Supplementary material for :

Automatic Tuning of Denoising Algorithms Parameters without Ground Truth

Arthur Floquet, Sayantan Dutta,
Emmanuel Soubies, Duong-Hung Pham, Denis Kouame

This is a supplementary material file for the article: Automatic Tuning of Denoising Algorithms Parameters without Ground Truth.

GitLab repository of this article can be found here.

1 Detailled algorithms

In the article, we presented four methods to tune the algorithm parameters $\boldsymbol{\theta}$. Here, we present a decision tree (Fig. 4) allowing one to choose the most suitable method according to the available data, as well as the considered denoising algorithm.
Note that Fig. 4 includes all methods presented in the article. Empirically, we found $\hat{\mathbf{x}}^{\text{Nr2N}}$ to perform poorly, and $\hat{\mathbf{x}}^{\text{NaC}}$ to be acceptable only for small noise levels. We believe $\hat{\mathbf{x}}^{\text{NaC}}$ might be a suitable candidate for denoisers that somehow don't have parameters related to noise level.

2 Hyperparameters

The proposed method relies on some hyperparameters that include the learning rate, the number of iterations, the initial point $\boldsymbol{\theta}_0$. For the results reported in Fig. 2 of the main article, we fixed the number of iterations to 100 and considered Adam optimizer with default parameters and a learning rate of 1. Moreover, for $\hat{\mathbf{x}}^{\text{Nr2N}}$, we fixed $\alpha = 1$, and for $\hat{\mathbf{x}}^{\text{R2R}}$ we used $\mathbf{D} = \frac{1}{2}\mathbf{I}$, and $M = 50$. Regarding the initial point $\boldsymbol{\theta}_0$ we fixed it through coarse manual tuning on a *single image*. Yet, we were able to obtain high-quality denoised images which show the robustness of the approach to this initialization. Finally, for $\mathbf{D}$ and M in R2R, we followed the recommendation from [1]. For Nr2N, the hyperparameter α was chosen equal to 1 for simplicity.

The take away message here is that (at least with the tested DeQuip denoising method) results were not sensitive to the choice of these parameters, which makes them easy to set.

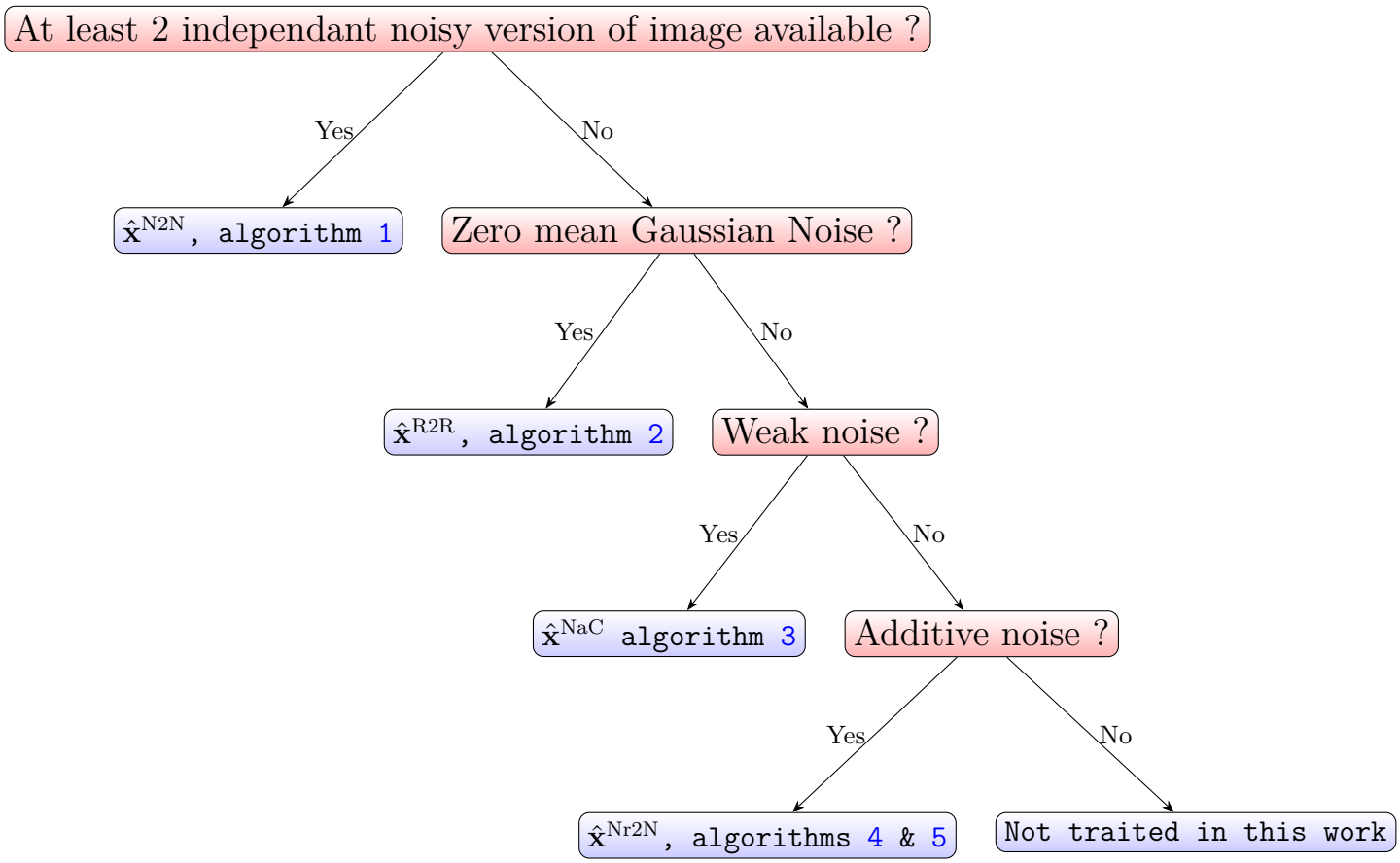

Figure 4: Help in choosing appropriate method

Algorithm 1 Computing $\hat{\mathbf{x}}^{\mathrm{N2N}}$

Require:
Denoising algorithm $A_{\boldsymbol{\theta}}$
$\mathbf{y}_1, \mathbf{y}_2$ two noisy version of $\mathbf{x}$, with independant noises.
Initial parameters $\boldsymbol{\theta}_0$
Learning rate l_r
Discrepancy measure $\mathcal{L}$ ▷ Adapted to noise type [2]
Optimizer $\mathcal{O}$
Number of iterations K
for $k \leftarrow 1$ to K **do**
  $\mathbf{u}_k \leftarrow A_{\boldsymbol{\theta}_k}(\mathbf{y}_1)$
  Compute $\mathcal{L}(\mathbf{u}_k, \mathbf{y}_2)$
  Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}$
  $\boldsymbol{\theta}_{k+1} \leftarrow \mathcal{O}\left(\boldsymbol{\theta}_k, l_r, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}\right)$ ▷ Optimizer step
end for
$\hat{\mathbf{x}}^{\mathrm{N2N}} \leftarrow A_{\boldsymbol{\theta}_K}(\mathbf{y}_1)$

Algorithm 2 Computing $\hat{\mathbf{x}}^{\mathrm{R2R}}$

Require:
Denoising algorithm $A_{\boldsymbol{\theta}}$
$\mathbf{y} = \mathbf{x} + \mathbf{n}$, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$
Invertible matrix $\mathbf{D}$
Initial parameters $\boldsymbol{\theta}_0$
Learning rate l_r
Optimizer $\mathcal{O}$
Number of iterations K
Inference number M
for $k \leftarrow 1$ to K **do**
  $\mathbf{n}_s^k \leftarrow \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$
  $\mathbf{z}_1^k = \mathbf{y} + \mathbf{D}^T\mathbf{n}_s^k$, $\mathbf{z}_2^k = \mathbf{y} - \mathbf{D}^{-1}\mathbf{n}_s^k$
  $\mathbf{u}_k \leftarrow A_{\boldsymbol{\theta}_k}(\mathbf{z}_1^k)$
  $\mathcal{L} \leftarrow ||\mathbf{u}_k - \mathbf{z}_2^k||_2^2$
  Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}$
  $\boldsymbol{\theta}_{k+1} \leftarrow \mathcal{O}\left(\boldsymbol{\theta}_k, l_r, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}\right)$ ▷ Optimizer step
end for
$\hat{\mathbf{x}}^{\mathrm{R2R}} \leftarrow \mathbf{0}$
for $m \leftarrow 1$ to M **do** ▷ Monte-Carlo inference
  $\mathbf{n}_s^m \leftarrow \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$
  $\mathbf{z}_1^m = \mathbf{y} + \mathbf{D}^T\mathbf{n}_s^m$
  $\hat{\mathbf{x}}^{\mathrm{R2R}} \leftarrow \hat{\mathbf{x}}^{\mathrm{R2R}} + A_{\boldsymbol{\theta}_K}(\mathbf{z}_1^m)$
end for
$\hat{\mathbf{x}}^{\mathrm{R2R}} \leftarrow \frac{1}{M}\hat{\mathbf{x}}^{\mathrm{R2R}}$

Algorithm 3 Computing $\hat{\mathbf{x}}^{\mathrm{NaC}}$

Require:
Denoising algorithm $A_{\boldsymbol{\theta}}$
$\mathbf{y} = \mathbf{x} + \mathbf{n}$, $\mathbf{n}$ weak noise from distribution $\mathcal{D}$ ▷ $\mathbf{n}$ *does not need to be additive*
Initial parameters $\boldsymbol{\theta}_0$
Learning rate l_r
Optimizer $\mathcal{O}$
Number of iterations K
for $k \leftarrow 1$ to K **do**
  $\mathbf{n}_s^k \sim \mathcal{D}$
  $\mathbf{z}^k = \mathbf{y} + \mathbf{n}_s^k$ ▷ can be done with other noise types
  $\mathbf{u}_k \leftarrow A_{\boldsymbol{\theta}_k}(\mathbf{z}^k)$
  $\mathcal{L} \leftarrow ||\mathbf{u}_k - \mathbf{y}||_2^2$
  Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}$
  $\boldsymbol{\theta}_{k+1} \leftarrow \mathcal{O}\left(\boldsymbol{\theta}_k, l_r, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}\right)$ ▷ Optimizer step
end for
$\hat{\mathbf{x}}^{\mathrm{NaC}} \leftarrow A_{\boldsymbol{\theta}_K}(\mathbf{y}_1)$

Algorithm 4 Computing $\hat{\mathbf{x}}^{\text{Nr2N}}$

Require:
Denoising algorithm $A_{\boldsymbol{\theta}}$
$\mathbf{y} = \mathbf{x} + \mathbf{n}$, $\mathbf{n} \sim \mathcal{D}$ ▷ For $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$, see Algorithm 5
Initial parameters $\boldsymbol{\theta}_0$
Learning rate l_r
Optimizer $\mathcal{O}$
Number of iterations K
for $k \leftarrow 1$ to K **do**
  $\mathbf{n}_s^k \sim \mathcal{D}$
  $\mathbf{z}^k = \mathbf{y} + \mathbf{n}_s^k$
  $\mathbf{u}_k \leftarrow A_{\boldsymbol{\theta}_k}(\mathbf{z}^k)$
  $\mathcal{L} \leftarrow ||\mathbf{u}_k - \mathbf{y}||_2^2$
  Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}$
  $\boldsymbol{\theta}_{k+1} \leftarrow \mathcal{O}\left(\boldsymbol{\theta}_k, l_r, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}\right)$ ▷ Optimizer step
end for
$\mathbf{n}_s \sim \mathcal{D}$
$\mathbf{z}^k = \mathbf{y} + \mathbf{n}_s$
$\hat{\mathbf{x}}^{\text{NaC}} \leftarrow 2A_{\boldsymbol{\theta}_K}(\mathbf{z}) - \mathbf{z}$

Algorithm 5 Computing $\hat{\mathbf{x}}^{\text{Nr2N}}$ (Gaussian noise case)

Require:
Denoising algorithm $A_{\boldsymbol{\theta}}$
$\mathbf{y} = \mathbf{x} + \mathbf{n}$, $\mathbf{n} \sim \mathcal{N}(0, \sigma)$
Variance parameter $\alpha \in]0, 1]$
Initial parameters $\boldsymbol{\theta}_0$
Learning rate l_r
Optimizer $\mathcal{O}$
Number of iterations K
for $k \leftarrow 1$ to K **do**
  $\mathbf{n}_s^k \leftarrow \mathcal{N}(\mathbf{0}, \alpha\sigma\mathbf{I})$
  $\mathbf{z}^k = \mathbf{y} + \mathbf{n}_s^k$
  $\mathbf{u}_k \leftarrow A_{\boldsymbol{\theta}_k}(\mathbf{z}^k)$
  $\mathcal{L} \leftarrow ||\mathbf{u}_k - \mathbf{y}||_2^2$
  Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}$
  $\boldsymbol{\theta}_{k+1} \leftarrow \mathcal{O}\left(\boldsymbol{\theta}_k, l_r, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k}\right)$ ▷ Optimizer step
end for
$\mathbf{n}_s \sim \mathcal{D}$
$\mathbf{z}^k = \mathbf{y} + \mathbf{n}_s$
$\hat{\mathbf{x}}^{\text{NaC}} \leftarrow \frac{(1+\alpha^2)A_{\boldsymbol{\theta}_K}(\mathbf{z}) - \mathbf{z}}{\alpha^2}$

References

[1] T. Pang, H. Zheng, Y. Quan, and H. Ji, "Recorrupted-to-Recorrupted: Unsupervised deep learning for image denoising," in *Conference on Computer Vision and Pattern Recognition*, IEEE/CVF, 2021, pp. 2043–2052.

[2] J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," in *International Conference on Machine Learning*, PMLR, Mar. 2018.