



UNIVERSITÉ DE MONTPELLIER
FACULTÉ DES SCIENCES

MASTER 2 - IMAGINE

Projet Image, sécurité et deep learning

Sujet 12 - Débruitage ou restauration d'images par CNN

CR4 - Processus sans CNN terminé et début de la méthode avec CNN

[HAI918I]

*BES Jean-Baptiste
COMBOT Evan*

10/11/2024

[Lien GitHub de notre projet image](#)

1 Creation de la base de donnees d'images

Pour pouvoir tester les methodes classiques (sans CNN) et les methodes avec CNN, nous avons constitu une petite base de donnees compose d'images en noir et blanc datant de la perode comprise entre le 19^e sicle et la moitie du 20^e sicle. Les photographies, qui sont anciennes, sont a la fois en noir et blanc et peuvent parfois presenter des degradations.

Ces images proviennent des archives nationales, dont le ministre de la Culture permet la visualisation via ce lien : <https://pop.culture.gouv.fr/search/list?base=%5B%22Photographies%20%28M%C3%A9moire%29%22%5D>



FIGURE 1 – Image de la BDD

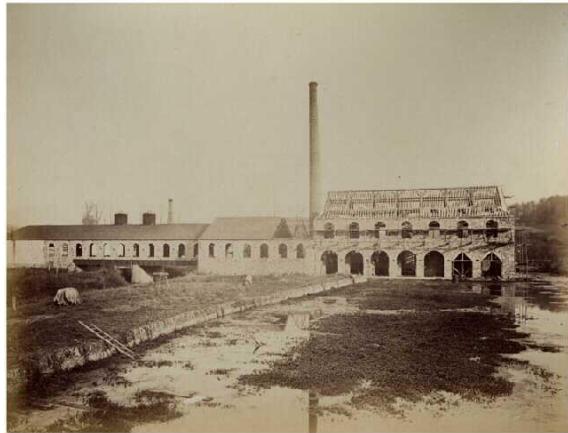


FIGURE 2 – Image de la BDD



FIGURE 3 – Image de la BDD



FIGURE 4 – Image de la BDD

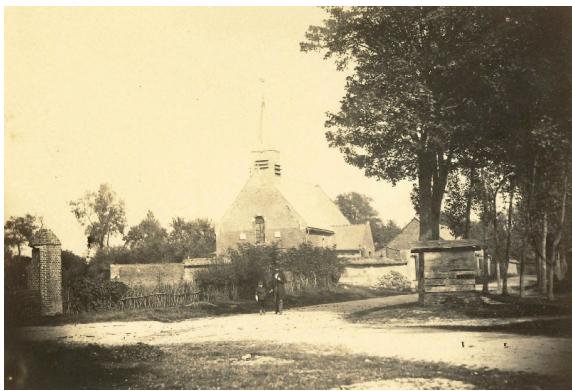


FIGURE 5 – Image de la BDD

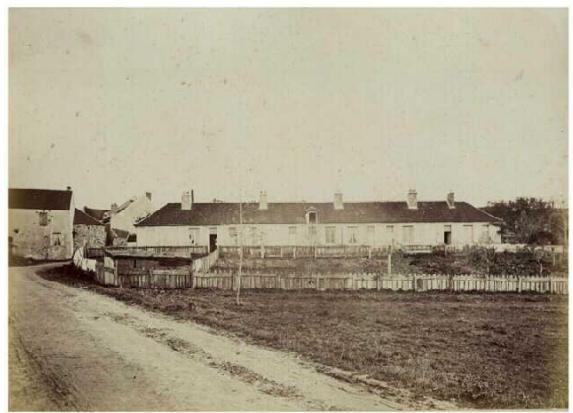


FIGURE 6 – Image de la BDD



FIGURE 7 – Image de la BDD



FIGURE 8 – Image de la BDD



FIGURE 9 – Image de la BDD



FIGURE 10 – Image de la BDD

Les images ci-dessus ne représentent qu'une partie de la base de données, il y en a d'autres. Les images de cette base de données ne sont pas toutes dégradées, bien qu'il y en ait, mais il y a aussi des images sur lesquelles nous pourrons tester différentes parties de la restauration (le débruitage, l'inpainting, la colorisation) ainsi que la super-résolution.

2 Avancements et fin du processus classique (sans CNN)

2.1 Floutage

Pour la phase de prétraitement, nous avons implémenté et testé 4 filtres. L'objectif de cette étape est d'atténuer les bruits afin d'aider le traitement de l'image.

2.1.1 Filtre Gaussien



FIGURE 11 – Image Originale



FIGURE 12 – Image filtrée avec un noyau 15x15 (Filtre Gaussien)



FIGURE 13 – Image Originale



FIGURE 14 – Image filtrée avec un noyau 7x7 (Filtre Gaussien)

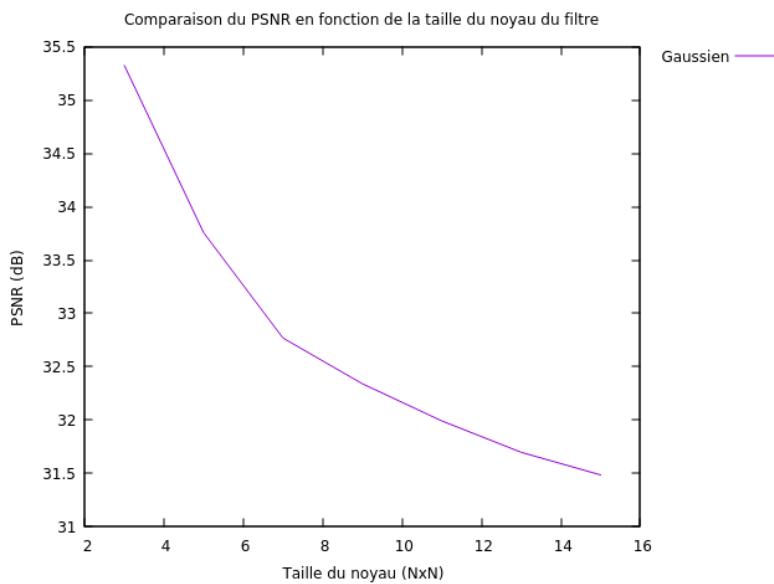


FIGURE 15 – Courbe du PSNR en fonction de la taille du noyau
(Filtre Gaussien)

2.1.2 Filtre Médian



FIGURE 16 – Image Originale



FIGURE 17 – Image filtrée avec un
noyau 3x3 (Filtre Médian)

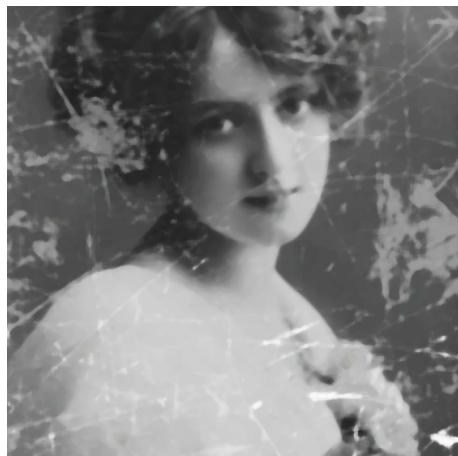


FIGURE 18 – Image filtrée avec un
noyau 7x7 (Filtre Médian)

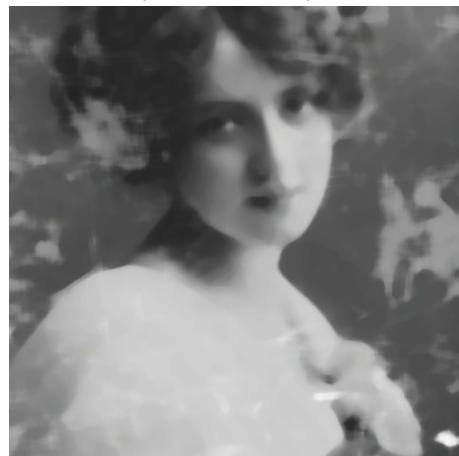


FIGURE 19 – Image filtrée avec un
noyau 15x15 (Filtre Médian)

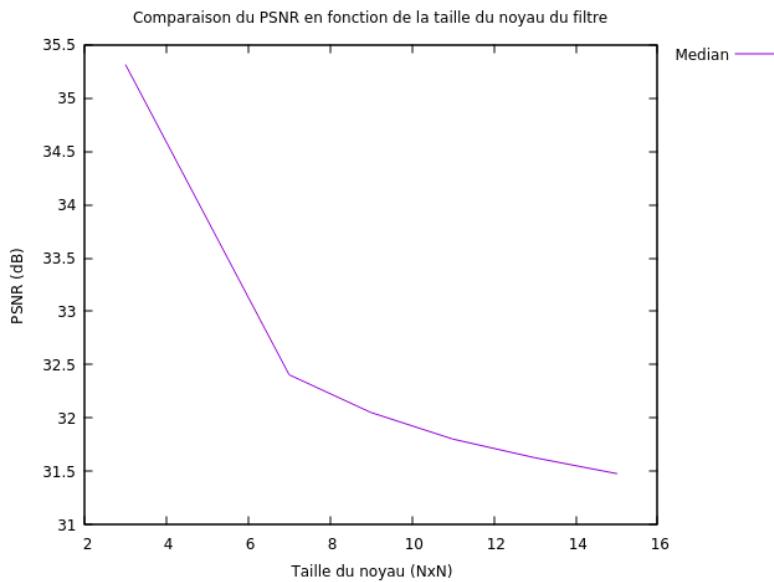


FIGURE 20 – Courbe du PSNR en fonction de la taille du noyau (Filtre Médian)

2.1.3 Filtre Bilatéral



FIGURE 21 – Image Originale



FIGURE 22 – Image filtrée avec un noyau 3x3 (Filtre Bilatéral)



FIGURE 23 – Image filtrée avec un noyau 7x7 (Filtre Bilatéral)



FIGURE 24 – Image filtrée avec un noyau 15x15 (Filtre Bilatéral)

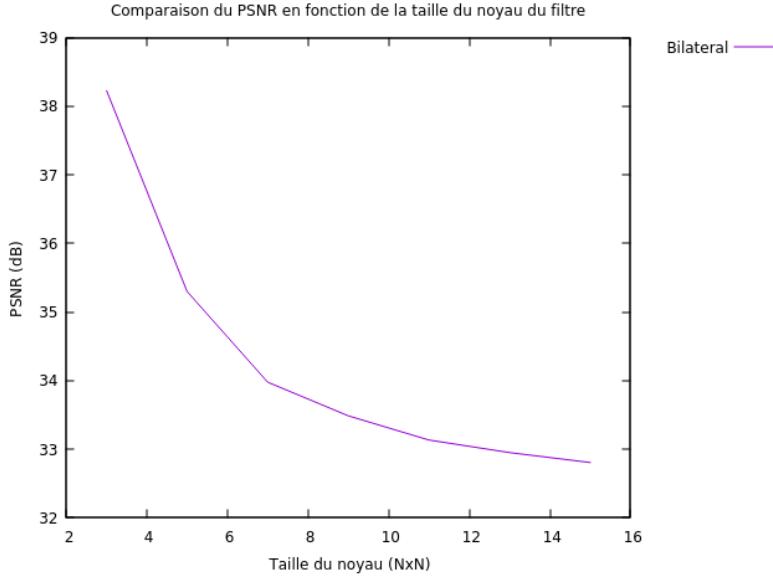


FIGURE 25 – Courbe du PSNR en fonction de la taille du noyau (Filtre Bilatéral)

2.1.4 Comparaison des filtres

Afin de comparer les filtres entre eux, nous avons décidé de calculer le PSNR entre l'image originale et l'image filtrée pour différentes tailles de noyau. Nous avons fait ce choix car cela permet de les comparer sur une caractéristique modulable qu'ils ont en commun.

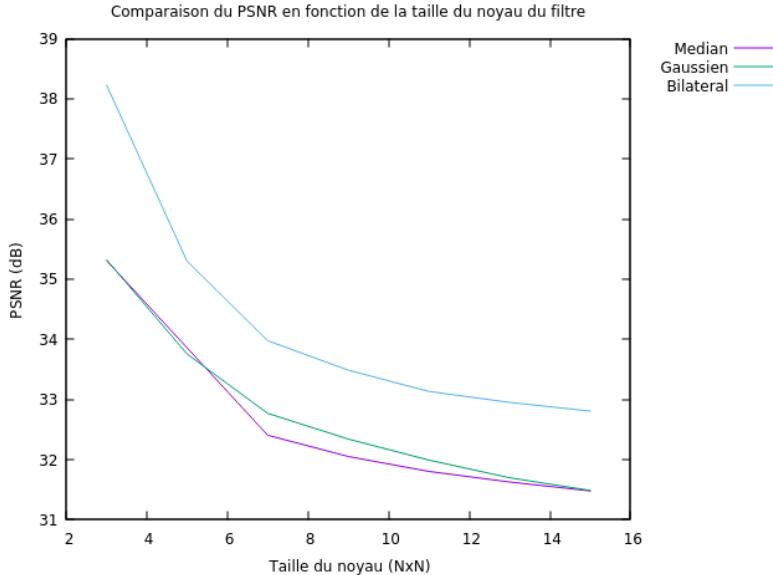


FIGURE 26 – Comparaison de la courbe PSNR/taille noyau de chaque filtre

On remarque que le filtre bilatéral se démarque des deux autres filtres par sa caractéristique à avoir le plus haut PSNR. Mais cela peut aussi impliquer que ce filtre débruite moins fortement que les deux autres. Quant aux filtres Médian et Gaussien, ils démontrent un comportement similaire au niveau de la perte de qualité de l'image.

On peut en conclure que la taille du noyau peut être ajusté afin de traiter différentes intensités de bruit ou de dégradation. De même, le choix du filtre peut dépendre de quelle type d'image on veut traiter. Le filtre bilatéral semble plus adapté à des images faiblement dégradées ou contenant de fins détails. Là où les filtres médian et gaussien pourraient servir à traiter des images fortement dégradées.

2.1.5 Filtre passe-bas

Nous avons voulu tester de filtrer les hautes fréquences dans une image car ces fréquences correspondent en grande partie aux bruits. Pour cela, nous transformons l'image dans le domaine fréquentiel afin de les supprimer à l'aide d'un filtre passe-bas.



FIGURE 27 – Image Originale



FIGURE 28 – Image filtrée

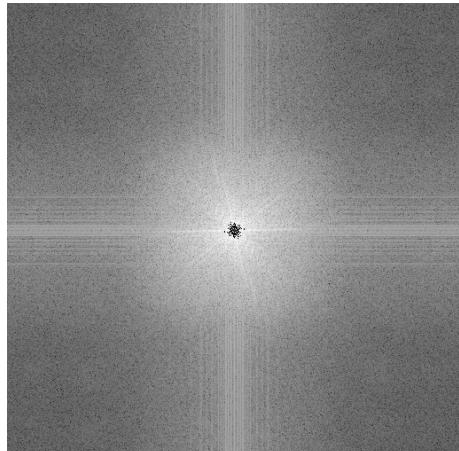


FIGURE 29 – Spectre des fréquences de l'image avant filtrage

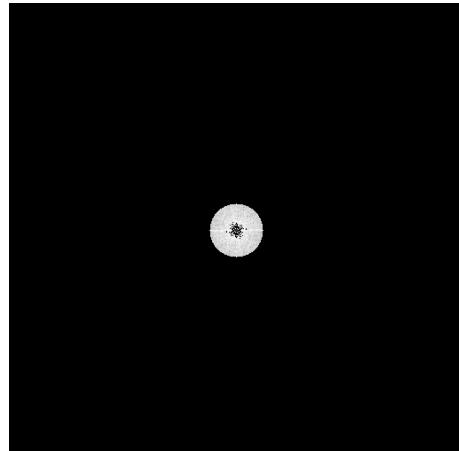


FIGURE 30 – Spectre des fréquences de l'image après filtrage

Nous avons par la suite pensé à une optimisation au niveau du rayon du filtre. L'idée consiste à tester plusieurs rayons de filtrage et à garder l'image filtrée qui a le plus haut PSNR. Mais pour cela, il nous fallait une vérité de terrain car sinon le rayon choisi serait simplement le plus grand. Donc nous avons décidé de rajouter du bruit à l'image d'entrée.



FIGURE 31 – Vérité de terrain



FIGURE 32 – Image bruitée en entrée



FIGURE 33 – Image floutée avec le plus haut PSNR

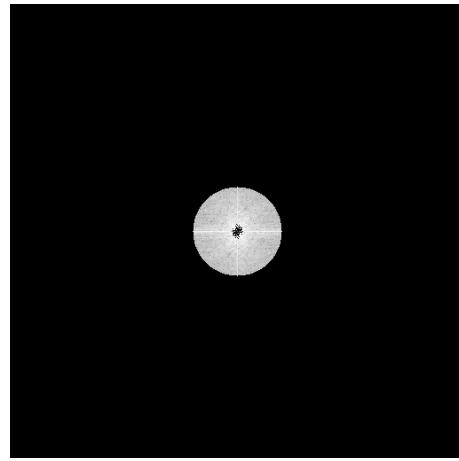


FIGURE 34 – Spectre des fréquences de l'image après filtrage

Nous avons obtenu les résultats suivants : **PSNR** = 32.51 dB pour un rayon de 50 pixels.

2.2 Égalisation d'histogramme

Cette étape permet d'accentuer les contrastes dans l'image floutée.



FIGURE 35 – Image filtrée avec un filtre bilatéral 7x7



FIGURE 36 – Image obtenue après égalisation de son histogramme

On remarque que les dégradations sont mieux visible notamment en bas de l'image (Figure-36), cela permet de faciliter leur detection lors de l'inpainting.

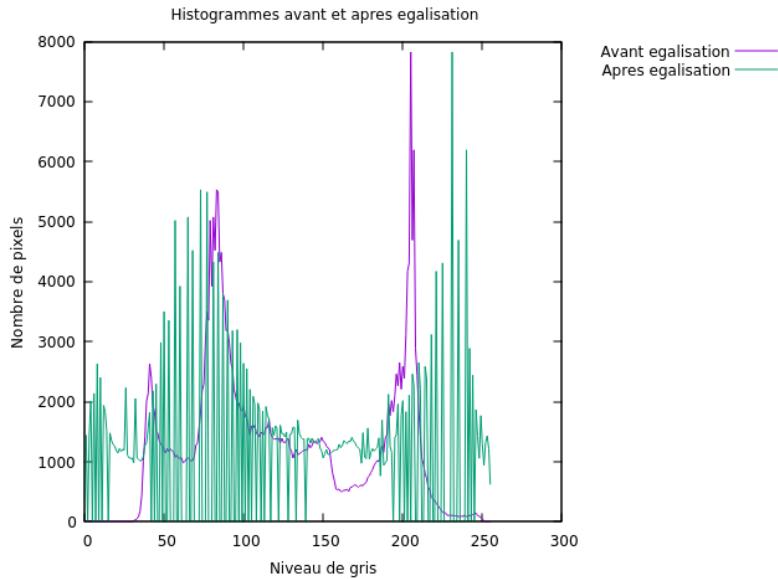


FIGURE 37 – Histogrammes avant et après égalisation

2.3 Déconvolution

Cette étape permet de déflouter l'image, elle est appliquée dans le post-traitement. Pour cela, nous l'avons testé sur une image floutée et après égalisation de son histogramme.



FIGURE 38 – Image floutée après égalisation de son histogramme



FIGURE 39 – Image obtenue après déconvolution

Le défloutage est assez léger mais c'est une étape qui arrive en fin de traitement afin d'ajout de la netteté dans l'image finale.

2.4 Processus de restauration

Nous avons terminé l'étape d'inpainting. Pour la création du masque binaire, que le processus d'inpainting utilise pour reconstituer les zones dégradées, nous avons développé deux méthodes permettant de le générer automatiquement (bien qu'elles ne soient pas très efficaces) et une méthode pour peindre le masque manuellement.

2.4.1 Méthodes de création du masque binaire

La 1ère méthode de création du masque binaire est une méthode qui provient du document : **Multidirectional Scratch Detection and Restoration in Digitized Old Images** [1]. Le processus de détection des dégradations consiste à :

1. Convertir l'image en niveaux de gris

2. Appliquer un filtre "Sharpen" de netteté pour éliminer les variations de luminosité de fond et pour mettre en évidence les détails.
3. Appliquer plusieurs filtres passe-bande dans différentes directions (12 directions au total avec un intervalle de 15°) afin de mettre en valeur les lignes de l'images.
4. Application d'un seuillage basé sur la moyenne et l'écart-type de l'intensité de l'image.
5. Application d'une transformée de Hough pour identifier les lignes droites (représentant les rayures), puis filtrage des lignes candidates afin d'éviter que la détection des rayures ne se superpose.
6. Détection du cœur de la rayure et croissance de la région

Voici des exemples sur des images non prétraitées (pas de débruitage).



FIGURE 40 – Image originale

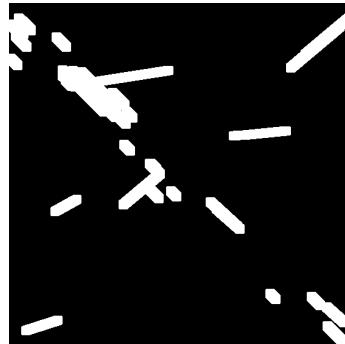


FIGURE 41 – Masque obtenu



FIGURE 42 – Image finale

PSNR = 28.92 dB

SSIM = 0.98



FIGURE 43 – Image originale

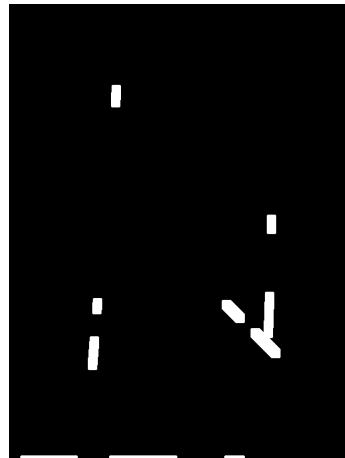


FIGURE 44 – Masque obtenu



FIGURE 45 – Image finale

PSNR = 32.08 dB

SSIM = 0.99



FIGURE 46 – Image originale

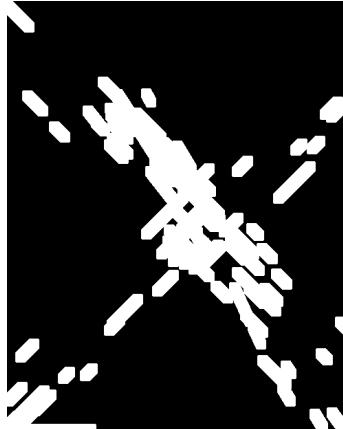


FIGURE 47 – Masque obtenu



FIGURE 48 – Image finale

PSNR = 25.06 dB

SSIM = 0.92

Il apparaît que cette méthode n'est pas du tout efficace dans la détection de dégradations d'une image ancienne ce qui entraîne une mauvaise reconstruction de l'image finale. Cette méthode est surtout efficace pour détecter des dégradations qui sont droites, voir très peu courbées. On remarque également que les valeurs de SSIM sont très proches de 1 ce qui indique qu'il n'y a pas eu de réels changements au niveau de l'image (ce que l'on observe visuellement sur les résultats ci-dessus).

L'autre méthode, pour générer automatiquement le masque binaire, est de suivre ce processus :

1. Application d'un flou gaussien sur l'image suivi du calcul de la magnitude du gradient (avec Sobel)
2. Application d'un seuil pour détecter les zones à fortes variations
3. On ferme les petits trous dans les rayures
4. On utilise un classifieur de Haar (pré entrainé)

Voici des exemples sur des images non prétraitées (pas de débruitage).



FIGURE 49 – Image originale

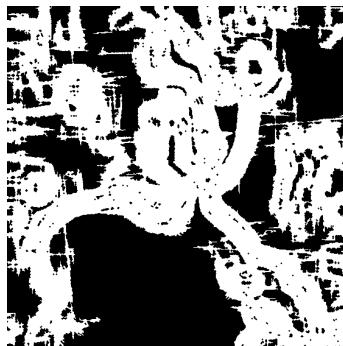


FIGURE 50 – Masque obtenu

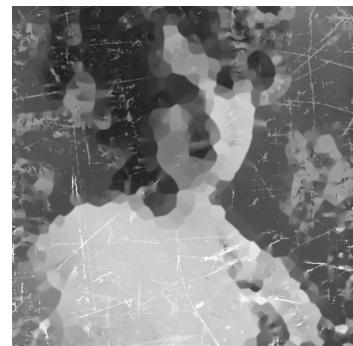


FIGURE 51 – Image finale

PSNR = 22.16 dB

SSIM = 0.94

En plus d'avoir un PSNR bas et un SSIM élevé, l'image présente tellement de défauts que le visage n'est pas reconnu.



FIGURE 52 – Image originale

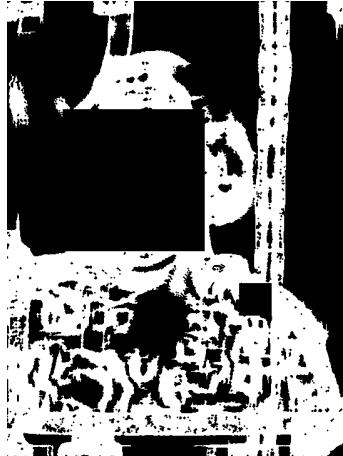


FIGURE 53 – Masque obtenu



FIGURE 54 – Image finale

PSNR = 19.68 dB

SSIM = 0.90



FIGURE 55 – Image originale

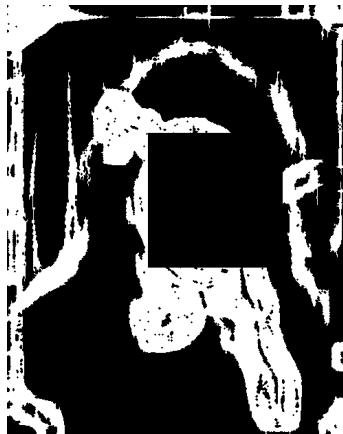


FIGURE 56 – Masque obtenu



FIGURE 57 – Image finale

PSNR = 24.16 dB

SSIM = 0.91

Cette méthode est une méthode qui utilise un classifieur Haar pour détecter les visages sans l'image en traçant un rectangle noir sur le masque binaire. On retrouve des PSNR très bas et des SSIM très élevés ce qui correspond bien avec la visualisation des images obtenues après l'application de l'inpainting. Les dégradations sont un peu mieux comparé à la méthode précédente.

La dernière méthode pour obtenir le masque binaire des dégradations est la méthode qui permet de peindre manuellement le masque binaire et c'est de loin la meilleure. Certes il y a l'action humaine qui entraîne des erreurs dans le dessin mais c'est cette méthode qui permet d'obtenir le masque le plus précis possible sans que des contours inutiles ne soient pris en compte dans le masque.

Voici des exemples sur des images non prétraitées (pas de débruitage).



FIGURE 58 – Image originale

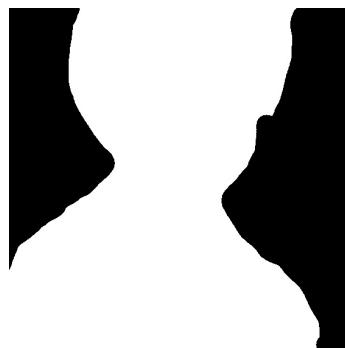


FIGURE 59 – Masque obtenu



FIGURE 60 – Image finale

PSNR = 19.18 dB

SSIM = 0.88



FIGURE 61 – Image originale

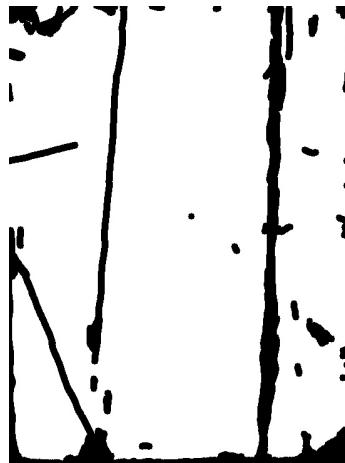


FIGURE 62 – Masque obtenu



FIGURE 63 – Image finale

PSNR = 18.3 dB

SSIM = 0.86



FIGURE 64 – Image originale



FIGURE 65 – Masque obtenu



FIGURE 66 – Image finale

PSNR = 23.19 dB

SSIM = 0.88

On peut constater que les masques et les résultats obtenus sont bien meilleurs avec la méthode manuelle qu'avec les méthodes automatiques. Certes, le processus n'est plus automatique mais les résultats obtenus sont meilleurs.

2.5 Tout le processus de restauration

Pour obtenir un résultat réaliste, bien que cela dépende beaucoup de l'image d'entrée, il est nécessaire de prétraiter l'image en suivant les étapes citées ci-dessus. Pour ce faire, il convient de simuler le processus comme si une méthode basée sur un réseau de neurones convolutifs (CNN) était utilisée, en ajoutant notamment du bruit à l'image. Pour les courbes ci-dessous, une étape dans le processus correspond à un traitement.



FIGURE 67 – Image originale

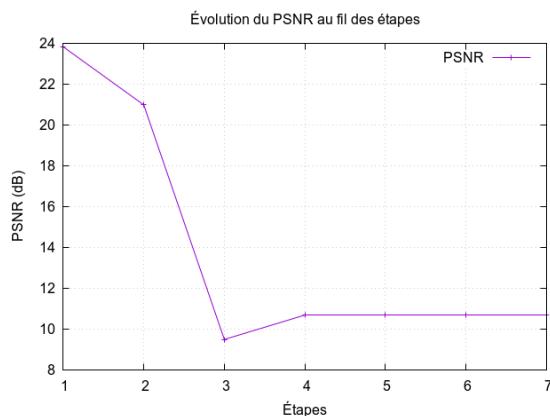


FIGURE 68 – Evolution du PSNR au fil du processus

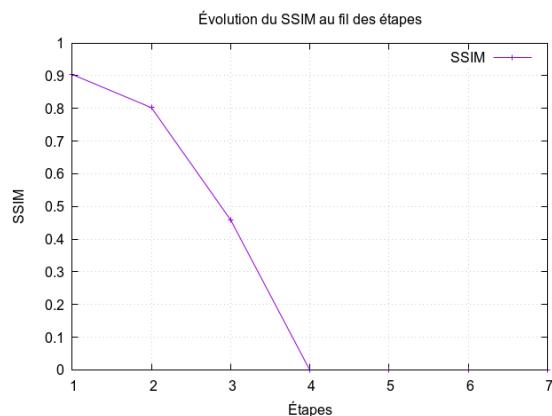


FIGURE 69 – Evolution du SSIM au fil du processus



FIGURE 70 – Image originale

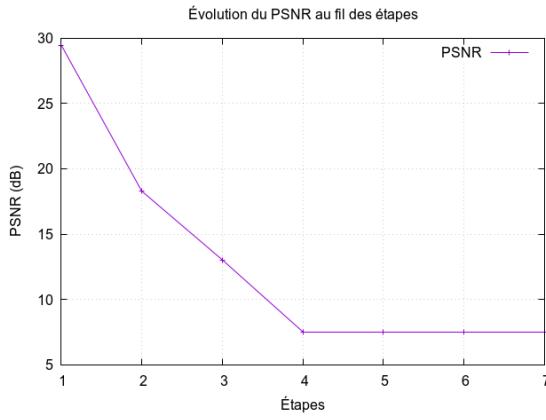


FIGURE 71 – Evolution du PSNR au fil du processus

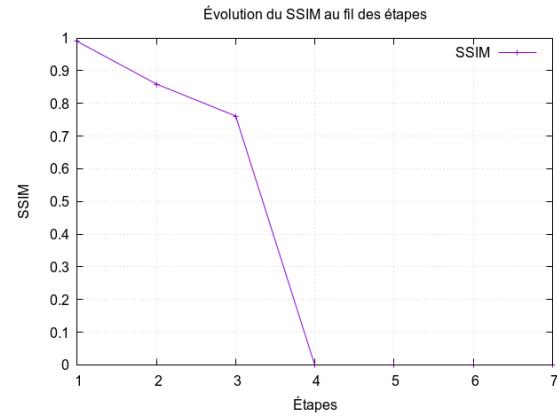


FIGURE 72 – Evolution du SSIM au fil du processus

On remarque que le flou, peut importe son type fait perdre beaucoup d'informations à l'image.

3 Choix de la méthode (avec CNN)

Après de multiples réflexions, nous avons choisis de nous orienter vers l'utilisation de VAEs (auto-encodeur variationnel) ou plus précisément de CVAEs (auto-encodeur variationnel convolutif).

La structure d'un VAE est constitué comme ceci :

1. Encodeur : Compresse les données d'entrée dans un espace latent (un espace de dimension réduite)
2. Décodeur : Prend des échantillons de l'espace latent et reconstruit les données d'origine

Un VAE utilise une approche probabiliste. L'encodeur génère une distribution des variables latentes plutôt qu'une valeur déterministe unique puis il génère ensuite les données à partir de cette distribution. Un VAE utilise également une fonction de coût qui combine la **reconstruction loss** qui mesure la différence entre l'entrée et la sortie reconstruite ainsi que de la divergence de Kullback-Leibler qui mesure la différence entre la distribution des variables latentes et une distribution prior (généralement une distribution normale)

3.1 Choix du dataset

Nous allons nous orienter sur le dataset **Pascal VOC 2012 Dataset**. Voici quelques images du dataset.



FIGURE 73 – Image du dataset



FIGURE 74 – Image du dataset



FIGURE 75 – Image du dataset



FIGURE 76 – Image du dataset



FIGURE 77 – Image du dataset



FIGURE 78 – Image du dataset



FIGURE 79 – Image du dataset



FIGURE 80 – Image du dataset



FIGURE 81 – Image du dataset



FIGURE 82 – Image du dataset

4 Conclusion

Nous avons déjà le processus sans CNN en place. Ce processus est très modulable car suivant la photo d'entrée, il faut oui ou non appliquer du débruitage, appliquer une égalisation d'histogramme, etc... ou bien simplement faire le processus d'inpainting (masque réalisé manuellement).

5 Planning prévisionnel

Notre planning prévisionnel, sur les semaines à venir, est le suivant :

- **CR5** (semaine 6) : Début de création de l'interface et implémentation du processus de restauration avec VAE
- **CR6** (semaine 7) : Amélioration de l'interface et poursuite de l'implémentation du processus avec VAE

Références

- [1] H. Dindo E. ARDIZZONE et G. MAZZOLA. “Multidirectional Scratch Detection and Restoration in Digitized Old Images”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Oct. 2010. URL : <https://jivp-eurasipjournals.springeropen.com/counter/pdf/10.1155/2010/680429.pdf>.