Denoising Images Project

ELABORAZIONE DI IMMAGINI

DARIO CIVALE - 0622701620 - D.CIVALE1@STUDENTI.UNISA.IT

PAOLO MANSI - 0622701542 - P.MANSI5@STUDENTI.UNISA.IT

VINCENZO SALVATI - 0622701550 - V.SALVATI10@STUDENTI.UNISA.IT

Guided Filter

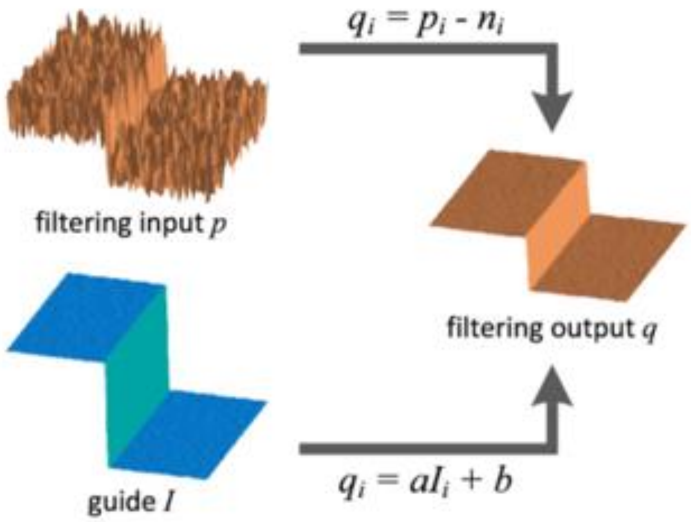
Filter which use a **guided image** to filter the input.

Main use:
- texture transfer;
- **denoising**;
- etc…

Local Linear Model

$$q_i = a_k * I_i + b_k, \forall i \in w_k$$

<u>Edge preserved</u>: $\nabla q = \nabla I$



q = output
p = input
a, b = unknown parameters
I = guided image

Parameters

Cost function to be minimized:

$$E(a_k, b_k) = \sum_{i \in w_k} \left((a_k * I_k + b_k - p_i)^2 + \varepsilon * a_k^2 \right)$$, where ε is a *regularization parameter*

Deriving the function, we get the parameters :

$$a_k = \frac{\frac{1}{|w_k|} * \sum_{i \in w_k} I_i * p_i - \mu_k * \bar{p}_k}{\sigma_k^2 + \varepsilon} \qquad b_k = \bar{p}_k - a_k * \mu_k$$, where:

μ_k = *mean* of I in w_k
σ_k^2 = *variance* of I in w_k
$\bar{p}$ = *mean* of p in w_k
$|w_k|$ = pixels in the window

Because of **overlapping windows for each pixel**, we consider the **mean of a and b**:

$$\bar{a}_i = \frac{1}{|w_k|} \sum_{k \in w_i} a_k \qquad \bar{b}_i = \frac{1}{|w_k|} \sum_{k \in w_i} b_k \quad \Rightarrow \quad q_i = \bar{a}_i * I_i + \bar{b}_i$$

Denoising

Denoising predicts that $\mathbf{I} = \mathbf{p}$:

$$a_k = \frac{\sigma_k^2}{\sigma_k^2 + \varepsilon} \qquad b_k = (1 - a_k) * \mu_k$$

$$\boldsymbol{\varepsilon} = \mathbf{0} \;\rightarrow\; a_k = 1, b_k = 0$$

$$\boldsymbol{\varepsilon} > \mathbf{0} \; \begin{cases} \sigma_k^2 \gg \varepsilon \;\rightarrow\; a_k \approx 1, b_k \approx 0 & \textbf{High variance}\text{: preserving edge} \\ \sigma_k^2 \ll \varepsilon \;\rightarrow\; a_k \approx 0, b_k \approx \mu_k & \textbf{Flat Patch}\text{: smoothing} \end{cases}$$

Performance indexes b&w

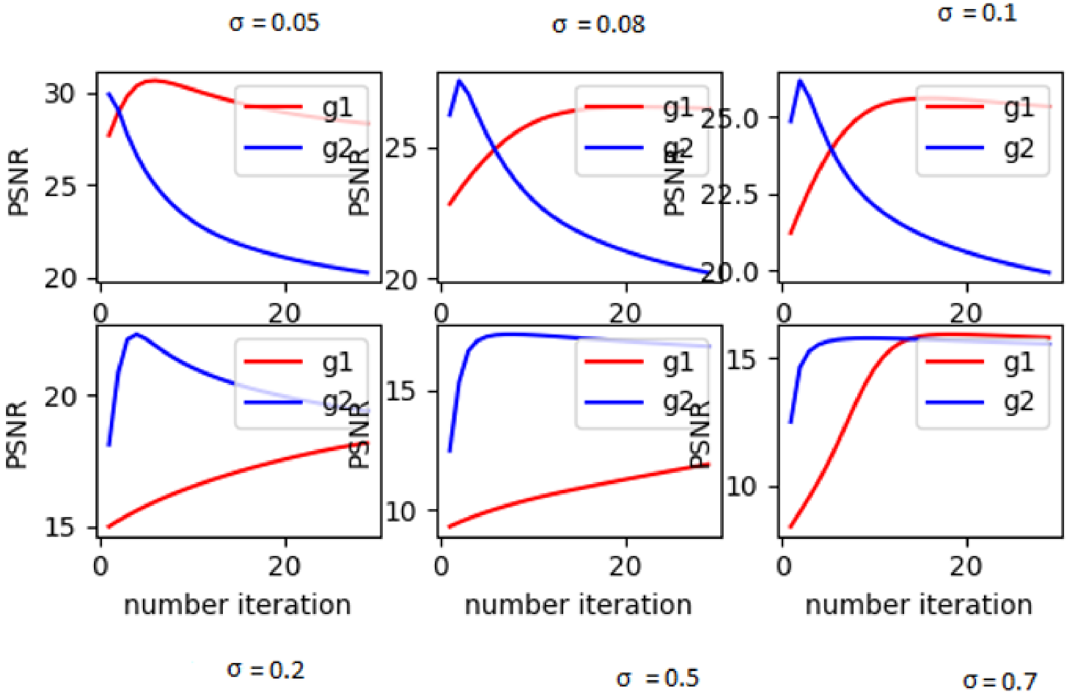
Performance indexes RGB

Anisotropic filter

It is a filtering technique that aims to reduce noise in images without removing significant parts of the image content, typically edges, lines, or other details that are important for image interpretation.

$$Img^{n} = Img^{n-1} + \frac{1}{\eta}\sum_{0}^{c} g(\delta_i(Img^{n-1}), K) * \delta_i(Img^{n-1})$$

Evaluation of conductivity functions



- $g_1 = e^{-\left(\frac{|\delta_i|}{K}\right)^2}$
- $g_2 = 1/(1 + \frac{|\delta_i|^2}{K^2}))$

Automatic calculation of the gradient threshold

- Canny noise estimator

- MAD estimator

$$K = 1.4826 * MAD(\nabla \mathrm{Img} - \mathrm{median}(\nabla \mathrm{Img}))$$

- Morphological estimator

$$K = \sum_{i,j \in Img} Img \circ \frac{\mathrm{st}}{\mathrm{row}} * \mathrm{col} - \sum_{i,j \in Img} Img \bullet \frac{st}{row} * col$$

Automatic calculation of the number of iterations

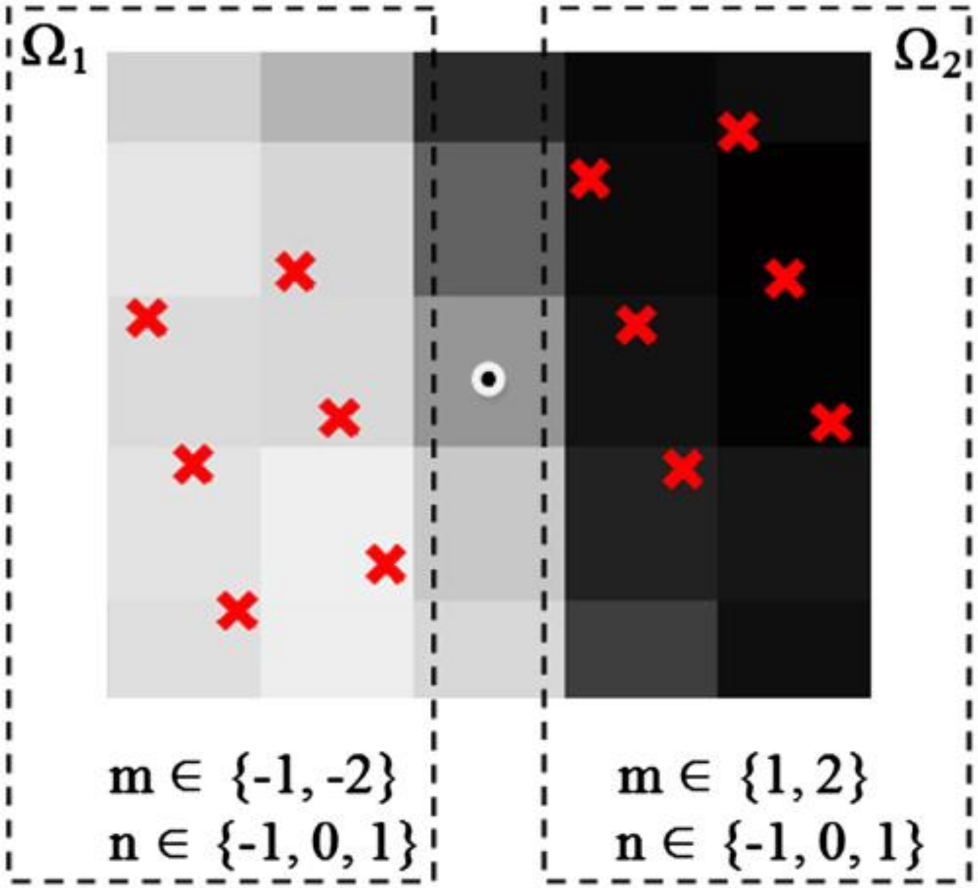
The algorithm consists of several steps:

1. Identification of the N edges

2. For each edge, a local area characterizing the edge is defined, in which the interpixel points are identified

$$x_{m,n} = x_k + m * cos(\theta) - n * sin(\theta) \qquad per\ m = \{-1, -1\ , 1\ , 2\}, n = \{-1,0,1\}$$
$$y_{m.n} = y_k - m * sin(\theta) - n * cos(\theta) \qquad per\ m = \{-2, -1,1,2\}, n = \{-1,0,1\}$$

3. Identification of interpixel regions



4. Calculation of the edge quality function Q for each edge identified in step 1

$$Q = |\, \mu_1 - \mu_2| - \alpha * |\sigma_1 + \sigma_2|$$

$$\alpha = 10 * \sigma/\mu 0, \mu 0 = (\frac{1}{\mathrm{N}}) \sum_1^N |\, \mu_1(\mathrm{t}) - \mu_2(t)|$$

5. Calculating the image quality function Qm at each iteration

$$Qm(t) = (1/N) \sum_{i=1}^{N} Q_i(t)$$

6. Estimation of the iteration instant

$$T = arg \max_t \quad Qm(t)$$

Filter behaviour b&w

Original

Gaussian noise

Morpho

- MSE: 0.01
- PSNR: 68.65
- SSIM: 0.69

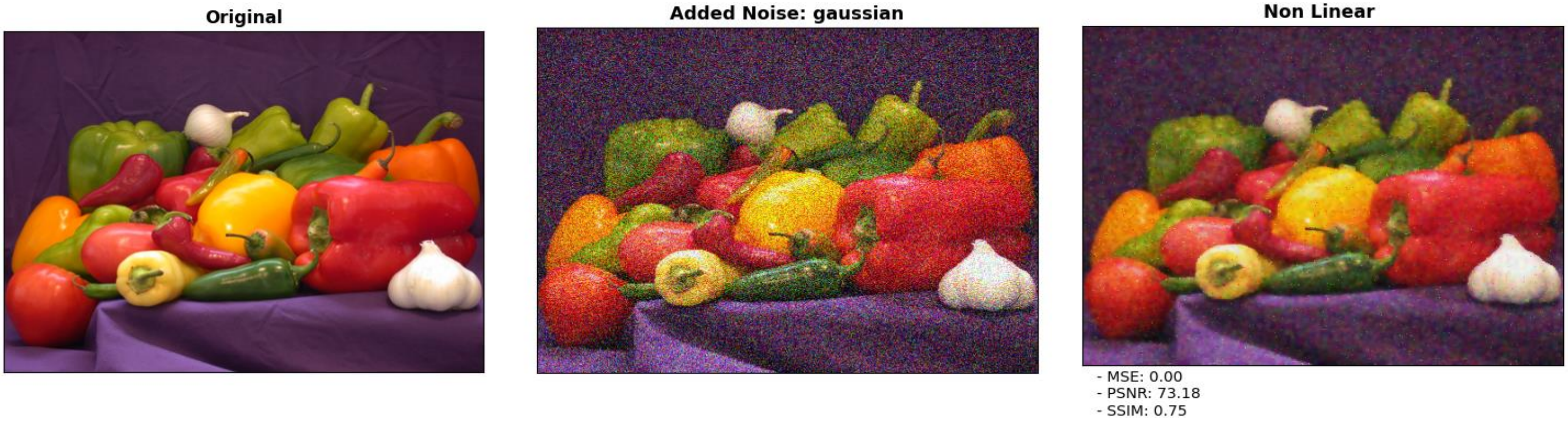
MAD

- MSE: 0.02
- PSNR: 66.33
- SSIM: 0.65

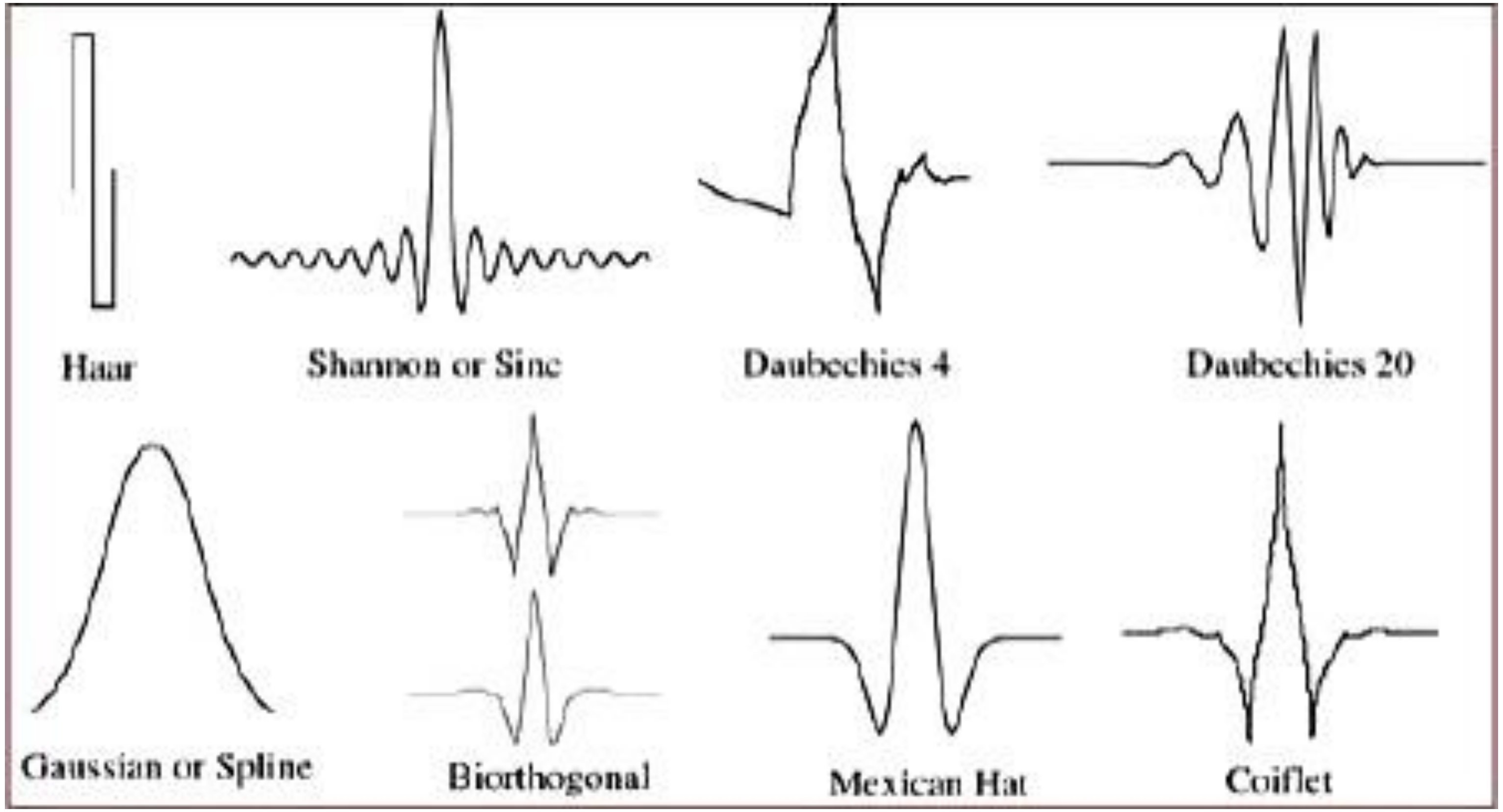
Canny noise estimator

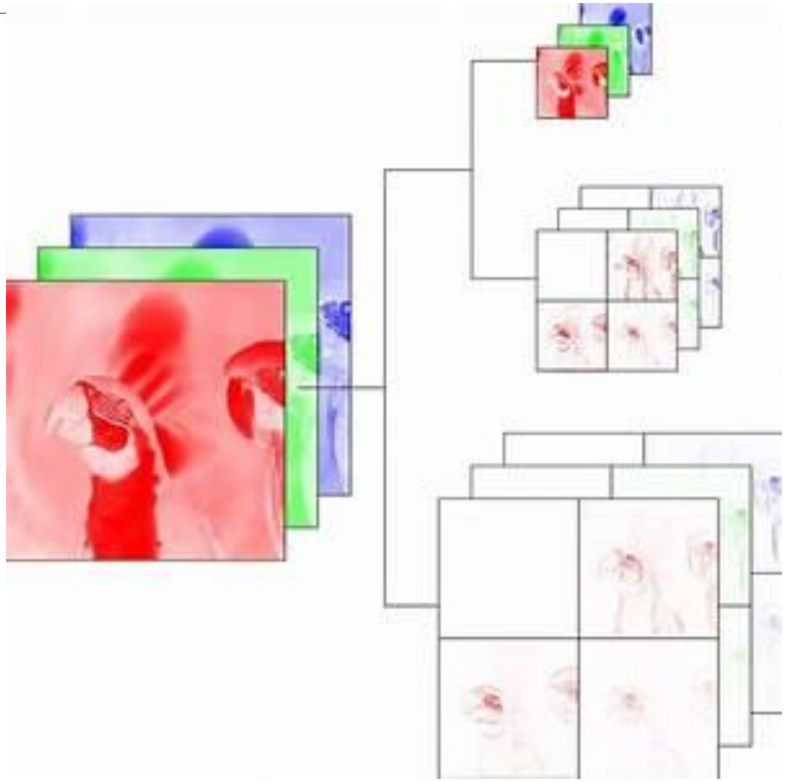
- MSE: 0.01
- PSNR: 70.53
- SSIM: 0.66

Filter behaviour RGB



Wavelet Denoising





Wavelet Denoising

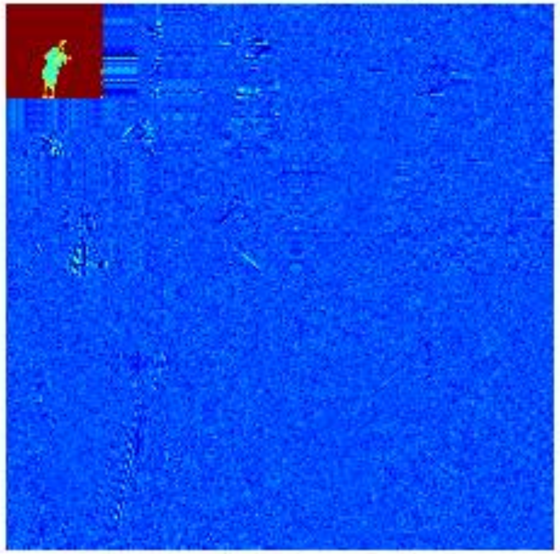
- Universal Threshold

$$\lambda = \sigma * \sqrt{2 * \log n^2}$$

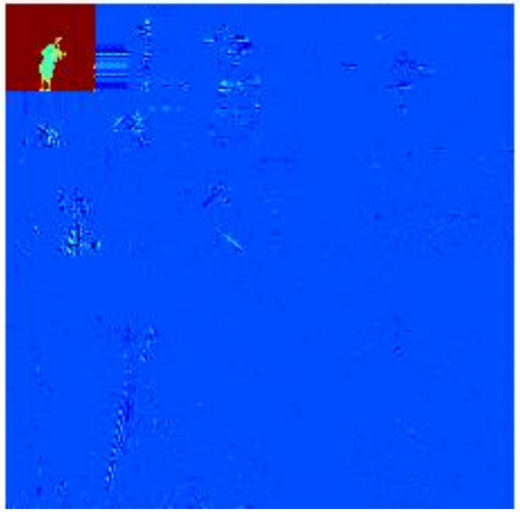
- NeighShrink

$$d_{j,k} = d_{j,k} * \left(1 - \frac{\lambda^2}{\sum_{(i,j)\in B_{j,k}} d_{i,l}^2}\right)$$

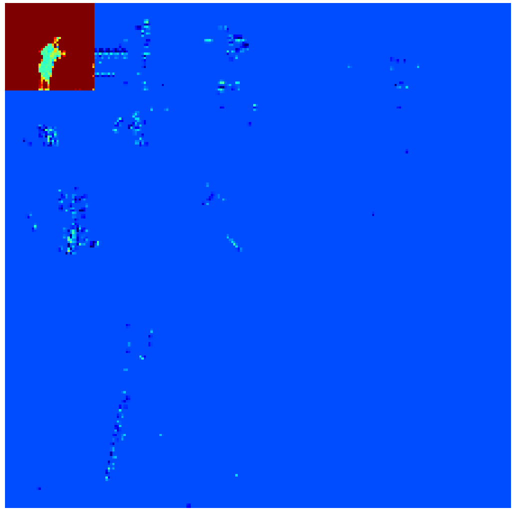
Pre Denoising

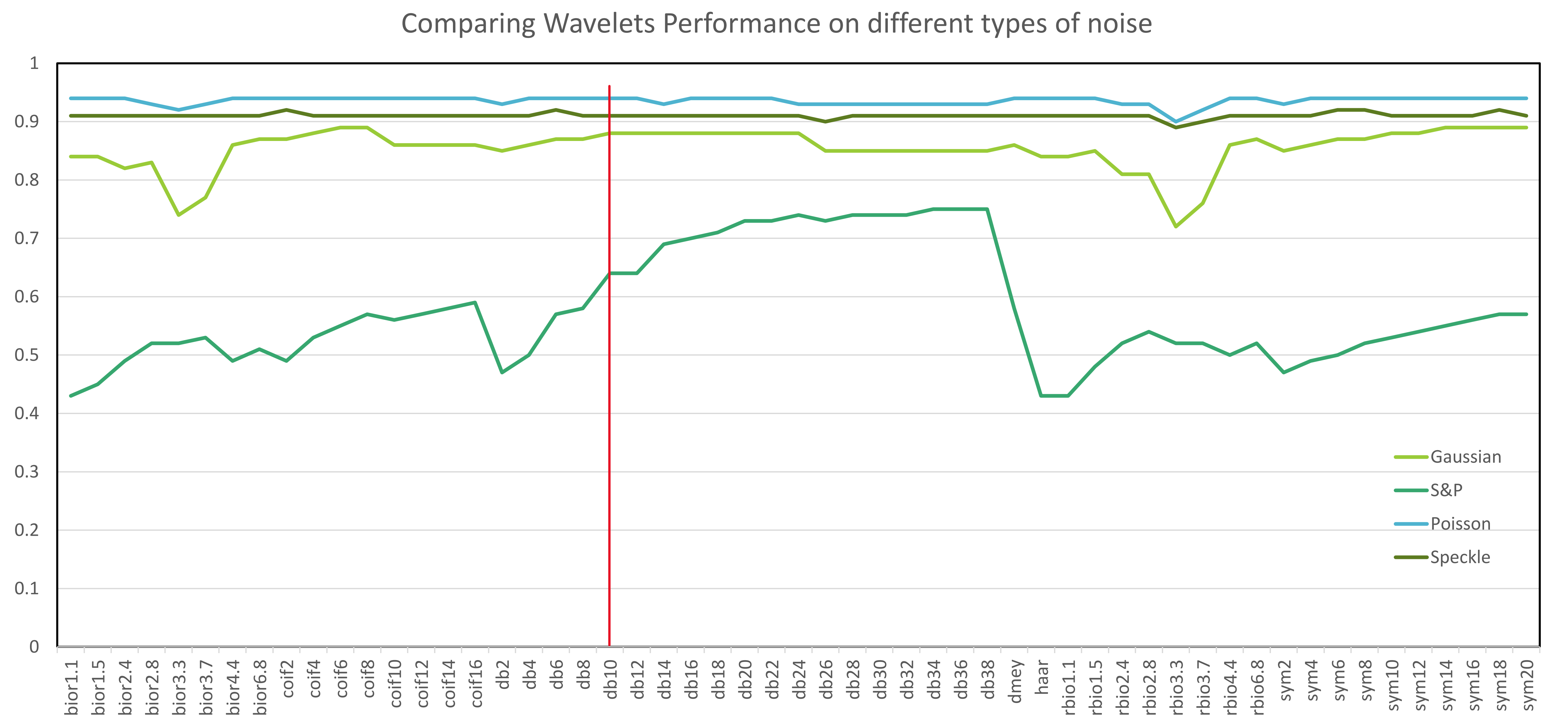


Universal Threshold



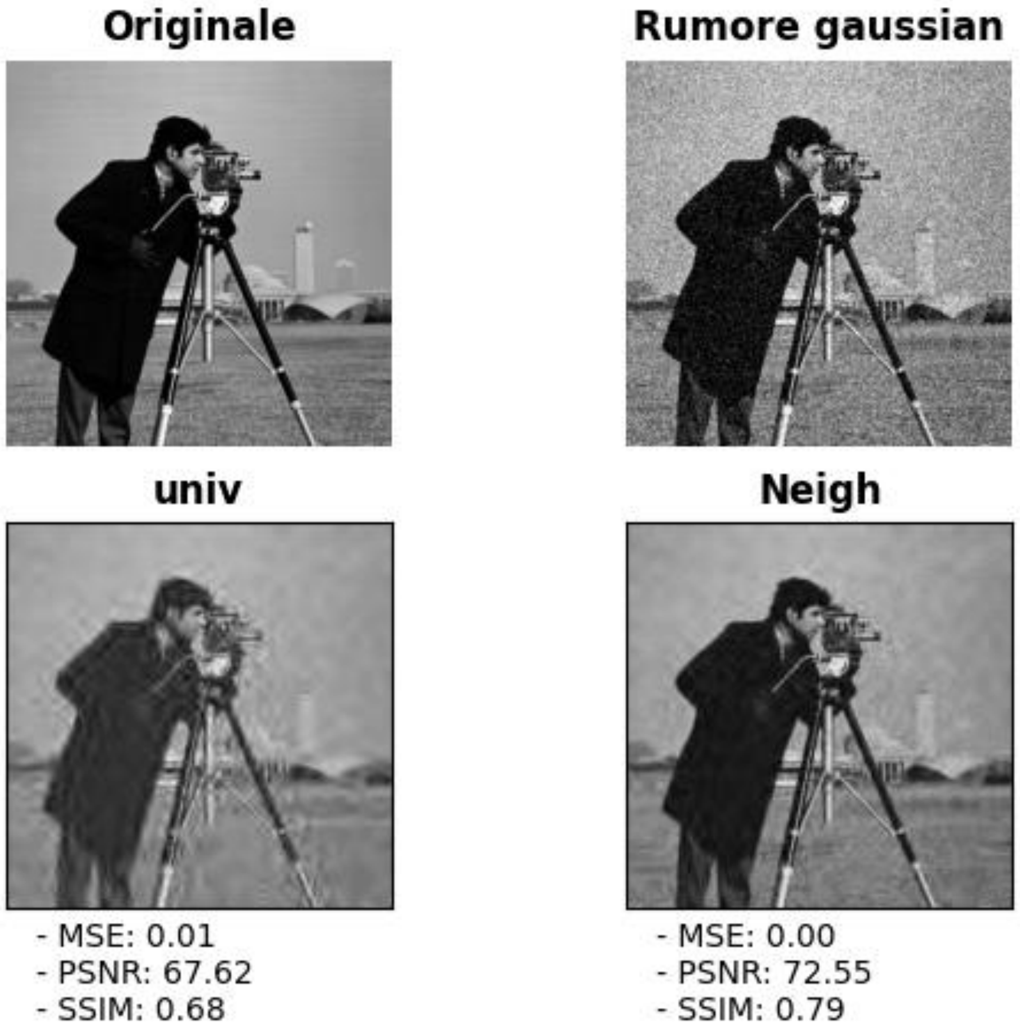
NeighShrink Threshold

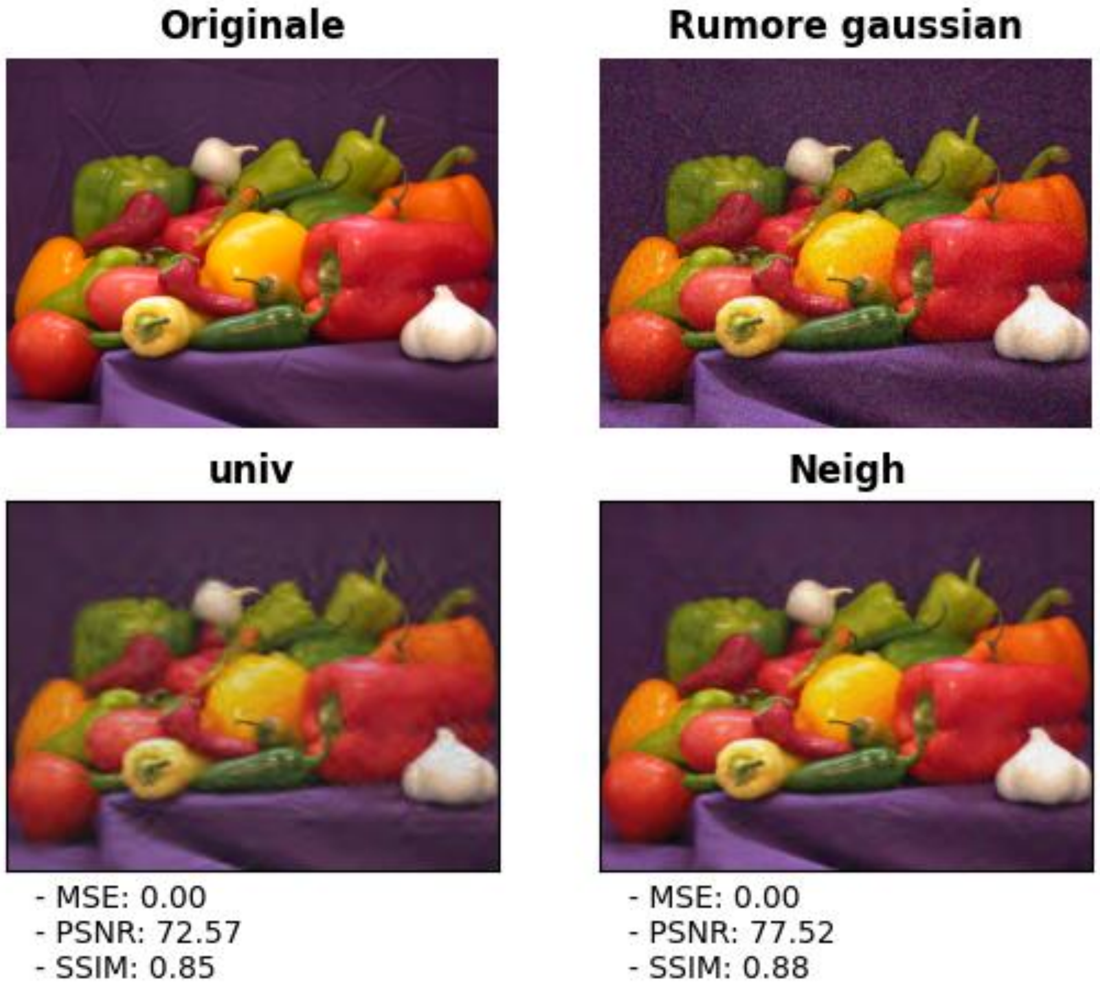


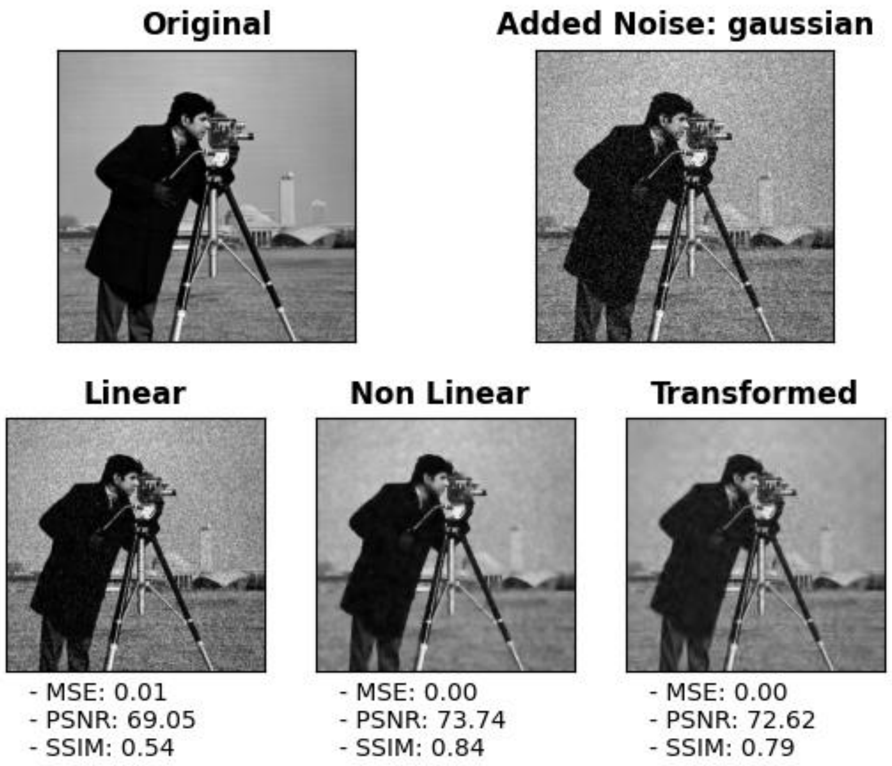


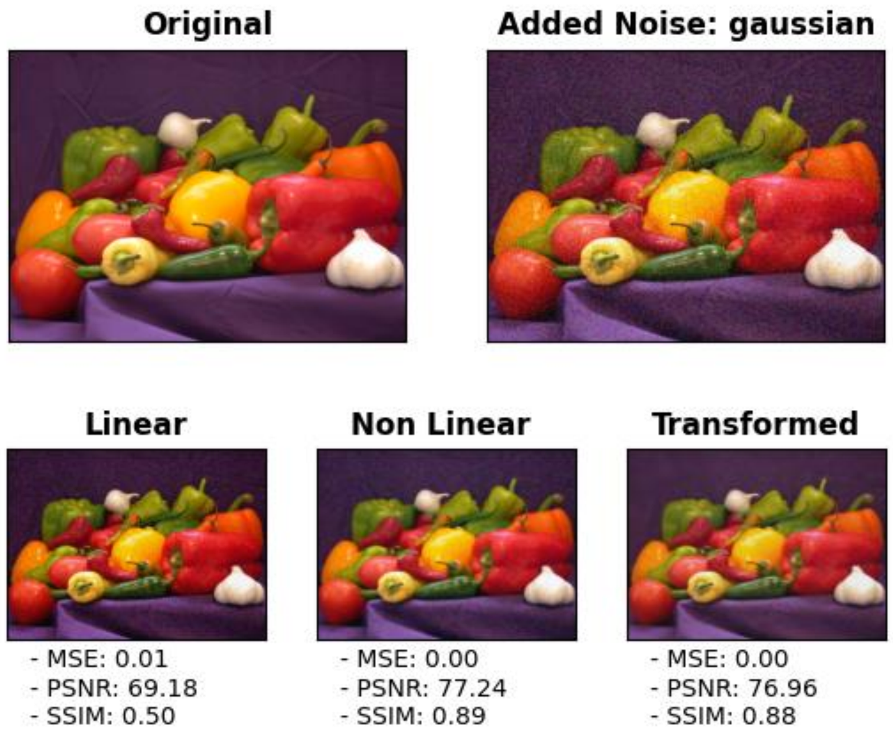
Comparing Wavelets Performance on different types of noise
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Gaussian
S&P
Poisson
Speckle
bior1.1
bior1.5
bior2.4
bior2.8
bior3.3
bior3.7
bior4.4
bior6.8
coif2
coif4
coif6
coif8
coif10
coif12
coif14
coif16
db2
db4
db6
db8
db10
db12
db14
db16
db18
db20
db22
db24
db26
db28
db30
db32
db34
db36
db38
dmey
haar
rbio1.1
rbio1.5
rbio2.4
rbio2.8
rbio3.3
rbio3.7
rbio4.4
rbio6.8
sym2
sym4
sym6
sym8
sym10
sym12
sym14
sym16
sym18
sym20

Wavelet Denoising

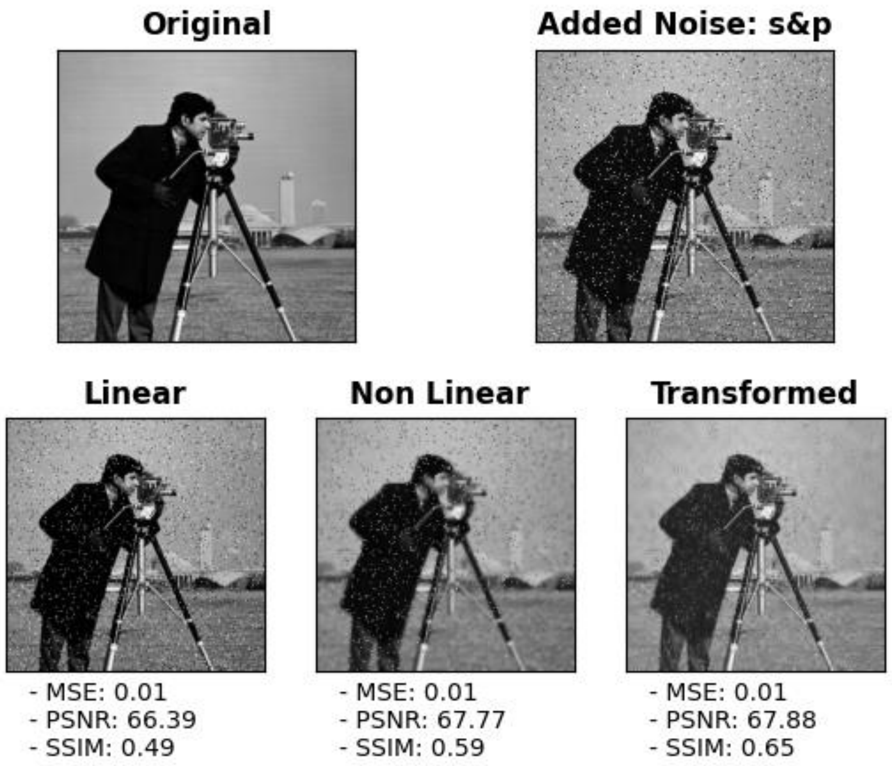


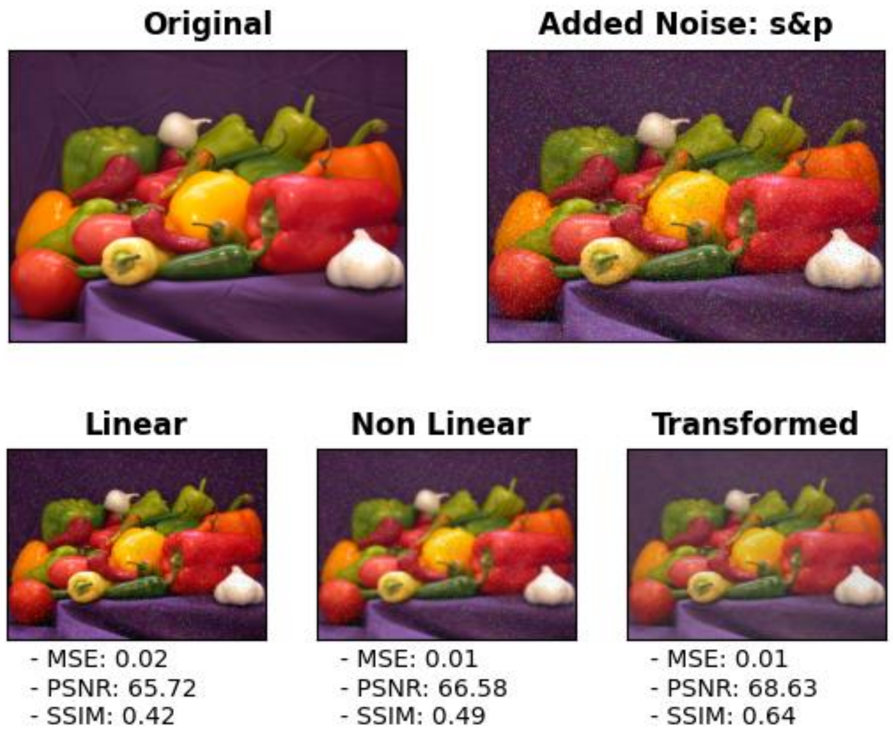






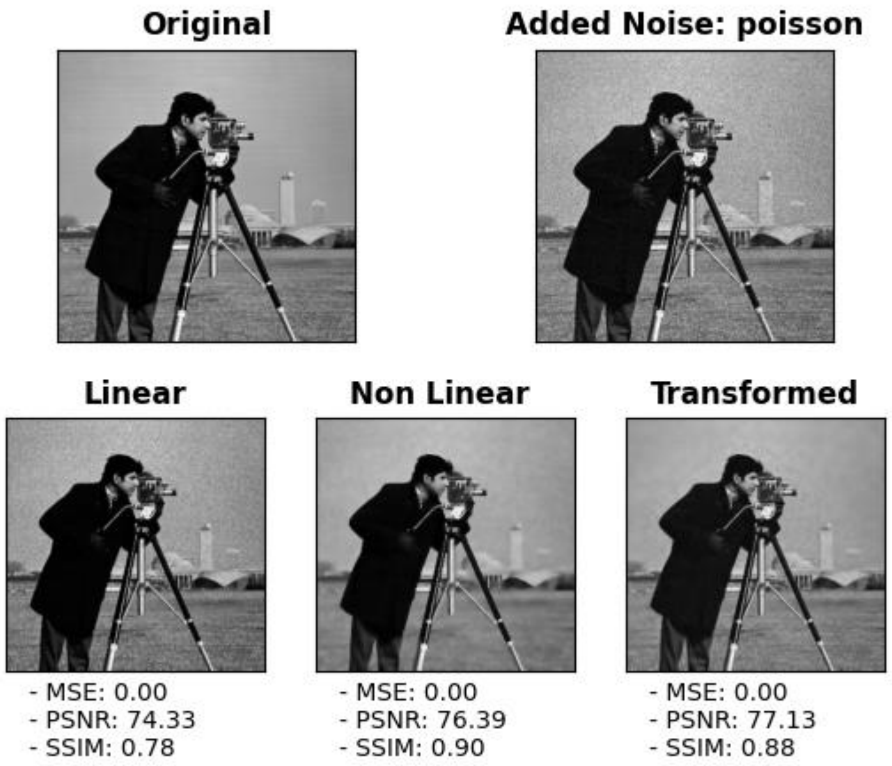
Filters comparation – Gaussian

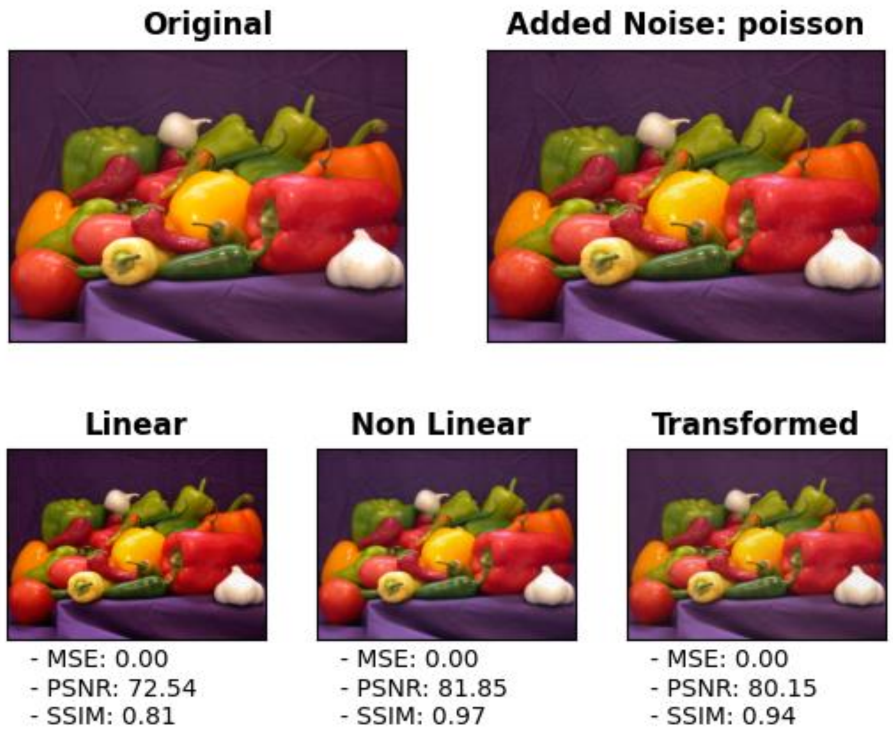


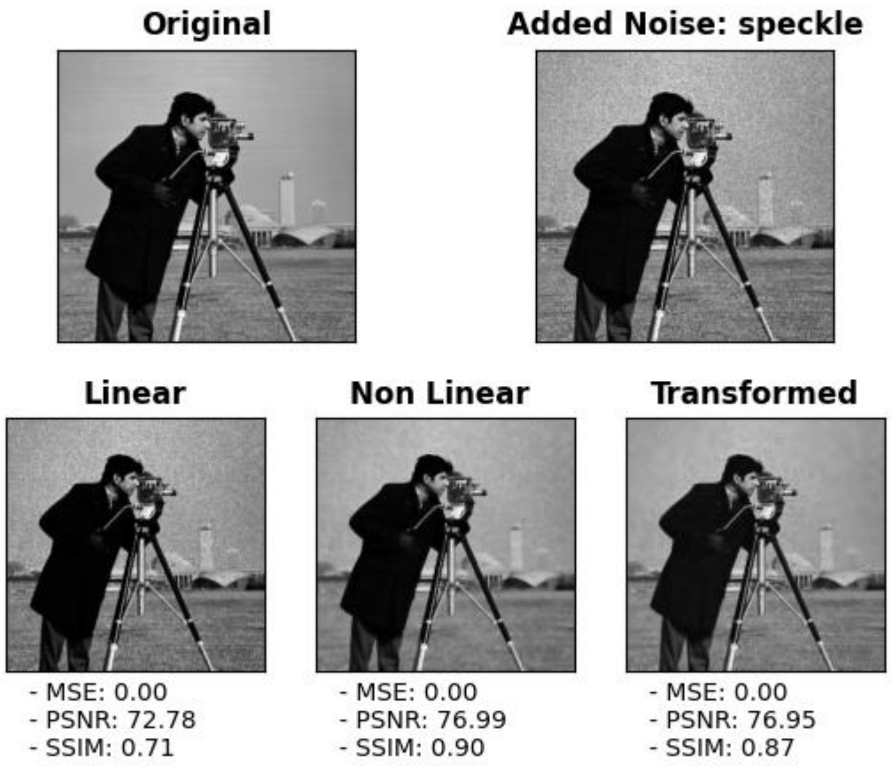


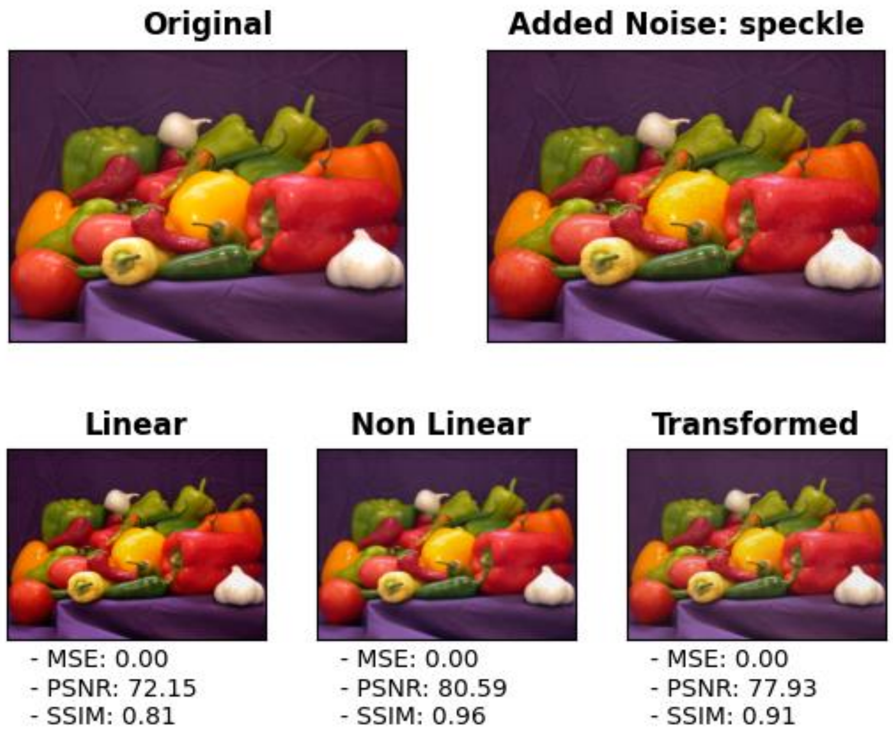
Filters comparation – Salt & Pepper

Filters comparation – Poisson









Filters comparation – Speckle