



HAI918I Image, sécurité et deep learning

Projet-CR5: Ajout de méthodes classiques, analyse de résultats et début CNN

Luna BOSSU
Jean Louis DEURVEILHER

Master IMAGINE
Faculté des Sciences
Université de Montpellier

November 20, 2024

1 Introduction

Cette semaine, nous avons implémenté de nouvelles méthodes classiques, une nouvelle méthode d'évaluation sans référence, et nous avons effectué une analyse des résultats obtenus par méthode classique ainsi qu'un début de recherche pour les méthodes utilisant des CNN.

Voici les **nouvelles méthodes classiques** implémentées :

- Filtre médian
- Débruitage par patches

Analyse des résultats :

- Courbes d'évolution de l'amélioration du PSNR en fonction du bruit sur les images ou des sigmas des filtres utilisés
- Nouvelle méthode d'évaluation sans référence : NIMA

2 Nouvelles Méthodes de Débruitage

Dans un premier temps, nous avons implémenté deux nouvelles techniques de débruitage : le filtre médian et le débruitage par patch.

2.1 Filtre Médian

Le filtre médian est relativement simple : il consiste, pour chaque pixel, à prendre la valeur médiane d'un certain nombre de pixels voisins (dans un rayon). Ce filtre est particulièrement efficace contre le bruit sel et poivre, car il élimine les valeurs aberrantes.



Figure 1: Image bruitée avec sel et poivre (gauche) et image débruitée avec un filtre médian (droite)

2.2 Débruitage par patch

Le débruitage par patch consiste à utiliser tous les pixels de l'image, pondérés par une fonction de poids, pour réaliser le débruitage. Ce type de débruitage permet de mieux conserver les détails de l'image originale. Un inconvénient majeur de ce type de débruitage est qu'il est extrêmement lent pour les grandes images, étant donné qu'il faut itérer sur tous les pixels pour chaque pixel de l'image (le filtre a une complexité temporelle de $\mathcal{O}(n^2)$).

3 Analyse des Résultats

Une fois les méthodes de bruitage et de débruitage mises en place, nous avons voulu pousser l'analyse des résultats afin de trouver les paramètres optimaux pour nos filtres et l'impact de ces filtres en fonction du niveau de bruit.

3.1 Analyse de l'Impact du Niveau de Bruit sur la Qualité de Débruitage

Dans un premier temps, nous avons analysé l'impact du niveau de bruit sur la qualité du débruitage. Pour cela, nous avons fait varier le gain de PSNR entre l'image bruitée et l'image de base ainsi que l'image débruitée par rapport à l'image originale.

Voici ce que l'on obtient pour les différents types de bruits :

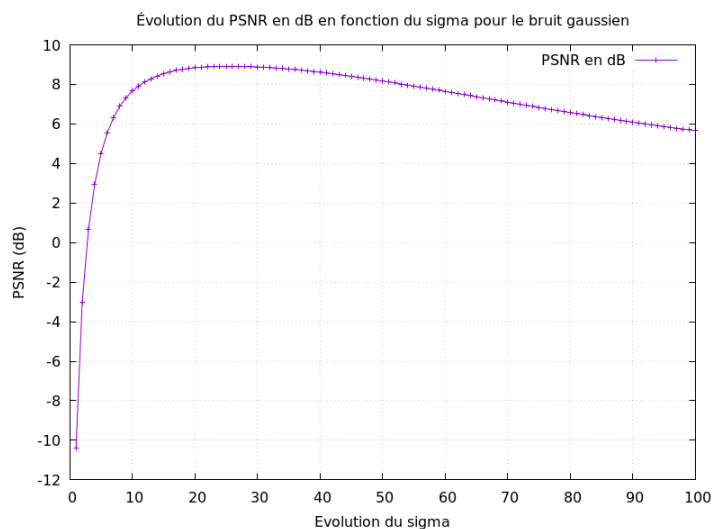


Figure 2: Évolution du gain de PSNR en fonction du sigma du bruit gaussien

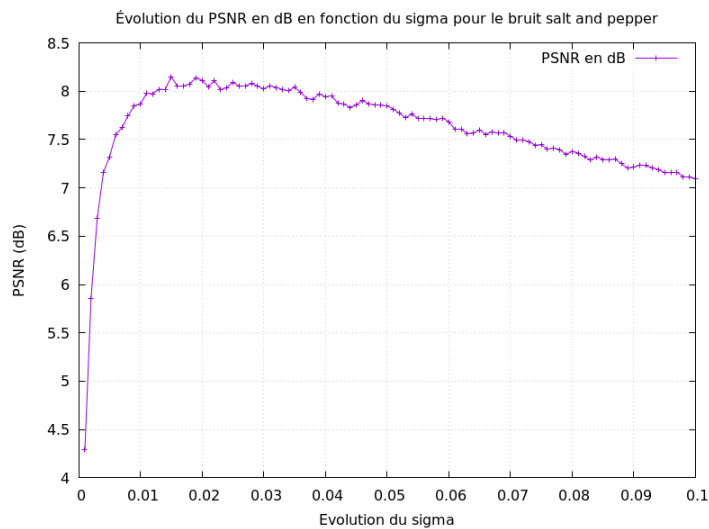


Figure 3: Évolution du gain de PSNR en fonction du sigma du bruit SaltAndPepper

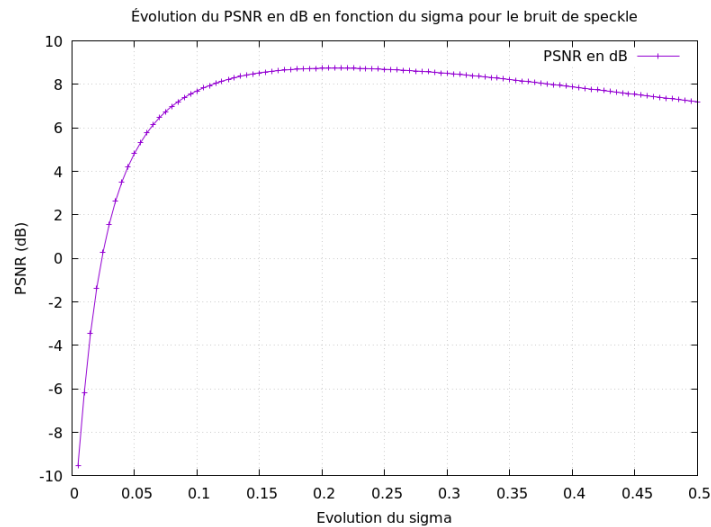


Figure 4: Évolution du gain de PSNR en fonction du sigma du bruit de Speckle

Pour chaque type de bruit, on remarque la même tendance, avec une forte amélioration de l'efficacité du filtre si le bruit est faible, jusqu'à une efficacité maximale pour un bruit faible à moyen.

En effet, si le bruit est trop faible, on obtient de mauvais résultats, car le filtre va débruiter une image à peine bruitée, ce qui risque d'empirer l'image en lissant les bords ou en appliquant un flou.

Cependant, pour un bruit plus important, le filtre atteint son efficacité maximale avant de décroître progressivement. Cela montre que les filtres perdent en efficacité lorsque le niveau de bruit devient trop important.

Cependant, cette analyse a été réalisée uniquement avec le filtre gaussien et avec la métrique du PSNR. Cela n'est pas suffisant et, d'ailleurs, visuellement, cela n'est pas toujours représentatif. Nous devons approfondir ces analyses avec d'autres métriques comme le SSIM ou NIMA, et pour chaque méthode, une fois notre base de données d'images construite. Néanmoins, cela montre une tendance générale dont nous pouvons tirer des conclusions.

3.2 Analyse de l'Efficacité des Filtres en Fonction de leurs Sigma Respectifs

Dans un second temps, nous avons analysé l'efficacité de chaque filtre selon le sigma afin de déterminer les meilleurs paramètres pour leur application.

Voici les résultats obtenus :

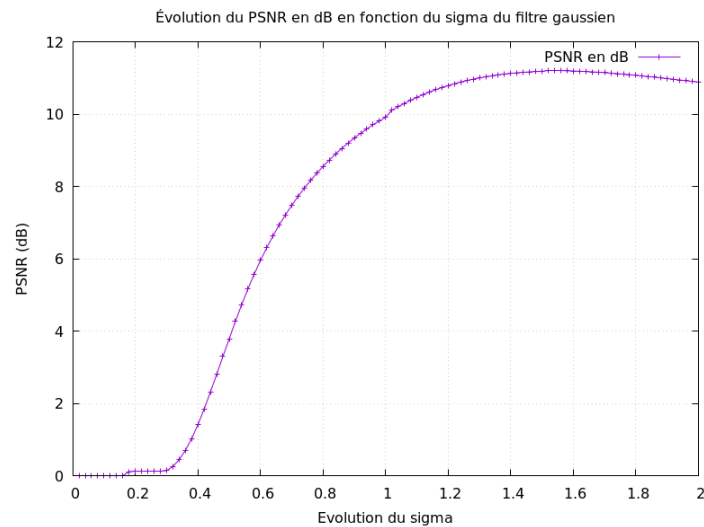


Figure 5: Évolution du gain de PSNR en fonction du sigma du filtre gaussien

Pour le filtre gaussien, on remarque un gain de PSNR croissant avec l'augmentation du sigma. Cependant, avec un sigma trop élevé, le gain de PSNR diminue à nouveau. Cela est cohérent, car le filtre gaussien a tendance à flouter l'image : un sigma trop élevé détériore donc l'image en appliquant un flou excessif. On peut ainsi observer que le sigma optimal est aux alentours de 1.54.

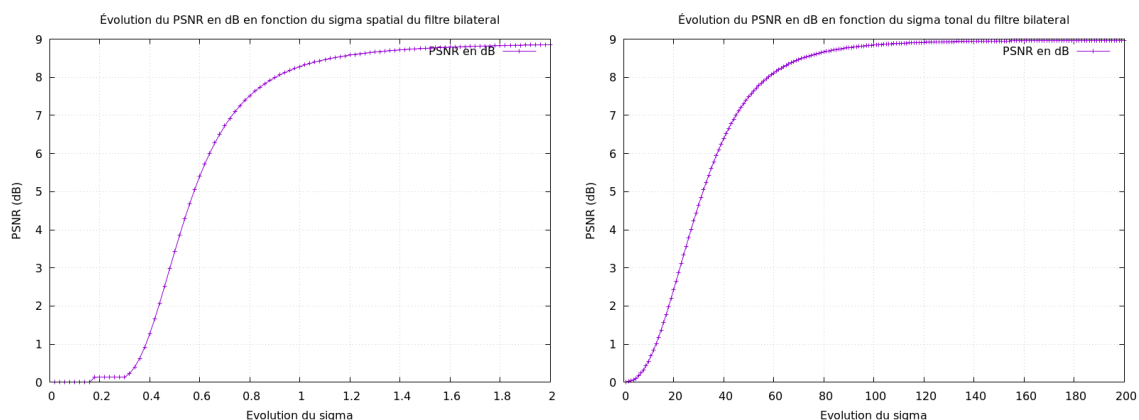


Figure 6: Évolution du gain de PSNR en fonction du sigma spatial (gauche) et tonal (droite) du filtre bilatéral

Pour le filtre bilatéral, on observe une tendance similaire, sauf qu'un sigma élevé ne détériore pas le gain de PSNR mais plafonne à une certaine valeur. On peut ainsi observer que les valeurs de sigma optimales sont d'environ 1.2 pour le sigma spatial et environ 120 pour le sigma tonal.

3.3 Nouvelle Méthode d'Évaluation sans Référence : NIMA

NIMA est une mesure d'évaluation de la qualité "esthétique" d'une image générée par un réseau de neurones. Contrairement au PSNR et au SSIM, NIMA ne compare pas une image à une référence pour obtenir un score de différence. NIMA attribue un score de 1 à 10 représentant la valeur esthétique d'une image grâce à un réseau de neurones.

Notre implémentation actuelle utilise un réseau pré-entraîné que nous utilisons pour calculer la NIMA de l'image filtrée et de l'image originale.

Malheureusement, il semblerait qu'il y ait un bug dans notre approche, car la valeur de la NIMA semble fluctuer lorsqu'on la calcule plusieurs fois sur la même image. Par exemple, en calculant la valeur NIMA pour une image I une première fois, nous obtenons 4.74. En calculant la même valeur une deuxième fois pour la même image I , nous obtenons 5.68. Cela est étrange, car les données d'entrée sont identiques, et le réseau devrait fournir la même valeur. Nous continuerons à chercher une solution dans les jours à venir.

4 Planning Prévisionnel

Dans les jours à venir, nous aimerions commencer à travailler sur l'approche réseau de neurones. Nous avons déjà identifié un modèle appelé FFDNet, qui semble bien adapté à nos besoins. Une version pré-entraînée de ce modèle est disponible en ligne. Notre objectif initial sera d'utiliser cette version pour une première comparaison avec notre propre version, que nous entraînerons sur nos données.

De plus, nous avons prévu de commencer à créer notre base d'images. Pour cela, nous avons repéré un lanceur de rayon paramétrable open source appelé "Tungsten". Ce lanceur de rayon nous permettra de générer des images avec différents niveaux de bruit en faisant varier le nombre d'échantillons par pixel.