

Débruitage par réseau convolutionnel profond avec un terme non-local

Master MVA: Projet d'imagerie numérique

AMEKOE Kodjo Mawuena

MBIA NDI Marie Thérèse

January 2021

Abstract

Ce projet est basé sur le papier <https://arxiv.org/pdf/1803.02112.pdf>. Cet article ayant pour titre "Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising" combine les méthodes par patches non locales et les méthodes de débruitage par les réseaux de neurones convolutifs (CNN) pour le débruitage d'image. Dans ce projet nous analysons dans un premier temps l'article sus-cité et ensuite proposons une implémentation de leur algorithme sur python en utilisant principalement le réseau FFDNET.

1 Introduction

Le débruitage d'image est une problématique au coeur du traitement d'image. Son but est d'estimer une image à partir de son observation bruitée. Pour cela plusieurs algorithmes ont été développés, notamment le DCT denoising, Non local means, etc. Avec l'avènement des réseaux de neurones, de nouvelles techniques ont été mises en place pour le débruitage d'image. Ce projet se base sur l'analyse et l'implémentation de l'article Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising disponible sur ce lien <https://arxiv.org/pdf/1803.02112.pdf>.

2 Analyse de l'article

En littérature, nous retrouvons plusieurs méthodes pour le débruitage d'image. Nous avons par exemple le DCT denoising qui est une méthode de débruitage basée sur la *Discrete Cosine Transform* (DCT) et le seuillage. Il décompose l'image en patches puis projette ces patches sur la base DCT et garde les coefficients à grande fréquence en appliquant un seuillage à ces coefficients et applique la DCT inverse, puis calcule la valeur moyenne de chaque pixel dans les patches. Cet algorithme a du mal avec les basses fréquences et les débruite mal. C'est le plus qu'apporte le DCT multi scale qui applique le même algorithme mais à différents niveaux de zoom de l'image afin de mieux débruiter les basses fréquences. Une implémentation de ces algorithmes est disponible sur les liens suivants: <http://www.ipol.im/pub/art/2011/ys-dct/> , <http://www.ipol.im/pub/art/2011/ys-dct/>.

Les deux algorithmes précédemment présentés n'exploitent pas le fait que l'on puisse débruiter une image en utilisant l'auto similarité des patches de cette image, ce qui est la particularité des algorithmes comme le BM3D et le *non local means* (NLM). Ces deux algorithmes sont utilisés pour faire du débruitage. Les deux méthodes agissent de manière similaire en regroupant entre eux les morceaux d'images qui se ressemblent pour obtenir un débruitage du pixel central. La différence entre NLM et BM3D est que la première se contente d'un calcul simple pour décider de la nouvelle valeur: on moyennise tous les pixels centraux des voisinages ressemblants. La seconde regroupe tous les voisinages ressemblants, produisant un bloc tri-dimensionnel, qui est filtré avec l'algorithme NLM et les patches sont débruités avec l'algorithme DCT. En general le BM3D est meilleur que le NLM. Une implémentation de ces algorithmes est disponible sur les liens suivants: http://www.ipol.im/pub/art/2011/bcm_nlm/ , <http://www.ipol.im/pub/art/2017/184/>.

Nous notons aussi des techniques de débruitage en utilisant des méthodes bayésiennes. Le but de ces techniques est d'estimer la probabilité qu'un patch soit similaire à un autre. L'utilisation des méthodes de débruitage précédentes demande de faire plusieurs hypothèses sur la nature du bruit. Ces hypothèses ne sont pas toujours vérifiées en réalité. Avec l'avènement des réseaux de neurones profonds, on a développé de nouvelles méthodes pour débruiter des images basées sur des réseaux de neurones convolutionnels. Avec ces méthodes, l'on n'a pas besoin de faire d'hypothèses sur la nature du bruit. L'article étudié tout au long de ce travail allie les forces des algorithmes présentés précédemment pour proposer une nouvelle méthode de débruitage.

La problématique d'intérêt dans cet article est le débruitage des images. Pour ce faire, les auteurs relient deux algorithmes de débruitage d'image. L'aspect le plus intéressant de cette méthode est que les auteurs utilisent deux algorithmes avec des inconvénients complémentaires. Il s'agit des algorithmes : réseaux de neurones profonds convolutionnels et le non local filtering. Le premier algorithme a des performances inférieures sur des textures à forte auto-similarité alors que le second performe bien lorsque la texture a de forte auto-similarité et a de mauvais résultats le cas échéant. Ainsi mettre ces deux algorithmes ensemble permet d'avoir de bons résultats quel que soit le type de texture utilisé. L'algorithme proposé par les auteurs est le suivant:

Algorithm 1 Proposed framework	
Require: z	noisy signal
Require: σ	noise standard deviation
Require: K	number of iterations
Require: $\lambda_k, k = 1, \dots, K$	iteration steps
Require: $\tau_k, k = 1, \dots, K$	NLF thresholds
1: for $k = 1$ to K do	
2: $\bar{z}_k = \lambda_k z + (1 - \lambda_k) \hat{y}_{k-1}$	convex combination
3: $\tilde{y}_k = \text{CNNF}(\bar{z}_k, \lambda_k \sigma)$	CNN-based filter
4: $\hat{y}_k = \text{NLF}(\tilde{y}_k, \tau_k)$	Nonlocal filter
5: end for	
6: return \hat{y}_K	return final estimate

Dans un premier temps, on effectue une combinaison convexe entre l'image bruitée fournie en entrée et le résultat de l'image débruitée obtenue à l'itération $k-1$. Ensuite l'on applique un réseau de neurone convolutionnel au résultat obtenu et enfin l'algorithme du non local filtering. Pour effectuer le non local filtering à une image, on la regroupe tout d'abord en bloc matching. Ce dernier consiste à choisir pour un patch i d'une image de taille $N_1 \times N_1$, les N_2 patches qui lui sont similaires. la notion de similarité ici fait référence à la distance entre les deux patches. Après avoir fait le bloc matching, pour chaque bloc on fait une transformation 1D suivant la troisième dimension, on applique un seuillage et la transformation 1D inverse. L'image est agrégée en faisant une moyenne pondérée des différents patches.

Dans la suite nous allons présenter les résultats qui émanent de l'implémentation de cet article.

3 Implémentation de l'article et résultats

Pour implémenter l'algorithme proposé par cet article, nous avons utilisé le FFDnet qui est un algorithme de débruitage par réseaux convolutionnels. Pour les hyperparamètres nous avons considéré ceci : $K=2$, $\lambda_k = 1/k$, $\tau_k = \sigma \lambda_k / 4$

La transformation 1D utilisée est les ondellettes de Haar. Nous avons utilisé une version pré-entraînée du FFDnet. Cette version provient de ce lien <https://github.com/gfacciolo/MVAdenoising>. Nous avons découpé les différentes images en patches de taille 10×10 et pour chaque patch nous avons considéré ses 32 patches similaires. Pour juger la qualité du débruitage, nous allons utiliser le PSNR comme mesure.

En moyenne notre algorithme met 187 secondes pour s'exécuter sur des images de taille 256 x 256.

Expérience 1

Notre première expérimentation avec cette image a été de la débruiter avec trois niveaux de bruit. Nous avons donc considéré trois variances pour le bruit: 30, 50, 75. Force est de constater que l'algorithme a du mal à débruiter correctement lorsque la variance du bruit augmente ; ce résultat est en adéquation avec le résultat de l'article. Les images débruitées sont reportées sur la figure ci-dessous:

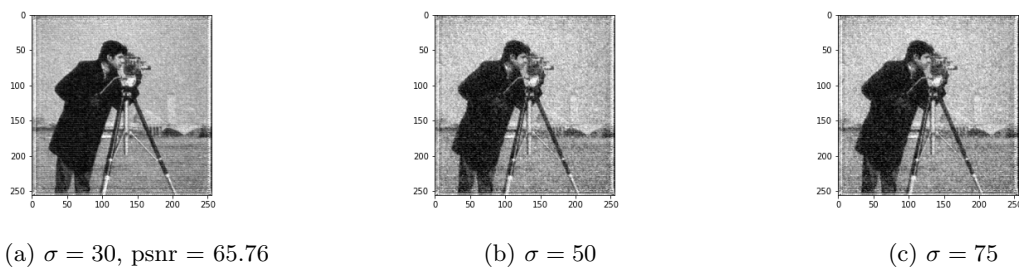


Figure 1: Débruitage suivant le niveau de bruit σ

Dans la suite nous allons considérer 30 comme variance du bruit.

Expérience 2

Notre seconde expérience a été de débruiter une image avec l'algorithme proposé par l'article. L'algorithme ici a du mal à bien débruiter les surfaces uniformes. On observe des artefacts sur les surfaces uniformes.

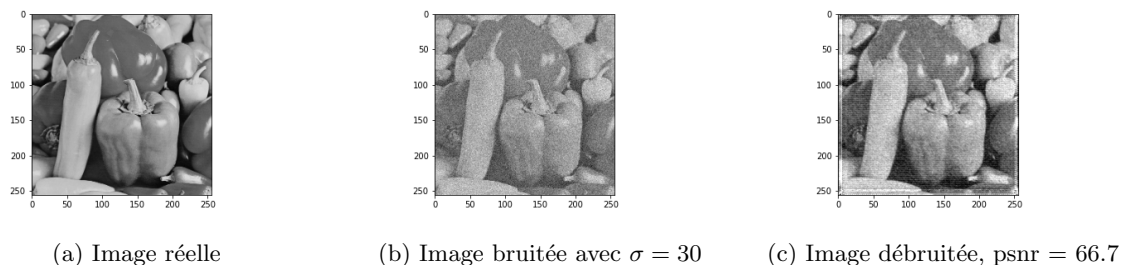
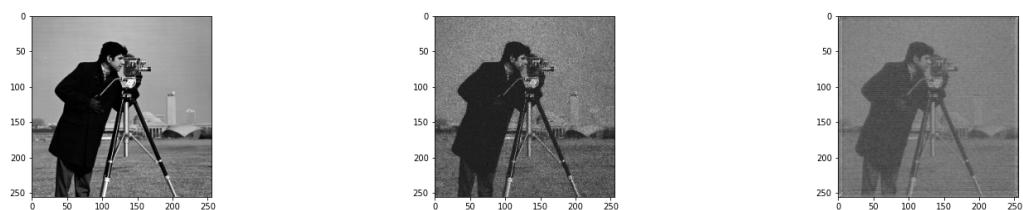


Figure 2: Débruitage FFDnet + NLF , $\sigma = 30$

Expérience 3

Dans cette expérience nous avons voulu apprécier le NLF et le FFDnet séparément. Nous constatons dans ce cas que le NLF débruite mieux que le FFDnet qui ajoute une sorte de flou sur toute l'image. Néanmoins pour le NLF nous observons du flou sur des zones où l'intensité lumineuse de l'image change(notamment la zone de transition entre la veste du photographe et le ciel). D'autre part ces algorithmes combinés à deux donnent un résultat visuel moins satisfaisant. Ici le débruitage a tendance à éclaircir le ciel.



(a) Image réelle

(b) Image débruitée avec notre NLF

(c) Image débruitée par FFDnet

Figure 3: Débruitage à niveau de bruit $\sigma = 30$

3.1 Conclusion

En somme, le but de ce travail était d'implémenter une méthode de débruitage par réseau convolutionnel profond avec un terme non-local en se basant sur l'article <https://arxiv.org/pdf/1803.02112.pdf>. Nous l'avons testé sur des images en niveau de gris. D'après nos résultats, cet algorithme a une performance décroissante avec le niveau de bruit ; ce qui est bien en adéquation avec le résultat de l'article. Aussi cet algorithme a du mal à débruiter les surfaces claires en les rendant encore plus claires et aussi sur les zones de changement de texture où nous avons observé du flou. Nous constatons aussi que dans ce cas, le NLF performe mieux que le FFDnet. Une piste d'amélioration de ce travail serait de bien tuner le FFDnet avant de l'utiliser et aussi d'optimiser l'algorithme de NLF.