



UNIVERSITÉ DE MONTPELLIER  
FACULTÉ DES SCIENCES

MASTER 2 - IMAGINE

---

## Projet Image, sécurité et deep learning

*Sujet 12 - Débruitage ou restauration d'images par CNN*

*CR3 - Approfondissement des recherches sur les méthodes avec CNN  
et début d'implémentation de la méthode sans CNN*

[HAI918I]

---

*BES Jean-Baptiste  
COMBOT Evan*

03/11/2024

[Lien GitHub de notre projet image](#)

# 1 Approfondissement de nos recherches

## 1.1 CNN + transformation en ondelettes

Dans l'article *Multi-Level Wavelet-CNN for Image Restoration* [1] est décrit une méthode (MWCNN) qui combine des transformations en ondelettes avec des réseaux de neurones convolutifs. Même si l'article est focalisé dans le débruitage d'images, cela peut continuer une approche dans la restauration d'images anciennes.

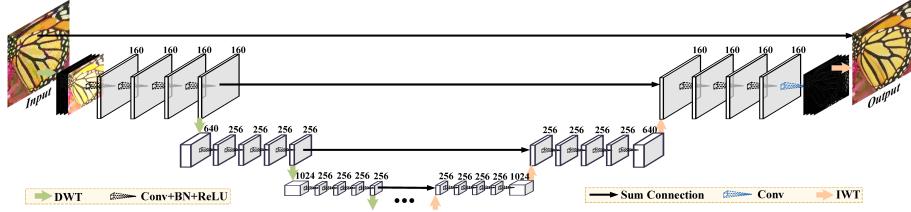


FIGURE 1 – Architecture Multi-level wavelet-CNN

Le principe est d'appliquer une décomposition par ondelettes afin de traiter les différentes sous-bandes de différentes fréquences indépendamment dans le CNN. Pour ensuite, reconstituer l'image débruitée à partir des sous-bandes obtenues après les convolutions. Cette architecture est composée de 24 couches convolutives, qui peuvent être étendu par tranche de 4 couches.

Cette méthode permet de conserver les détails dans l'image grâce à la transformation en ondelettes tout en éliminant le bruit et les artéfacts.

## 1.2 AdaptBIR

L'article *AdaptBIR : Adaptive Blind Image Restoration with latent diffusion prior for higher fidelity* [2] présente une méthode similaire à la méthode *Deep Latent Space Translation*, énoncée dans le CR2.

Comme pour la méthode DLSP, AdaptBIR utilise deux auto-encodeurs variationnels afin d'extraire les informations pertinentes de l'image. Ainsi qu'une architecture U-Net pour restaurer l'image corrigée.

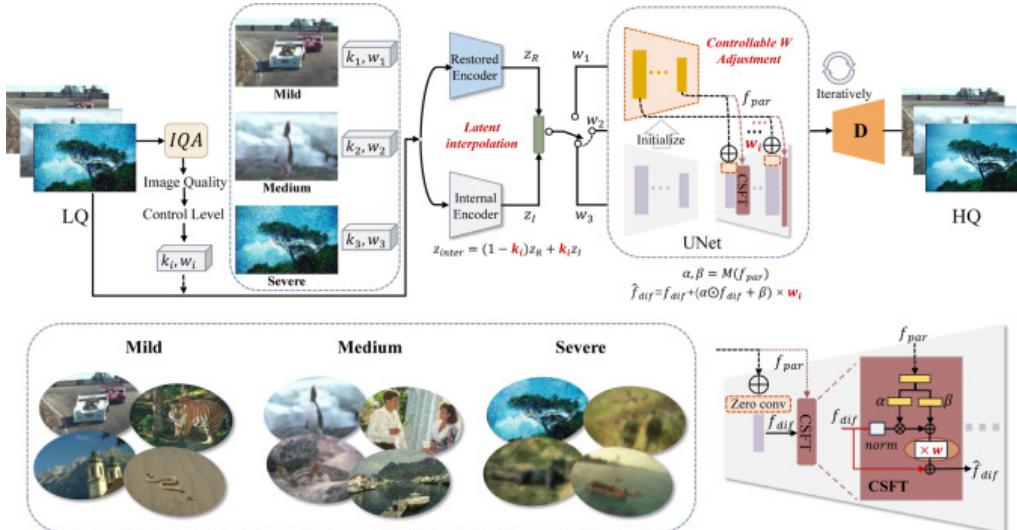


FIGURE 2 – Framework d'AdaptBIR

Ce pour quoi nous avons trouvé pertinent de répertorier cet article dans cet état de l'art, c'est par rapport à l'évaluation du degré de dégradation. Dans cet article, ils présentent trois catégories de dégradation : **Mild**, **Medium** et **Severe**.

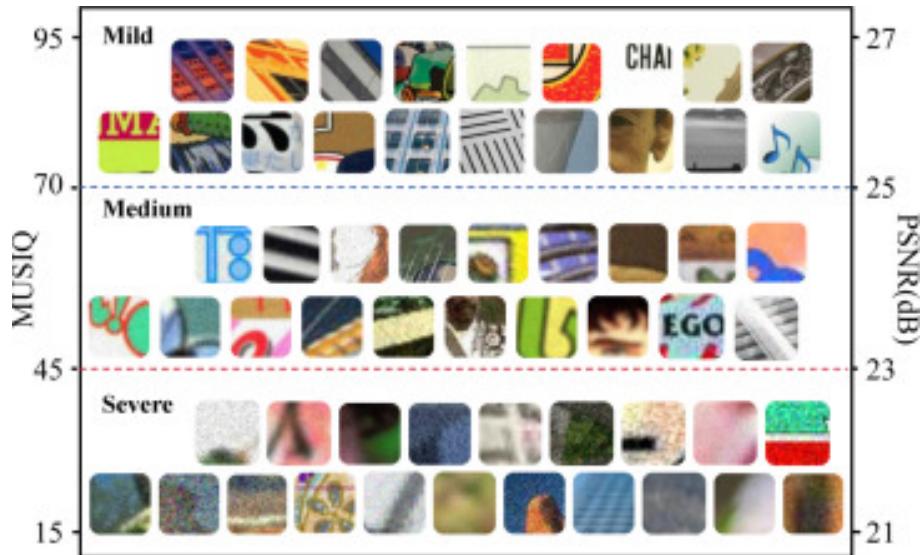


FIGURE 3 – Les catégories de dégradation

Cette étape de classification est cruciale dans le pipeline car elle permet d'attribuer des paramètres pour l'étape d'extraction d'informations de l'image. Cela permet de s'adapter à différents niveaux de dégradation et donc d'améliorer la robustesse de cette méthode. Mais il y a un inconvénient, s'il y a une erreur lors de la classification de la dégradation alors la correction de l'image en sera fortement perturbée.

## 2 Processus (sans CNN)

Pour commencer à programmer le processus de restauration et de colorisation d'images sans utiliser de CNN, nous nous sommes ici consacrés aux éléments principaux : la restauration par inpainting et la recolorisation.

### 2.1 Recolorisation

#### 2.1.1 K-Means

Tout d'abord, on initialise une image en couleur, on la transforme en niveaux de gris, puis on applique un K-Means avec un certain nombre de clusters. La couleur assignée à chaque pixel d'un cluster est la couleur moyenne des pixels de ce cluster.

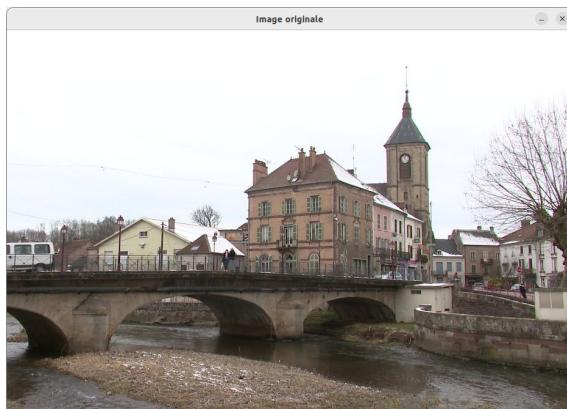


FIGURE 4 – Image originale



FIGURE 5 – Image recolorisée (K = 30)



FIGURE 6 – Image originale



FIGURE 7 – Image recolorisée ( $K = 30$ )



FIGURE 8 – Image originale

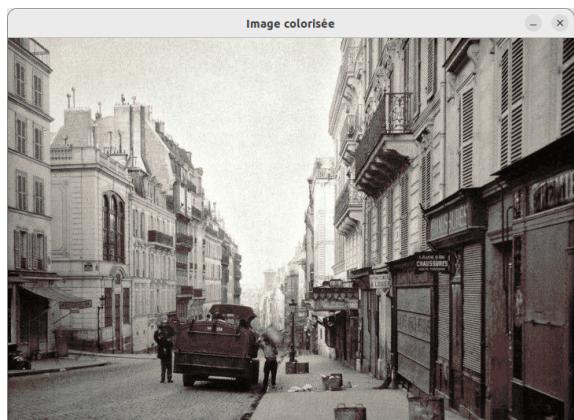


FIGURE 9 – Image recolorisée ( $K = 30$ )



FIGURE 10 – Image originale



FIGURE 11 – Image recolorisée ( $K = 30$ )



FIGURE 12 – Image originale

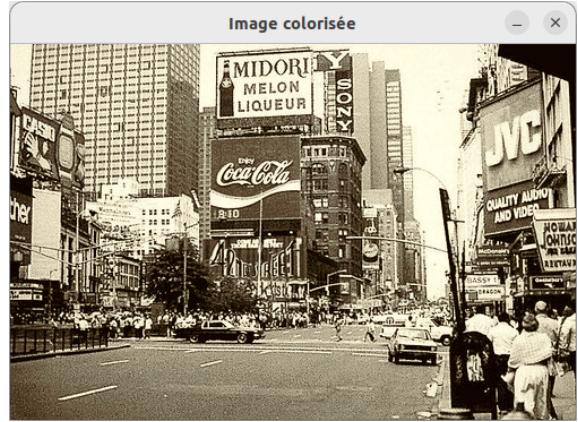


FIGURE 13 – Image recolorisée ( $K = 30$ )

Cette méthode donne des résultats corrects. Cependant, elle nécessite une image colorisée dès le départ, ce qui n'est vraiment pas idéal dans le cas de la restauration d'images anciennes où la couleur n'est pas présente dans les très vieilles photos.

### 2.1.2 Histogramme

Nous avons testé une méthode basée sur les histogrammes (RGB) d'une image de référence, que nous transférons à une image originale que nous transformons en niveaux de gris, mais qui n'est pas très concluante.



FIGURE 14 – Image originale



FIGURE 15 – Image de référence



FIGURE 16 – Image originale

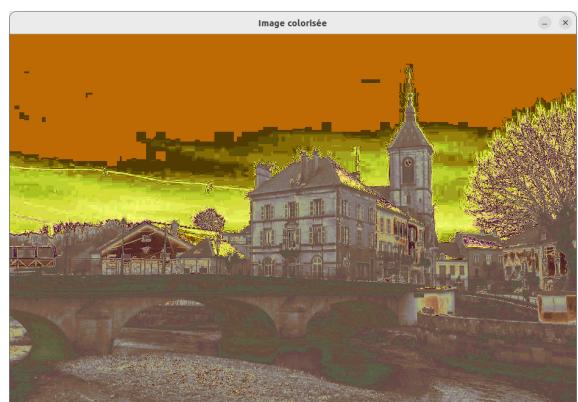


FIGURE 17 – Image obtenue



FIGURE 18 – Image de référence

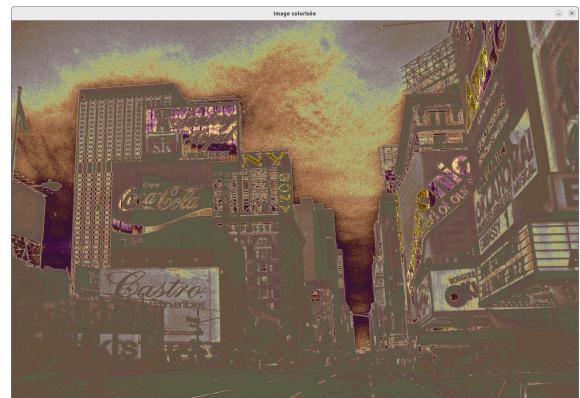


FIGURE 19 – Image obtenue

## 2.2 Restauration (Inpainting)

Concernant l'inpainting, nous avons commencé à l'implémenter. Cependant, dans le code proposé, il est obligatoire de charger une image binaire représentant l'endroit où se trouvent les dégradations dans l'image, ce qui n'est pas notre objectif. Notre but est de détecter automatiquement, ou au moins de pouvoir sélectionner les zones de dégradations afin que l'algorithme d'inpainting puisse faire son travail.



FIGURE 20 – Image originale

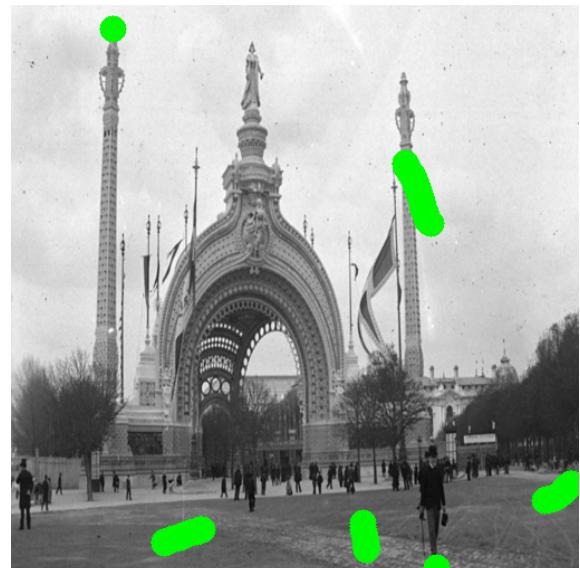


FIGURE 21 – Image obtenue après restauration



FIGURE 22 – Image originale



FIGURE 23 – Image obtenue après restauration

L'exemple ci-dessus provient d'un logiciel qui permet de visualiser, au fil des itérations, la restauration de l'image. La taille du patch est de 4, c'est-à-dire que l'image va être restaurée 4 pixels par 4 pixels. On remarque tout de même quelques incohérences dans la reconstruction. Plus on réduit la taille du patch, meilleure est la restauration.

## Références

- [1] Pengju LIU et al. “Multi-Level Wavelet-CNN for Image Restoration”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Juin 2018. URL : [https://openaccess.thecvf.com/content\\_cvpr\\_2018\\_workshops/papers/w13/Liu\\_Multi-Level\\_Wavelet-CNN\\_for\\_CVPR\\_2018\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w13/Liu_Multi-Level_Wavelet-CNN_for_CVPR_2018_paper.pdf).
- [2] Yingqi LIU et al. “AdaptBIR : Adaptive Blind Image Restoration with latent diffusion prior for higher fidelity”. In : *Pattern Recognition* 155 (2024), p. 110659. ISSN : 0031-3203. DOI : <https://doi.org/10.1016/j.patcog.2024.110659>. URL : <https://www.sciencedirect.com/science/article/pii/S0031320324004102>.