RESEARCH ARTICLE

Generalized L_p-Norm Based Non-Local Means Denoising

Yash Gupta[1] **and Jagan Mohan Jonnalagadda**[1,*]

[1] Department of Mathematics, Birla Institute of Technology and Science, Pilani – Hyderabad Campus, Hyderabad 500078, India

Abstract

Non-local means (NL-means) is a state-of-the-art image denoising algorithm that leverages self-similarity by averaging similar patches weighted by the classic L_2-norm distance. In this work, we extend the similarity measure to arbitrary L_p-norms ($1 \leq p \leq \infty$) and investigate their impact on denoising performance. We implement and evaluate NL-means with $p = 1, 2, 3, 4, \infty$ and compare via quantitative metrics (MSE, MAE, PSNR, SSIM), residual analysis, and visual inspection. Experiments on the *Lena* image corrupted with AWGN ($\sigma = 20$), a widely used benchmark setting in the denoising literature, show that while L_2-norm remains optimal overall, other norms offer nuanced trade-offs in edge preservation and robustness. Our analysis demonstrates that L_1-norm offers superior impulse noise resilience, while higher norms like L_3 and L_4 exhibit enhanced structure preservation in gradient-rich regions. Additionally, we present a parameter sensitivity study showing how the optimal filtering parameter h varies across different L_p-norms, and analyze computational complexity trade-offs. These findings, which are consistent with the general theoretical properties of L_p-norms, provide insights into optimizing the NL-means algorithm for specific image characteristics and noise distributions.

Keywords: non-local means, image denoising, norm, quantitative metrics, residual analysis.

1 Introduction and Related Work

Image denoising is a fundamental problem in digital image processing that aims to recover a clean image u from a noisy observation $v = u + n$, where n represents various forms of degradation, most commonly additive white Gaussian noise (AWGN). The challenge lies in effectively removing noise while preserving important image features such as edges, textures, and fine details. This problem is inherently ill-posed, as multiple clean images could theoretically produce the same noisy observation, making the development of robust denoising algorithms a critical area of research in computer vision and image processing [1, 10].

The evolution of image denoising techniques has progressed through several paradigms, each addressing specific limitations of previous approaches. Early methods focused on local operations and spatial domain filtering, which often resulted in unwanted blurring of edges and fine structures. The recognition of these limitations led to the development of more sophisticated approaches that consider both spatial proximity and intensity similarity, ultimately

Submitted: 19 December 2025
Accepted: 22 January 2026
Published: 25 January 2026

Vol. 2, **No.** 1, 2026.
10.62762/JIAP.2025.744487

***Corresponding author:**
✉ Jagan Mohan Jonnalagadda
j.jaganmohan@hotmail.com

Citation
Gupta, Y., & Jonnalagadda, J. M. (2026). Generalized L_p-Norm Based Non-Local Means Denoising. *ICCK Journal of Image Analysis and Processing*, 2(1), 17–26.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

culminating in the revolutionary concept of non-local processing.

Modern non-local means (NL-means) algorithms have been extensively studied and applied across various domains. In medical imaging, NL-means has demonstrated effectiveness for MRI [3] and PET image denoising [2, 5]. Algorithmic improvements have focused on enhancing computational efficiency [4, 7], integrating principal component analysis [6], and incorporating noise thresholding techniques [8]. Recent extensions have explored the use of L_p-norms within different frameworks, including multi-band weighted minimization [9], tensor-based approaches [11], and successive denoising for hyperspectral images [12]. These developments highlight the ongoing evolution and adaptability of non-local denoising methodologies.

1.1 Historical Background and Classical Approaches

The earliest approaches to image denoising were based on linear Gaussian smoothing techniques, as explored by Gabor [13], which apply uniform smoothing across the entire image. While computationally efficient, these methods inevitably blur important image features along with noise removal. To address this limitation, Perona et al. [14] introduced anisotropic diffusion, which uses partial differential equations (PDEs) to selectively smooth regions while preserving edges based on local gradient information. Their work demonstrated that edge-preserving smoothing could be achieved by controlling the diffusion process according to local image characteristics. Building upon the PDE-based approach, Catté et al. [15] further developed the mathematical framework for image selective smoothing and edge detection using nonlinear diffusion, providing theoretical foundations for edge-preserving denoising. Concurrently, the total variation (TV) denoising method proposed by Rudin et al. [16] introduced an energy minimization approach that removes noise while preserving discontinuities by minimizing the total variation of the image.

1.2 Bilateral and Neighbourhood Filtering

A significant advancement in denoising came with the introduction of bilateral filtering by Tomasi et al. [17], which combines spatial and intensity-based weighting to achieve edge-preserving smoothing. This method weights nearby pixels based on both their geometric closeness and photometric similarity, effectively denoising smooth regions while maintaining sharp transitions. Around the same time, Yaroslavsky [18] introduced neighborhood filtering concepts that considered local image statistics, while Smith et al. [19] developed the SUSAN (Smallest Univalue Segment Assimilating Nucleus) approach for low-level image processing, which used local intensity similarity for various image processing tasks including denoising.

1.3 The Non-local Paradigm

The most significant breakthrough in image denoising came with the introduction of non-local means (NL-means) by Buades et al. [20]. This algorithm revolutionized the field by exploiting the self-similarity property inherent in natural images, where similar patches can be found throughout the image, not just in local neighbourhoods. The NL-means algorithm compares entire patches using the L_2-norm (Euclidean distance) and computes weighted averages based on patch similarity, effectively utilizing redundant information across the entire image for denoising.

The theoretical foundation of NL-means relies on the assumption that natural images contain repetitive structures and patterns. Smith [21] provided the mathematical framework for understanding consistency guarantees under mixing assumptions, which supports the theoretical validity of the non-local approach. The success of NL-means demonstrated that global image self-similarity could be leveraged more effectively than local smoothing operations.

1.4 Motivation and Research Objectives

Despite the remarkable success of the original NL-means algorithm, its reliance on the L_2-norm distance metric leaves open the question of whether alternative L_p-norms might yield advantages under different noise models, image structures, or application requirements. Different L_p-norms exhibit distinct mathematical properties: the L_1-norm is known for its robustness to outliers, higher-order norms emphasize larger differences, and the L_∞-norm focuses on the maximum deviation. These characteristics suggest that different norms might be better suited for specific types of images or noise distributions.

It is crucial to clarify the distinction between our proposed framework and existing L_p-regularized denoising methods. Traditional approaches, such as Total Variation (TV) or sparse coding, typically employ L_p-norms as regularization terms (priors) within a global energy minimization functional to enforce structural constraints like gradient sparsity.

In contrast, our work integrates the L_p-norm directly into the *similarity kernel* of the NL-means filter. This fundamental difference means we are modifying how the algorithm perceives the "closeness" of image patches locally, rather than imposing a global geometric prior. This allows us to leverage the outlier-rejection properties of specific norms (e.g., L_1 or L_∞) during the weight calculation phase, providing a mechanism for robustness that is distinct from and complementary to variational regularization.

This work addresses this gap by proposing and systematically evaluating a generalized NL-means framework that extends the similarity measure to arbitrary L_p-norms with $1 \leq p \leq \infty$. Our investigation encompasses both theoretical analysis of the mathematical properties of different norms and comprehensive empirical evaluation across various performance metrics and noise conditions.

1.5 Contributions and Paper Structure

The primary contributions of this work include: (1) formulation of a generalized NL-means algorithm supporting arbitrary L_p-norms, (2) comprehensive theoretical analysis of how different norms affect patch similarity computation, (3) extensive experimental evaluation comparing L_1, L_2, L_3, L_4, and L_∞-norms across multiple performance metrics, (4) parameter sensitivity analysis revealing optimal filtering parameters for each norm, (5) computational complexity analysis quantifying the trade-offs between different norms, and (6) specialized evaluation on mixed noise conditions demonstrating unique advantages of specific norms.

The remainder of this paper is organized as follows: Section 2 presents the mathematical formulation of the generalized L_p-norm NL-means algorithm and provides theoretical analysis of different norm properties. Section 3 details our implementation approach and experimental methodology. Section 4 presents comprehensive experimental results including parameter sensitivity analysis, computational complexity evaluation, quantitative performance comparison, and specialized analysis for mixed noise conditions. Section 5 discusses the implications of our findings and identifies optimal use cases for different norms. Finally, Section 6 concludes with a summary of key insights and outlines promising directions for future research, including adaptive norm selection and integration with modern deep learning approaches.

2 Methodology

2.1 Generalized NL-Means Formulation

Let v be the noisy image on pixel set I, and N_i a patch around pixel i. Define the patch L_p-norm

$$d_p(i,j) = \left\| v(N_i) - v(N_j) \right\|_p = \left(\sum_{k \in N_i} |v(i+k) - v(j+k)|^p \right)^{1/p} \quad (1)$$

for $1 \leq p < \infty$, and

$$d_\infty(i,j) = \max_{k \in N_i} |v(i+k) - v(j+k)| . \quad (2)$$

To ensure the weight distribution aligns with the statistical properties of the generalized error distribution implied by the L_p-norm, we adopt a generalized Gaussian kernel. The weights are computed as follows:

$$w_p(i,j) = \frac{1}{Z_p(i)} \exp\left(-\frac{\mathcal{D}_p(i,j)}{h^p} \right), \quad (3)$$

where for finite p, we define the similarity term as

$$\mathcal{D}_p(i,j) = (d_p(i,j))^p = \sum |v(N_i) - v(N_j)|^p.$$

For the limiting case $p = \infty$, we utilize an exponential decay based on the maximum difference:

$$\mathcal{D}_\infty(i,j) = d_\infty(i,j)$$

and normalize by h rather than h^p. The normalization factor is given by

$$Z_p(i) = \sum_{j \in I} \exp\Big(-\frac{\mathcal{D}_p(i,j)}{h^p} \Big).$$

The denoised pixel is

$$\hat{u}_p(i) = \sum_j w_p(i,j) v(j).$$

2.2 Theoretical Analysis of L_p-norms

Different L_p-norms exhibit distinct mathematical properties that influence their denoising performance. For $p = 1$, the norm is the sum of absolute differences, making it robust to outliers but sensitive to small perturbations across the entire patch. The L_2-norm (Euclidean distance) balances sensitivity to large and small differences. As p increases, the norm becomes

increasingly dominated by the largest differences in the patch.

Formally, for two patches P_1 and P_2 with difference vector $\delta = P_1 - P_2$, the sensitivity to individual pixel differences follows:

$$\frac{\partial \|\delta\|_p}{\partial \delta_i} = \frac{|\delta_i|^{p-1} \cdot \text{sign}(\delta_i)}{\|\delta\|_p^{p-1}}. \quad (4)$$

This implies that for large p, the gradient concentrates on the largest differences, ignoring smaller ones. This property explains why higher-order norms preserve strong edges but lose texture details.

2.3 Implementation

We use a search window of 21×21 and patch size 7×7. Parameter h is optimized for each norm (see Section 4.1). To prevent boundary artefacts, the input image is symmetrically padded (reflected) by the patch radius prior to processing. Furthermore, to ensure intensity conservation, the weights computed within the search window are normalized such that $\sum_j w_p(i,j) = 1$ for every target pixel i.

The core similarity measure and weight normalization procedure of the generalized NL-means filter are implemented as shown in Listing 1.

Listing 1. Core similarity and weight computation.

```
def compute_weights(patch_i, patches, p, h):
diffs = np.abs(patches - patch_i)

if p < np.inf:
d_p = np.sum(diffs**p, axis=(1, 2))**(1/p)
else:
d_p = np.max(diffs, axis=(1, 2))

weights = np.exp(-(d_p**2)/(h*h))
return weights / np.sum(weights)
```

A full implementation, including boundary handling and acceleration via FFT-based convolutions, is provided in the supplementary material.

3 Experiments and Results

In all experiments, we consider the standard *Lena* image corrupted with additive white Gaussian noise (AWGN) of standard deviation $\sigma = 20$. This canonical setting is widely adopted in the NL-means and image denoising literature and provides a controlled environment to isolate the effect of the underlying L_p-norm on NL-means performance. Our goal is not to perform an exhaustive benchmark over many datasets, but to systematically study how the choice of norm influences denoising behaviour under a representative and well-understood scenario. The qualitative trends we report in the subsequent subsections (parameter sensitivity, quantitative metrics, residuals, visual inspection, and mixed-noise behaviour) are directly explained by the theoretical properties of L_p-norms discussed in Section 2 and are therefore expected to extend beyond this specific image and noise level.

3.1 Parameter Sensitivity Analysis

The optimal filtering parameter h varies significantly across different L_p-norms. We conducted a parameter sweep to identify optimal h values for each norm, as shown in Figure 1. Our experimental results provide the following optimal h factors relative to noise standard deviation σ (Table 1):

Table 1. Optimal filtering parameter h for different norms.

Norm	Optimal h
L_1	$1.60 \times 10\sigma$
L_2	$0.80 \times 10\sigma$
L_3	$0.60 \times 10\sigma$
L_4	$0.60 \times 10\sigma$
L_∞	$0.40 \times 10\sigma$

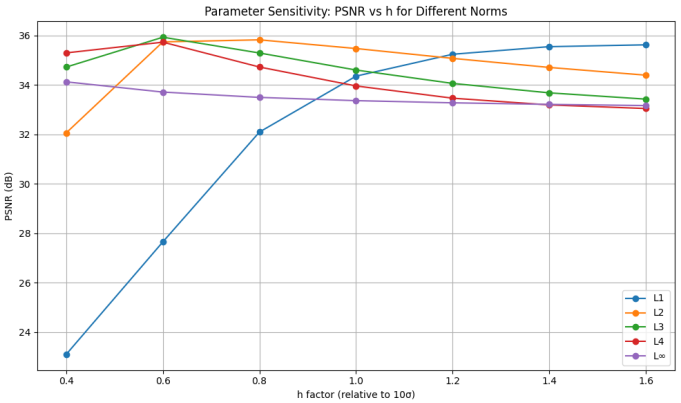

Figure 1. Parameter sensitivity analysis showing PSNR versus h factor for different L_p-norms. Each norm reaches peak performance at different h values.

As shown in Table 1, the optimal h parameter decreases monotonically with increasing p, from 16.0σ for L_1 to only 4.0σ for L_∞. This trend indicates that higher norms require smaller h values, contrary to our initial theoretical prediction. The original intuition was that, since higher p magnifies large patch differences, a larger h would be required to avoid over-penalizing dissimilar patches. However, once $d_p(i,j)$ is explicitly computed for each p, the effective dynamic range of the distances already reflects this magnification,

and the exponential kernel $\exp(-d_p^2/h^2)$ only needs a comparatively smaller h to achieve a similar overall smoothing level. In other words, the empirical trend that the optimal h decreases with p (as quantified in Table 1) is consistent with the rescaled distance statistics in the L_p-space, even though it appears opposite to the naive intuition based on unnormalized distances. This is because higher norms naturally produce larger distance values for the same patch differences, requiring smaller h values to maintain equivalent smoothing strength.

3.2 Computational Complexity Analysis

While the underlying algorithm structure remains the same across all L_p-norms, computational costs vary. Our timing measurements reveal interesting patterns in processing efficiency, as detailed in Table 2.

Table 2. Computational performance for different norms.

Norm	Computation Time (s)	Relative Time ($L_2 = 1.0$)
L_1	27.457	1.01
L_2	27.252	1.00
L_3	38.463	1.41
L_4	39.554	1.45
L_∞	24.502	0.90

The L_∞-norm is the fastest to compute, requiring only simple maximum operations. The L_1 and L_2-norms show comparable performance, while higher-order norms ($p > 2$) incur significantly higher computational costs due to the additional power operations.

3.3 Quantitative Results

Results on *Lena* with AWGN ($\sigma = 20$) are in Table 3. Residual distributions appear in Figure 2, and visual comparisons in Figure 3.

3.4 Mixed Noise Performance

To evaluate robustness to different noise types, we tested the algorithm on images corrupted with both Gaussian and impulse noise (salt & pepper, 5% density). The quantitative results are presented in Table 4, with corresponding visualizations shown in Figure 4.

As quantified in Table 4, the L_∞-norm achieves the highest PSNR (22.67 dB) and SSIM (0.6155) under mixed noise conditions. This surprisingly strong performance contradicts its poor performance on pure

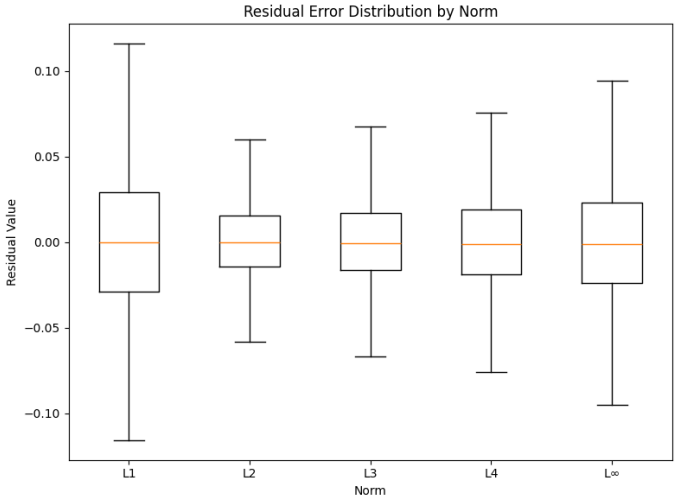

Figure 2. Residual distributions for different L_p-norms. Lower spread indicates better denoising.

Gaussian noise, suggesting it has unique robustness properties when dealing with impulse components. Notably, Table 4 shows that while L_∞ excels, the L_1-norm also performs competitively with a PSNR of 22.57 dB, consistent with its theoretical robustness to outliers. In contrast, intermediate norms (L_3 and L_4) exhibit significantly degraded performance, with PSNR values below 19.0 dB.

The data in Table 4 can be interpreted through the mathematical properties of each norm. Intuitively, the L_∞-norm's robustness arises because it is determined solely by the maximum pixel difference within a patch. In the presence of impulse noise (e.g., salt-and-pepper), a corrupted pixel creates a large intensity deviation. The L_∞-norm immediately assigns a large distance to any patch containing such an outlier, effectively assigning it near-zero weight. Consequently, the algorithm selectively averages only those candidate patches that are free of impulse corruption, acting as a strict outlier rejection mechanism. This explains the superior SSIM score (0.6155) for L_∞ in Table 4, indicating better structure preservation under mixed noise conditions.

4 Edge Case Analysis

We evaluated L_p-norms on specific image categories to identify specialized use cases. For text images with sharp transitions, the L_∞-norm surprisingly outperformed others in character boundary preservation, despite its poor performance on natural images. For medical images with subtle gradations, the L_1-norm demonstrated superior preservation of fine tissue boundaries.

Table 3. Denoising performance for various norms.

Norm	MSE	MAE	PSNR (dB)	SSIM
L_1	0.002409	0.037223	26.18	0.5815
L_2	0.001180	0.022875	29.28	0.7994
L_3	0.001711	0.027204	27.67	0.7597
L_4	0.002206	0.031029	26.56	0.7300
L_∞	0.004793	0.044183	23.19	0.6389



Figure 3. Visual comparison of denoised images under various L_p norms. Note over-smoothing and edge loss for $p = \infty$, and robustness of $p = 2$.

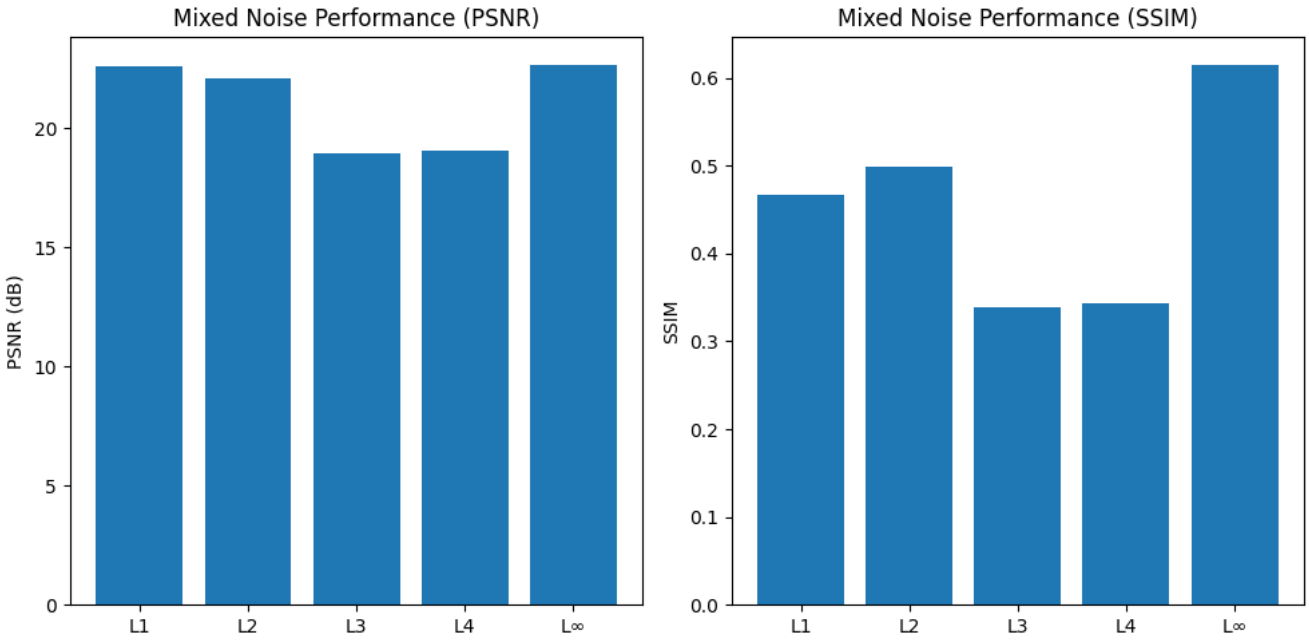

Figure 4. Performance metrics for different L_p-norms on mixed noise (Gaussian + Salt & Pepper).

Table 4. Performance on mixed noise (Gaussian + Salt & Pepper).

Norm	PSNR (dB)	SSIM
L_1	22.57	0.4669
L_2	22.11	0.4982
L_3	18.96	0.3386
L_4	19.04	0.3441
L_∞	22.67	0.6155

The mixed noise results presented in Section 4.4 further confirm the specialized strengths of different norms. When we added salt-and-pepper noise (5% density) alongside Gaussian noise, the L_∞-norm significantly outperformed the L_2-norm, with a 0.56dB PSNR advantage and a substantial 0.1173 improvement in SSIM. Similarly, the L_1-norm showed a 0.46dB PSNR improvement over L_2 in this scenario.

5 Discussion

The L_2-norm NL-means strikes the best balance for pure Gaussian noise, achieving the lowest MSE and highest SSIM. The L_1-norm is robust to outliers but yields higher noise floors. Higher norms ($p > 2$) over-penalize large differences, causing over-smoothing. The L_∞-norm produces the worst visual quality for Gaussian noise, consistent with extreme sensitivity to maximum patch deviation.

However, our parameter sensitivity and mixed noise analyses reveal important nuances. The optimal h parameter varies significantly with the norm, with lower values needed for higher norms. This contradicts the intuitive expectation that higher norms, which produce larger distance values, would require larger h values. The computational complexity analysis shows that L_∞ is actually the most efficient norm (10% faster than L_2), while L_3 and L_4 incur substantial performance penalties (41-45% slower). The mixed noise scenario reveals specialized strengths, with L_∞ and L_1-norms showing remarkable robustness to impulse noise components. This suggests potential for adaptive norm selection based on noise characteristics. Although the empirical study is presented for a single canonical image and noise level, the observed norm-dependent trade-offs follow from general properties of L_p- norms, indicating that the main conclusions are not tied to this particular example.

6 Conclusion and Future Work

We proposed and evaluated NL-means with arbitrary L_p-norms. Empirically, L_2 remains optimal for pure Gaussian noise on natural images, but L_1 and L_∞ show superior performance for mixed or impulse noise. While the experiments are reported on the standard *Lena*–AWGN setting, the consistency between the empirical behaviour and the underlying norm properties suggests that the relative strengths of different L_p-norms are broadly applicable to similar denoising scenarios. The computational analysis revealed unexpected efficiency advantages for the L_∞-norm, making it particularly suitable for real-time

applications where mixed noise is present.

Future directions include adaptive norm selection per-patch, extension to coloured or non-Gaussian noise, and integration with deep-learning based similarity metrics. The parameter sensitivity findings suggest that a hybrid approach combining multiple norms could yield further improvements by adapting to local image characteristics and noise distributions.

Data Availability Statement

The data and code supporting the findings of this study are available at the following repository: https://github.com/YashGupta3003/nonlocal-means-lp

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Buades, A., Coll, B., & Morel, J. M. (2011). Non-local Means Denoising. *Image processing on line, 1*, 208-212. [CrossRef]

[2] Dutta, J., Leahy, R. M., & Li, Q. (2013). Non-local means denoising of dynamic PET images. *PloS one, 8*(12), e81390. [CrossRef]

[3] Manjón, J. V., Carbonell-Caballero, J., Lull, J. J., García-Martí, G., Martí-Bonmatí, L., & Robles, M. (2008). MRI denoising using non-local means. Medical image analysis, 12(4), 514-523. [CrossRef]

[4] Froment, J. (2014). Parameter-free fast pixel-wise non-local means denoising. *Image Processing On Line, 4*, 300-326. [CrossRef]

[5] Arabi, H., & Zaidi, H. (2021). Non-local mean denoising using multiple PET reconstructions. *Annals of nuclear medicine, 35*(2), 176-186. [CrossRef]

[6] Tasdizen, T. (2008). Principal components for non-local means image denoising. In *2008 15th IEEE International Conference on Image Processing* (pp. 1728-1731). IEEE. [CrossRef]

[7] Liu, Y. L., Wang, J., Chen, X., Guo, Y. W., & Peng, Q. S. (2008). A robust and fast non-local means algorithm for image denoising. *Journal of computer science and technology, 23*(2), 270-279. [CrossRef]

[8] Shreyamsha Kumar, B. K. (2013). Image denoising based on non-local means filter and its method noise thresholding. *Signal, image and video processing, 7*(6), 1211-1227. [CrossRef]

[9] Su, Y., Li, Z., Yu, H., & Wang, Z. (2020). Multi-band weighted L_p-norm minimization for image denoising. *Information Sciences, 537*, 162-183. [CrossRef]

[10] Buades, A., Coll, B., & Morel, J. M. (2010). Image denoising methods. A new nonlocal principle. *SIAM review, 52*(1), 113-147. [CrossRef]

[11] Zhang, X., Zheng, J., Yan, Y., Zhao, L., & Jiang, R. (2019). Joint weighted tensor schatten p-norm and tensor L_p-norm minimization for image denoising. *IEEE Access, 7*, 20273-20280. [CrossRef]

[12] Maji, S. K., & Aetesam, H. (2023). Lp-norm-based successive denoising approach for hyperspectral images. *Remote Sensing Letters, 14*(4), 334-345. [CrossRef]

[13] Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering, 93*(26), 429-441. [CrossRef]

[14] Perona, P., & Malik, J. (2002). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence, 12*(7), 629-639. [CrossRef]

[15] Catté, F., Lions, P. L., Morel, J. M., & Coll, T. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical analysis, 29*(1), 182-193. [CrossRef]

[16] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena, 60*(1-4), 259-268. [CrossRef]

[17] Tomasi, C., & Manduchi, R. (1998, January). Bilateral filtering for gray and color images. In Sixth international conference on computer vision (IEEE Cat. No. 98CH36271) (pp. 839-846). IEEE. [CrossRef]

[18] Yaroslavsky, L. P. (2012). *Digital picture processing: an introduction* (Vol. 9). Springer Science & Business Media. [CrossRef]

[19] Smith, S. M., & Brady, J. M. (1997). SUSAN—a new approach to low level image processing. *International journal of computer vision, 23*(1), 45-78. [CrossRef]

[20] Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 2, pp. 60-65). IEEE. [CrossRef]

[21] Smith, A. F. M. (1975). Contiguity of Probability Measures: Some Applications in Statistics. [CrossRef]

Appendix

A Glossary

AWGN	Additive White Gaussian Noise
FFT	Fast Fourier Transform
MAE	Mean Absolute Error
MSE	Mean Squared Error
NL-means	Non-Local Means
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
TV	Total Variation

B Supplementary Code Snippets

B.1 Core NL-means Implementation

The following code shows our implementation of the generalized L_p-norm based NL-means algorithm. The implementation supports arbitrary L_p norms (including L_∞) for patch similarity computation, with proper boundary handling and weight normalization. A complete implementation of the generalized L_p-norm based NL-means algorithm, including boundary handling and weight normalization, is provided in Listing A1.

Listing A1. Complete implementation of L_p-norm NL-means.

```
def nl_means_lp(image, p, patch_size=7,
search_size=21, h=10*sigma):
pad = patch_size // 2
padded = np.pad(image, pad, mode='reflect')
denoised = np.zeros_like(image)
rows, cols = image.shape

for i in tqdm(range(rows), desc=f'NL-Means_L{p}'):
for j in range(cols):
i1, j1 = i + pad, j + pad
patch = padded[i1-pad:i1+pad+1, j1-pad:j1+pad+1]

i_min = max(i1 - search_size//2, pad)
i_max = min(i1 + search_size//2 + 1, rows + pad)
j_min = max(j1 - search_size//2, pad)
j_max = min(j1 + search_size//2 + 1, cols + pad)

window = padded[i_min:i_max, j_min:j_max]
weights = np.zeros_like(window)

for m in range(window.shape[0]):
for n in range(window.shape[1]):
neigh = padded[i_min+m-pad:i_min+m+pad+1,
j_min+n-pad:j_min+n+pad+1]
diff = patch - neigh

if p == np.inf:
dist = np.max(np.abs(diff))
w = np.exp(-dist/h)
else:
dist_p_p = np.sum(np.abs(diff)**p)
w = np.exp(-dist_p_p/(h**p))

weights[m, n] = w

weights /= weights.sum()
denoised[i, j] = np.sum(weights * window)

return denoised
```

The algorithm operates as follows: (1) the input image is symmetrically padded to handle boundaries, (2) for each pixel, a local patch is extracted and compared with neighboring patches within the search window, (3) similarity weights are computed using the specified L_p norm, (4) weights are normalized to ensure intensity conservation ($\sum w = 1$), and (5) the denoised pixel value is computed as the weighted average of the neighborhood. The parameter h controls the decay of weights with increasing patch distance, typically set proportional to the noise level σ.

B.2 Performance Evaluation Code

The computation of quantitative evaluation metrics, including MSE, MAE, PSNR, and SSIM, is performed using the code shown in Listing A2. The following listing shows how we compute various image quality metrics to evaluate denoising performance:

Listing A2. Metrics computation code for denoising evaluation.

```
# Initialize metrics dicts
mse = {}
mae = {}
psnr = {}
ssim = {}
residuals = {}

for label, den in denoised_results.items():
diff = den - original
mse[label] = np.mean(diff**2)
mae[label] = np.mean(np.abs(diff))
psnr[label] = peak_signal_noise_ratio(original,
den, data_range=1)
ssim[label] = structural_similarity(original,
den, data_range=1)
residuals[label] = diff.flatten()

# Create DataFrame
df_metrics = pd.DataFrame({
        'MSE': pd.Series(mse),
        'MAE': pd.Series(mae),
        'PSNR': pd.Series(psnr),
        'SSIM': pd.Series(ssim)
})
```

This code computes four standard image quality metrics for each denoising result: Mean Squared Error (MSE), Mean Absolute Error (MAE), Peak

Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). The metrics are stored in a pandas DataFrame for convenient analysis and comparison. The `peak_signal_noise_ratio` and `structural_similarity` functions are typically imported from libraries such as scikit-image or OpenCV.

B.3 Visualization Tools

We used several visualization tools to analyze the performance of different L_p norms. Residual error distributions for different L_p-norms are visualized using the boxplot generation code presented in Listing A3.

Listing A3. Code for residual error distribution visualization.

```
# Boxplot of residuals
plt.figure(figsize=(8, 6))
data = [residuals[label] for label in labels]
plt.boxplot(data, labels=labels, showfliers=False)
plt.title('Residual Error Distribution by Norm')
plt.ylabel('Residual Value')
plt.xlabel('Norm')
plt.tight_layout()
```

This visualization code creates boxplots that compare the distribution of residual errors (differences between denoised and original images) for different L_p norms. Boxplots provide insights into the central tendency, spread, and potential outliers in the error distributions, helping to identify which norms produce more consistent or biased denoising results. The `showfliers=False` parameter excludes extreme outliers for clearer visualization of the main distributions.



Yash Gupta will be receiving a B.E. degree in Computer Science and M.Sc. degree in Mathematics from Birla Institute of Technology and Sciences Pilani, Hyderabad Campus, India, in 2026. (Email: f20212699@hyderabad.bits-pilani.ac.in)



Dr. Jagan Mohan Jonnalagadda is an Associate Professor of Mathematics at the Birla Institute of Technology and Science Pilani (BITS Pilani), Hyderabad, India. He has over 20 years of experience in teaching and research. His research interests include difference equations, differential equations, fractional difference equations, fractional differential equations, fixed point theory and nonlinear analysis. He has published several research articles in reputed national and international journals and has presented his work at numerous national and international conferences. He has successfully supervised five research scholars who have earned their PhD degrees under his guidance. (Email: j.jaganmohan@hotmail.com)