

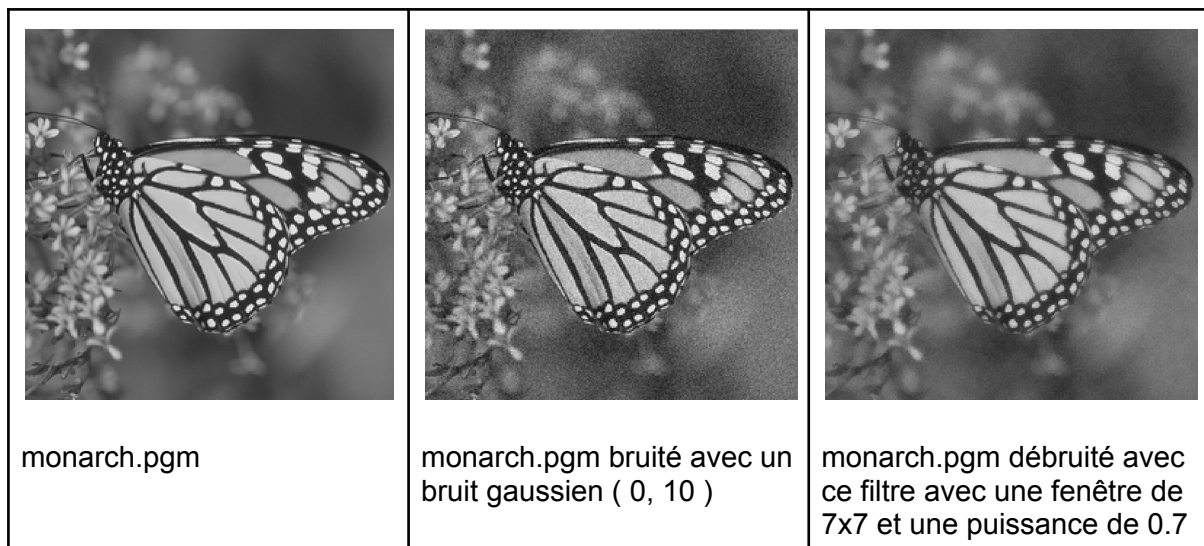
[HAI927I] Projet Image - Compte Rendu n°4

Projet #12.3 : Débruitage d'images

1. Filtres de débruitage




Nous avons fini d'implémenter 2 filtres de "faits maison" :

- Le premier reprend le filtre moyenneur et pondère la somme des valeurs de pixels par la distance entre la couleur du pixel courant et le pixel voisin, mise à la place *power* qui est un argument. Plus la couleur du pixel voisin est proche de la couleur du pixel courant, plus le poids de cette couleur est fort et vice-versa.



PSNR : 26.977840

- Le deuxième utilise la carte des gradient. On applique un filtre moyennneur ou un filtre gaussien centré réduit seulement aux pixels connexes de la même couleur dans la carte de gradient seuillée.



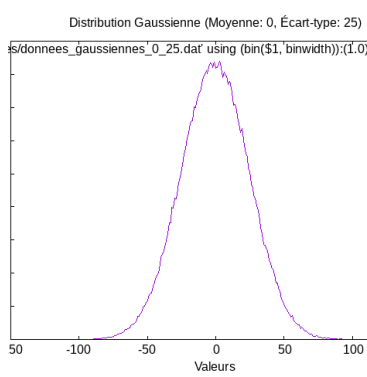
		
monarch.pgm	monarch.pgm bruité avec un bruit gaussien (0, 10)	monarch.pgm débruité avec ce filtre avec une fenêtre de 3x3, un seuil de 80 et un filtre Moyennneur

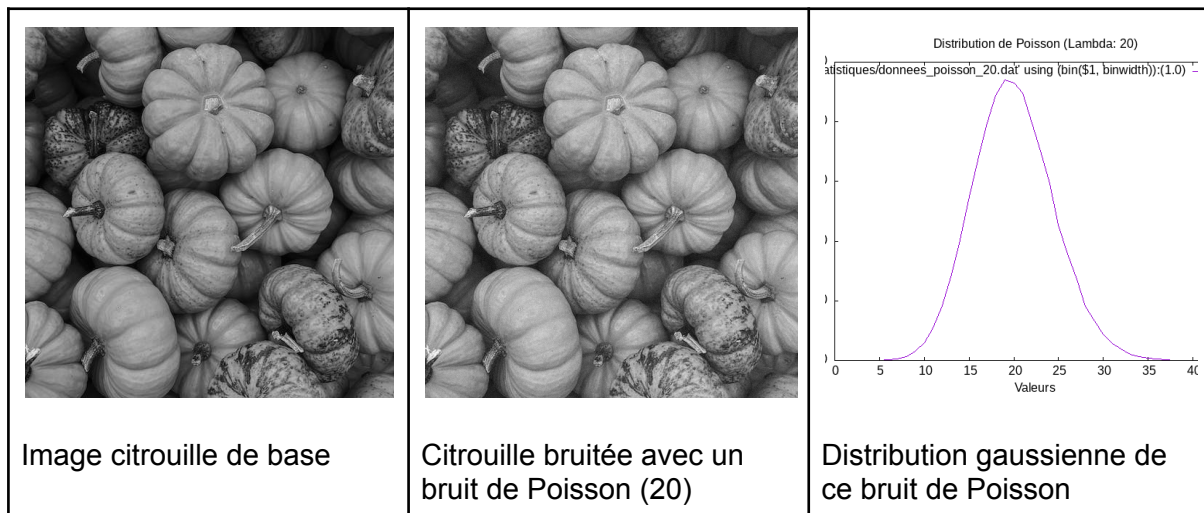
PSNR : 27.771883

2. Méthodes de vérifications de la validité des bruits

Nous avons tracé les distributions Gaussiennes et de Poisson pour vérifier la validité de nos bruits.

Cela nous a permis de corriger nos images avec du bruit de Poisson.

		
Image citrouille de base	Citrouille bruitée avec un bruit gaussien (0, 25)	Distribution gaussienne de ce bruit gaussien



On voit que l'image avec du bruit de Poisson est plus claire que l'image de base.

3. Méthodes d'analyse de résultats

On a créé deux scripts afin de générer des données d'analyse entre une image intacte et une image débruitée sur de grandes bases de données. Le premier script GenImg.sh permet de :

- Créer un dossier avec des images bruitées à partir d'un dossier contenant des images (pgm ou ppm) et un type de bruit particulier
- Créer un dossier avec des images débruitées à partir d'un dossier contenant des images (pgm ou ppm) et un mode de débruitage particulier
- Evaluer le PSNR, SNR, SSIM et RMSE entre deux images du même nom dans deux dossiers différents et stocker les résultats dans un nouveau fichier texte
- Supprimer un dossier et son contenu

Le deuxième script GenData.sh utilise le premier afin d'enchaîner plusieurs opérations de création, d'évaluation et de suppression d'images. C'est ce dernier qu'il faudra modifier pour renseigner quelles bases de données, quelles méthodes de bruitage, quelles méthodes de débruitage et quels arguments utiliser dans la génération de données.

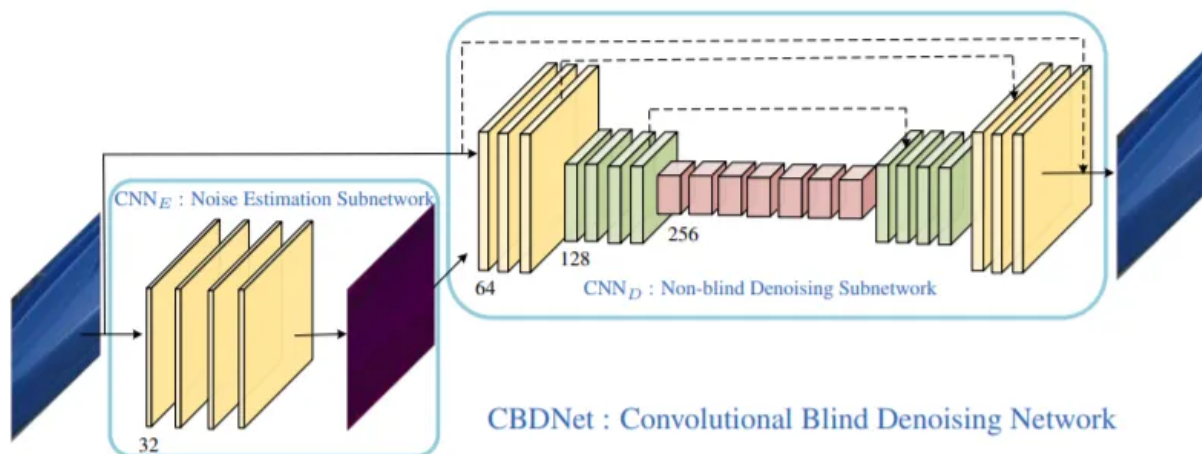
Nous avons déjà récupéré deux bases de données sur internet : une base de données d'image pgm de taille 256 par 256 pixels contenant des images de l'extérieur et une base de données d'image ppm de taille 128 par 128 pixels contenant des images de fleurs.

4. Mise en place du CNN

Nous avons commencé à mettre en place un CBDNet sur un google colab en nous basant sur l'implémentation faite sur ce site web :

<https://medium.com/analytics-vidhya/image-denoising-using-deep-learning-dc2b19a3fd54>

Voici le fonctionnement du CBDNet :



La première partie du réseau consiste à estimer le bruit de l'image et dans la seconde partie à débruiter l'image donnée en entrée.

Pour l'instant nous avons commencé l'entraînement en nous basant sur des images en couleurs que nous avons bruitées avec nos algorithmes. Nous l'avons entraîné sur 10 epochs avec un batch_size de 10 sur 13 images en couleur.

Voici quelques exemples de résultats obtenus pour l'instant :

Image bruitée



Image débruitée



Image bruitée



Image débruitée



Image bruitée



Image débruitée



Nous n'avons pas encore de résultat satisfaisant, nous allons donc continuer à l'améliorer (notamment en l'entraînant sur plus de données).

5. Objectifs de la semaine

- Générer les données pour mesurer l'efficacité de chaque méthode de débruitage
- Améliorer notre CBDNet pour avoir de meilleurs résultats
- Faire en sorte que notre CBDNet puisse corriger le bruit sur des images en niveaux de gris
- Commencer à mettre en place l'application