nous permettrait de mieux suivre l'évolution de la biodiversité aquatique. Dans le contexte actuel d'érosion de la biodiversité, ceci pourrait être très utile.

Dans ce projet, nous avons implémenté trois algorithmes de restauration d'images sous-marines. Le premier provient de Bianco et al. [3], et utilise des techniques de correction d'image classiques. Les deux autres sont des réseaux de neurones convolutifs.

2. État de l'art

Pour simplifier la lecture de ce rapport, nous définissons une méthode moderne par une technique d'analyse ou de transformation d'images à base de réseaux de neurones, ou apprentissage profond. À l'inverse, les méthodes classiques englobent toutes les autres méthodes d'analyse d'image. Il s'agit bien évidemment d'un abus de langage.

2.1. Méthodes Classiques

En 2010, Schettini et Corchs publièrent *Underwater Image Processing: State of the Art Restoration and Image Enhancement Methods* [13]. Cet article de synthèse décrit les méthodes les plus performantes de restauration et de correction d'images sous-marines. Il décrit aussi les possibles modélisations de la lumière bruitée par l'eau et les moyens utilisés pour évaluer une technique de restauration d'image. Nous invitons le lecteur à lire l'article pour s'immerger dans le monde de l'imagerie sous-marine avant 2010.

Au cours des dernières années, trois méthodes sont devenues populaires dans le domaine. La première, publiée en 2011 par Chiang et Chen [5], se base sur le modèle d'acquisition d'images brouillardeuses [11]. Ce modèle, classique en imagerie, décrit les images en utilisant l'équation suivante:

$$I_\lambda(x) = J_\lambda(x)t_\lambda(x) + A(1 - t_\lambda(x)) \quad (1)$$

avec $\lambda \in R, G, B$. On dénote par x un point dans l'image, par $I_\lambda(x)$ l'image aquise par la caméra et par $J_\lambda(x)$ la radiance au point x . $t_\lambda(x)$ est la matrice de transition définie par:

$$t_\lambda(x) = e^{-\beta d(x)} \quad (2)$$

où $d(x)$ est la distance entre la caméra et le point x et β le coefficient de diffusion atmosphérique. Dans leurs

méthode, Chiang et Chen tentent, avec succès, de dé-bruitier les images sous-marines en optimisant l'équation ci-dessus. Ils estiment t_λ et J_λ afin de recréer I_λ , le but étant de minimiser la différence entre l'image originale et celle produite par leurs estimations de t_λ et J_λ .

Plus récemment, en 2017, Bianco et al. se basèrent sur la correction d'image en utilisant l'espace de couleur $l\alpha b$ pour restaurer les images sous-marines. Cette méthode a été implémentée dans ce projet, et sera détaillée plus tard dans le rapport.

Enfin, en 2019, Akkaynak et Treibitz présentèrent Seathru [1]. Cet algorithme est vraisemblablement le plus performant à ce jour. Il fonctionne avec des images RGBD. Dans un premier temps, l'algorithme estime la rétrodiffusion d'une image en utilisant les pixels foncés. Il estime ensuite l'atténuation de la lumière par l'eau à chaque point. La figure 2 contient certains de leurs résultats. Nous pouvons notamment voir dans la deuxième image une palette de couleurs, placée en dessous du corail. Leur algorithme restaure extrêmement bien l'image. La couleur de la palette est quasiment identique à sa couleur sur terre. Nous avons donc utilisé ces images pour juger la performance de nos algorithmes par rapport à l'état de l'art actuel.

2.2. Méthodes Modernes

À l'inverse, il existe très peu d'algorithmes de restauration d'images sous-marines à base de réseaux de neurones. Bien évidemment, cela est en partie dû au fait qu'il est très difficile de construire un jeu de données qui permettrait à un réseau d'apprentissage profond de corriger les images bleutées. Après tout, comment pouvons-nous distinguer une image correcte d'une image avec brouillard alors que nous ne savons même pas à quoi ressemblerait la scène sous lumière normale ?

Pourtant, une équipe de chercheurs s'est penché sur l'idée en 2019. Dans leurs papier *An underwater image enhancement benchmark dataset and beyond*, Li et al. [10] proposent un jeu de données pour la correction d'images sous-marines. Ce jeu de données est composé de 950 images. Leur objectif : un jeu de données de grande taille, qui présente une diversité d'images de qualité et de luminosité variable. Les images "référentes", ou correctes, sont générées par 12 algorithmes classiques de restauration d'images aquatiques. Ensuite, 50 personnes ont été consultées pour choisir la restauration de meilleure qualité. Celle-ci a été choisie comme image référente dans leur jeu de données. Notons tout-de-même que 60 images n'ont pas de références. Aucuns des algorithmes classiques ne sont parvenus à proprement restaurer l'image initiale. Nous avons donc utilisé certaines de ces images afin de visualiser la performance de nos algorithmes. Nos algorithmes à base de réseaux de neurones ont été entraîné sur les 890 autres images.

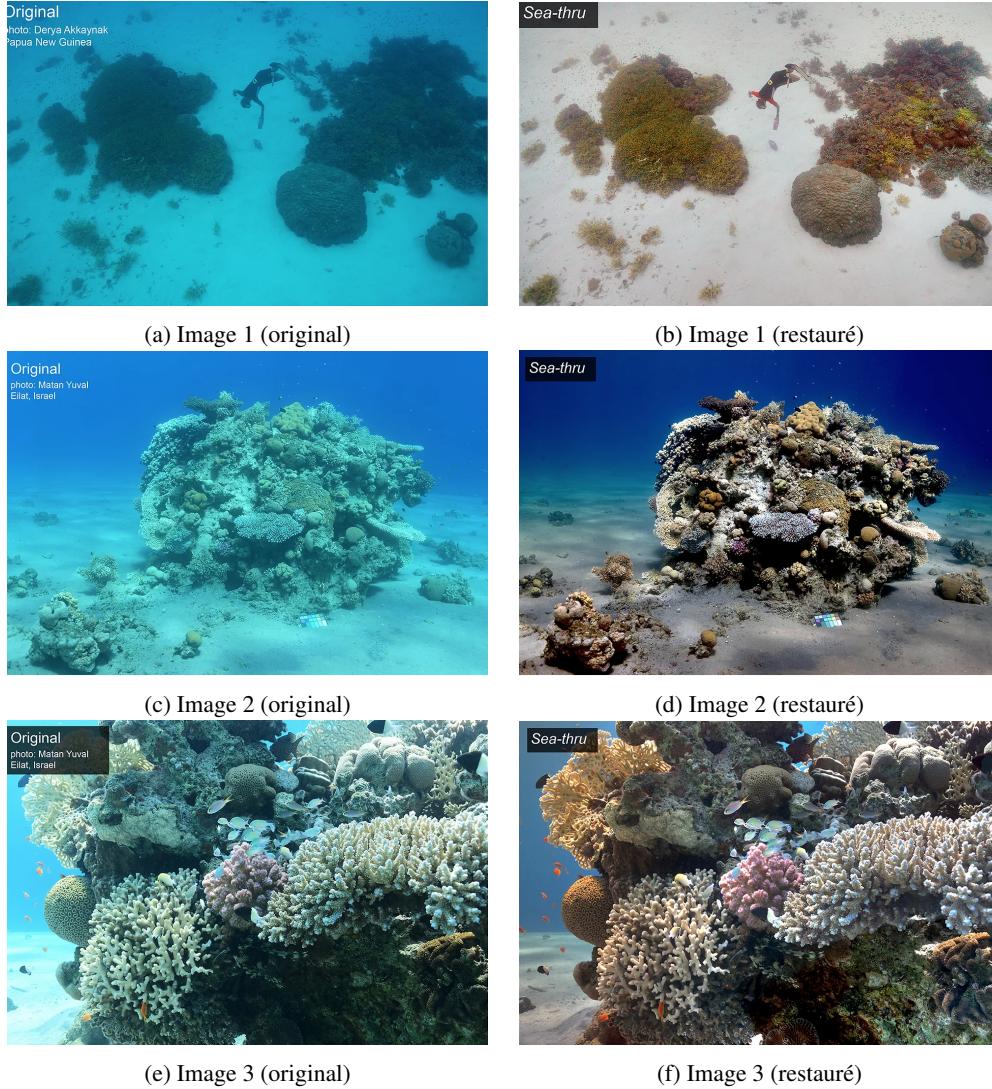


Figure 2: Résultats de l'algorithme Sea-thru sur trois images [1]

Il existe cependant des algorithmes à base de réseaux de neurones très performants dans d'autres domaines connexes au notre. Dans le domaine de la correction de la couleur d'images, il existe une pléthore de modèles très performants. Nous en avons choisi un, proposé récemment par Steffens et al. [14], pour restaurer nos images aquatiques en utilisant le jeu de données de Li et al. [10]. Celui-ci tente, avec succès, d'éclairer les images sombrement éclairées. Nous avons aussi utilisé le réseau de neurones de Zhangyang et al. [9] pour corriger nos images. Ce réseau est un classique dans le domaine de l'élimination du brouillard dans les images naturelles. Il se base sur le modèle d'imagerie introduit dans l'équation 1. Nous présenterons l'architecture de chacun de ces réseaux plus tard dans le rapport.

3. Méthodologie

3.1. Méthode Classique de Bianco et al.

Dans ce projet, nous avons implémenté la méthode proposée par Bianco et al. dans [3]. Cette méthode est basée sur une correction de l'image dans l'espace de couleur $l\alpha\beta$. Cette correction traduit l'hypothèse d'un "monde gris" afin de corriger la teinte bleutée de l'image.

Cette méthode se décompose en trois temps :

1. Passage dans l'espace $l\alpha\beta$
2. Application de la correction d'image
3. Passage inverse pour retourner dans l'espace RGB

3.1.1 Passage dans l'espace de couleur $l\alpha\beta$

L'espace $l\alpha\beta$ est un espace à trois canaux :

- 1 décrit la luminance, qui est proportionnel à la somme des canaux r, g, et b.
- α : qui décrit les couleurs bleue et jaune, proportionnel à $(r + g - b)$
- β : qui décrit les couleurs rouge et verte, proportionnel à $(r - g)$

Pour passer d'une image en RGB à une image en $l\alpha\beta$, il faut appliquer une série de transformations, schématisée dans la figure 3. Tout d'abord, une correction γ est appliquée. Notons que le coefficient γ n'est pas donné dans l'article, nous l'avons donc calibrer manuellement pour nous rapprocher des résultats de l'article. Ensuite, une série de transformations linéaires sont appliquées pour arriver dans l'espace LMS. Une fois dans cet espace, nous devons prendre le log pour chaque pixel. Nous avons appliqué un effet de seuillage sur les valeurs proches de zéro qui n'est pas mentionné dans l'article. Nous avons essayé avec un seuil simple et avec un déplacement d'histogramme, qui donnait des résultats très similaires, nous avons donc garder un simple seuil. Enfin, une dernière transformation linéaire permet de passer dans l'espace $l\alpha\beta$.

3.2. Correction traduisant l'hypothèse de "monde-gris"

L'hypothèse de "monde-gris" introduite par Buchsbaum en 1980 [4] stipule que la moyenne de l'image doit être grise. De cette hypothèse, on déduit la correction suivante dans l'espace $l\alpha\beta$ pour chaque pixel (m, n) :

$$\begin{aligned} l_\alpha^*(m, n) &= l_\alpha(m, n) - \bar{l}_\alpha \\ l_\beta^*(m, n) &= l_\beta(m, n) - \bar{l}_\beta \end{aligned}$$

Où $\bar{l}_\alpha(m, n)$ et $\bar{l}_\beta(m, n)$ sont les moyennes des canaux α et β avant correction, $l_\alpha(m, n)$ et $l_\beta(m, n)$ les valeurs du pixel (m, n) avant correction.

Concernant le canal de luminance, il est corrigé en appliquant un filtrage d'histogramme sur les 1% des valeurs extremes de luminance les plus faibles et les plus fortes puis un étirement de l'histogramme.

3.2.1 Retour dans l'espace RGB

Ensuite, il faut repasser l'image dans l'espace RGB. Pour cela, on applique les transformation inverses décrites section 3.1.1.

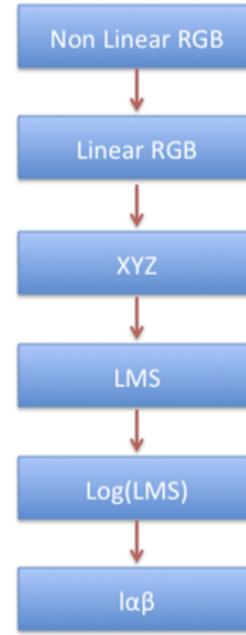


Figure 3: Transformations à appliquer pour convertir une image RGB en $l\alpha\beta$ [3]

3.3. Réseaux de neurones

3.3.1 Architecture de Steffens et al.

Le réseau de neurones de Steffens et al. est composé de plusieurs branches. Chaque branche est composée d'une série de filtres convolutifs. Ceux-ci sont des filtres avec un petit noyau, de tailles 3×3 ou 1×1 , avec un stride et une dilation variable. Chaque convolution est suivie d'une activation ELU, définie par :

$$f(x) = \begin{cases} x, & ,x \geq 0 \\ \alpha(e^x - 1), & ,x < 0 \end{cases} \quad (3)$$

Chaque branche contient aussi un bloc de filtres convolutifs. Tous les filtres dans le bloc ont la même entrée, mais transforment l'image différemment, de par la valeur variable de leur dilation. Ces filtres, populaires dans les architectures de réseau de neurones de rectification d'image, permettent d'extraire un maximum d'information à chaque niveau du réseau sans avoir trop de paramètres à régler pendant l'entraînement. Le réseau interpole aussi l'image plusieurs fois afin de maintenir la même taille d'image en sortie. La figure 4 montre l'architecture du réseau. Le lecteur est bien sûr libre de consulter notre code pour voir une implémentation détaillée utilisant PyTorch [12].

Pour entraîner ce réseau, nous avons utilisé

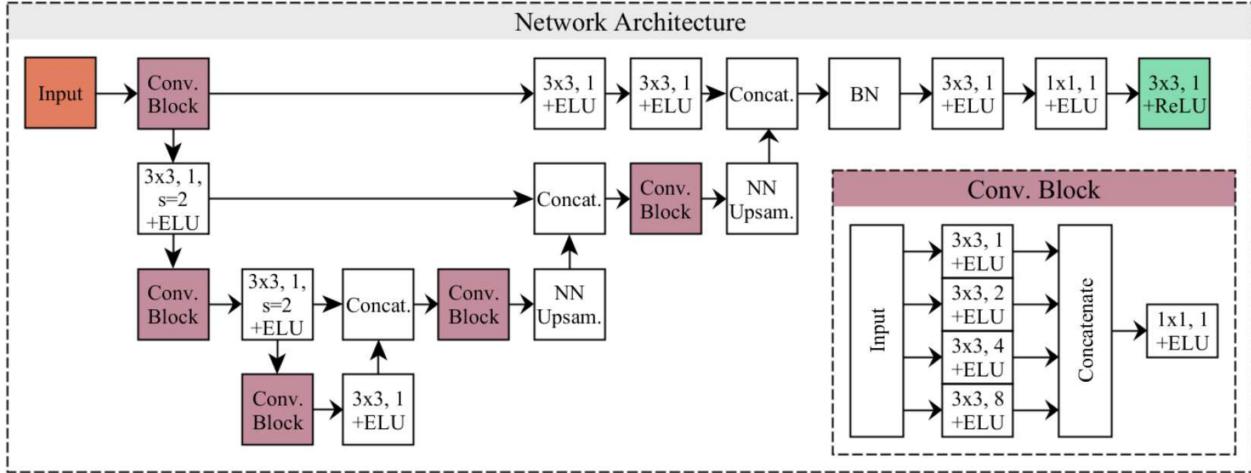


Figure 4: Architecture du réseau de neurones de Steffens et al. [14]

l’optimiseur ADAM [8] avec un taux d’apprentissage de 0,001. Ce taux a été multiplié par 0,1 toutes les 15 époques pendant les 40 époques d’entraînement. On utilise une taille de batch de 80 images, toutes coupées et dimensionnées avec une taille de 200 pixels en longueur et en largeur. Nous avons utilisé 800 images pour entraîner le réseau et 90 images pour le valider. À chaque époque, l’erreur moyenne quadratique, définie par :

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (4)$$

a été minimisée. Le réseau avec la plus petite erreur moyenne sur les 90 images de validation a été sauvegardé et testé.

3.3.2 Architecture de Zhangyang et al.

L’architecture du réseau de neurones de Zhangyang et al. est assez similaire, bien que le modèle soit beaucoup plus léger. Celui-ci n’utilise que cinq filtres convolutifs, avec des noyaux de dimensions grandissantes (1×1 pour le premier filtre, 7×7 pour le dernier). Les images sont transformées par un premier filtre. La sortie de celui-ci est alors utilisée comme entrée pour le deuxième filtre. La sortie du second est ensuite utilisée comme entrée pour le troisième, et ainsi de suite. Les sorties de chaque filtre sont ensuite mises en commun et utilisées pour la restauration de l’image originale. Une fonction d’activation ReLU est utilisée après chaque filtre. Celle-ci est définie par :

$$f(x) = \max(0, x) \quad (5)$$

pour tout $x \in R$. La figure 5 montre cette architecture.

Le réseau de neurones a été entraîné de la même manière que celui de Steffens et al. La seule différence : ce réseau a

besoin de beaucoup plus de données pour obtenir un résultat convenable. Ainsi, nos images ont été re-dimensionnées avec une largeur et une longueur de 800 pixels. Notre taille de batch a donc dû être diminuée à 32.

4. Résultats

Comme nous pouvons le voir dans les figures 6, 7, et 8, la méthode classique est bien plus performante que les méthodes à base de réseaux de neurones pour corriger les images bleutées. Les résultats obtenus sur les images 1, 2, et 3 s’approchent de ceux obtenus par l’algorithme Sea-thru [1], qui est aujourd’hui l’algorithme le plus performant dans le domaine. Ceci est impressionnant car l’algorithme n’utilise pas des images RGBD pour fonctionner. La méthode Sea-thru ne peut que marcher en utilisant ce type d’image, qui est assez rare dans le monde réel. Cela montre aussi que l’hypothèse de “monde-gris” est applicable à la restauration d’images sous-marines. La correction de la couleur dans l’espace *lab* se prête clairement à ce type de rectification.

À l’inverse, les deux méthodes à base de réseau de neurones que nous avons utilisées ne sont pas très performantes. Elles parviennent à corriger un minimum le teint bleuté de nos images. Cependant, cette correction est trop faible, et ne nous aiderait pas pour une tâche de reconnaissance d’objets, par exemple. La palette de couleur qui se trouve dans l’image 2 illustre bien cette idée. Alors que les algorithmes classiques parviennent à la rectifier pour qu’elle ait une couleur semblable à celle que l’on trouverait hors de l’eau, nos réseaux de neurones lui donnent toujours une couleur bleutée. Plus précisément, le réseau de Zhangyang et al. ne corrige quasiment pas l’image originale. Ceci est vraisemblablement dû à la légèreté du réseau, qui ne comporte que 1761 paramètres à entraîner. En comparaison,

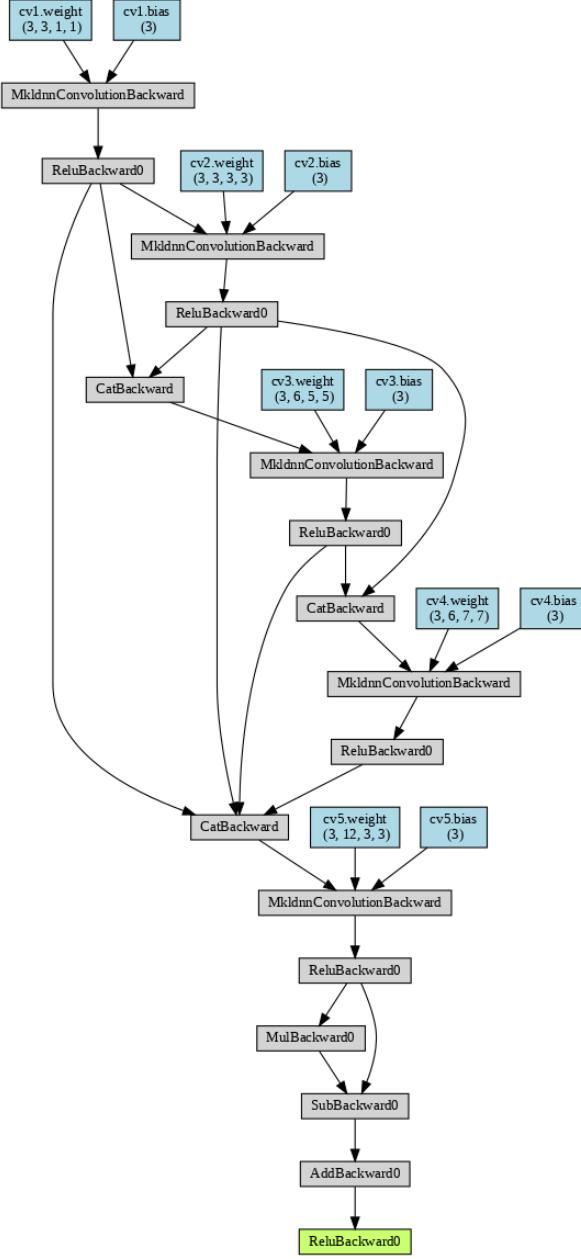


Figure 5: Architecture du réseau de neurones de Zhangyang et al.

l'architecture de Steffens et al., qui contient plus 380 000 paramètres à entraîner, est plus performante. On peut toutefois observer quelques distorsions en coin d'image, dû aux interpolations successives qu'on trouve dans certaines couches du réseau. Ceci est très problématique.

La méthode de Bianco et al. a elle quelques limites. Sur certaines de nos images, l'algorithme a tendance à distordre les couleurs des images. On observe par exemple une couleur violette dans l'image 5, un bleu turquoise peu naturel dans l'image 4, et une image 6 bien trop

grisée. Les méthodes à base de réseau de neurones, bien qu'elles soient moins performantes sur la plupart des images, ont moins tendance à distordre les couleurs originales de manière anormale. Elles évitent probablement ce type de problème de par le fait qu'elles sont entraînées sur une panoplie d'images plus diverse, qui comprend des images à luminosité variable. Nous invitons le lecteur à consulter notre Github, qui comprend un plus grand échantillon de résultats.



(a) Méthode classique de Bianco et al.



(b) Méthode de Steffens et al.



(c) Méthode de Zhangyang et al.

Figure 6: Image 1 (restauré)



(a) Méthode classique de Bianco et al.

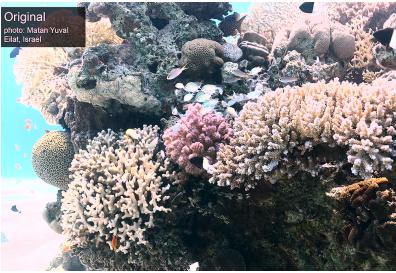


(b) Méthode de Steffens et al.



(c) Méthode de Zhangyang et al.

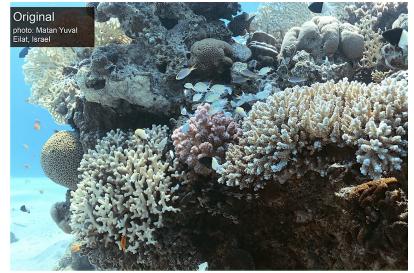
Figure 7: Image 2 (restauré)



(a) Méthode classique de Bianco et al.



(b) Méthode de Steffens et al.



(c) Méthode de Zhangyang et al.

Figure 8: Image 3 (restauré)

5. Prochaines étapes

Dans un premier temps, nous pensons qu'il est nécessaire d'étudier l'algorithme Sea-thru [1] d'une manière plus détaillée. Nous aimerais pouvoir le tester sur d'autres images afin d'explorer ses possibles limites. Bianco et al. ne présentent qu'un très petit nombre d'images dans leur article [3]. Notre projet a permis d'illustrer certaines de ses faiblesses. En utilisant un jeu de données avec une variété d'images sous différentes luminosités, nous pouvons mettre en avant de futures axes de recherche dans le domaine de la restauration d'images sous-marines, et même la correction d'images en général.

Nous pensons aussi que les nouveaux algorithmes comme Sea-thru et celui de Bianco et al. nous permet-

traient de créer un nouveau jeu de données, avec des images restaurées, ou "correctes", de meilleure qualité. Cela permettrait à nos réseaux de neurones d'obtenir de meilleurs résultats. Dans leurs papier, Akkaynak et al. [1] affirment avoir plus de 1100 images corrigées par l'algorithme Sea-thru. Nous pourrions construire un jeu de données de la même manière que Li et al. [10], avec des algorithmes plus modernes. Nos réseaux de neurones montrent après tout qu'ils sont plus résistants aux corrections anormales de la couleur. Pourquoi ne pas utiliser à la fois des algorithmes classiques et modernes pour mieux corriger les images aquatiques ?

Enfin, nous pensons qu'il est nécessaire d'explorer une architecture plus complexe de réseau de neurones. Finalement, nos réseaux n'ont pas beaucoup de couches, et sont



(a) Original



(b) Méthode classique de Bianco et al.



(c) Méthode de Steffens et al.



(d) Méthode de Zhangyang et al.

Figure 9: Image 4



(a) Original



(b) Méthode classique de Bianco et al.



(c) Méthode de Steffens et al.



(d) Méthode de Zhangyang et al.

Figure 10: Image 5



(a) Original



(b) Méthode classique de Bianco et al.



(c) Méthode de Steffens et al.



(d) Méthode de Zhangyang et al.

Figure 11: Image 6



(a) Original



(b) Méthode classique de Bianco et al.



(c) Méthode de Steffens et al.



(d) Méthode de Zhangyang et al.

Figure 12: Image 7

pourtant assez performants. Les résultats récents en apprentissage profond ont démontré que plus le réseau est profond, plus les résultats sont impressionnantes sur des tâches de vision par ordinateur. Nous pourrions aussi explorer d'autres quantités à minimiser pendant l'entraînement, tel que le peak signal to noise ratio (PSNR) et le structural similarity (SSIM), assez populaires dans le domaine de la correction d'images.

6. Conclusion

Dans ce projet, nous avons implémenté trois algorithmes de restauration d'images sous-marines. Le premier est

basé sur la correction de la couleur dans l'espace *lab* et l'hypothèse d'un "monde gris". Les deux autres sont des réseaux de neurones dont l'architecture est populaire dans des domaines connexes au notre. Nos résultats démontrent que la première méthode est bien plus performante que les deux autres, et s'approche de l'état de l'art actuel. Nous pensons cependant qu'avec un jeu de données plus moderne, nos réseaux de neurones pourraient être plus performants. Nous invitons le lecteur à consulter notre Github¹ pour plus de résultats. Le dépôt contient aussi le code que nous avons utilisé, qui peut être utilisé pour d'autres tâches.

¹https://github.com/d-dawg78/MVA_ImaNum

References

- [1] Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691, 2019.
- [2] Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1170–1177. IEEE, 2012.
- [3] Gianfranco Bianco, Maurizio Muzzupappa, Fabio Bruno, Rodrigo Garcia, and Luká Neumann. A new color correction method for underwater imaging. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):25, 2015.
- [4] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, pages 1–26, 1980.
- [5] John Y Chiang and Ying-Ching Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE transactions on image processing*, 21(4):1756–1769, 2011.
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [7] Byrne Robert Howard Duxbury, Alyn C. and Fred T. Mackenzie. Seawater. In *Encyclopedia Britannica*. Ieee, 2020.
- [8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [9] Boyi Li, Xulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.
- [10] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing*, 29:4376–4389, 2019.
- [11] Earl J McCartney. Optics of the atmosphere: scattering by molecules and particles. *New York*, 1976.
- [12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [13] Raimondo Schettini and Silvia Corchs. Underwater image processing: state of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, 2010:1–14, 2010.
- [14] Cristiano Rafael Steffens, Valquiria Huttner, Lucas Ricardo Vieira Messias, PLJ Drews, Silvia SC Botelho, and RS Guerra. Cnn-based luminance and color correction for ill-exposed images. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3252–3256. IEEE, 2019.
- [15] Paul Tresson, Philippe Tixier, William Puech, and Dominique Carval. Insect interaction analysis based on object detection and cnn. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019.
- [16] Peiqin Zhuang, Yali Wang, and Yu Qiao. Wildfish: A large benchmark for fish recognition in the wild. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1301–1309, 2018.