

Débruitage d'images

BONBON Adam, HUNOT-MARTIN Alaric, JEAN Louis

Faculté des Sciences, Université de Montpellier

13 novembre 2024



Illustration des types de bruit



Figure: Image originale



Figure: Bruit sel et poivre



Figure: Bruit gaussien

- **Bruit sel et poivre** : pixels blancs et noirs aléatoires
- **Bruit gaussien** : ajoute des variations aléatoires suivant une distribution normale

Implémentation des méthodes classiques de débruitage

- **Filtre médian :**

- Très efficace pour le bruit impulsionnel (sel et poivre)
- **PSNR amélioré :** de 20.1 dB à 33.1 dB pour le bruit sel et poivre

- **Transformée en ondelettes de Haar :**

- Décompose l'image en sous-bandes de fréquence pour un débruitage ciblé
- **PSNR amélioré :** de 20.1 dB à 26.0 dB pour le bruit gaussien

- **Variation totale :**

- Minimise la variation globale tout en conservant les bords nets
- **PSNR amélioré :** de 20.1 dB à 29.6 dB pour le bruit gaussien



De gauche à droite : filtre médian, ondelettes de Haar, variation totale

Implémentation des méthodes classiques de débruitage

- **Filtre moyeneur** : Effectue une convolution entre une image et un filtre moyeneur.
 - Réduit le bruit, mais introduit un flou dans l'image
 - **PSNR amélioré** : de 20.1 dB à 24.4 dB
- **Transformée de Fourier avec filtre passe-bas gaussien** :
 - Conserve uniquement les basses fréquences
 - **PSNR amélioré** : de 20.1 dB à 28.7 dB
- **Filtre basé sur la densité de probabilité** :
 - Pondère les pixels voisins selon une distribution normale
 - **PSNR amélioré** : de 20.1 dB à 24.2 dB

Test avec du **bruit gaussien** :



- **Filtrage de Wiener :**

- Optimise le rapport signal/bruit tout en préservant les détails.
- **PSNR amélioré** : de 20.1 dB à 28.93 dB.

- **Non-Local Means (NLM) :**

- Exploite la redondance des motifs pour estimer les pixels.
- **PSNR amélioré** : de 20.1 dB à 29.9 dB.

- **BM3D (Block-Matching and 3D Filtering) :**

- Regroupe des blocs similaires pour le filtrage collaboratif.
- **PSNR amélioré** : de 20.1 dB à 31.84 dB.

- **Filtre guidé :**

- Préserve les bords tout en lissant les régions homogènes.
- **PSNR amélioré :** de 20.1 dB à 25.20 dB.

- **Filtre bilatéral non adaptatif :**

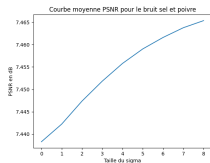
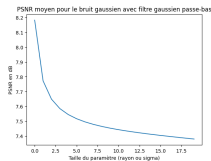
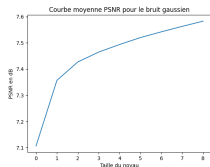
- Lisse tout en préservant les contours en utilisant une fenêtre locale.
- **PSNR amélioré :** de 20.1 dB à 28.54 dB.

- **Filtre bilatéral adaptatif :**

- Ajuste dynamiquement les paramètres en fonction des caractéristiques locales.
- **PSNR amélioré :** de 20.1 dB à 28.70 dB.

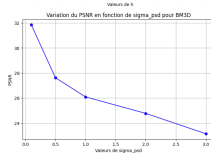
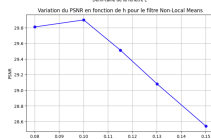
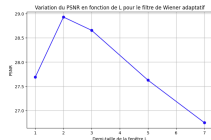
Comparaison des méthodes classiques avec une courbe PSNR moyenne

- **Filtre moyenneur :**
Amélioration modérée
- **Transformée de Fourier :**
PSNR décroît avec
l'augmentation du rayon
- **Densité de probabilité :**
Amélioration très lente



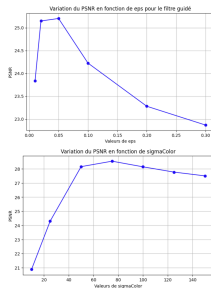
Comparaison des méthodes classiques avec une courbe PSNR moyenne

- **Filtre Wiener** : PSNR optimal à *demi-taille de fenêtre* $L = 2$
- **Non-Local Means (NLM)** : PSNR optimal pour $h = 0.09$
- **BM3D** : PSNR décroît avec l'augmentation de σ_{psd}



Comparaison des méthodes classiques avec une courbe PSNR moyenne

- **Filtre Guidé :**
PSNR optimal pour $\epsilon = 0.05$
- **Filtre Bilatéral Simple :**
PSNR optimal pour $\sigma_{\text{Color}} = 60$



Première exploration de CGNet

- Utilisation de **CGNet** pour le débruitage basé sur un modèle d'apprentissage profond pré-entraîné.
- Dataset utilisé : **SIDD** (Smartphone Image Denoising Dataset).
- Principaux défis rencontrés :
 - **Ressources limitées** : Difficulté à entraîner le modèle localement.
 - **Délais d'entraînement** : Plus d'une semaine nécessaire pour obtenir des résultats concrets sur notre machine personnel sans modifier les paramètres.
- Perspectives : Ajuster et optimiser l'entraînement pour le rendre accessible et réduire les délais.

Réimplémentation basique du modèle CGNet

Objectif : Développer une version du modèle **CGNet** pour le débruitage d'images en utilisant le dataset **BSDS500**.

- Le modèle utilise des **convolutions profondes** pour capturer des informations contextuelles à différentes échelles.
- Utilisation de **blocs résiduels partiels** pour améliorer la propagation des gradients.
- Expérimentation avec des **fonctions d'activation avancées** pour optimiser la capture des caractéristiques.

Dataset utilisé : BSDS500

Présentation du dataset

- Dataset : **BSDS500** (Berkeley Segmentation Dataset).
- Contient environ **200 images** d'animaux, paysages et environnements naturels.
- Chaque image est annotée pour une meilleure compréhension des contours et des segments.

Pourquoi ce dataset ?

- Choisi pour sa **diversité** et sa compatibilité avec l'architecture CGNet.
- Garantit une **variété de scènes et textures**, utile pour un modèle de débruitage général.
- Contrairement à d'autres datasets thématiques, il évite de biaiser les résultats en offrant une meilleure diversité d'images.

Performances et résultats préliminaires

Résultats actuels :

- Amélioration du **PSNR** (Peak Signal-to-Noise Ratio) allant de 4 à 10 dB par rapport aux images bruitées initiales.
- En moyenne, le PSNR passe de 20 dB (image bruitée) à 28 dB (image débruitée).

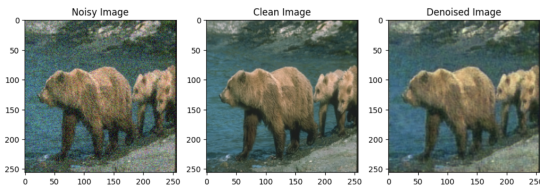


Figure: Résultats sur un exemple du dataset BSDS500

PSNR bruité = 20.10 dB, PSNR débruité = 28.11 dB

Réseau encore en développement :

- L'entraînement du modèle n'est pas encore **finalisé**.
- Les **hyperparamètres** n'ont pas encore été entièrement optimisés pour des performances maximales.
- Potentiel d'amélioration avec un ajustement plus précis des paramètres d'apprentissage.

Features manquantes à implémenter prochainement :

- Cependant, il reste des différences par rapport à un CGNet complet :
 - ① Manque de **modules contextuels** pour l'agrégation d'informations à différentes échelles.
 - ② Pas encore de **supervision multi-échelle** pour guider l'apprentissage.
 - ③ Absence de **convolutions dilatées**, limitant la capacité à capturer des détails globaux.

Réécriture des méthodes classiques et premières comparaisons

Objectif : comparer notre implémentation de CGNet avec des méthodes classiques de débruitage

Méthodes réimplémentées (adaptées pour les images couleur) :

- Filtre médian
- Transformation en ondelettes de Haar
- BM3D

Pipeline de test :

- Bruitage d'une image (bruit gaussien, sel et poivre)
- Débruitage avec chaque méthode + CGNet
- Évaluation via PSNR

Observations :

- CGNet performant, mais améliorable
- Filtre médian efficace pour le bruit sel et poivre
- BM3D à revoir (performances inférieures)

Résultats de la comparaison (exemple avec *lena.ppm*)

Méthode	PSNR (dB)	
	Bruit gaussien	Bruit sel et poivre
CGNet	25.8	25
Filtre médian	25.9	30
Ondelettes de Haar	26	21
BM3D	20.2	18.1

Table: Comparaison du PSNR entre l'image originale et débruitée

Observations :

- Le filtre médian est le meilleur pour le bruit sel et poivre
- CGNet montre un potentiel prometteur même avec une implémentation basique

Développement de l'interface graphique

Objectif : simplifier l'utilisation des méthodes de débruitage

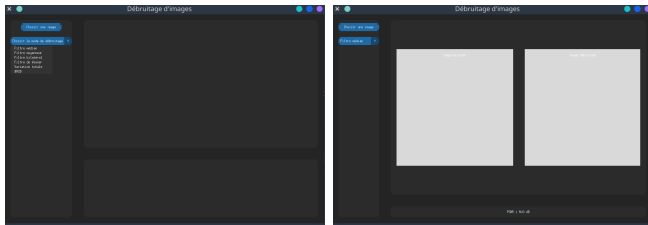
Fonctionnalités actuelles :

- Sélection d'images bruitées ou neutres
- Choix de la méthode de débruitage

Fonctionnalités à venir :

- Ajout d'options pour bruiteur une image avec différents types de bruit
- Lancement du débruitage avec calcul des métriques (PSNR, SSIM, etc.)

Aperçu de l'application :



Merci pour votre attention !

Des questions ?