



Edge-preserving image denoising using noise-enhanced patch-based non-local means

Deepak Dhillon¹ · Rajlaxmi Chouhan¹

Received: 30 June 2022 / Accepted: 29 November 2022 / Published online: 19 December 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Preserving edges while denoising an image is a crucial and challenging necessity. In this paper, a noise-aided edge-preserving denoising algorithm is proposed by extending the classical patch-based Non Local Means (NLM) algorithm. The classical NLM algorithm uses similarity weights to ensure edge preservation. In the proposed algorithm, these similarity weights are enhanced by processing the similarity distances using stochastic resonance (SR). SR refers to a phenomenon where the performance metric in a nonlinear system counterintuitively increases to a peak and then decreases (like a bell curve) in the presence of a controlled amount of noise. The similarity weights derived from NLM are iteratively processed using the discretized SR equation. For the iteratively reconstructed images, the local maxima of the corresponding quality metric, PSNR, is selected as the optimal output. The iterative processing results in a nonlinear scaling of the similarity distances. This processing, in effect, ensures that similarity weights of similar patches are high and those of dissimilar patches are low, thereby producing enhanced edge preservation. The performance of the proposed algorithm is demonstrated by presenting the comparative results for a variety of images corrupted by a wide range of AWGN noise. The proposed algorithm is found to handle the spurious artifacts near the edges more efficiently. Moreover, the obtained edges are sharper and better preserved even in the presence of high noise deviation. Benchmarking results on SET12 and BSD68 datasets show an improvement of 14.5% and 12.1% respectively over that of NLM for high noise.

Keywords Image denoising · Stochastic resonance · Edge · Non-local means · Image filtering

1 Introduction

In the current world, digital multimedia has emerged more prominently than it was a few years ago. However, an image could undergo degradation in the forms of perturbation at the image sensor level, ill-focusing, motion of the imaging device, unfavorable environmental conditions, or limitations of the communication channel, etc. The challenge with denoising is that information is lost in the form of edges at the cost of removing noise. In an effort to remove noise, the edge details become blurred or some new artifacts inadvertently get introduced during the process of denoising. In the purview of this paper, the image-denoising problem is formulated using the equation below:

$$v(i) = u(i) + n(i) \quad (1)$$

where at each 2-dimensional pixel location i , the pristine image u is corrupted by an i.i.d. Additive White Gaussian Noise (AWGN) of zero mean and standard deviation (s.d.) σ (i.e. $n \sim \mathcal{N}(0, \sigma)$), to generate the noisy image v . The purpose of a denoising algorithm is to obtain the best possible estimate \hat{u} of the pristine image u , when provided with the noisy image v .

Buades et al. [1] presented a seminal image denoising algorithm called Non-Local Means (NLM) that denoises an image based on neighborhood similarity and uses pixels that are not even adjacent. NLM has two versions—pixel-based [1] and patch-based [2]. There are various developments on the pixel-based version of NLM, but comparatively, fewer algorithms have been developed on the patch-based version (the proposed algorithm is patch-based). Wang et al. [35] sped up the NLM (pixel-based) by using FFT. Dabov

✉ Deepak Dhillon
deepak.2@iitj.ac.in

¹ Department of Electrical Engineering, Indian Institute of Technology Jodhpur, Jodhpur, Rajasthan 342037, India

et al. [8] stacked similar patches and did shrinkage using the Wiener filter to improve the denoising performance. Khmag et al. [24] used representations in dictionary learning to denoise an image. Zhang et al. [38] used local-pixel-grouping (LPG) with principal-component-analysis (PCA). Khamag [20] used collaborative patches and shrinkage in the SVD domain to improve denoising. Deledalle et al. [9] increased the number of candidate patches in NLM by including arbitrary-shaped patches. Chaudhury and Singer [4] modified the NLM by replacing the mean with the median, based on the view that the median has better robustness against the outliers. Chen et al. [5] increased the number of candidate patches by expanding the sample space to not just the current image but multiple images. Fedorov and Ballester [15] improved the NLM by including the affine transformed patches (scaled, rotated, etc.). Huang et al. [17] sped up the NLM by developing a parallel implementation. DnCNN [36] is an end-to-end residual-learning-based deep network designed to predict residual images for denoising. Liu et al. [25] created a recurrent NN for denoising using the concept of NLM. Recently, Khmag et al. [22] [21] improved the quality as well as the speed of the NLM by first clustering using K-means, followed by NLM in the wavelet domain, and then using a Hidden Markov Model (HMM). Another recent development with pixel-based NLM includes the work by D'Elia et al. [13] that uses bilevel optimization to estimate the parameters in non-local image denoising. Regarding the patch-based NLM, recently, Frosio and Kautz [16] sped up the NLM by allowing only those patches whose euclidean distance with the reference patch is close to the expectation. Shi et al. [30] reconstructed the low-frequency part using SVD and the high-frequency part using sparse representation. Valsesia et al. [34] used graph convolution to learn self-similarities. Khmag [19] developed a wavelet-assisted GAN network that can generate better-denoised images.

Patch-based Non-local means (NLM) [2] algorithm is an elegant way of performing denoising in the spatial domain. NLM deviates from the concept of classical averaging that uses pixels only from the adjacent local neighborhood around the concerned pixel. Instead, NLM uses pixels that are not adjacent and can be present anywhere in a larger area denoted by the research window. The resembling (similar) pixel neighborhoods are searched from the whole research window even if those pixels are *non-local* to the target pixel. As compared to traditional local filters like box filter/gaussian filter/median filter, etc., the NLM has proven to be a better edge-preserving algorithm for AWGN-corrupted images [2]. In this paper, we have proposed an algorithm that improves the factor that play the most crucial role in the NLM algorithm—the similarity weights. The similarity distances (and consequently the similarity weights) are enhanced by iteratively processing

the original similarity distances using a noise-aided iterative process based on the counter-intuitive phenomenon of *Stochastic Resonance* where noise is ‘utilized’ to enhance a non-linear system. The proposed hypothesis is that enhancing the similarity distances using SR-based iterative processing would enhance the edge-preserving capability of the patch-based NLM algorithm [2]. The hypothesis is validated in this paper through both qualitative and quantitative evaluations.

Stochastic Resonance (SR) is a phenomenon observed only in non-linear systems, whereby a feeble signal is amplified with the *assistance* of noise [3]. Noise is not always detrimental for the system and can enhance the effect of a weak signal when present in a controlled and properly-tuned quantities [3]. In the realm of image processing, SR has been successfully utilized in wavelet-based image restoration [6], image enhancement [7], watermark extraction [18], and edge detection [11, 12]. There are mainly two manifestations of SR—one by externally adding noise to the system and controlling the amount that gives the best performance; second is by tuning the system parameters without the addition of external noise and maximizing the performance metric when the process is translated into an iterative equation. This paper uses the second approach where the peak in the performance metric w.r.t. iteration curve is used to determine the optimal output.

Our earlier work [10] was the first instance where the application of SR was reported for improving the edge-preserving nature of pixel-based NLM denoising algorithm [1]. Conventionally, to induce SR in any image processing related application, a specific feature would be iteratively processed. The design of the proposed framework follows this strategy by iteratively processing the NLM-based similarity distances. In the current work, we attempt to further improve the denoising performance of the patch-based NLM using the iterative processing of SR. Although both pixel-based NLM [1] and patch-based NLM [2] share a very similar formulation for finding the resemblance weights, both algorithms follow different toolchains and are different from each other in the following aspects: In the pixel-based approach, weights for a single pixel is estimated, while in the patch-based approach, the weights for the entire patch is estimated. There is no final aggregation step in the pixel-based NLM, which the patch-based NLM, in contrast, claims as the primary cause of its own performance in reducing spurious oscillations around the edges. Further, the proposed patch-based NLM algorithm introduces a tolerance of $2\sigma^2$ in the distance calculation while omitting the need of gaussian weighting of neighborhood. Both algorithms use a different set of values of filtering parameter h for different noise values. With all these differences between pixel-based NLM and patch-based NLM, it does not remain obvious that SR can indeed be used to enhance the similarity weights (the

most crucial step in NLM), and needs experimentation and analysis as presented in this manuscript.

1.1 Key contribution

The primary contribution of this work is the improvement in edge-preserving image denoising performance of the patch-based NLM using SR-based iterative processing. Similarity weights are the most crucial part of the NLM algorithm. The proposed algorithm successfully enhances these similarity weights by inducing stochastic resonance. Whenever SR is established in a non-linear system in the presence of a weak signal and noise, the performance metric always manifests itself in a bell-shaped curve. Since the images in question are noisy images, the similarity weights can be considered a continuum of signal and noise in the image, and are iteratively processed using the discretized Duffing equation. The iteration is terminated at the peak of the metric PSNR, thus resulting in the ‘optimal state’ of the weights.

When tested on the complete BSD68 dataset for AWGN $\sim \mathcal{N}(0, 50)$, a similarity weight ($w \in [0, 1]$), on average, is updated by about 0.0001, but the cumulative effect is reflected as an improvement of 12.1% in the image quality along with sharper edges. As per the authors’ knowledge, the proposed algorithm is the first reporting of improving patch-based NLM with SR, and the second reporting of using SR with any version of the NLM (the first being authors’ work [10] on pixel-based NLM). Regarding image processing, SR has mainly been used for image enhancement, and there are only a few works using SR for denoising.

The results showcase that the proposed algorithm handles the spurious artifacts near the edges much more efficiently, and the edges are sharper and better preserved even at high levels of noise. The establishment of the SR curve with the performance metric validates the authors’ hypothesis along with empirical results and comparisons.

2 Proposed analogy for double well for image denoising application

In the analogy proposed by Chouhan et al. [7] for image enhancement, the double-well represents the *contrast* of the image where one stable state represents a low-contrast image, and the other stable state represents an enhanced image. Here, the transition from low to high contrast is effected by iteratively processing the frequency coefficients or intensity values of the input image. The authors propose a similar analogy for the current application of image denoising (Fig. 1). The ‘state’ of the NLM similarity distance is analogous to the position of the particle in the bistable potential well. As similar patches have smaller similarity distances, the similarity distances represent the weak signal

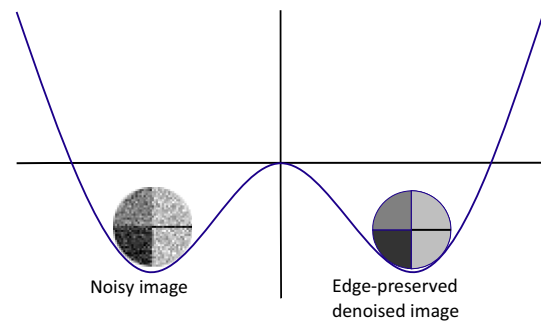


Fig. 1 Proposed double-well analogy of image denoising and SR

here, and the AWGN inherently present in the image plays the role of noise. Since the signal and noise are a continuum, the term *signal + noise* from the Duffing’s equation (with signal present) [29] can be replaced by the *noisy similarity distances* as discussed in Sect. 3.2.

$$\frac{dx(t)}{dt} = -\frac{dU(x)}{dx} + (\text{signal} + \sqrt{D}\xi(t)) \quad (2)$$

The initial state of the patches is represented by one stable state, whereas the other stable state represents the *enhanced* state of the patches. The SR-induced iterations non-linearly scale the similarity distances in accordance with the inherent noise, thus shifting the patches’ state to an ‘enhanced’ state. This goes in line with our hypothesis that enhancing the similarity distances using SR-based iterative processing would enhance the edge-preserving capability of the patch-based NLM algorithm [2]. The hypothesis is validated in this paper through both qualitative and quantitative evaluations. For more details on the double well analogy, readers may refer to Chouhan et al. [7].

3 Proposed framework for SR-based edge-preserving denoising

Figure 2 presents a high-level block diagram of the proposed framework. First, the similarity distances are calculated from the given noisy image using classical NLM. Then, these similarity distances are enhanced using SR-based iterative processing, thus obtaining the ‘enhanced’ similarity weights. These new weights are used to estimate the patches. Pixels from these estimated patches are then aggregated to construct a denoised estimation at pixel locations.

Like in traditional SR-based algorithms, denoising performance (in terms of quantitative metric) is noted for images obtained after each iteration. Of all the iterated images, the one with the best (maximum) performance, corresponding to iteration count, n_{opt} , is chosen as the final denoised output.

Fig. 2 High-level block diagram of the proposed framework

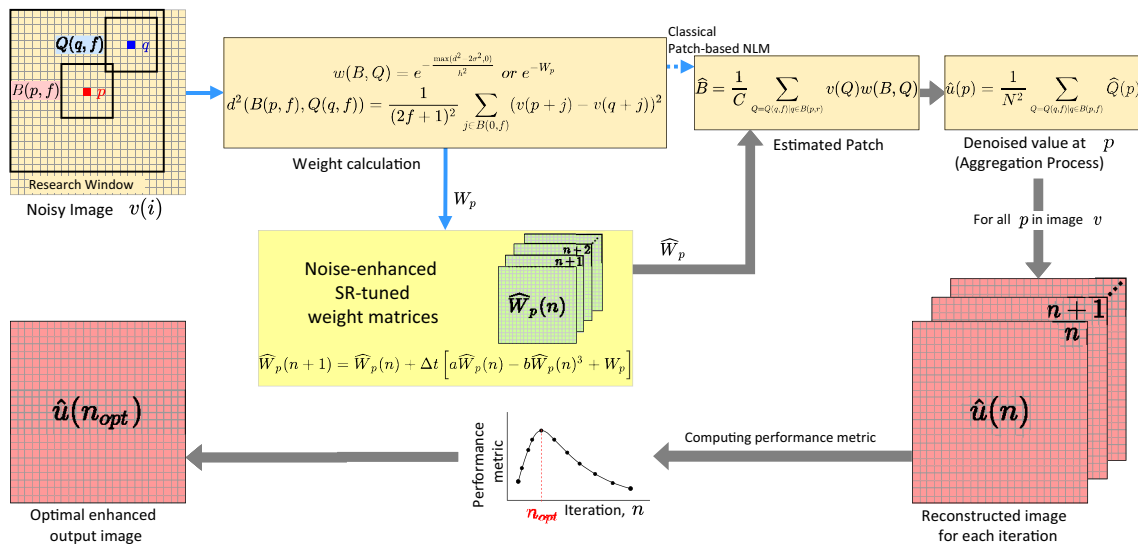
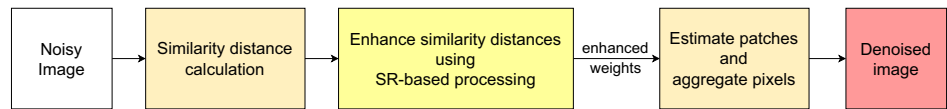


Fig. 3 Proposed algorithm

Enhancing the similarity distances (and thus similarity weights) of the patch-based NLM [2] using the iterative SR-based processing to obtain better denoising with better edge preservation capability, and developing a working algorithm around this construct is the main contribution of the proposed algorithm. As per the authors’ knowledge, the proposed algorithm is the first reporting of improving patch-based NLM with SR, and second reporting of using SR with any version of the NLM (the first being authors’ work [10] on pixel-based NLM).

Figure 3 presents the detailed block diagram, and the following subsections present the proposed algorithm in details. In addition, the pseudocode, including all the parameter values, is presented for a better understanding and reproducibility of the proposed algorithm. The authors have

used Matlab R2020b software to implement the proposed algorithm.

Parameter values of the proposed algorithm are determined using the noise standard deviation (sd). Parameters like patch window size, research window size, and double-well parameters a ($a = 2\sigma_0^2$) and b ($b = 0.01 * (4/27) a^3$), increase with increasing noise sd, whereas the filtering parameter that maneuvers the decreasing exponential kernel decreases with increasing noise sd. These parameters (elaborated in Table 1, pseudocode) and their usage are explained in detail in the subsequent sections. Since the noisy images are generated as phantom images from pristine images, the value of noise sd is used directly (without estimation). Using a preprocessing module to estimate the AWGN noise from the noisy image is part of the future work of the proposed algorithm.

Table 1 Parameters

σ	Patch window	Research window	$h(\sigma)$
$0 < \sigma \leq 15$	3×3	21×21	0.40σ
$15 < \sigma \leq 30$	5×5	21×21	0.40σ
$30 < \sigma \leq 45$	7×7	35×35	0.35σ
$45 < \sigma \leq 75$	9×9	35×35	0.35σ
$75 < \sigma \leq 100$	11×11	35×35	0.30σ

3.1 Initial similarity calculation

To denoise a pixel p in the noisy image $v(i)$ (where both p and i are respectively the 2-dimensional locations of these pixels; and $v(i)$ denotes image intensity at pixel location i); first, all the squared patches that contain p are estimated. Following the classical patch-based NLM, to estimate a low-noise version of a patch $B(p, f)$ —a square neighborhood of pixels centered at p and sized $(2f + 1) \times (2f + 1)$ (as

Pseudocode: SR-NLM Image Denoising

```

SR-NLM(noisyimage, sd, refimage)
if sd > 0 & sd ≤ 15 then
     $f \leftarrow 1, r \leftarrow 10, k \leftarrow 0.4, n_{iter} \leftarrow 30$  ▷ f sets patch window, r sets
    research window, k sets filtering parameter,  $n_{iter}$  sets the SR iterations
else if sd > 15 & sd ≤ 30 then
     $f \leftarrow 2, r \leftarrow 10, k \leftarrow 0.4, n_{iter} \leftarrow 40$ 
else if sd > 30 & sd ≤ 45 then
     $f \leftarrow 3, r \leftarrow 17, k \leftarrow 0.35, n_{iter} \leftarrow 50$ 
else if sd > 45 & sd ≤ 75 then
     $f \leftarrow 4, r \leftarrow 17, k \leftarrow 0.35, n_{iter} \leftarrow 60$ 
else if sd > 75 & sd ≤ 100 then
     $f \leftarrow 5, r \leftarrow 17, k \leftarrow 0.30, n_{iter} \leftarrow 70$ 
end if
 $h \leftarrow k * sd$  ▷ filtering parameter
for each pixel p in noisyimage do
    for each pixel q in research window do
         $B_p \leftarrow (2f + 1) \times (2f + 1)$  patch around pixel p
         $B_q \leftarrow (2f + 1) \times (2f + 1)$  patch around pixel q
         $d^2 \leftarrow \frac{1}{(2f+1)^2} \sum (B_p - B_q)^2$ 
         $W_p \leftarrow \frac{\max(d^2 - 2(sd)^2, 0)}{h^2}$  ▷ similarity distance
         $array\_W_p[] \leftarrow W_p$ 
    end for
▷ SR iterative processing
     $a \leftarrow 2 * (sd/100)^2$ 
     $b \leftarrow 0.01 * (4/27) * a^3$ 
     $\Delta t \leftarrow 0.02$ 
     $V[] \leftarrow 0$ 
    for iter = 1 to  $n_{iter}$  do
         $V[iter + 1] \leftarrow V[iter] + \Delta t [a * V[iter] - b * V[iter]^3 + array\_W_p[]]$ 
         $matrix\_V[] \leftarrow V[iter + 1]$ 
    end for
     $matrix\_V[] \leftarrow \exp\{-matrix\_V[]\}$ 
     $matrix\_w_p[] \leftarrow \text{normalize}\{matrix\_V[]\}$  ▷ similarity weight
     $\widehat{B}_p[] \leftarrow matrix\_w_p[] * B_q$  ▷ patch estimation
    for iter = 1 to  $n_{iter}$  do
         $denoisedimage[p, iter] \leftarrow \text{avg}\{\text{corr. pxl from est. patches}\}$ 
    end for
end for
 $denoisedimage \leftarrow denoisedimage[..., iter] \Big|_{iter=iter(\text{psnr maxima})}$ 
return denoisedimage

```

shown in Figure 3)—all the squared patches $Q(q, f)$ that fall within the *research window* $B(p, r)$ are utilized. The research window $B(p, r)$ is a $(2r + 1) \times (2r + 1)$ square window with the centre at pixel p , as shown in Fig. 3. A research window is used instead of the entire image to reduce the computations involved. For example, for an AWGN of s.d. 25, the patches are 5×5 , and the research window is 21×21 . It should be noted that this example is only for illustration as the proposed algorithm does not include AWGN estimation.

The estimated low-noise version of a patch, $\hat{B}(p, f)$ or \hat{B} , is a weighted linear combination of all the noisy patches $Q(q, f)$ or Q , whose center q falls within the research window $((2r + 1) \times (2r + 1))$, as shown in Equation 3, where C is a weight normalization constant, and $v(Q)$ represents intensity of the noisy image for all the square patches falling within the research window.

$$\hat{B} = \frac{1}{C} \sum_{Q=Q(q,f)|q \in B(p,r)} v(Q)w(B, Q) \quad (3)$$

$$C = \sum_{Q=Q(q,f)|q \in B(p,r)} w(B, Q) \quad (4)$$

Similarity distance (W_p) between the two patches $B(p, f)$ (a square patch with center at pixel p) and $Q(q, f)$ (a square patch with center at pixel q) is a function of (i) Euclidean distance (d) between the two patches, (ii) standard deviation (σ) of the noise, and (iii) the filtering parameter (h), as shown in Eq. 5. The similarity weight $w(B, Q)$ and the similarity distance W_p are monotonically related, though inversely, as shown in Eq. 6. Similarity weight is the decreasing exponential of the similarity distance. Thus, as the similarity *distance* decreases, similarity *weight* increases monotonically.

All the patches whose Euclidean distance is less than $2\sigma^2$ are weighed maximally, the weight being the same as the weight for the patch at p itself. For the remaining patches, weight decreases monotonically with the increasing distance, thus providing an auto cut-off for the larger distances.

$$W_p = \frac{\max(d^2 - 2\sigma^2, 0)}{h^2} \quad (5)$$

$$w(B, Q) = e^{-W_p} \quad (6)$$

$$d^2(B(p, f), Q(q, f)) = \frac{1}{(2f + 1)^2} \sum_{j \in B(0, f)} (v(p + j) - v(q + j))^2 \quad (7)$$

The curve of the decreasing exponential kernel is further maneuvered using a filtering parameter h , which is a function of noise standard deviation σ as shown in Table 1.

3.2 SR-based iterative processing of similarity matrix

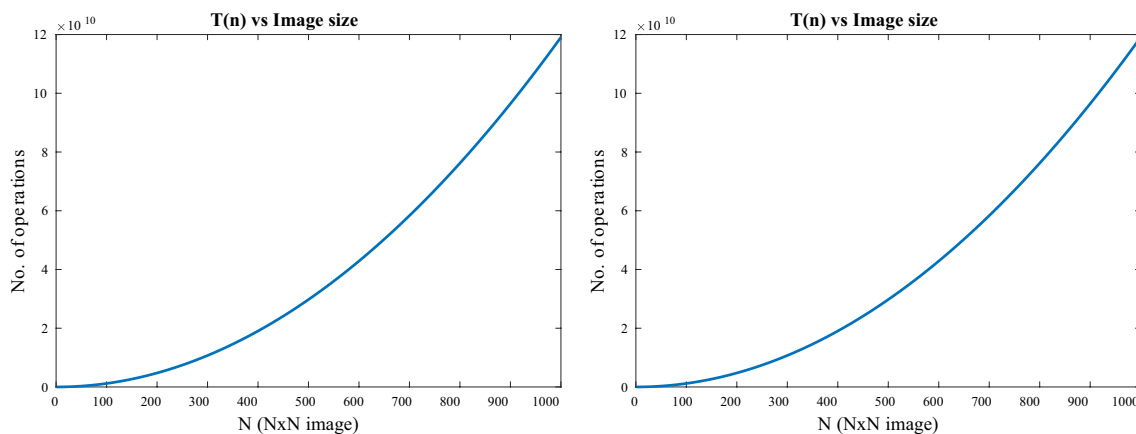
A weight matrix for a patch $B(p, f)$ is a matrix encompassing similarity weights, $w(B, Q)$, for all the patches that lie within the corresponding research window. Weight matrix calculation is the most crucial step of the classical patch-based NLM algorithm. These weights are the attributes that decides the resemblance between two patches to assess which patches are more similar out of the pool from the research window. Higher the similarity between two patches, higher the weight. More the number of similar patches found, more would be the capability to reduce the noise variance (and consequently better denoising). In this work, SR-based iterative processing is used to enhance this very ingredient of the classical NLM algorithm—the similarity weight matrix. As mentioned, the similarity weight ($w(B, Q)$) is simply a function of similarity distance (W_p) and both are monotonically and inversely related. Patched neighborhoods that are more similar have smaller similarity distances and thus higher similarity weights. Considering the Duffing's equation,

$$\frac{dx(t)}{dt} = -\frac{dU(x)}{dx} + \sqrt{D} \xi(t) \quad (8)$$

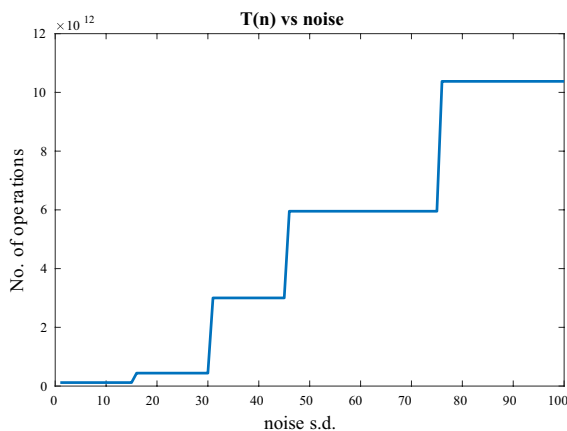
where $x(t)$ denotes the state of a particle in a bistable (or multistable) quartic potential $U(x)$ in presence of an additive stochastic force $\xi(t)$ of intensity D . Based on this, an SR-based discrete model for enhancement of low-contrast images was developed by Chouhan et al. [7], which was further utilized in Dhillon and Chouhan [10] for enhancing the weight matrix of the pixel-based NLM algorithm [1]. The SR-based enhanced similarity matrices generated [10] were:

$$\hat{W}_p(n + 1) = \hat{W}_p(n) + \Delta t \left[a\hat{W}_p(n) - b\hat{W}_p(n)^3 + W_p \right] \quad (9)$$

where W_p denotes the initial similarity distance matrix for a pixel p corresponding to its research window, and $\hat{W}_p(n)$ denotes the SR-tuned similarity distance after n iterations. Parameters $a = 2\sigma_0^2$ and $b = 0.01 * (4/27) a^3$ (should be $< (4/27) a^3$) were deduced by SNR maximization [7], where σ_0 is the normalized noise standard deviation. The discretization step-size, Δt , was chosen experimentally to be 0.02 [10]. This formulation of SR-based weight enhancement [10] has been extended in this paper for patch estimation in a patch-based NLM algorithm. Tuning the similarity distance matrix (using SR) would enhance the similarity value between the patches and help preserve the edges better since similar patches would be assigned higher weights, and dissimilar patches would be assigned the lower weights.



(a) w.r.t. image size



(b) w.r.t. noise s.d.

Fig. 4 Order of growth of running time of SR-NLM

3.3 Final aggregation process

While computing the distance $d^2(B(p, f), B(q, f))$ (Eq. 7) or similarity distance (Eq. 5) or similarity weights (Eq. 6), no special importance is given to different pixels and all the pixels in a patch window are treated equally. This leads to the idea that $d^2(B, Q)$ should be used to denoise *all* the pixels present in patch $B(p, f)$ and not just pixel p . For a single SR iteration number, each pixel p falls within $N^2 = (2f + 1)^2$ patches, and thus can be estimated using the aggregation:

$$\hat{u}(p) = \frac{1}{N^2} \sum_{Q=Q(q,f)|q \in B(p,f)} \hat{Q}(p) \tag{10}$$

where $\hat{Q}(p)$ represents intensity at pixel location p extracted from the estimated low-noise square patch \hat{Q} , and q represents center of such a patch that includes pixel p . This aggregation step defines how the pixel-level estimates are

generated from the patch-level estimates. Using N^2 estimates for a single pixel also adds to the noise-handling capability of the algorithm.

3.4 Finding the optimally-denoised output image

Like in the traditional SR-based algorithms, the output is taken at an optimal iteration count based on the maximization of a performance metric. In the case of denoising, a maximized image quality assessment (IQA) metric would serve as a good choice. In this work, since noisy images are generated as phantom images from pristine images that are available for comparison, the simple *PSNR* metric has been used to decide the optimal iteration count for the output denoised image. Thus, n_{opt} is chosen to be the n at which $PSNR(n)$ is maximum, and $\hat{u}(n_{opt})$ is the final denoised version of the noisy image v .

Fig. 5 NLM vs SR-NLM on BSD68 dataset for AWGN of s.d. 50



3.5 Computational complexity

The asymptotic time complexity of SR-NLM is observed to be the same as that of NLM: $\mathcal{O}_{NLM} = \mathcal{O}_{SR-NLM}$. While enhancing the similarity matrices, only a fixed number of iterations are computed (say 40). The number of iterations are independent of the input image size, and does not grow with the increasing image size (number of pixels). In this paper, various images of different sizes (from different

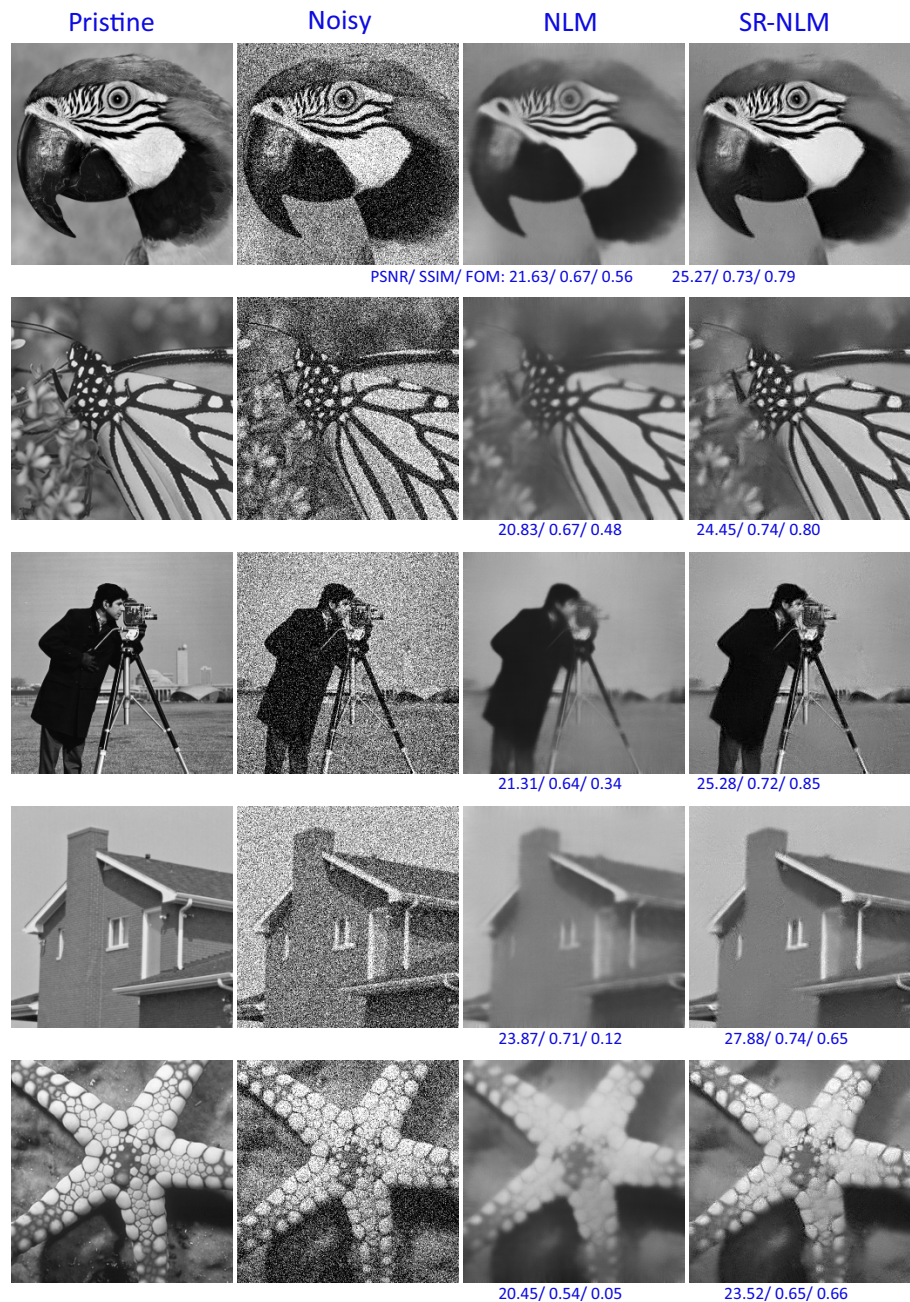
datasets) have been taken, but the number of iterations is fixed and does not change with image size.

Specifically, in terms of the array notation, the running time (rate of growth) $\mathcal{T}(n)$ of the proposed algorithm (SR-NLM) is linear in terms of the number of pixels in the image:

$$\mathcal{T}(n) \sim \mathcal{O}(n_{iter} \cdot R^2 \cdot P^2 \cdot n) \quad (11)$$

$$\mathcal{T}(n) \sim \mathcal{O}(n) \quad (12)$$

Fig. 6 NLM vs SR-NLM on SET12 dataset for AWGN of s.d. 50



where, n_{iter} is the no. of SR iterations, $R \times R$ is the research window size, and $P \times P$ is the patch window size. Often, the notation used in image processing for $N \times N$ sized images is in terms of N , and not in terms of the total number of pixels n ($n = N^2$). Accordingly, the running time of SR-NLM is quadratic in terms of N (for $N \times N$ image):

$$\mathcal{T}(n) \sim \mathcal{O}(n_{iter} \cdot R^2 \cdot P^2 \cdot N^2) \quad (13)$$

$$\mathcal{T}(n) \sim \mathcal{O}(N^2) \quad (14)$$

Since the parameters of the SR-NLM are a function of noise standard deviation (s.d.), the running time also depends upon noise s.d. in a staircase manner. The order of growth of the running time of SR-NLM w.r.t. total no. of pixels, or image height/width N , or noise s.d. is shown in Fig. 4 for better visualization.

The running time of NLM (and thus SR-NLM) is considered a demerit and that is why many researchers have focused solely on improving the speed of NLM, as discussed in the Introduction section.

Table 2 Quantitative comparison for some BSD68 images corrupted with $\sim \mathcal{N}(0, 50)$

Image		NLM	SR-NLM
<i>Jackal</i>	PSNR	22.44	29.00
	SSIM	0.70	0.77
	FOM	0.35	0.52
	BRISQUE	40.53	25.61
<i>Rail bridge</i>	PSNR	20.26	23.63
	SSIM	0.59	0.72
	FOM	0.06	0.72
	BRISQUE	22.50	21.85
<i>Racing cars</i>	PSNR	21.55	24.91
	SSIM	0.55	0.62
	FOM	0.23	0.68
	BRISQUE	24.74	24.94
<i>Cowboy</i>	PSNR	21.16	24.98
	SSIM	0.64	0.70
	FOM	0.23	0.79
	BRISQUE	18.71	17.96
<i>Blessings</i>	PSNR	19.76	24.31
	SSIM	0.62	0.74
	FOM	0.28	0.76
	BRISQUE	31.17	29.76
<i>Cheetah</i>	PSNR	19.32	22.58
	SSIM	0.48	0.64
	FOM	0.14	0.75
	BRISQUE	34.16	26.54
<i>Human face</i>	PSNR	22.67	28.47
	SSIM	0.68	0.83
	FOM	0.65	0.84
	BRISQUE	41.03	15.30

Bold values indicate better values of the respective image quality metrics

4 Results and discussions

The experiments were conducted on grayscale images derived from standard datasets—LIVE [26], TID2013 [33], BSD68 [28] and SET12 [36]. The images were corrupted with a wide range of Additive White Gaussian Noise (AWGN) of zero mean and ten different standard deviations (σ): 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. Since the proposed algorithm is not learning-based, So, all the images work as the testing images (no training data) i.e., the complete dataset of BSD68 [28] and SET12 [36] are used as the testing dataset with no image used for training purposes.

4.1 SR-NLM vs NLM

Figure 5 compares the visual results from the NLM and the proposed algorithm (SR-NLM) on images from BSD68

Table 3 Quantitative comparison for some SET12 images corrupted with $\sim \mathcal{N}(0, 50)$

Image		NLM	SR-NLM
<i>Parrot</i>	PSNR	21.63	25.27
	SSIM	0.67	0.73
	FOM	0.56	0.79
	BRISQUE	18.89	17.81
<i>Butterfly</i>	PSNR	20.83	24.45
	SSIM	0.67	0.74
	FOM	0.48	0.80
	BRISQUE	42.21	30.83
<i>Cameraman</i>	PSNR	21.31	25.28
	SSIM	0.64	0.72
	FOM	0.34	0.85
	BRISQUE	18.76	20.25
<i>House</i>	PSNR	23.87	27.88
	SSIM	0.71	0.74
	FOM	0.12	0.65
	BRISQUE	33.38	27.88
<i>Starfish</i>	PSNR	20.45	23.52
	SSIM	0.54	0.65
	FOM	0.05	0.66
	BRISQUE	33.36	16.96

Bold values indicate better values of the respective image quality metrics

Table 4 Weight change analysis of the proposed algorithm on BSD68 dataset

	sd = 15	sd = 25	sd = 50
Average weight change	0.0009	0.0004	0.0001
Maximum weight change	0.45	0.38	0.24
Research window	21 × 21	21 × 21	35 × 35

corrupted with AWGN of s.d. 50. In Fig. 5, the reader may observe that the sharpness around the animal *Jackal* (first image), or the *Rail Bridge* (second image) is better in SR-NLM denoised images. The word *Castrol* (in third image) can be better read from that of SR-NLM result. For other images as well, it can be observed that the proposed algorithm produces images with better image quality and significantly sharper/ crispier edges. The same is reflected by the *PSNR*, *SSIM*, and *FOM* written below each image. These metrics, along with a no-reference perceptual quality metric, *BRISQUE* [27], are included in Table 2. Better PSNR denotes less mean square error; better SSIM denotes better structural similarity; better FOM [19] denotes better edge retaining/extraction capabilities; and better *BRISQUE* denotes better naturalness in the image. The settings for FOM ($\in [0, 1]$) used throughout this work are *threshold* = 0.1 with the Sobel edge detector (including thinning). For *PSNR*, *SSIM*, and *FOM* metrics, the higher,

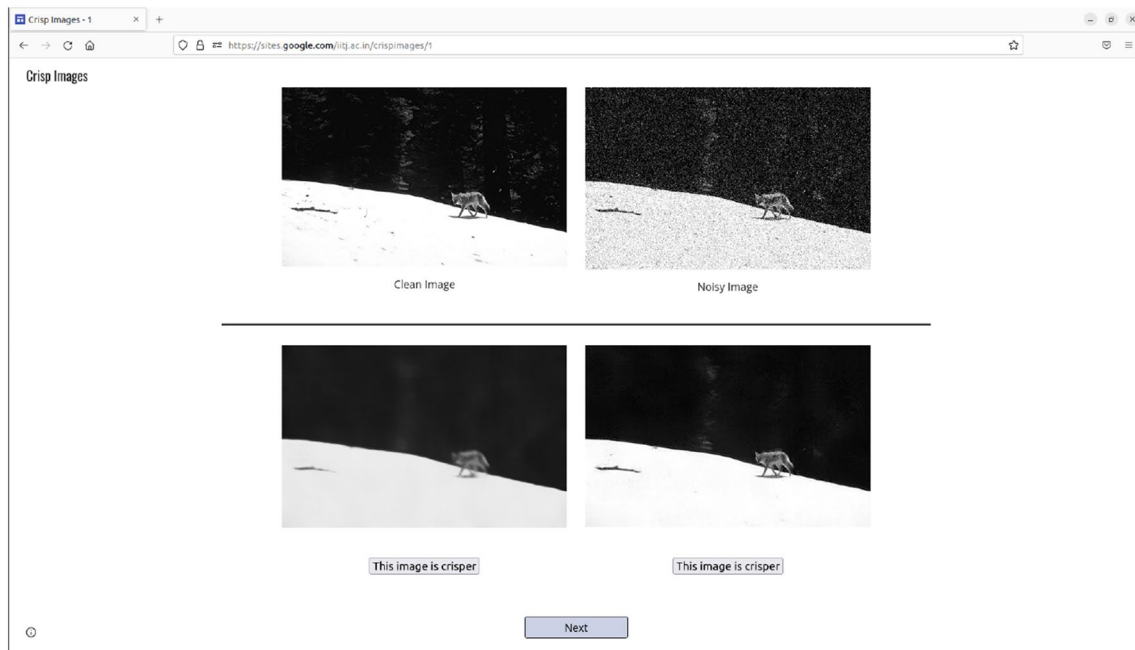


Fig. 7 A screenshot from online visual comparison experiment

Table 5 Quantitative comparison for images corrupted with $\sim \mathcal{N}(0, 50)$

Image		Proposed	NLM [2]	K-SVD [14]	BM3D [8]	LPG-PCA [38]	DnCNN [36]	FFDNet [37]
<i>Human Face</i>	PSNR	28.47	22.67	28.89	29.28	28.60	25.06	25.27
	SSIM	0.83	0.68	0.82	0.85	0.78	0.73	0.74
	FOM	0.84	0.65	0.89	0.89	0.86	0.91	0.90
	BRISQUE	15.30	41.03	32.20	30.42	24.09	31.83	37.75
<i>House</i>	PSNR	27.88	23.87	27.92	29.29	28.35	29.94	30.24
	SSIM	0.74	0.71	0.77	0.80	0.72	0.83	0.83
	FOM	0.65	0.12	0.70	0.79	0.65	0.84	0.84
	BRISQUE	27.88	33.38	41.53	34.36	27.87	32.70	37.48

Bold values indicate better values of the respective image quality metrics.

the better. For *BRISQUE* metric, a lower value denotes a better image quality. Table 2 shows that the proposed algorithm performs better in terms of all the four metrics used.

Similarly, Fig. 6 (with Table 3) compares both the algorithms on images from SET12 corrupted with AWGN of s.d. 50. The reader can observe that the sharpness around the *Parrot's* eye (first image), or the *Butterfly's* veins (second image), or the *Cameraman's* camera stand (third image), are better in the SR-NLM results.

Compared to the NLM, the images from SR-NLM are visually better in terms of visual quality, image quality metrics, and sharpness of edges. The same is confirmed by an improvement of 14.5% and 12.01% in benchmarking results (AWGN s.d. 50) on SET12 and BSD68 datasets, respectively (Table 6).

4.1.1 Subjective analysis

For proper subjective analysis, ten random images from Figs. 5 and 6 were tested for visual comparison by human observers. For an image, the participants were presented with four images—noisy, pristine, NLM denoised, and SR-NLM denoised; and were asked to choose (from the NLM and SR-NLM denoised images) the one which is sharper/crisper and better-looking. A screenshot of the same is shown in Fig. 7. Both textual as well visual instructions were provided in the experiment for a better understanding of the participants. A total of 40 people participated in this study (Age group: 20–36 years, median age: 27 years, 10 females, 30 males). Of the subject pool, 5 were researchers in the area of image processing.

Fig. 8 Illustration: Similarity weights (w_1 and w_2) rearrangement by SR-based processing

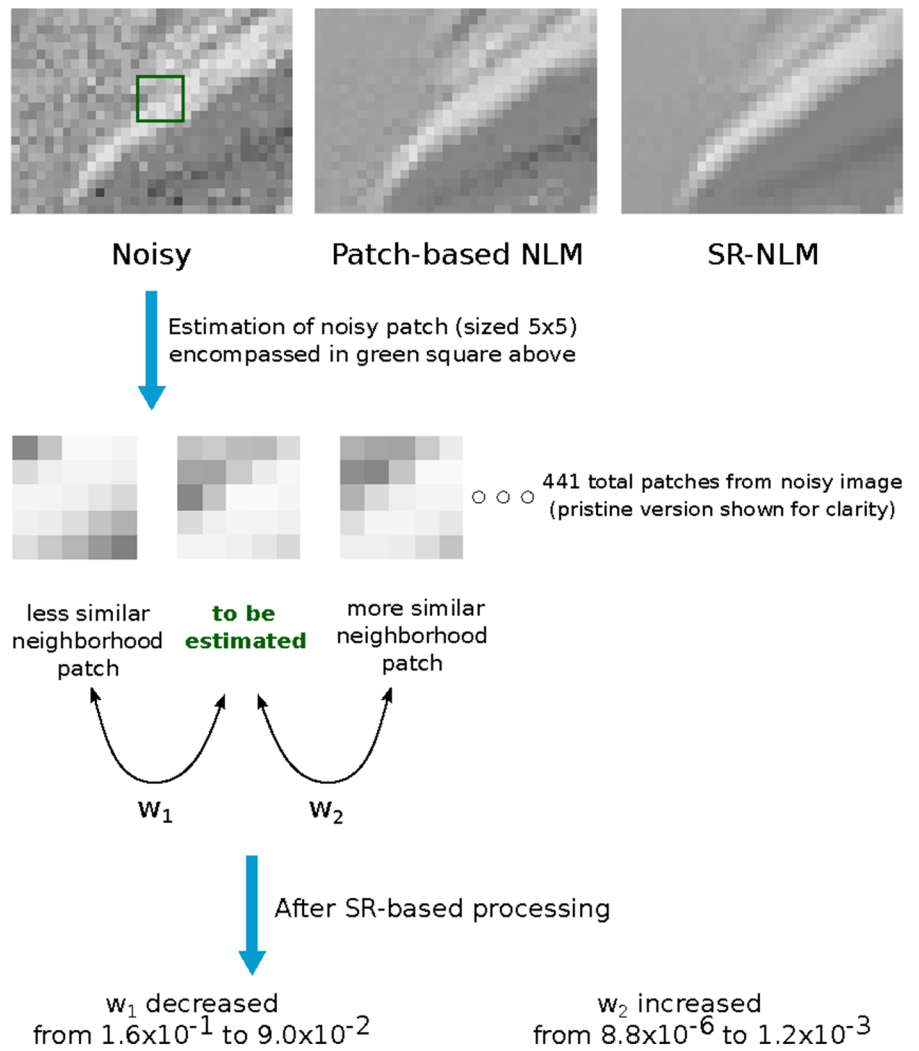


Table 6 Average PSNR (dB) on benchmarking datasets

Dataset	Noise σ	Proposed	NLM [2]	K-SVD [14]	BM3D [8]	DnCNN [36]	FFDNet [37]	BRDNet [32]	ADNet [31]
BSD68	15	30.42	30.13	30.89	31.07	31.72	31.62	31.79	31.74
	25	27.93	27.10	28.30	28.57	29.23	29.19	29.29	29.25
	50	24.72	22.05	25.17	25.62	26.23	26.3	26.36	26.29
SET12	15	31.35	30.95	31.97	32.37	32.86	32.77	33.03	32.98
	25	28.98	28.24	29.42	29.97	30.43	30.44	30.61	30.58
	50	25.38	22.16	25.81	26.72	27.18	27.32	27.45	27.37

From this subjective evaluation, the SR-NLM denoised images were selected as the crispier/sharper and better-looking images for above 99% of the selections.

4.2 How exactly does SR help in denoising here?

The denoised version of a patch is estimated as the linear combination of weighted patches in the research window (Eq. 3). These weights are the decreasing exponential of

the similarity distance (Eq. 6). Stochastic Resonance (SR) enhances exactly these similarity distances, which in turn leads to the similarity weight enhancement and thus better denoising.

An illustration of how the similarity weight rearrangement is done by the SR-based processing is shown in Fig. 8. For this illustration, consider zooming into a small portion of image *Ladyliberty*. In Fig. 8, the top row shows respectively the corresponding area in the noisy input, output of

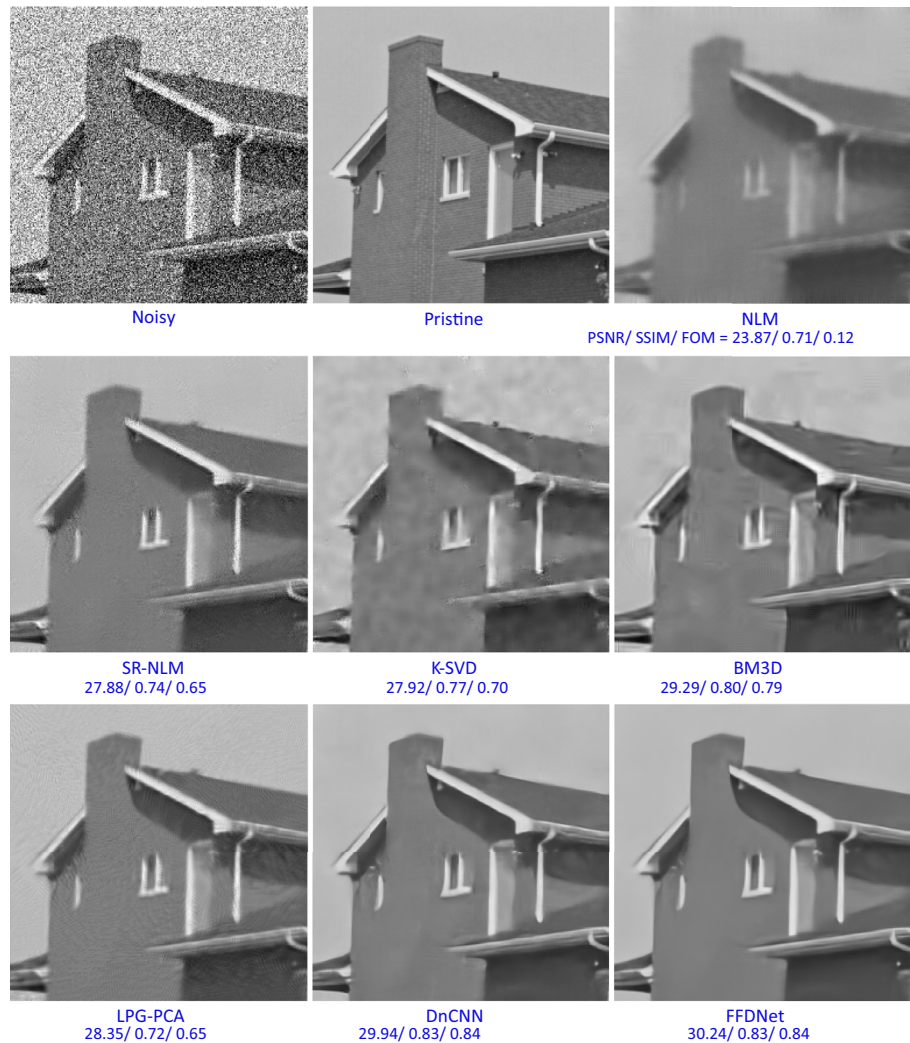
Fig. 9 Comparison of SR-NLM denoised *Human Face* image (s.d. = 50) with existing algorithms



patch-based NLM, and the output of the proposed SR-NLM, in this order. Consider a 5×5 sized noisy patch, encompassed in the green square in the noisy area, which needs to be estimated. The pristine version of this patch is again shown in the next row (second row of Fig. 8) as the second patch from left, henceforth mentioned as the ‘middle patch’. Let us call the patch on the left of the ‘middle patch’ as ‘left patch’, and the one on the right of the ‘middle patch’ as ‘right patch’. To estimate this ‘middle patch’, 441 patches (each 5×5 sized) from the noisy image, that lie within the corresponding research window, are used. The left and the

right patches are two of those 441 patches. For the purpose of clarity, the corresponding pristine version of the patches have been shown instead of the noisy ones; however, in the algorithm, only the patches from the noisy image are actually used in the patch estimation process. As clearly visible, the similarity between the ‘right patch’ and the ‘middle patch’ is more as compared to that between the ‘left patch’ and the ‘middle patch’. Patch-based NLM [2] has weighted these similarities as w_2 (right) and w_1 (left) respectively, as shown in Fig. 8. Here, the primary effect of the SR-based processing is to rearrange these similarity weights. SR-based

Fig. 10 Comparison of SR-NLM denoised *House* image (s.d. = 50) with existing algorithms



processing has increased the weight for the similar patch and decreased the weight for the dissimilar patch. This way, the contribution of the similar patch (right patch) in estimating the required patch (middle patch) has been increased, and the contribution from the dissimilar patch (left patch) has been reduced.

Table 4 presents the weight change analysis on the complete BSD68 dataset. Weight between two patches ranges from zero to one ($w \in [0, 1]$). SR-NLM induces a maximum weight change of about 0.45, 0.38, and 0.24 for AWGN noise s.d. of 15, 25, and 50, respectively. For these noise levels, although the average change in weights is about 0.0009, 0.0004, and 0.0001 respectively, the effects cumulate to an improvement of about 12.1% (for s.d. 50) in the image quality with better-preserved edges. It is also to be noted that the average weight change for noise s.d. 50 is lower than that of s.d. 25. One of the reasons for this change could be the larger research window since the weights are shared between all the patches of a research window (Table 5).

4.3 Comparative characterization

It has been demonstrated that the proposed algorithm performs better denoising than the patch-based NLM. In this section, the focus will be on how the proposed algorithm stands in comparison to various state-of-the-art algorithms. Figures 9 and 10 show the denoising results of the SR-NLM compared with the NLM [2], K-SVD [14], BM3D [8], LPG-PCA [38], DnCNN [36], and FFDNet [37] algorithms for the image *Human Face* and *House*, respectively (corrupted with a high AWGN of s.d. 50). For Figs. 9 and 10, the official codes provided by the respective authors have been used to generate these results. As can be seen in Fig. 9, although K-SVD and BM3D have better *PSNR*, cloudy or stain or splash-like artifacts can be observed in the visual images. Similarly, LPG-PCA also has better *PSNR* but certain lines or stains-like artifacts are observed in the denoised images. DnCNN and FFDNet are DL-based denoising algorithms; although they produce smooth-looking images but on many instances, they result in false stain-like artifact generation,

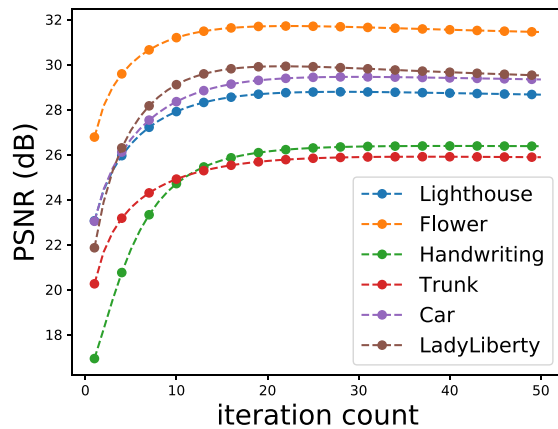


Fig. 11 Performance metric vs iteration count (n) for six test images ($\sigma = 25$), exhibiting a well-defined resonant behaviour in the image quality

like the false nose and lips in Fig. 9. Similarly, in Figure 10, K-SVD and BM3D are observed to produce a stain-like artifact; LPG-PCA produces a line-like artifact; DnCNN also produces some artifacts, but the result of FFDNet is appreciable for this image. Therefore, while the results of DL-based algorithms (DnCNN and FFDNet) are found to significantly vary from image to image, the results of classical algorithms are uniform across the images. Further, in reference to benchmarking, these state-of-the-art algorithms fare better in terms of *PSNR*, *SSIM*, *FOM*, *BRISQUE*, etc.. However, many of them produce new unwanted artifacts during the denoising process, especially at high noise.

These figures are used for illustration purpose while benchmarking results on large datasets are reported in the next subsection.

The curve in Fig. 11, *PSNR vs iteration count* exhibits a well-defined resonant-peak at/around the optimal iteration count (e.g. at $n = 23$ for image *Flower*). This iteration count is an indicator of noise as the inherent noise (in the similarity distance) is scaled non-linearly with each iteration. This resonant nature of the curve validates the authors' initial hypothesis that a *noise-induced resonance* has been established with SR-NLM iterative processing for denoising.

4.3.1 Benchmarking

Table 6 shows the denoising benchmarking results on the widely-known datasets, BSD68 [28] and SET12 [36]. The proposed algorithm is not a learning-based algorithm. The complete dataset of BSD68 and SET12 with 68 and 12 images respectively act as the testing datasets. The proposed algorithm exhibits better results than NLM over the complete range of noise, especially at high noise. Although SR-based similarity weight enhancement has improved the

underlying NLM algorithm, the other state-of-the-art algorithms are found to have higher reported PSNR (as conventionally reported in benchmarking data). Though PSNR is known to not be suitable metric of visual quality and perceptual attributes, in terms of conventional use, an improvement of 14.5% and 12.1% over NLM is observed for the proposed algorithm on SET12 and BSD68 benchmarking, respectively, for the noise deviation of 50. This reinforces our claim that the proposed algorithm performs better at higher noise.

5 Conclusions and future scope

An edge-preserving image-denoising algorithm was proposed in this paper by *utilizing* the presence of noise in an image using iterative processing of weights of the conventional patch-based NLM. Since noisy images were generated as phantom images from pristine images, the simple metric of *PSNR* was used to terminate the SR-based iterative processing. Comparative characterization however was conducted using both full-reference and no-reference image quality metrics. Observations show that the iterative processing nonlinearly enhances these similarity weights, and as a result the proposed algorithm handles the spurious artifacts near the edges significantly better than the classical NLM algorithm. Experiments were conducted on images of different sizes, and corrupted with nine different values of AWGN. Objective comparison with the popular edge-preserving image denoising algorithms—NLM, BM3D, K-SVD, LPG-PCA, DnCNN, FFDNet, BRDNet, ADNet—was reported and the proposed algorithm was found to display noteworthy performance, especially in the presence of large noise intensities without generating spurious artifacts.

Due to the dependence of this algorithm on the induction of SR, the algorithm performs well in a significantly noisy environment but only marginally well in low noise conditions. This counter-intuitive nature of the performance is a limitation of the proposed approach which does not include the addition of external noise in the SR iterative processing. This could possibly be circumvented by exploring the alternate ways of inducing SR (adding controlled amounts of external noise in the iterative steps) for low-noise environments. The proposed algorithm has also inherited a demerit from the NLM—its high computational complexity. Various researchers have worked on improving the running time of NLM, and those works could guide the efforts to reduce the time complexity of SR-NLM as a potential future direction of this work. Learning (training) from large datasets could also be included to improve the performance of the proposed algorithm. Another possible future direction includes using

the automatic estimation of AWGN noise s.d. [23], as a preprocessing module to streamline the complete algorithm.

The characteristic curve shows a resonant peak, thus confirming that *stochastic resonance* has indeed been established. Improved results also validate the proposed hypothesis that enhancing the similarity distances using SR-based iterative processing would enhance the edge-preserving capability of the patch-based NLM algorithm.

Funding Funded by the Department of Science and Technology, Government of India, under the project titled Noise-enhanced Edge-preserving Image Denoising using Stochastic Resonance (#ECR/2016/001606).

Data availability Publicly available data has been used in this work. Image results can be made available on request.

Code availability This work uses a custom code which is not publicly available.

Declarations

Conflict of interest The authors declare that they have no conflict of interest. This work is supported by Department of Science and Technology, Government of India (#ECR/2016/001606).

References

- Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE CVPR, pp 60–65 (2005)
- Buades, A., Coll, B., Morel, J.M.: Non-local means denoising. *Image Process. On Line* **1**, 208–212 (2011)
- Bulsara, A.R.: Tuning in to noise. *Phys. Today* **49**(3), 39–45 (1996)
- Chaudhury, K.N., Singer, A.: Non-local Euclidean medians. *IEEE Signal Process. Lett.* **19**(11), 745–748 (2012)
- Chen, G., Zhang, P., Wu, Y., et al.: Denoising magnetic resonance images using collaborative non-local means. *Neurocomputing* **177**, 215–227 (2016)
- Chouhan, R., Jha, R.K., Biswas, P.K.: (2012) Image denoising using dynamic stochastic resonance in wavelet domain. In: Proc, pp. 58–63. IEEE ISDA, IEEE (2012)
- Chouhan, R., Jha, R.K., Biswas, P.K.: Enhancement of dark and low-contrast images using dynamic stochastic resonance. *IET Image Process.* **7**(2), 174–184 (2013)
- Dabov, K., Foi, A., Katkovnik, V., et al.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Deledalle, C.A., Duval, V., Salmon, J.: Non-local methods with shape-adaptive patches (nlm-sap). *Journal of Mathematical Imaging and Vision* **43**(2), 103–120 (2012)
- Dhillon, D., Chouhan, R.: Noise-aided edge-preserving image denoising using non-local means with stochastic resonance. In: 2018 IEEE GlobalSIP, pp 21–25 (2018)
- Dhillon, D., Chouhan, R.: Enhanced edge detection using sr-guided threshold maneuvering and window mapping: Handling broken edges and noisy structures in canny edges. *IEEE Access* **10**(11), 11191–11205 (2022a)
- Dhillon, D., Chouhan, R.: Exhibition of noise-aided stochastic resonance by discontinuity detectors in smartphone images. *Fluctuat. Noise Lett.* **21**(4), 2250038 (2022)
- D’Elia, M., De Los Reyes, J.C., Miniguano-Trujillo, A.: Bilevel parameter learning for nonlocal image denoising models. *J. Math. Imaging Vision* **63**(6), 753–775 (2021)
- Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing* **15**(12), 3736–3745 (2006)
- Fedorov, V., Ballester, C.: Affine non-local means image denoising. *IEEE Trans. Image Process.* **26**(5), 2137–2148 (2017)
- Frosio, I., Kautz, J.: Statistical nearest neighbors for image denoising. *IEEE Trans. Image Process.* **28**(2), 723–738 (2019)
- Huang, F., Lan, B., Tao, J., et al.: A parallel nonlocal means algorithm for remote sensing image denoising on an intel xeon phi platform. *IEEE Access* **5**, 8559–8567 (2017)
- Jha, R.K., Chouhan, R.: Dynamic stochastic resonance-based grayscale logo extraction in hybrid svd-dct domain. *J. Franklin Inst.* **351**(5), 2938–2965 (2014)
- Khmag, A.: Additive gaussian noise removal based on generative adversarial network model and semi-soft thresholding approach. *Multimedia Tools and Applications* pp 1–21 (2022a)
- Khmag, A.: Digital image noise removal based on collaborative filtering approach and singular value decomposition. *Multimed. Tools Appl.* **81**(12), 16645–16660 (2022)
- Khmag, A., Al Haddad, S.A.R., Ramlee, R.A., et al.: Natural image noise removal using non local means and hidden Markov models in stationary wavelet transform domain. *Multimed. Tools Appl.* **77**(15), 20065–20086 (2018)
- Khmag, A., Al Haddad, S.A.R., Ramlee, R.A., et al.: Natural image noise removal using nonlocal means and hidden Markov models in transform domain. *Vis. Comput.* **34**(12), 1661–1675 (2018)
- Khmag, A., Ramli, A.R., Al-Haddad, S., et al.: Natural image noise level estimation based on local statistics for blind noise reduction. *Vis. Comput.* **34**(4), 575–587 (2018)
- Khmag, A., Ramli, A.R., Kamarudin, N.: Clustering-based natural image denoising using dictionary learning approach in wavelet domain. *Soft Comput.* **23**(17), 8013–8027 (2019)
- Liu, D., Wen, B., Fan, Y., et al.: Non-local recurrent network for image restoration. In: *Advances in Neural Information Processing Systems* 31 (2018)
- LIVE (2005) <http://live.ece.utexas.edu/research/quality>
- Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **21**(12), 4695–4708 (2012)
- Roth, S., Black, M.J.: Fields of experts. *Int. J. Comput. Vis.* **82**(2), 205 (2009)
- Sanjuán, M.A.: *Stochastic Resonance. From Suprathreshold Stochastic Resonance to Stochastic Signal Quantization*, by MD McDonnell, NG Stocks, CEM Pearce and D. Abbott: Scope: monograph. Level: graduate students and researchers. Taylor & Francis (2010)
- Shi, M., Zhang, F., Wang, S., et al.: Detail preserving image denoising with patch-based structure similarity via sparse representation and svd. *Comput. Vis. Image Understanding* **206**(103), 173 (2021)
- Tian, C., Xu, Y., Li, Z., et al.: Attention-guided cnn for image denoising. *Neural Netw.* **124**, 117–129 (2020)
- Tian, C., Xu, Y., Zuo, W.: Image denoising using deep CNN with batch renormalization. *Neural Netw.* **121**, 461–473 (2020)
- TID 2013 (2013) <http://ponomarenko.info/tid2013.htm>
- Valsesia, D., Fracastoro, G., Magli, E.: Deep graph-convolutional image denoising. *IEEE Trans. Image Process.* **29**, 8226–8237 (2020)
- Wang, J., Guo, Y., Ying, Y., et al.: Fast non-local algorithm for image denoising. In: 2006 International Conference on Image Processing, pp 1429–1432 (2006)

36. Zhang, K., Zuo, W., Chen, Y., et al.: Beyond a gaussian denoiser: residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)
37. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.* **27**(9), 4608–4622 (2018)
38. Zhang, L., Dong, W., Zhang, D., et al.: Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recogn.* **43**(4), 1531–1549 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Deepak Dhillon received the bachelor's degree from Panjab University, Chandigarh, in 2011, and the master's degree from Thapar University, Patiala, in 2015. He is currently pursuing the Ph.D. degree in image processing with the Department of Electrical Engineering, IIT Jodhpur, India, under the Visvesvaraya Ph.D. Fellowship. From 2011 to 2012, he worked as a Software Analyst at Newgen Software Technologies Ltd., New Delhi. He has served IEEE Student Branch of IIT Jodhpur as

the Vice-Chair from 2017 to 2020. For two years, he was a member of the Rotaract Club of IIT Jodhpur. He received the IEEE Signal Processing Society Travel Grant for presenting his paper at GlobalsIP 2018, Anaheim, CA, USA.



Rajlaxmi Chouhan is an Assistant Professor in the Department of Electrical Engineering at Indian Institute of Technology Jodhpur. She received her PhD from the Indian Institute of Technology Kharagpur (India) in 2015. Her current research interests include image quality assessment, noise-aided image processing applications, education technology and technology-enhanced learning. Dr. Chouhan received the IEEE Region 10 WIE Student Volunteer Award in 2014, and the National Award for Best M.Tech.

Thesis in Electronics and Electrical Engineering in 2012 by Indian Society of Technical Education, New Delhi. For her teaching contributions at IIT Jodhpur, she received the Teaching Excellence Award in 2019 and the Dr. Vandana Sharma Memorial Award for Teaching Innovation in 2021. Dr. Chouhan is a Senior Member of the IEEE and currently serves on the international Subcommittee of IEEE Women in Signal Processing (2022–24).