REPORT FOR THE OPTIMIZED BLOCKWISE NON-LOCAL MEANS DENOISING FILTER FOR 3-D IMAGES

Xavier Sumba
40086818
Department of ECE
Concordia University

Masuda Akter Tonima
40094221
Department of MIAE
Concordia University

December 27, 2019

ABSTRACT

Noise is a problem that affects all kinds of signals. This problem becomes a dangerous issue when it comes to medical imaging, if denoising is not done properly, the results could lead to faulty diagnosis and thus faulty treatment. One crucial issue is denoising the image without compromising the original data which not only increases chances of proper diagnosis but also improves performance of the quantitative image analysis. This project implements 2 types of Non-Local means and a two optimization methods to filter the image's noise and recreate the results achieved by Coupé et al. [1]. We demonstrate that the OBNLM performs as good as the NLM at a lower computational cost.

Keywords Non-Local Means Filter · Blockwise Non-Local Means Filter · Optimizations

1 Introduction

Noise is any unwanted random signal that is produced due to random variations or electronic noise and might cause problems during the interpretation of an image. Specifically, in the medical context, this added noise might result in misdiagnosis or misinterpretations, which can be not only risky but life-threatening. There can be different kinds of noise; for example, to name a few we have periodic, aperiodic, Gaussian, fractal, salt and pepper, speckle, photon, gamma, Rayleigh and many more. In practice, noise is found to be always present in digital images during image acquisition or transmission [2]. Removing these noise types can be achieved in various ways.
Early image processing techniques utilized linear filters for enhancement and restoration of the images due to the ease of implementation and its performance during application. Although this can be true for many instances, linear filters fail to perform well in presence of additive, nonlinear or Gaussian noise; [3], points out that removing this types of noise can result in blurriness and according to [1], noise removal while maintaining the image integrity is crucial, especially for MRI images due to small structures and the importance of each pixel; thus even a small amount of noise deteriorates the useful data. To achieve proper denoising, in their work, they propose the Optimized Non-Local Means filter for 3-D images. They built on top of the classical Non-Local Means (NLM) and propose three main contributions: 1) Blockwise Non-Local Means (BNLM); 2) automatic smoothing parameter; 3) optimal selection of voxels in the search volume. In this work, we re-implement their work and analyze different implementations of NLM. First, in Section 2 we summarize the main contributions of the paper. Section 3 contains some notes during the implementation. Finally, we show some results and conclusions in Section 4 and 5 respectively.

2 Denoising filters

Medical images commonly contain noisy measurements and the goal of denoising filters is to remove this noise, facilitating the interpretability of a specific diagnosis. In general, several filters have been proposed and attempted to recover a cleaned intensity value by averaging different pixels. These filters can be categorised as such:

1. Data-independent: applies the same filter to the image and tends to be computationally effective but resulting images are blurred. Some examples of filters in this category are the box filtering [4] and the well known Gaussian filter.
2. Data-dependent: recovers the original pixel based on the noisy image's pixel. Filters in this category range from the ones that use simple calculations (median filter) to filters that consider a neighborhood (Non-Local Means (NLM) and Blockwise Non-Local Means (BNLM)) these methods are effective since they preserve the edges while successfully removing noise.

In this section, we describe the main contributions of the paper [1] and the Optimised BNLM upon which our report is built.

2.1 Non-Local Means

The Non-Local Means (NLM) filter [5] is based on the intuition that if we take multiple noisy snapshots and average them, the noise is removed. This is true, since the value of the pixel depends on the spatial neighbors of the pixels of the same community. This method can be applied either in 2D or 3D images.

Thus, given a noisy image $u = \{u(x_i) \mid x_i \in I^3\}$, the denoised pixels can be determined by calculating the weighted average of the voxel intensities. For instance, $NL(u)(x_i) = \sum_{x_j \in \mathcal{N}^3} w(x_j, x_i)u(x_i)$, where the weights w quantifies the similarities of local neighbourhoods N_i and N_j with voxel x_j and x_i, under the assumption that the weights of block x_i is between [0,1] and the sum of weights is 1 while restoring $u(x_i)$.

Classically, voxels are allowed to be connected to others but practically the number of voxels that are considered in the weighted average is limited to the search volume V_i.
For computational simplicity, this search volume V_i is defined to be of size $(2M+1)^3$, centered at the current voxel x_i. Since pixels might be different, we calculate the weights based on the similarity of different neighbors N_i of size $(2d+1)^3$, see Figure 1[1].

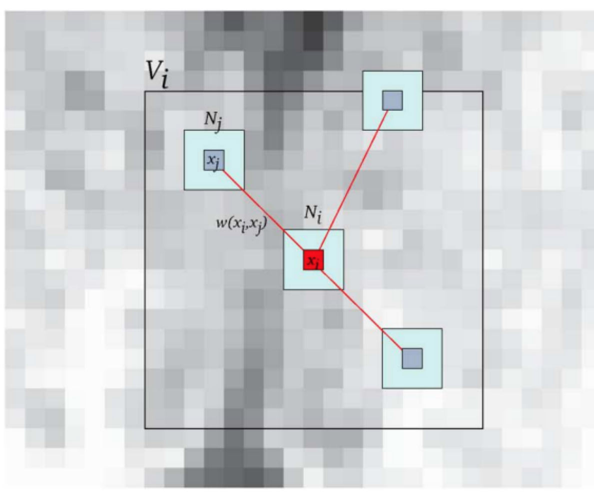


Figure 1: Non-Local Means Filter with values $d = 1$ and $M = 8$.

Consequently, the similarity distance is computed with the Gaussian-weighted Euclidean distance with standard deviation a for the Gaussian kernel, an smoothing parameter h, which controls the decaying and Z_i, and the normalization constant.

$$w(x_i, x_j) = \frac{1}{Z_i} \exp \frac{\| u(N_i) - u(N_j) \|_{2,a}^2}{h^2} \tag{1}$$

Large values of h result in all x_j's in the search volume to have the same weight in comparison to x_i, as a result, the restored intensity becomes the average of all the voxels in the search volume V_i, thus the image is nicely smoothed; whereas a small h causes decay to be strong, so only a small amount of voxels in the search volume have significant

[1] Figure taken from the original paper [1]

weight, and the restored intensity will be the weighted avg of some voxels with similar neighborhoods, thus very weak smoothing [1].

2.2 Block-wise Non-Local Means

The Block-Wise NLM is computationally cheaper compared with NLM. In the BLNM filter, first, the volume Ω^3 is partitioned in overlapping blocks B_{i_k} while making sure none of the overlapping blocks are empty and that the size is $(2\alpha+1)^3$. Additionally, centres of these blocks are x_{i_k}—a subset of Ω^3 that is equally distributed at i_k with n being the distance between the centre voxels, where $i_k = (k_1 n, k_2 n, k_3 n), (k_1, k_2, k_3) \in N^3$

Furthermore, BNLM finds the restored blocks. This process is similar to finding restored value of voxels except that the voxels are replaced with blocks such that the new restored block are $NL(u)(B_{i_k}) = \sum_{B_j \in V_{i_k}} w(B_{i_k}, B_j) u(B_j)$, and instead of calculating the distance between voxels as in NLM (see eq 1), weights are computed between blocks.

$$w(B_{i_k}, B_j) = \frac{1}{Z_{i_k}} \exp \frac{\| u(B_{i_k}) - u(B_j) \|^2_{2,a}}{h^2} \tag{2}$$

Finally, since a voxel x_i might be included in several estimations $A_i = NL(u)(B_{i_k})$, we average them in order to obtain the final restored voxel. For instance, $NL(u)(x_i) = \frac{1}{|A|} \sum_{p \in A_i} A_i(p)$. This procedure is illustrated in Figure 2[2].

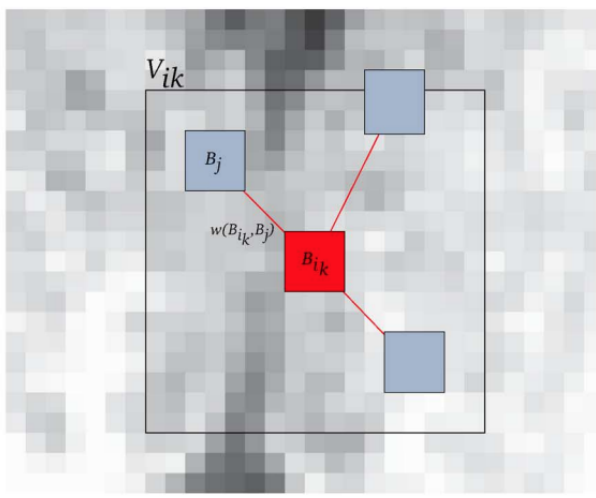


Figure 2: Block-wise Non-Local Means with values $\alpha = 1$ and $M = 8$

2.3 Optimizations

In order to speed up computation and hyper-parameter tuning, the authors also proposed a method to find an optimal smoothing parameter and selection of important voxels in the search volume. The execution strategy of these methods are given below, which is followed by their imposition and affect on denoising of the images.

2.3.1 Automatic tuning of the smoothing parameter

Automatic tuning of smoothing parameter instead of a fixed h; this allows the system to get an optimal value of h by adjusting σ and β. Ideally h is expected to be 10σ.
According to [1] the relationship between h and σ, and β is estimated by finding the standard deviation $\hat{\sigma}$, which can be done by using pseudo-residuals ϵ_i [6].
We calculate $\hat{\sigma}^2 = \frac{1}{|\Omega^3|} \sum_{i \in \Omega^3} \epsilon_i^2$ such that the smoothing parameter is automated.
Finally, in order to compute the weights in NLM or BNLM, the authors suggest the use the Euclidean distance, and the estimated smoothing parameter:

$$h = 2\beta\hat{\sigma}^2 \tag{3}$$

[2]Figure taken from the original paper [1]

Although Coupé et al. [1], takes into account the cubic local neighbourhood size $|N_i|$ in order to make the filter independent of the local neighbourhood; we, however, position this independence factor with the weighing rather than the smoothing parameter calculation; this results in no adverse affect on our results.

2.3.2 Voxel selection in search volume

Among all the voxels in a 3-d image, only the most relevant voxels, i.e voxels with the highest weight in a given volume, are pre-selected in this method. These pre-selected voxels, and the subject voxel are compared and their Euclidean distance is calculated. Any voxel with minimal weight is completely ignored.
This, simple rule, allows the system to avoid computation for all the irrelevant voxels in of the image and thus speeds up the system. Selection of these similar voxels are based on the mean value of voxels containing the intensities i and j and their respective neighbourhoods.The mathematical representation of this idea can be written as:

$$w(x_i, x_j) = \begin{cases} \frac{1}{Z_i} \exp \frac{\|u(N_i)-u(N_j)\|_2^2}{2\beta\sigma^2|N_i|}, & \text{if } \mu_1 < \frac{u(N_i)}{u(N_j)} < \frac{1}{\mu_1} \text{ and } \sigma_1^2 < \frac{Var(u(N_i))}{Var(u(N_j))} < \frac{1}{\sigma_1^2} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

This condition allows the NLM filter to differentiate between the edges and noise pixels in flat regions, thus preserving the details better, although this might result in flat regions being overly smooth, but it can be negligible in cases where the pixels of the flat area are similar.

2.4 Theoretical comparison

According to the Coupé et al [1], the usage of these methods somewhat, albeit not greatly, varies the results of denoising. There are some advantages and some disadvantages to each of the methods. The summary of their expectations are noted in Table 1:

Approaches		Advantage	Disadvantage
NLM		reduces noise with really good performance for 2-d images	computationally expensive
ONLM	auto smoothing parameter	more efficient denoising	dependent upon β - manual input
	voxel pre-selection	reduces unnecessary computation	must pre-calculate voxels for each different kind of image
BNLM		performs really well with 3-d images	For 2-d images this is a heavy computation
OBNLM		performs best of all	Not known

Table 1: Comparison of the systems and their expected dis/advantage

3 Implementation

We implemented the methods proposed in the paper and created a library in MATLAB for NLM, BNLM, ONLM, and OBNLM. Here, we describe the main methods of our library along with the problems faced while developing it.

3.1 Execution

The project contain three main functions with the following implementations:

nlm(*img, d, M, h, a*) $\rightarrow$ **restored image** : Implementation of NLM that receives an image with noise. M and d are used to calculate the neighborhood and search volume. h is the smoothing parameter, and a is the standard deviation for the Gaussian kernel.

bnlm(*img, alpha, M, n, h, a*) $\rightarrow$ **restored image** : Implementation of BNLM that receives a noisy image, and alpha, M and d are used to compute the overlapping blocks and search volume. h is the smoothing parameter, and a is the standard deviation for the Gaussian kernel.

onlm : Implementation of the tasks in optimization were done separately. Where part (a) automates the smoothing parameter and (b) performs the voxel pre-selection. Both systems are designed as independent blocks and free of external functions, following the theory mentioned in section 2.3.

obnlm : Implements the combination of BNLM and the optimization methods.

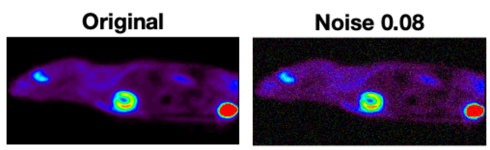
3.2 Changes made

In order to make the 3d MRI image more visually pleasing, the dimensions were assigned to colour channels RGB. This allows, although not altering any result, to be able to view the images fully using simple command. The smoothing parameter calculation in this paper does not utilize the cubic local neighbourhood size to make the smoothing parameter independent of the latter, instead it was adjusted during the weight calculation as stated before. The system was tested using two different types of image entirely to test its performance and efficiency, unlike its mother paper which only does so on brain images (This verifies that the system is capable of denoising any image).

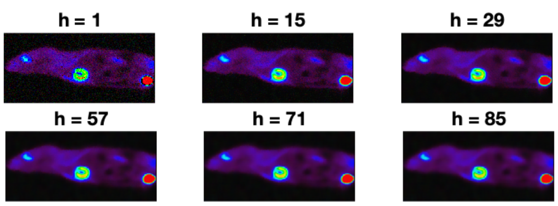
4 Results

In this section, we present some of the results and analysis done with NLM. We tested our implementation with two colour images, a PET image and a brain image. The experiment is as follows: we perturb the original image by adding Gaussian noise and perform image denoising with the different improvements of the vanilla NLM.

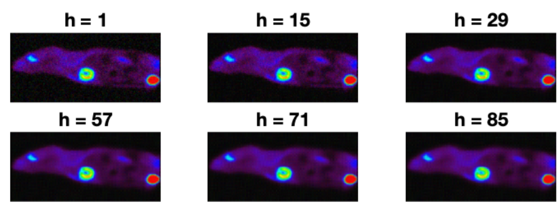
Figure 3 and 4 illustrates the performance of the systems with a a PET image, where Figure 3(a) is the side by side comparison of the original image and its noisy version. Figure 3(b) show the implementation of Non-Local Means and Blockwise Non-Local Means for different values of the smoothing parameter. We can notice that for bigger values the image is smoother. This is furthered by Figure 4, that show the results of the implementation of automated smoothing parameter sub-system (a) and the voxel pre-selection subsystem (b), respectively.



(a) Original and Noisy image

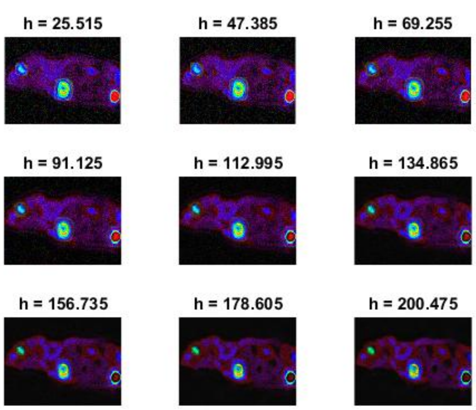


(b) NLM

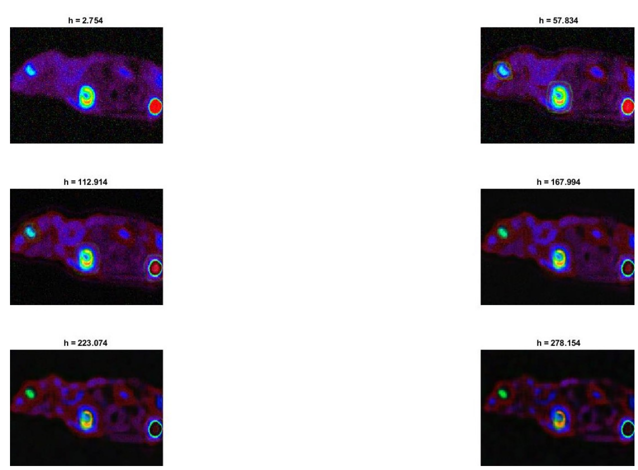


(c) BNLM

Figure 3: Results of variations of NLM filter in a 3D PET image



(a) ONLM optimal voxel selection



(b) ONLM automatic smoothing parameter

Figure 4: Results of variations of ONLM filter in a 3D PET image

Both of these implementations show a point in the smoothing parameter that visually appeases the noisiness (centre image in Figure 4a, where h=112.995, and right top image for Figure 4b,where $h = 57.83$) after which however, both the systems work in a counter productive manner and fail to maintain the integrity of the original image itself, by loosing details, edge information and over-smoothing. In contrast to this, the Optimized Blockwise NLM performs better and solves this issue as it can be visually verified in Figure 5, despite the rising values of the estimated smoothing parameter, the OBNLM filter smooths the noise without compromising, detail, edge/colour information, i.e. the integrity of the original image.

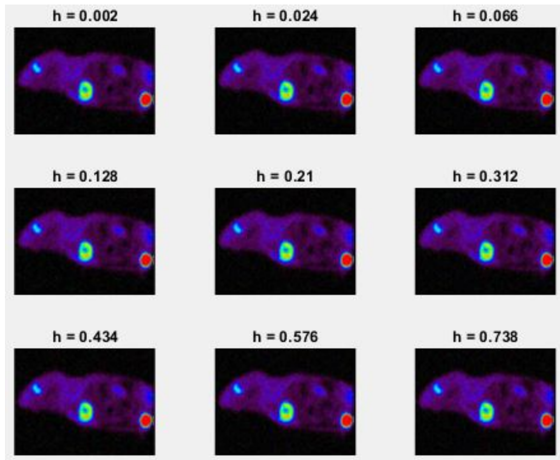
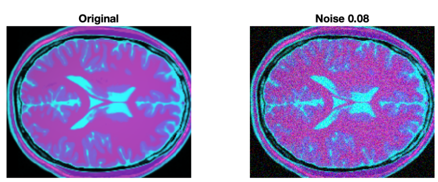
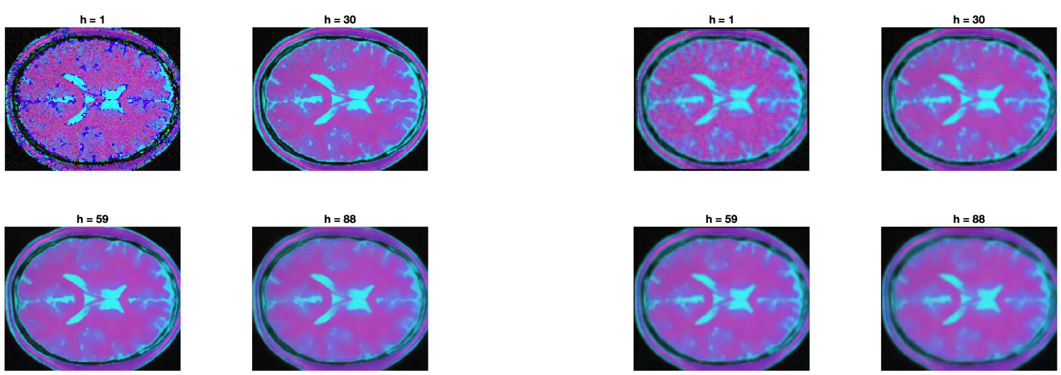


Figure 5: Result of Optimised Blockwise Non-Local Means filter implemented on PET image

Next, we experimented with a brain image and combined three different sequences of the same image to convert it to a colour image. The ground truth for brain image and its noisy counterpart (perturbed with a standard deviation of 0.08 for the Gaussian noise added), is compared side by side in figure 6a. This is then followed by the performance of NLM and BNLM with an increasing smoothing parameter. Although the smoothing effect of this factor is visually evident in both Figure 6b and 6c, we can also visually verify that with higher smoothing parameter, edge detail is lost.



(a) Original and Noisy image

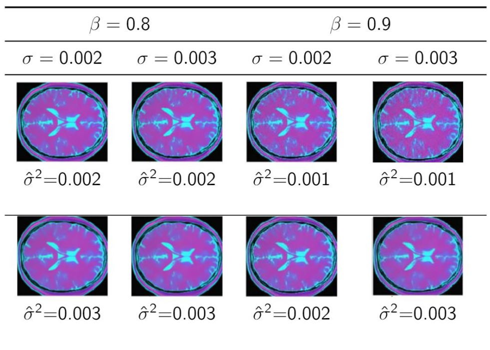


(b) NLM

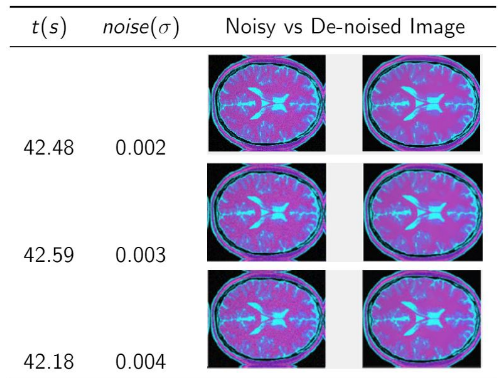
(c) BNLM

Figure 6: Results of variations of NLM filter in a 3D Brain image

This issue is remedied in Figure 7, which shows the comparison of performance between the two optimizing methods. It is visually verified that both methods successfully denoise the brain image without losing any colour or edge details. This, along with the Table 2, confirms the theoretical suggestions table 1 to be true for the automatic smoothing parameter choosing. For instance, better denoiser and voxel pre-selector save time from useless computation.



(a) ONLM automatic smoothing parameter



(b) ONLM optimal voxel selection

Figure 7: Results of variations of ONLM filter in a 3D Brain image

Finally, similarly to the PET results, we combine both optimization methods with BNLM, resulting in the optimized BNLM. Results of this filter are shown in Figure 8 along with its estimated smoothing parameter. We can observe that very large values smooths the image to the point that some edges are lost.

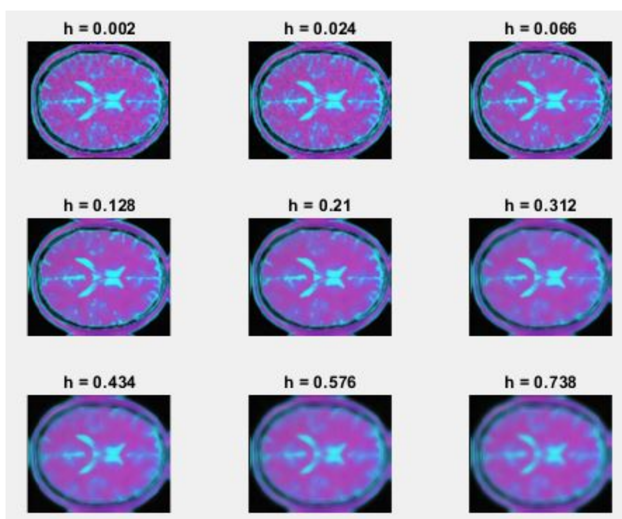


Figure 8: Result of optimised blockwise Non-Local means filter implemented on 3-d brain image

Since one of the main contributions of his work was to build a computational efficient filter for 3D images, we compare the time that takes to denoise with different variations of NLM. Table 2 shows the time taken on average per iteration for each image and method, in two different machines,[3] For both of these implementations it can be seen that the Optimized BNLM reduces computation by a large margin in contrast to their non-optimised counterpart. Furthermore, it is noteworthy that time efficiency is highly dependent on the capabilities of the machine being used.

[3] This computation was done using an Intel i7 processor, ram 2GB, partially dedicated, using only CPU. Further, upon implementing the same code on a upgraded computer, 9th generation 2.6GHz six-core Intel Core i7-9750H processor that clocks up to 4.5GHz, 16GB of speedy DDR4 RAM, and a GPU, the results vary noticeably.

Method	Time (CPU sec)		Time (GPU sec)	
	PET	Brain	PET	Brain
NLM	77.18	66.28	57.26	46.23
ONLM (Automatic h tuning)	46.33	56.27	31.22	31.82
ONLM (voxel selection)	**44.5**	41.85	30.30	32.16
BNLM	46.94	41.31	29.20	32.22
OBNLM	47.88	**40.66**	**28.5**	**31.20**

Table 2: Time efficiency: comparison of different implementations of NLMs with PET and Brain images. Results are averaged over several runs on simulations run on two different machines. NLM: Non-Local Means; BNLM: Blockwise Non-Local Means; ONLM: Optimized Non-Local Means; OBNLM: Optimized Blockwise Non-Local Means.

5 Conclusions

In this work, we analyze the OBNLM filter and implement the main paper fully. We have replicated the methods of inserting noise and all the denoising techniques from the sourcer paper and compared the results of these implementations. We notice the affect of smoothing parameter in classical NLM and how the results of this denoising filter changes with varying factors implemented here. Finally we see that the proposed optimsed method performs just as well as the vanilla NLM filter but only at a lower computational cost, which fulfils the aim of this project.

The application of the denoising techniques administered here is including but not limited to the medical field. Combinational techniques such as the optimized blockwise Non-Local means perform well enough to be used in vision-based control surveillance, control, depth estimation and etc. With an added layer of classification, based on a simple regression model, a mother system may be built that can effectively classify the image and use the most effective method to de-noise the image.

Statement of Contributions

Both members contributed to the project equally. The Non-Local Means and Blockwise Non-Local Means were implemented by Xavier. Tonima worked on optimization for both the algorithms. Both the authors contributed to experimentation, presentation, and writing the paper.

References

[1] Pierrick Coupé, Pierre Yger, Sylvain Prima, Pierre Hellier, Charles Kervrann, and Christian Barillot. An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images. *IEEE transactions on medical imaging*, 27(4):425–441, 2008.

[2] Ajay Kumar Boyat and Brijendra Kumar Joshi. A review paper: noise models in digital image processing. *arXiv preprint arXiv:1505.03489*, 2015.

[3] Gnanambal Ilango and R Marudhachalam. New hybrid filtering techniques for removal of gaussian noise from medical images. *ARPN Journal of Engineering and Applied Sciences*, 6(2):8–12, 2011.

[4] MJ McDonnell. Box-filtering techniques. *Computer Graphics and Image Processing*, 17(1):65–70, 1981.

[5] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[6] Theo Gasser, Lothar Sroka, and Christine Jennen-Steinmetz. Residual variance and residual pattern in nonlinear regression. *Biometrika*, 73(3):625–633, 1986.