Image Restoration using the Bilateral and Non-Local means Filters

Julia Bazarbachian
26781357

Kajanthy Subramaniam
40063712

***Abstract*— In this report, we discuss the noise-reduction capabilities of the bilateral filter and the non-local means filter with respect to gaussian & salt and pepper noise. The bilateral filter for image denoising applications succeeds in smoothing images while preserving edges. It does so by leveraging a nonlinear combination of neighbouring image values. The non-local means filter restores images by removing unwanted frequencies while retaining image textures that would be blurred out by other denoising algorithms. The goal of this report is to perform image restoration using both the bilateral and the non-local means filter in order to collect experimental data. The result of their performance is then compared with those of median and mean filters for denoising grayscale and colored images.**

Keywords—image denoising, bilateral filter, non-local means filter, mean filter, median filter, MSE, PSNR, SSIM

I. INTRODUCTION

A. Image Denoising

The goal of image restoration is to attempt to recover a corrupted image by leveraging knowledge of its degradation. Restoration techniques therefore tend to start by modelling the noise and then follow with applying the inverse process in an attempt to recover the original image [1]. Several different kinds of image restoration solutions have been introduced in order to remove noise from digital images, including linear, nonlinear, and adaptive spatial filters. Where many of them fall short is that, in addition to removing unwanted frequencies they also remove fine details and structure of the images being processed. The bilateral and non-local means filters are two denoising algorithms that were introduced to blur images while respectively preserving edges and texture.

This work will analyze the performances of the bilateral and non-local means filters and compare experimental results with other chosen denoising filters, specifically the mean and median filters. The degradation we will attempt to remove from natural images will be that caused by additive noise. The noise models we will be using for the purpose of corruption simulation are Gaussian noise and Salt-and-Pepper noise.

B. Overview of Research Paper

In this report, we will conduct an image noise reduction comparison between filters. In the following section, we formalize the notion of bilateral and non-local means filtering by presenting a theoretical introduction to these algorithms. Section 3 shows experimental results for our chosen image denoising filters by displaying visual quality and performance metrics comparisons. Section 4 discusses the performance of the bilateral and non-local means filters according to our chosen criteria. Lastly, the results of our experiment are summarized in section 5.

II. IMAGE DENOISING METHODS

The bilateral filter is an image restoration solution that uses a non-linear technique to blur an image while preserving edges. It has become ubiquitous in the fields of image processing and computer vision, with applications including image interpolation, medical imaging, and video enhancement [3]. The Non-Local Means Filter denoises edges without losing the finer details of the image structure. It does this by utilizing the similarity of local patches to determine pixel weights [5].

In this section, we provide a theoretical foundation to bilateral filtering (BF) and non-local means (NL-means) filtering.

A. Bilateral Filter

The BF algorithm for image denoising is an edge preserving image smoothing technique that is currently widely used in computational photography. The advantages offered by the Bilateral Filter can be summarized by the following main factors [1]:

1. It has a simple edge-preserving image restoration algorithm consisting of replacing each pixel by a weighted average of its neighbours. This allows for easy implementation and domain specific applications.
2. It relies on two parameters: the size and contrast of the features to preserve
3. It can be used non iteratively, making parameters easy to set.

The bilateral filter outputs a smoother version of an input image by removing noise, texture, and small details while preserving large sharp edges.

Before describing the bilateral filter, we will introduce the Gaussian blur, also referred to Gaussian Convolution. The latter is a simpler smoothing filter that does not preserve edges but is needed to introduce the concept of local averaging. The bilateral filter then combines image blurring with edge preservation [1].

Smoothing images with Gaussian Convolution

The equation for an image smoothed with the Gaussian Convolution is as follows [1]:

$$GC\ [I]_p = \sum_{q \in S} G_\sigma(\|p-q\|) I_q$$

Where:

- $GC\ [I]_p$ is the gaussian convolution of the image value at pixel position p
- $\sum_{q \in S}$ is the sum over all image pixels indexed by q
- $\|p-q\|$ is the Euclidean distance between pixel locations p and q
- σ is a parameter defining neighborhood size
- $G_\sigma(x)$ denotes the 2D Gaussian kernel

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2}{2\sigma^2}\right)$$

To smooth an image using the Gaussian blur, each output image pixel value is a weighted sum of its neighbours in the input image. The influence of one pixel over another depends solely on their distance in the image, irrespective of the actual image values. Consequently, image edges are blurred at high values of σ because averaging is performed over larger areas, as can be observed in Figure 2.1.

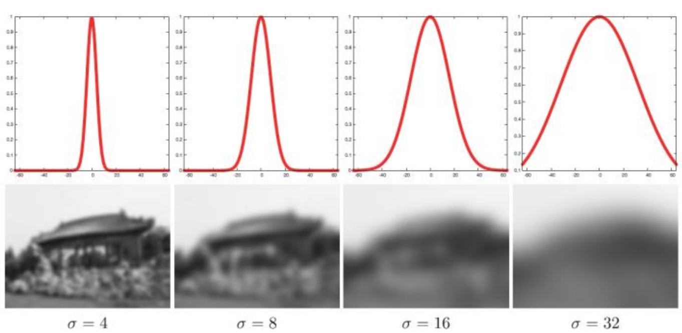


Fig 2.1 Gaussian Convolution with different values of σ. This figure is reproduced from [1].

Edge-preserving smoothing with the Bilateral Filter

The bilateral filter combines the smoothing abilities of the Gaussian blur along with edge preservation. In addition to computing the weighted average of nearby pixels, the bilateral filter also considers the difference in value with the neighbours.

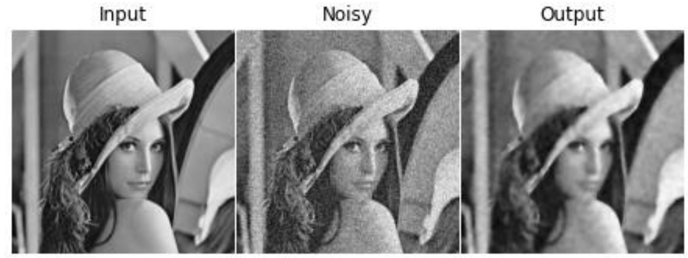


Fig 2.2 The Bilateral Filter applied on Lena image.

We model the Bilateral Filter as follows [1]:

$$BF\ [I]_P = \frac{1}{W_p} \sum_{q \epsilon S} G_{\sigma_s}(\|p-q\|)\ G_{\sigma_r}(|I_p - I_q|)\ I_q$$

Where:

- G_{σ_s} is a spatial domain kernel that decreases the influence of a distant pixel
- G_{σ_r} is a range domain kernel that decreases the influence of pixels q when their intensity values differ from I_p.
- W_p denotes the normalization factor ensuring that pixel weights sum to 1.0

$$W_P = \sum_{q \epsilon S} G_{\sigma_s}(\|p-q\|)\ G_{\sigma_r}(|I_p - I_q|)$$

The spatial component penalizes distant pixels while the range component penalizes pixels with a different intensity. Combining both components, we end up with only nearby similar pixels contributing to the final result, as illustrated in Figure 2.3 [1].

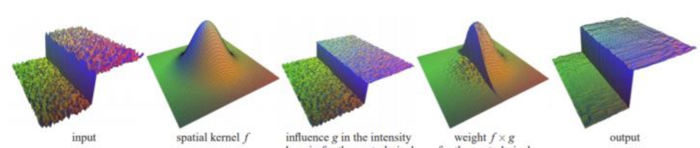


Fig 2.3 Illustration of Bilateral Filter. This figure is reproduced from [4].

The product of G_{σ_s} and G_{σ_r} determines the contribution of a pixel. As a result, no smoothing occurs if either of the weights σ_s/σ_r reach values near 0. Conversely, edges are no longer well preserved if those values become too large [3], as can be observed in Figure 2.4.

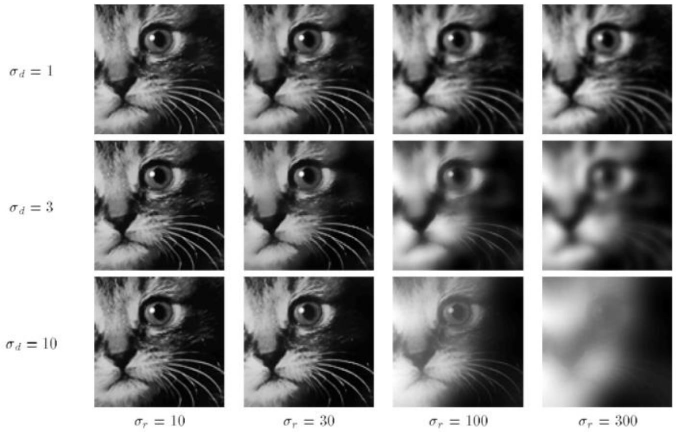


Fig 2.4. Bilateral Filter applied on cat image with varying parameter values. This figure is reproduced from [2].

Image denoising with the bilateral filter succeeds in noise removal while maintaining detailed edges. This advantage offered by the BF is due to its combination of two sources of information: weights determined by spatial proximity and range proximity. Unlike the Gaussian Blur, BF manages to preserve sharp details in the image contours while removing unwanted frequencies.

B. Non-Local Means Filter

The Non-Local Means Filter (NL-means) was proposed with the aim of preserving the fine structures, details and textures of images while maintaining a strong denoising effect [6]. The assumption behind the NL-means algorithm is that of self-similarity. In other words, we assume that there is a high level of redundancy in a natural image that can be leveraged to make self-predictions about the image itself. Thus, for every small frame in the image, we can find many other similar frames in other parts of the image [6].

Fig 2.5 The Non-Local Means Filter applied on Lena image.

We model the Non-Local Means Filter as follows [6]:

Letting v be the noisy image observation defined on domain $\Omega \subset \mathbb{R}^2$ and $x \in \Omega$

$$NL\,(v)(x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |v(X+.)-v(Y+.)|^2)(0)}{h^2}} v(y)dy$$

Where:

- G_a is a Gaussian kernel with standard deviation a
- h is a filtering parameter
- $C(x)$ is the normalizing factor

  $$C(x) = \int_{\Omega} e^{-\frac{(G_a * |v(X+.)-v(Z+.)|^2)(0)}{h^2}} dz$$
- and

  $$(G_a * |v(x+.) - v(y+.)|^2)(0) = \int_{\mathbb{R}^2} G_a(t)|v(x+t) - v(y+t)|^2 dt$$

The denoising effect of this algorithm is to estimate the value of x as an average of all pixel values whose gaussian neighborhood resembles the neighborhood of x.

Given a discrete noisy image, the equation for the NL-means algorithm becomes as follows [6]:

Let $v = \{v(i) |\, i \in I\}$

$$NL\,(v)(i) = \sum_{j \in I} w(i,j)v(j)$$

Where:

- $\{w(i,j)\}_j$ are weights that depend on the similarity between pixels i and j and satisfy the conditions

  $0 \leq w(i,j) \leq 1 \,, \sum_j w(i,j) = 1.$

  They are computed as

  $$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i)-v(N_j)\|_{2,a}^2}{h^2}}$$

Where:

- $Z(i)$ is the normalizing factor

$$Z(i) = \sum_{j} e^{-\frac{\|v(N_i)-v(N_j)\|_{2,a}^2}{h^2}}$$

- h is a parameter that controls the decay of the weights as a function of the Euclidean distances

For the purpose of computation, we restrict our averaging to the neighborhood window centered on the pixel of interest i [5].

Figure 2.6 displays an example of neighborhood windows on the Lena image. Comparing patches q1 and q2 to patch p, we can observe that they are very similar. Consequently, w(p, q1) and w(p, q2) are large. On the other hand, the weight of q3 is smaller because the intensity values in patch q3 are different from those in patch p [6].

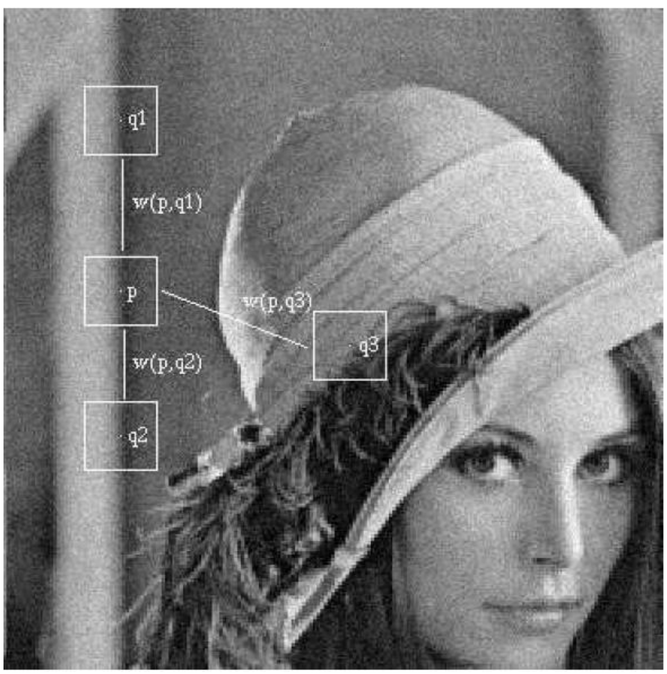


Fig 2.6 Patch Similarities with the NL-means Filter. This figure is reproduced from [6].

The result of the Non-Local means algorithm is therefore a filtered image that has preserved its texture and fine details. Figure 2.7 shows an example of the NL-means filter applied on a colored image of an aircraft. The noisy image and the restored image are displayed side by side for a comparison of the images' visual quality.

Fig 2.7 Non-Local Means filtering. The image on the left is the noisy input, while the one on the right is the restored output. This figure is reproduced from [6].

Due to its advantage of filtering noise while retaining image characteristics, the non-local means filter is favored by many researchers since Buardes and al. first introduced it in 2005 [6], and several improvements have been proposed throughout the years with the aim of improving its performance.

III. Experimental Results for Image Denoising

In this section, we compare the performance of four filters: mean, bilateral, nonlocal means, and median. We will also be looking at the performance of mean and median filters since they are more traditional filters. We will be applying the filters on four pictures: an MRI scan, a portrait picture of an astronaut, a dog, and a landscape picture of a scenery.

We will also be turning the same pictures to grayscale and applying the same filters. Before applying the filters, we will be introducing gaussian noise to the pictures before the use of the mean, bilateral, and nonlocal means filters. Salt and pepper noise will be introduced before the use of the median filter.

Once this is done, we will calculate the root mean square error, peak signal to noise ratio, and structural similarity index and use these as metrics to compare and evaluate the performance of the filters.

Code for the application of the filters and the calculation of the metrics can be found here in our public GitHub repository. We will also be conducting a visual comparison of the outputs to judge the performance of each of these filters.

A. Visualization of the filter results

TABLE I. MEAN FILTER

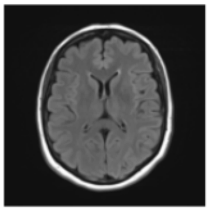
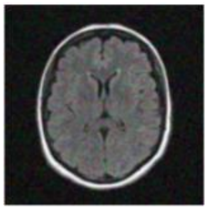
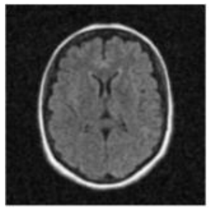
Original	Colour	Grayscale
		
		
		
		

TABLE II. BILATERAL FILTER

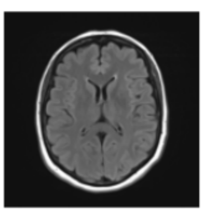
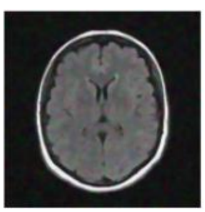
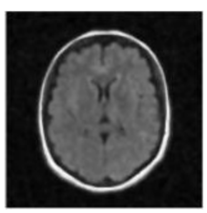
Original	Colour	Grayscale
		
		
		
		

TABLE III. NON-LOCAL MEANS FILTER

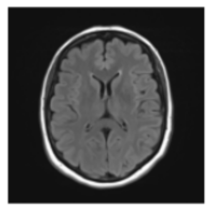
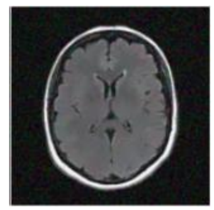
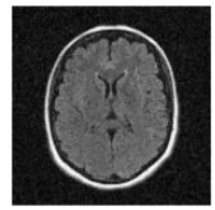
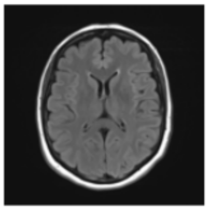
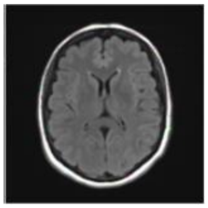
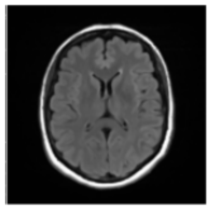
Original	Colour	Grayscale
		
		
		
		

TABLE IV. MEDIAN FILTER

Original	Colour	Grayscale
		
		
		
		

B. Performance Metrics

This section presents the performance metrics used to evaluate image filtering performance among our chosen denoising algorithms, and then displays the results of these metrics in their respective tables.

1) Root Mean square error (RMSE)

Original image f and restored image $\hat{f}$
Mean square error (MSE):

$$\text{MSE} = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left[f(m,n) - \hat{f}(m,n)\right]^2$$

The RMSE is a commonly used measure of image quality. It estimates the perceived errors. The closer this value is to zero, the more similar the two images are. This metric is influenced by the image intensity scaling. For example, a MSE of 100.0 for an 8-bit image (with pixel values in the range 0-255) is much more significant than a MSE of 100.0 for a 10-bit image (pixel values in [0,1023]) which is barely noticeable [7].

2) Peak Signal-to-Noise-Ratio (PSNR)

$$PSNR = -10log_{10}\frac{eMSE}{S^2}$$

The PSNR circumvents the problem seen with MSE by scaling the MSE according to the image range (S is the maximum pixel value). The higher the value of PSNR the better, when comparing the same image (denoised versions of the same image) [7].

3) Structural similarity index (SSIM)

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{({\mu_x}^2 + {\mu_y}^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

This formula was developed by Wang et al. The SSIM attempts to model the perceived change in the structural information of the image. SSIM values can vary between -1 to 1. A value of 1 would indicate perfect similarity. The lower the SSIM value, the more similar it is to the input image. In this equation, two windows are compared instead of the entire image as in MSE. This leads to having a more robust approach that is able to account for changes in the structure of the image, rather than just the perceived image [8]. As the RMSE *increases* the images are *less similar*, as opposed to the SSIM where *smaller values* indicate *less similarity*. By default, this value is estimated from the image datatype.

TABLE V. MEAN FILTER METRICS

	RMSE	**SSIM**	**PSNR**
MRI (Grayscale)	77.106	0.000312	10.389
MRI (Colour)	77.105	0.000324	10.389
Astronaut (Grayscale)	14.412	0.603	24.955
Astronaut (Colour)	14.713	0.601	24.776

Dog (Grayscale)	12.710	0.554	26.0477
Dog (Colour)	13.0537	0.553	25.816
Landscape (Grayscale)	28.855	0.459	18.926
Landscape (Colour)	28.823	0.473	18.935

TABLE VI. BILATERAL FILTER METRICS

	RMSE	**SSIM**	**PSNR**
MRI (Grayscale)	10.205	0.802	27.954
MRI (Colour)	8.421	0.761	29.623
Astronaut (Grayscale)	11.560	0.703	26.871
Astronaut (Colour)	10.0440	0.675	28.092
Dog (Grayscale)	8.449	0.742	29.594
Dog (Colour)	8.818	0.692	29.222
Landscape (Grayscale)	21.864	0.648	21.336
Landscape (Colour)	14.322	0.808	25.0106

TABLE VII. NON-LOCAL MEANS FILTER METRICS

	RMSE	**SSIM**	**PSNR**
MRI (Grayscale)	21.471	0.323	21.493
MRI (Colour)	21.471	0.323	21.493
Astronaut (Grayscale)	17.551	0.588	23.244
Astronaut (Colour)	15.187	0.653	24.501
Dog (Grayscale)	20.0870	0.464	22.0724
Dog (Colour)	17.129	0.527	23.455
Landscape (Grayscale)	20.973	0.750	21.697
Landscape (Colour)	17.420	0.798	23.309

TABLE VIII. MEDIAN FILTER METRICS

	RMSE	**SSIM**	**PSNR**
MRI (Grayscale)	4.296	0.977	35.468
MRI (Colour)	4.316	0.978	35.428
Astronaut (Grayscale)	6.810	0.953	31.467
Astronaut (Colour)	6.891	0.945	31.365
Dog (Grayscale)	4.305	0.932	35.449
Dog (Colour)	4.667	0.921	34.748
Landscape (Grayscale)	27.908	0.597	19.216
Landscape (Colour)	27.600	0.620	19.312

IV. DISCUSSION AND ANALYSIS

A. Visual Analysis

Visually, the median filter seems to have rendered the best results. Denoising using the bilateral filter seems to have left a blurring effect that we can clearly observe on the output images. The nonlocal means filter seems to leave behind a blurring effect as well. This can be observed with the picture of the astronaut, her skin seems smoothed out.

The mean filter seems to have performed the least best with the output images looking grainy after the application of the filter. This graininess can be especially seen in the MRI image. Visually, there does not seem to be a significant difference between the filtered grayscale images and the filtered coloured images. Something peculiar with the colour MRI output image (with all filters with the exception of nonlocal means and median) is that there seems to be some specks of colours.

B. Mean Filter

Comparisons between grayscale and colour images: There does seem to be a huge difference between the colour output and its grayscale counterpart. However, the colour images do seem to be denoised slightly better than the grayscale images.

Performance Metrics:

In terms of metrics, the mean filter is performing pretty poorly, especially with the MRI image where the value is very far away from 0. When looking at the SSIM value, given that a value of 1 would indicate perfect similarity between the input and the output image, the performance is pretty average as most of the SSIM values are in the middle of the possible range of values of -1 to 1. The highest performing image is astronaut in grayscale and the lowest is MRI in grayscale.

C. Bilateral Filter

Comparisons between grayscale and colour images: Again, there is no significant difference between grayscale and colour images but according to the SSIM (which is more reliable than the RMSE) the grayscale images with less details such as: MRI, astronaut, and dog images, performed better than the colour image. Whereas the Landscape image which has a lot of colour details performed better in grayscale.

Performance Metrics:

With the bilateral filter, from looking at the SSIM metric, we can see that this is a high performing filter because of how close the values are to 1. As the RMSE *increases* the images are *less similar*, as opposed to the SSIM where *smaller values* indicate *less similarity*. This relationship is clearly seen in the results we have collected. In comparison to the mean filter, this filter clearly outperforms the latter. The highest performing image is landscape in colour conversely, the lowest is landscape in grayscale.

D. Nonlocal means Filter

Comparisons between grayscale and colour images: There is no significant difference between the grayscale and the colour images however the filter performs slightly better in colour on all images.

Performance Metrics:

The highest performing image is landscape in colour conversely, the lowest is MRI. This is the only filter where MRI in grayscale and MRI in colour received the same metrics. Overall, in terms of metrics this filter underperformed in comparison to the bilateral filter.

E. Median Filter

Comparisons between grayscale and colour images: There seems to be no significant difference between the grayscale and colour outputs. All metrics have very little difference between the colour and grayscale counterparts.

Performance Metrics:

Based on the metrics we can see that this filter performed the best out of all the other filters with relatively low RMSE values and SSIM values close to 1. This filter outperformed the bilateral filter on all images except for the landscape image. The highest performing image is MRI in colour conversely, the lowest is landscape in grayscale. Overall, this filter performed the best out of all four filters, having the highest the highest metrics for all pictures with the exception of the landscape picture.

V. Conclusion

In conclusion, in this paper we have tried to compare the performance of the mean, bilateral, nonlocal means, and median filters. To do so, we applied these filters to four images and their grayscale counterpart. After introducing either gaussian or salt and pepper noise to the image, we calculated the following metrics: root mean squared error (RMSE), peak signal to noise ratio (PNSR), and the structural similarity index (SSIM). We then conducted a visual comparison analysis and then took a look at the metrics to the differences in performance between the type of image (grayscale or colour), the genre of image (human, animal, medical imaging, and scenery), and finally the types of filters. From our results we were able to conclude that overall the landscape is always the most difficult to denoise probably because of how busy an image it is with all the colour and the small details. The highest performing image is usually the MRI scan image; this is most likely due to the fact that this image is not very detailed or busy. The best performances were seen with the median filter with the exception of the landscape image which was better denoised by the bilateral filter. The lowest performing filter was the mean filter.

References

[1] Sylvain Paris; Pierre Kornprobst; Jack Tumblin; Frédo Durand, Bilateral Filtering: Theory and Applications, 2009, pp. 1-73, doi: 10.1561/0600000020

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998, pp. 839-846, doi: 10.1109/ICCV.1998.710815.

[3] Lukac, R. (2011) *Computational photography : methods and applications*. Boca Raton, FL: CRC Press (Digital imaging and computer vision series). doi: 10.1201/b10284.

[4] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of highdynamic-range images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 257–266, Proceedings of the ACM SIGGRAPH conference, 2002, *DOI:https://doi.org/10.1145/566654.566574*

[5] B. K., Shreyamsha Kumar. (2013). Image denoising based on non-local means filter and its method noise thresholding. Signal, Image and Video Processing. 7. 1211-1227. 10.1007/s11760-012-0389-y.

[6] A. Buades, B. Coll, and J. M. Morel, “A Review of Image Denoising Algorithms, with a New One,” Multiscale Model. Simul., vol. 4, no. 2, pp. 490–530, Jan. 2005, doi: 10.1137/040616024.

[7] “Measures of image quality.” https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node18.html (accessed Apr. 20, 2021).

[8] I. D. with OpenCV and P.-P. says, “How-To: Python Compare Two Images,” PyImageSearch, Sep. 15, 2014. https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/ (accessed Apr. 20, 2021).