Image restoration model using Jaya-Bat optimization-enabled noise prediction map

Premnath S. P1 | J. Arokia Renjit2

1 Department of Electronics and Communication Engineering, Sri Krishna College of Engineering and Technology, Kuniamuthur, Coimbatore 641008, India
2 Department of Computer Science and Engineering, Jeppiaar Engineering College, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai 600 119, India

Abstract
Image restoration approaches are introduced to restore the latent clear images from the degraded images. However, the performance of the existing approaches remains an open problem, which may lead to the further development of advanced image restoration techniques. Therefore, an effective image restoration method is developed for restoring the input image from various noises, like impulse noise and random noise. The generation of pixel map, identification of noisy pixel, and the enhancement of pixel are the three major phases involved in the proposed method. Initially, the noisy pixel map generation is performed from the input image, and then the noisy pixels are identified based on deep convolutional neural network, which is trained by the proposed Jaya-Bat algorithm. The Jaya-Bat algorithm is developed by combing the Jaya optimization algorithm and Bat algorithm. Once the noisy pixels are identified, the pixel enhancement is done using the neuro fuzzy system. The experiment is carried out using Statlog (landsat satellite) dataset, and the developed method achieves the maximal peak signal to noise ratio of 51.03 dB, maximal structural similarity index of 0.848 for the image with random noise, and the maximal second derivative like measure of enhancement 62.96 dB with impulse noise, respectively.

1 INTRODUCTION

The images taken in dim light is unsatisfactory, and noisy while increasing the ISO at short duration. Using the flash light may improve lighting, but still creates unwanted shadow or changing the tone in image [1, 2]. Image restoration plays a very important role in the field of computer vision, graphic design, remote sensing, agricultural buildings [3], thermal analysis [4] and the medical imaging [5]. The image restoration aims to recover the high-quality images from their degraded observation [6, 7]. In the image restoration, the degradation process is generally defined using linear transformation [7]. The degraded images are restored using various methods in signal processing environment. Here, the clean patches are predicted separately [8] or collaborating other same patches using non-local schemes [9]. Then, the computed patches in the image are returned to their raw positions and the overlapped patches get average to obtain the reconstructed image [10]. While denoising the image, the overlapping images are initially decomposed and denoising each patch separately and at last, the result is obtained by simple averaging [11, 12]. In the last few years, several pre-processing restoration approaches for HIS are introduced to achieve optimal results. The previous restoration approaches are broadly categorized into three. They are morphology or filter-based techniques, spectral and spatial methods, and the sparse representation and low-rank (SR-LR)-based methods. The filter-based techniques are the type of most practical and traditional HIS restoration approaches [13]. Non-local techniques are performed based on intrinsic similarities present in image patches. This leads to improve the development of various non-local image restoration approaches introduced in [9, 10, 14–16]. Non-local restoration capabilities are used by most of the research works for grouping similar images in image. In addition, the sparse representations are obtained based on sparsity methods [10, 11, 17].

Gaussian mixture models (GMM) is utilized in several signal processing tasks, like video applications, audio processing [18], image segmentation, and image denoising [19].
Nowadays, GMM is utilized to tackle the recovery issue caused in nonlinear and sparse computation, as GMM is the integration of linear estimation. Piecewise linear estimation (PLE) is another image restoration method employed for image patches. Expected patch log likelihood (EPLL) is introduced in [20] for weight initialization [10]. Introducing global model for the entire image is very hard because of the curse of dimensionality. Hence, several image restoration algorithms are introduced nowadays, for addressing modelling with patch priors [11]. This patch priors are easy to model, readily available and low dimensional. In [11], GMM, sparsity inspired model, and ICA are introduced to identify the same patches. In global GMM image restoration the similar patches are measured by Gaussian probability density function [10, 12].

In this research, the image restoration is carried out using the proposed Jaya-Bat optimization algorithm. The proposed approach involves three phases. The initial phase is generation of pixel map, the secondary phase is the identification of noisy pixel, and the tertiary phase is the enhancement of pixel. At first, the noisy pixel map generation is performed from the input image, and then the noisy pixels are identified based on Deep CNN, which is trained by the proposed optimization algorithm, termed Jaya-Bat algorithm. The Jaya-Bat algorithm is the combination of Jaya optimization algorithm and Bat algorithm. After the identification of noisy pixels, the pixel enhancement is done on the basis of fuzzy neuro system.

The contribution of this research is the development of Jaya-Bat-based Deep CNN for eliminating the impulse and random noise from remote sensing images. The Jaya-Bat-based deep CNN is designed by integrating Jaya optimization algorithm and Bat algorithm. The developed method detects the noise in corrupted images into non-corrupted images.

The paper is structured in the following manner: Section 2 discusses existing methods of image restoration with challenges of the methods that remains the motivation for the research. The proposed method of Jaya-Bat-based Deep CNN is demonstrated in Section 3, and Section 4 provides the results of developed methods. At last, concludes the research work in Section 5.

2 | LITERATURE SURVEY

Several methods related to image restoration are described, and analysed as follows: Papyan and Elad [11] developed multi-scale expected patch log likelihood (EPLL) method that imposes very small patch on various scale patches extracted from the image. This method attained better performance in both visually and quantitatively, however, denoising the noisy image is difficult. Zeng et al. [21] presented Average Image-induced nonlocal means (aviNLM) for image restoration. In this approach, the prior image was generated from the reconstructed image in every energy bin. Here, the performance was found better, but the deformation of tissue is very challenging. Niknejad et al. [10] modelled Gaussian mixture model (GMM) where the neighbourhood similar patches were derived based on multivariate Gaussian probability distribution with covariance and mean. However, zooming is very challenging from the randomly observed pixels. Teng et al. [13] developed adaptive morphological filtering (AMF) and the fusion-enabled restoration method. This method was very flexible enhancement component for local filters, but failed to consider HSI restoration for both pre-processing and post-processing areas.

Zhang and Destrosiers [7] presented an approach for image restoration that integrates the global structure sparsity and non-local self-similarity in the single efficient model. Here, the performance was found better, but developing effective image regularization priors was very difficult. Shen et al. [2] developed multispectral joint image restoration framework for handling all the structure divergence and finding the edges and transitions. It was more robust and to tackle multispectral restoration issues. The method failed to consider other optimization algorithms for improving the system performance. Wang and Tao [1] modelled Non-local auto-encoder that uses self-similar information of natural images for the stability. This method achieved comparable denoising performance to other state-of-the-art methods and outperformed other leading super resolution methods, but the performance of image denoising was found poor. Peng and Cosman [22] developed depth estimation approach for underwater scenes using light absorption and image blurriness. This method estimates underwater depth scene accurately, but the variation of physical properties was difficult for restoring underwater images. Asif et al. [23] developed a deep learning (DL)-based image evaluation method (W-LDMM) for the effective image restoration. This method enhanced the quality of the endoscopy images used to examine internal body organs. Anyway, the performance of this method was evaluated using small dataset. Wang et al. [24] developed a constrained iterative support shrinking algorithm with proximal linearization (C-ISSAPL) for enhancing the image restoration. This method had maximum convergence speed and easy to implement. Also, this method was applicable for natural image restoration and piecewise constant. This method was required more processing time. However, in this method, the calculation of box constraint was difficult.

3 | PROPOSED JAYA-BAT ALGORITHM-BASED DEEP CONVOLUTIONAL NEURAL NETWORK FOR IMAGE RESTORATION

This research primarily focuses to model a new method for image restoration using Jaya-Bat algorithm-based DCNN. Initially, the input image is fed to the noisy pixel map generation stage, where the noisy pixel map is effectively generated. The generated noisy pixel map is subjected to the noisy pixel identification stage, where the noisy pixels are identified based on deep convolutional neural network (deep CNN) classifier, which is trained by the Jaya-Bat optimization algorithm. Here, the proposed Jaya-Bat algorithm is the integration of Jaya optimization algorithm [25], and Bat Inspired algorithm [26]. Finally, the identified noisy pixels are sent to the pixel enhancement stage, where the enhancement of the pixel is carried out using the
Figure 1: Block diagram of the proposed Jaya-Bat algorithm for image restoration

neuro fuzzy system. Figure 1 depicts the schematic view of the developed model for image restoration.

Assume the input image \( J \) with size \( U \times V \), and the pixel location of the input image is represented as \( J(x, y) \). Due to the noise present in image, certain part of the image is corrupted and the remaining pixels are considered as the noise free. The detected noisy pixels are very challenging, hence in this paper, the Deep CNN is introduced to identify the noisy pixel by considering the neighbourhood of the image.

### 3.1 Noisy pixel identification: Constructing the noisy pixel map using the DCNN classifier

This DCNN is utilized for pixel identification. The DCNN is utilized for generating the prediction map for image pixels. The DCNN classifier [27] generates the prediction map based on input image. The architecture of DCNN and the algorithmic steps of the Jaya-Bat algorithm are described below.

#### 3.1.1 Architecture of the DCNN

The basic architecture of the DCNN [27] is discussed in this section with its architecture in Figure 2. The DCNN comprises of the number of convolutional (conv) layers, pooling (POOL). The architecture of the Deep CNN is deliberated in Figure 3 and the architecture of DCNN comprises of three layers, like pooling (POOL), convolutional (conv), and the fully connected (FC) layers. Among three layers of DCNN, each of the layers constitutes specific function. The main function of the conv layers is to generate the feature maps from the segments of the pre-processed image and these feature maps are further subsampled down in the pool layers. The third layer is FC layer, where the classification is progressed. The convolutional layer engages in mapping the input such that the input maps undergo convolution with the convolutional kernels in order to develop the output map. The size of the output map is similar to the kernel number, and the size of the kernel matrix is, \( 3 \times 3 \). Thus, making it clear that the conv layers is the multilayer loop of input maps, kernel weights, and output maps. In the first conv layer,

\[
(M^i)^{ij} = (B^i)^{ij} + \sum_{d=1}^{w-1} \sum_{k=-w}^{w} \sum_{n=-w}^{w} (\theta^i_{cd})_{k,n} \ast (J)_{i+k,j+n}
\]

where, the symbol \( \ast \) indicates the convolutional operator, which enables the pattern extraction from the outputs obtained from the adjacent conv layers, \( (B^i)^{ij} \) indicates the fixed feature maps. The output from previous \( (b-1) \)th layer forms the input to \( i \)th conv layer. The conv layer weight is denoted as \( \theta^i_{cd} \), which is
the weights of $l$th conv layer and the bias of $l$th conv layer is denoted as, $b^l$. Let us consider $d$, $k$, and $u$ as the notations of feature maps.

ReLU layer: ReLU is the activation function for ensuring effectiveness and simplicity and the significance of ReLU layer is that DCNN with ReLU works faster dealing with large networks. The output from the ReLU layer when fed with the feature maps is given as,

$$ G^b_r = A f u \left(M^{-1}_r\right) $$

where, $A f u()$ refers to the activation function in $l$th layer.

POOL layers: It is a non-parametric layer with no weights, and bias, undergoing a fixed operation. The importance of POOL layer is to mitigate the spatial dimensions of the input and minimizes the computational complexity.

FC layers: The patterns generated using the pooling and the conv layers form the input to the fully connected layers that are subjected to the high-level reasoning. The output from the fully-connected layers is given as,

$$ H^b_r = \delta \left(G^b_r\right) \text{ with } M^b_r $$

$$ = \sum \sum \sum \left(\tilde{c}^b_{r,c} \right)_{k,u} \ast \left(f_{l+1,k+u}\right) $$

where, $(\tilde{c}^b_{r,c})_{k,u}$ denotes the weight.

3.1.2 Training of DCNN based on Jaya-Bat algorithm

This section briefly explains the developed Jaya-Bat algorithm for image pixel identification. Here, the proposed Jaya-Bat algorithm is developed by integrating the Jaya optimization algorithm [25] and Bat algorithm [26]. The BA is motivated by echolocation characteristics of microbats. This method is very efficient for generating improved features to solve the multi-objective optimization problems. Moreover, it can solve the highly non-linear problems with complex constraints. Jaya is performed using candidate solutions, and this framework works independently of any parameters. The Jaya algorithm is simpler in operation; because it works in a single phase. Here, the update equation of Bat algorithm is modified using the update equation of Jaya optimization algorithm. The modification makes the solution update to be more efficient, and it further improves the performance of the proposed optimization algorithm. The update equation of Bat is expressed as,

$$ D^j_{r+1} = D^j_r + u^j_r $$

where, the term $D^j_{r+1}$ denotes the $j$th bat position in $(r + 1)$th iteration, $D^j_r$ indicates the $j$th bat position in $r$th iteration, and $u^j_r$ indicates the $j$th bat velocity in $r$th iteration. The velocity of the bat at $r$th iteration is given as,

$$ u^j_r = u^j_{r+1} - \left(D^j_r - D^r\right)p^j $$

(Substitute $u^j_r$ in Equation (4), the solution becomes,

$$ D^j_{r+1} = D^j_r + u^j_{r+1} - \left(D^j_r - D^r\right)p^j $$

where, $D^r$ refer to the best position of the bat, $p^j$ signifies the frequency of the $j$th bat, and $u^j_{r+1}$ indicates the velocity of the bat in $(r + 1)$th iteration. Then, the above Equation (6) is modified using the Jaya for improving the effectiveness of approach and to solve several optimization problems. The standard equation of the Jaya optimization is given by,

$$ D^j_{r+1} = D^j_r + v_1 \left(D^r - \left|D^j_r\right|\right) - v_2 \left(D^\text{worst} - \left|D^j_r\right|\right) $$

where, the term $D^j_{r+1}$ indicates the value of $j$th variable in $(r + 1)$th iteration, $v_1$ and $v_2$ refers to the random numbers ranging from 0 to 1. The worst candidate solution is represented as $D^\text{worst}$.

Let us assume $D^j_r$ is positive,

$$ D^j_{r+1} = D^j_r + v_1 \left(D^r - D^j_r\right) - v_2 \left(D^\text{worst} - D^j_r\right) $$

$$ D^j_{r+1} = D^j_r + v_1 D^r - v_1 D^j_r - v_2 D^\text{worst} + v_2 D^j_r $$

$$ D^r = \frac{1}{v_1} \left[D^j_{r+1} - D^j_r + v_1 D^j_r + v_2 D^\text{worst} - v_2 D^j_r\right] $$

Substituting Equation (10) in Equation (6) the final equation is,

$$ D^j_{r+1} = \frac{u^j_{r+1} + D^j_r (v_1 - (1 + v_2) p^j) + v_2 D^\text{worst} p^j}{v_1 - p^j} $$

where, $p^j = p_{\text{min}} + (p_{\text{max}} - p_{\text{min}}) \alpha$. Here, $\alpha$ denotes the random vector ranging from $(0, 1)$, and $f_{\text{min}}$ is considered as zero and the value of $f_{\text{max}}$ tends to $0(1)$.

The steps followed in the developed algorithm are illustrated below,

Step 1: Initialization: The initial step of the proposed Jaya-Bat algorithm is the initialization of bat population in the search space as,

$$ \text{Bat population}, D^j_r \quad \left(1 \leq j \leq a\right) $$

where, $a$ denotes the total bats, and $D^j_r$ refer to the location of $j$th bat in the search space, and $D \in \tilde{c}^b_{r,c} + \tilde{d}^b_{r,d} + b^b.$

Step 2: Fitness function evaluation: The selection of the optimal location of the bat is performed based on minimization problem. The minimal value of the objective
function describes the better solution and therefore, the solution with the minimum value of the error is chosen as the best solution. The error is determined as,

$$\text{MSE} = \frac{1}{X} \left[ \sum_{i=1}^{X} H_{\text{target}} - H_i^p \right]$$  \hspace{1cm} (13)

where, \(H_{\text{target}}\) and \(H_i^p\) are the estimated and target output of classifier. The term \(X\) denotes the total number of samples.

Step 3: Update the solution based on Jaya-Bat algorithm:
After computing the objective function, the solution undergoes the location update on the basis of Bat algorithm. While updating the solution, there exist two conditions, first based on pulse emission rate and the second based on loudness and the fitness. In the first condition, the random number is compared with the pulse emission rate, and if the random number exceeds the pulse emission rate, the position is updated based on random walk. In the second condition, if the random number is less than the pulse emission rate, then it is compared with the fitness and the loudness of the bat. If the random number exceeds the loudness and fitness exceeds the fitness of best solution, then the new solution is updated using Equation (22). After position update, the loudness decreases and the pulse emission rate increases for successive iterations. The evaluation of loudness and emission rate for the \(j^{th}\) bat is expressed as,

$$D^j_{r+1} = \tau D^j_r$$  \hspace{1cm} (14)

$$F^j_r = F^j_0 [1 - \exp (-\omega j)]$$  \hspace{1cm} (15)

where, \(D^j_r\) represents the loudness in previous iteration, \(F^j_r\) is the emission rate, \(F^j_0\) is the initial emission rate, and \(\tau\) and \(\omega\) are the constants.

Step 4: Ranking of bats based on fitness function: The solutions are ranked based on fitness and the solution with highest fitness measure is chosen as the best solution, \(D^*\).

Step 5: Termination: The process is continued for maximum number of iterations and stopped after the generation of global optimal solution. The pseudocode of the proposed Jaya-Bat algorithm is depicted in Algorithm 1.

**ALGORITHM 1 Pseudocode of proposed Jaya-Bat approach**

Input: Population of bats \(D^j\); \((1 \leq j \leq a)\)

Output: Best position of Bat \(D^*\)

Start

1. **Initialization**
   - For \(\forall i\)
     - Calculate the pulse emission rate and the loudness
   - While (\(i < \) Maximum iterations)
     - Produce new solutions using equation (24)
     - If (Rand > \(F^j\))
       - Update position based on random walk
     - End if
     - Generate new solutions based on random fly
     - If (Rand < \(F^j\) \&\& Fitness\((D^j) < \) Fitness\((D^*)\))
       - Increment pulse emission rate
       - Update the bat position
     - End if
   - Rank the solution based on fitness
   - Estimate the current best solution
   - End While
   - Produce the best solution \(D^*\)

Stop

3.2 Restoration of noisy pixel based on statistical model

After the identification of noisy pixel, the novel pixel values are estimated for corresponding noise pixels. When the noisy centre pixel is detected, the noisy pixel is removed using the below expression.

$$f_{np}(x, y) = \begin{cases} f_x(x, y) & \text{if } M(x, y) = 1 \\ f(x, y) & \text{otherwise} \end{cases}$$  \hspace{1cm} (16)

where, the term \(f_x(x, y)\) refer to the new pixel image value. The selected noisy pixel is only applicable to compute the new pixel value. At first, the \(3 \times 3\) window \(f(u, v)\) is generated, and then the matching is done with the noisy pixel of \(3 \times 3\) window \(f(u, v)\) and the input image \(f(x, y)\). From the matching output, the same matched pixels \(N_x\) are considered for further processing.

When \(N_x\) is higher than threshold value \(T_1\), the novel pixel value is produced or else the noisy pixels are changed by the raw image in the selected \(3 \times 3\) windows. The value of \(T_1\) is considered as 21. Here, the new pixel values are computed based on \(X_{np}, Z,\) and \(B\) parameters, and is expressed below.

$$E(x, y) = e(X_{np}, Z, B)$$  \hspace{1cm} (17)
where, the term $Z$ refer to the absolute function of $3 \times 3$ window based on neighbour pixel and is given below,

$$Z = \text{abs}(f(u,v) - f(u+x-3,v+y-3))$$  \hfill (18)

The value of $X_b$ is derived by the absolute output of Equation (29) with prefixed window. The value of $X_b$ is initially set to zero in which $b$ varies on the basis of selected prefixed window,

$$X_b = X_b + Q_1(x,y) \ast Z; \quad 1 \leq b \leq \omega$$  \hfill (19)

The rounding process is carried out for $X_b$ and is round by $\frac{X_b}{8}$, which is represented as $g$. After rounding, the sorting process is done for the result obtained from $X_b$ where the initial value is selected as $E$.

$$E(x,y) = \begin{cases} 1 - \frac{(E - 1)}{4} & E > 10 \\ 1 & \text{otherwise} \end{cases}$$  \hfill (20)

The result of Equation (31) is compared with $T_2$. Here, $T_2$ value is set to 0.2. When the value of $E(x,y)$ is higher than the second threshold value, the function is produced.

$$RP(x,y) = \begin{cases} \sum p(x) / q_1 & s > T_3 \\ E(x,y) & \text{otherwise} \end{cases}$$  \hfill (21)

where, the term $\alpha$ refer to the mean value of neighbourhool pixel, and the term $q_1$ signifies the constant in which the $q_1$ value is set to 4. The mean value is evaluated using the below expression,

$$\alpha = e^{(Y_a + Y_b)/0.001 + 2\beta^2}$$  \hfill (22)

where, the term $\beta$ signifies the standard deviation. When the output obtained from Equation (32) is compared with threshold value $T_3$. Here, the $T_3$ value is fixed to 23. If it is greater than $T_3$, the new pixel value is generated using the equation given below,

$$J_n(x,y) = \frac{E \cdot P_n}{E \cdot P_h} = \frac{\sum f(x,y) \cdot M(x,y)}{\sum M(x,y)}$$  \hfill (23)

Algorithm 2 depicts the pseudo code of novel pixel computation from noisy pixels.

**ALGORITHM 2 Pseudocode for computing new pixel**

**Input:** $J_n(x,y)$  
**Output:** $J_n(x,y)$  
**Computation of parameter based on Jaya-Bat algorithm**  
**Procedure:**  
**Begin**  
**Selection of $3 \times 3$ window by considering the noise pixel as the centre pixel**

### 3.3 Image enhancement based on neuro fuzzy system

After the identification of new pixels, the image enhancement process is performed based on neuro fuzzy system. The image enhancement process is utilized for modifying input image by changing a visual impact. The enhancement image provides the detailed information about boundaries, edges with increased dynamic range. In this work, the enhancement procedure is carried out on the basis of noisy pixels. Once the noisy pixels position $K$ are identified, that identified pixels $s$ only applied for the enhancement process. After that, the new image matrix is produced on the basis of nearest neighbour for the noisy pixel is expressed by,

$$J_{a1}(x,y) = \frac{1}{5} \sum_{n=1}^{1} \sum_{m=1}^{1} J_n(x + u, y + v)$$  \hfill (24)

In order to solve the uncertainty issues, the generated new pixel values are subjected to the neuro fuzzy system.

$$J_{a2}(x,y) = N_{\text{fuzzy}}(J_n(x,y))$$  \hfill (25)

where, the term $N_{\text{fuzzy}}$ signifies the neuro fuzzy system. Once the neuro fuzzy set model is generated, the new pixel $J_n(x,y)$ is considered as the input and the new image is generated using the noise pixel-based location. At last, the restored image is obtained from the below equation.

$$J_{G}(x,y) = \left[ \frac{J_{a1}(x,y) + J_{a2}(x,y)}{\forall K} \right]$$  \hfill (26)

The structure of ANFIS [28] composed of consequence and premise parts. The ANFIS is trained for determining the parameters related to the parts employed in optimization.
approach. Here, the ANFIS uses output and input data pairs in the training. The layers present in ANFIS are fuzzification, rule, normalization, defuzzification, and summation layer. Here, the fuzzification layer employs the membership functions for obtaining fuzzy clusters from the input values where the firing strengths are produced based on the membership values evaluated in the fuzzification, whereas the normalization layer is utilized to compute the firing strengths corresponding to every rule. The normalized value refers to the ratio of firing strength to ith rule to the total number of firing strengths. The defuzzification layer computes the weighted values for every node, while the summation layer summed the results obtained from the defuzzification layer.

4 | DISCUSSION OF RESULTS

The section demonstrates the results obtained by the developed Jaya-Bat-based DeepCNN model and the performance is evaluated based on the PSNR, SDME, and SSIM.

4.1 | Experimental arrangement

The experimentation of the proposed model is done in MATLAB, operating in the PC with Windows 10 OS and 2Gb RAM.

4.2 | Dataset description

The experimentation is done using Statlog (Landsat satellite) dataset [29], and the description of the datasets is given below:

Statlog (Landsat satellite) Dataset: It is a multivariate dataset with integer attribute characteristic. It contains 36 attributes, 6435 number of instances and 138348 number of web hits. The Landsat satellite data is one of the many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions including multispectral and radar data, maps indicating topography, land use etc. One frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region and two are in the infra-red. Each pixel is an 8-bit binary word, with 0 corresponding to black and 255 to white.

4.3 | Experimental results

Figure 4 depicts experimental results of the proposed Jaya-Bat-based DeepCNN for image restoration. Figure 4(a) depicts the input image 1. Figure 4(b) represents image with the presence of impulse noise, and Figure 4(c) depicts the reconstructed image 1. the input image 2 is depicted in Figure 4(d), and the presence of random noise of image 2 is shown in Figure 4(e), and the reconstructed image 2 is depicted in Figure 4(f).

4.4 | Performance metrics

The metrics utilized for the analysis, such as PSNR, SDME, and SSIM.

4.5 | Comparative techniques

The methods, such as cost-based image reconstruction [30], LOTV-PADMM [31], DBAIN [32], DeepCNN [33], AFSWA [34], T2FCS [6], CNN [27], W-LDMM [23], and C-ISSAPL [24] are utilized for the comparison with the developed Jaya-Bat-based DeepCNN for the analysis.

4.5.1 | Comparative analysis for Impulse noise

Based on image 1

The comparative analysis in terms of PSNR, SDME, and SSIM, and using image 1 with impulse noise is depicted in Figure 5. Figure 5(a) illustrates the analysis based on PSNR by varying the noise level. When the noise level is 0.1, then the corresponding PSNR values computed by existing cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed Jaya-Bat-based DeepCNN are 8.66, 16.55, 29.50, 29.85, 29.86, 43.65, 43.65, 39.38, 45.54, and 49.87 dB, respectively. For the noise level 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the PSNR of 8.63, 16.02, 25.55, 27.99, 29.86, 43.65, 43.65, 39.38, 45.54, and 49.87 dB, respectively. For the noise level 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the PSNR of 8.63, 16.02, 25.55, 27.99, 34.64, 34.65, 35.96, and 39.11 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the PSNR of 48.65 dB.

The comparative analysis in terms of SDME metric is depicted in Figure 5(b). When the compression ratio is 0.2, the SDME values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS,
FIGURE 5  Comparative analysis by varying the noise level using image 1 (a) PSNR, (b) SDME, and (c) SSIM
FIGURE 6 Comparative analysis by varying the noise level using image 2
(a) PSNR, (b) SDME, and (c) SSIM

W-LDMM, C-ISSAPL, and proposed model are 0.043, 0.183, 0.397, 0.398, 0.666, 0.676, 0.711, 0.555, 0.635, and 0.712, respectively. For noise level $\sigma = 0.3$, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SSIM of 0.041, 0.152, 0.396, 0.397, 0.499, 0.640, 0.642, 0.462, and 0.581, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SSIM of 0.664.

Based on image 3

The comparative analysis based on SDME, PSNR, and SSIM using image 3 is shown in Figure 7. Figure 7(a) illustrates the analysis based on PSNR by varying the noise level. When the noise level is 0.1, then the corresponding PSNR values computed by existing cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed Jaya-Bat-based DeepCNN are 12.30, 13.89, 24.06, 24.69, 24.71, 32.72, 33.18, 37.84, 39.68, and 44.74 dB, respectively. For the noise level $\sigma = 0.3$, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the PSNR of 9.97, 13.84, 20.81, 21.19, 21.26, 32.71, 32.85, 37.25, and 36.41 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the PSNR of 44.70 dB.

The comparative analysis in terms of SDME metric is depicted in Figure 7(b). When the compression ratio is 0.2, the SDME values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 1.50, 24.99, 37.33, 39.41, 39.57, 46.71, 46.80, 36.20, 44.17, and 47.70 dB, respectively. For noise level $\sigma = 0.3$, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SDME of 1.48, 23.94, 37.29, 37.87, 38.24, 46.49, 46.65, 35.40, and 42.17 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SDME of 47.38 dB.

The comparative analysis in terms of SSIM metric is depicted in Figure 7(c). When the compression ratio is 0.1, the SSIM values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 0.038, 0.164, 0.323, 0.325, 0.574, 0.589, 0.613, 0.503, 0.541, and 0.614, respectively. For noise level $\sigma = 0.3$, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SSIM of 0.037, 0.136, 0.322, 0.324, 0.425, 0.533, 0.533, 0.427, and 0.433, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SSIM of 0.575.
4.5.2 Comparative analysis for random noise

**Based on image 1**

The comparative analysis based on SDME, PSNR, and SSIM, and using image 1 with random noise is shown in Figure 8. Figure 8(a) illustrates the analysis based on PSNR by varying the noise level. When the noise level is 0.1, then the corresponding PSNR values computed by existing cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed Jaya-Bat-based DeepCNN are 8.65, 15.95, 27.15, 28.71, 28.88, 43.65, 43.65, 42.27, 43.50, and 51.03 dB, respectively. For the noise level 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possess the PSNR of 8.64, 14.81, 19.54, 19.68, 21.10, 43.63, 43.65, 34.77, and 38.53 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the PSNR of 49.61 dB.

The comparative analysis in terms of SDME metric is depicted in Figure 8(b). When the compression ratio is 0.2, the SDME values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 19.02, 38.16, 46.21, 46.31, 46.36, 57.69, 57.71, 54.28, 52.88, and 62.44 dB, respectively. For noise level = 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SDME of 18.89, 37.61, 39.19, 39.23, 43.77, 57.64, 57.69, 50.54, and 51.54 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SDME of 62.06 dB.

The comparative analysis in terms of SSIM metric is depicted in Figure 8(c). When the compression ratio is 0.1, the SSIM values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 0.343, 0.378, 0.620, 0.620, 0.725, 0.815, 0.746, 0.761, and 0.848, respectively. For noise level = 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SSIM of 0.206, 0.378, 0.419, 0.449, 0.454, 0.619, 0.621, 0.617, and 0.693, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SSIM of 0.838.

**Based on image 2**

The comparative analysis in terms of SDME, PSNR, and SSIM, and using image 2 is shown in Figure 9. Figure 9(a) illustrates the analysis based on PSNR by varying the noise level. When the noise level is 0.2, then the corresponding PSNR values computed by existing cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS,
FIGURE 8  Comparative analysis by varying the noise level using image 1 (a) PSNR, (b) SDME, and (c) SSIM

FIGURE 9  Comparative analysis by varying the noise level using image 2 (a) PSNR, (b) SDME, and (c) SSIM
W-LDMM, C-ISSAPL, and proposed Jaya-Bat-based DeepCNN are 13.24, 13.85, 17.90, 18.62, 23.71, 33.63, 33.99, 32.53, 34.26, and 43.06 dB, respectively. For the noise level 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the PSNR of 13.17, 13.75, 13.77, 14.05, 21.95, 33.62, 33.83, 31.95, and 32.45 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the PSNR of 42.76 dB.

The comparative analysis in terms of SDME metric is depicted in Figure 9(b). When the compression ratio is 0.2, the SDME values achieved by cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 2.41, 33.05, 33.38, 34.03, 36.11, 43.12, 43.63, 39.57, 39.37, and 44.51 dB, respectively. For noise level $\theta = 0.3$, the existing techniques, like cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SDME of 2.37, 28.16, 28.47, 32.79, 34.15, 43.08, 43.60, 37.89, and 39.12 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SDME of 43.99 dB.

The comparative analysis in terms of SSIM metric is depicted in Figure 9(c). When the compression ratio is 0.1, the SSIM values achieved by cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 0.043, 0.196, 0.399, 0.399, 0.671, 0.690, 0.691, 0.557, 0.625, and 0.756, respectively. For noise level $\theta = 0.3$, the existing techniques, like cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SSIM of 0.041, 0.187, 0.323, 0.332, 0.400, 0.401, 0.586, 0.465, and 0.510, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SSIM of 0.638.

Based on image 3
The comparative analysis in terms of PSNR, SDME, and SSIM, and using image 3 is shown in Figure 10. Figure 10(a) illustrates the analysis based on PSNR by varying the noise level. When the noise level is 0.1, then the corresponding PSNR values computed by existing cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed Jaya-Bat-based DeepCNN are 13.90, 14.25, 23.54, 23.91, 25.24, 32.70, 33.33, 33.54, 38.09, and 44.56 dB, respectively. For the noise level 0.3, the existing techniques, like cost-based image reconstruction, DBAIN, AFSAW, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the PSNR of 13.86, 14, 14.11, 14.19, 22.07, 32.67, 32.88, 29.80, and 34.72 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the PSNR of 44.03 dB.

FIGURE 10  Comparative analysis by varying the noise level using image 3 (a) PSNR, (b) SDME, and (c) SSIM
TABLE 1 Comparative discussion

| Metrics/Methods               | Cost-based image reconstruction | DBAIN | AFSWA | CNN | LOTV-PADMM | DeepCNN | T2FCS | W-LDMM | C-ISSAPL | Proposed AGWO-based Deep stacked auto encoder |
|------------------------------|--------------------------------|-------|-------|-----|------------|---------|-------|--------|----------|---------------------------------------------|
| With impulse noise           |                                |       |       |     |            |         |       |        |          |                                             |
| PSNR (dB)                    | 8.660                          | 16.55 | 29.50 | 29.85 | 29.86      | 43.65   | 43.65 | 39.38  | 45.54     | 49.87                                      |
| SDME (dB)                    | 20.57                          | 41.57 | 51.53 | 52.63 | 52.71      | 57.51   | 57.95 | 54.04  | 56.51     | 62.96                                      |
| SSIM                         | 0.306                          | 0.463 | 0.619 | 0.621 | 0.790      | 0.826   | 0.827 | 0.663  | 0.775      | 0.829                                      |
| With random noise            |                                |       |       |     |            |         |       |        |          |                                             |
| PSNR (dB)                    | 8.659                          | 15.95 | 27.15 | 28.71 | 28.88      | 43.65   | 43.65 | 42.27  | 43.50     | 51.03                                      |
| SDME (dB)                    | 19.06                          | 39.26 | 49.50 | 51.76 | 51.85      | 57.91   | 57.94 | 55.08  | 55.98     | 62.45                                      |
| SSIM                         | 0.343                          | 0.378 | 0.620 | 0.620 | 0.725      | 0.815   | 0.818 | 0.746  | 0.767      | 0.848                                      |

The comparative analysis in terms of SDME metric is depicted in Figure 10(b). When the compression ratio is 0.2, the SDME values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, C-ISSAPL, and proposed model are 1.47, 33.22, 34.13, 34.82, 36.57, 45.65, 45.89, 42.49, 42.03, and 46.64 dB, respectively. For noise level \( = 0.3 \), the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SDME of 1.46, 28.91, 29.01, 33.29, 34.24, 44.57, 45.77, 41.25, and 41.45 dB, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SDME of 46.49 dB.

The comparative analysis in terms of SSIM metric is depicted in Figure 10(c). When the compression ratio is 0.1, the SSIM values achieved by cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, and proposed model are 0.038, 0.184, 0.324, 0.324, 0.585, 0.589, 0.594, 0.453, 0.513, and 0.660, respectively. For noise level \( = 0.3 \), the existing techniques, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, possesses the SSIM of 0.037, 0.174, 0.286, 0.291, 0.326, 0.328, 0.512, 0.379, and 0.428, respectively, which is comparatively lower than the Jaya-Bat-based DeepCNN. For the same noise level, the developed Jaya-Bat-based DeepCNN acquired the SSIM of 0.552.

4.6 Comparative discussion

Table 1 represents the comparative discussion of the developed model based on with impulse noise and random noise. The PSNR obtained by the existing methods, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, with random noise by varying the noise level is specified as 8.659, 15.95, 27.15, 28.71, 28.88, 43.65, 43.65, 42.27, and 43.50 dB, while the proposed Jaya-Bat-based DeepCNN obtained the better PSNR of 51.03 dB, respectively for with random noise. The SSIM obtained by the existing methods, like cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, by varying the noise level is specified as 0.343, 0.378, 0.620, 0.620, 0.725, 0.815, 0.818, 0.746, and 0.767, while the proposed Jaya-Bat-based DeepCNN obtained better SDME of 0.848, respectively. It is clearly specified that the proposed Jaya-Bat-based DeepCNN has attained better performance with random noise by varying the noise level, respectively. The SDME obtained by the existing methods, such as cost-based image reconstruction, DBAIN, AFSWA, CNN, LOTV-PADMM, DeepCNN, T2FCS, W-LDMM, and C-ISSAPL, with impulse noise is specified as 20.57, 41.57, 51.53, 52.63, 52.71, 57.51, 57.95, 55.08, and 55.98 dB, while the proposed Jaya-Bat-based DeepCNN obtained better SSIM of 62.96 dB, respectively.

5 CONCLUSION

This paper presents the restoration model based on Jaya-Bat-based Deep CNN in order to restore the input images from impulse noise and the random noise. Here, the image restoration process is performed by three stages. At first, the Deep CNN is introduced to identify the noisy pixels from input image. Subsequently, the discovered noise pixels get eliminated in second phase based on Jaya-Bat optimization algorithm. In tertiary phase, the pixel enhancement is carried out using the proposed neuro fuzzy system. Thus, the developed model restores the image by removing several types of noises, like random noise and impulse noise. The performance of the Jaya-Bat-based Deep CNN is evaluated based on PSNR, SDME, and SSIM. The proposed method produces the maximal PSNR of 51.03 dB, maximal SSIM of 0.848 with random noise, maximal SDME 62.96 dB with impulse noise in the image that indicates the superiority of the developed model. In future, extending the analysis based on other standard databases with the highly advanced features.
REFERENCES

1. Zhang, Y., et al.: Adaptive residual networks for high-quality image restoration. IEEE Trans. Image Process. 27(7), 3150–3163 (2018).
2. Shen, X., et al.: Multispectral joint image restoration via optimizing a scale map. IEEE Trans. Pattern Anal. Mach. Intell. 37(12), 2518–2530 (2015).
3. Conti, L., Barbati, M., Monti, M.: Design of sustainable agricultural buildings. A case study of a wine cellar in Tuscany, Italy. Buildings 6(2), 17 (2016).
4. Conti, L., et al.: Simplified method for the characterization of rectangular straw bales (RSB) thermal conductivity. IOP Conf. Ser.: Mater. Sci. Eng. 245(5), 052035 (2017).
5. Santosh Kumar, B.P., Venkata Ramanaiah, K.: An efficient hybrid optimization algorithm for image compression. Multimedia Research 2(4), 1–11 (2019).
6. Kumar, S.V., Nagaraju, C.: T2FCS filter: Type 2 fuzzy and cuckoo search-based filter design for image restoration. J. Visual Commun. Image Represent. 58, 619–641 (2019).
7. Zhang, M., Desrosiers, C.: High-quality image restoration using low-rank patch regularization and global structure sparsity. IEEE Trans. Image Process. 28(2), 868–879 (2018).
8. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. 15(12), 3736–3745 (2006).
9. Dabov, K., Foi, A., Egiazarian, K.: Video denoising by sparse 3D transform-domain collaborative filtering. In: IEEE 15th European Signal Processing Conference, pp. 145–149. IEEE, Piscataway, NJ (2007).
10. Niknejad, M., Rabbani, H., Babaie-Zadeh, M.: Image restoration using Gaussian mixture models with spatially constrained patch clustering. IEEE Trans. Image Process. 24(11), 3624–3636 (2015).
11. Papyan, V., Elad, M.: Multi-scale patch-based image restoration. IEEE Trans. Image Process. 25(1), 249–261 (2015).
12. Wang, R., Tao, D.: Non-local auto-encoder with collaborative stabilization for image restoration. IEEE Trans. Image Process. 25(5), 2117–2129 (2016).
13. Teng, Y., et al.: Adaptive morphological filtering method for structural fusion restoration of hyperspectral images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9(2), 655–667 (2015).
14. Dong, W., Zhang, L., Shi, G., Li, X.: Nonlocally centralized sparse representation for image restoration. IEEE Trans. Image Process. 22(4), 1620–1630 (2012).
15. Li, X.: Patch-based image interpolation: Algorithms and applications. In: International Workshop on Local and Non-Local Approximation in Image Processing, pp. 1–6. IEEE, Piscataway, NJ (2008).
16. Elmoataz, A., Lezoray, O., Bougleux, S.: Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. IEEE Trans. Image Process. 17(7), 1047–1060 (2008).
17. Donoho, D.L., Huo, X.: Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inf. Theory 47(7), 2845–2865 (2001).
18. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Digit. Signal Process. 10(1–3), 19–41 (2000).
19. Rabbani, H., et al.: Speckle noise reduction of medical ultrasound images in complex wavelet domain using mixture priors. IEEE Trans. Biomed. Eng. 55(9), 2152–2160 (2008).
20. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: IEEE International Conference on Computer Vision, pp. 479–486. IEEE, Piscataway, NJ (2011).
21. Zeng, D., et al.: Spectral CT image restoration via an average image-induced nonlocal means filter. IEEE Trans. Biomed. Eng. 63(5), 1044–1057 (2015).
22. Peng, Y.T., Cosman, P.C.: Underwater image restoration based on image blurriness and light absorption. IEEE Trans. Image Process. 26(4), 1579–1594 (2017).
23. Asif, M., et al.: An automatic framework for endoscopic image restoration and enhancement. Applied Intelligence (2020). https://link.springer.com/article/10.1007/s10489-020-01923-w#:~:text=The%20proposed%20framework%20consists%20of%20illumination%20of%20the%20endoscopy%20images
24. Wang, W., Wu, C., Tai, X.-C.: A globally convergent algorithm for a constrained non-Lipschitz image restoration model. J. Sci. Comput. 83, 14 (2020).
25. Rao, R.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations 7(1), 19–34 (2016).
26. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74. Springer, Berlin, Heidelberg (2010).
27. Zhang, K., et al.: Learning deep CNN denoiser prior for image restoration. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2808–2817. IEEE, Piscataway, NJ (2017).
28. Karaboga, D., Kaya, E.: Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. Artificial Intelligence Review 52(4), 2263–2293 (2019).
29. Statlog (Landsat Satellite) Data Set, https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29. Accessed 25 November 2020.
30. Chen, B.H., Yin, J.L., Li, Y.: High accurate image reconstruction for multimodal noise suppression using semi supervised learning on big data. IEEE Transaction on Multimedia 1–12 (2018).
31. Yuan, G., Ghanem, B.: I2FTV: A new method for image restoration in the presence of impulse noise. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 5369–5377. IEEE, Piscataway, NJ (2015).
32. Ng, P.E., Ma, K.K.: A switching median filter with boundary discriminative noise detection for extremely corrupted images. IEEE Transaction on Image Processing 15(6), 1507–1516 (2006).
33. Zhang, K., et al.: Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Trans. Image Process. 26(7), 3142–3155 (2017).
34. Varghese, J., et al.: Efficient adaptive fuzzy-based switching weighted average filter for the restoration of impulse corrupted digital images. IET Image Proc. 8(4), 199–206 (2013).

How to cite this article: S.P, Renjit J.A. Image restoration model using Jaya-Bat optimization-enabled noise prediction map. IET Image Process. 2021;15:1926–1939.
https://doi.org/10.1049/ipr2.12162