Review of Computer Engineering Studies
Vol. 8, No. 1, March, 2021, pp. 14-21
Journal homepage: http://iieta.org/journals/rces

# Image Denoising Based on Improved Hybrid Genetic Algorithm

Nail Alaoui*, Amel Baha Houda Adamou-Mitiche, Lahcène Mitiche, Lakhdar Bouhamla

Laboratoire de Recherche Modélisation, Simulation et Optimisation des Systèmes Complexes Réels, Université ZIANE Achour de Djelfa, Ain Chih, Djelfa 17000, Algeria

Corresponding Author Email: n.alaoui@univ-djelfa.dz

https://doi.org/10.18280/rces.080103

**Received:** 23 January 2021
**Accepted:** 15 March 2021

***Keywords:***
*image denoising, hybrid genetic algorithm, edge preservation, optimization, digital images*

**ABSTRACT**

Digital images can be degraded through noise during the transmission and process of acquisition, it is still a fundamental challenge is to eliminate as much noise as possible while preserving the main features of the image, for instance, edges, texture, and corners. This paper proposes for image denoising a new Improved Hybrid Genetic Algorithm (IHGA), whose combined a Genetic Algorithm (GA), with some image denoising methods. Wherein this approach uses mutation operators, crossover, and population reinitialization as default operators available in evolutionary methods with applied some state-of-the-art image denoising methods, such as local search. Tests are conducted on some digital images, commonly used as a benchmark by the scientific community, where different standard deviations are used for digital images. Experimental results indicate that the proposed method is very effective and competitive in comparison with previously published works.

## 1. INTRODUCTION

Image denoising is one of the exemplary issues of image handling, numerous methodologies have been acquainted with eliminate commotion from advanced image literature [1], yet eliminating noise from computerized images stay a difficult issue.

Digital images can be collected from various instruments, such as laser scanners, medical scanners, cameras, and weather satellites [2]. It is therefore important to remove the noise while maintaining the important features of the image, such as edges and corners. Noise can eventually corrupt images during processing, transmission, and compression processes.

This research describes a method for suppressing noise with an Improved Hybrid Genetic Algorithm (IHGA) for a digital image that combines genetic algorithm with some image denouncing techniques from BM3D [3], Anisotropic Diffusion [4], and Wiener-chop [5] literature. In this work, the IHGA implemented Improved Genetic Hybrid Algorithm which eliminates Gaussian noise in digital images. Our experimental results display that IHGA improves in general.

The remainder of the paper is as below. Section 2 describes the proposed Improved Genetic Hybrid Algorithm in this paper, and we present in detail the reviews of different techniques to denoise pictures. Section 3 summarizes the findings of the Section 4 experiments, and ends remarks in Section 5.

## 2. BACKGROUND

The key aim of the image denoising approach is to restore an original picture that has been polluted with additive noise without losing the image's edge information, such as texture and corners.

Some linear filtering [6] was suggested in the original images to eliminate the uniform and Gaussian noise. The filters used to eliminate the noise in optical images are known as linear filters, for example, is the Wiener filter, while non-linear filters are categorized as a median filter, for example. In linear filters, a kernel filter is transformed to the required result through a noise signal, whereas non-linear filters [7] cannot be regarded as a convolution process [1]. These filters are used to eliminate the noise in the image with white Gaussian noise applied without the need for any previous information.

Rational operators [8] have been applied to progress denoising techniques. Approaches based on computational fluid dynamics (CFD) and partial differential equations (PDE) have also been advanced, total variation (TV) methods [9], level set methods [10] non-linear isotropic and anisotropic diffusion [11].

Other methods have combined filtering techniques to remove impulse to suppress noise and local adaptive filtering in the transform domain [9]. Non-local filtering has been confirmed to be strong for image denoising, one of these methods is the BM3D [3] filter Singular Value Decomposition (SVD) has also been applied in the filtering of image noise [12]. Other methods collected wavelet transformations, spatially adaptive methods and hidden models of Markov [13].

In recent years several methods have been proposed using Evolutionary Algorithms for image denoising. Such methods generally attempt to implement the shrinkage rule by estimating thresholds on an image for the noisy wavelet coefficients [14-16].

A genetic algorithm is used to eliminate noise from image in the process suggested by de Paiva et al. [17, 18]. In this approach, a noisy picture is used as a contribution and certain methods of denotation are used to initialize mutation operators, crossover processing and population growth.

This work improves on several essential aspects of the approach introduced by de Paiva et al. [17]. First, it uses a new collection of mutations based on methods of image restoration. Secondly, a whole new range of crossovers. Second, a new approach to initializing a population is implemented, in this method by randomly crossing two people from the initial population group. In addition, there are other significant differences such as using a different selection method as selection of roulette wheels.

## 3. METHODOLOGIES

**Algorithm**: Proposed Improved Hybrid Genetic Algorithm

**Input**: Noisy image I.

Step 1: (Initialization) Create a group of three new individuals G= {IBM3D, IAD, IWiener -Chop} as the initial population by Apply filters BM3D, AD and Wiener-Chop over input image I.

Step 2: while the initial population size is less than PopSize do

Step 3: Select a two individual randomly of individuals from a set G.

Step 4: Procedure a random crossing for two individual of this selected pair and integrate each individua of the resulting individuals into the initial population.

Step 5: end while

Step 6: (Evaluation) Each individual of the initial population is evaluated by a fitness function.

Step 7: while the Runtime is less than MaxTime and the iteration number is less than MaxIter do.

Step 8: repeat

Step 9: (Parent selection) Select a pair of individuals from the population using a Roulette Wheel Selection.

Step 10: (Crossover) The offspring are created by recombining pairs of the selected parents to a new generation.

Step 11: (Mutation) Mutate to each offspring using one of three mutations are proposed which are also selection randomly to be used with probability Pm.

Step 12: (Evaluation) Evaluate the fitness of each offspring.

Step 13: (Local Search) If a randomly selected value from [0, 1] is Less than the local search rate, apply local Search operator at the end of each evolutionary step to the best individual found.

Step 14: end if

Step 15: (Elitism) generate a new generation of PopSize individuals using deterministic fitness-based replacement.

Step 16: (Reset population) if the runtime is less than MaxTime integrate the best individual of the previous generation previous with created a new population by the same process used to the initial population.

Step 17: end if

Step 18: (Evaluation) Evaluate each new population's fitness

Step 19: until complete PopSize generations.

Step 20: end while

Step 21: the best image of the last generation returns.

This section describes IHGA, our proposed Improved Genetic Hybrid Algorithm which suppresses image noise. The input of the proposed algorithm is a gray-scale image that I was interrupted by Gaussian noise. The Denoised image of I is the production of the initial population each person in IHGA is represented as a denoised image of I. The proposed algorithm outlines.

### 3.1 Initialization

Initial populations with PopSize are generated by Lines 1-5 of our algorithms. In which a double-dimensional (2D) pixel array of values within [0,255] range represents each person in the population. An updated version of image I entry reflects all users. After each of the subsequent denoising filters, the first three population individuals use a denoted graphic. BM3D, AD, Wiener Chop.

These methods are classified as computationally fast filters to take advantage of their strength due to their image denoising competency as well as their short computational time. Those methods are considered to be the best literature findings.

The algorithm IHGA creates the other individuals of the initial population by selecting from the set {IBM3D, IAD, Iwiener-Chop} two individuals IX and IY. The outputs are submitted by a random crossing between the two individuals, which exchanges pixels point-to- point. This new individual recombination operation output is included in the initial population and used this operation repetitively until PopSize individuals were attained by the initial population. Figure 1 shows a block diagram of the initial population created.

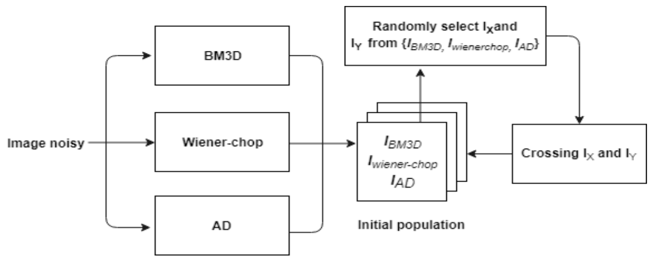



**Figure 1.** Creation of the initial population of IHGA

### 3.2 Fitness function

Lines 6, 12 and 18 of our Algorithm evaluate the fitness of the population. The algorithm is guided based on a fitness function represented by minimizing Eq. (1). As stated in the study [19].

$$fitness(i) = \{\sum_{\Omega} \sqrt{1+\beta^2|\nabla I|^2} + \frac{\lambda}{2}(I - I_0)^2\} \tag{1}$$

Mindful the edges of the image and attempts to save significant highlights of the image work portrayed in the parameter, I is the picture being assessed, I0 the loud picture, β and λ are adjusting boundaries and Ω is the group of all focuses in the image.

Full names of authors are required. The middle name can be abbreviated.

### 3.3 Parent selection

Line 9 of our Algorithm create parents by selects pairs of individuals who are selected through roulette wheel selection.

### 3.4 Crossover

Following a selection of our algorithm's step parents in line

9, line 10, the new person is generated by randomly selecting one of three crossover operators shows next:

Single-point: On both parents a single crossover picked a point, one of the two that we randomly pick in this process.

One-point column: Random arrangement of a progression of pixels. The pixels over this line originate from one parent. The pixels from the subsequent parent are all beneath this line.

One-point row: Similar to the past methodology yet rather than picking a segment from a column.

Two-point: Two-point hybrid chose two focuses on the two guardians; we arbitrarily pick one of the two in this cycle.

Two point column: two hybrid focuses are chosen haphazardly from the exhibit, all pixels duplicated from the beginning of the chromosome to a parent's first hybrid point, at that point all pixels are replicated from the principal hybrid purpose of the parent to the second traverse purpose of the parent, and the rest of replicated from the first of the parent.

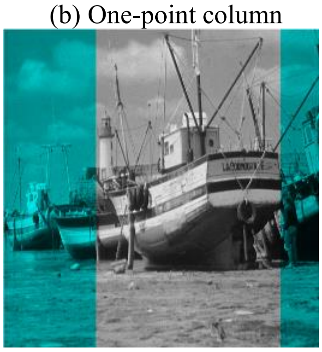
Two-point row: segment like the past structure, yet favor a segment as opposed to a column.

Cross grid: a solitary point and one-point administrator blend is utilized to fragment each image into four quadrants, however not actually equivalent measurements. In several image, he shares a quadrant. In Figure 2, this Fusion reveals the effect.

(a) One-point row

(b) One-point column

(c) Two-point row



(d) Two-point: column

(e) Cross Grid

**Figure 2.** Examples of the crossover

### 3.5 Mutation

Line 11 of the proposed Algorithm executes a grid mutation operator which provides population diversity. This operator takes a single Ix with probability mutation rate as its input. Next, it selects two rows and two Ix columns at random. Then it labels the rectangle formed by the rows and columns that were selected. Second, by randomly applying one of the three filters presented next, it treated the area:

Filter motion blur: filters a filter motion in the picture, this filter produces a motion blur.

Median filter: filters the image by means of a median filter, the size is randomly selected between 3 pixels and 5 pixels.

Intensity: Each pixel of the image is multiplied by the same factor chosen randomly between the interval [0.8, 1.2].

This operator created a mutated offspring Ix', Figure 3 shows a result of this mutation operator.

**Figure 3.** Example of the grid mutation operator

### 3.6 Local search

Line 13 of our algorithm explains that when the randomly selected value of [0, 1] is less than the local search rate of the algorithm, a local search operator is applied to the best individual found in a new individual using the denoising method of the three described previously BM3D [3], AD [4], and Wiener-chop [5].

### 3.7 Population replacement

Line 15 of the proposed algorithm is a modified step that only guarantees that the right person is available. The fitness replacement scheme is formed by the union of some of the parents of the previous generation and some of their offspring's in order to perform with a sorting algorithm to choose certain people.

### 3.8 Reset population

The population is reset in line 16 of our algorithm to retain the best people and build the majority of the new people in the same method with the first generation.

### 3.9 Termination condition

Lines 8 to 19 of our Algorithm repeats the algorithm until it completes Pop Size generations even a condition is met in line 7. Next, the algorithm returns the best individual present in the last generation (see Line 21).

## 4. EXPERIMENTAL RESULTS

The experimental results from the proposed improved genetic hybrid algorithm (IHGA) for image denoise are presented in this section with the intention of testing the efficiency of our proposed developmental algorithm and compared the proposed algorithm to state-of-the-art images denoise methods. For this function the additive Gaussian noise disrupted each of the seven images with 11 different standard variants σ = 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60. For this purpose we used 7 images.

We measure the objective quality metrics to evaluate the quality of the image restored after a filtering process. Eq. (2) presented the Peak Signal to Noise Ratio (PSNR) measure via the Mean Square Error (MSE) of Eq. (3).

$$PSNR = 10\ log_{10}\ (\frac{255^2}{MSE}) \tag{2}$$

The MSE is the mean squared error between the original (O) and the recovered images (K). M and N It is dimensions of the image.

$$MSE = \frac{1}{M\ N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[O(i,j) - K(i,j)]^2 \tag{3}$$

### 4.1 Setting parameter

To test the performance of IHGA parameters, tests were performed using different parameters for each test, and the target quality metrics were calculated to select the best parameter.

The basic configuration of the IHGA after performing all these tests are shown in Table 1.

**Table 1.** The basic configuration of the IHGA

| Selection Pressure | 8 |
|---|---|
| Mutation rate | 0.2 |
| Population size | 15 |
| β | 1.5 |
| MaxTime | 20 minutes |

MaxIter = 5, locale search rate=0.8, and λ was $1/\sqrt{(v)}$ where ν is the estimated variance of the noise, except for the parameters maxIter, λ and local search rate which were chosen empirically.

### 4.2 Comparison of the results

The results of the IHGA in this section against the other approaches used in the literature used in contrast were Bayes [20], Wiener [1], median [1], TV [9]. Wavelets. The results of the PSNR were given in Table 2., Wiener-chop [5], AD [4], BM3D [3], and HGA [17]. For the noise ratio, the value displayed in bold is the highest value and the underlined values are the lowest.

Results also demonstrated that IHGA is comparable with some of the best picture denoising approaches available in the literature, even though in some cases its worst results (IHGA Min) also yielded better results. In most cases, IHGA proposed innovative technique gives superior results than those techniques used as local search operators. In addition, the effects of this hybrid technique would be entitled to outperform other available approaches published in the literature.

In order to validate this process, IHGA is compared with HGA, which has been able to obtain better results than the ones described in the study [17]. This indicates that some changes in the HGA and its combining with other techniques will help to provide the better output solution. The suggested HGA was introduced for the same runtime of the IHGA run at [17]. This amendment was introduced in order to allow for a rational distinction of the two approaches.

Table 2 illustrates the minimum (Min), maximum (Max) and average (Avg) PSNR obtained by the IHGA. It is also presented the HGA and other methods found in the literature. When examining the maximum results, the proposed method IHGA was the optimum method in terms of PSNR presented the greater results in 59 out of 77 tests (77%). When examining the number of times that the IHGA, it was top PSNR than the other methods with all tested noise levels, against the 88 results for the other methods. The proposed IHGA is greater than other methods in 84 times for Man (96% of the cases), 96% for Boat image, 97% for Lenna image, 99% for Glasses image, 96% for Peppers image, 99% for Lightning image, and 96% for Cameraman image.

Instead of analyzing the best cases Such as those mentioned in the previous paragraph, we conducted an analysis of the worst cases and the average cases, with taking into account the same comparisons the proposed method. IHGA was best than the other methods. In the average and worst cases, respectively, PSNR found by the proposed IHGA is greater than PSNR values of other methods, for the Man image at 81% and 60% of the time, 73% and 58% for Boat image, 82% and 76% for Lenna image, 97% and 93% for Glasses image, 85% and 77% for Peppers image, 96% and 93% for Lightning image, and 78% and 70% for Cameraman image.

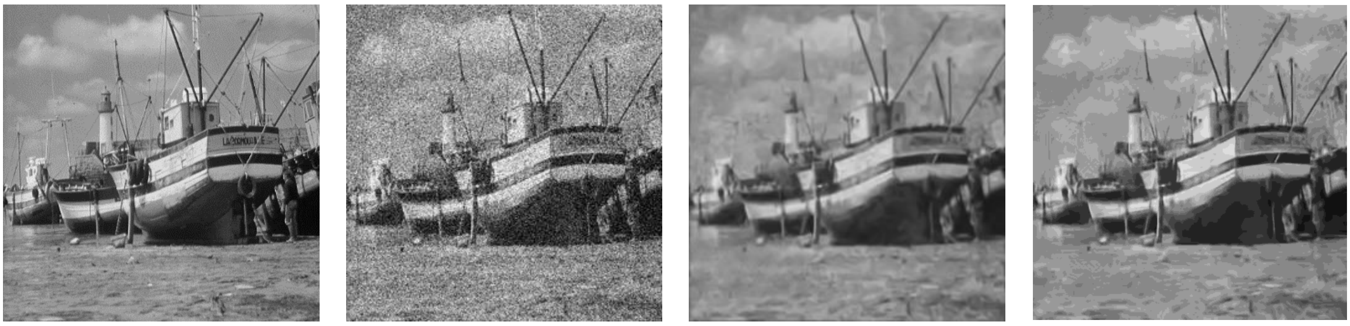
IHGA is now compared against Best methods for denoising images it uses as local search, when making the same comparisons as those that were made in the previous, but taking into account it against only BM3D, AD, and Wiener-chop. The maximum PSNR outperforms these methods for the Man image at 90% of the time, 87% for Boat image, 90% for Lenna image, 100% for Glasses image, 90% for Peppers image, 100% for Lightning image, and 87% for Cameraman image. In the average case and worst cases. Respectively, IHGA has the best PSNR than the theses methods at for the Man image at 63% and 36% of the time, 73% and 30% for Boat image, 66% and 54% for Lenna image, 90% and 78% for Glasses image, 66% and 54% for Peppers image, 87% and 87% for Lightning image, and 51% and 42% for Cameraman image.

When Comparison of the results of the proposed method IHGA for it against only HGA show that the Image quality is improved without losing image features, indicates that our technique has an advantage over HGA. For PSNR metric, our analysis shows that the IHGA is better than HGA 94% of the time in the best case, 92% in the average case, and 72% in the worst result for the 10 executions.

The IHGA algorithm is more efficient in eliminating Gaussian noise than a IHGA, especially at the high noise level For example σ =60 (see Figures 4, 5, 6, 7, 8, 9, and 10).

**Table 2.** PSNR results for all the tested methods

| Images | σ | Bayes | WC | Median | Wiener | AD | BM3D | TV | HGA avg | HGA max | HGA min | IHGA avg | IHGA max | IHGA min |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Man | 10 | 31,31 | 32,11 | 30,204 | 30,49 | 32,57 | **33,62** | 30,74 | 32,51 | 32,66 | 32,27 | 32,58 | 33,61 | 32,10 |
| | 15 | 29,75 | 30,50 | 28,744 | 29,485 | 30,465 | **31,718** | 30,268 | 30,556 | 30,719 | 30,374 | 30,554 | 31,605 | 30,328 |
| | 20 | 28,69 | 29,45 | 27,402 | 28,562 | 29,158 | **30,452** | 29,182 | 29,461 | 29,937 | 29,215 | 29,885 | 30,012 | 29,248 |
| | 25 | 27,75 | 28,56 | 26,106 | 27,641 | 28,136 | 29,542 | 27,601 | 28,990 | **29,576** | 28,146 | 28,980 | 29,56 | 28,143 |
| | 30 | 26,96 | 27,82 | 24,964 | 26,811 | 27,311 | 28,821 | 25,884 | 27,75 | 28,526 | 27,266 | 28,530 | **28,846** | 27,228 |
| | 35 | 26,41 | 27,12 | 23,872 | 26,063 | 26,655 | 28,176 | 24,232 | 27,091 | 27,637 | 26,407 | 27,160 | **28,197** | 26,824 |
| | 40 | 25,85 | 26,61 | 22,984 | 25,354 | 26,129 | 27,598 | 22,814 | 26,301 | 26,677 | 25,879 | 26,570 | **27,625** | 26,411 |
| | 45 | 25,40 | 26,05 | 22,039 | 24,635 | 25,528 | 27,057 | 21,414 | 25,887 | 26,236 | 25,498 | 26,310 | **27,088** | 26,060 |
| | 50 | 25,00 | 25,55 | 21,294 | 24,027 | 25,064 | 26,616 | 20,307 | 25,465 | 25,778 | 24,975 | 25,880 | **26,627** | 25,574 |
| | 55 | 24,604 | 24,967 | 20,566 | 23,467 | 24,623 | 26,237 | 19,217 | 25,039 | 25,518 | 24,182 | 25,559 | **26,245** | 24,98 |
| | 60 | 24,192 | 24,534 | 19,907 | 22,969 | 24,175 | 25,742 | 18,397 | 24,651 | 25,087 | 24,24 | 25,135 | **25,756** | 24,77 |
| Boat | 10 | 30,32 | 32,24 | 29,412 | 30,042 | 32,338 | **33,587** | 30,461 | 32,30 | 32,382 | 32,081 | 32,252 | 33,297 | 32,080 |
| | 15 | 29,05 | 30,55 | 28,178 | 29,013 | 30,388 | **31,930** | 29,867 | 30,481 | 30,598 | 30,099 | 30,58 | 31,631 | 30,002 |
| | 20 | 27,94 | 29,37 | 26,958 | 28,120 | 28,986 | **30,791** | 28,879 | 29,223 | 30,047 | 28,711 | 29,415 | **30,791** | 28,652 |
| | 25 | 26,95 | 28,39 | 25,768 | 27,185 | 27,867 | 29,782 | 27,355 | 28,281 | 29,056 | 27,785 | 28,405 | **29,785** | 27,65 |
| | 30 | 26,19 | 27,60 | 24,673 | 26,354 | 26,971 | 28,978 | 25,72 | 27,589 | 28,678 | 26,985 | 27,602 | **29,252** | 26,971 |
| | 35 | 25,59 | 26,96 | 23,721 | 25,627 | 26,26 | 28,362 | 24,125 | 26,392 | 27,390 | 25,410 | 27,100 | **28,452** | 26,884 |
| | 40 | 24,99 | 26,35 | 22,820 | 24,871 | 25,599 | 27,585 | 22,687 | 25,859 | 26,615 | 24,725 | 26,560 | **27,674** | 26,150 |
| | 45 | 24,58 | 25,72 | 21,954 | 24,200 | 25,026 | 26,906 | 21,329 | 25,393 | 25,854 | 24,847 | 26,156 | **26,987** | 25,706 |
| | 50 | 24,15 | 25,22 | 21,225 | 23,626 | 24,512 | 26,476 | 20,228 | 24,753 | 25,433 | 24,122 | 25,700 | **26,547** | 25,420 |
| | 55 | 23,79 | 24,66 | 20,504 | 23,035 | 24,003 | 25,925 | 19,142 | 24,389 | 24,995 | 23,016 | 25,147 | **26,025** | 24,720 |
| | 60 | 23,37 | 24,22 | 19,877 | 22,558 | 23,611 | 25,426 | 18,250 | 23,909 | 24,585 | 23,138 | 24,865 | **25,777** | 24,289 |
| Lenna | 10 | 33,33 | 34,35 | 32,116 | 32,661 | 34,177 | **35,873** | 33,06 | 34,287 | 34,481 | 34,084 | 34,344 | 35,773 | 34,178 |
| | 15 | 31,84 | 32,72 | 30,042 | 31,215 | 32,155 | **34,248** | 32,227 | 32,456 | 32,633 | 32,171 | 32,720 | 34,107 | 32,047 |
| | 20 | 30,56 | 31,50 | 28,361 | 29,996 | 30,786 | **32,999** | 30,586 | 31,809 | 32,717 | 31,213 | 32,414 | 32,901 | 31,191 |
| | 25 | 29,48 | 30,47 | 26,815 | 28,847 | 29,672 | 32,014 | 28,457 | 31,209 | **32,025** | 30,095 | 31,144 | 32,014 | 30,017 |
| | 30 | 28,75 | 29,65 | 25,533 | 27,868 | 28,831 | 31,184 | 26,478 | 30,09 | 30,735 | 29,207 | 30,012 | **31,304** | 29,101 |
| | 35 | 27,98 | 28,87 | 24,38 | 26,992 | 28,162 | 30,507 | 24,63 | 28,742 | 29,079 | 28,248 | 30,104 | **30,611** | 29,301 |
| | 40 | 27,51 | 28,07 | 23,297 | 26,084 | 27,38 | 29,932 | 22,991 | 28,195 | 28,910 | 27,599 | 29,411 | **29,999** | 28,61 |
| | 45 | 27,02 | 27,423 | 22,432 | 25,366 | 26,867 | 29,338 | 21,628 | 27,477 | 28,239 | 27,019 | 28,438 | **29,414** | 28,251 |
| | 50 | 26,50 | 26,753 | 21,589 | 24,632 | 26,264 | 28,775 | 20,388 | 26,916 | 27,372 | 26,341 | 28,215 | **28,810** | 27,321 |
| | 55 | 25,99 | 26,227 | 20,802 | 24,043 | 25,798 | 28,284 | 19,369 | 26,639 | 27,333 | 25,643 | 27,64 | **28,381** | 26,984 |
| | 60 | 25,47 | 25,645 | 20,117 | 23,485 | 25,288 | 27,746 | 18,448 | 25,966 | 26,853 | 25,137 | 27,248 | **27,819** | 26,61 |
| Glasses | 10 | 39,64 | 40,961 | 35,166 | 37,74 | 40,088 | 43,348 | 40,165 | 40,933 | 41,296 | 40,632 | 43,347 | **43,610** | 42,21 |
| | 15 | 37,32 | 38,333 | 31,944 | 34,895 | 37,665 | 41,455 | 37,928 | 39,457 | 40,702 | 38,457 | 41,445 | **41,601** | 40,278 |
| | 20 | 35,53 | 36,018 | 29,494 | 32,555 | 35,844 | 39,727 | 33,684 | 38,709 | 39,453 | 37,155 | 39,727 | **39,793** | 38,415 |
| | 25 | 34,78 | 34,574 | 27,652 | 30,978 | 34,366 | 38,458 | 30,207 | 38,496 | **38,501** | 38,443 | 38,453 | **38,501** | 37,458 |
| | 30 | 34,05 | 33,485 | 26,16 | 29,798 | 33,352 | 37,286 | 27,566 | 35,374 | 36,575 | 33,669 | 37,346 | **37,483** | 37,286 |
| | 35 | 33,25 | 32,475 | 24,885 | 28,765 | 32,36 | 36,379 | 25,492 | 34,165 | 35,116 | 32,667 | 36,489 | **36,616** | 36,379 |
| | 40 | 32,38 | 31,077 | 23,627 | 27,633 | 31,226 | 35,390 | 23,627 | 32,935 | 33,825 | 31,773 | 35,405 | **35,495** | 35,39 |
| | 45 | 31,56 | 30,511 | 22,779 | 26,863 | 30,614 | 34,840 | 22,344 | 32,566 | 33,256 | 32,121 | 34,842 | **34,999** | 34,105 |
| | 50 | 30,91 | 29,502 | 21,856 | 26,044 | 29,696 | 33,760 | 21,156 | 31,577 | 31,922 | 30,727 | 33,765 | **33,820** | 33,76 |
| | 55 | 29,85 | 28,567 | 21,024 | 25,148 | 28,912 | 32,455 | 20,122 | 30,567 | 30,975 | 30,078 | 32,590 | **32,877** | 32,255 |
| | 60 | 28,94 | 27,646 | 20,351 | 24,502 | 27,998 | 31,448 | 19,21 | 29,739 | 30,496 | 28,747 | 31,865 | **31,968** | 31,148 |
| Peppers | 10 | 30,93 | 32,617 | 30,101 | 30,641 | 33,378 | **34,500** | 32,097 | 33,444 | 33,601 | 33,246 | 33,517 | 34,420 | 33,114 |
| | 15 | 28,33 | 30,735 | 28,63 | 29,55 | 31,204 | **32,578** | 31,161 | 31,279 | 31,701 | 30,244 | 31,571 | 32,575 | 30,114 |
| | 20 | 26,52 | 29,261 | 27,285 | 28,422 | 29,607 | **31,115** | 29,646 | 29,999 | 30,322 | 29,798 | 30,301 | 31,105 | 29,625 |
| | 25 | 24,94 | 28,017 | 25,908 | 27,334 | 28,379 | **30,073** | 27,667 | 28,795 | 29,451 | 28,267 | 29,412 | 30,240 | 28,171 |
| | 30 | 24,38 | 27,114 | 24,821 | 26,493 | 27,269 | 29,070 | 25,98 | 27,926 | 28,521 | 26,992 | 28,415 | **29,178** | 27,601 |
| | 35 | 23,75 | 26,152 | 23,7 | 25,605 | 26,412 | 28,328 | 24,269 | 26,566 | 27,018 | 26,184 | 27,61 | **28,521** | 27,008 |
| | 40 | 23,26 | 25,253 | 22,774 | 24,852 | 25,637 | 27,531 | 22,708 | 25,769 | 26,339 | 25,215 | 27,031 | **27,610** | 26,328 |
| | 45 | 23,03 | 24,555 | 21,935 | 24,19 | 24,857 | 26,784 | 21,401 | 25,181 | 25,501 | 24,762 | 26,251 | **26,888** | 25,48 |
| | 50 | 22,48 | 23,887 | 21,212 | 23,555 | 24,257 | 25,971 | 20,372 | 24,316 | 25,026 | 23,847 | 25,441 | **26,101** | 24,81 |
| | 55 | 22,13 | 23,329 | 20,415 | 22,947 | 23,644 | 25,675 | 19,228 | 23,719 | 24,115 | 23,194 | 25,257 | **25,901** | 24,644 |
| | 60 | 21,90 | 22,834 | 19,837 | 22,446 | 23,151 | 25,101 | 18,389 | 23,172 | 23,575 | 22,762 | 24,611 | **25,301** | 24,045 |
| Lightning | 10 | 33,52 | 37,535 | 33,73 | 36,009 | 38,359 | 40,169 | 37,477 | 38,939 | 39,059 | 38,695 | 40,16 | **40,369** | 40,017 |
| | 15 | 31,61 | 35,202 | 31,144 | 33,848 | 36,101 | 38,171 | 35,988 | 36,909 | 37,271 | 36,593 | 38,247 | **38,341** | 38,17 |
| | 20 | 30,29 | 33,301 | 28,989 | 31,902 | 34,37 | 36,662 | 32,769 | 35,879 | 36,457 | 35,218 | 36,742 | **36,777** | 36,66 |
| | 25 | 29,43 | 32,127 | 27,298 | 30,445 | 33,062 | 35,533 | 29,819 | 35,449 | **35,542** | 34,933 | 35,341 | **35,542** | 34,633 |
| | 30 | 28,86 | 31,117 | 25,828 | 29,106 | 32,006 | 34,707 | 27,214 | 33,756 | 34,618 | 33,037 | 34,814 | **34,847** | 34,41 |
| | 35 | 28,52 | 30,148 | 24,581 | 27,967 | 31,062 | 33,659 | 25,132 | 32,168 | 32,548 | 31,61 | 33,74 | **33,744** | 33,659 |
| | 40 | 27,74 | 29,415 | 23,535 | 26,898 | 30,182 | 33,166 | 23,441 | 31,155 | 31,809 | 29,927 | 33,168 | **33,216** | 33,166 |
| | 45 | 27,09 | 28,692 | 22,597 | 26,102 | 29,483 | 32,507 | 22,012 | 30,802 | 31,208 | 29,395 | 32,617 | **32,807** | 32,507 |
| | 50 | 26,46 | 27,928 | 21,667 | 25,164 | 28,481 | 31,711 | 20,765 | 29,901 | 30,422 | 28,784 | 31,715 | **31,810** | 31,715 |
| | 55 | 25,90 | 27,461 | 20,987 | 24,655 | 28,084 | 31,152 | 19,787 | 29,495 | 30,084 | 28,427 | 31,156 | **31,262** | 30,156 |
| | 60 | 25,64 | 26,61 | 20,231 | 23,866 | 27,100 | 30,466 | 18,816 | 28,379 | 29,021 | 27,411 | 30,468 | **30,567** | 30,466 |
| Cameraman | 10 | 31,36 | 31.194 | 26.567 | 29.146 | 33.086 | **33.548** | 29.177 | 32.747 | 33.045 | 31.185 | 33.087 | 33.414 | 32.577 |
| | 15 | 28,55 | 29.255 | 23.810 | 29.255 | 30.807 | **31.538** | 28.891 | 30.613 | 30.839 | 30.134 | 30.447 | 31.417 | 30,038 |
| | 20 | 26,98 | 27.874 | 23.810 | 29.255 | 29.236 | **30.200** | 27.907 | 29.381 | 29.591 | 29.176 | 29.477 | 30.110 | 29,047 |
| | 25 | 25,76 | 26.806 | 23.810 | 29.255 | 27.906 | **29.246** | 26.768 | 28.218 | 27.666 | 27.929 | 28.147 | 29.141 | 27,87 |
| | 30 | 25,03 | 26.001 | 23.810 | 29.255 | 26.797 | 28.260 | 26.421 | 27.136 | 27.481 | 26.924 | 27.571 | **28.310** | 26,722 |
| | 35 | 24,25 | 25.184 | 23.810 | 29.255 | 25.817 | 27.535 | 23.882 | 26.044 | 26.297 | 25.531 | 26.018 | **27.644** | 25,541 |
| | 40 | 23,67 | 24.563 | 23.810 | 29.255 | 24.992 | 26.748 | 22.638 | 25.230 | 25.448 | 24.427 | 25.351 | **26.887** | 24,86 |
| | 45 | 22,98 | 23.904 | 23.810 | 29.255 | 24.170 | 25.789 | 21.315 | 24.266 | 24.662 | 23.363 | 24.630 | **25.977** | 24,135 |
| | 50 | 22,25 | 23.288 | 23.810 | 29.255 | 23.389 | 25.299 | 20.237 | 23.651 | 23.838 | 23.185 | 23.515 | **25.319** | 23,425 |
| | 55 | 21,90 | 22.833 | 19.940 | 21.644 | 22.785 | 24.514 | 19.231 | 23.166 | 23.322 | 22.808 | 23.208 | **24.718** | 23,052 |
| | 60 | 21,53 | 22.454 | 19.481 | 21.212 | 22.320 | 24.162 | 18.482 | 22.631 | 22.851 | 22.193 | 22.744 | **24.241** | 22,425 |

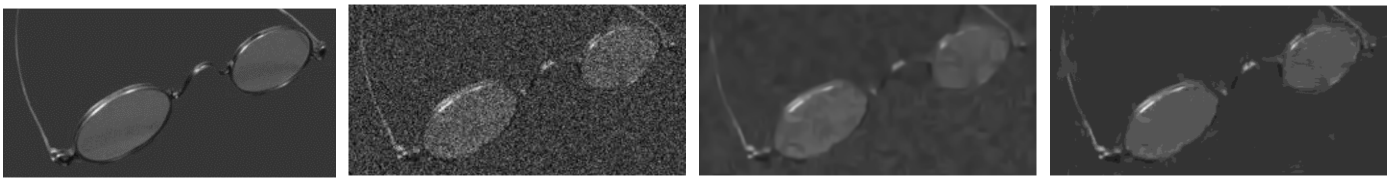


(a) image Original (b) image Noisy (σ =60) (c) HGA (d) IHGA

**Figure 4.** Results of the methods for Boat image by HGA and IHGA

(a) image Original (b) image Noisy (σ =60) (c) HGA (d) IHGA

**Figure 5.** Results of the methods for Cameraman image by HGA and IHGA



(a) image Original (b) image Noisy (σ =60) (c) HGA (d) IHGA

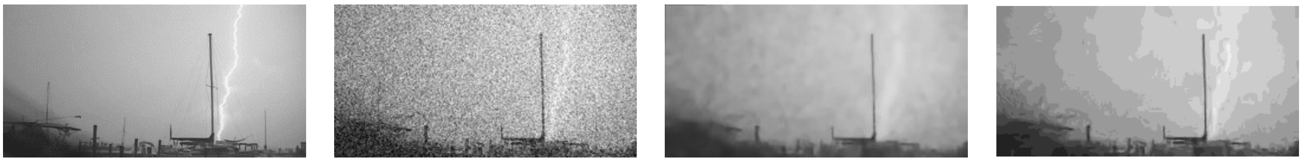
**Figure 6.** Results of the methods for Cameraman image by HGA and IHGA

(a) image Original (b) image Noisy (σ =60) (c) HGA (d) IHGA

**Figure 7.** Results of the methods for Lenna image by HGA and IHGA

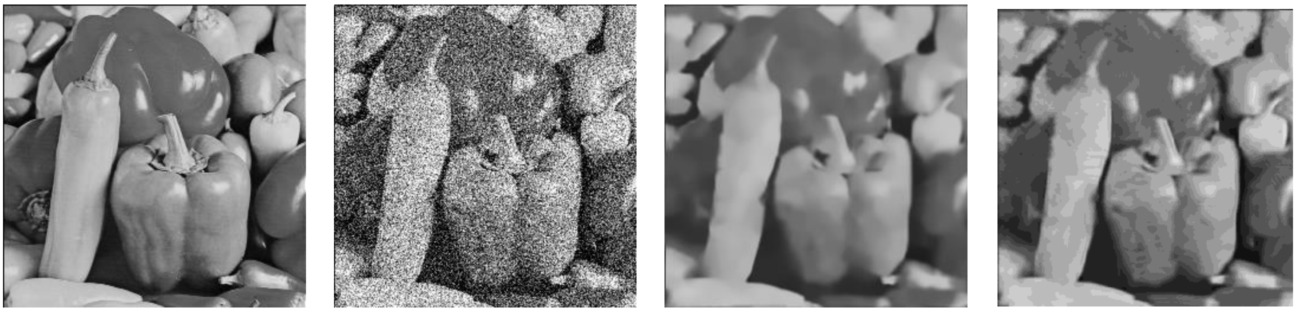
(a) image Origin (b) image Noisy (σ =60) (c) HGA (d) IHGA

**Figure 8.** Results of the methods for Man image by HGA and IHGA



(a) image Original (b) image Noisy (σ =60) (c) HGA (d) IHGA

**Figure 9.** Results of the methods for Lightning image by HGA and IHGA



(a) image Original (b) image Noisy (σ =60) (c) HGA (d) IHGA

**Figure 10.** Results of the methods for Peppers image by HGA and IHGA

## 5. CONCLUSIONS

In this paper, we have presented an Improved Hybrid Genetic Algorithm (IHGA), this method, although inspired by HGA has a number of fundamental changes, such as the use different operators of mutation and crossover, local search operators used only the best candidate that was identified at the conclusion of and evolutionary process. We also have changed the method selection process. IHGA was evaluated against other denoising methods, where were used seven different images with the 11 levels of noise. Experimental results present that IGHA outperformed a previous an approach based on HGA, which indicates that we have found an improves solution for image denoising problem. In comparison with the other denoising image found.

In the literature, especially images with high noise levels. Taking the best solutions into consideration, the average and the worst solutions found, which measured using PSNR. IHGA is still slow compared to some image denoising methods. This problem becomes more apparent when several executions. As future work, we intend to reduce the computational cost through proposed new fitness functions and other image denoising techniques can be proposed as local search.


## ACKNOWLEDGMENT

The research being reported in this publication was supported by the Algerian Directorate General for Scientific Research and Technological Development (DGRSDT).


## REFERENCES


[1] Masters, B.R., Gonzalez, R.C., Woods, R., (2009). Book Review: Digital Image Processing, Third Edition. Digital Image Processing. https://doi.org/10.1117/1.3115362

[2] Sakthidasan, K., Nagappan, N.V. (2016). Noise free image restoration using hybrid filter with adaptive genetic algorithm. Computers & Electrical Engineering, 54: 382-392. http://dx.doi.org/10.1016/j.compeleceng.2015.12.011

[3] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K. (2006). Image denoising with block-matching and 3D filtering. In Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, 6064: 606414. http://dx.doi.org/10.1117/12.643267

[4] Black, M.J., Sapiro, G., Marimont, D.H., Heeger, D. (1998). Robust anisotropic diffusion. IEEE Transactions on Image Processing, 7(3): 421-432. http://dx.doi.org/10.1109/83.661192

[5] Ghael, S., Sayeed, A.M., Baraniuk, R.G. (1997). Improved wavelet denoising via empirical Wiener filtering. In SPIE Technical Conference on Wavelet Applications in Signal Processing. https://doi.org/10.1117/12.292799

[6] Russo, F. (2004). Image filtering based on piecewise linear models. In 2004 IEEE International Workshop on Imaging Systems and Techniques (IST) (IEEE Cat. No. 04EX896), pp. 7-12. http://dx.doi.org/10.1109/IST.2004.1397271

[7] Kouri, D.J., Zhang, M.M., Zhang, D.S. (2017). On a new nonlinear image filtering technique. In 2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1-5. http://dx.doi.org/10.1109/ICSPCS.2017.8270495

[8] Ramponi, G. (1996). The rational filter for image smoothing. IEEE Signal Processing Letters, 3(3): 63-65. http://dx.doi.org/10.1109/97.481156

[9] Rudin, L.I., Osher, S., Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. Physica D: nonlinear Phenomena, 60(1-4): 259-268. http://dx.doi.org/10.1016/0167-2789(92)90242-F

[10] Sethian, J.A. (1999). Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press. https://doi.org/10.1017/S0263574799212404

[11] Barbu, T. (2017). Nonlinear anisotropic diffusion-based structural inpainting framework. In 2017 13th International Conference on Advanced Technologies,

Systems and Services in Telecommunications (TELSIKS), pp. 207-210. https://doi.org/10.1109/TELSKS.2017.8246264

[12] Orchard, J., Ebrahimi, M., Wong, A. (2008). Efficient nonlocal-means denoising using the SVD. In 2008 15th IEEE International Conference on Image Processing, pp. 1732-1735. http://dx.doi.org/10.1109/ICIP.2008.4712109

[13] Amini, M., Ahmad, M.O., Swamy, M.N.S. (2014). Image denoising in wavelet domain using the vector-based hidden Markov model. In 2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS), pp. 29-32. http://dx.doi.org/10.1109/NEWCAS.2014.6933977

[14] Gupta, V., Chan, C.C., Sian, P.T. (2007). A differential evolution approach to PET image de-noising. In 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4173-4176. http://dx.doi.org/10.1109/IEMBS.2007.4353256

[15] Da Silva, R.D., Minetto, R., Schwartz, W.R., Pedrini, H. (2013). Adaptive edge-preserving image denoising using wavelet transforms. Pattern Analysis and Applications, 16(4): 567-580. http://dx.doi.org/10.1007/s10044-012-0266-x

[16] Alaoui, N., Adamou-Mitiche, A.B.H., Mitiche, L. (2019). Effective hybrid genetic algorithm for removing salt and pepper noise. IET Image Processing, 14(2): 289-296. https://doi.org/10.1049/IET-IPR.2019.0566

[17] de Paiva, J.L., Toledo, C.F., Pedrini, H. (2016). An approach based on hybrid genetic algorithm applied to image denoising problem. Applied Soft Computing, 46: 778-791. http://dx.doi.org/10.1016/j.asoc.2015.09.013

[18] de Paiva, J.L., Toledo, C.F., Pedrini, H. (2015). A hybrid genetic algorithm for image denoising. In 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 2444-2451. http://dx.doi.org/10.1109/CEC.2015.7257188

[19] Zosso, D., Bustin, A. (2014). A primal-dual projected gradient algorithm for efficient Beltrami regularization. Computer Vision and Image Understanding, pp. 14-52.

[20] Chang, S.G., Yu, B., Vetterli, M. (2000). Adaptive wavelet thresholding for image denoising and compression. IEEE Transactions on Image Processing, 9(9): 1532-1546. http://dx.doi.org/10.1109/83.862633