Assignment 2 - Medical Image Analysis

2020/2021

Francesco Corti & Julie Mazzella

12002944 12040967

Graz, May 16, 2021

Contact: Francesco Corti & Julie Mazzella, corti@student.tugraz.at julie.mazzella@student.tugraz.at

1 Introduction

Image denoising is one of the main problems in medical image analysis, since it is considered an ill-posed problem.
Ill-posed problems have either no solution, infinite solutions, or solutions that change discontinuously depending on the initial situations. Therefore, to solve the problem, it is necessary to limit the space of possible solutions to only meaningful ones. This is done by applying regularization to the considered issue, and for this a priori knowledge is necessary.

A statistical interpretation of this problem is given by the Bayesian Inference: a random variable u taken from a probability distribution is considered and the objective is to find u', that is the maximally probably hypothesis u solving the inverse problem, given an image f. This solution is the maximum a posteriori estimation (MAP) method, written through the Bayes rule, that exploits the conditional probabilities:

$$u' = \max_u\{p(u|f)\} = \frac{p(f|u)p(u)}{p(f)} \tag{1}$$

In the equation $p(f|u)$ is the data model, which is the probability of an observation f given that we know the correct u, $p(u)$ is the prior model, which is what is known about u, and $p(f)$ is a normalization factor to ensure that the posterior probability $p(u|f)$ sums to one.
The Bayes rule describes how to update the probability of u, given new observations f.

1.1 Tikhonov-regularized denoising

This regularization problem can be solved through the Tikhonov method.
First, the probabilistic model is defined as:

$$max\left\{p(u|f) = \prod_{\Omega} p(u)p(f|u)\right\} \tag{2}$$

where Ω describes the whole image domain. For each pixel in Ω, independence of u is assumed. The probabilities $p(u)$ and $p(f|u)$ are modeled as normal distributions:

$$p(u) = exp(-\frac{\|\nabla u\|_2^2}{2}), \quad p(f|u) = exp(-\frac{\lambda}{2}(u-f)^2) \tag{3}$$

$\|\nabla u\|_2^2$ is used as an argument in the prior model: the gradient is squared, so large gradients (image edges in f) are penalized more. As a consequence, it is costly to create edges in the solution u due to this modeling.
$p(f|u)$, on the other hand, is modeled as a normal distribution with pixel-wise differences between u and f as an argument: this represents a penalization factor.
Since maximizing exponential functions is equal to minimizing their arguments, it is possible to obtain the following energy functional:

$$min\left\{E(u) = \frac{1}{2}\int_{\Omega}\|\nabla u\|_2^2 d\vec{x} + \frac{\lambda}{2}\int_{\Omega}(u-f)^2 d\vec{x}\right\} \tag{4}$$

This minimization problem can be solved by considering the associated Euler-Lagrange equation:

$$\Delta u + \lambda(u - f) = 0 \tag{5}$$

which is a partial differential equation (PDE). Using the Calculus of Variations, a theorem to describe the functional at a stationary point is obtained. Setting the functional derivative to zero leads to the condition of equation 5 that the functional has to satisfy.

To solve the Tikhonov-regularized model, discretization of the continuous variables is required. One way is to assume that the solution of 5 is a stationary point of an artificially introduced time-dependent diffusion equation, and solving it by using the Gradient Descent Optimization:

$$u^{t+1} = u^t - \tau(-\Delta u^t + \lambda(u^t - f)) \tag{6}$$

where τ is the time step. The result moves in the direction of the gradient after every iteration step.

1.2 Perona-Malik denoising

By solving the equation 5 numerically, a blurring filter related to Gauss convolution is implemented, smoothing out edges of the original image.
In the Perona-Malik approach, however, edges can be preserved by incorporating the gradient of f: this is done by introducing in the generalized quadratic regularization a diffusion tensor D that encodes axis-specific information:

$$\int_\Omega \|\nabla u\|_2^2 d\vec{x} = \int_\Omega \nabla u^T \nabla u d\vec{x} = \int_\Omega \nabla u^T D \nabla u d\vec{x} \tag{7}$$

In the Perona-Malik approach, D is identified as a function $g(\|\nabla f\|_2)$, determined by the image gradient ∇f in the main diagonal:

$$g(\|\nabla f\|_2 = \frac{1}{1 + \left(\frac{\|\nabla f\|_2}{K}\right)^2} \tag{8}$$

with K being a constant related to the noise level. In the Tikhonov method, on the other hand, D is identified as the identity matrix.

1.3 Total Variation denoising

An alternative option for preserving the edges is given in the Total Variation denoising method. Compared to Tikhonov, which uses a quadratic prior, TV uses a different and more robust norm for the prior. It encodes the variation of u in x and y direction squared:

$$\int_\Omega \|\nabla u\|_2 d\vec{x} = \int_\Omega \sqrt{(\frac{\partial u}{\partial x})^2 + \frac{\partial u}{\partial y})^2} d\vec{x} \tag{9}$$

However, TV is harder to minimize because the derivative is undefined at zero, and this is the case in homogeneous regions. The Euler-Lagrange equation for the Total Variation (or ROF model) is:

$$-\nabla \cdot \frac{\nabla u}{\||\nabla u\||_2} + \lambda(u - f) = 0 \tag{10}$$

The solution to this PDE is given by the explicit gradient descent optimization, which introduces a small term ϵ to avoid that the divisor is zero. However, the choice of ϵ is critical: a large value yields slow convergence and smoothes over the edges, while a small value leads to instability.
Total Variation is a method able to preserve edges in the image while denoising and it is based on the principle that, depending on a trade-off parameter, large gradients do not incur any penalty, while small gradients are removed.
The problem is that the TV norm is only defined for smooth functions of u, since the gradient is infinite for discontinuous functions. As a consequence, it is difficult to reconstruct images that contain such edges.

1.4 Primal-Dual denoising

Total Variation can also be expressed by introducing a dual variable $\vec{p}$:

$$TV(u) = \int_\Omega \|\nabla u\|_2 d\vec{x} = \max_{\|\vec{p}\|_2 \leq 1} \left\{ \int_\Omega \vec{p} \cdot \nabla u d\vec{x} \right\} \tag{11}$$

TV is related to the sum of all perimeters of the iso-intensity curves through the normal vector field $\vec{p}$, which is the 1-normalized image gradient ∇u:

$$\|\nabla u\|_2 = \max_{\|\vec{p}\|_2 \leq 1} \left\{ \vec{p} \cdot \nabla u \right\} \tag{12}$$

whose solution is:

$$\vec{p} = \begin{cases} \frac{\nabla u}{\|\nabla u\|_2} & \text{if } \nabla u \neq 0 \\ \text{arbitrary} & \text{if } \nabla u \neq 0 \end{cases} \tag{13}$$

The equivalence of:

$$-\int_\Omega u \nabla \cdot \vec{p} d\vec{x} = \int_\Omega \vec{p} \cdot \nabla u d\vec{x} \tag{14}$$

is obtained through the Divergence property. The dual relationship is then applied into the primal solution, resulting in a saddle point problem that can be solved by optimizing u and $\vec{p}$ alternatively, through gradient ascent and descent:

$$\min_u \max_{\|\vec{p}\|_2 \leq 1} \left\{ \int_\Omega \vec{p} \cdot \nabla u d\vec{x} + \frac{\lambda}{2} \int_\Omega (u - f)^2 d\vec{x} \right\} \tag{15}$$

2 Methods

The aim of the project is to implement algorithms for the four denoising methods: Tikhonov, Perona-Malik, Total Variation ROF and Primal-Dual ROF.
To do that a Python code was implemented, using Numpy [1] to handle matrices and vectors.
The noisy image considered was the following:

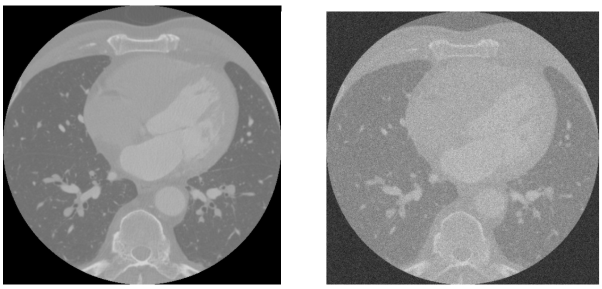

Figure 1: Heart axial slice: original image and noisy image

The algorithms were performed with different values of the regularization parameter λ, in order to evaluate how the denoising is acting on the considered image. The following values were considered:

- $\lambda \in \{5, 0.5, 0.05\}$.

2.1 Tikhonov-regularized denoising

As described in Sec. 1, in order to implement the Tikhonov denoising method, a random variable u is considered and the objective is to find the optimal u' that replicates the original image f. In order to do so, the gradient of the random variable u is computed in the x and y directions. Then, the Laplacian Δu is computed using the differential backward operation. Finally, the optimal u' is computed by applying iteratively the equation 6 of the gradient descent optimization.

2.2 Perona-Malik denoising

Since the Tikhonov algorithm smoothes the edges and blurs the image, the Perona-Malik method introduces a diffusion tensor that encodes gradient orientation.
To obtain the diffusion tensor, the gradients of f in both x and y directions are computed, and then the magnitude of the gradient is calculated. Afterward, the equation 8 is applied, knowing the parameter K. Finally, the Laplacian Δu is obtained by multiplying the diffusion tensor for the backward differentials of u in both directions and, as for the previous method, the optimal u' is obtained iteratively through the equation 6.

2.3 Total Variation denoising

The Total Variation algorithm is more efficient for edge preservation, since it uses a more robust norm for the prior. In fact, it does not use the quadratic norm anymore, and this allows to encode the variation of u in the x and y directions better. However, since the derivative is undefined in zero in homogeneous regions, the TV is harder to minimize. This leads to the introduction of the ϵ parameter.
To implement this method, first the gradients in the x and y directions of u are computed and then the magnitude is obtained. The following step is to calculate the normalized gradients: they are obtained dividing the previously computed gradients for the magnitude plus the ϵ parameter. This allows to avoid that the divisor is ever zero.
As for the previous methods, the Laplacian of u is computed and the optimal u' is obtained through equation 6.

2.4 Primal-Dual denoising

TV provides edge preserving denoising, but the TV norm is only defined for smooth functions of u, as the gradient is infinite for non-smooth functions. For this reason the dual variable $\vec{p}$ is introduced, which is the 1-normalized image gradient Δu. The dual relationship between $\vec{p}$ and u is exploited in an optimization problem that alternatively computes the optimized values of both variables.
To implement the algorithm [2], first the divergence of $\vec{p}$ is computed through the backward differentiation of $\vec{p}$ in both x and y directions. Then, the divergence of $\vec{p}$ is used to computed the update value of u according to the following equation:

$$u^{(n+1)} = u^n - \tau_P(-\nabla \cdot \vec{p} + \lambda(u^n - f)) \tag{16}$$

The gradients in x and y directions of this new u are calculated and used to compute the updated value of $\vec{p}$ according to the equation:

$$\tilde{p}^{(n+1)} = p^n + \tau_D(\nabla u) \tag{17}$$

Finally, the new value of $\vec{p}$ is obtained by dividing every element of p^{n+1} for the maximum element-wise value between the norm of $\vec{p}$ and one:

$$p^{(n+1)} = \frac{\tilde{p}^{(n+1)}}{max(1, \|\tilde{p}^{n+1}\|_2)} \tag{18}$$

This iterative method of alternating u and $\vec{p}$'s optimization is then repeated until convergence.

For each of these algorithms, the code has been runned until a stop criterion was met. The considered condition takes into account the normalized difference of the old u and the updated u'. If the value is lower than a fixed value, then the algorithm stops.

3 Results

For the four algorithms developed the same calculations have been conducted: first an analysis of the optimal regularization parameter λ over the same time step τ. Then, the best λ has been chosen and the denoising has been applied on the noisy image, to observe the resulting clean image. This process has been repeated for 4 different noisy images: one with Gaussian noise with variance equal to 10, one with variance equal to 20, one with variance equal to 40 and one with Salt and Pepper noise.
In the following subsections the corresponding results for the 4 methods are shown.

3.1 Tikhonov results

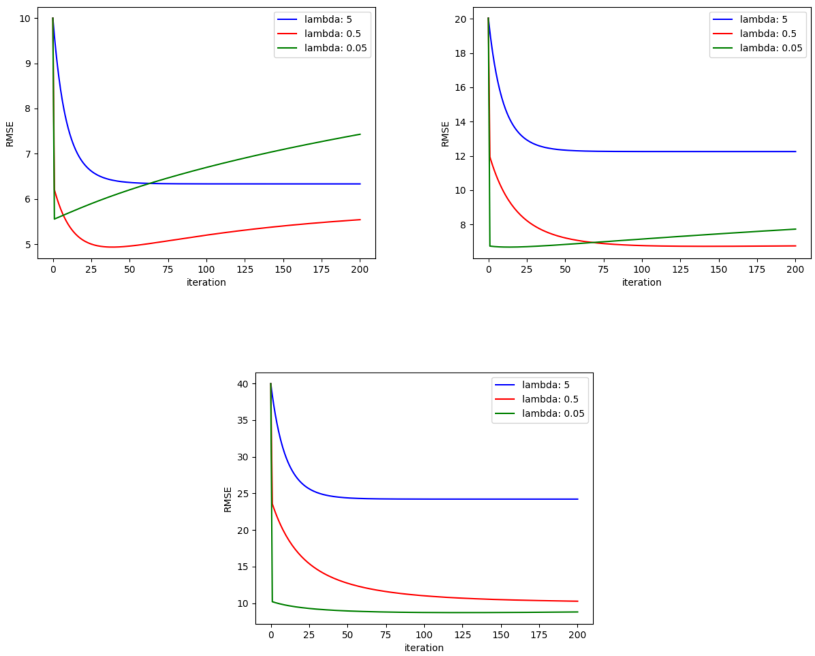

Figure 2: RMSE behavior with different λ values, for Tikhonov images with σ equal to 10, 20 and 40.

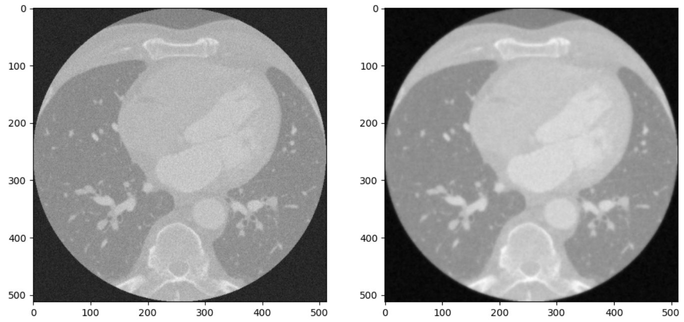

Figure 3: Tikhonov denoised image with variance equal to 10.

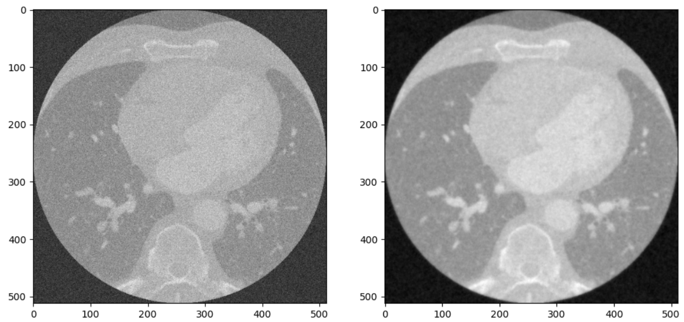

Figure 4: Tikhonov denoised image with variance equal to 20.

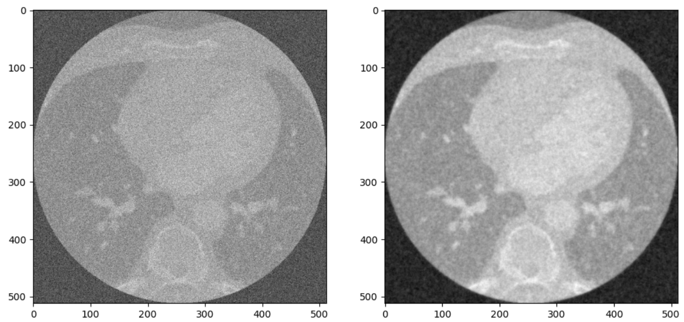

Figure 5: Tikhonov denoised image with variance equal to 40.

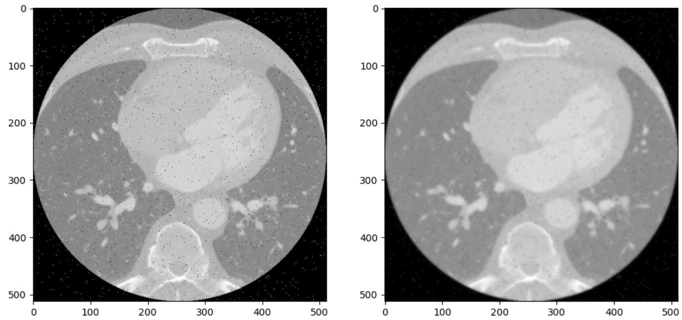

Figure 6: Tikhonov denoised image with Salt & Pepper noise

3.2 Perona-Malik results

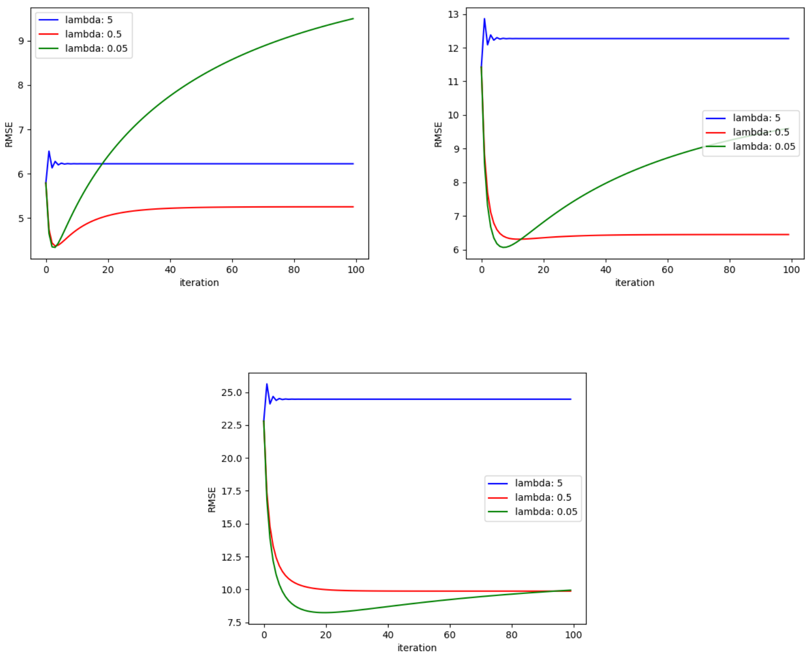

Figure 7: RMSE behavior with different λ values, for Perona-Malik images with σ equal to 10, 20 and 40.

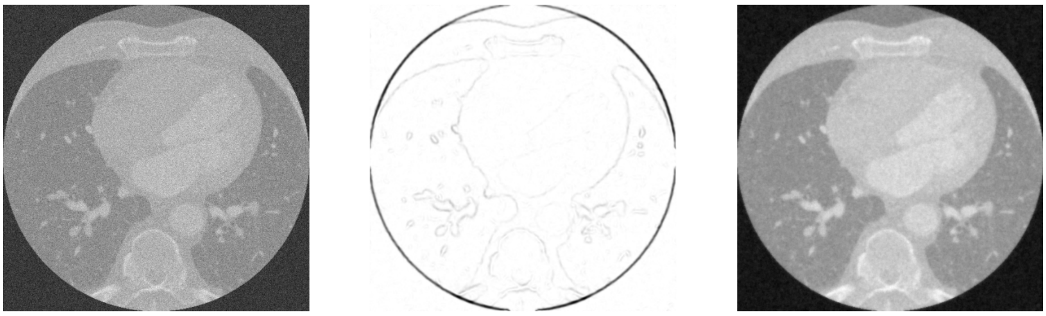

Figure 8: Perona-Malik denoised image with variance equal to 20.

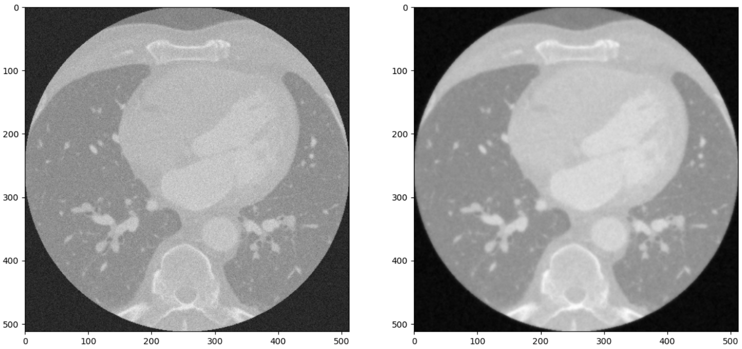

Figure 9: Perona-Malik denoised image with variance equal to 10.

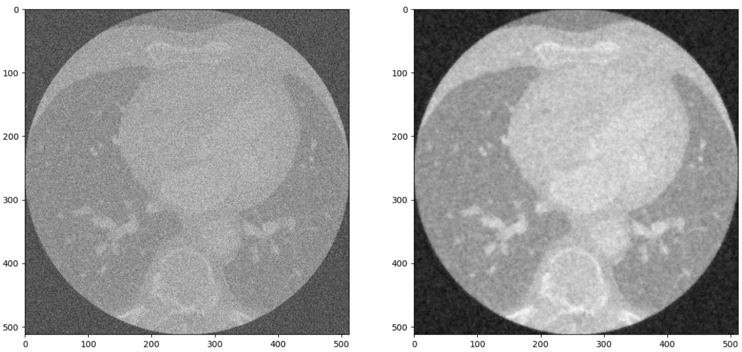

Figure 10: Perona-Malik denoised image with variance equal to 40.

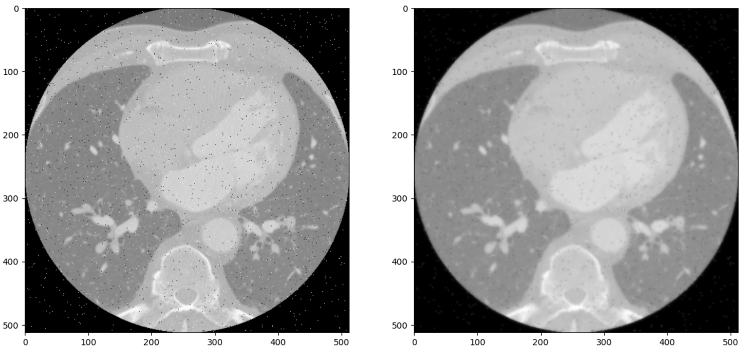

Figure 11: Perona-Malik denoised image with Salt & Pepper noise.

3.3 Total Variation results

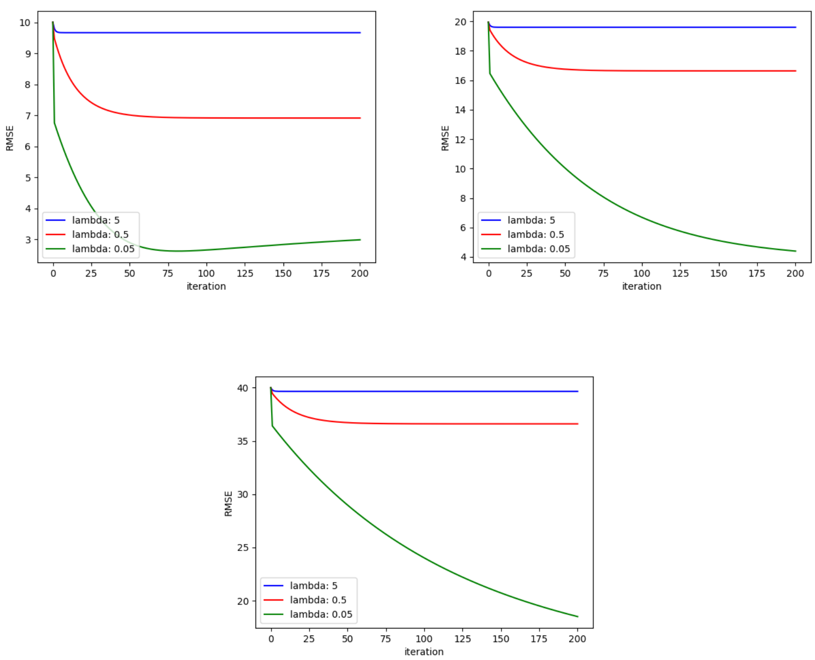

Figure 12: RMSE behaviour with different λ values, for TV images with σ equal to 10, 20 and 40.

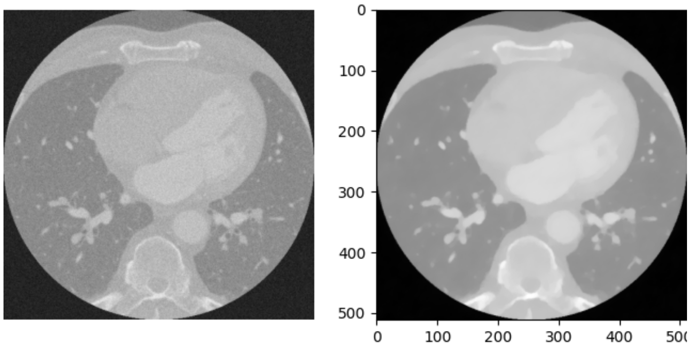

Figure 13: TV denoised image with variance equal to 10.

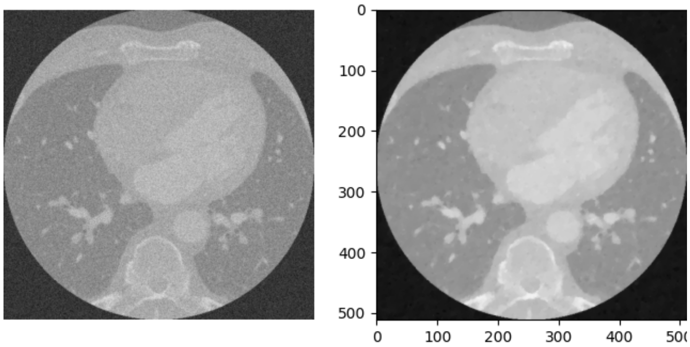

Figure 14: TV denoised image with variance equal to 20.

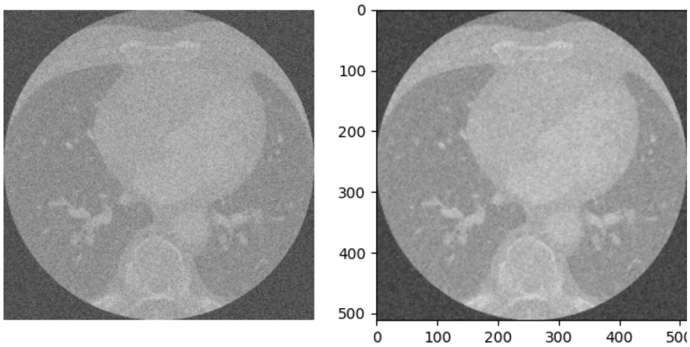

Figure 15: TV denoised image with variance equal to 40.

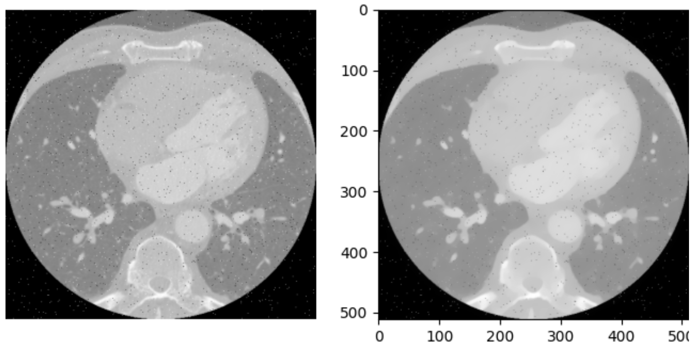

Figure 16: TV denoised image with Salt & Pepper noise.

3.4 Primal-Dual results

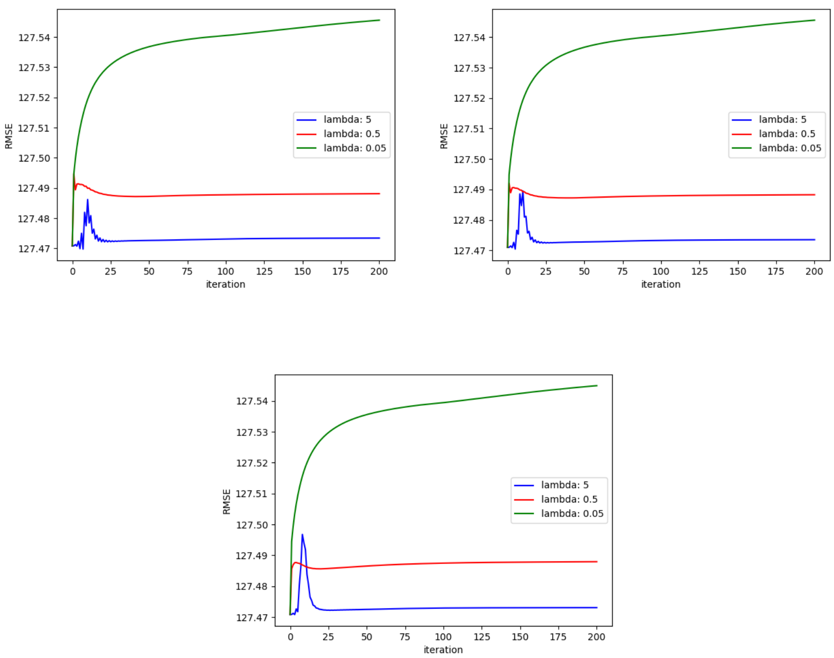

Figure 17: RMSE behavior with different λ values, for Primal-Dual images with σ equal to 10, 20 and 40.

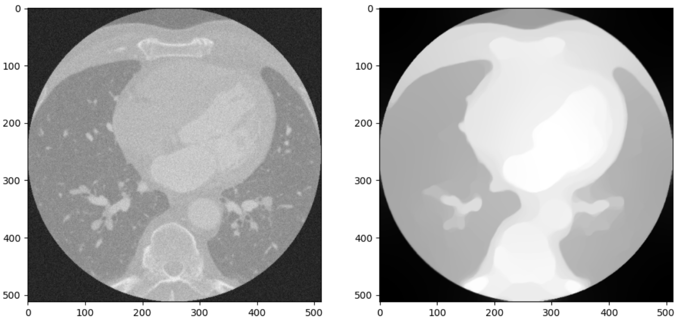

Figure 18: Primal-Dual denoised image with variance equal to 10.

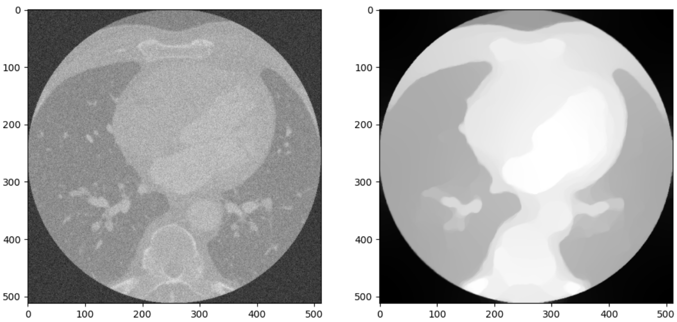

Figure 19: Primal-Dual denoised image with variance equal to 20.

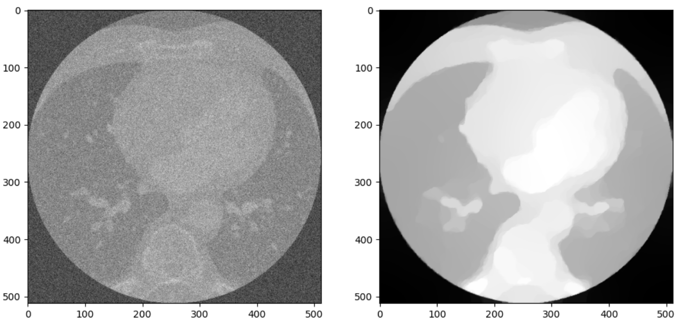

Figure 20: Primal-Dual denoised image with variance equal to 40.

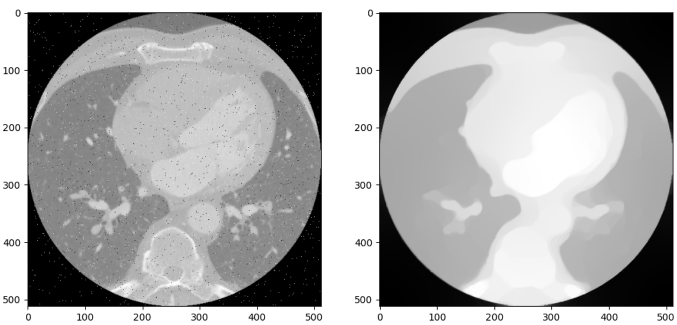

Figure 21: Primal-Dual denoised image with Salt & Pepper noise.

4 Discussion

4.1 Tikhonov-regularized & Perona-Malik denoising

As it can be seen from Figure 2, as the regularization parameter λ decreases, the space of possible solutions increases, so that the RMSE decreases until convergence for images with Gaussian noise with variance equal to 10 and 20. For the Gaussian noise equal to 40, on the other hand, the best performance in terms of RMSE is achieved with λ equal to 0.05.
As a consequence, the more the image is noisy, the better it is to look for a wider range of possible solutions, thus using a lower value of the regularization parameter λ.
Differently from Tikhonov, the Perona-Malik algorithm uses a diffusion tensor dependent on the image gradient Δf. This allows to preserve the edges better and to obtain a less blurred image.

After selecting as optimal λ the value 0.5, it is possible to observe the effect of the denoising algorithm on the different noisy images: as already mentioned, the chosen parameter acts correctly on the first 2 images. For the third image (with noise equal to 40), the result is not satisfying. Choosing a lower λ would probably lead to a better final image.
Finally, the algorithm has been applied also to the image with Salt & Pepper noise and as it can be seen, the present outliers are not removed in the final result. This is because the Tikhonov and the Perona-Malik denoising methods imply the use of the quadratic L2-norm of the variable u, thus not being able to remove random outliers present in the image.

4.2 Total Variation denoising

As it can be seen from Figure 12, for the Total Variation method the regularization parameter that better performs in terms of RMSE is the one equal to 0.05. In fact, the smaller the regularization term is, the more weak edges are removed.
The results also show how slowly is the method converging compared to the other algorithms. In fact, the TV algorithm is harder to minimize with respect to other methods, since it exploits only the L2-norm and not its quadratic version. This is the reason why an additional variable ϵ is introduced: in fact, ϵ avoids the divisor to reach a zero value, thus allowing the convergence of the derivative.
Additionally, by minimizing the L2-norm instead of its quadratic version, the method preserves better the sharp edges of the image without over smoothing them and this can be seen by comparing the denoised results of Tikhonov (Figure 6) with the results of the TV method

(Figure 14). If we consider a Salt & Pepper noise, however, the method is not performing optimally for the chosen λ, but an even smaller value would be necessary.

4.3 Primal-Dual denoising

For this last method, the best performant λ is the one equal to 2. This is a convex functional problem, so a bigger λ is sufficient to achieve a good result in denoising and this can be seen in the obtained images for all the considered noises (Figure 18, 19, 20). In fact, this method is not dependent anymore on the gradient of u, but on the dual variable $\vec{p}$ that depends on the iso-intensity curves. This different implementation makes the method more robust, and allows to obtain good performance for higher values of the regularization term λ.
Additionally, this last method is better also in the removal of Salt & Pepper noise, with respect to the other algorithms (Figure 21). The choice of a high λ is sufficient to remove this type of noise: a smaller parameter would not be good, since it would remove not only the noise, but also relevant edges of the original image.

5 Conclusion

Image denoising is one of the main issues in Medical Image Analysis and many methods can be exploited to face it. However, denoising problems are task and image-dependent. Methods like Tikhonov denoising can be sufficiently performant for certain types of noise and images, while more sophisticated methods such as Primal-Dual can be adopted for more complex noises like Salt and Pepper.
Additionally, each method necessarily requires an accurate choice of its parameters to achieve a good result, like the regularization λ and the time step τ for the convergence of the solution.
In conclusion, it can be said that image denoising analysis needs to be solved through a trial and error approach, by looking how does the solution change, varying the denoising method and its chosen parameters.

References

[1] Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[2] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 34:8–34, 2008.