Neurocomputing 174 (2016) 698–721

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Optimal sub-band adaptive thresholding based edge preserved satellite image denoising using adaptive differential evolution algorithm

A.K. Bhandari [a,b,*], D. Kumar [a], A. Kumar [a], G.K. Singh [c]

[a] PDPM Indian Institute of Information Technology Design and Manufacturing, Jabalpur 482005, Madhya Pradesh, India
[b] Department of Electronics and Communication Engineering, National Institute of Technology, Patna, Bihar-800005, India
[c] Department of Electrical Engineering, Indian Institute of Technology Roorkee, Uttarakhand 247667, India

ARTICLE INFO

Article history:
Received 21 November 2014
Received in revised form
13 May 2015
Accepted 23 September 2015
Communicated by: Paolo Remagnino
Available online 8 October 2015

Keywords:
Image denoising
DWT
Optimization algorithms
Adaptive thresholding function
Adaptive differential evolution algorithm

ABSTRACT

An image is often corrupted by different kinds of noise during its acquisition and transmission. Conventional denoising methods can suppress the Gaussian noise effectively, but fail to maintain the quality of denoised images and may blur edges in an image. To address these short comings, this paper aims to develop an optimized adaptive thresholding function based framework for edge preserved satellite image denoising using different nature inspired algorithms which is capable of effectively removing the Gaussian noise from images without over smoothing edge details. Image denoising using adaptive thresholding functions selects the suitable threshold values to separate noise from the actual image without affecting the actual features of the image. In this approach, most widely used nature inspired optimization algorithms are exploited for learning the parameters of adaptive thresholding function required for optimum performance. It was found that the proposed adaptive differential evolution algorithm (JADE) algorithm based denoising approach has superior features and give better performance in terms of PSNR, MSE, SSIM and FSIM as compared to other methods.

1. Introduction

Image denoising is a classical image processing problem. Digital images are often distorted by the variety of noise. The term noise in digital image processing is referred to any quantity that deflects an observed pixel from its original value. The numerous kinds of noise and artifact in imaging modalities corrupt the images and reduce the image quality. Such artifacts have considerable impact on the image appearance and affect the human interpretation as well as accuracy of the computer assisted methods in various image processing applications [1]. Moreover, image enhancement, image segmentation, image classification and quantitative measurement becomes complicated and unpredictable because of mixture of noise parameters. It is clear that the removal of noise from the image facilitates the processing. Thus, denoising of images has become the fundamental step in several practical applications such as satellite imaging [2]. As a result, in order to minimize the effect of noise and improve the image quality for higher level processing, denoising pre-treatment of image signal is carried out.

In practice, most frequent distortions are due to corruption by additive noise (Gaussian), salt and pepper noise and multiplicative noise (speckle) with different characteristics. In Gaussian noise, each pixel of image is changed by a small amount from its original value. The Gaussian (normal) distribution is a very good model to represent this type of noise. This is due to central limit theorem, which states that the sum of different noises tends to approach a Gaussian distribution. Conventional linear filters remove the Gaussian noises with detriments for edge and texture details in an image. To address the problem of edge-preserving, a variety of modified Gaussian noise removal techniques has been presented [3–7]. Multiplicative noise is generally more difficult to be removed from the images than additive noise because the intensity of noise varies with the signal intensity. In salt and pepper noise, pixels in the image are very different in color or intensity unlike their surrounding pixels. A noisy pixel does not have any relation to the color of neighboring pixels. This type of artifact is caused by sharp and sudden disturbance in the image signal. This noise affects only a small number of pixels. It contains dark and white dots [8]. Remote sensing images are an essential source of information, which are used in several environmental assessment and monitoring such as climate studies, assessment of forest resources, examining marine environments agriculture, metrology, mapping, military, etc [9–11].

[*] Corresponding author.
E-mail addresses: bhandari.iiitj@gmail.com (A.K. Bhandari), deepak.n.sonukumar@gmail.com (D. Kumar), anilkdee@gmail.com (A. Kumar), gksngfee@gmail.com (G.K. Singh).

http://dx.doi.org/10.1016/j.neucom.2015.09.079
0925-2312/© 2015 Elsevier B.V. All rights reserved.

In denoising, traditional spatial filters not only smooth the data and reduce the noise, but also blur the data to some extent. The main aim of denoising is to eliminate the noise particles and to retain the actual image features as much as possible. At present, many new denoising techniques have been developed and explored, such as wavelet based approach [12–14], non-local means algorithm [15–16], Bivariate Shrinkage function [17], Sparse coding shrinkage [18], Bayesian approach [19–21], principal component analysis [22], support vector machine [23], Support vector regression [24–25], compressive sensing theory [26], Bilateral filtering [27–28], Wavelet shrinkage [29] soft thresholding and hard thresholding.

These techniques can perform image smoothing/denoising and preserve the useful features and edges to a certain extent. However, each of these methods has certain limitations in terms of image quality or computational efficiency. In general, non-local means [15–16] obtains the denoised image with fine image quality but it takes relatively high computational cost in the global search for pixels with similar intensity. On the other hand, wavelet thresholding based methods and basis pursuit denoising schemes can effectively suppress the noise because of having sparse representation in most natural images when they are expressed in wavelets or a set of basis. However, these techniques are likely to be affected by the ripple artifacts. Image denoising using bilateral filter [27–28] produces fine results with advantage of easy implementation, but it has not yet attained a desirable level of applicability in terms of image quality [26]. In 2014, Zhang et al. have proposed an adaptive bilateral filter [30] based framework for image denoising which is capable of eliminating the universal noise efficiently, i.e. impulses, Gaussian noise or mixture of the two types of noises, from the images without over smoothing edge details.

Images are often affected by another type of noise known as impulse noise, which replaces the value of certain region of pixel with random value. This type of noise generally arises because of transmission error. To deal with such kind of problems, median filters are designed which can remove the impulse noise to a certain extent, with some of its enhanced performance and better feature preserving rate [31–32]. In 2005, Chan et al. [33] proposed a two-phase scheme for removing salt-and-pepper impulse noise. In the first phase, an adaptive median filter is used to identify pixels, which are likely to be contaminated by the noise. In the second phase, the image is restored using a specialized regularization method that applies only to those selected noise candidates. In case of impulse noise, median filter is found to be most effective nonlinear filter due to its powerful denoising capability and fast processing. However, when the noise level is over 50%, some details and edges are degraded by the filter. In practice, during transmission and acquisition, mixed noise together with the Gaussian and impulse noise arise simultaneously. To remove such kind of noise from noisy image, various efforts have been made [34–36].

In recent years, wavelet thresholding algorithm is found to be one of the most favorable approaches for image denoising. Many filtering techniques have been designed to get better denoised image such as averaging filter, median filter, Wiener filter, adaptive filter, etc. In these classical methods, median filter is most frequently used in nonlinear spatial filters to suppress salt and pepper noise due to its effective denoising performance. But this filtering approach does not give satisfactory results in case of Gaussian noise because it generates a blurred and smoothed image with poor feature localization and incomplete noise suppression. These artifacts take place in denoised image because median filter replaces the noisy pixel by a median value in their vicinity without taking into account the local features such as the presence of edges. In past few years, a number of efforts have been made to remove the speckle noise using wavelet transform [37–39]. Recently, some new approach has been presented using wavelet transform which works in transformed version of the noisy image and obtain the denoised image in transformed domain [40].

In literature, a wide range of wavelet thresholding approaches have been presented. Denoising in the wavelet domain may be stated as *thresholding* of DWT detailed coefficients of the noisy image, either *hard* or *soft* [41]. Hard or soft thresholding of DWT coefficients is commonly used to achieve denoising. In hard thresholding, image is preserved if it is higher than the threshold value; otherwise it is set to zero, and in soft thresholding, image is shrunk to zero by an amount of threshold. Because of having the properties like sparsity and multiscale decomposition in wavelet transform coefficients; it has attracted a number of researchers to work in the wavelet domain. These features of the wavelet domain provide flexibility to represent main energy of signal by few large coefficients and remaining energy by many small coefficients. Since, most of the noise powers are present in many small coefficients; therefore, it is essential to modify these coefficients by a certain rule to achieve the denoised image. To improve this process of denoising, researchers have tried to develop a number of advanced thresholding function.

Nasri and Pour [42] have introduced a new adaptive thresholding function based on wavelet transform based thresholding neural network (WT-TNN) methodology. The proposed function is further used in a new subband-adaptive thresholding neural network to improve the efficiency of denoising procedure. They have reported that the suggested technique outperforms the many well known other thresholding techniques such as soft, hard, garrote and other existing thresholding functions in WT-TNN methodology. Further, they claimed that the presented scheme eliminates the noise regardless of its distribution and modeling of the distribution of image wavelet coefficients. The beauty of this approach using the new adaptive thresholding function is due to simultaneous learning of parameters of thresholding function and threshold value in each sub-band of WT. But the drawback of this technique is more computational cost. Due to the usage of a steepest descent gradient technique in WT-TNN approach, computational time is increased considerably. Basically in this approach, proper initialization of threshold and thresholding parameters is very difficult to achieve fast convergence of the learning process in order to obtain the optimum values of these parameters. Therefore, to overcome the prime limitation of WT-TNN based denoising methodology, an optimized adaptive thresholding function based framework for satellite image denoising using JADE and other optimization techniques have been presented in this paper. The effective denoised results of the proposed JADE algorithms based adaptive thresholding approach is achieved due to the formulation and implementation of optimization algorithms instead of a steepest descent gradient-based LMS technique, which corresponds to superior performance with fast convergence. But more importantly, it has been examined that the proposed optimized adaptive thresholding based denoised image yields better edge preserving performance at high noise levels. After successful performance of the new adaptive thresholding function, in 2011, Bhutada et al. presented an edge preserved image enhancement technique using adaptive fusion of images which is denoised by wavelet and curvelet transform [43]. Furthermore, in 2011, wavelet transform-based thresholding neural network (WT-TNN) methodology has been proposed by Bhutada et al. to improve the computational cost of denoising problem. In this approach, they have adaptively selected the learning step size for tuning the parameter of thresholding function [44]. Subsequently, in 2012, authors have presented the PSO-based learning of sub-band adaptive thresholding function for denoising of Gaussian noise [45]. In these papers, the existing approaches do not obtain the thresholding value and the thresholding function simultaneously. None of the presented works considered remarkably for edge preserving for the Gaussian noise. Many of the existing techniques need to adopt few priori assumptions regarding the statistical distribution of subbands of noise-free wavelet coefficients. Due to that, proposing a scheme to reduce the noise regardless of its distribution and modeling of the distribution of the image wavelet coefficients will be very valuable.

From the above discussion, it is clear that the wavelet transform yields better denoising, particularly in homogeneous region, and adaptive thresholding function based on wavelet transform gives effective results. However, it was found that the existing literature does not address the following aspects simultaneously: (i) the existing techniques do not obtain the thresholding value and the thresholding function simultaneously, (ii) None of the existing approaches deals with the Gaussian noise effectively with edge preserving quality, and (iii) the most important feature of the proposed methodology is fast computational time to determine the denoised results using optimized adaptive thresholding function as compared to a WT-TNN method. On the other hand, the optimized adaptive thresholding function based proposed approach overcomes many classical methods and yields superior results in Gaussian noise reduction cases. In addition, to enhance the ability of function, three shape tuning parameters have been introduced which correspond to a comprehensive-thresholding function that can be adjusted to any desired thresholding function. This function is utilized in an adaptive way that is motivated by the Zhang's Thresholding Neural Network (TNN) schemes [46–47]. In this paper, various optimization algorithms such as differential evolution (DE), particle swarm optimization (PSO), wind driven optimization (WDO), firefly algorithm (FA), cuckoo search algorithm (CS) and adaptive differential evolution algorithm (JADE) are used for learning the parameters of adaptive thresholding function required for optimum performance.

The rest of the paper is organized as follows. Section 2 gives a brief introduction of some threshold estimation techniques with formulation of Gaussian noise. Section 3 presents the problem formulation of adaptive thresholding function. In Section 4, adaptive differential evolution algorithm and other algorithm based optimized adaptive thresholding function has been presented for satellite image denoising. Section 5 gives brief formulation of the assessment parameters for comparison purpose. Section 6 reports the visual, qualitative and quantitative results of the proposed and existing methodology supported by peak-signal-to-noise-ratio (PSNR), mean square error (MSE), structural similarity index measure (SSIM) and feature similarity index measure (FSIM) with a brief explanation of results. Finally, conclusions are drawn in Section 7.

2. Wavelet-based thresholding methodology

The generalized and simple way of denoising of an image using wavelet thresholding (WT) is as follows: At first, WT of a noisy image is obtained. Then, wavelet coefficient is modified by the suitable thresholding function. Finally, after modification of wavelet coefficient, inverse WT is applied to obtain the reconstructed image. In a wavelet-based thresholding technique, thresholding function has major effect on the quality of image. Therefore, many types of thresholding functions were introduced having the property of hard, soft, semi-soft and garrote. The garrote and semi-soft thresholding functions are an improved way of thresholding. These thresholding functions have property and advantage of hard and soft thresholding both.

In this section, the fundamentals of noise suppression in wavelet domain are given for both types of Gaussian noise.

2.1. Gaussian noise

Assume that the observed data vector $x = [x_0, x_1, \ldots, x_{N-1}]^T$ which is corrupted by additive Gaussian noise is given by:

$$x_i = v_i + n_i \qquad i = 0, 1, 2, \ldots, N-1 \tag{1}$$

where, v_i represents the noise-free wavelet coefficient and n_i stands for the element of (independent and identically distributed) *i.i.d* Gaussian noise.

Now, consider that the noise-free data vector $V = [v_0, v_1, \ldots, v_{N-1}]^T$ in wavelet domain and $\hat{V} = [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_{N-1}]^T$corresponds to output of thresholding function in wavelet domain. The aim of denoising is to estimate function f with minimum mean square error (MSE), and the MSE risk can be determined by Eq. (2):

$$f_{MSE} = \frac{1}{2}E||\hat{V} - V|| = \frac{1}{2N}\sum_{i=0}^{N-1}((\hat{v}_i - v_i)^2 \tag{2}$$

where, N is the size of sub-band, v_i is the WT coefficients of noise free image, $\hat{v}_i$ is the thresholded WT coefficients of noisy image.

In thresholding based denoising methods, various attempts have been made to formulate better thresholding functions. In Visu Shrink method, the universal threshold is computed based on noise statistics:

$$thr_{universal} = \sigma\sqrt{2\,\ln(n)} \tag{3}$$

Here, σ represents the noise standard deviation that defines the median absolute deviation. A robust median estimator is used from the finest scale wavelet coefficients:

$$\hat{\sigma} = \frac{\text{median}\left[\left|\{x_{i,j} : i,j \in HH_1\}\right|\right]}{0.6745} \tag{4}$$

where, HH_1 is wavelet subband.

After applying the above shrinkage factor to wavelet coefficients, it has been observed that some details in the image are lost and sometimes, the reconstructed image becomes blurred due to its very large and global threshold.

2.2. Hard thresholding

Hard thresholding can be expressed as follows:

$$D(U,\lambda) = \left\{ \begin{array}{ll} U & \text{for all } |D| > \lambda \\ 0 & \text{otherwise} \end{array} \right\} \tag{5}$$

Hard threshold is considered a "keep or kill" process and is more intuitively interesting. It seems to be a natural procedure. Sometimes, pure noise coefficients may pass the hard threshold and appear as annoying "blips" in the output image [40].

2.3. Soft thresholding

Soft thresholding can be expressed as:

$$D(U,\lambda) = \text{sgn}(U) \quad \max(0,\ |U| - \lambda) \tag{6}$$

Here, soft threshold shrink coefficients represent the above threshold in absolute value. The false structures in hard thresholding can be overcome by soft thresholding.

3. Problem formulation of adaptive thresholding function

To improve the effectiveness of thresholding functions; in some papers, numerous class of thresholding functions has been presented with several shape tuning parameters. For example, Zhang functions in Fig. 1 are extension to soft thresholding function

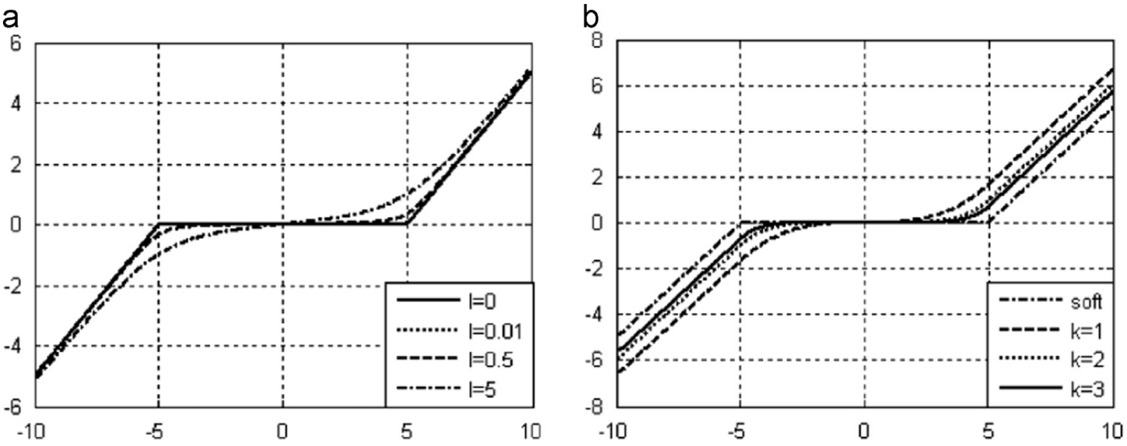

Fig. 1. Zhang thresholding functions with different values of shape tuning factor: (a) Ref. [46] and (b) Ref. [47].

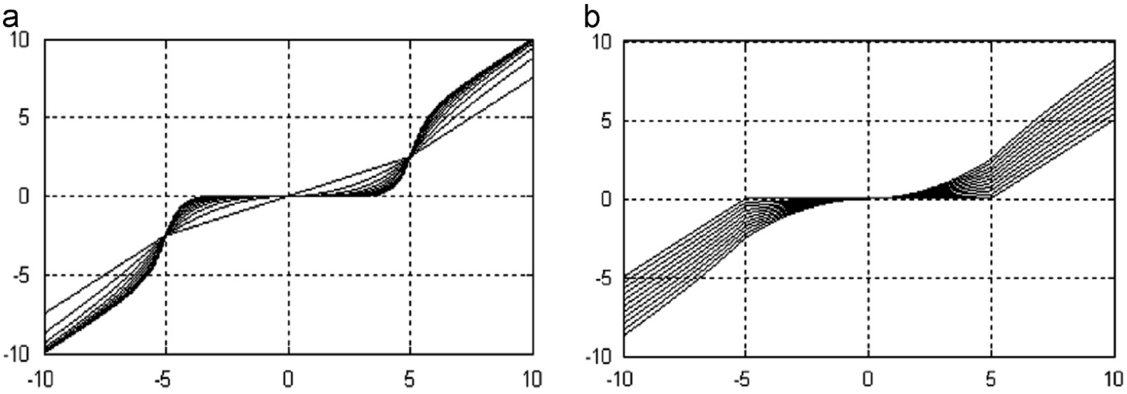

Fig. 2. (a) Behavior of adaptive thresholding function for $k=1$ and different values of m and n in the range [2, 10] and (b) behavior of adaptive thresholding function for $n=m=2$ and $k \in [0, 1]$, note that for $k\to 0$ the function tends to soft thresholding function [42].

[46,47], which are defined as:

$$\eta_{Zhang(1998)}(x,thr,k)=\begin{Bmatrix} x+thr-\frac{thr}{2k+1} & x<-thr \\ \frac{1}{(2k+1)thr^{2k}}x^{2k+1} & |x|\le thr \\ x-thr+\frac{thr}{2k+1} & x>thr \end{Bmatrix} \tag{7}$$

where, k is a positive integer. Note that the limit of $\eta_{Zhang}(x, thr)$ when k→∞1 is just the commonly used soft-thresholding function $\eta_s(x, thr)$. Therefore, the similar smoothness property of the estimate using new thresholding functions can be expressed as.

$$\eta_{Zhang(2001)}(x,thr,\lambda)=x+0.5\left(\sqrt{\left(x-thr\right)^2+\lambda}-\sqrt{\left(x+thr\right)^2+\lambda}\right) \tag{8}$$

where thr is the threshold and $\lambda>0$ is a user-defined (fixed) function parameter.

3.1. New nonlinear thresholding function

The new nonlinear thresholding function is formulated in Eq. (9). The foremost difference of this thresholding function is found in non-important coefficients. Previous thresholding functions set the coefficients below threshold value to zero, but the new nonlinear thresholding function tunes the coefficients by a polynomial function. Due to tuning property, capability of the thresholding function is increased; since, the coefficients can be attenuated easily in a manner that are below the threshold value and close to it to a value less than the far coefficients. In addition, for important coefficients, the function is garrote-type. Therefore, it leads to a more powerful function:

$$\eta(x,thr)=\begin{Bmatrix} x-0.5\dfrac{thr^2}{x} & |x|>thr \\ 0.5\dfrac{x^3}{thr^2} & |x|\le thr \end{Bmatrix} \tag{9}$$

Here, x is noisy WT coefficient and thr is threshold.

3.2. Adaptive thresholding function with three shape tuning parameters

In Eq. (9), three shape tuning parameters are added to improve flexibility and ability of new nonlinear thresholding function, which is formulated in Eq. (10).

$$\eta(x,thr,m,n,k)=\begin{Bmatrix} x-0.5\dfrac{thr^m\times k}{x^{m-1}}+(k-1)thr & x>thr \\ 0.5\dfrac{k\times|x|^n}{thr^{n-1}}\,\mathrm{sign}(x) & |x|\le thr \\ x+0.5\dfrac{(-thr)^m\times k}{x^{m-1}}-(k-1)thr & x<-thr \end{Bmatrix} \tag{10}$$

where, parameters n and m are used to calculate the shape of function for coefficients, which are lower and higher as compared to absolute threshold value, respectively; whereas k is used to compute the asymptote of the function. The parameter k determines the asymptote of the function. By varying parameter $k \in [0, 1]$, the thresholding function (η) behavior varies between hard and soft thresholding as shown in Fig. 2a. The parameter m decides the shape of thresholding function, which adds more flexibility in its behavior as shown in Fig. 2b. Here, x is noisy WT coefficient and thr is threshold.

4. Different evolutionary techniques for denoising of satellite images

4.1. Differential Evolution Algorithm (DE)

In recent years, much attention has been paid towards the application of evolutionary algorithms in various fields of image processing. The Differential Evolution (DE) Algorithm is proposed by Storn and Price [48] which is recently emerged as a simple yet very competitive evolutionary optimizer. The algorithm has effectively been applied to many multilevel thresholding and image processing problems, and has received a wider range of acceptance and popularity [49–50].

The version of DE algorithm employed in this work is known as the DE/best/l/exp or "DE1". The DE algorithm starts with the initialization of population of N_p, and an D-dimension vectors with parameter values, which are randomly and uniformly distributed between the pre-specified lower initial parameter bound $x_{j,low}$ and the upper initial parameter bound $x_{j,high}$:

$$x_{j,i,G} = x_{j,low} + \text{rand}(0,1)\cdot(x_{j,high} - x_{j,low}); \\ j = 1,2,...,D; i = 1,2,...,N_p; \quad G = 0. \tag{11}$$

The subscript G is the generation index to which the population belongs, where index i represents the ith solution of population index and j indicates the parameter index. There are three main operators in DE algorithm which are mutation, crossover and selection.

Mutation: To produce a trial solution, the mutation operator is applied to generate mutant vector v_i for each target vector χ_i in the current population. Therefore, initially DE algorithm mutates a best solution vector from the current population by adding to the scaled difference of two other vectors from the current population. A mutant vector is generated according to

$$V_{i,G} = X_{r1,G} + F \times (x_{r2,G} - x_{r3,G}); \quad \text{where, } r_1, r_2, r_3 \in \{1, 2, ..., N_p\} \tag{12}$$

where, $i = 1...NP$, r_1, r_2 are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$, and F is the mutation scale factor such that $F \in [01]$.

Crossover: once the perturbed individual $V_{i,G} = (v_{1,i,G}, ..., v_{n,i,G})$ is produced, it belongs to crossover operation with target individual$X_{i,G} = (x_{1,i,G}, ..., x_{n,i,G})$, which generates trial solution in the end $U_{i,G} = (u_{1,i,G}, ..., u_{n,i,G})$ as follows:

$$u_{j,i,G} = \begin{Bmatrix} v_{j,i,G} & \text{if } rand\ C_r \vee j = jj \\ x_{j,i,G} & \text{otherwise} \end{Bmatrix} \tag{13}$$

where, $j = 1, ..., n, jj \in \{1, ..., n\}$ is a random parameter's index, selected once for each i. The crossover rate $Cr \in [0,1]$ is set by the user that is used to control the section of parameters belonging to mutant vector which contributes to the trial vector.

Selection: at last, a selection scheme is used to improve the solutions. The selection process in DE algorithm is different from the other evolutionary algorithms. If the cost function of trial vector is less or equal to target vector, then trial vector replaces the target vector in the next generation. Otherwise, target vector remains in the population for at least one new generation:

$$X_{i,G+1} = \begin{Bmatrix} U_{i,G} & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{Bmatrix} \tag{14}$$

Here, f denotes the cost function. The overall procedures are repeated until a termination criterion is fulfilled or a predetermined generation number is attained.

A complete flowchart routine of the optimized adaptive thresholding function based methodology, depicting the detailed steps of overall algorithm for image denoising, is shown in Fig. 3.

4.2. Particle swarm optimization (PSO)

In PSO [51], possible solution and collection of possible solutions (search space) are known as particle position and swarm correspondingly. The PSO is dominated by two basic updating equations for particle position i, first is velocity updating equation, which is formulated as

$$V_i(k+1) = w \times V_i(k) + c_1\phi_1(P_{ibest}(k) - P_i(k)) + c_2\phi_2(G_{best}(k) - P_i(k)) \tag{15}$$

and second is position updating equation defined by

$$P_i(k+1) = P_i(k) + V_i(k+1) \tag{16}$$

where, w represents the inertia weight factor, which varies linearly from 0 to 1; while c_1 and c_2 are cognitive and social acceleration factors respectively, ϕ_1 and ϕ_2 indicates the uniformly distributed random numbers with range 0 to 1. The next velocity $V_i(k+1)$ of particle i can be computed by Eq. (15), and the next position $_{Pi}(k+1)$ is tracked by Eq. (16).

Fundamentally, particle position $_{\mathrm{Pi}}$ indicates one possible solution of the optimization problem and at each iteration, the objective function (fitness function) is measured by position vector $_{Pi}(k)$. The position vector corresponding to best fitness is called '*pbest*' and the overall best solution of all the particles in population is known as '*gbest*'. In this algorithm, the success of finding global optimum depends exceptionally on the initial value of control parameters such as w, c_1, c_2, ϕ_1, ϕ_2, swarm size (s) and the maximum iteration number. In [52–53], a detailed description of PSO is given.

4.3. Wind driven optimization (WDO) technique

WDO is a modern nature inspired global search optimization technique proposed by Zikri Bayraktar in 2010 [54–55]. It is basically motivated by the motion of infinitesimally tiny air particle present in earth's environment. In WDO, using Newton's second law of motion, trajectory of these particles can be expressed. In this optimization algorithm, "wind" corresponds to horizontal air motion in lowest layer of the earth's environment which is also known as troposphere. In atmosphere, wind blows in an effort to make equal air pressure. More exclusively, the air is used to move from high pressure to low pressure at a velocity, which is proportional to the pressure gradient. The beginning step of WDO is supported by Newton's second law of motion, which is used to

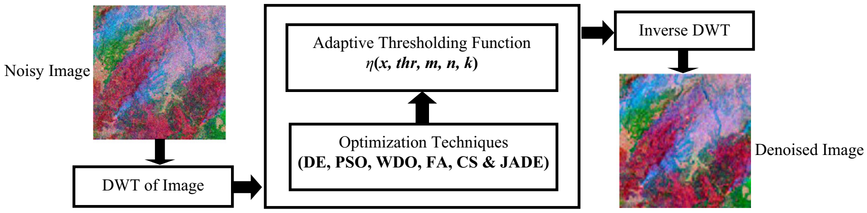

Fig. 3. An optimized adaptive thresholding function based image denoising using nature inspired algorithms.

determine the accurate results specifically for the analysis of atmospheric motion.

Now, Newton's Law states that the net force applied on air parcel causes acceleration a in the direction of net force applied, and it is mathematically represented as

$$\rho\vec{a} = \sum \vec{F}_i \tag{17}$$

where, a is the acceleration vector, ρ is the air density for an infinitesimal element of volume, and F_i are the individual forces acting on the air parcel. The equation that relates air pressure to its density and temperature is given by the ideal gas law:

$$P = \rho RT \tag{18}$$

where P is the pressure, R is the universal gas constant and T is the temperature.

This motion of air particle in Eq. (17) is basically affected by the four major forces that either cause the wind to move in a specific direction or deflect it from its path. The most observable force causing air to move is the pressure gradient force (***FPG***), although the friction force (***FF***) simply acts to oppose such motion as described in Eq. (18). Whereas, the gravitational force (***FG***) acts as a vertical force in three-dimensional atmosphere when it is mapped to N-dimensional space. The Coriolis force (***FC***) is caused by the rotation of earth, and deflects the path of wind from one dimension to another. In WDO, it is implemented as a motion in one dimension that affects the velocity in another [54].

The physical equations that govern each of these forces are given below:

$$\vec{F}_{PG} = -\nabla P \delta V \tag{19}$$

$$\vec{F}_G = \rho \delta V \vec{g} \tag{20}$$

$$\vec{F}_c = -2\Omega \times \vec{u} \tag{21}$$

$$\vec{F}_F = -\rho\alpha\vec{u} \tag{22}$$

where, ∇P is the pressure gradient, δV represents an infinitesimal air volume, Ω represents rotation of the earth, g is the gravitational acceleration constant, α is friction coefficient and u is velocity vector of the wind. By putting all the above mentioned forces in Eq. (17), the summation of all four forces can be obtained by Eq. (23):

$$\rho\vec{u}\Delta t = (\rho\delta V\vec{g}) + (-\nabla P\delta V) + (-\rho\alpha\vec{u}) + (-2\Omega \times \vec{u}) \tag{23}$$

If an infinitesimal air parcel is considered which is moving with wind, then the velocity update equation can be determined by (31). On the basis of ideal gas law equation from (18), ρ can be written in terms of pressure, and a unity time step ($\Delta t = 1$) can be assumed for simplicity. The velocity update equation is:

$$\vec{u}_{new} = \left((1-\alpha)\vec{u}_{old}\right) + g\left(-\vec{x}_{old}\right) + \left[\left|\frac{P_{\max}}{P_{old}} - 1\right| RT(x_{\max} - x_{old})\right] + \left[\frac{-c_{old}^{otherdim}}{P_{old}}\right] \tag{24}$$

which is the velocity and position updating equation for WDO respectively. Here, P is the pressure, R is universal gas constant, T is temperature, $c = -2\Omega RT$ called Coriolis coefficient. In Eq. (24), updated velocity for the next iteration, $\boldsymbol{u_{new}}$, depends on the current iteration velocity ($\boldsymbol{u_{old}}$), current location of air parcel in search space ($\boldsymbol{x_{old}}$), distance from the highest pressure point that has been found ($\boldsymbol{x_{max}}$), maximum pressure (P_{max}), pressure at the current location (P_{old}), temperature (T), gravitational acceleration (g), and constants R**,** α, and c.

In WDO expression (24), the pressure term is analogous to the fitness of a chromosome in GA terminology. If WDO is compared with PSO, similar velocity update equations can be realized. After the velocity of parcel is updated by Eq. (24), the position of air parcel can be updated by Eq. (25):

$$\vec{x}_{new} = \vec{x}_{old} + (\vec{u}_{new} \times \Delta t) \tag{25}$$

where, $\vec{x}_{new}$ and $\vec{x}_{old}$ are the current and new position of air parcel in search space respectively. In Eq. (25), $\vec{x}_{old}$ represents that the air parcel would continue to move on its previous path with some opposition that is created due to friction. $\vec{u}_{new}$ is an attractive force that pulls against the center of coordinate system. Δt indicates the force against position of maximum pressure that is assumed to be the global best position for the optimization problem. $\vec{x}_{new}$ follows the Coriolis force, which is a deflecting force. In this manner, WDO offers a simple yet highly effective way to solve the complex optimization problems. From the above discussion, it can be concluded that WDO is controlled by RT, c, α, g and number of iteration. Detailed description of WDO is given in [55–56].

4.4. Firefly algorithm (FA)

The firefly algorithm is a recently developed swarm-based approach for optimization, which is inspired by the social behavior of fireflies and the phenomenon of bioluminescent communication. In this concept, basically the variation of light intensity and formulation of attractiveness are two essential issues. Yang [57] has explored the attractiveness of a firefly on the basis of their brightness, which in turn is associated with the encoded objective function. The attractiveness is proportional to their brightness. Moreover, each member x_i of the firefly swarm is recognized through the brightness I_i that can be simply described as an inverse of a cost function for a minimization problem.

In the firefly algorithm, there are three idealized rules: (i) all the fireflies are unisex in order that one firefly will be fascinated to other fireflies despite of their sex; (ii) attractiveness is proportional to their brightness; hence, out of any two flashing fireflies, the less brighter one will move in the direction of brighter one, if there is no brighter firefly in comparison with an individual one, then it will move randomly. Since; firefly attractiveness needs to choose any monotonically decreasing function of the distance $r_{i,j} = d(x_j,\ x_i)$ to the selected j-th firefly, in such a way that the exponential function is followed as [58]:

$$r_{i,j}\|x_i - x_j\| = \sqrt{\sum_{k=1}^{c}(x_{i,k} - x_{j,k})^2} \tag{26}$$

$$\beta \leftarrow \beta_0 e^{-\gamma r_{i,j}} \tag{27}$$

in which, β_0 represents the attractiveness at $r_{i,j} = 0$ and c stands for light absorption coefficient at the source.

The movement of a firefly i is fascinated to another more attractive firefly j which is computed by

$$x_{i,k} \leftarrow (1-\beta)x_{i,k} + x_{j,k} + u_{i,k} \tag{28}$$

$$u_{i,k} = \alpha\left(rand1 - \frac{1}{2}\right) \tag{29}$$

If there is no brighter firefly in comparison with an individual firefly x_i, then it will move randomly according the given criteria:

$$x_{i^{max},k} \leftarrow x_{i^{max},k} + u_{i^{max},k} \qquad \text{for } k = 1, 2, ..., c \tag{30}$$

$$u_{i^{max},k} = \alpha\left(rand2 - \frac{1}{2}\right) \tag{31}$$

where, $rand1 \approx U(0,\ 1)$ $rand2 \approx U(0,\ 1)$ indicates the random values, which are determined from the uniform distribution. (iii)

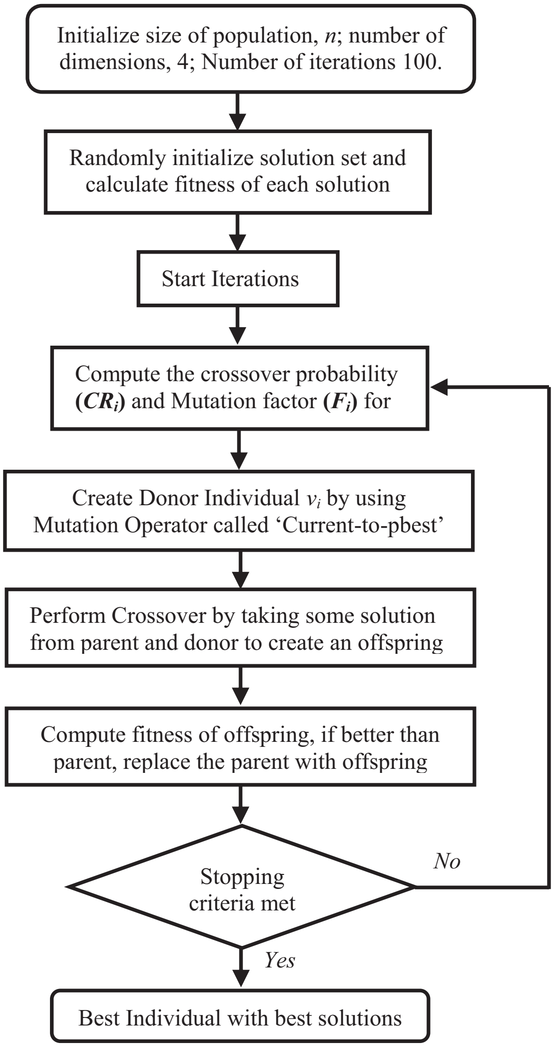
Initialize size of population, n; number of dimensions, 4; Number of iterations 100.
Randomly initialize solution set and calculate fitness of each solution
Start Iterations
Compute the crossover probability (CRi) and Mutation factor (Fi) for
Create Donor Individual vi by using Mutation Operator called 'Current-to-pbest'
Perform Crossover by taking some solution from parent and donor to create an offspring
Compute fitness of offspring, if better than parent, replace the parent with offspring
Stopping criteria met
No
Yes
Best Individual with best solutions

Fig. 4. Flowchart of JADE algorithm.

The brightness of a firefly is affected or calculated by the landscape of the fitness function $\varphi(.)$. In other word, the brightness I of a firefly at a specific position x can be selected as $I(x)$, which is proportional to the value of fitness function $u(x)$.

4.5. Cuckoo search (CS) algorithm

In this paper, a new evolutionary optimization algorithm is used which is inspired by the lifestyle of a bird family called cuckoo. CS algorithm is a population based elitist search algorithm [59–60]. In this algorithm, each new solution attempts to search around the best solution found previously. CS has only two control parameters. The general structure and problem solving efficiency of CS technique has been explored in detail in [56]. Cuckoo birds are most popular because of their attractive voice and fascinating singing style; moreover, their reproduction policy is also one of the most aggressive among the birds. Cuckoos can engage indirect conflict with the host birds. Usually, host birds either throw the alien eggs from the nest or they leave the nest to make a new nest.

There are three idealized rules in the CS algorithm.

i. Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
ii. The best nests with high quality of eggs will carry over to the next generation.

Table 1
Specific values for given parameters used in each algorithm.

Algorithms	Parameters	Values
DE	Number of objectives	1
	Number of constraints	0
	Number of decision variables	4
	Scaling factor	0.5
	Crossover probability	0.2
	Maximum number of generations	100
	Function evaluations bound	150
PSO	Swam size	200
	No. of iterations	100
	Cognitive, social and neighborhood acceleration (C_1, C_2, C_3)	2, 2, 1
	Lower bound lb (W_{min}) and Upper bound ub (W_{max})	1 and 256
	Error goal and Max Trial limit	1e-7 and 500
	Value of velocity weight at the beginning	0.95
	Value of velocity weight at the end of the PSO iterations	0.4
	The fraction of maximum iterations, for which W is linearly varied	0.7
	Maximum velocity step, Constriction factor and Neighborhood size	1
	Value of Global Minima	0
WDO	Population size	20
	Maximum no. of iteration	100
	RT coefficient	2
	Gravitational constant (g)	0.2
	Constant in the update equation	0.5
	Coriolis effect coefficient (c)	0.4
	Maximum allowed speed or velocity limit	0.3
FA	No. of iterations	100
	Alpha, betamin and gamma	0.5, 0.2 and 1
	Lower bound Lb	[0,0,0.1,0.1]
	Upper bound Ub	[1000, 1, 20, 4]
	Number of dimensional problems	4
CS	Number of nests	25
	No. of iterations	100
	Lower bound Lb and Upper bound Ub	1 and 256
	Step size (α)	1
	Mutation probability value (p_a)	0.25
	Scale factor (β)	1.5
JADE	No. of iterations	100
	Scale parameter and standard deviation	0.1
	Lower bound Lb	[0,0,0.1,0.1]
	Upper bound Ub	[1000, 1, 20, 4]
	Constant c and p,	0.1 and 0.05
	Crossover probability CR_i	[0, 1]
	Independent normal distribution with mean *CRm* and *Fm*	0.5

iii. The number of available host nest is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability of $p_a \in [0, 1]$.

Now, in such cases, the host bird either abandons the nest to build a completely new nest at new location or throws the egg away. In order to simplify the last assumption, many authors have assumed that it can be approximated by a probability p_a of the n nest, which are replaced by the latest nests (with random new solutions). Every egg in a nest stands for one solution and a cuckoo egg indicates a new solution, which aims to exploits new and potentially better solutions (cuckoos). Start iteration, generate new nest by Levy flight but keep the current best. For generating new solution, $x_i(t+1)$ for cuckoo i, a Lévy flight is performed by the Eq. (32):

$$x_i(t+1) = x_i(t) + \alpha \oplus \text{Lévy}(\lambda) \tag{32}$$

where, α is the step size ($\alpha > 1$) and is related to the size of the problem. In most cases, $\alpha = 1$ is used. The product $\oplus$deals with entry wise multiplications process. However, Lévy flights provide a random walk whereas their random step lengths are drawn from a Lévy distribution for large steps defined by (33):

$$Lévy(\lambda) = t^{-\lambda} \qquad \text{where } 1 < \lambda \le 3 \tag{33}$$

Lévy function can be changed according to application. Mantegna's algorithm is one of the Lévy function. It has an infinite variance and infinite mean. Lévy flights are more proficient in case of extensive search space in comparison with Brownian random walks because variance (σ^2) of Levy flights increases at greater rate than Brownian random walks. Moreover, variance of Levy flights distribution is expressed as:

$$\sigma^2(t) \sim t^{2-\beta} \qquad \text{where } 1 \le \beta \le 2 \tag{34}$$

Which expands at a greater rate than the linear relationship of Brownian random walks followed by $\sigma^2(t) \to t$. On the basis of above expression, it can be noticed that the CS algorithm is mainly controlled by four parameters: number of iterations, number of host nests (*NS*), step size (α) and probability (p_a).

4.6. Differential Evolution Algorithm (JADE)

JADE is an adaptive version of DE. In 2011, a new differential evolution (DE) algorithm, JADE, is proposed by Zhang and Sanderson [61] to improve the optimization performance by implementing a new mutation strategy "DE/current-to-*p*best" with optional external archive and updating control parameters in an adaptive manner. Basically, JADE algorithm includes a new mutation strategy (i.e., DE/current-to-pbest) to be used with standard DE. This approach implements a randomly chosen individual to evolve the best solution in the population at that moment.

JADE algorithm differs from DE in 3 aspects. First, JADE can optionally utilize an archive of parent solutions recently replaced with more successful offspring. The archive is used in the JADE mutation process. The second one is a special mutation operator, which is known as "current-to-*p*best". DE/current-to-pbest, a mutation vector is generated in the following manner:

$$v_i = x_i + F_i \cdot (x^p_{best} - x_i) + F_i \cdot (x_{r1} - (x_{r2}) \tag{35}$$

where, x_i is the parent individual, where x^p_{best} is randomly selected as one of the top 100 *p*% individuals in the current population with $p \in [0, 1]$. x_{r1} and x_{r2} are individuals randomly chosen from the population and from the union of current population and archive respectively. F_i is the mutation factor that is associated with x_i and is re-generated at each generation by the adaptation process.

The third and most important difference is the adaptation of *F* and C_r. In classic DE, both factors are usually constant (or sampled from a static distribution). In JADE, mutation coefficient is produced with $F \sim N$ (μ_c, 0.1) and crossover coefficient with $C_r \sim$Cauchy (μ_F, 0.1). Here, *N* (μ_c, 0.1) is a normal distribution function with average value of μ_c and standard deviation value of 0.1, and Cauchy (μ_F, 0.1) is a cauchy distribution function with local parameter value of μ_F and scale parameter value of 0.1. JADE algorithm solves the real-valued optimization problems much more successfully than the standard DE algorithm. The detailed knowledge of JADE algorithm can be found in [61]. In addition, a generalized flowchart of JADE algorithm is given in Fig. 4.

Based on the above-mentioned rules, the basic steps of JADE algorithm are summarized as follows:

Step 1 : Select the size of population, *n*, number of different solutions. The number of solutions has been taken as 15. Next, set the stopping criteria as either the fitness value less than some fixed value or the number of iteration. In this problem, the fixed number of iteration has been chosen. Set the dimension of problem. Here, the dimension of problem is four. Also, set the limiting values for each dimension.

Step 2: Randomly, initialize the solution set by generating population of size *n*, in which each are having 4 dimensions.

Step 3: Evaluate fitness for each of the initialized solution. Then sort it in ascending order on the basis of their fitness values.

Step 4: Start iteration. Compute the crossover probability, CR_i, from a normal distribution with mean, μ_{cr} and, Standard deviation of 0.1. Similarly, compute Mutation Factor, F_i, from a Cauchy Distribution with location parameter μ_f and, Scale parameter 0.1. The parameters, μ_{cr} and μ_f, are updated in each generation using Arithmetic and Contra-harmonic mean respectively, of CR_i and F_i.

Step 5: Next a donor individual, V_i, is created by using the mutation operator called "Current-to-pbest",

$$V_i = X_i + F_i * (X^p{}_{best} - X_i) + F_i * (X_{r1} – X_{r2}) \tag{36}$$

where, X_i is the parent individual, X^p_{best} is an individual randomly chosen from the best 100% individual in current population, p € (0,1], X_{r1} and X_{r2} are individuals randomly chosen from the population and from the union of current population and the archive, respectively. The *Fi* is the mutation factor. The individuals X^p_{best}, X_{r1} and X_{r2}, and the value of F_i are chosen as a new for each mutation.

Step 6: After Mutation, Crossover is performed by taking some solution components from the parent X_i and other components from the donor V_i , resulting in the creation of an offspring.

Step 7: The offspring u_i=(u^i,1,...,u_i, 4) is as follows:

$$\begin{aligned} u_{i,j} &= v_{i,j} \quad \text{if} \quad r_j \le CR_i \quad \text{or} \quad j = j_i,\ rand, \\ & x_{i,j} \qquad \text{otherwise,} \end{aligned} \tag{37}$$

where, r_j is a random number uniformly distributed in [0, 1], $CR_i \in$ [0, 1] is the crossover probability representing the average proportion of components that the offspring gets from its donor, and j_i, *rand* is the randomly chosen index of the solution component surely taken from the donor.

Step 8: The new population so obtained is sorted in ascending order on the basis of their fitness, and the global best solution is updated.

Step 9: The above steps are repeated until the stopping criteria is achieved, thereby giving the best fitness and corresponding best individual.

4.7. Details steps of the proposed denoising algorithm

Step 1: Take an input image, extract the red, green and blue pixel matrix and put Gaussian noise of mean zero and some predetermined variance in each of them.

Step 2: Apply DWT for both input and noisy image matrixes and obtain coefficients upto level 2, by again taking wavelet transform of the approximation matrix so obtained from level 1 transform.

Step 3: Set the required parameters for the optimization algorithm **(JADE)** such as number of solutions as 4, number of generations as 40 etc.

Step 4: Now put these noisy image coefficients (1st and 2nd level) to the optimizing algorithm **(JADE)** which includes adaptive thresholding function. Obtain solutions from the optimization algorithm and after computing it through three shape adaptive thresholding function, calculate the fitness value for each solution and for each subband of noisy image.

Step 5: Obtain the best solution by Eq. 10 (*thr, k, m, n*) through the optimization algorithm (**JADE**), and route it to thresholding function along with noisy coefficients, to obtain the required

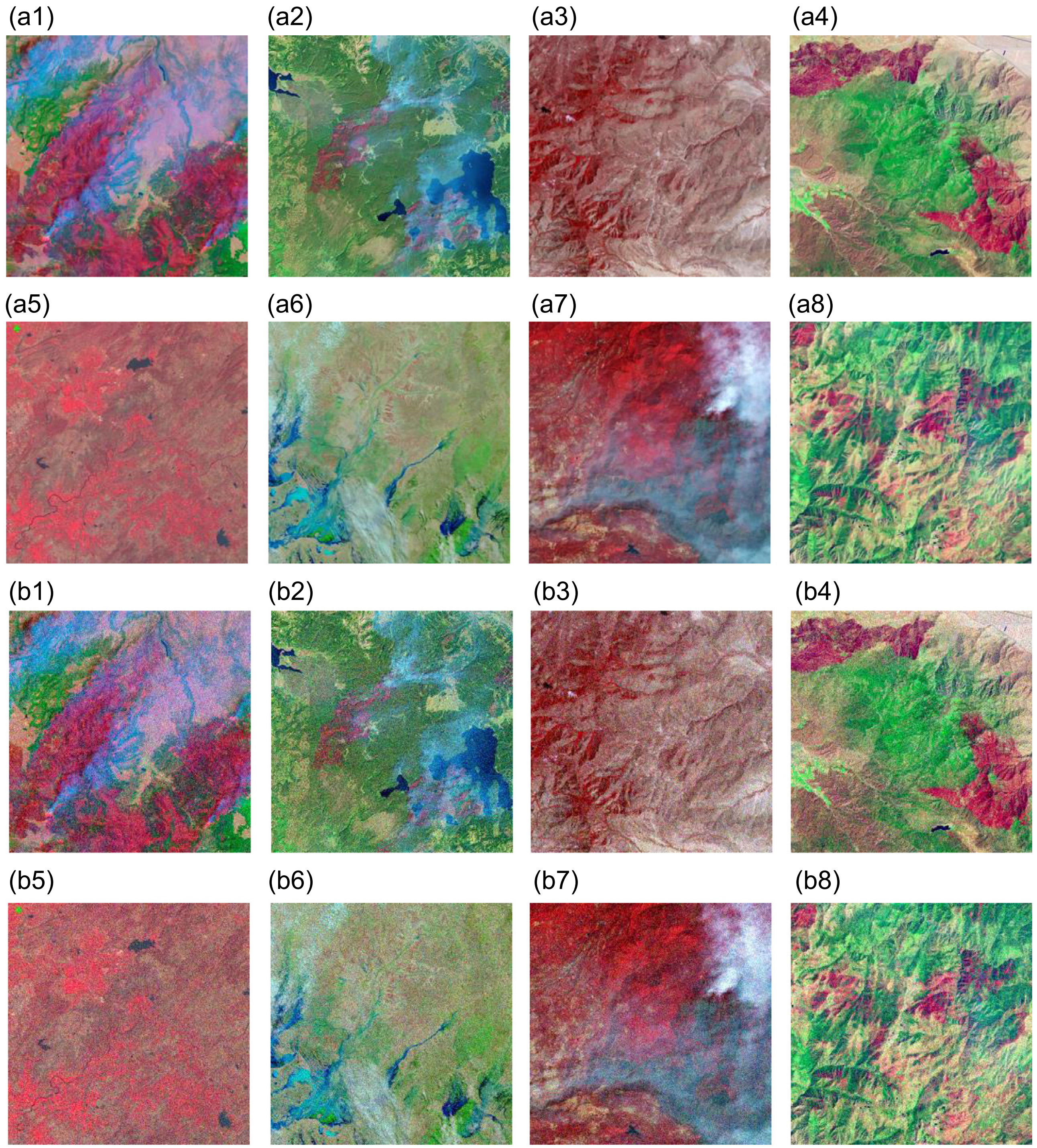

Fig. 5. Satellite images, (a1–a8) original satellite images of size 256 × 256 pixels [62], and (b1–b8) corresponding noisy images with σ=20 Gaussian noise.

matrix of coefficients, say **Thresholded matrix**. In this function by applying differentiability property to the proposed thresholding function, a nonlinear differentiable function with two shape tuning parameters k and m is determined. By adjusting k parameter, a near-optimum function between hard and soft functions is resulted. Moreover, via tuning parameter m, the near-optimum thresholding function is adjusted to the optimum one by applying small changes. In common practice, optimizing parameter k works similar to a global search and optimizing parameter m works like local search in finding the best thresholding function.

Step 6: Use these thresholded matrixes and apply inverse DWT to first obtain the coefficient matrix of level 1, and then with the help of other matrixes of coefficients of level 1 obtain back the modified, denoised image.

Step 7: Calculate the required performance parameters such as PSNR, MSE, SSIM, FSIM and CPU time.

Since the proposed technique contains the optimization algorithms as one of the important constituent of our image denoising algorithm and we have experimented our technique with each of them so we have provided the basic introduction of each of the meta-heuristic algorithm used, to provide a better understanding of what each algorithm doing and why our proposed technique is best suited with JADE Algorithm.

The main Focus of our paper is the study of various optimization algorithms for optimizing the parameters lambda; Threshold value, *k*; a measure between soft and hard thresholding, m and n; the exact shape tuning parameters of the thresholding function giving the freedom to thresholding function to modify the coefficients adaptively, of the adaptive thresholding function, a modification of zhang functions, to remove the noise in an image. For finding the optimum value of the above mentioned coefficients we have used six most used meta heuristic algorithms and upon comparing the results we found that JADE gives the best results.

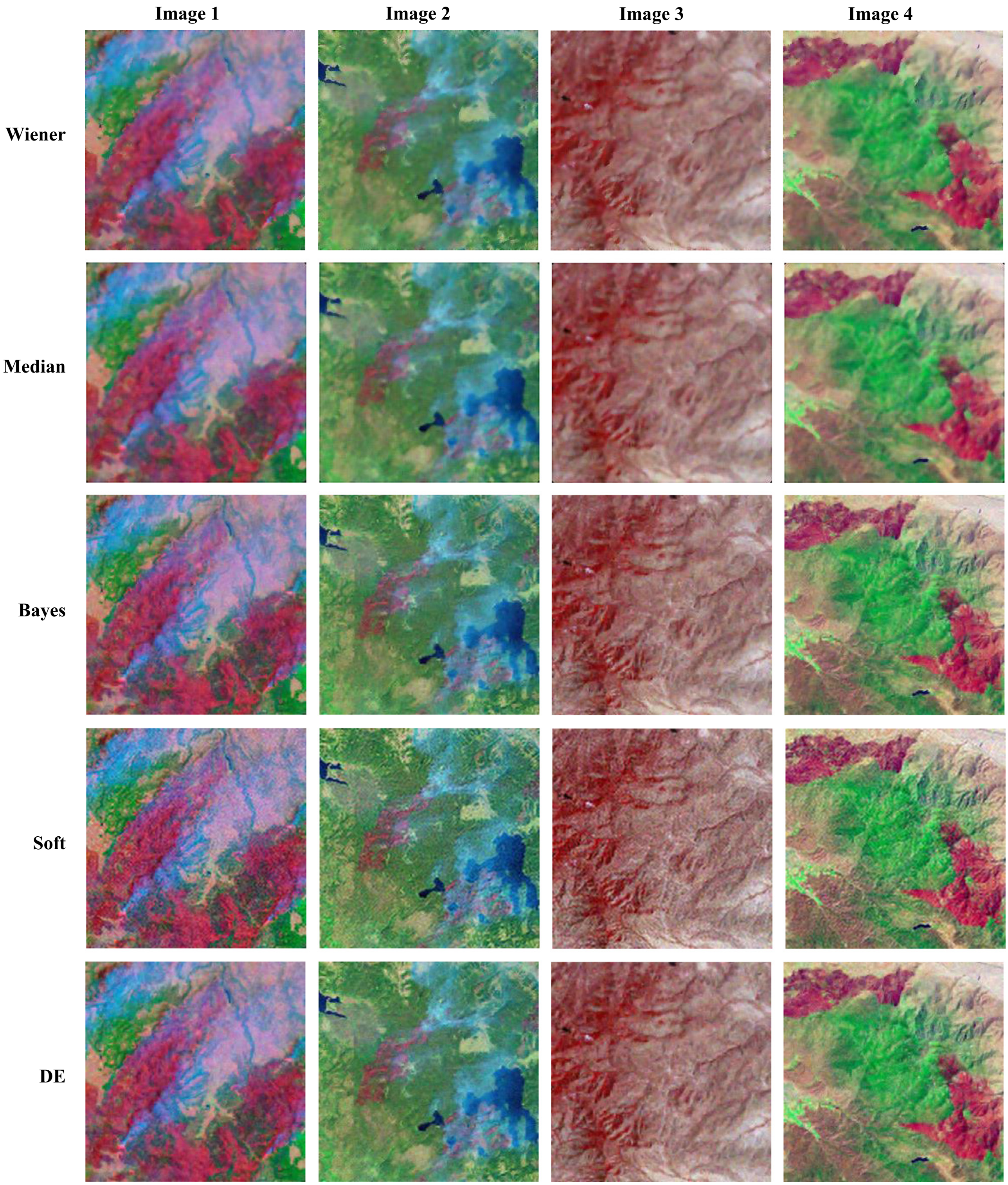

Fig. 6. Comparison of different classic filters (Weiner, Median, Bayes and Soft Threshold) images, DE, PSO, WDO, FA, CS and proposed **JADE** algorithm based adaptive thresholding function results with σ=**20** Gaussian denoising for test image 1, test image 2, test image 3 and test image 4.

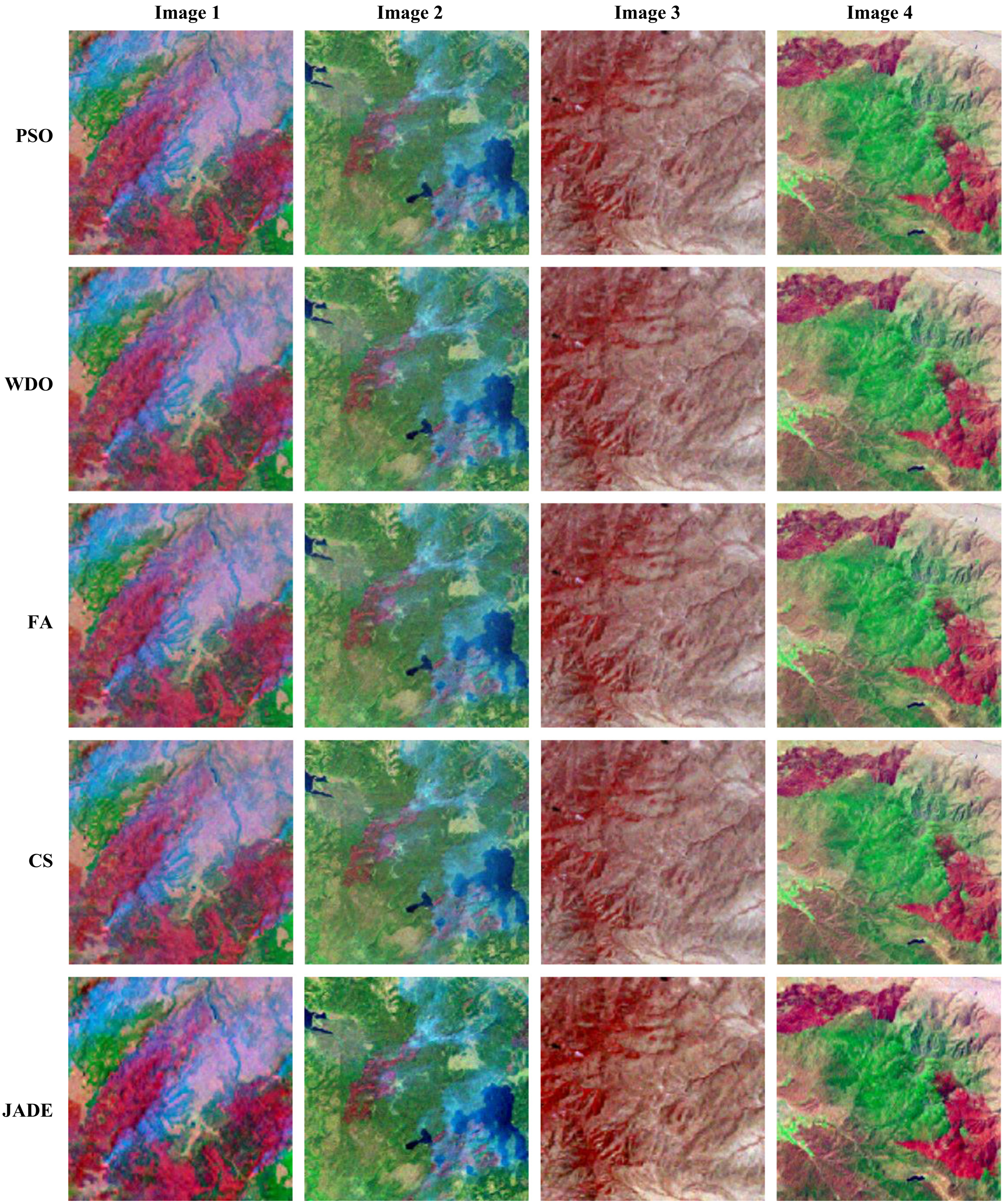

Fig. 6. (*continued*)

5. Assessment parameters and discussion

In this experimental section, different well known image processing matrices such as peak signal-to-noise ratio (PSNR) and mean-squared error (MSE), structural-similarity index (SSIM) and feature similarity index (FSIM) are used to compare the image denoising quality. PSNR and MSE depend directly on the image intensity values, which usually indicate the strength and accuracy of final reconstructed signal or image. The MSE represents the cumulative squared error between the denoised and original image. Lower the value of MSE, lower will be the error. Performance of this method is measured in terms of following significant parameters:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - \tilde{I}(i,j)]^2 \quad (38)$$

PSNR represents the measure of peak error and is expressed in decibels, which is defined by:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad (39)$$

where, M, N are the size of image, I is the original image and $\tilde{I}$ is the denoised image at a particular noise variance.

On the other hand, SSIM and FSIM are exploited to measure the similarity among the images computed as a result of denoised image carried out by the proposed methodology and the original images. *SSIM* is an image quality assessment based on the degradation of structural information. The SSIM is used to compare the

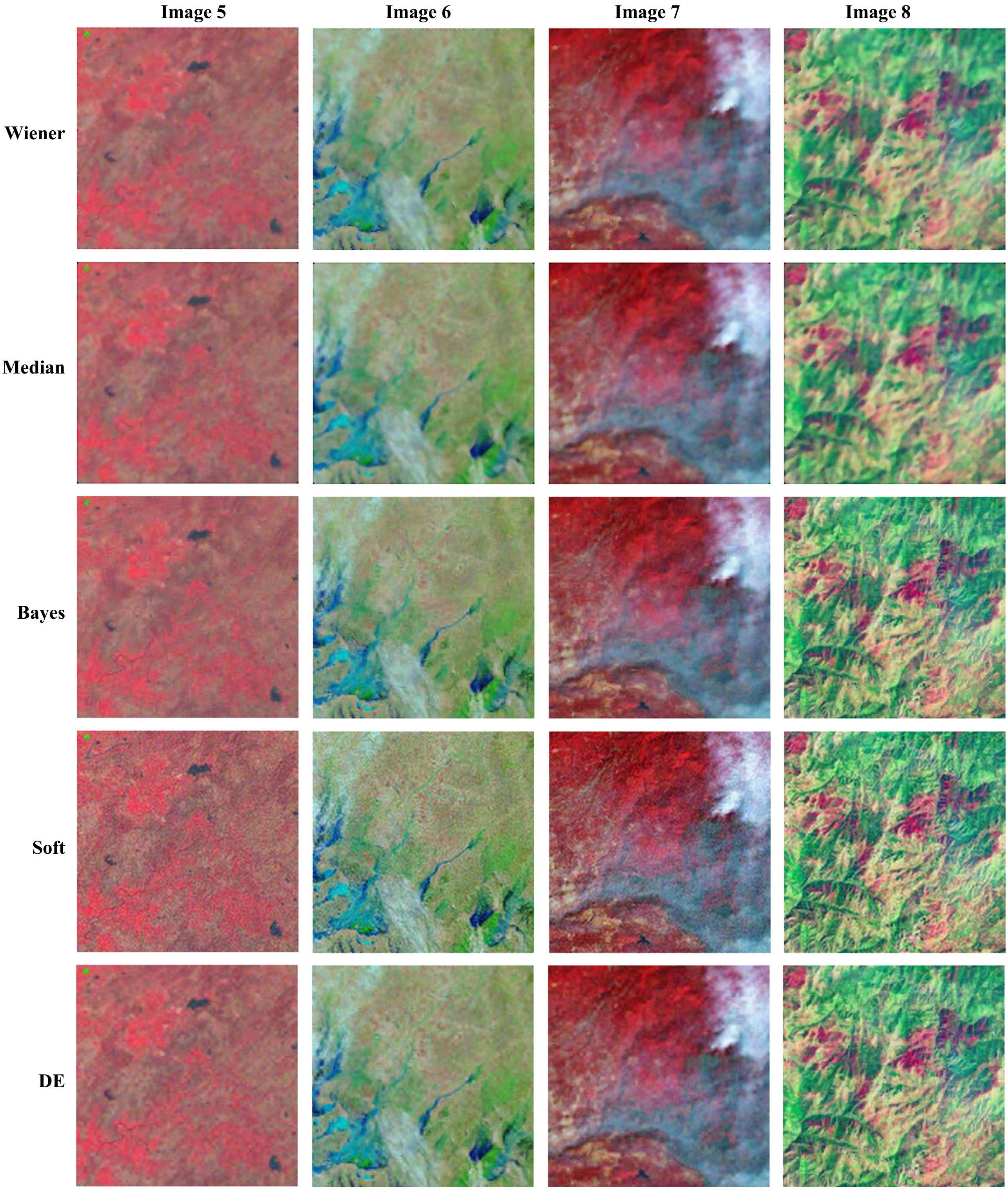

Fig. 7. Comparison of different classic filters (Weiner, Median, Bayes and Soft Threshold) images, DE, PSO, WDO, FA, CS and proposed **JADE** algorithm based adaptive thresholding function results with σ=**20** Gaussian denoising for test image 5, test image 6, test image 7 and test image 8.

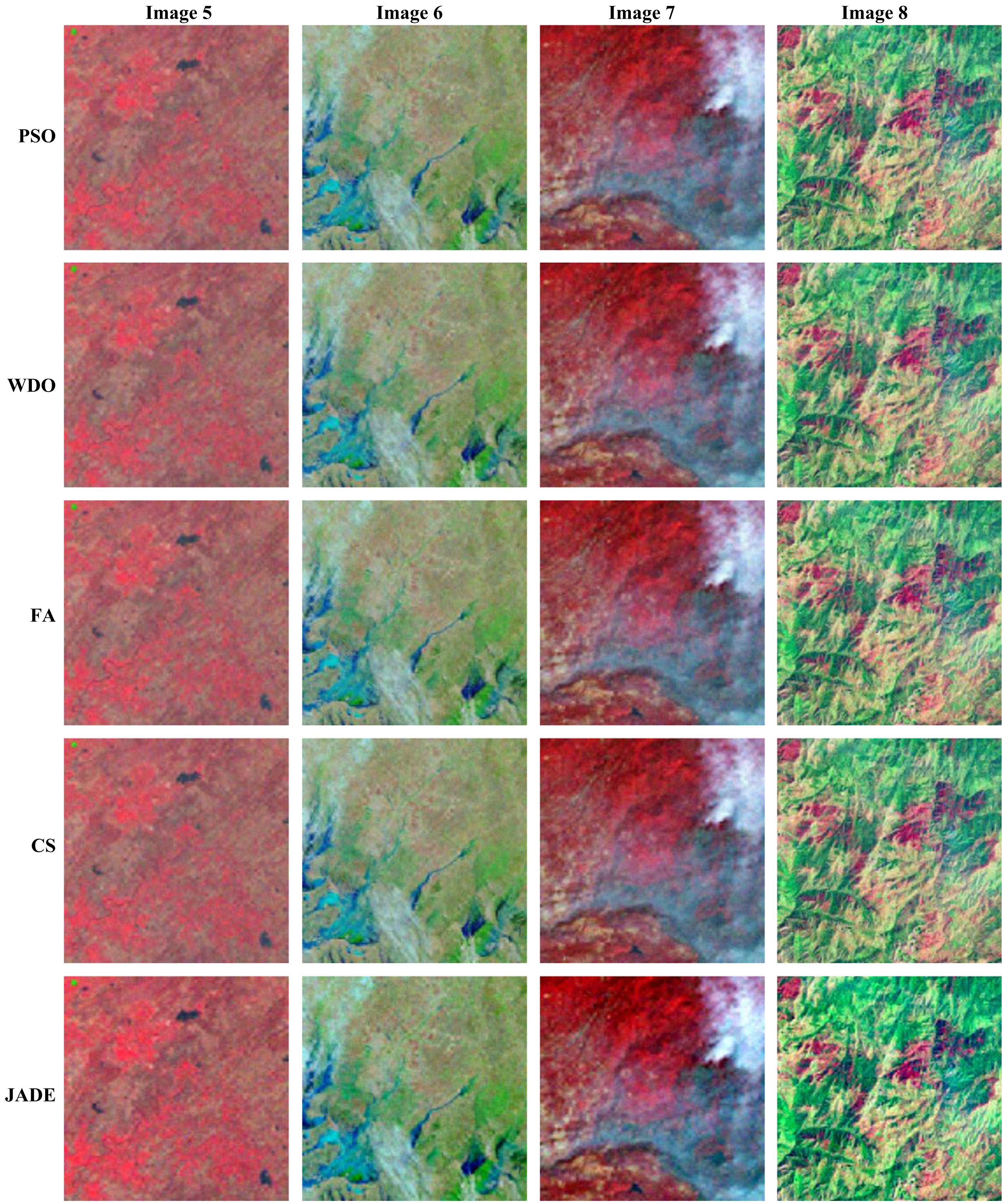

Fig. 7. (*continued*)

structures of original and thresholded image [56]. The SSIM index is calculated as:

$$\mathrm{SSIM}(x,y)=\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu^2{}_x+\mu^2{}_y+C_1)(\sigma^2{}_x+\sigma^2{}_y+C_2)} \tag{40}$$

where, μ_x and μ_y stands for mean intensity of image of x and y respectively, σ_x and σ_y indicates the standard deviations of x and y respectively, σ_{xy} is the local sample correlation coefficient between x and y. C_1 and C_2 are the constants, and are included to avoid instability when $\mu^2{}_x+\mu^2{}_y$are very close to zero. Here, $C_1=C_2=0.6$. The SSIM can take values in $[-1, 1]$ range, and a higher value of SSIM shows better performance. The SSIM metric can be extended for true color RGB images as following:

$$\mathrm{SSIM}=\sum_c \mathrm{SSIM}(x^c,y^c) \tag{41}$$

Table 2
PSNR result comparison for various test images and different noise variance (σ) values with classical filtering techniques, different optimization and proposed **JADE** algorithm based adaptive thresholding function.

Test images	Σ	**PSNR**									
		Weiner	Median	Bayes [40]	Soft threshold	DE	PSO	WDO	FA	CS	**JADE**
1	10	32.8338	32.6134	34.9900	33.8485	35.1850	34.8082	35.0222	35.1635	35.2505	**36.1797**
	20	32.1047	31.6973	32.1209	31.2545	32.6954	32.4770	32.7229	32.5651	32.6674	**34.2669**
	30	31.3394	30.9232	31.1701	30.0073	31.4346	31.2635	31.3133	31.3729	31.4176	**31.6350**
2	10	31.8894	31.2372	33.3907	32.4280	33.7110	34.8082	33.5953	33.6680	33.8497	**36.9225**
	20	31.3523	30.7934	31.4085	30.7795	31.4885	31.5123	31.5741	31.4537	31.6528	**32.8589**
	30	30.8510	30.3403	30.0043	29.7947	30.5831	30.6387	30.7578	30.5747	30.7723	**32.3617**
3	10	31.9323	31.6623	34.1883	33.4106	34.3716	34.2625	34.1239	34.3780	34.4662	**36.6633**
	20	31.4192	31.0006	31.8989	31.1750	32.0102	31.9557	31.9149	31.9672	31.9949	**34.2927**
	30	30.9632	30.4382	31.0241	29.9807	31.0256	31.0765	31.0649	31.0524	31.0987	**32.8392**
4	10	31.0291	30.8029	33.2332	32.1517	33.6619	33.5664	33.9853	33.6407	33.7090	**37.6411**
	20	30.6887	30.3497	31.2544	30.7035	31.3869	31.3740	31.3308	31.3286	31.4122	**33.2235**
	30	30.3820	29.9414	30.5580	29.7839	30.5276	30.5094	30.6208	57.0916	30.5647	**32.0605**
5	10	34.0106	33.5081	35.2673	34.5668	35.7434	35.6335	35.9582	35.6868	35.7644	**40.7673**
	20	32.9737	32.4075	33.1983	31.4882	33.2894	33.1890	33.7130	33.2911	33.3798	**34.2579**
	30	31.8208	31.4734	31.9016	29.9983	32.0490	31.9509	31.9622	31.9271	32.2385	**32.4373**
6	10	33.2364	32.8662	34.7944	33.9581	34.9366	34.7732	34.7560	34.7689	34.9396	**41.8929**
	20	32.4152	31.8660	32.4986	31.3535	32.5909	32.5174	32.7323	32.4992	32.6164	**36.8412**
	30	31.1510	31.0402	31.4501	30.0083	31.5798	31.4818	31.5490	31.4726	31.5019	**34.6800**
7	10	33.9070	33.3304	35.4473	34.7232	35.6634	35.5672	35.6976	35.7304	35.8015	**44.7822**
	20	32.9634	32.2816	33.0729	31.5524	33.2217	33.1244	33.1470	33.1805	33.2722	**34.4170**
	30	31.7877	31.4230	31.8820	30.1373	31.9356	31.7458	32.1646	31.9524	32.0173	**32.3324**
8	10	30.1947	30.1616	33.1014	31.8288	33.4248	33.2258	33.7779	33.3608	33.5097	**36.3314**
	20	29.9271	29.7890	30.9825	30.5035	31.1320	31.0584	31.3459	31.0666	31.1385	**33.5781**
	30	29.7426	29.4832	30.2081	29.6592	30.2662	30.2626	30.2948	30.2600	30.2625	**31.4885**

Here, x^c, y^c correspond to c^{th} channel of the original image and the denoised image at a particular noise variance; where, c stands for channel number (i.e., $c\ = 1, 2, 3$ in true color RGB images).

FSIM is used to calculate the feature similarity between input and the denoised images [56] which is formulated as:

$$\text{FSIM} = \frac{\sum_{X \in \Omega} S_L(X) PC_m(X)}{\sum_{X \in \Omega} PC_m(X)} \tag{42}$$

where, Ω represents the entire image, and $S_L(x)$ indicates the similarity between denoised image and original image. The FSIM metric can be extended for true color RGB images as following:

$$\text{FSIM} = \sum_{c} \text{FSIM}(x^c, y^c) \tag{43}$$

The motivation of our proposed techniques comes from the beauty of Meta heuristic optimization algorithms which can find the best possible values of the parameters in accordance with a cost or fitness function. Since for denoising we require a thresholding function which can find an optimum threshold value such that there is no compromise with the details (edges, boundaries and other features) of the image and give good denoised results.

The main Focus of our paper is the study of various optimization algorithms for optimizing the parameters lambda; Threshold value, k; a measure between soft and hard thresholding, m and n; the exact shape tuning parameters of the thresholding function giving the freedom to thresholding function to modify the coefficients adaptively, of the adaptive thresholding function, a modification of zhang functions, to remove the noise in an image. For finding the optimum value of the above mentioned coefficients we have used six most used meta heuristic algorithms and upon comparing the results we found that JADE gives the best results.

JADE is one of the Meta-heuristic optimization algorithms that optimize the values of shape tuning parameters lambda, k, m and n respectively. It itself forms an important part of our proposed denoising technique as the one which is optimizing the above mentioned values. Since our algorithm is performing denoising in the frequency domain with the help of an adaptive thresholding function which along with threshold, modifies the value of the wavelet coefficients hence denoising the image. The basic idea of our proposed method is to first transform the image from spatial domain to frequency domain as the noise mostly reside in the higher frequency bands along with the detail coefficients (edges, boundaries and other features). Now when the wavelet coefficients have been obtained (upto two levels), finding an optimum coefficient value (Threshold) with a thresholding or shrinkage function is the next step. Since the image features vary with different images, so noise effects different image differently, so by using an adaptive thresholding function noise content can be removed efficiently. Hence we chose an adaptive function, a modification of Zhang function, which has four variables i.e shape tuning parameters of which choosing optimum values can modify coefficients effectively without compromising Image details. Now for finding these best suited values, we are using various Meta-heuristic optimization algorithms like JADE Cuckoo Search, Differential Evolution etc.

6. Experimental results

To investigate the effectiveness of proposed optimization based adaptive thresholding approach and the other image denoising methods, Gaussian noise corrupted images are considered and compared with the original ones. The wavelet transform used in all the methods is *Db*4 at one level of decomposition. To evaluate Gaussian denoising methods, this noise is added to the different satellite images with different noise variances such as $\sigma{=}10$, $\sigma{=}20$ and $\sigma{=}30$. Table 1 represent the parameters used for DE, PSO, WDO, FA, CS and JADE algorithms.

To analyze the robustness of proposed approach, numerous satellite images with different features, shown in Fig. 5(a1–a8), have been used for experimentation whereas Fig. 5(b1–b8) shows the corresponding noisy images with $\sigma{=}10$, $\sigma{=}20$ and $\sigma{=}30$ Gaussian noise. There are several methods, which have been used for satellite image denoising. In this paper, various conventional

Table 3
MSE result comparison for various test images and different noise variance (σ) values with classical filtering techniques, different optimization and proposed **JADE** algorithm based adaptive thresholding function.

Test images	Σ	**MSE**									
		Weiner	Median	Bayes [40]	Soft threshold	DE	PSO	WDO	FA	CS	**JADE**
1	10	33.8609	35.6239	20.6101	26.8059	19.7052	21.4909	20.4578	19.8030	19.4100	**15.6716**
	20	40.0506	43.9898	37.7526	48.7117	34.9574	36.7599	34.7366	36.0224	35.1834	**24.3441**
	30	47.7680	52.5722	49.2738	64.9156	46.7322	48.6103	48.0557	47.4017	46.9155	**44.6255**
2	10	42.0861	48.9059	29.7857	37.1774	27.6682	21.4909	28.4146	27.9434	26.7987	**13.2077**
	20	47.6269	54.1683	47.8987	54.3414	46.1563	45.9032	45.2552	46.5274	44.4423	**33.6660**
	30	53.4535	60.1242	57.5998	68.1730	56.8546	56.1315	54.6130	56.9650	54.4321	**37.7498**
3	10	41.6730	44.3457	24.7886	29.6495	23.7642	24.3680	25.1584	23.7288	34.4662	**14.0201**
	20	46.8985	51.6437	41.9944	49.6111	40.9318	41.4481	41.8391	41.3389	41.0762	**24.1997**
	30	52.0908	58.7838	52.3658	65.3147	51.3477	51.9302	50.8847	52.2199	51.6664	**33.8191**
4	10	51.3058	54.0499	30.8861	39.6198	27.9828	28.6045	25.9742	28.1195	27.6806	**11.1935**
	20	55.4898	59.9939	48.7130	55.3007	47.2491	47.3888	47.8623	47.8873	46.9741	**30.9549**
	30	59.5502	65.9081	57.7845	68.3425	57.5861	57.8279	56.3631	57.0916	57.0961	**40.4601**
5	10	25.8236	28.9913	19.3355	22.7197	17.3277	17.7716	16.4911	17.5549	17.2442	**5.44940**
	20	32.7879	37.3532	31.7335	46.1593	30.4886	31.2016	27.6548	30.4767	29.8604	**24.3944**
	30	41.7828	46.3167	41.7211	65.0505	40.5673	41.4940	41.3863	41.7226	39.8355	**39.7515**
6	10	30.8628	33.6092	21.5597	26.1381	20.8651	21.6647	21.7507	21.6865	20.8507	**4.20530**
	20	37.2873	42.3138	36.7458	47.6139	35.8092	36.4195	34.6611	36.5811	34.9396	**13.4573**
	30	47.4965	51.1750	46.4584	64.9012	45.1960	46.2265	45.5172	46.3258	46.0139	**22.1351**
7	10	26.4470	30.2023	18.5503	21.9161	17.6500	18.0448	17.5109	17.3798	17.0974	**2.16200**
	20	32.8656	38.4524	31.3175	45.4819	30.9678	31.6689	31.5045	31.2626	30.6098	**23.5171**
	30	44.1443	46.8579	43.4487	63.0014	41.6405	43.5008	39.5013	41.4802	40.8648	**38.0054**
8	10	62.1745	62.6497	31.8374	42.6774	29.5528	30.9384	27.2447	29.9919	28.9806	**15.1335**
	20	66.1263	68.2622	51.8592	57.9065	50.1053	50.9608	47.6960	50.8653	50.0298	**28.5279**
	30	68.9953	73.2419	62.9825	70.3327	61.1592	61.2090	62.1719	61.2462	61.2109	**46.1561**

classical denoising techniques such as Weiner filter, Median filter, Bayes shrink and soft threshold methods have been used for comparison purposes. In order to show the effectiveness of proposed method over other optimization algorithms, five different robust and well known techniques such as DE, PSO, WDO, FA and CS are used for comparison. To investigate the effectiveness of noise suppression; commonly used performance indices like PSNR, MSE have been used, and to evaluate edge preservation, SSIM and FSIM has been used.

For Gaussian noise suppression, the proposed optimized thresholding function based image denoising methodology is compared with the four classic methods such as Weiner filter, Median filter, Bayes shrink and soft threshold with Gaussian random noise of standard deviation 10, 20 and 30. Then, the proposed JADE algorithm based method is compared with some other optimization algorithm DE, PSO, WDO, FA and CS based image denoising methods with three shape adaptive thresholding function.

In order to investigate the efficiency of proposed thresholding function, well known denoising techniques such as Weiner filter, Median filter, Bayes shrink and soft threshold methods are exploited. Figs. 6 and 7 show the denoised images with $\boldsymbol{\sigma=20}$ Gaussian noise using classic Weiner filter, Median filter, Bayes shrink and soft threshold methods besides the proposed JADE and other optimization algorithm such as DE, PSO, WDO, FA and CS based denoised images. On the other hand, Tables 2–6 show the PSNR, MSE, SSIM, FSIM and CPU time results respectively with $\boldsymbol{\sigma=10}$, $\boldsymbol{\sigma=20}$, $\boldsymbol{\sigma=30}$ Gaussian noise using classic Weiner filter, Median filter, Bayes shrink and soft threshold methods besides the proposed JADE and other optimization algorithm such as DE, PSO, WDO, FA and CS based denoised images.

In this paper, to get better denoised result with superior quality of features, the proposed thresholding function is used in subband-adaptive scenario using optimization method. In the optimization based approaches, only the threshold value is learnt and shape tuning factors (***x, thr, m, n, k***) of function are kept fixed. Figs. 8 and 9 show the corresponding conversion rate with respect to MSE for each optimization based denosing methods (DE, PSO, WDO, FA and CS) along with the proposed JADE algorithm based approach with $\sigma=20$ in the proposed method. Most importantly, Fig. 10 presents the denoised images for the test image 1 with $\boldsymbol{\sigma=10}$, $\boldsymbol{\sigma=20}$, $\boldsymbol{\sigma=30}$ Gaussian noise which demonstrate the complete visual comparison of the proposed methodology with Weiner filter, Median filter, Bayes shrink, soft threshold, DE, PSO, WDO, FA and CS based denoising methods.

The proposed optimized thresholding function has produced better results in Gaussian noise reduction compared to other methods. The denoised images of Weiner filter and Median filter are poorly affected by the over smoothing and blurring with $\boldsymbol{\sigma=10}$, $\boldsymbol{\sigma=20}$, $\boldsymbol{\sigma=30}$ Gaussian noise; and due to these artifacts, the available features in the image are lost. In case of Bayes shrink and soft threshold methods, the denoised results are found to be good with $\boldsymbol{\sigma=10}$ Gaussian noise; but at high ($\boldsymbol{\sigma=20}$, $\boldsymbol{\sigma=30}$**)** Gaussian noise, the final denoised images are distorted. In this paper, to show the preserved edges and features of the denoised image, two dedicated fidelity criteria are considered known as SSIM and FSIM. The SSIM and FSIM are reported in Tables 4 and 5, which indicates the feature preserving capability of the proposed methodology. These two assessment parameters demonstrate the drastic improvement of features, whereas minimum MSE and high PSNR values indicate the overall quality and strength of the final denoised images.

Both the quantitative and qualitative results indicate that the proposed method effectively suppresses Gaussian noise without smoothing the important image details at higher nose level. Experiments demonstrates that the new method produces superior results compared to the methods based on the other optimization and results comparable to other well-known denoising methods.

The cost function which we have defined is the Mean square error risk between the original image and the estimated image. This cost function is very suitable for use for our optimization

Table 4
SSIM result comparison for various test images and different noise variance (σ) values with classical filtering techniques, different optimization and proposed **JADE** algorithm based adaptive thresholding function.

Test images	σ	**SSIM**									
		Weiner	Median	Bayes [40]	Soft threshold	DE	PSO	WDO	FA	CS	**JADE**
1	10	0.9952	0.9986	0.9988	0.9983	0.9991	0.9991	0.999	0.9991	0.9991	**0.9993**
	20	0.9915	0.9984	0.9925	0.9919	0.9981	0.9981	0.998	0.9981	0.9981	**0.9985**
	30	0.9803	0.9966	0.9851	0.9819	0.9972	0.9971	0.9971	0.9972	0.9972	**0.9976**
2	10	0.9801	0.9685	0.9948	0.9951	0.9986	0.9991	0.9983	0.9985	0.9986	**0.9991**
	20	0.9635	0.9557	0.9839	0.9913	0.9966	0.9965	0.9965	0.9964	0.9966	**0.9969**
	30	0.9516	0.9388	0.9482	0.9861	0.9949	0.995	0.9949	0.9949	0.995	**0.9951**
3	10	0.9458	0.9362	0.9939	0.9974	0.9987	0.9986	0.9985	0.9987	0.9987	**0.9988**
	20	0.9351	0.9388	0.9535	0.9937	0.997	0.9969	0.9969	0.9969	0.997	**0.9972**
	30	0.8755	0.9309	0.9303	0.9803	0.9956	0.9955	0.9955	0.9955	0.9955	**0.9958**
4	10	0.9879	0.9971	0.998	0.9985	0.9985	0.9985	0.9982	0.9985	0.9985	**0.9987**
	20	0.9786	0.9918	0.9943	0.9983	0.9963	0.9962	0.9962	0.9962	0.9963	**0.9964**
	30	0.9686	0.9928	0.9802	0.995	0.9944	0.9944	0.9943	0.9944	0.9955	**0.9958**
5	10	0.9685	0.8636	0.9503	0.9987	0.999	0.999	0.9989	0.999	0.999	**0.9992**
	20	0.9613	0.8471	0.8408	0.9881	0.9982	0.9981	0.9981	0.9982	0.9982	**0.9985**
	30	0.8546	0.8996	0.7716	0.8825	0.997	0.9975	0.9975	0.8451	0.9976	**0.9978**
6	10	0.9841	0.9844	0.9957	0.9965	0.9989	0.9988	0.9987	0.9988	0.9989	**0.9991**
	20	0.9781	0.9751	0.9747	0.9941	0.9976	0.9975	0.9975	0.9976	0.9976	**0.9978**
	30	0.9682	0.9743	0.9478	0.9824	0.9966	0.9965	0.9965	0.9965	0.9965	**0.9969**
7	10	0.9987	0.9982	0.9988	0.999	0.999	0.999	0.9989	0.999	0.9991	**0.9992**
	20	0.998	0.9994	0.9961	0.9988	0.998	0.998	0.998	0.9981	0.9981	**0.9982**
	30	0.9972	0.9952	0.9945	0.9969	0.9972	0.9972	0.9972	0.8784	0.9972	**0.9977**
8	10	0.9605	0.9511	0.9956	0.9983	0.9985	0.9984	0.9985	0.9984	0.9985	**0.9988**
	20	0.9744	0.9416	0.9832	0.9955	0.9959	0.9959	0.9958	0.9959	0.996	**0.996**
	30	0.9715	0.9449	0.9246	0.9891	0.9937	0.9937	0.9935	0.9936	0.9936	**0.9939**

Table 5
FSIM result comparison for various test images and different noise variance (σ) values with classical filtering techniques, different optimization and proposed **JADE** algorithm based adaptive thresholding function.

Test images	σ	**FSIM**									
		Weiner	Median	Bayes [40]	Soft threshold	DE	PSO	WDO	FA	CS	**JADE**
1	10	0.8598	0.8427	0.9374	0.9162	0.9568	0.9565	0.9522	0.9556	0.9563	**0.9427**
	20	0.8532	0.8299	0.8916	0.8369	0.9136	0.9151	0.9093	0.9127	0.9155	**0.9031**
	30	0.8403	0.8114	0.8589	0.764	0.8819	0.8826	0.8826	0.8829	0.8829	**0.8836**
2	10	0.8082	0.7783	0.9229	0.9048	0.9521	0.9565	0.9445	0.9493	0.9526	**0.9356**
	20	0.8098	0.782	0.8616	0.8474	0.8947	0.8943	0.8936	0.8917	0.8954	**0.8832**
	30	0.8099	0.7786	0.8217	0.7884	0.855	0.8576	0.8527	0.8545	0.8567	**0.8595**
3	10	0.8064	0.7916	0.9312	0.9189	0.9549	0.9542	0.9468	0.9535	0.9537	**0.9554**
	20	0.8073	0.7876	0.8706	0.8618	0.9022	0.9022	0.8989	0.9001	0.9013	**0.9028**
	30	0.808	0.7839	0.8311	0.8026	0.8653	0.8631	0.8617	0.8654	0.8641	**0.8673**
4	10	0.7857	0.7691	0.9287	0.9104	0.958	0.958	0.9515	0.9573	0.9575	**0.9584**
	20	0.7905	0.7703	0.8701	0.8627	0.9023	0.902	0.902	0.9028	0.9039	**0.9045**
	30	0.7939	0.7709	0.8281	0.8126	0.8653	0.8672	0.8646	0.8646	0.8652	**0.8689**
5	10	0.8129	0.804	0.9035	0.9048	0.9265	0.9303	0.9175	0.9266	0.9266	**0.9267**
	20	0.826	0.8105	0.8382	0.8155	0.8698	0.8689	0.8689	0.8712	0.8724	**0.8762**
	30	0.824	0.8051	0.7854	0.7288	0.8483	0.8457	0.8443	0.8451	0.8451	**0.8457**
6	10	0.8321	0.8106	0.9241	0.9122	0.9431	0.945	0.9406	0.9411	0.944	**0.9481**
	20	0.8344	0.8076	0.8688	0.8428	0.8882	0.8899	0.8861	0.8886	0.8879	**0.8922**
	30	0.8284	0.8007	0.8298	0.7716	0.8552	0.8571	0.8566	0.8566	0.8544	**0.8568**
7	10	0.8607	0.8472	0.929	0.9168	0.9485	0.9496	0.9432	0.9472	0.949	**0.9493**
	20	0.8602	0.8389	0.8803	0.8327	0.9023	0.9037	0.9017	0.9039	0.9025	**0.9049**
	30	0.8505	0.8264	0.8476	0.7533	0.8775	0.8758	0.8772	0.8784	0.8783	**0.879**
8	10	0.7889	0.7752	0.9412	0.9205	0.9691	0.9698	0.9655	0.9686	0.9695	**0.9698**
	20	0.7902	0.7717	0.887	0.8809	0.9242	0.9257	0.9219	0.925	0.9251	**0.9256**
	30	0.7904	0.7647	0.8496	0.8378	0.8894	0.8936	0.8899	0.889	0.89	**0.892**

algorithm as there exist a Statistical technique called Stein's unbiased risk Estimator (SURE) which is an unbiased estimator of the mean-squared error of a nearly arbitrary, nonlinear biased estimator.

$$SURE(h) = d\sigma^2 + \|g(x)\|^2 + 2\sigma^2 \sum_{i=1}^{d} \frac{\partial}{\partial x_i} g_i(x) \tag{44}$$

In Eq. (44), σ^2 is the variance and mu is the mean of the noisy image, d is the number of pixels in an image, $g_i(x)$ is the ith component of the function $g(x)$ (image) and ||.|| is the Euclidean norm.

The importance of SURE is that it is an unbiased estimate of the mean-squared error (or squared error risk) of $h(x)$ (estimated denoised image) i.e.

$$E_\mu\{SURE(h)\} = MSE(h) \tag{45}$$

Table 6
CPU Time comparison for various test images and different noise variance (σ) values with classical filtering techniques, different optimization and proposed **JADE** algorithm based adaptive thresholding function.

Test Images	σ	**CPU TIME**									
		Weiner	Median	Bayes [40]	Soft threshold	DE	PSO	WDO	FA	CS	**JADE**
1	10	0.692784	0.187907	0.982341	0.607182	899.4933	870.2179	106.2316	916.0361	1206.838	636.5686
	20	0.743468	0.17546	0.986429	0.576398	893.3093	927.3882	100.6611	881.3291	1295.6856	886.6048
	30	0.764474	0.165952	1.009205	0.57295	903.307	908.5422	82.6689	902.3618	1748.099	926.6066
2	10	0.779874	0.170531	0.996654	0.626243	911.8942	870.2179	116.1871	911.5593	1254.3151	646.5569
	20	0.723023	0.179115	1.009097	0.578435	656.208	924.5459	105.0162	619.123	1239.7113	661.2742
	30	0.707449	0.179579	0.952206	0.579171	888.7511	637.1325	121.2399	637.7071	1768.4288	943.7039
3	10	0.765562	0.17719	1.008721	0.617951	882.1061	878.7669	128.7295	652.9663	1242.067	659.2825
	20	0.70882	0.194342	1.006854	0.585681	886.8923	888.4022	82.23149	863.5765	1738.9881	636.5267
	30	0.746033	0.160666	1.000949	0.543674	649.4997	885.9046	118.8511	616.5182	1699.1206	915.8764
4	10	0.697499	0.168277	0.974318	0.579352	926.2485	880.1201	128.8659	678.0716	1243.0733	894.1175
	20	0.675285	0.162689	0.963448	0.573525	948.9743	930.2213	109.9766	891.7059	1793.0306	666.8474
	30	0.643251	0.162295	0.963824	0.584728	676.1184	897.7797	83.87221	872.2186	1735.983	671.8323
5	10	0.70988	0.17524	0.938624	0.586604	648.0865	835.762	90.64159	661.3702	1661.3812	879.5087
	20	0.691129	0.181147	0.920292	0.579681	644.0929	903.7071	103.1967	628.1854	1634.9425	912.1832
	30	0.700508	0.166603	0.972729	0.571403	861.5219	927.8554	104.3867	608.7738	1136.3861	649.145
6	10	0.714711	0.169498	0.97456	0.5971	942.3628	884.0189	133.6923	910.4236	1728.7048	901.6794
	20	0.702783	0.167182	0.964966	0.600535	932.0979	641.6533	84.82182	937.7443	1687.2844	656.4891
	30	0.697243	0.166616	1.001217	0.677412	885.2152	920.6708	126.5071	874.3157	1266.6693	684.9808
7	10	0.687356	0.168775	0.994768	0.62918	641.6394	620.6255	110.1259	667.7944	1183.1514	638.441
	20	0.687856	0.159701	0.967738	0.609024	870.3114	904.4827	122.5203	614.8468	1142.9044	681.2647
	30	0.684347	0.174034	0.993865	0.60384	614.373	604.2582	100.7066	630.6488	1214.0186	859.3658
8	10	0.712688	0.163881	0.975456	0.610312	892.1928	882.5932	111.7647	672.6176	1195.6923	659.344
	20	0.724784	0.172328	0.976588	0.575127	903.7654	905.51	82.84719	882.1714	1670.355	680.4376
	30	0.699146	0.16818	0.96338	0.572535	673.417	903.3006	121.4759	894.2541	1158.3873	952.9065

With

$$MSE(h) = E_{\mu}\|h(x) - \mu\|^2 \tag{46}$$

Thus, minimizing SURE can act as a surrogate for minimizing the MSE. So, instead of using SURE risk estimator researchers are directly using mean square error to show the superiority of their techniques. Also, using SURE itself is a different area of Statistical research and our main focus of this paper is to show the effectiveness of our proposed denoising algorithm with the help of Meta-heuristic optimization Algorithm JADE. Our paper also acts as a study of various evolutionary Algorithms for the optimum use of a four parameter shrinkage Function. It's important to keep in mind that SURE is **not** a surrogate for MSE. What is meant is that **minimizing** SURE is a surrogate for **minimizing** MSE. When choosing a statistical estimator, we often want the one that will minimize MSE, but we cannot compute MSE without knowing the true parameters. SURE gives us an unbiased estimate of what this MSE is without needing to know the true parameters. Also find below the research papers that have used MSE risk as cost function. Sir, please add references of papers using MSE as cost function.

While the learning method that you proposed with the use of first forming a data set of required parameters with the help of a learning system (like Neural nets, Support vector Machine etc.) with N fold evaluation would itself require the training data to be created by repetitive run of our proposed technique with optimization algorithm(JADE) and then obtaining the parameters which would constitute the training data for a machine learning system. But the issue with this model is that the training data we obtained is only valid for a particular image as our shrinkage function is adaptive and for every run of our proposed algorithm, the training data (parameters of shrinkage function) are obtained in accordance with that particular noisy image. Hence, your suggestion is very much helpful for a shrinkage function which is universal so that by training a learning model with an appropriate training data (N groups of parameters of universal function) with N-1 groups for training and one disjoint group for cross-validating or/and testing would give us optimized weights(neural nets)/thetas so that any noisy image that is passed through the leaning model will reduce the noise effectively thereby preserving the important details as boundaries, edges and other image features.

6.1. *Optimization technique*

The main aim of the proposed methodology is to preserve the available features and edges in the denoised results. Therefore, the optimization techniques are employed to get minimum MSE as an objective function for computing the best thresholded parameters (***x, thr, m, n, k***) of the adaptive thresholding function. For this purpose, the present paper proposes to utilize relatively recent category of Swarm and evolutionary algorithms (DE, PSO, WDO, FA, CS and JADE), where JADE algorithm based result was found to be superior to other optimization based methods. As an algorithm, the main strength of JADE algorithm is its fast convergence and easy implementation.

6.2. *Subband-adaptive method*

After investigating the superiority of proposed optimization based adaptive thresholding function, results of utilizing the function in optimization manner show that this method has produced better smooth images with highly preserved edges and features which reveals that the proposed method can be used as an efficient denoising method in Gaussian noise reduction.

6.3. *Threshold learning*

Threshold learning is a conventional method of image denoising and applying this method in an optimized adaptive thresholding function yields it as a more effective approach in image denoising.

6.4. *Shape parameter learning*

The three shape learning objective functions can be called as improved *Visu Shrink*; in which the proposed optimization based

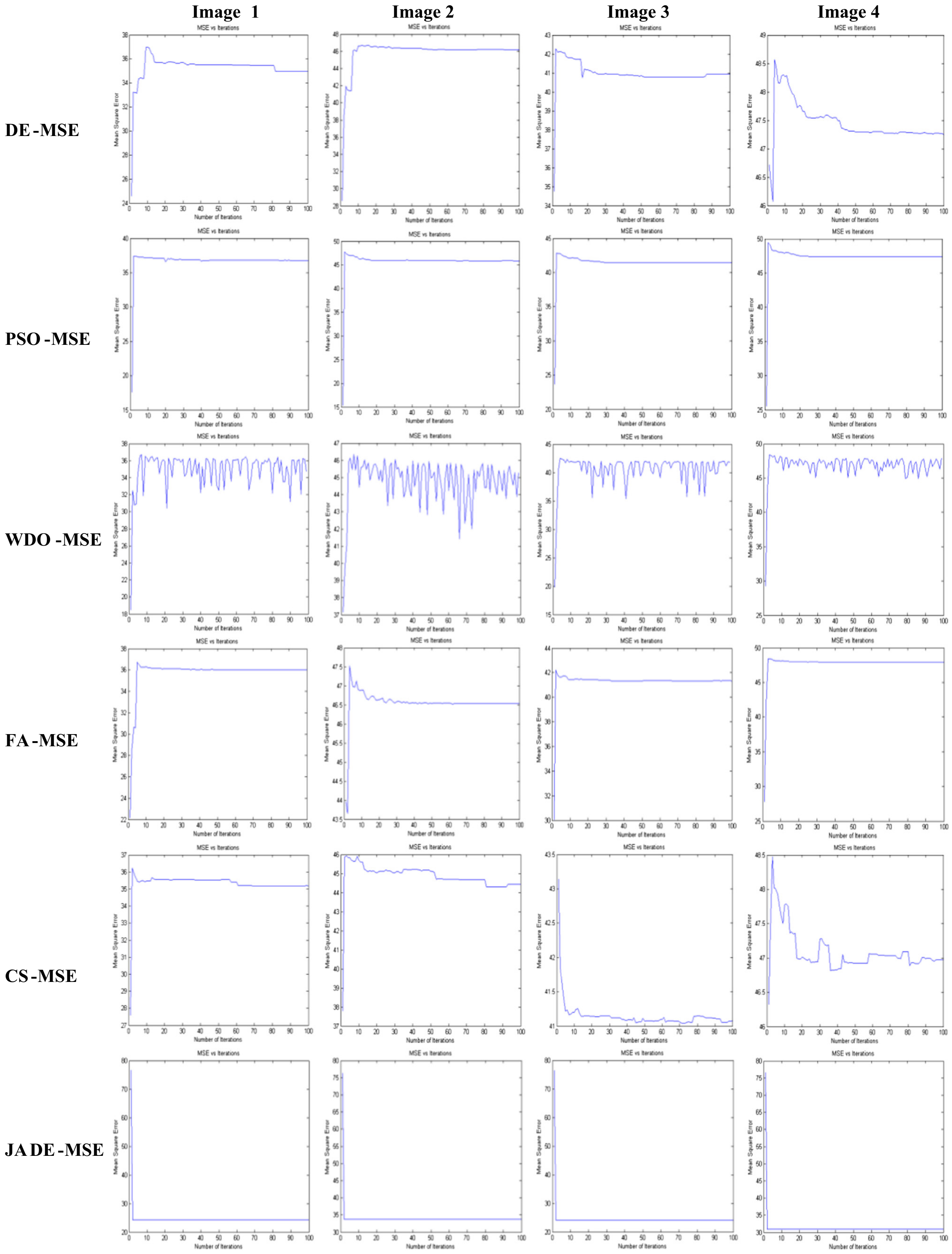

Fig. 8. Comparative performance of the **MSE convergence rate** for DE, PSO, WDO, FA, CS algorithms and proposed **JADE** algorithm based adaptive thresholding function results with σ=**20** Gaussian denoising for test image 1, test image 2, test image 3 and test image 4.

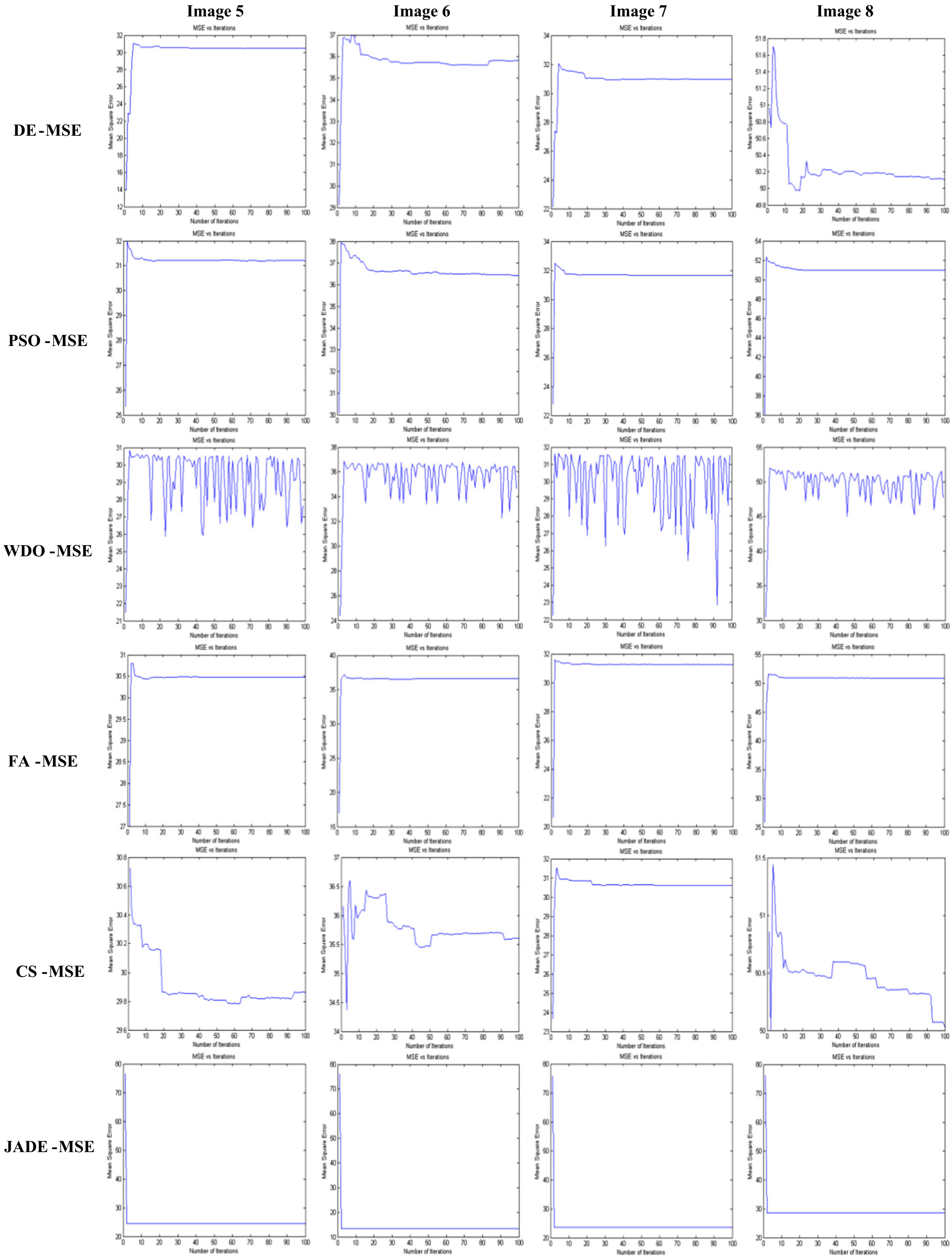

Fig. 9. Comparative performance of the **MSE convergence rate** for DE, PSO, WDO, FA, CS algorithms and proposed **JADE** algorithm based adaptive thresholding function results with $\sigma = \mathbf{20}$ Gaussian denoising for test image 5, test image 6, test image 7 and test image 8.

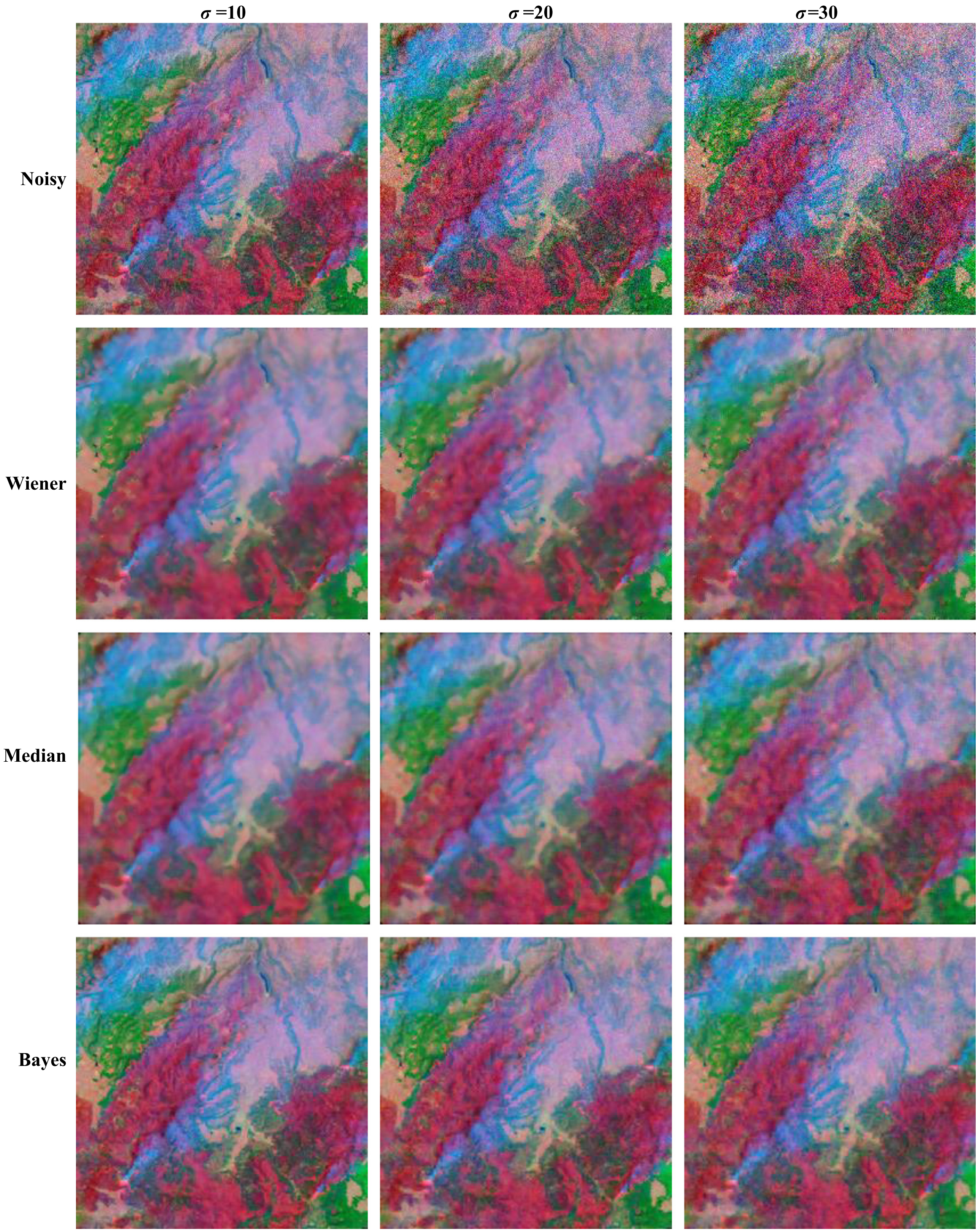

Fig. 10. Comparison of different classic filters (Weiner, Median, Bayes and Soft Threshold) images, DE, PSO, WDO, FA, CS and proposed **JADE** algorithm based adaptive thresholding function results with $\sigma = \mathbf{10}$, $\sigma = \mathbf{20}$ and $\sigma = \mathbf{30}$ Gaussian denoising for test image 1.

approach depends on learning the parameter k in the fixed Visu Shrink threshold values that is used to produce powerful results. Here, the role of optimization technique is very important because it produces the optimized (best) values of the three shape learning parameters which are responsible to suppress noise from the image.

7. Conclusion

In this paper, an exhaustive study for satellite image denoising has been presented. A remarkable point is that Gaussian noise reduction at higher noise variance for edge preserve subjects is

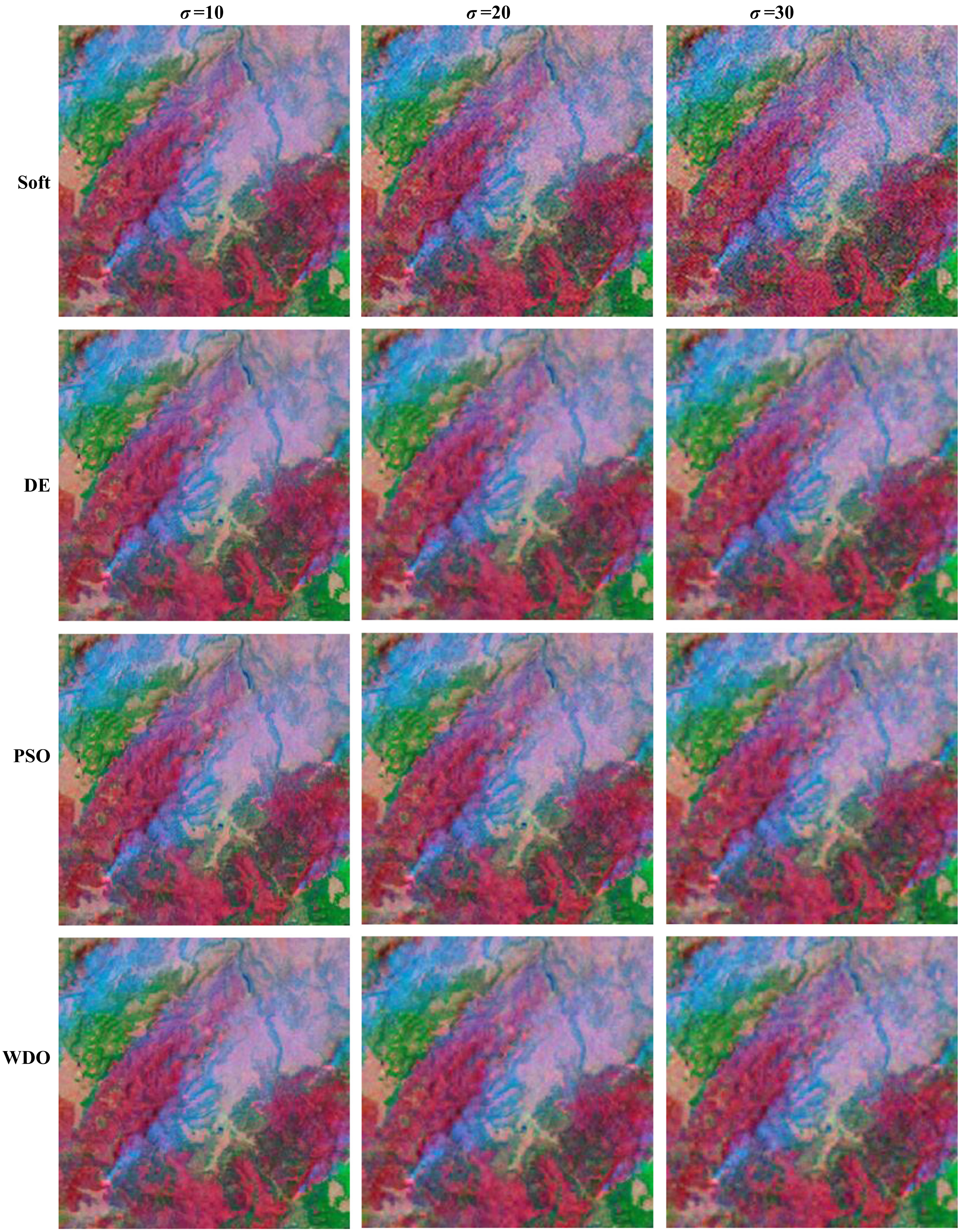

Fig. 10. (*continued*)

considered together in this paper. In this study, different classic (Weiner, Median, Bayes and Soft Threshold) methods and well known optimization (DE, PSO, WDO, FA, CS and JADE) algorithms have been employed to optimize the three shape adaptive thresholding function to compute the feature preserve denoised image. When the noise variance is less, the classic methods give satisfactory results; but at high noise variance level, these methods are failed to provide better denoised output. Therefore, in this

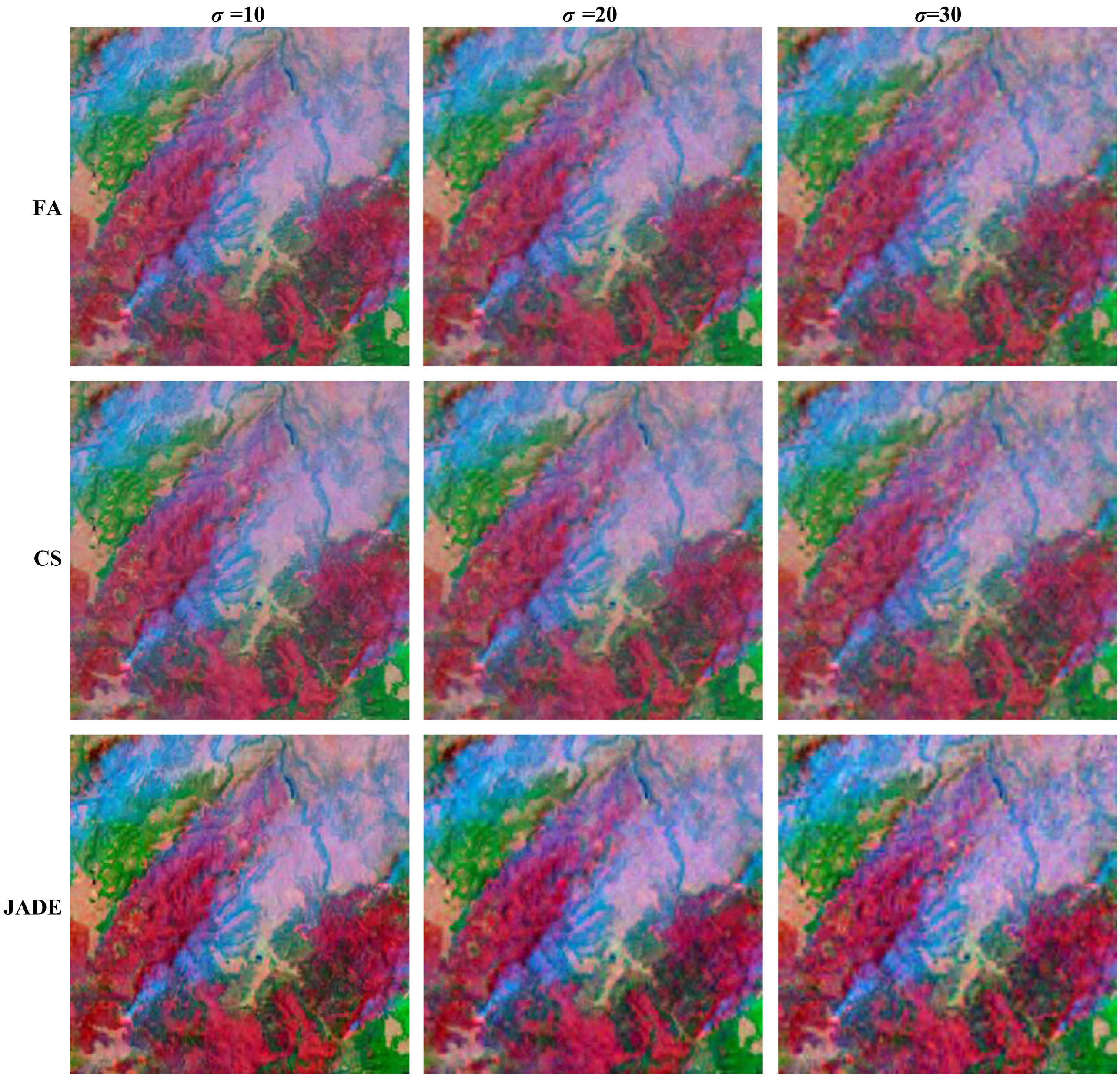

Fig. 10. (*continued*)

paper, feature preserved satellite image denoising using optimized adaptive thresholding function based on adaptive differential evolution algorithms has been proposed. A remarkable point is that this method is highly effective where edge preservation is the basic need such as in case of satellite imaging application. It was found that the proposed JADE algorithm based denoising approach has superior features and give better performance in terms of PSNR, MSE, SSIM and FSIM as compared to other methods. The experimental results show that the performance of JADE based denoising approach is obviously better than the other algorithms with the increase of Gaussian noise variance. This makes it an efficient method in satellite image denoising applications. Especially, the proposed method can preserve the edges very well while removing the high noise levels. The denoised results of JADE with adaptive thresholding function are promising and prompt further research for applying these concepts to complex and real-time image analysis problems such as object classification, automatic target recognition and color image processing. Utilizing other objective functions with JADE algorithm or apply some new optimization techniques can be another topic for future work and may improve the efficiency of the method.

Acknowledgment

We would like to thank the reviewers very much whose comments and suggestions will greatly improve this manuscript.

References

[1] C. Yu, X. Chen, Remote sensing image denoising application by generalized morphological component analysis, Int. J. Appl. Earth Obs. Geoinf. 33 (2014) 83–97.

[2] V. Soni, A.K. Bhandari, A. Kumar, G.K. Singh, Improved sub-band adaptive thresholding function for denoising of satellite image based on evolutionary algorithms, IET Signal Process. **7** (8) (2013) 720–730.

[3] A. De Decker, D. François, M. Verleysen, J.A. Lee, Mode estimation in high-dimensional spaces with flat-top kernels: Application to image denoising, Neurocomputing 74 (9) (2011) 1402–1410.

[4] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (11) (2003) 1338–1351.

[5] F. Luisier, T. Blu, M. Unser, A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding, IEEE Trans. Image Process. 16 (3) (2007) 593–606.

[6] F. Liu, J. Liu, Anisotropic diffusion for image denoising based on diffusion tensors, J. Vis. Commun. Image Represent. 23 (3) (2012) 516–521.

[7] X. Lu, Y. Yuan, P. Yan, Sparse coding for image denoising using spike and slab prior, Neurocomputing 106 (2013) 12–20.

[8] A. Jaiswal, J. Upadhyay, A. Somkuwar, Image denoising and quality measurements by using filtering and wavelet based techniques, AEU – Int. J. Electron. Commun. 68 (8) (2014) 699–705.

[9] A.K. Bhandari, A. Kumar, P.K. Padhy, Enhancement of low contrast satellite images using discrete cosine transform and singular value decomposition, World Acad. Sci. Eng. Technol. (2011) 35–41.

[10] A. Kumar, A.K. Bhandari, P. Padhy, Improved normalised difference vegetation index method based on discrete cosine transform and singular value decomposition for satellite image processing, IET Signal Process. 6 (7) (2012) 617–625.

[11] A.K. Bhandari, A. Kumar, G.K. Singh, Feature extraction using Normalized Difference Vegetation Index (NDVI): A case study of Jabalpur city, Procedia Technol. 6 (2012) 612–621.

[12] M.A. Figueiredo, R.D. Nowak, Wavelet-based image estimation: an empirical Bayes approach using Jeffrey's noninformative prior, IEEE Trans. Image Process. 10 (9) (2001) 1322–1331.

[13] S.G. Chang, B. Yu, M. Vetterli, Spatially adaptive wavelet thresholding with context modeling for image denoising, IEEE Trans. Image Process. 9 (9) (2000) 1522–1531.

[14] J.L. Starck, E.J. Candès, D.L. Donoho, The curvelet transform for image denoising, IEEE Trans. Image Process. **11** (6) (2002) 670–684.

[15] A. Buades, B. Coll, J.M. Morel. A non-local algorithm for image denoising, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), 2005, 2, pp. 60–65.

[16] Y.L. Liu, J. Wang, X. Chen, Y.W. Guo, Q.S. Peng, A robust and fast non-local means algorithm for image denoising, J. Comput. Sci. Technol. 23 (2) (2008) 270–279.

[17] L. Sendur, I.W. Selesnick, Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency, IEEE Trans. Signal. Process. 50 (11) (2002) 2744–2756.

[18] M. Elad, M. Aharon, 'Image denoising via sparse and redundant representations over learned dictionaries', IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.

[19] F. Abramovich, T. Sapatinas, B.W. Silverman, Wavelet thresholding via a Bayesian approach, J. R. Stat. Soc. Ser. B (Stat. Methodol.) 60 (4) (1998) 725–749.

[20] A. Pizurica, W. Philips, I. Lemahieu, M. Acheroy, A joint inter-and intrascale statistical model for Bayesian wavelet based image denoising, IEEE Trans. Image Process. 11 (5) (2002) 545–557.

[21] A. Achim, P. Tsakalides, A. Bezerianos, SAR image denoising via Bayesian wavelet shrinkage based on heavy-tailed modeling, IEEE Trans. Geosci. Remote. Sens. 41 (8) (2003) 1773–1784.

[22] P. Chatterjee, P. Milanfar, Clustering-based denoising with locally learned dictionaries, IEEE Trans. Image Process. 18 (7) (2009) 1438–1451.

[23] X.Y. Wang, H.Y. Yang, Z.K. Fu, A New wavelet-based image denoising using undecimated discrete wavelet transform and least squares support vector machine, Expert. Syst. Appl. 37 (10) (2010) 7040–7049.

[24] H. Cheng, J.W. Tian, J. Liu, Q.Z. Yu, Wavelet domain image denoising via support vector regression, Electron. Lett. 40 (23) (2004) 1479–1481.

[25] D. Li, Support vector regression based image denoising, Image Vis. Comput. 27 (6) (2009) 623–627.

[26] J. Jin, B. Yang, K. Liang, X. Wang, General image denoising framework based on compressive sensing theory, Comput. Graph. 38 (2014) 382–391.

[27] M. Zhang, B.K. Gunturk, Multiresolution bilateral filtering for image denoising, IEEE Trans. Image Process. 17 (12) (2008) 2324–2333.

[28] H. Yu, L. Zhao, H. Wang, Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain, IEEE Trans. Image Process. 18 (10) (2009) 2364–2369.

[29] H.Y. Gao, Wavelet shrinkage denoising using the non-negative garrote, J. Comput. Graph. Stat. 7 (4) (1998) 469–488.

[30] Y. Zhang, X. Tian, P. Ren, An adaptive bilateral filter based framework for image denoising, Neurocomputing 140 (2014) 299–316.

[31] Y. Dong, S. Xu, A new directional weighted median filter for removal of random-valued impulse noise, IEEE Signal Process. Lett. 14 (3) (2007) 193–196.

[32] T.C. Lin, A new adaptive center weighted median filter for suppressing impulsive noise in images, Inf. Sci. 177 (4) (2007) 1073–1087.

[33] R.H. Chan, C.W. Ho, M. Nikolova, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, IEEE Trans. Image Process. 14 (10) (2005) 1479–1485.

[34] R. Garnett, T. Huegerich, C. Chui, W. He, A universal noise removal algorithm with an impulse detector, IEEE Trans. Image Process. 14 (11) (2005) 1747–1754.

[35] C.H. Lin, J.S. Tsai, C.T. Chiu, Switching bilateral filter with a texture/noise detector for universal noise removal, IEEE Trans. Image Process. 19 (9) (2010) 2307–2320.

[36] L. Jin, C. Xiong, H. Liu, Improved bilateral filter for suppressing mixed noise in color images, Digit. Signal Process. 22 (6) (2012) 903–912.

[37] A. Achim, A. Bezerianos, P. Tsakalides, Novel Bayesian multiscale method for speckle removal in medical ultrasound images, IEEE Trans. Med. Imaging 20 (8) (2001) 772–783.

[38] F. Argenti, L. Alparone, Speckle removal from SAR images in the undecimated wavelet domain, IEEE Trans. Geosci. Remote Sens. 40 (11) (2002) 2363–2374.

[39] H. Xie, L.E. Pierce, F.T. Ulaby, SAR speckle reduction using wavelet denoising and Markov random field modeling, IEEE Trans. Geosci. Remote Sens. 40 (10) (2002) 2196–2212.

[40] S.G. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, IEEE Trans. Image Process. 9 (9) (2000) 1532–1546.

[41] D.L. Donoho, De-noising by soft-thresholding, IEEE Trans. Inf. Theory 41 (3) (1995) 613–627.

[42] M. Nasri, H. Nezamabadi-pour, Image denoising in the wavelet domain using a new adaptive thresholding function, Neurocomputing 72 (4) (2009) 1012–1025.

[43] G.G. Bhutada, R.S. Anand, S.C. Saxena, Edge preserved image enhancement using adaptive fusion of images denoised by wavelet and curvelet transform, Digit. Signal. Process. 21 (1) (2011) 118–130.

[44] G.G. Bhutada, R.S. Anand, S.C. Saxena., Image enhancement by wavelet-based thresholding neural network with adaptive learning rate, IET image Process. 5 (7) (2011) 573–582.

[45] G.G. Bhutada, R.S. Anand, S.C. Saxena, PSO-based learning of sub-band adaptive thresholding function for image denoising, Signal Image Video Process. 6 (1) (2012) 1–7.

[46] X.P. Zhang, M.D. Desai, Adaptive denoising based on SURE risk, IEEE Signal. Process. Lett. 5 (10) (1998) 265–267.

[47] X.P. Zhang, Thresholding neural network for adaptive noise reductio, IEEE Trans. Neural Netw. **12** (3) (2001) 567–584.

[48] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.

[49] E. Cuevas, D. Zaldivar, M. Pérez-Cisneros, A novel multi-threshold segmentation approach based on differential evolution optimization, Expert Syst. Appl. 37 (7) (2010) 5265–5271.

[50] M. Ali, P. Siarry, M. Pant, An efficient differential evolution based algorithm for solving multi-objective optimization problems, Eur. J. Oper. Res. 217 (2) (2012) 404–416.

[51] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization: an overview, Swarm Intell. 1 (1) (2007) 33–57.

[52] A.K. Bhandari, A. Kumar, G.K. Singh, Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions, Expert Syst. Appl. 42 (3) (2015) 1573–1601.

[53] A.K. Bhandari, V. Soni, A. Kumar, G.K. Singh, Artificial Bee Colony-based satellite image contrast and brightness enhancement technique using DWT-SVD, Int. J. Remote Sens. 35 (5) (2014) 1601–1624.

[54] Z. Bayraktar, J.P. Turpin, D.H. Werner, Nature-inspired optimization of high-impedance metasurfaces with ultrasmall interwoven unit cells, IEEE Lett. Antennas Wirel. Propag. 10 (2011) 1563–1566.

[55] Z. Bayraktar, M. Komurcu, J.A. Bossard, D.H. Werner, The wind driven optimization technique and its application in electromagnetic, IEEE Trans. Antennas Propag. 61 (5) (2013) 2745–2757.

[56] A.K. Bhandari, V.K. Singh, A. Kumar, G.K. Singh, Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy, Expert Syst. Appl. 41 (7) (2014) 3538–3560.

[57] X.S. Yang, Firefly algorithm, Levy flights and global optimization, in: Max Bramer, Richard Ellis, Miltos Petridis (Eds.), Research and Development in Intelligent Systems XXVI, Springer, London, 2010, pp. 209–218.

[58] M.H. Horng, R.J. Liou, Multilevel minimum cross entropy threshold selection based on the firefly algorithm, Expert. Syst. Appl. 38 (12) (2011) 14805–14811.

[59] A.K. Bhandari, V. Soni, A. Kumar, G.K. Singh, Cuckoo search algorithm based satellite image contrast and brightness enhancement using DWT–SVD, ISA Trans. 53 (4) (2014) 1286–1296.

[60] X.S. Yang, S. Deb, Cuckoo search via levey flights, in: Proceedings of the World Congress on Nature and Biologically Inspired Computing, NABIC: Coimbatore, 2009, 4, pp. 210– 214.

[61] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[62] 〈http://earthobservatory.nasa.gov/Images/?eocn=topnav&eoci=images〉.

A. K. Bhandari received his B. Tech in Electronics and Communication Engineering from Mahatma Jyotiba Phule Rohilkhand University, Bareilly, India, in 2009 and M. Tech and Ph.D. (Gold Medalist) Degree from Indian Institute of Information Technology Jabalpur, India in 2011 and 2015, respectively. Currently, he is Assistant Professor in the Electronics and Communication Engineering Department, National Institute of Technology Patna, Bihar, (India). His research interests include almost all aspects of image processing with remote sensing image classification, satellite image enhancement, image segmentation, image denoising, image compression and optimization techniques.

D. Kumar is pursuing his Bachelor of Technology degree in Electronics and Communication Engineering at the Indian Institute of Information Technology Jabalpur, India. He is currently in the seventh semester of the program and is expected to graduate in July 2016.

Anil Kumar has received the B.E. from Army Institute of Technology (AIT) Pune, Pune University in Electronics and Telecommunication Engineering and M.Tech and Ph.D. Degree from Indian Institute of Technology Roorkee, India in 2002, 2006 and 2010, respectively. Currently, he is Assistant Professor in the discipline of Electronics and Communication Engineering, PDPM IIITDM, Jabalpur, India. His research interest is design of digital filters and multirate filter bank, multirate signal processing, biomedical signal processing, image processing, and speech processing.

G. K. Singh received the B.Tech. degree from G. B. Pant University of Agriculture and Technology, Pantnagar, India, in 1981, and the Ph.D. degree from Banaras Hindu University, Varanasi, India, in 1991, both in electrical engineering. He worked in industry for nearly five and a half years. Currently, he is Professor in the Electrical Engineering Department, Indian Institute of Technology Roorkee, India. His academic and research interest is design and analysis of electrical machines and biomedical signal processing. He has coordinated a number of research projects sponsored by the CSIR and UGC, Government of India.