Université de Montpellier

M2 - Projet Image
Compte Rendu 4

Verniol Baptiste
Fournier Alexandre

Année 2023-2024

Sommaire

1. CNN en débruitage - état de l'art

Nous avons trouvé plusieurs techniques utilisant des réseaux de neurones pour le débruitage. Les voici :

1.1 DnCNN (Deep Convolutional Neural Network for Image Denoising)

DnCNN est un type de réseau de neurones convolutifs spécialement conçu pour éliminer le bruit des images. DnCNN fonctionne en utilisant des blocs résiduels, ce qui permet au réseau d'apprendre à distinguer les caractéristiques du bruit tout en préservant les détails importants de l'image. Lors de l'entraînement, DnCNN utilise des paires d'images, une version bruitée et sa version débruitée correspondante, pour apprendre à modéliser la relation entre les deux. Ainsi, le réseau peut ensuite prendre une image bruitée en entrée et générer une estimation débruitée. Cette approche a démontré une grande efficacité dans le débruitage d'images, en particulier pour le bruit gaussien, et a contribué de manière significative aux avancées en matière de débruitage basé sur des réseaux de neurones.

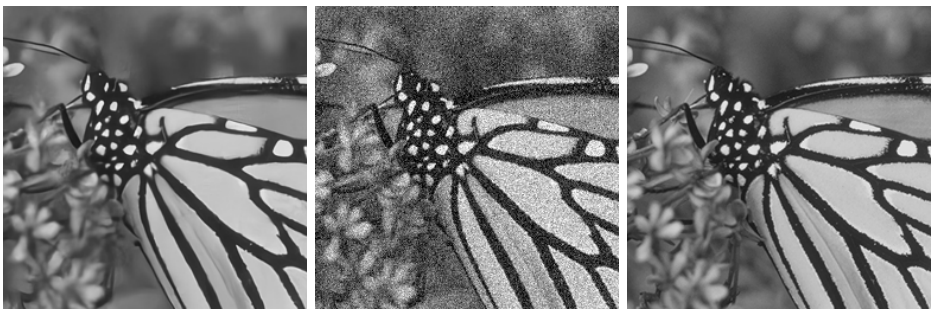

FIGURE 1.1 – Source. gauche bruitée, milieu débruité par DnCNN, droite image de base

1.2 FFDNet (Fast and Lightweight Denoising Network)

FFDNet est une architecture de réseau de neurones conçue pour le débruitage rapide et efficace des images. Elle utilise le même principe que DnCNN tout en maintenant une architecture légère pour une exécution rapide. Comparé à DnCNN, FFDNet vise à atteindre un bon équilibre entre la qualité du débruitage et l'efficacité computationnelle, ce qui en fait une option attractive pour des applications nécessitant une détection de bruit en temps réel ou à faible coût computationnel.

1.3 REDNet (Residual Encoder-Decoder Network)

Une autre méthode que nous avons trouvé. Mais nous n'avons pas parfaitement compris ce qui la différenciait. Elle a l'air d'utiliser les canaux de rouleur RGB pour effectuer son traitement.

1.4 CBDNet (Convolutional Blind Denoising Network)

Ce modèle aborde le défi du débruitage lorsque des informations sur le type spécifique de bruit ne sont pas disponibles pendant l'entraînement. CBDNet utilise des blocs résiduels et des mécanismes d'attention pour apprendre à débruiter les images de manière robuste, même en l'absence de connaissances préalables sur le bruit. Contrairement à des approches nécessitant des informations détaillées sur le bruit, CBDNet est capable de traiter différents types de bruit de manière adaptative. En comparaison avec DnCNN, CBDNet se distingue par son aptitude à gérer des scénarios de débruitage où le modèle de bruit est inconnu, en faisant ainsi une option pertinente pour des applications pratiques nécessitant une adaptabilité aux diverses conditions de bruit. Comme les vraies photographies.

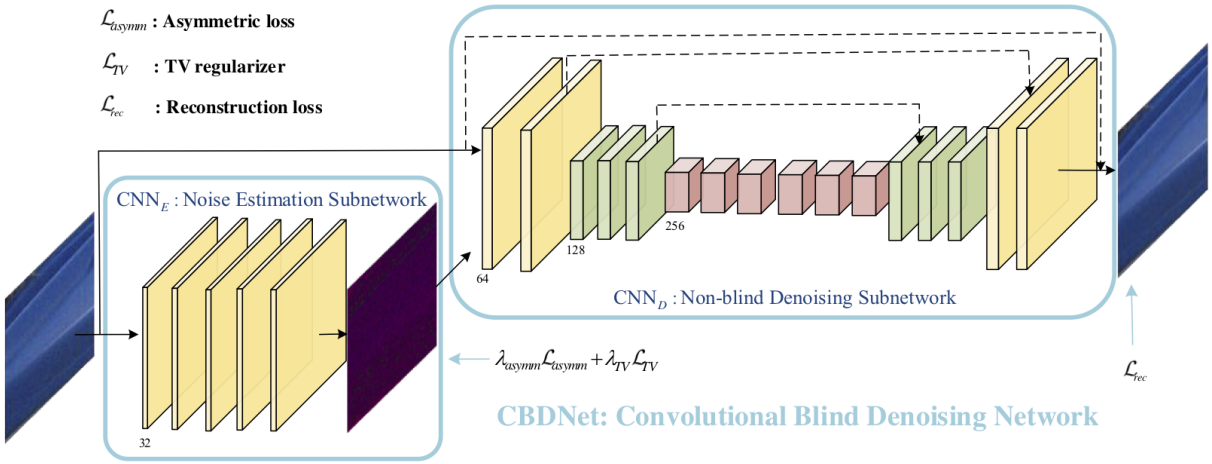

FIGURE 1.2 – Source. Network Structure

1.5 Noise2Noise

Noise2Noise est une approche novatrice de débruitage. Contrairement à de nombreuses méthodes traditionnelles de débruitage qui nécessitent des paires d'images bruitées et débruitées pour l'entraînement, Noise2Noise a la particularité d'apprendre à débruiter en utilisant uniquement des paires d'images bruitées. Cette approche est rendue possible en exploitant les propriétés statistiques du bruit dans les images, permettant au réseau d'apprendre à partir de multiples instances du bruit sans nécessiter des versions débruitées de référence. Noise2Noise a montré sa capacité à obtenir des résultats de débruitage convaincants dans des situations dans lesquelles des données débruitées de référence ne sont pas disponibles, ce qui en fait une technique précieuse pour des scénarios réels où l'acquisition de données débruitées peut être coûteuse ou impossible. Comparé aux méthodes traditionnelles de débruitage, Noise2Noise offre une approche plus souple et adaptative.

2. Mise en place

2.1 Base de données de test

Pour la base de données que nous allons utiliser, nous avons pensé à créer la nôtre à partir de rendu d'image. Mais la quantité et le temps nécessaire pourrait être un peu long. Nous avons donc regardé qu'elle base utilisait le code du CNN qui nous intéressait. Ils en proposaient deux, une petite que nous avons récupérée. Et une plus grosse pour laquelle il faut contacter ceux qui la possèdent pour pouvoir y avoir accès et l'utiliser. Baptiste a donc contacté IMAGENET, en espérant que ceux-ci nous répondent positivement. Si ce n'était pas le cas, nous nous en ferions une à partir de rendu et de bruit générer nous-mêmes.

2.2 CNN choisi

Nous avons décidé de choisir Noise2Noise comme modèle de CNN à implémenter. Car avec les images rendues, il n'y a pas toujours la possibilité sans prendre un très long temps d'avoir des images parfaitement propre en sortie. Utilisé un modèle tel que Noise2Noise qui s'entraine sur des paires d'images bruitées nous a semblé une meilleure idée. Il est possible aussi que celui-ci permet de traiter plus de variété de bruit efficacement qu'avec la méthode DnCNN.

2.3 Test

Avant de commencer notre propre version, nous avons décidé de lire le papier et de tester le code réalisé par ces auteurs mis en référence. Le test de leur code nous a permis de mieux comprendre le papier même si nous n'avons pas fini de le comprendre.

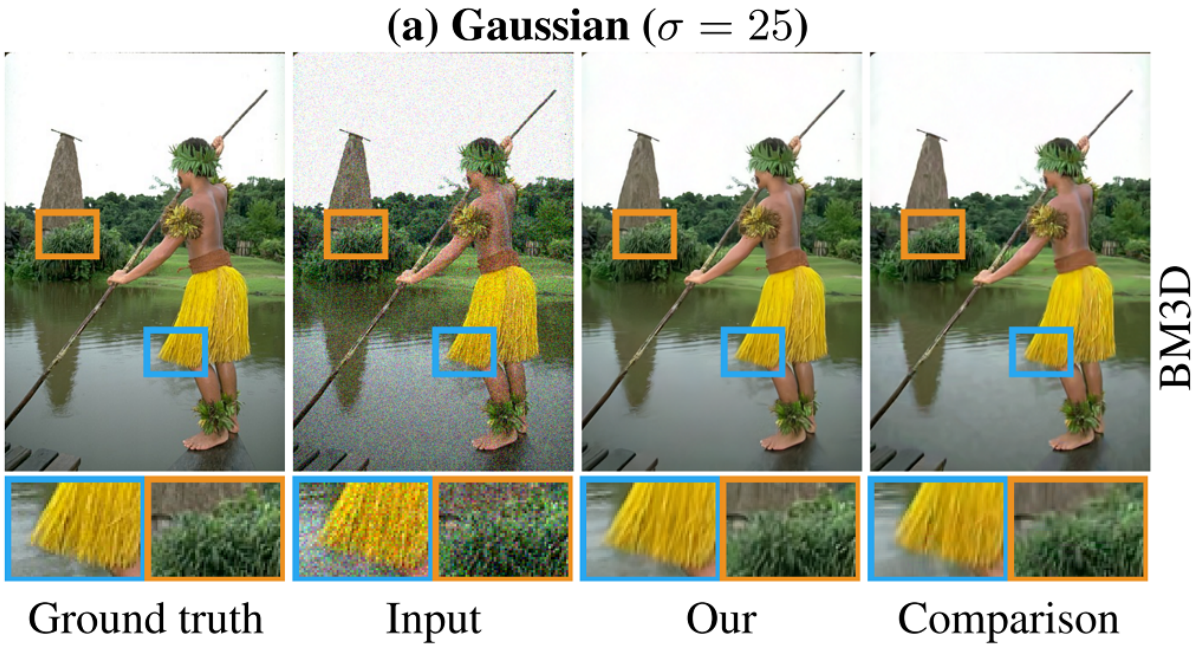

FIGURE 2.1 – Source. Exemple de leur résultat

Nous allons essayer de terminer de le comprendre cette semaine et de mettre en place notre propre version du modèle Noise2Noise.

3. Ajout au projet

3.1 Métriques de qualité d'image pour l'évaluation : SSIM

Nous avons normalement complété et terminé la méthode SSIM qui servira comme troisième métrique pour évaluer nos images.

$$SSIM(x,y) = l(x,y).c(x,y).s(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)(cov_{xy} + c_3)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)(\sigma_x\sigma_y + c_3)}$$

avec

- μ_x la moyenne de x ;
- μ_y la moyenne de y ;
- σ_x^2 la variance de x ;
- σ_y^2 la variance de y ;
- cov_{xy} la covariance de x et y ;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ et $c_3 = \frac{c_2}{2}$ trois variables destinées à stabiliser la division quand le dénominateur est très faible ;
- L la dynamique des valeurs des pixels, soit 255 pour des images codées sur 8 bits ;
- k_1 = 0,01 et k_2 = 0,03 par défaut.

FIGURE 3.1 – Formule SSIM

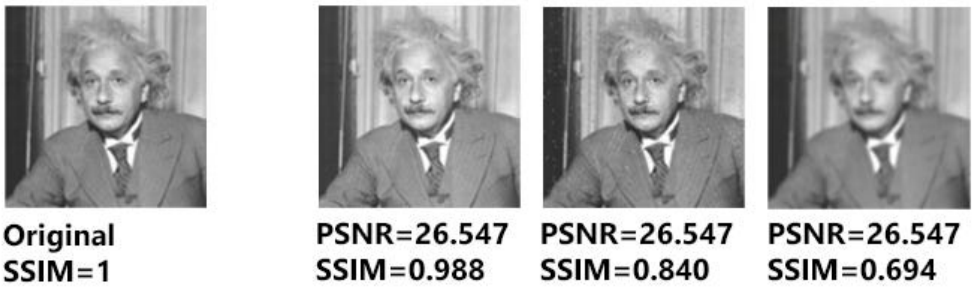

FIGURE 3.2 – Source. Exemple SSIM

4. Références

4.1 Wikipédia :

4.1.1 Méthode d'évaluation du résultat

Structural Similarity — Wikipédia

4.1.2 CNN

Generative adversarial network - Wikipedia

4.2 Site :

DnCNN

FFDNet

REDNet

CBDNet

Noise2Noise code
Noise2Noise papier

Image Denoising with GAN. 1. Introduction | by La Javaness R&D