

# HAI92TI - Projet image : Débruitage

MAURIN Christina - COQUERON Solal

# Etat de l'art : méthodes traditionnelles

Plusieurs type de bruit : Poivre et sel, Gaussien, Chromatique...

Plusieurs type de filtres :

1/16 x

1	2	1
2	4	2
1	2	1

Gaussien

a	b	c
d	e	f
g	h	i

e remplacé par médiane de [a,i]

Médian

1/16 x

1	2	1
2	4	2
1	2	1

Moyenneur

$$W(i, j) = e^{-\frac{(i-x)^2 + (j-y)^2}{2s^2}} \cdot e^{-\frac{(I(i, j) - I(x, y))^2}{2r^2}}$$

Bilatéral

0	1	0
1	-4	1
0	1	0

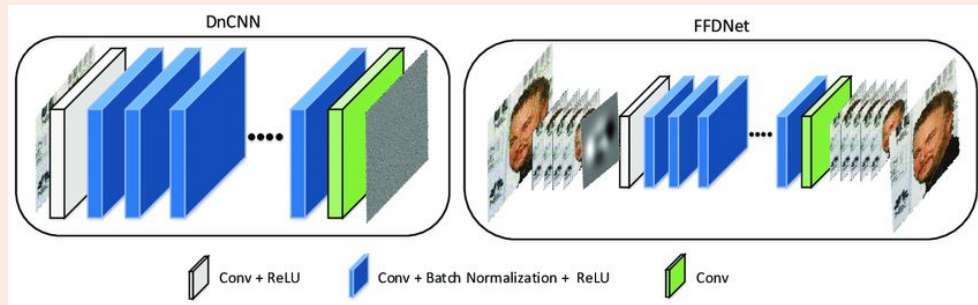
Laplacien

Papier : *Efficient poisson denoising for photography*

- Le bruit des capteurs d'images est dominé par le bruit de poisson
- La plupart des techniques partent du principe d'un bruit Gaussien
- L'idée est de transformer ce bruit de poisson pour le rapprocher d'un bruit Gaussien
- Cela va permettre d'améliorer l'efficacité des filtres classiques



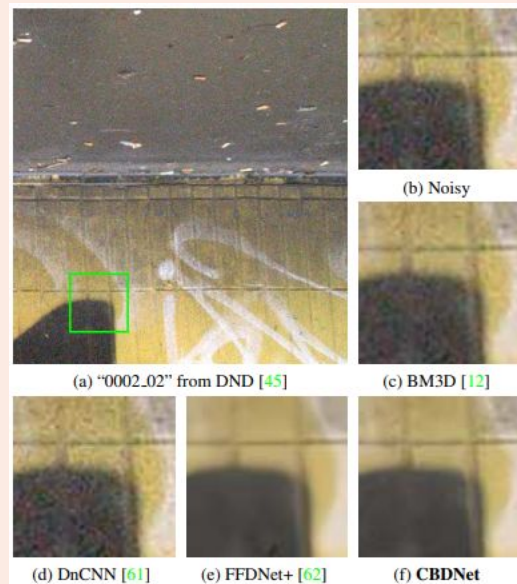
# Etat de l'art : réseau de neurones



- DnCNN et FFDNet ont des bons résultats pour des images avec du bruit simple
- Bruits plus complexes en photographie

Papier : *Toward Convolutional Blind Denoising of Real Photographs*

- CNN moins efficace sur de vraies photos
- Améliorer la généralisation grâce à la prise en compte des bruits liés au signal de l'image et au traitement que l'appareil photo effectue.
- Entraînement sur des paires de photos bruitées et non bruitées



# Résultats obtenus



Image originale



Ajout bruit poivre et sel



Ajout bruit gaussien



Ajout bruit chromatique



Filtrage médian

26,2 dB



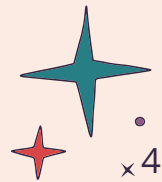
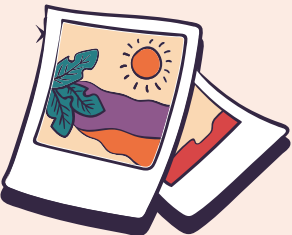
Filtrage gaussien

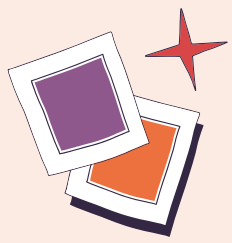
25,9 dB



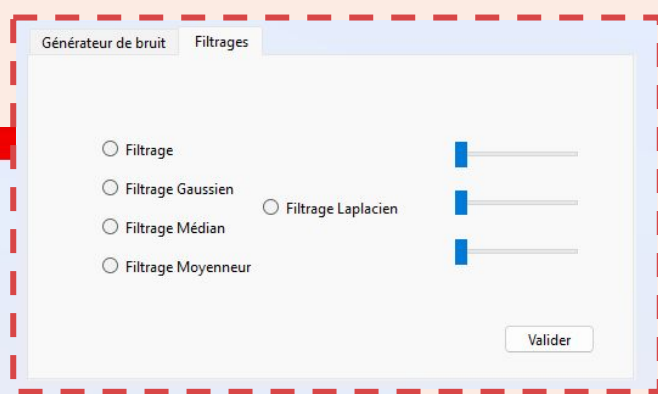
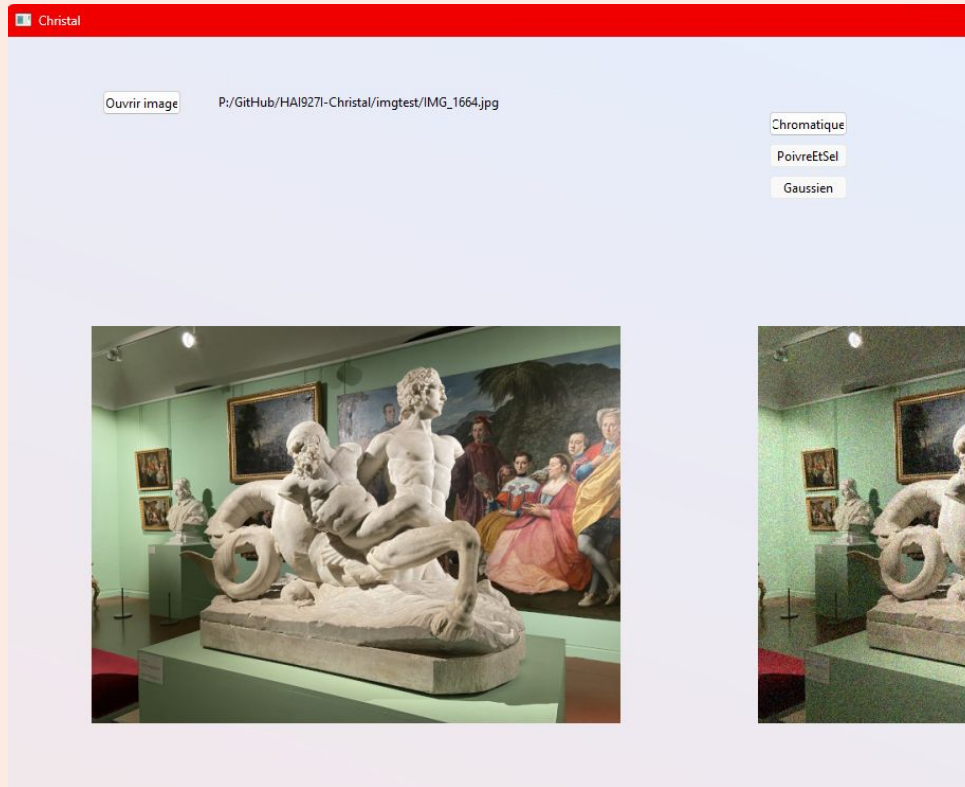
Filtrage bilatéral

27,8 dB





# Résultats obtenus



Création base de code pour l'application développée sur QtCreator Python



Merci pour votre attention !

