Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(11): 32578–32628.
DOI: 10.3934/math.20241561
Received: 05 September 2024
Revised: 30 October 2024
Accepted: 04 November 2024
Published: 18 November 2024

Research article

L0-Norm based Image Pansharpening by using population-based algorithms

Mehmet Akif Günen[1], **María-Luisa Pérez-Delgado**[2,*]**and Erkan Beşdok**[3]

[1] Department of Geomatics Engineering, Faculty of Engineering and Natural Sciences, Gümüşhane University, 29100, Gümüşhane, Türkiye

[2] Department of Computer Science and Automatics, Universidad de Salamanca, Escuela Politécnica Superior de Zamora, Av. Requejo, 33, Zamora, 49022, Spain

[3] Engineering Faculty, Department of Geomatics Engineering, Erciyes University, Kayseri, Türkiye

* **Correspondence:** Email: mlperez@usal.es; Tel: +34980545000.

Abstract: Earth observation satellites capture panchromatic images at high spatial resolution and multispectral images at lower resolution to optimize the use of their onboard energy sources. This results in a technical necessity to synthesize high-resolution multispectral images from these data. Pansharpening techniques aim to combine the spatial detail of panchromatic images with the spectral information of multispectral images. However, due to the discrete nature of these images and their varying local statistical properties, many pansharpening methods suffer from numerical artifacts such as chromatic and spatial distortions. This paper introduces the L0-Norm-based pansharpening method (L0pan), which addressed these challenges by maximizing the number of similar pixels between the synthesized pansharpened image and the original panchromatic and multispectral images. L0pan was optimized using a population-based colony search algorithm, enabling it to effectively balance both chromatic fidelity and spatial resolution. Extensive experiments across nine different datasets and comparison with nine other pansharpening methods using ten quality metrics demonstrated that L0pan significantly outperformed its counterparts. Notably, the colony search algorithm yielded the best overall results, highlighting the algorithm's strength in refining pansharpening accuracy. This study contributed to the advancement of pansharpening techniques, offering a method that preserved both chromatic and spatial details more effectively than existing approaches.

Keywords: Pansharpening; image fusion; Population-based algorithms; Colony Search Algorithm
Mathematics Subject Classification: 68410, 68T05, 68T20, 68W50, 94a08

1. Introduction

Pansharpening is an image fusion and super resolution technique that combines a high-resolution panchromatic image, PAN, with a lower-resolution multi-spectral image, MSI, to synthesize a single high-resolution, multispectral image. This process enhances the spatial resolution of the multispectral image using the detail from the panchromatic image. Earth observation satellites take high-resolution panchromatic images and low-resolution color images to save energy and extend their lifetime. This method increases energy efficiency while allowing the acquisition of high-resolution color images. The acquired images are then combined using a pansharpening method to produce images with both high spatial and spectral resolution. Pansharpening methods aim to better preserve both the detail and color information of the images by combining the advantages of multispectral and panchromatic images [1]. Multispectral images provide rich spectral information, while panchromatic images provide high spatial resolution. The combination of these two image types is crucial in remote sensing applications. The effectiveness of this method in both academic research and industrial applications greatly facilitates the analysis and interpretation of images. High-resolution imagery is used in a variety of applications, such as crop health monitoring in agriculture, detailed analysis of structures and roads in urban planning, tracking environmental changes, and more. These images make it possible to detect even small-scale changes, enabling more accurate and detailed analysis. For example, these methods offer great benefits in critical areas such as early detection of diseases in vegetation, water resource management or monitoring the effects of natural disasters [2, 3].

Given the advantages of pansharpening, the development of more efficient and effective pansharpening methods is a scientific and industrial necessity. Different pansharpening techniques in use today are based on different algorithms and mathematical models. Pansharpening methods can be classified into several categories based on their underlying approaches and techniques. Two prominent classifications are component substitution (CS) methods and multi-resolution analysis (MRA) methods. CS methods involve transforming the multispectral and panchromatic images into a new component space, where the high-resolution spatial details from the panchromatic image can be injected into the multispectral image. After substitution, the images are transformed back to the original space. This approach aims to enhance the spatial resolution of the multispectral image while preserving its spectral properties. Common CS methods include the intensity-hue-saturation (IHS) transformation and the principal component analysis (PCA). These methods are popular due to their simplicity and efficiency, but they may sometimes result in spectral distortion. MRA methods, on the other hand, decompose the images into different frequency components using techniques such as wavelet transforms or Laplacian pyramids. The high-frequency components of the panchromatic image, which contain detailed spatial information, are then fused with the low-frequency components of the multispectral image. This approach allows for the preservation of both spatial and spectral information, leading to a more accurate and visually appealing pansharpened image. MRA methods are generally more sophisticated and computationally intensive compared to CS methods but tend to produce better quality results in terms of maintaining spectral integrity and spatial detail [4–6].

Each pansharpening method has its own advantages and disadvantages. Therefore, the need to research and apply new and improved pansharpening methods continues, both in academic circles and in industrial applications. A suitable pansharpening method should possess certain characteristics, including spectral accuracy, spatial detail accuracy, computational efficiency, and enhanced interpretability. It should maintain the original spectral properties of the multispectral image while accurately conveying the spatial details of the panchromatic image. It should optimize both spectral and spatial resolution and balance the two. Likewise, it should be fast and efficient when working with large datasets, and the results obtained should be easily interpretable by the user [7, 8]. Among the most commonly used pansharpening methods in the literature are the IHS, PCA, Brovey transformation, Gram-Schmidt transformation, and Wavelet transformation. The IHS method converts red-green-blue (RGB) images to the IHS space, combines the high-resolution component with the panchromatic image, and then obtains the final image with an inverse transformation [9–11]. The PCA method performs principal component analysis of multispectral images and combines them with the panchromatic image [12]. The Brovey transformation employs a linear combination of multispectral and panchromatic images to create a high-resolution image [13]. The Gram-Schmidt transformation is based on rearranging the spectral components and combining them with the panchromatic image [14]. The wavelet transformation involves separating images into different frequency components and combining these components with the panchromatic image to obtain high-resolution images [15]. Pansharpening is a crucial technique in remote sensing that enables more detailed and accurate analyses by enhancing both the spatial and spectral resolution of images. A plethora of pansharpening methods is available in the literature, each with its distinctive advantages and application areas [16].

This article presents a new multiparameter and multimodal pansharpening method: L0-Norm-based pansharpening method (L0pan). The analytical structure of L0pan is defined using a thresholded L0-Norm-based objective function. In addition, the structural parameters of the optimization method are calculated using population-based algorithms.

Comparing two images on a pixel-by-pixel basis has a multimodal nature due to the discrete nature of the images. Therefore, it is challenging to perform a robust comparison using pixel-by-pixel comparison-based image quality measures. Using a thresholded L0-Norm-based quality metric in solving multimodal problems facilitates obtaining relatively robust results, thanks to the outlier exclusion feature of this approach. The L0-Norm returns the number of nonzero differences in the sequence of pixel-by-pixel differences. Images are continuously affected by low-level noise of varying nature and amplitude due to chaotic thermal fluctuations in the operational environments of sensors. Generally, in images with high jitter values, such as satellite images, low-amplitude numerical oscillations in pixel values do not significantly affect the visual quality of the image as perceived by the human eye. In images with high jitter values, changes in pixel values greater than ±10 intensity levels can be easily detected by a healthy human eye.

Parametric pansharpening problems are typically multimodal due to their multiparameter nature. Robust solutions to these problems require calculating the optimal values of the relevant parameters. Classical optimization methods rely on the use of the functional gradient of the problem and thus assume that the obtained solution is optimal. However, the pansharpening problem is based on image processing, where images are defined by pixel values in discrete space. This means that images, even at high resolution, are numerically discontinuous. Therefore, a direct functional gradient cannot be produced for images. In the image processing community, approximate solutions such as numerical gradients are commonly used to solve related optimization problems. However, the efficiency of numerical gradients decreases as the image size increases. Relatively, numerical gradients are inefficient for solving the pansharpening problem in large 2D images. Consequently, in the past decade, the use of population-based optimization methods, which do not require a functional gradient and yield much more efficient results than numerical gradient methods, has become widespread in image processing applications. The advantages provided by the numerical collective intelligence of swarm-based methods make it possible to calculate the values of the structural parameters in parametric pansharpening problems with high robustness and accuracy.

Swarm intelligence algorithms are inspired by the collective behavior of decentralized, self-organizing systems in nature. These algorithms have proven effective in solving complex optimization problems. The particle swarm optimization (PSO), the ant colony optimization and the artificial bee colony algorithm are among the first methods based on this idea, although many more have been proposed in recent years [17, 18]. These algorithms use simple agents that interact locally to achieve global optimization, and the operations performed can balance exploration and exploitation. Cooperative interaction among agents enables faster convergence to optimal solutions. The algorithms are highly parallelizable, making them suitable for large-scale distributed systems. Their decentralized nature makes them fault-tolerant in real-world applications. They also exhibit strong scalability, handling high-dimensional search spaces effectively. The stochastic nature of these algorithms provides robustness in dynamic environments. Swarm intelligence algorithms have been successfully applied in fields like robotics, data mining, and network optimization. Recent developments have focused on hybrid approaches, combining swarm intelligence with genetic algorithms. Hybrid swarm-evolutionary algorithms leverage the strengths of both approaches for improved performance. Swarm-based evolutionary computing methods partially exhibit characteristics of collective intelligence. Thanks to this feature, these algorithms can more easily escape from local optima and can access solutions relatively closer to the global optimum. Research continues to refine these algorithms for higher accuracy and efficiency in global optimization tasks.

Five population-based methods were considered to optimize the structural parameters of L0pan: four swarm-based methods – namely, PSO [19], the colony-based search algorithm (CSA) [20], teaching-learning-based optimization algorithm (TLBO) [21], and fruit fly optimization algorithm (FOA) [22, 23] – alongside one differential evolution method, adaptive differential evolution with optional external archive (JADE) [24]. These five methods were selected due to their efficacy in solving complex numerical problems. PSO, TLBO, FOA, and JADE are methods widely used in solving numerical optimization problems. PSO has provided efficient solutions for countless image processing problems. TLBO and FOA are preferred for their highly effective problem-solving capabilities and impressive swarm-based structural features. JADE, among differential evolution methods, stands out as a state-of-the-art method with superior problem-solving abilities. CSA is a relatively new swarm-based evolutionary method. It demonstrates high robustness and can effectively solve different types of problems. Additionally, CSA is not highly sensitive to initial conditions and is practically considered a universal swarm-based method.

The innovations provided by L0pan are detailed below:

1. L0pan synthesizes a pansharpened image by preserving color information supplied by the MSI and brightness details from the PAN at a high level. Since the analytical structure of L0pan employs a thresholded L0-Norm technique, it effectively avoids outlier data. This approach is unique in the literature and grants L0pan advanced capabilities in both color and spatial data preservation.
2. L0pan is a parametric method. Its structural parameters govern the degree of fusion between respective image bands, and the optimal numerical values for these parameters can be calculated

through an evolutionary computing approach. In the experiments conducted for this study, the optimal values for the structural parameters of L0pan were determined with relative success using a swarm-based method known as CSA.

3. The analytical structure of L0pan is represented through an objective function utilizing the L0-Norm and a newly developed image quality metric, fidelity-deformation (FD). FD is designed to optimize the similarity of the pansharpened image to both the original MSI and PAN images simultaneously.
4. The effectiveness of L0pan in pansharpened image synthesis, based on the FD metric, has been rigorously analyzed and statistically compared with solutions from widely recognized, relatively advanced classical methods.

The remainder of this document is structured as follows. Section 2 describes the outline of the pansharpening method proposed in the article, while Section 3 describes the population-based methods that have been combined with the L0pan method. Then, Section 4 presents the computational experiments performed with the new method and the discussion of the results. Finally, Section 5 includes the conclusions of the article.

2. L0-Norm-based pansharpening method (L0pan)

The number of cells that detect color in the human eye is significantly lower than those that detect brightness. The human brain combines color and brightness information to achieve high-resolution color vision. This concept also applies to the efficient processing of optical Earth observation satellite data. Satellites capture multispectral image data at relatively low spatial resolution to efficiently use on-board energy resources, while panchromatic images, containing only brightness information, are captured at high spatial resolution. These multispectral and panchromatic images are then fused using pansharpening methods to synthesize a high spatial super resolution color image, known as a pansharpened image. The pansharpened image synthesis process enhances the technical accuracy of data processing in many remote sensing applications and contributes to energy efficiency, thus extending the operational lifespan of expensive optical Earth observation satellites. The ongoing technical and economic demands in the field of remote sensing serve as the primary motivation for developing more efficient pansharpening methods.

In this section, the framework of the L0-Norm-based pansharpening method is introduced, and its main components are detailed. The L0pan method is based on the concept of L0-Norm maximization, which aims to preserve the structural details and enhance the spatial resolution of multispectral images while maintaining their spectral integrity. The following steps outline the process:

1. Image Preprocessing: The initial step involves preprocessing the multispectral and panchromatic images to ensure they are aligned and have the same dimensions. This may include geometric corrections, resampling, and histogram matching.
2. L0-Norm Maximization: The core of the L0pan method is the maximization of the L0-Norm, which promotes sparsity in the gradient domain. This step enhances the high-frequency details in the multispectral images.
3. Fusion Process: The enhanced details from the panchromatic image are injected into the multispectral images using an optimization-based approach. The objective function is designed to balance the trade-off between spatial and spectral quality.

4. Post-Processing: The final step involves post-processing to remove any artifacts introduced during the fusion process and to ensure the output images have high visual quality and are free of distortions.

In this article, the variables PAN and MSI are used to describe 8-bit panchromatic and multispectral images, respectively. Each spectral band of PAN and MSI has a pixel size of $M \times L$, as indicated in Eq (1).

$$[M \times L] \leftarrow size(PAN) \equiv size(MSI) \tag{1}$$

The multispectral images used in the experiments have only red, green, and blue bands. Nevertheless, the proposed pansharpening method can be readily adapted to more bands within the MSI, thereby offering enhanced analytical flexibility. The initial step of the L0pan algorithm is to normalize the image pixel values within the range [0, 1] using Eq (2).

$$[pan0, msi0] = [PAN/255, MSI/255] \tag{2}$$

The normalized panchromatic image (pan) and the normalized multispectral image (msi) are obtained using Eq (3), where $\mu^{(pan0)}$, $\mu_k^{(msi0)}$, $\sigma^{(pan0)}$, and $\sigma_k^{(msi0)}$ are computed by Eqs (4), (5), (6) and (7), respectively, and k is an integer value between 1 and 3.

$$[pan, msi_k] = \left[\frac{pan0 - \mu^{(pan0)}}{\sigma^{(pan0)}}, \frac{msi0 - \mu_k^{(msi0)}}{\sigma_k^{(msi0)}} \right] \tag{3}$$

$$\mu^{(pan0)} = \left[\frac{1}{M \cdot L} \sum_{i=1}^{M} \sum_{j=1}^{L} pan0_{i,j} \right] \tag{4}$$

$$\mu^{(msi0)} = \left[\frac{1}{M \cdot L} \sum_{i=1}^{M} \sum_{j=1}^{L} msi0_{i,j,k} \right] \tag{5}$$

$$\sigma^{(pan0)} = \sqrt{ \frac{1}{M \cdot L} \sum_{i=1}^{M} \sum_{j=1}^{L} \left(pan0_{i,j} - \mu^{(pan0)} \right)^2 } \tag{6}$$

$$\sigma^{(msi0)} = \sqrt{ \frac{1}{M \cdot L} \sum_{i=1}^{M} \sum_{j=1}^{L} \left(msi0_{i,j,k} - \mu_k^{(msi0)} \right)^2 } \tag{7}$$

The chromatically reorganized images, ms and ps, are calculated using Eqs (8) and (9), respectively, where $h = \{1, 2, 3, ...8\}$. The expression $h[c \in \{1 : 8\}]$ represents the c^{th} element of h, for example, $h[2] = 2$.

$$ms_s = X_{h[2s-1]} \cdot (msi_s + X_{h[2s]}) \mid s = \{1, 2, 3\} \tag{8}$$

$$ps = X_{h[2s-1]} \cdot (pan + X_{h[2s]}) \mid s = 4 \tag{9}$$

The synthesized pansharpened image (PSI) is then calculated by Eq (10), where $\lceil\ \rceil$ denotes the ceiling function and PSI_s denotes the s^{th} spectral band of PSI.

$$PSI_s = \lceil 255 \cdot (ps + ms_s) \rceil \mid s = \{1, 2, 3\} \tag{10}$$

Pixel values must belong to the integer-valued Galois space, ranging from 0 to 255. Therefore, we used the ceiling function in Eq (10) to truncate numerical values that exceed this range. See Eq (8) for ms_s used in Eq (10). Since $max(s) = 4$, there are eight X values that need to be optimized for use in Eqs (8) and (9).

The common L0-Norm value between PSI, MSI, and PAN was calculated using Eq (11).

$$err_{L0} = \frac{1}{M \cdot L} \sum (|PSI - MSI| \leq T) + \frac{1}{M \cdot L} \sum (|PSI - PAN| \leq T) \mid T = 10 \tag{11}$$

Common mean squared error (MSE) values between PSI and MSI, and between PSI and PAN were calculated using Eq (12), where the $MSE(a, b)$ function returns the MSE value for input variables (a, b). The MSE function is described using Eq (13).

$$err_{MSE} = MSE(PSI, MSI) + MSE(PSI, PAN) \tag{12}$$

$$MSE(a, b) = \frac{1}{u} \sum_{i=1}^{u} (a_u - b_u)^2 \tag{13}$$

In this article, the sum of $-err_{L0}$ and err_{MSE} defines a new image quality metric called FD, as expressed in Eq (14). A pansharpening method having a relatively low FD value indicates that the method effectively preserves both chromatic (i.e., color) and spatial (i.e., image primitives or details) information at a high level in the pansharpened image. FD maximizes the number of pixels that preserve chromatic information for a specific threshold value, T, in PSI and ensures the preservation of spatial information. Because of the inherent properties of human visual system, the numerical value of hyperparameter $T = 10$ has been used in the experiments performed in this article.

$$FD = -err_{L0} + err_{MSE} \tag{14}$$

The analytical model for the production process of L0pan is described using Eq (15).

$$\underbrace{argmin}_{X_{\{1:8\}}} FD \tag{15}$$

L0pan was developed to provide a scientific contribution to meeting the technical and economic needs for a relatively more efficient pansharpening method. The structure of L0pan, defined by Eq (15), is based on the L0-Norm, making L0pan less sensitive to outlier data. The technical analysis of the results obtained from detailed experiments demonstrates that the pansharpened image synthesized by L0pan effectively preserves and fuses the information provided by the MSI and PAN at a high quality. Images are discrete signals expressed in integer-valued matrices. Therefore, they do not have continuous derivatives. While numerical derivative-based optimization methods can be somewhat efficient in simple operations such as edge detection, they are inefficient in complex image processing operations like pansharpening. For this reason, the best way to calculate the values of

$X_{\{1:8\}}$ is to use optimization methods based on evolutionary computation. Although many experimental evolutionary computation methods have been introduced in the literature, few are able to sufficiently avoid local solutions and have a better and relatively faster convergence toward a global solution for hybrid and complex problems. For our research, five population-based methods commonly used to solve optimization problems related to image processing have been selected. The selected methods (CSA, TLBO, PSO, FOA, and JADE) are described in the following section. These methods were used to solve the optimization problem defined in Eq (15) and calculate the values of $X_{\{1:8\}}$. In this study, for the selected population-based method, the L0-Norm operator returns the number of nonzero differences among the absolute values of the related pixel-by-pixel differences that are smaller than the threshold value, T. This allows minimizing chromatic and spatial distortions in the pansharpened image (PSI) synthesized by the population-based method, depending on the amplitude of T. The structural parameter values of the L0pan method proposed in this article are optimized based on the relevant $T = 10$ hyperparameter.

The five population-based methods were used to optimize the structural parameters of L0pan. Computational experiments showed that the variant that uses CSA obtains better results than the other four variants. Thanks to its unique L0-Norm usage strategy, L0pan combined with CSA can relatively better preserve chromatic and spatial data compared to comparison methods.

3. Population-based methods

Equation (15) defines the objective function of an original pansharpening method with high chromatic and spatial detail-preserving capability. Various evolutionary computing methods can be employed to calculate the necessary X values for synthesizing PSI using Eq (15). There are many evolutionary computing methods available for solving single-objective, multivariate, real-valued numerical problems. Evolutionary computing methods are stochastic global search tools. Differential evolution-based evolutionary computing algorithms produce new trial solutions by performing numerical interactions among randomly selected vectors from the existing solution vectors using deterministic equations. Swarm-based evolutionary computing methods tend to exhibit relatively more intelligent behaviors. Swarm methods typically determine which numerical interactions among the available solution vectors could be efficient based on an elitist probability model or a bio-interaction model to generate trial solutions. In this study, four different swarm computing methods – PSO, TLBO, FOA, and CSA – and a differential evolution-based method – JADE –, were used to calculate the X values that define the solution space of Eq (15). PSO, TLBO, and FOA are successful swarm methods with proven problem-solving capabilities and structural parameters used in many applications. CSA is a relatively new universal swarm computing method that does not have structural parameters determined by random processes. Instead, it is an inherently universal swarm computing method without practical structural parameters. JADE is a highly successful differential evolution optimizer whose elitist nature provides it with swarm behavior characteristics. As CSA is a universal swarm method, it does not require a trial-and-error process to set the initial values of its inherent parameters. In contrast, PSO, TLBO, and JADE require the adjustment of structural parameters. This process is quite time-consuming, and the relevant parameters are empirically determined according to the literature. In the experiments conducted in this study, CSA provided solutions that were relatively better than those provided by other evolutionary computing methods. Therefore, the CSA solution was accepted as the reference solution. The other evolutionary optimizer methods used in this study (i.e., PSO, TLBO, FOA, and JADE) were used as comparison methods against CSA. Below, the technical characteristics of CSA and the other comparison methods are presented. Before describing these methods, several basic concepts referred to in said description are presented.

Optimization aims to determine the numerical values of the parameters of a problem that best satisfy a predefined objective function. Let us consider an optimization problem defined in a solution space of dimension D. The problem has an associated function, $f(x)$, called the objective function, where $x \in \mathbb{R}^D \mid [1 \times D] \leftarrow size(x)$ is a vector that represents a feasible solution to the problem. The objective of a solution method is to find the solution that generates the best value of the objective function (that is, the maximum value for a maximization problem, and the minimum value for a minimization problem). The methods considered in this article use a population of solutions to try to obtain the best possible solution to a problem. Let N denote the population size. Each algorithm starts by defining a set of N feasible solutions that define the initial individuals in the population. In general, initial solutions take random values from the problem solution space, $low \leq x \leq up \mid x \in \mathbb{R}^D$. low and up are vectors of size D that include the minimum and maximum allowed values, respectively, for each component of a solution vector. Each new solution generated by an algorithm must always fit within this interval. The quality or fitness of a solution is calculated by applying the objective function of the problem. The solution of this initial set that has the best fitness is selected as the initial solution obtained by the population for the problem of interest. Then, each algorithm applies an iterative process that updates the set of solutions, trying to obtain solutions that improve over the course of iterations. The iterative process can be applied a predetermined number of times ($TMAX$), or it can be concluded when the solution converges to a predefined error. The final solution defined by the algorithm is the best among those found throughout the iterations (the one with the best value of the objective function).

3.1. *Particle swarm optimization*

PSO is a very popular swarm-based method proposed by Kennedy and Eberhart, inspired by the movement of a flock of birds [17]. The movement of each bird in a flock is conditioned not only by its previous movement, but also by the movement of the other birds in the group and the individuals that lead the group. PSO is one of the most popular swarm-based algorithms. To solve a problem, the method considers a set of N particles. The position of each particle represents a feasible solution to the problem. The quality of a particle i is determined by the fitness of its position, fit_i, which is calculated by applying the objective function of the problem. At each iteration t of the algorithm, each particle i has a position $x_i(t)$ and a velocity $v_i(t)$. Furthermore, the particle stores its personal best position $pbest_i(t)$, which is the position with the best fitness among those associated with the particle until iteration t. The solution to the problem defined by the swarm is the global best solution, $gbest$, which is the personal best position with the best fitness among the N particles.

The initial stage of the algorithm defines the initial values of the variables associated with each particle i. The initial position of the particle, $x_i(0)$, takes random values from the problem solution space. The initial velocity of the particle, $v_i(0)$, can take random values from a predetermined interval or can be set to 0. The value $x_i(0)$ is stored as the initial personal best position of the particle ($pbest_i(0) = x_i(0)$). Finally, the fitness of the particle is computed ($fit_i = f(x_i(0))$). To complete the operations of the initial stage, the position of the particle with the best fitness defines the initial value of $gbest$. After this, an iterative process is applied to try to improve the solutions associated with the particles. At each iteration, the velocity of each particle is updated by Eq (16), where ϵ_1 and ϵ_2 are vectors including random values in $[0, 1]$. On the other hand, ω, c_1 and c_2 are the parameters of the PSO method that determine the relative influence of each addend. c_1 determines the effect of the previous experience of the particle, and c_2 determines the effect of the previous experience of the swarm.

$$v_i(t) = \omega v_i(t-1) + c_1\epsilon_1(pbest_i(t-1) - x_i(t-1)) + c_2\epsilon_2(gbest - x_i(t-1)) \tag{16}$$

After updating the velocity, the new position of each particle i, $x_i(t)$, is computed by Eq (17).

$$x_i(t) = x_i(t-1) + v_i(t) \tag{17}$$

The fitness of the new position is computed before checking whether the personal best position of the particle needs to be updated. If the fitness of $x_i(t)$ is better than the fitness of $pbest_i(t-1)$, then the value of $x_i(t)$ is taken as the new value of $pbest_i(t)$; otherwise, $pbest_i(t)$ keeps its previous value. The last operation of each iteration checks whether $gbest$ must be updated. If the new personal best position of some particle j improves the quality of the current value of $gbest$, then the value $pbest_j(t)$ replaces the current value of $gbest$.

Algorithm 1 PSO

```
INPUT: $N$
OUTPUT: $gbest$

Define $x_i(0)$ and $v_i(0)$ for each particle $i$
Set $pbest_i(0) = x_i(0)$ for each particle $i$
Compute $fit_i$ for the position $x_i(0)$ of each particle $i$
Identify the particle $j$ with the best fitness and set $gbest = pbest_j(0)$
for $t = 1$ to $TMAX$ do
    Update the velocity of each particle $i$ by Eq (16)
    Update the position of each particle $i$ by Eq (17)
    Compute $fit_i$ for the position $x_i(t)$ of each particle $i$
    for each particle $i$ do
        if $fit_i$ is better than the fitness of $pbest_i(t-1)$ then
            Set $pbest_i(t) = x_i(t)$
        end if
    end for
    Identify the particle $j$ that has the $pbest_j(t)$ with the best fitness
    if the fitness of $pbest_j(t)$ is better than the fitness of $gbest$ then
        Set $gbest = pbest_j(t)$
    end if
end for
```

3.2. Adaptive differential evolution algorithm

JADE is a differential evolution algorithm that uses a novel mutation strategy as an adaptive strategy [24]. The algorithm works on a population of N individuals that represent solutions to the problem.

An initial population is defined, and successive iterations of the algorithm define new generations of the population. Each individual i represents a solution $x_i(t)$. It also has a crossover ratio value, CR_i, and a scaling factor, F_i, which are randomly generated in each generation of the algorithm. CR_i is a random value from a normal distribution of mean μ_{CR} and standard deviation 0.1. F_i is a random value from a Cauchy distribution with location parameter μ_F and scale parameter 0.1. The sets S_F and S_{CR} are used to update the values μ_F and μ_{CR}, respectively. S_{CR} includes the successful CR_i values at the current generation, and S_F includes the successful F_i values at the current generation, for $i = 1, ..., N$.

The algorithm uses the set A to include the archived inferior solutions. This set is initially empty. The operations of the algorithm include in the set each solution $x_i(t-1)$ that is improved at generation t. Before defining the next generation of the population, randomly selected solutions are eliminated from A until the set includes N elements as a maximum.

Initially, the algorithm sets $\mu_{CR} = 0.5$ and $\mu_F = 0.5$ and considers the set A to be empty. In addition, a random initial population is created. After this, an iterative process is applied that defines successive generations of the population. At the beginning of each iteration, the sets S_{CR} and S_F are empty. Mutation and crossover operations are applied to each individual to try to improve the solution it represents.

Equation (18) is used to compute a candidate solution x'_i (mutation vector) for each individual i. This equation uses the position of three randomly selected individuals. $best$ represents an individual taken from the subset of the best individuals in the population, whose size is defined by the parameter p. $r1$ is an individual of the current population, different from i. $r2$ is selected from the current population or the set A , and it must be different from $r1$ and i .

$$x'_i = x_i(t-1) + F_i(x_{best}(t-1) - x_i(t-1)) + F_i(x_{r1}(t-1) - x_{r2}(t-1)) \tag{18}$$

Then, the crossover operator is applied to define each element of the solution u_{ij}, with $j = 1, ..., D$. Let ε_1 be a random integer in the interval $[0, D]$ and ε_2 be a random real in the interval $[0, 1]$. Eq (19) is used to compute each u_{ij} value.

$$u_{ij} = \begin{cases} x'_{ij} & \text{if } \varepsilon_1 = 1 \text{ or } \varepsilon_2 < CR_i \\ x_{ij}(t-1) & \text{in other case} \end{cases} \tag{19}$$

After this, the fitness of $x_i(t-1)$ and u_i is compared. If the first is better than the second, $x_i(t)$ maintains its previous value. Otherwise, $x_i(t-1)$ is included in the set A, CR_i is included in the set S_{CR}, F_i is included in the set S_F, and $x_i(t) = u_i$.

Once the previous operations have been applied to all the individuals, the set A is processed to eliminate solutions randomly until the set includes a maximum of N solutions. After this, μ_{CR} and μ_F are updated by Eqs (20) and (21), respectively, where $mean_A(S_{CR})$ is the arithmetic mean of the values in the set S_{CR}, $mean_L(S_F)$ is the Lehmer mean of the values in the set S_F, and c is a positive constant.

$$\mu_{CR} = (1-c)\mu_{CR} + c\, mean_A(S_{CR}) \tag{20}$$

$$\mu_F = (1-c)\mu_F + c\, mean_L(S_F) \tag{21}$$

The final solution obtained by the algorithm is the solution found throughout the iterations that provides the best value of the objective function. JADE is a stochastic evolutionary search method.

Therefore, its problem-solving success does not inherently exhibit excessive dependence on the initial values of its parameters. In evolutionary algorithms, the values of structural parameters are typically determined empirically through a time-consuming trial-and-error process.

Algorithm 2 JADE

```
INPUT: $N$
OUTPUT: $gbest$

Sets $\mu_{CR} = 0.5$ and $\mu_F = 0.5$
Set $A = \emptyset$ (empty set)
Define $x_i(0)$ for each individual $i$
Compute $fit_i$ for the solution $x_i(0)$ of each individual $i$
Identify the solution $j$ with the best fitness and set $gbest = x_j(0)$
for $t = 1$ to $TMAX$ do
    Set $S_{CR} = \emptyset$ and $S_F = \emptyset$
    for each individual $i$ do
        Compute $x'_i$ by Eq (18)
        Compute $u_i$ by Eq (19)
        Compute the fitness of $u_i$, $fitu_i$
        if $fit_i$ is worse than $fitu_i$ then
            Include $x_i(t-1)$ in the set $A$
            Include $CR_i$ in the set $S_{CR}$
            Include $F$ in the set $S_F$
            Set $x_i(t) = u_i$ and $fit_i = fitu_i$
        end if
    end for
    Randomly remove solutions from $A$ until the set includes at most $N$ solutions
    Update $\mu_{CR}$ by Eq (20)
    Update $\mu_F$ by Eq (21)
    Identify the individual $j$ with the best fitness
    if $fit_j$ is better than the fitness of $gbest$ then
        Set $gbest = x_j(t)$
    end if
end for
```

3.3. Teaching-Learning-Based optimization algorithm

TLBO imitates the effect of the influence of a teacher on learners [21]. It considers a population of solutions to look for the best solution. The population represents a set of learners, and the operations of the algorithm include two main phases: the teacher phase and the learner phase. In the first phase, the learner learns from the teacher, while in the second he or she learns from other learners. The population size N indicates the number of learners, and the D dimensions of the solution space indicate the subjects (courses) offered. $x_i(t)$ represents the learner i at iteration t. The result of the learner is equivalent to the quality or fitness of the solution associated with this learner ($f(x_i(t))$).

Algorithm 3 TLBO

```
INPUT: $N$
OUTPUT: $gbest$

Define $x_i(0)$ for each learner $i$
Create $P$ including the vectors $x_i(0)$, $i = 1, ...N$, as rows of the array
Compute $fit_i$ for the solution $x_i(0)$ of each learner $i$
Identify the learner $j$ with the best fitness and set $gbest = x_j(0)$
for $t = 1$ to $TMAX$ do
  Select the solution $k$ with the best fitness and set $x_{teacher} = x_k(t-1)$
  Compute the vector $A$ (mean of each row in $P$)
  Set $A_{new} = x_{teacher}$
  Compute $DM$ by Eq (22)
  for each learner $i$ do
    Compute $x'_i$ by Eq (23)
    Compute the fitness of $x'_i$, $f'_i$
    if $f'_i$ is better than $fit_i$ then
      Set $x_i(t) = x'_i$ and $fit_i = f'_i$
    end if
  end for
  for each learner $i$ do
    Select a partner $j$ randomly, with $i \neq j$
    if $fit_i$ is better than $fit_j$ then
      Compute a candidate solution $x'_i$ by Eq (24)
    else
      Compute a candidate solution $x'_i$ by Eq (25)
    end if
    Compute the fitness of $x'_i$ , $f'_i$
    if $f'_i$ is better than $fit_i$ then
      Set $x_i(t) = x'_i$ and $fit_i = f'_i$
    end if
  end for
  Identify the learner $j$ with the best fitness
  if $fit_j$ is better than the fitness of $gbest$ then
    Set $gbest = x_j(t)$
  end if
end for
```

The first operation of the algorithm initializes the solution associated with each individual i, $x_i(0)$, with random values from the solution space. The information is stored in a matrix P with N rows and D columns, where each row corresponds to the information of a learner and each column to the information of a course. The iterative stage of the algorithm first applies the teacher phase and then applies the learner phase. First, the quality of the current solutions is determined by applying the objective function of the problem. This information is used to identify the best solution, k , which will be considered the teacher for the current iteration ($x_{teacher} = x_k(t-1)$). On the other hand, the mean of each row in P is calculated. Let A be the vector of size D that contains the mean values computed.

The teacher tries to shift the mean from A toward $x_{teacher}$, which is considered a new mean for the iteration ($A_{new} = x_{teacher}$). The difference between both means is computed by Eq (22), where ε_1 is a random number in the interval $[0, 1]$ and T_F is a teaching factor that determines the value of the mean to be changed. The value of T_F is randomly selected between 1 or 2.

$$DM = \varepsilon_1(A_{new} - T_F A) \tag{22}$$

A new solution is computed for each individual i by Eq (23). If the quality of the new solution ($f(x'_i)$) improves the quality of the current solution ($f(x_i(t-1))$), the new solution is accepted ($x_i(t) = x'_i$); otherwise, it is rejected ($x_i(t) = x_i(t-1)$).

$$x'_i = x_i(t-1) + DM \tag{23}$$

The learner phase is applied after the teacher phase. Learners increase their knowledge by interacting with their peers. A learner learns something new from a partner if that person has more knowledge than him or her. During the learner phase, a partner j is randomly chosen for each learner i, with $i \neq j$. If solution $x_i(t)$ is better than $x_j(t)$, Eq (24) is used to compute a new solution for the learner i; otherwise, Eq (25) is used for this purpose. In these equations, ε_2 is a random number in the interval $[0, 1]$.

$$x'_i = x_i(t) + \varepsilon_2(x_i(t) - x_j(t)) \tag{24}$$

$$x'_i = x_i(t) + \varepsilon_2(x_j(t) - x_i(t)) \tag{25}$$

If the quality of the new solution ($f(x'_i)$) improves the quality of the current solution ($f(x_i(t))$), the new solution is accepted ($x_i(t) = x'_i$); otherwise, it is rejected ($x_i(t)$ is not modified). The iterative process continues until the predefined number of generations is achieved. The final solution found by the algorithm is the best of the solutions obtained throughout the iterations.

3.4. Fruit fly optimization algorithm

This algorithm is inspired by the foraging behavior of fruit flies [22]. These flies are attracted to ripened or fermenting food through their sensing and perception characteristics, especially osphresis and vision. The foraging process of the fruit fly can be summarized as follows. To start, it smells the food source through the osphresis organ and flies to that place. After approaching the location of food, the fly uses sensitive vision to find food and the location where other fruit flies gather. With this information, it flies toward that direction.

Although the original FOA was described for two-dimensional problems, the same author adapted it to solve high-dimensional problems [23]. The algorithm uses a population of N fruit flies that represent feasible solutions to the problem. Additionally, it uses the fruit fly swarm location $\Delta = (\delta_1, ..., \delta_D)$, which represents the solution obtained by the algorithm. It should be noted that the variable Δ of this algorithm represents the same information as the variable $gbest$ used in the previous algorithms. The fruit fly swarm location is initialized by randomly assigning values from the problem solution space. This initial location represents the initial solution to the problem found by the algorithm. After the initialization phase, the iterative process of the algorithm begins. The first step of this process determines the random direction and distance for the search of food using osphresis. The position of each fruit fly i in the population is computed by Eq (26), where ε is a random value from a uniform distribution.

$$x_{ij}(t) = \delta_j + \varepsilon, \text{ for } j = 1, ..., D \tag{26}$$

The second step of the iterative process is the vision foraging phase, which applies a greedy selection procedure. The smell concentration of each fruit fly i, $Smell_i$, is obtained by applying the objective function to its current position ($Smell_i = f(x_i(t))$). It should be noted that the variable $Smell_i$ of this algorithm represents the same information as the variable fit_i used in the previous algorithms. Then, the individual with the best smell concentration is determined. Let bI denote the index of this individual and bS denote its smell concentration ($bS = Smell_{bI}$). If bS is better than the quality or fitness of the fruit fly swarm location, the current position of the fruit fly bI is used to replace the current fruit fly swarm location by Eq (27).

Algorithm 4 FOA

```
INPUT: $N$
OUTPUT: $\Delta$

Define the initial value of $\Delta$ randomly
for $t = 1$ to $TMAX$ do
    Compute the position $x_i(t)$ of each fly $i$ by Eq (26)
    Compute $Smell_i$ for the position $x_i(t)$ of each fly $i$
    Identify the fly $bI$ with the best smell concentration and set $bS = Smell_{bI}$
    if $bS$ is better than the fitness of $\Delta$ then
        Set $\Delta = x_{bI}(t)$
    end if
end for
```

$$\Delta = x_{bI}(t) \tag{27}$$

The final solution generated by the algorithm is the one corresponding to the fruit fly swarm location. It should be noted that this algorithm does not include special parameters. It is only necessary to define the population size.

3.5. Colony search algorithm

CSA is an evolution-based global minimizer developed to solve single-objective, bounded/unbounded real-valued numerical problems [20]. The algorithm uses two matrices to store feasible solutions: the clan matrix, denoted p, and the pattern matrix or colony, denoted $p0$. Both matrices include D columns, but p includes N rows while $p0$ includes $Z \cdot N$ rows. Therefore, $p0$ contains Z times more random solution vectors (patterns) than the clan matrix. Z is called the population expanding factor, and it is set to 2 in [24]. CSA only deals with the evolution of the patterns of the clan matrix, which consists of randomly selected patterns from the parent population defined by the colony matrix. The mutation process of CSA is combined with the random crossover process, and both processes together are referred to as the morphogenesis process. The first operation of the algorithm initializes the colony $p0$ by assigning random values from the solution space. The initial quality of each vector $p0_i$ is computed by applying the objective function ($fitp0_i = f(p0_i)$). The element that has the best value of $fitp0$ is determined, and the associated pattern is stored as the initial global solution, $gbest$. To complete the initialization stage of the algorithm, the variable *moment* is set to 0, and *initindex* stores the N indices between 1 and N.

After the initialization stage, the algorithm applies the iterative search phase. The first operation of each iteration defines the clan matrix. To do this, a random permutation of the indices between 1 and $Z \cdot N$ is generated and the first N values of the sequence are stored in $index$. The sequence generation is repeated until the resulting sequence is different from $initindex$. When this is achieved, the sequence contained in $index$ is copied to $initindex$. After this, the elements of $p0$ identified in $index$ are copied to p (Eq 28). The operation is completed by copying into $fitp$ the value that indicates the quality of the patterns copied into p (Eq 29). In this case, $p_i \in \mathbb{R}^D$ and $[1, D] \leftarrow size(p_i)$, while $fitp_i \in \mathbb{R}^1$ and $[N, 1] \leftarrow size(fitp)$. The $size(a)$ function used here returns the number of rows and columns in a.

$$p_i = p0_{index_i}\ , \text{ for } i = 1, ..., N \tag{28}$$

$$fitp_i = fitp0_{index_i}\ , \text{ for } i = 1, ..., N \tag{29}$$

The second operation of the iterative process determines the direction scale factor used by the algorithm. The scale factor, *scale*, is defined using Eq (30), where $\oslash$ denotes the Hadamard division operator and $\circ$ denotes the Hadamard multiplication. R_1 and R_2 are two matrices of random values of a uniform distribution $U(0, 1)$, with N rows and c columns. R_3 includes the same type of values as R_1 and R_2, but the matrix has N rows and 1 column. The value of c is set to 1 if $\varepsilon_3 < \varepsilon_4$, and it is set to D in other cases. ε_1, ε_2, ε_3, and ε_4 are random reals with uniform distribution $U(0, 1)$. $U(a, b)$ denotes a continuous uniform distribution, and $U\{a, b\}$ indicates a discrete uniform distribution in discrete mathematics. In addition, $X \sim U(a, b)$ is used to denote that the variable X takes random values from the specified uniform distribution.

$$scale = \begin{cases} (R_1 - 0.05) \oslash (R_2 - 0.50) & \text{if } \varepsilon_1 < \varepsilon_2 \\ sign(R_3 - 0.50) \circ \Phi & \text{in other case} \end{cases} \tag{30}$$

For $c = 1$, the *scale* value is an $N \times 1$ sized column vector. For $c = D$, the *scale* value is an $N \times D$ sized matrix. The variable Φ is an $N \times 1$ sized column vector.

The values of Φ fit a Lévy distribution with position and shape parameters α and β, respectively. Lévy flights allow the algorithm to escape from local optima and access high-quality solutions at different locations in the solution space. In CSA, Φ is defined by Eq (31), where ε is a random real number with distribution $U(0, 1)$; ω is a value of a Gamma distribution $\Gamma(\alpha, k)$, with shape parameter

α and scale parameter k, ($k \sim U\{2,5\}$). α is a vector of size $N \times 1$ of random integers, where each element of α is distributed in $U\{2,5\}$. β is an array of size $N \times 1$, where each element is computed as a^b ($a \sim U\{1,10\}$) and b is the value -1 or 1 (chosen randomly).

$$\Phi = \beta \cdot (\varepsilon + 1) \oslash \omega^{1/\alpha} \tag{31}$$

The third operation of the iterative process computes the morphogenesis control matrix (mutation+crossover) of the algorithm. This operation calculates the binary-valued mutation control matrix, m, which includes N rows and D columns. The elements of this matrix are initially set to zero, and then some elements are set to 1 according to the following process. The process is repeated for each row j, where $j = \{1 : N\}$. First, a random vector-permutation operation is applied to the integer-valued vector $[1 : N]$ to generate indices, ind. Then, k is computed by Eq (32), based on $R_4 \sim U\{0,1\} \mid R_4 \in \mathbb{Z}^1$ and, $(\varepsilon_5)^{\varepsilon_6} \mid \varepsilon_5 \sim U(0,1)$, $\varepsilon_6 \sim U\{2,10\}$, $\{\varepsilon_5, \varepsilon_6\} \in \mathbb{Z}^1$. Finally, the elements of ind that are between positions 1 and $ceil(k \cdot D)$ are selected as S and the elements of the mutation control matrix m_{jS} are set to 1.

$$k = abs(R_4 - (\varepsilon_5)^{\varepsilon_6}) \tag{32}$$

The fourth operation of the iterative process computes the evolutionary direction vector, dx, of CSA using Eq (33). v is an integer value uniformly distributed in the range $[1,3]$ that determines the expression used to define dx, where $dx \in \mathbb{R}^{N,D}$. The notation $p[a]$ is used to indicate that the elements of the vector p are returned in the order indicated by a. For example, if $p = [3,5,7,9]$ and $a = [1,4,2,3]$, then $p[a] = [3,9,5,7]$.

$$dx = \begin{cases} p[v2] - p[v1] & v = 1 \\ p[v1] - p & v = 2 \\ p[v3] - p & v = 3 \end{cases} \tag{33}$$

In Eq (33), $v1$ and $v2$ are the responses of two distinct random vector-permutation processes applied to an integer-valued vector of size N, with $v1 \neq v2$. To define $v3$, first the indices of the elements of p are sorted by decreasing the value of $fitp$; let $index0$ denote this sorted list. Thus, $v3$ is defined using Eq (34), where $SI \sim U\{1, \lceil N/5 \rceil\}$.

$$v3 = index0[SI] \tag{34}$$

The next operation of the iterative process defines the morphogenesis pattern matrix, px, by Eq (35), where $\{R_5, R_6\} \sim U(0,1)$, $[N \times 1] \leftarrow size(R_5) = size(R_6)$, $\varepsilon_7 \sim U\{2,10\}$, and s is computed by Eq (36).

$$px = p + scale \circ m \circ dx + s \circ moment \tag{35}$$

$$s = (R_5 - 0.50) \circ (R_6)^{\varepsilon_7} \tag{36}$$

The values of all the elements in px must be adjusted to the limits established for a feasible solution to the problem. Therefore, each element px_{ij} (with $i = 1,...,N$ and $j = 1,...,D$) is analyzed. If

$px_{ij} < low_j$, then the value is adjusted by Eq (37); on the other hand, if $px_{ij} > up_j$, the value is updated by Eq (38), where $\{\varepsilon_8, \varepsilon_9\} \sim U(0, 1)$ and $\{\varepsilon_{10}, \varepsilon_{11}\} \sim U\{1, 5\}$.

$$px_{ij} = low_j + (\varepsilon_8)^{\varepsilon_9}(up_j - low_j) \tag{37}$$

$$px_{ij} = up_j + (\varepsilon_{10})^{\varepsilon_{11}}(low_j - up_j) \tag{38}$$

Algorithm 5 CSA

```
INPUT: $N$
OUTPUT: $gbest$

Define the initial $p0$
Compute $fitp0_i$ for each row $p0_i$ of $p0$
Identify the row $j$ with the best fitness and set $gbest = p0_j$
Set initial values for $moment$ and $initindex$
for $t = 1$ to $TMAX$ do
    Compute the clan matrix $p$ by Eq (28)
    Copy into $fitp$ the fitness of the patterns copied into $p$ (Eq 29)
    Compute $scale$ by Eq (30)
    Compute the matrix $m$
    Compute $dx$ by Eq (33)
    Compute the matrix $px$ by Eq (35)
    Clip the values of $px$ to the valid interval by Eqs (37) and (38)
    for each row $i$ of $px$ do
        Compute the fitness $fitpx_i$ of the row $px_i$
        if $fitpx_i$ is better than $fitp_i$ then
            Set $p_i = px_i$ and $fitp_i = fitpx_i$
        end if
    end for
    Update $p0$ and $fitp0$ by Eqs (39) and (40)
    Identify the row $k$ of $p0$ with the best fitness and set $gbest = p0_k$
    Update $moment$ by Eq (41)
end for
```

The objective function values of the px patterns are then calculated as $fitpx_i = f(px_i)$. Then, the clan is updated based on this information. The patterns in px which have better objective function value than those in p, are used to replace the corresponding element in p. When an element px_i is used to replace the current value of p_i, the objective value $fitpx_i$ is also used to replace the previous value of $fitp_i$. Once the clan has been updated, the next operation updates the colony, $p0$, using the clan information. This operation considers the list of indices previously used to create the clan, as it determines the colony element that is replaced with each of the N patterns in the clan with Eq (39). When a pattern of the colony is updated, the corresponding objective function value is also updated in Eq (40).

$$p0_{index_i} = p_i \text{ , for } i = 1, ..., N \tag{39}$$

$$fitp0_{index_i} = fitp_i \text{ , for } i = 1, ..., N \tag{40}$$

The global solution of the algorithm is then updated. To do this, the element that has the best value of $fitp0$ is determined; let k be the index of this element. Then, the associated pattern is stored as the new value of the global best solution ($gbest = p0_k$).

At the end of the current iteration, the *moment* value is updated using Eq (41), where R is an array of integer random values in the range [0, 1] with N rows and 1 column.

$$moment = (abs(R) - m) \circ dx \tag{41}$$

Readers can find the CSA implementations in MATLAB at https://github.com/BESDOK/CSA.

3.6. Integration of population-based algorithms into the L0pan method

Algorithm 6 summarizes the operations of L0pan combined with the application of a population-based method. The algorithm takes the PAN and MSI images as input and generates the PSI image as the result of the operations.

Algorithm 6 L0pan

```
INPUT: $PAN$, $MSI$
OUTPUT: $PSI$

Compute $pan0$, $msi0_1$, $msi0_2$, and $msi0_3$ by Eq (2)
Compute $pan$, $msi_1$, $msi_2$, and $msi_3$ by Eq (3)
Apply population-based method ($X$, $N$, $M$, $L$, $PAN$, $MSI$, $pan$, $msi_1$, $msi_2$, $msi_3$ )
Generate $PSI_1$, $PSI_2$, and $PSI_3$ by Eq (10)
```

The third operation of the L0pan algorithm applies one of the five methods described in this section (PSO, JADE, TLBO, FOA, or CSA). This operation is used to calculate the values of $X_{\{1:8\}}$ described in Section 2 (eight values used by the L0pan method that need to be fine-tuned by an optimization method). On the other hand, x denotes a feasible solution for a population-based method, where x is a vector including D components. Therefore, for the problem of interest, a feasible solution x used by a population-based method is a vector of size $D = 8$ including the eight values $X_{\{1:8\}}$ associated with the pansharpening problem (that is, $x = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$).

The algorithms that represent the operations of the PSO, JADE, TLBO, FOA, and CSA methods only include the population size (N) as input information. However, here it is necessary to add several additional parameters that must be passed as input to these algorithms when combined with the L0pan algorithm. These parameters are necessary to calculate the fitness of the solutions handled by the population-based algorithms. On the other hand, we denote X the output generated by the population-based algorithm.

Algorithm 7 shows the operations that must be performed to calculate the fitness of a feasible solution. The information used as input not only includes the feasible solution x, but also additional information about the images that is necessary to calculate the FD by Eq (14).

Algorithm 7 Compute the fitness of a feasible solution

```
INPUT: $x$, $M$, $L$, $PAN$, $MSI$, $pan$, $msi_1$, $msi_2$, $msi_3$
OUTPUT: $FD$

Compute $ms_1$, $ms_2$, and $ms_3$ by Eq (8)
Compute $ps$ by Eq (9)
Compute $PSI_1$, $PSI_2$, and $PSI_3$ by Eq (10)
Compute $err_{L0}$ by Eq (11)
Compute $err_{MSE}$ by Eq (12)
Compute $FD$ by Eq (14)
```

4. Results and discussions

4.1. Characteristics of the tests performed

In this article, the pansharping process has been performed using high resolution satellite images. The QuickBird-2 satellite was launched on October 18, 2001, and is a high-resolution Earth observation satellite. It has a panchromatic resolution of 0.61m and a multispectral resolution of 2.44m. It operates at 450 kilometers altitude and has a swath width of 16.5 kilometers at its lowest point. QuickBird-2's return time extends from 1 to 3.5 days. The satellite operates at a height of 681 kilometers and has a swath width of 15.2 kilometers at nadir. GeoEye-1, which was launched on September 6, 2008, is a very advanced Earth observation satellite. It has a panchromatic resolution of 0.41m and a multispectral resolution of 1.65m. The satellite operates at a height of 681 kilometers and has a swath width of 15.2 kilometers at nadir. GeoEye-1 has a revisit time of less than three days. WorldView-2 has a panchromatic resolution of 0.46m and a multispectral resolution of 1.84m. It operates at 770 kilometers altitude and has a swath width of 16.4 kilometers at its lowest point. The satellite has a return time of around 1.1 days. These satellite images are often excellent for precision mapping, agriculture, land use planning, environmental monitoring, and disaster response. In this study, 3 geo-images of 512×512 dimensions were randomly obtained from each of 3 high-spatial-resolution satellite systems to include different land uses. In this context, the statistics and visuals obtained as a result of the pansharping process of the test images are presented in this section. This article includes results from 9 tests, labeled Test 1 to Test 9, that work on each of the 9 selected images. Geo-images from WorldView-2 have been employed in Tests 1–3, QuickBird-2 in Tests 4–6, and GeoEye-1 in Tests 7–9.

This article compares the five methods described in Section 3 to address the pansharpening problem. To identify the five L0pan variants that use each of these methods, the ending '-PS' will be added to the acronyms already defined for the methods, indicating that the corresponding method is applied to the pansharpening problem. Therefore, the five methods are CSA-PS, TLBO-PS, PSO-PS, FOA-PS and JADE-PS. The population-based methods used in the study were repeated 30 times in 1500 iterations on the same seed, and statistics were generated. The population size considered in all cases was $N = 30$. On the other hand, the values considered for the specific parameters of each method are shown in Table 1.

Table 1. Control parameters of comparison algorithm.

Methods	Control Parameters
CSA-PS	No parameter
TLBO-PS	Teaching factor = [1 2]
PSO-PS	$c1 = c2 = 1.80$, $\omega = 0.60$
FOA-PS	No parameter
JADE-PS	$c \in [5, 20]$, $p \in [5\%N, 20\%N]$

To complete the study, five non-population-based pansharpening methods were also applied. The methods are IHS, wavelet transformation, Brovey transformation, hue saturation value (HSV), and smoothing filter-based intensity modulation (SFIM). These techniques were chosen for improving spatial resolution while maintaining spectral information due to their variety of approaches and demonstrated effectiveness. IHS is renowned for its ease of use and power in vegetative and urban analysis. When it comes to frequency decomposition, wavelet transformation appears, offering finely detailed spatial enhancement. The effectiveness with which Brovey transformation can combine spectral and spatial information is well-known. HSV improves spatial details while preserving color integrity [25]. Finally, SFIM is acknowledged for its capacity to enhance spatial resolution through intensity modulation, which combines these techniques into a comprehensive approach for a strong pansharpening analysis [26].

Figures 1 to 9 show the images of the test set and the resulting images of the process. Each figure contains 12 sub-figures. Sub-figure (a) shows the multispectral band (MSI) and sub-figure (b) shows the panchromatic band (PAN). The remaining sub-figures show the result obtained with each of the 10 methods tested. Tables 2 to 10 show the results obtained for each of the 9 images in the test set. Nine image quality measures (IQM) commonly used in literature related to the topic have been chosen [9, 10, 14, 27]. The tables show the mean ($Mean$) and standard deviation (Std) of each IQM. The IQM used are: the root mean squared error (RMSE), the correlation coefficient (CC), the peak signal-to-noise ratio (PSNR), the error relative global dimensionless synthesis (ERGAS) index, the relative average spectral error metric (RASE), the blind image spatial quality evaluator (BRISQUE), the naturalness image quality evaluator (NIQE), the structural similarity index (SSIM), and the universal image quality index (QAVE). Additionally, the FD defined in Eq (15) is used.

RMSE is the square root of MSE. MSE is the average squared difference between the pixels of the original image and the transformed image. An advantage of RMSE over MSE is that the value is measured in the same units as the response variable. PSNR is the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation [28–30]. ERGAS estimates the spectral quality of the resulting image [31]. This is one of the most popular indexes to assess both spectral and structural fidelity between the original and the transformed image. The spectral distortion introduced during the pansharpening process can be measured using the RASE metric, which is used to evaluate the quality of the images. It computes, on average and over all spectral bands, the relative error between the original multispectral image and the pansharpened image. Better spectral fidelity, or how closely the pansharpened image maintains the original spectral information, is indicated by lower RASE values [31]. The CC metric provides a quantitative assessment of the fidelity of the pansharpening outcome by indicating the degree of correlation between panchromatic and multispectral images. CC takes values in the range $[-1, 1]$. For ease of interpretation in the article, CC results are multiplied by 100. SSIM estimates the similarity between the original and the transformed image [30,32]. It is based on the differences in the luminance, the contrast, and the structure between both images. SSIM takes values in the interval $[0, 1]$. SSIM is close to one when the transformed image has a higher spectral quality. For ease of interpretation in the article, SSIM results are multiplied by 100.

BRISQUE and NIQE are no-reference image quality indices. BRISQUE operates in the spatial domain [33]. The index uses scene statistics of locally normalized luminance coefficients to quantify possible losses of naturalness in the image due to the presence of distortions. NIQE [34] uses measurable deviations from statistical regularities observed in natural images without training on human-rated distorted images and without any exposure to distorted images. It is based on the construction of a quality-aware collection of statistical features based on a simple and successful space-domain natural scene statistic model. QAVE is a comprehensive metric that takes into account both spectral and spatial aspects when assessing the quality of pansharpened images [35]. By using comparisons of luminance, contrast, and structure, it assesses how similar the pansharpened image is to the reference images while maintaining structural details like brightness and contrast. Furthermore, QAVE evaluates spectral fidelity to guarantee that color and intensity from the multispectral image are accurately preserved.

Lower values of RMSE, BRISQUE, NIQE, RASE, FD, and ERGAS indicate better results. On the contrary, larger values of CC, PSNR, SSIM, and QAVE correspond to better resulting images.

4.2. The tests

The discussion of the experimental results begins by describing the results of the 9 tests performed with the 9 images in the dataset. For this analysis, the IQM values presented in Tables 2 to 10 were examined within the 95% confidence limits of the normal distribution.

Table 2 shows the results for the Test 1, applied to the first image of the dataset. It is evident that CSA-PS outperformed IHS, TLBO-PS, PSO-PS and FOA-PS for all the indices, SFIM and Wavelet for 9 indices, and Brovey and HSV for 8 indices. The CSA-PS method produced relatively better results in terms of RMSE, CC, RASE, ERGAS, SSIM, PSNR, BRISQUE, and FD. When all the IQM values in Table 2 are considered together, CSA-PS achieved better results than the comparison methods at a level of 77.78%. CSA-PS and JADE-PS produced similar results. The average value obtained by JADE-PS was slightly better than that obtained by CSA-PS for all indices except QAVE, NIQE and FD. However, when examining the respective standard deviation values, CSA-PS produced statistically more robust results than JADE-PS. CSA-PS is significantly more robust than JADE-PS in terms of FD. Based on FD values, it can be stated that CSA-PS preserved the chromatic and spatial information more successfully and robustly than all comparison methods for Test 1. CSA-PS produced statistically better results than IHS, wavelet and the other population-based methods in terms of QAVE. In terms of NIQE, CSA-PS performed significantly better than IHS and yielded similar results to Brovey, HSV, or wavelet.

Table 2. Results of Test 1 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	20.08	69.10	19.64	36.11	12.30	10.24	21.53	33.10	60.29	9.30
	Std	0.00	0.00	0.00	0.00	0.00	0.07	7.02	11.90	7.77	2.23
CC	*Mean*	92.85	61.45	94.07	82.23	97.54	98.37	94.61	87.65	89.76	98.62
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.06	0.02	0.01
RASE	*Mean*	35.61	122.55	34.82	64.03	21.82	18.17	38.18	58.70	106.92	16.49
	Std	0.00	0.00	0.00	0.00	0.00	0.12	12.46	21.10	13.78	3.95
QAVE	*Mean*	0.73	0.04	0.76	0.91	0.69	0.71	0.14	0.07	0.10	0.68
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.16	0.07	0.08	0.12
ERGAS	*Mean*	9.04	480.30	9.80	14.64	5.44	4.57	13.57	27.94	18.76	4.15
	Std	0.00	20.00	0.00	0.00	0.00	0.03	7.44	15.68	0.95	0.99
SSIM	*Mean*	61.76	6.89	66.26	57.85	72.49	80.46	49.16	32.40	26.91	82.54
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.15	0.04	0.06
PSNR	*Mean*	22.25	11.42	22.41	17.53	26.66	28.09	22.08	18.47	12.73	29.21
	Std	0.00	0.00	0.00	0.00	0.00	0.06	2.95	3.45	1.08	2.33
NIQE	*Mean*	3.55	11.38	3.59	4.40	3.32	3.92	5.08	5.74	5.79	4.54
	Std	0.00	0.00	0.00	0.00	0.00	0.03	1.60	2.27	0.60	1.28
BRISQUE	*Mean*	20.08	69.10	19.64	36.11	12.30	10.24	21.53	33.10	60.29	9.30
	Std	0.00	0.00	0.00	0.00	0.00	0.07	7.02	11.90	7.77	2.23
FD	*Mean*	446.34	8636.70	452.59	3261.40	387.35	197.56	943.10	2421.70	8034.40	224.38
	Std	0.00	0.00	0.00	4.00	0.00	0.02	635.48	1634.30	2112.60	48.82

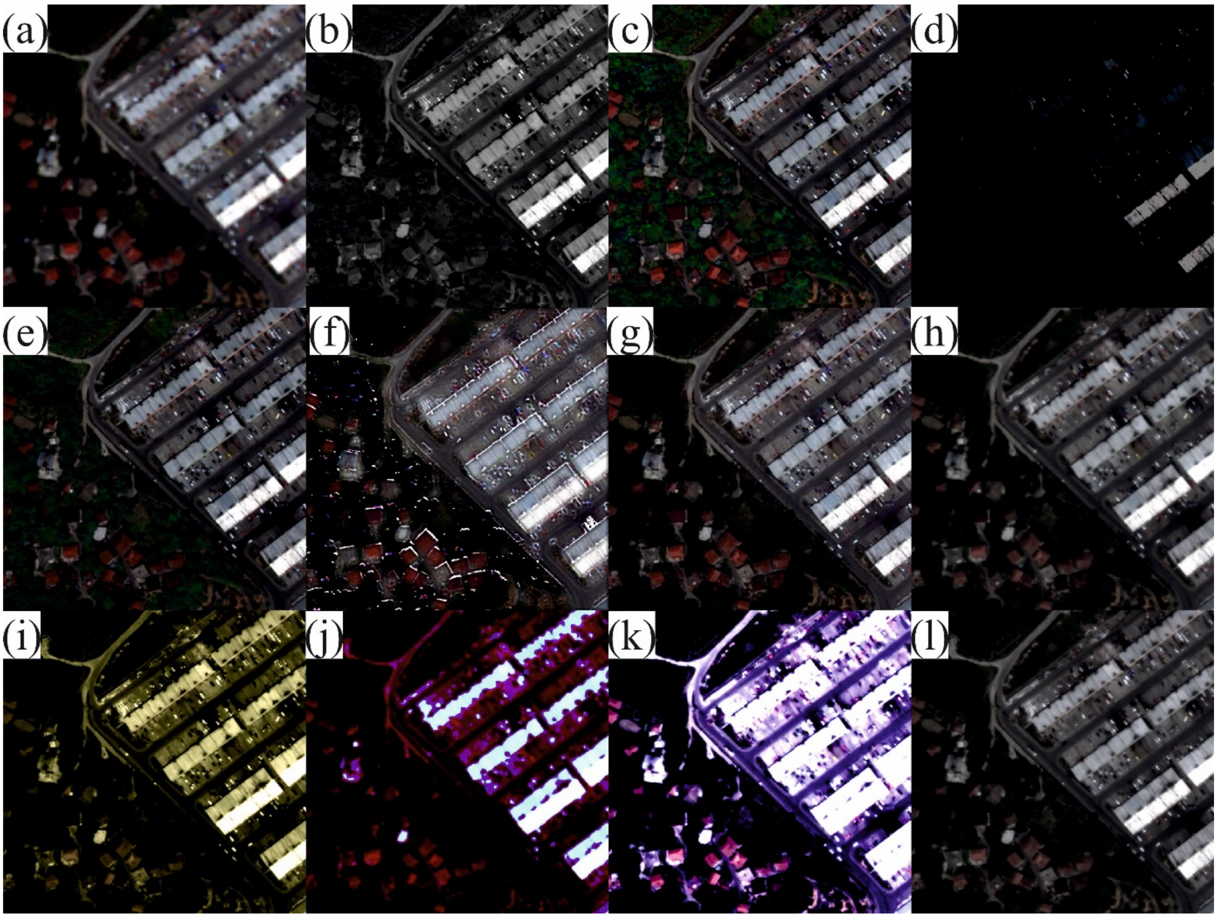

Figure 1. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 1.

Upon examining the values given in Table 3 for the second image of the dataset, it is evident that CSA-PS outperforms IHS, TLBO-PS, PSO-PS and FOA-PS for all the indices. It also outperforms Brovey and HSV for 9 indices, and SFIM for 8 indices. When all IQM values in Table 3 are considered together, CSA-PS achieved better results than the comparison methods at a level of 71.11%. According to the results presented in Table 3, the CSA-PS method provided better results for the RMSE, CC, RASE, ERGAS, SSIM, PSNR, BRISQUE, and FD metrics. Based on the experimental results obtained, CSA-PS generally provided higher chromatic and spatial accuracy than the compared methods, except for NIQE. Although CSA-PS and JADE-PS appeared to achieve quite close IQM values, CSA-PS obtained better average results for all the indices and more robust results due to relatively lower standard deviations. On the other hand, CSA-PS only improves the wavelet results for 2 indices (SSIM and FD).

Table 3. Results of Test 2 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	39.80	69.16	27.88	51.60	16.54	20.83	27.63	33.16	62.62	20.92
	Std	0.00	0.00	0.00	50.00	0.00	0.25	4.83	10.45	8.00	2.40
CC	*Mean*	84.54	72.54	88.32	69.84	96.09	95.23	89.87	87.64	88.89	94.93
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.05	0.02	0.02
RASE	*Mean*	66.96	116.36	46.90	86.90	27.83	35.04	46.48	55.78	105.36	35.19
	Std	0.00	0.00	0.00	0.00	0.00	0.43	8.13	17.59	13.46	4.04
QAVE	*Mean*	0.63	0.08	0.65	0.90	0.81	0.72	0.19	0.19	0.17	0.65
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.15	0.15	0.11	0.13
ERGAS	*Mean*	12.01	328.65	10.53	19.10	7.06	7.51	11.50	19.19	17.58	7.74
	Std	0.00	0.00	0.00	20.00	0.00	0.09	3.02	10.62	0.95	1.39
SSIM	*Mean*	53.83	7.34	56.99	65.07	74.95	75.84	42.31	37.26	31.50	73.41
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.15	0.06	0.08
PSNR	*Mean*	16.28	11.42	19.37	14.35	24.11	21.93	19.58	18.29	12.42	21.91
	Std	0.00	0.00	0.00	0.00	0.00	0.04	1.75	2.91	1.12	0.98
NIQE	*Mean*	3.01	8.73	3.36	2.91	2.67	3.46	3.48	4.02	4.05	3.54
	Std	0.00	0.00	0.00	0.00	0.00	0.02	0.40	1.57	0.32	0.35
BRISQUE	*Mean*	39.80	69.16	27.88	51.65	16.54	20.83	27.63	33.16	62.62	20.92
	Std	0.00	0.00	0.00	0.00	0.00	0.25	4.83	10.45	8.00	2.40
FD	*Mean*	2146.10	12317.00	1492.00	7003.00	1633.30	833.08	1565.90	3395.50	7562.00	919.09
	Std	0.00	0.00	0.00	0.00	0.00	0.04	717.02	2076.20	1599.20	340.28

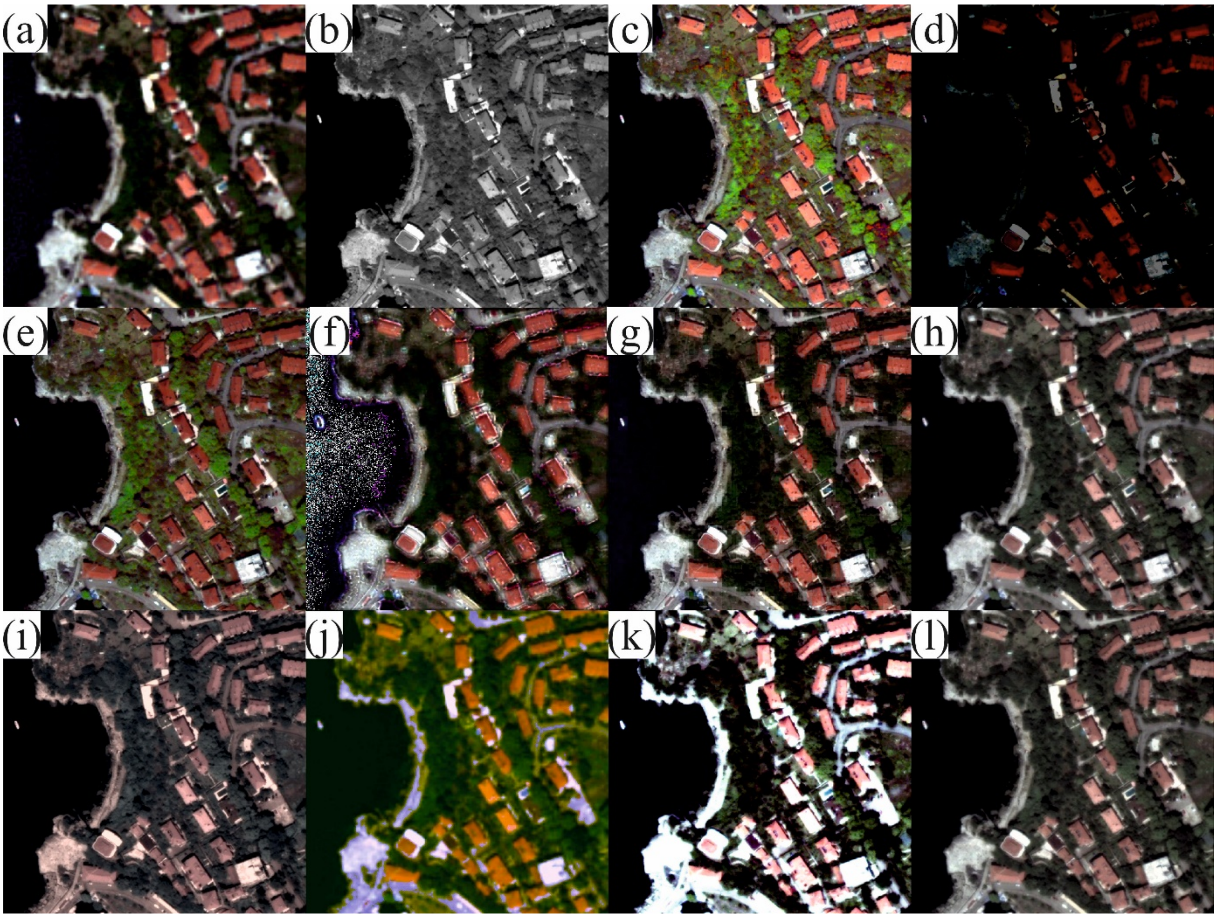

Figure 2. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 2.

The results included in Table 4 for image 3 indicate that CSA-PS outperforms IHS, PSO-PS, and FOA-PS for all the IQMs, Brovey, TBLO-PS and JADE-PS for 9 IQMs, and IHS and SFIM for 8 IQMs. It has been observed that CSA-PS generally provides better results than these methods in metrics such as RMSE, CC, RASE, ERGAS, SSIM, PSNR, BRISQUE, and FD. When considering all IQM values in Table 4 collectively, CSA-PS achieves superior results compared to the related comparison methods by a margin of 73.33%. Experimental results indicate that CSA-PS is more successful in delivering lower error rates and higher structural similarity overall. Specifically, CSA-PS exhibits better performance against IHS, TLBO-PS, PSO-PS, and FOA-PS in the QAVE metric, and achieves results comparable to other methods. The experiments summarized in Table 4 also show that the CSA-PS method delivers much more robust outcomes compared to JADE-PS. CSA-PS obtains the best results of the set of methods for the CC, ERGAS, SSIM, and FD indices. Although wavelet obtains the best results of the set of methods for all other IQMs except QAVE, differences with CSA-PS are small.

Table 4. Results of Test 3 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	35.47	76.89	24.96	43.67	17.10	18.22	27.65	35.18	58.68	18.95
	Std	0.00	0.00	0.00	0.00	0.00	0.07	5.54	11.54	7.05	2.06
CC	*Mean*	86.25	72.81	91.95	79.12	96.44	96.67	91.72	89.13	91.37	96.22
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.01	0.01
RASE	*Mean*	50.40	109.26	35.46	62.06	24.29	25.88	39.29	50.00	83.38	26.93
	Std	0.00	0.00	0.00	0.00	0.00	0.09	7.87	16.40	10.02	2.93
QAVE	*Mean*	0.69	0.12	0.75	0.88	0.68	0.72	0.20	0.16	0.16	0.64
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.11	0.18	0.07	0.15
ERGAS	*Mean*	10.23	211.55	8.71	14.13	6.09	5.87	10.79	16.76	16.01	6.11
	Std	0.00	0.00	0.00	0.00	0.00	0.02	3.66	7.54	0.92	0.65
SSIM	*Mean*	55.54	10.64	62.94	63.98	69.02	75.09	46.50	37.66	31.59	72.75
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.17	0.04	0.07
PSNR	*Mean*	17.34	10.59	20.37	15.71	23.77	23.07	19.70	17.96	13.03	22.78
	Std	0.00	0.00	0.00	0.00	0.00	0.03	1.88	3.39	1.04	0.84
NIQE	*Mean*	3.29	8.93	3.35	3.25	3.05	3.62	3.52	4.01	5.14	3.57
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.68	1.03	0.72	0.08
BRISQUE	*Mean*	35.47	76.89	24.96	43.67	17.10	18.22	27.65	35.18	58.68	18.95
	Std	0.00	0.00	0.00	0.00	0.00	0.07	5.54	11.54	7.05	2.06
FD	*Mean*	1691.00	12814.00	1209.60	5474.40	1228.20	637.34	1681.20	3611.20	7530.30	655.63
	Std	0.00	0.00	0.00	0.00	0.00	0.00	834.39	2127.40	1471.00	61.54

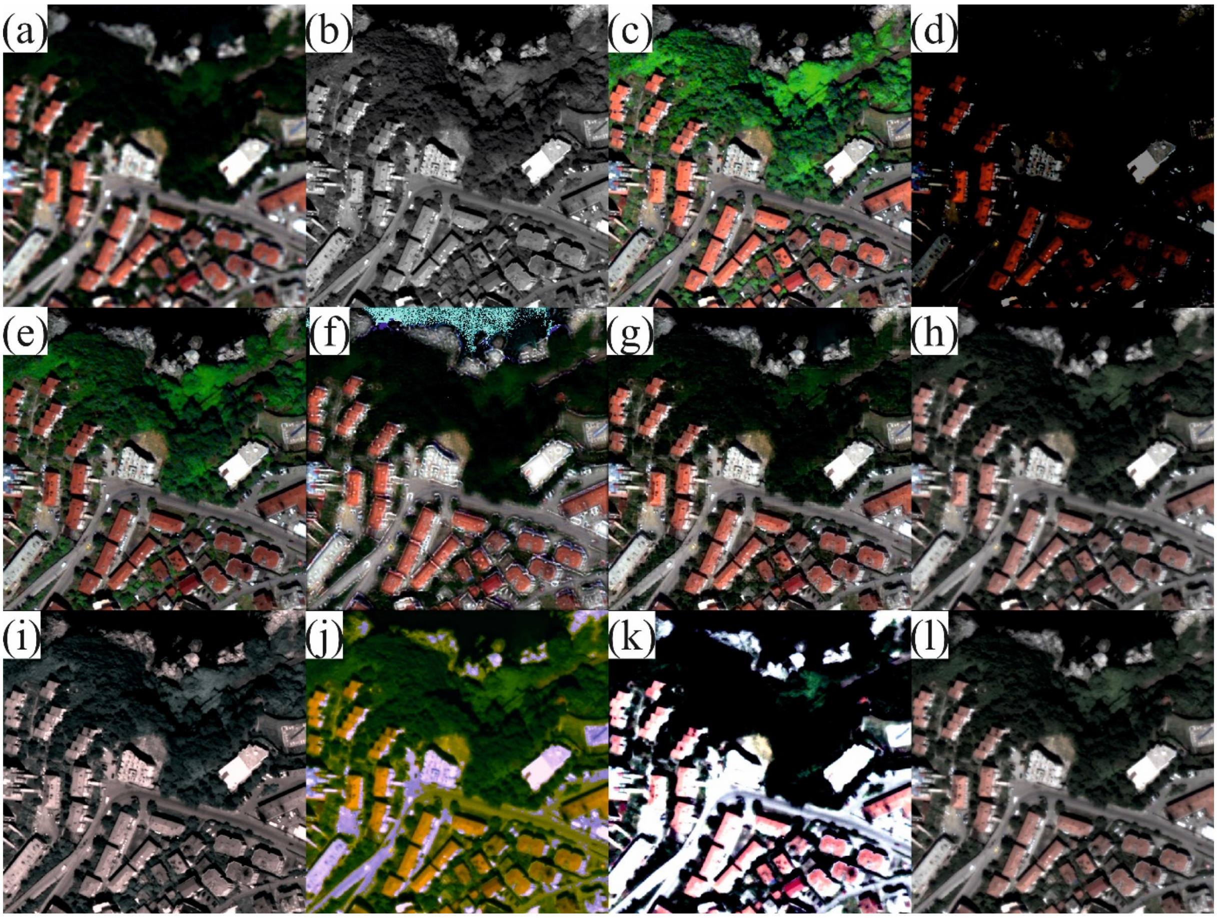

Figure 3. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 3.

Reviewing the values presented in Table 5, CSA-PS is shown to statistically outperform Brovey, HSV, PSO-PS, JADE-PS, TBLO-PS, FOA-PS and IHS. CSA-PS outperforms the initial four methods for all the indices, the next two methods for 9 indices, and the last method for 8 indices. The CSA-PS method demonstrates relatively better performance compared to other methods for the metrics CC, QAVE, ERGAS, SSIM, and FD. Considering all IQM values in Table 5 collectively, CSA-PS yields superior results compared to the related comparison methods by a margin of 62.22%. CSA-PS produces superior FD values compared to other methods, and while CSA-PS and JADE-PS yield similar results in terms of FD, the confidence interval for the results produced by CSA-PS is narrower, indicating more robust outcomes. This implies that users can place greater confidence in the results generated by CSA-PS in a random solution scenario. In these experiments, SFIM and wavelet achieved impressively successful results. Wavelet obtained better values than the rest of the methods for all indices except QAVE, SSIM, NIQE (in which cases SFIM obtained the best values), and FD. The best FD value was obtained by CSA-PS.

Table 5. Results of Test 4 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	84.59	54.27	64.65	35.81	26.96	41.95	52.87	46.15	64.05	42.93
	Std	0.00	0.00	0.00	0.00	0.00	0.10	7.21	13.19	9.74	3.46
CC	*Mean*	41.84	41.80	52.41	79.32	87.10	81.96	68.05	79.62	74.99	80.43
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.11	0.05	0.03
RASE	*Mean*	115.09	73.84	87.96	48.73	36.69	57.08	71.94	62.79	87.15	58.42
	Std	0.00	0.00	0.00	0.00	0.00	0.13	9.81	17.95	13.25	4.71
QAVE	*Mean*	0.59	0.67	0.65	0.92	0.60	0.67	0.25	0.19	0.23	0.58
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.17	0.21	0.13	0.15
ERGAS	*Mean*	15.73	25.19	13.95	11.41	9.08	10.12	14.83	14.46	17.19	10.55
	Std	0.00	0.00	0.00	0.00	0.00	0.02	3.97	6.20	1.19	1.11
SSIM	*Mean*	37.75	36.74	42.64	68.14	42.23	56.78	32.69	34.74	26.61	52.99
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.21	0.05	0.08
PSNR	*Mean*	9.71	13.54	12.00	17.35	19.67	15.73	13.87	15.30	12.25	15.56
	Std	0.00	0.00	0.00	0.00	0.00	0.02	1.20	2.43	1.34	0.60
NIQE	*Mean*	3.65	3.16	3.81	2.71	4.16	3.42	3.37	3.83	3.40	3.42
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.33	1.63	0.40	0.09
BRISQUE	*Mean*	84.59	54.27	64.65	35.81	26.96	41.95	52.87	46.15	64.05	42.93
	Std	0.00	0.00	0.00	0.00	0.00	0.10	7.21	13.19	9.74	3.46
FD	*Mean*	7684.50	9666.00	5160.00	10291.00	6669.00	3474.70	5796.50	7495.00	9729.70	3817.50
	Std	0.00	0.00	0.00	0.00	0.00	0.05	1954.90	3320.30	1173.80	1218.00

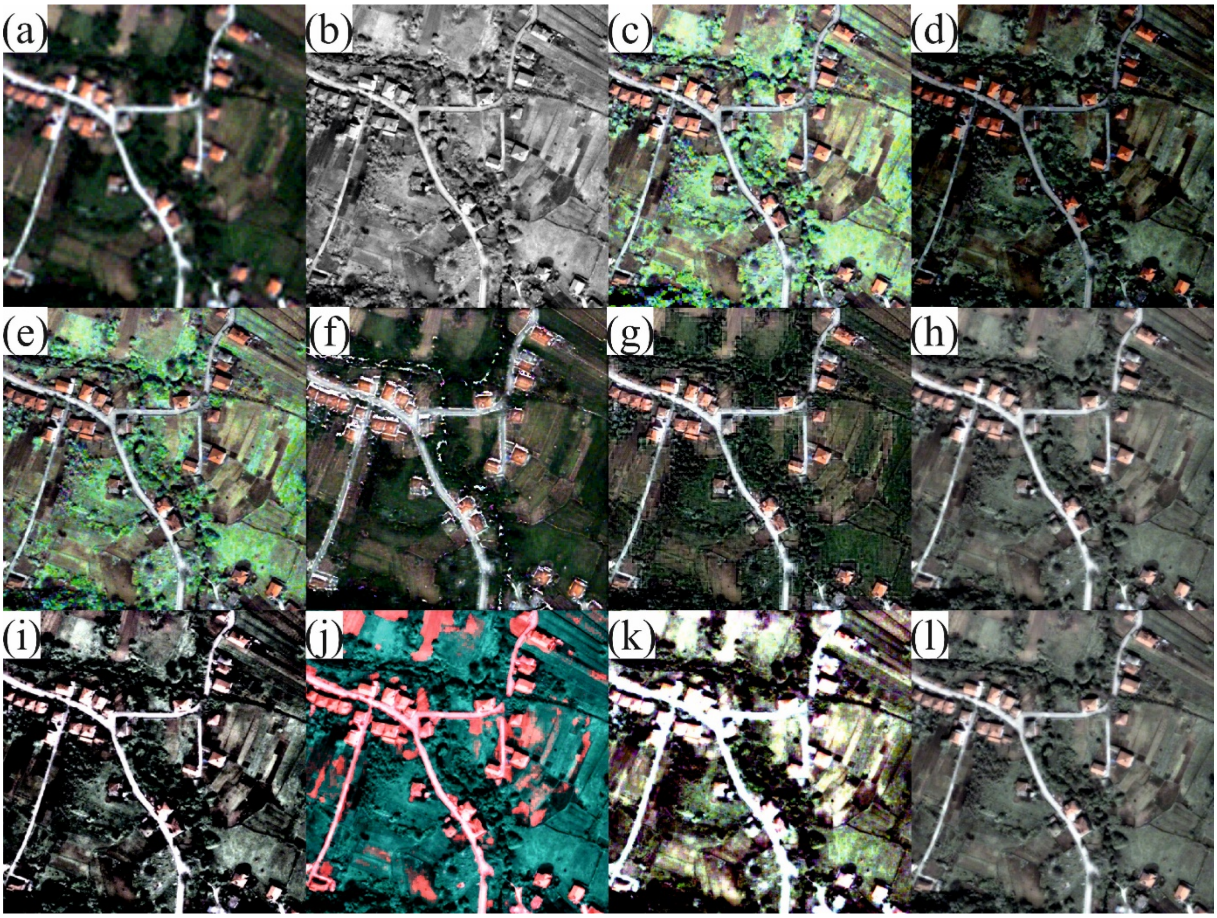

Figure 4. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 4.

Examining the values presented in Table 6, CSA-PS outperforms Browey and the four population-based methods for all the IQMs, HSV for 9 IQMs, and IHS and SFIM for 8 IQMs. Considering all IQM values in Table 6 collectively, CSA-PS achieves superior results compared to the related comparison methods by a margin of 68.89%. According to Table 6, the CSA-PS method demonstrates superior performance in numerous metrics compared to other methods. It achieves better results in metrics such as RMSE, CC, RASE, QAVE, ERGAS, SSIM, PSNR, BRISQUE and FD compared to the methods indicated above. On the other hand, the results of the experiments showed that wavelet provided more successful results than CSA-PS, since it obtained better results than CSA-PS for all indices except QAVE, SSIM, and FD.

Table 6. Results of Test 5 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	60.26	56.29	51.27	50.86	20.39	29.62	43.78	45.07	56.97	34.08
	Std	0.00	0.00	0.00	0.00	0.00	0.06	8.09	12.67	6.41	13.97
CC	*Mean*	56.21	57.30	64.59	74.43	94.87	88.28	81.76	82.98	86.00	85.23
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.07	0.03	0.10
RASE	*Mean*	63.93	59.72	54.40	53.96	21.64	31.42	46.45	47.82	60.44	36.16
	Std	0.00	0.00	0.00	0.00	0.00	0.06	8.58	13.45	6.80	14.82
QAVE	*Mean*	0.65	0.70	0.65	0.89	0.65	0.70	0.18	0.13	0.13	0.55
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.11	0.14	0.09	0.22
ERGAS	*Mean*	14.25	16.30	13.47	12.09	5.34	7.40	13.41	15.00	14.99	8.22
	Std	0.00	0.00	0.00	0.00	0.00	0.01	2.90	5.21	0.79	1.74
SSIM	*Mean*	43.35	40.78	43.21	57.74	55.02	64.70	34.08	27.47	23.46	56.83
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.16	0.05	0.15
PSNR	*Mean*	12.67	13.16	14.02	14.26	22.16	18.76	15.59	15.54	13.22	17.92
	Std	0.00	0.00	0.00	0.00	0.00	0.02	1.72	2.64	0.96	2.25
NIQE	*Mean*	2.94	2.75	2.70	2.65	2.37	2.76	2.88	3.30	3.45	2.77
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.47	1.09	0.35	0.10
BRISQUE	*Mean*	60.26	56.29	51.27	50.86	20.39	29.62	43.78	45.07	56.97	34.08
	Std	0.00	0.00	0.00	0.00	0.00	0.06	8.09	12.67	6.41	13.97
FD	*Mean*	3879.00	3604.50	2995.20	9958.50	3174.30	1728.16	4065.70	5639.60	7516.70	2712.60
	Std	0.00	0.00	0.00	0.00	0.00	0.02	1424.00	2748.70	1147.90	3369.00

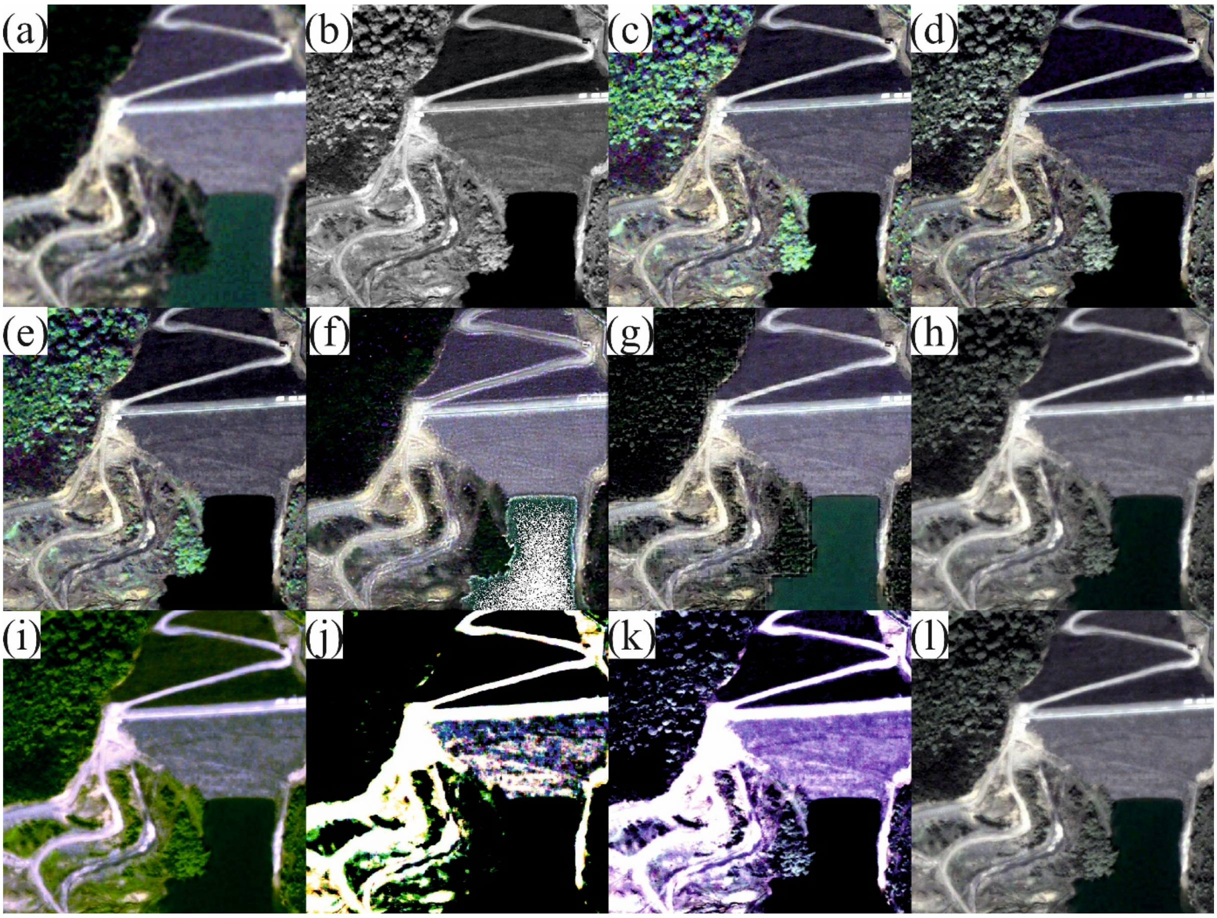

Figure 5. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 5.

After analyzing the results presented in Table 7, it is evident that CSA-PS outperforms IHS, TBLO-PS, PSO-PS, and FOA-PS for all indices. CSA-PS also outperforms Brovey and JADE-PS for 9 indices, and HSV and SFIM for 8 indices. When considering all IQM values in Table 7 collectively, CSA-PS achieved superior results by 74.44% compared to the related comparison methods. CSA-PS provides better results in terms of RMSE, CC, RASE, ERGAS, SSIM, PSNR, BRISQUE, and FD metrics. In particular, CSA-PS demonstrated significantly superior outcomes concerning SSIM and FD. In addition, wavelet outperforms CSA-PS for all the indices except QAVE, SSIM, and FD. JADE-PS also produced commendable results. However, the respective standard deviation values suggest that CSA-PS remained relatively more robust.

Table 7. Results of Test 6 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	49.89	62.71	38.28	39.19	20.42	24.34	34.76	37.46	55.65	24.85
	Std	0.00	0.00	0.00	0.00	0.00	0.10	7.19	12.38	6.12	2.33
CC	*Mean*	71.73	76.45	81.17	83.79	95.02	94.49	87.11	87.86	91.15	93.91
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.06	0.01	0.02
RASE	*Mean*	63.74	80.13	48.91	50.07	26.09	31.10	44.42	47.86	71.10	31.75
	Std	0.00	0.00	0.00	0.00	0.00	0.13	9.18	15.82	7.82	2.98
QAVE	*Mean*	0.67	0.40	0.70	0.91	0.56	0.69	0.15	0.11	0.13	0.62
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.12	0.11	0.08	0.12
ERGAS	*Mean*	12.40	50.71	10.82	11.56	6.45	6.84	12.24	14.43	15.14	6.99
	Std	0.00	0.00	0.00	0.00	0.00	0.03	3.25	5.40	0.73	0.64
SSIM	*Mean*	44.24	26.04	47.98	58.41	48.69	63.71	35.36	30.14	25.44	62.04
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.16	0.04	0.06
PSNR	*Mean*	14.33	12.29	16.64	16.60	22.13	20.59	17.64	17.30	13.43	20.41
	Std	0.00	0.00	0.00	0.00	0.00	0.04	1.92	3.12	0.95	0.80
NIQE	*Mean*	3.05	4.06	3.15	2.82	2.68	3.17	3.31	3.95	4.92	3.16
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.56	1.36	0.99	0.06
BRISQUE	*Mean*	49.89	62.71	38.28	39.19	20.42	24.34	34.76	37.46	55.65	24.85
	Std	0.00	0.00	0.00	0.00	0.00	0.10	7.19	12.38	6.12	2.33
FD	*Mean*	2773.90	9034.80	1850.90	6036.00	2107.80	1133.20	2981.10	4671.20	8034.30	1159.60
	Std	0.00	0.00	0.00	0.00	0.00	0.03	1280.70	2404.70	1258.00	64.80

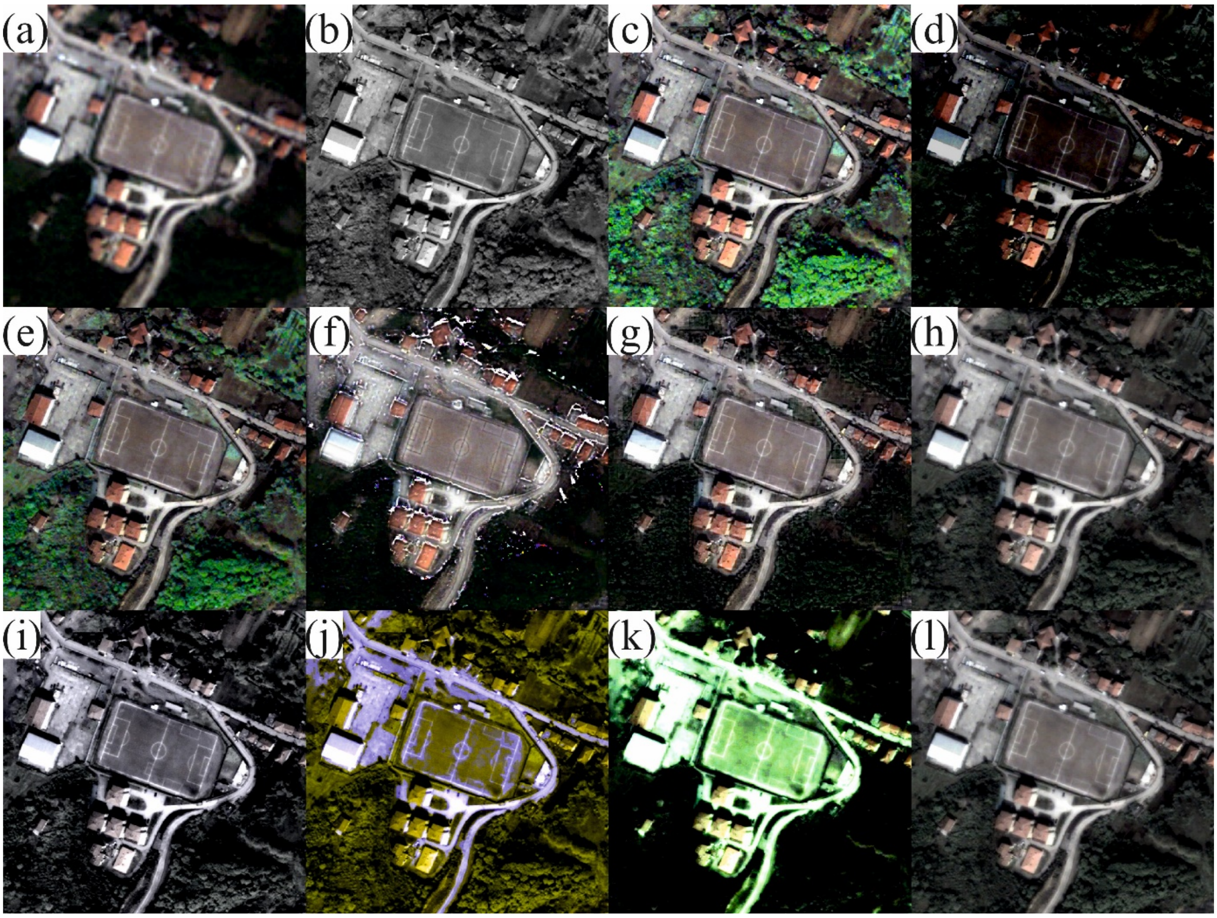

Figure 6. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 6.

The results reported in Table 8 indicate that CSA-PS outperforms IHS and the other population-based methods for all the indices. It also outperforms HSV, SFIM, and wavelet for 9 IQM values, respectively. Compared to Brovey, CSA-PS obtains the same result for the QAVE index but better results for the rest of the indexes. The CSA-PS method produced relatively better responses for RMSE, CC, RASE, ERGAS, SSIM, PSNR, BRISQUE, and FD. Considering all IQM values in Table 8 collectively, CSA-PS achieved superior results by 70% compared to the related compared methods. Notably, CSA-PS demonstrated a significant advantage over the IHS, PSO-PS, and FOA-PS methods in terms of CC, ERGAS, SSIM, PSNR, NIQE, and BRISQUE.

Table 8. Results of Test 7 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	29.87	67.34	27.78	46.19	21.53	16.66	27.77	40.48	66.78	19.92
	Std	0.00	0.00	0.00	0.00	0.00	0.09	6.11	18.09	8.14	18.54
CC	*Mean*	84.52	71.03	86.75	69.63	92.09	94.70	89.22	82.28	86.12	92.51
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.11	0.02	0.12
RASE	*Mean*	46.80	105.52	43.52	72.38	33.74	26.10	43.52	63.43	104.64	31.21
	Std	0.00	0.00	0.00	0.00	0.00	0.15	9.57	28.34	12.76	29.05
QAVE	*Mean*	0.76	0.11	0.73	0.86	0.84	0.76	0.23	0.18	0.19	0.71
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.19	0.15	0.11	0.17
ERGAS	*Mean*	11.46	241.23	12.99	16.01	8.50	6.51	13.59	28.83	18.40	6.68
	Std	0.00	0.00	0.00	0.00	0.00	0.04	5.23	16.19	0.90	1.24
SSIM	*Mean*	60.24	9.55	60.04	52.30	68.46	75.96	44.16	32.84	28.00	73.16
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.16	0.05	0.15
PSNR	*Mean*	18.82	11.66	19.43	15.15	21.80	23.94	19.64	16.88	11.85	23.43
	Std	0.00	0.00	0.00	0.00	0.00	0.05	2.11	3.76	1.06	3.23
NIQE	*Mean*	2.41	10.48	2.17	3.84	2.61	2.35	2.79	3.92	3.72	2.42
	Std	0.00	0.00	0.00	0.00	0.00	0.02	0.78	2.11	0.44	0.21
BRISQUE	*Mean*	29.87	67.34	27.78	46.19	21.53	16.66	27.77	40.48	66.78	19.92
	Std	0.00	0.00	0.00	0.00	0.00	0.09	6.11	18.09	8.14	18.54
FD	*Mean*	1193.00	9003.00	1165.00	6143.00	985.00	522.28	1396.20	4259.00	8770.40	1424.50
	Std	0.00	0.00	0.00	0.00	0.00	0.07	668.53	4603.00	2157.20	4787.00

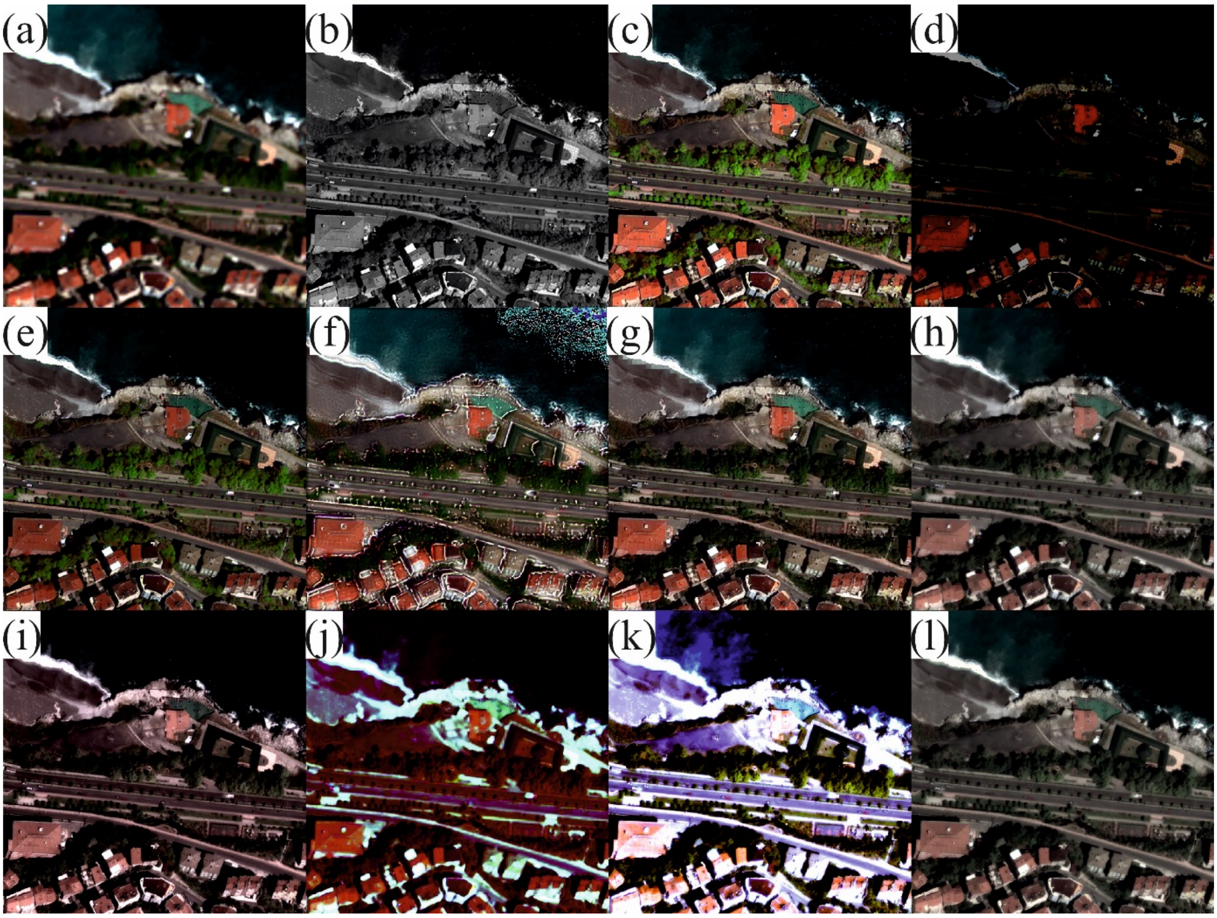

Figure 7. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 7.

The results included in Table 9 show that CSA-PS outperforms IHS, wavelet, TLBO-PS, PSO-PS, and FOA-PS for all the indices, SFIM for 9 indices, and Brovey and HSV for 8 indices. Considering all IQM values in Table 9 collectively, CSA-PS achieved superior results by 81.11% compared to the related comparison methods. CSA-PS provided relatively superior values for RMSE, CC, RASE, ERGAS, SSIM, PSNR, BRISQUE, and FD metrics. In this experiment, CSA-PS notably outperformed wavelets, and it also achieved more robust values compared to JADE-PS, although the average values obtained by JADE-PS are slightly better.

Table 9. Results of Test 8 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	31.96	80.02	33.76	62.43	27.08	18.79	34.36	38.53	60.84	18.25
	Std	0.00	0.00	0.00	0.00	0.00	0.13	9.24	14.87	7.89	1.91
CC	*Mean*	84.74	68.93	86.28	53.84	89.16	94.04	85.29	85.89	85.62	94.35
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.08	0.02	0.01
RASE	*Mean*	40.49	101.39	42.77	79.10	34.31	23.81	43.53	48.82	77.09	23.13
	Std	0.00	0.00	0.00	0.00	0.00	0.17	11.71	18.84	10.00	2.42
QAVE	*Mean*	0.80	0.14	0.78	0.78	0.63	0.76	0.25	0.26	0.24	0.74
	Std	0.00	0.00	0.00	0.00	0.00	0.01	0.13	0.13	0.08	0.04
ERGAS	*Mean*	10.96	243.26	14.03	16.75	8.87	6.17	13.92	25.82	15.73	5.99
	Std	0.00	0.00	0.00	0.00	0.00	0.05	6.75	19.70	0.81	0.65
SSIM	*Mean*	63.14	12.40	63.28	35.89	63.69	77.75	47.49	49.19	38.46	78.55
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.17	0.05	0.03
PSNR	*Mean*	18.23	10.14	17.70	12.42	19.65	22.80	17.86	17.28	12.66	23.13
	Std	0.00	0.00	0.00	0.00	0.00	0.06	2.39	3.85	1.11	1.25
NIQE	*Mean*	3.23	8.30	3.16	6.46	3.29	3.73	4.13	5.34	5.95	3.86
	Std	0.00	0.00	0.00	0.00	0.00	0.02	1.12	2.35	0.68	0.58
BRISQUE	*Mean*	31.96	80.02	33.76	62.43	27.08	18.79	34.36	38.53	60.84	18.25
	Std	0.00	0.00	0.00	0.00	0.00	0.13	9.24	14.87	7.89	1.91
FD	*Mean*	1502.40	11835.00	1750.90	11127.00	1287.40	678.60	1813.80	3908.60	7663.30	700.13
	Std	0.00	0.00	0.00	0.00	0.00	0.05	1047.70	2296.70	2013.20	98.51

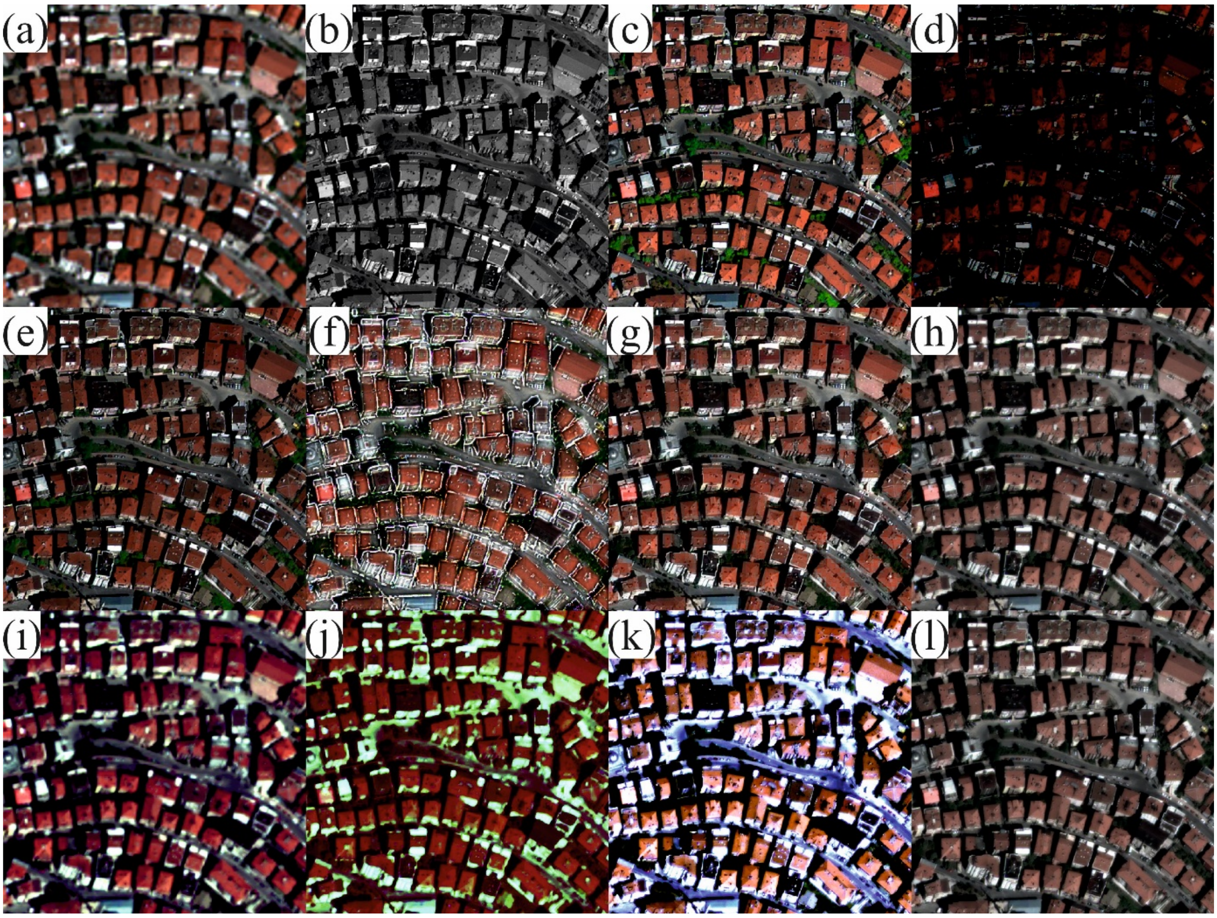

Figure 8. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 8.

The results of the last test, applied to image 9, are included in Table 10. This table clearly shows that CSA-PS outperformed all the methods for all the indices. Furthermore, the second-best method is TLBO-PS, which outperforms the other 8 methods for 6 of the indices. On the contrary, the worst results are obtained by the SFIM method, followed by the FOA-PS method.

Table 10. Results of Test 9 image.

IQM	Statistics	Method									
		Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*Mean*	27.65	43.44	24.62	100.71	19.22	15.68	18.61	35.31	72.24	21.47
	Std	0.00	0.00	0.00	0.00	0.00	0.18	5.47	20.60	7.23	21.80
CC	*Mean*	89.37	82.95	90.50	37.44	94.22	96.20	94.37	85.44	84.95	92.88
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.11	0.02	0.13
RASE	*Mean*	75.65	118.83	67.36	275.50	52.57	42.90	50.92	96.58	197.60	58.72
	Std	0.00	0.00	0.00	0.00	0.00	0.50	14.95	56.35	19.78	59.64
QAVE	*Mean*	0.63	0.13	0.58	0.70	0.58	0.72	0.26	0.08	0.11	0.58
	Std	0.00	0.00	0.00	0.00	0.00	0.02	0.14	0.08	0.04	0.24
ERGAS	*Mean*	17.62	121.83	19.95	35.34	13.40	10.45	13.92	37.12	25.06	11.61
	Std	0.00	0.00	0.00	0.00	0.00	0.10	4.29	21.19	1.12	2.23
SSIM	*Mean*	63.42	19.84	61.52	21.37	70.32	81.58	49.01	33.09	24.93	73.64
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.16	0.05	0.17
PSNR	*Mean*	19.85	15.66	20.75	8.78	22.82	24.80	23.54	18.31	11.32	23.35
	Std	0.00	0.00	0.00	0.00	0.00	0.04	3.13	3.53	0.88	3.83
NIQE	*Mean*	3.80	5.57	3.57	5.26	4.08	3.18	3.66	5.76	4.80	3.19
	Std	0.00	0.00	0.00	0.00	0.00	0.03	0.77	3.94	0.86	0.09
BRISQUE	*Mean*	27.65	43.44	24.62	100.71	19.22	15.68	18.61	35.31	72.24	21.47
	Std	0.00	0.00	0.00	0.00	0.00	0.18	5.47	20.60	7.23	21.80
FD	*Mean*	1007.00	3879.90	947.70	19218.00	795.30	426.14	699.86	3670.70	8997.30	1689.00
	Std	0.00	0.00	0.00	0.00	0.00	0.02	201.60	6138.00	1820.80	6152.80

When considering the results of all experiments together for the 95% normal distribution confidence limits, CSA-PS achieved better results by 65.12% (73.38% excluding JADE-PS) compared to the other methods. CSA-PS consistently produced more robust results than JADE-PS across all experiments. Experimental results related to the FD value indicate that CSA-PS has generally lower chromatic and spatial information distortion compared to the comparison methods. The multi-swarm-based architecture of CSA-PS limits the loss of numerical diversity during the optimization process. Hence, CSA-PS can perform more efficient and robust searches compared to other evolutionary calculators. CSA-PS employs an objective function based on L0-Norm, allowing it to maximize the number of pixels that better preserve chromatic and spatial values.

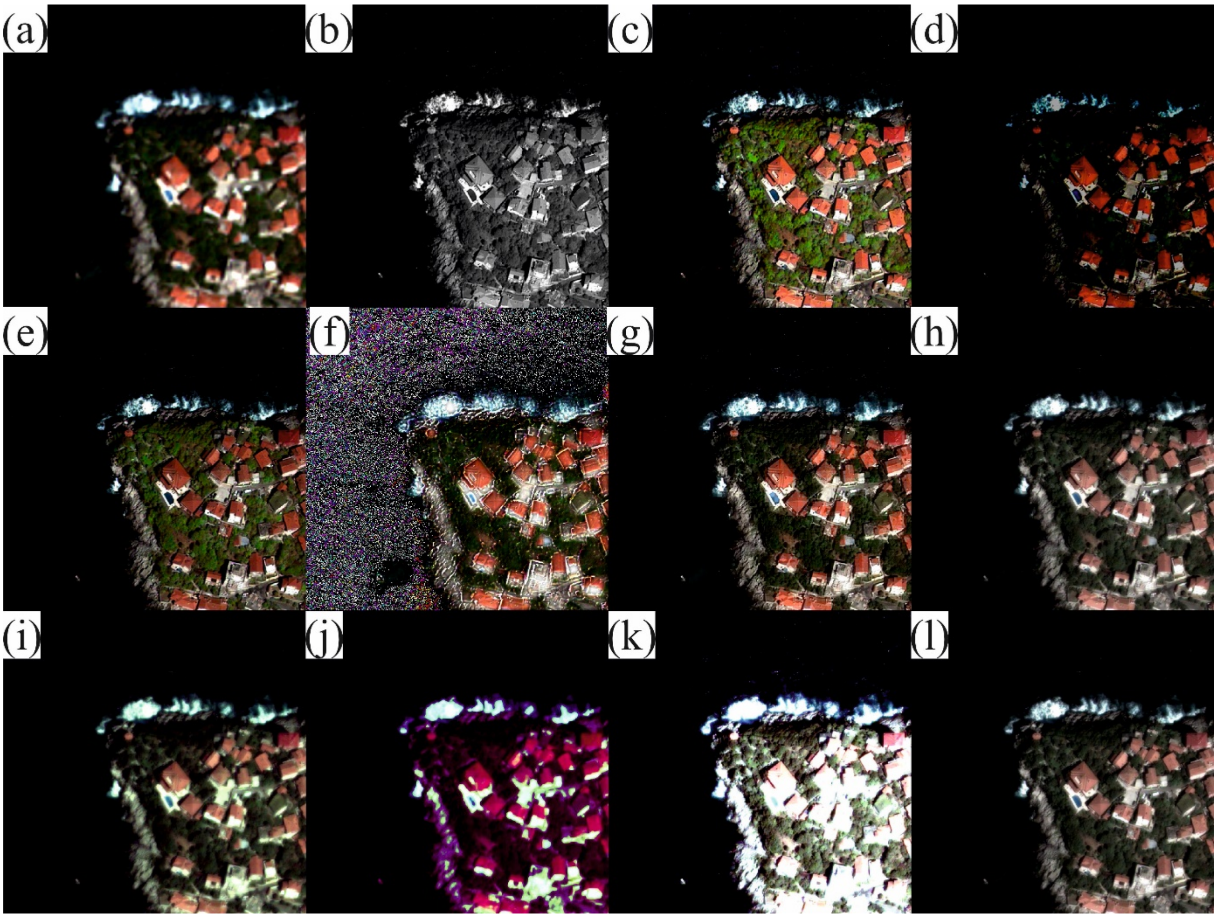

Figure 9. a) MSI, b) PAN, c) Brovey, d) IHS, e) HSV, f) SFIM, g) Wavelet, h) CSA-PS, i) TLBO-PS, j) PSO-PS, k) FOA-PS, and l) JADE-PS image of Test 9.

4.3. Analysis of the results

The discrete nature of the images necessitates not only numerical methods but also visual interpretation to assess the quality of the obtained results. This analysis can be performed by observing the images included in Figures 1 to 9. Visual evaluation results indicate that the pansharpened images obtained by CSA-PS are significantly less affected by aliasing, image halo, and image blur effects compared to the pansharpened images obtained by the comparison methods. The detailed analyses conducted indicate that CSA-PS has considerable potential for pansharpened image synthesis. According to the results of the experiments conducted in this study, CSA-PS has demonstrated success in synthesizing super resolution new images from remote sensing images.

Once the image results have been analyzed independently, a global analysis of the results obtained by each method can be made. Figures 10 and 11 compare the average results of the 5 classical methods with the results of the L0pan method that uses each of the 5 population-based methods. The information of these figures shows that wavelet is the best of the classical methods, while IHS is the worst. Wavelet is the best method according to all the indices, except for QAVE (for this index it is the fourth of the 5 classical methods) and for FD (in this case it is the second method, after HSV). In general, HSV is the second best among the classical methods, as it is the second best for all indices except for ERGAS (in which case Brovey obtains a slightly better result) and for FD (in which case it is the best method). IHS is the worst method according to all indices. In the case of the ERGAS index, a notable difference is observed among the result obtained by IHS and that obtained by the rest of the methods that are compared. SFIM is the second-worst classical method according to all indices except QAVE (in which case it is the best method).

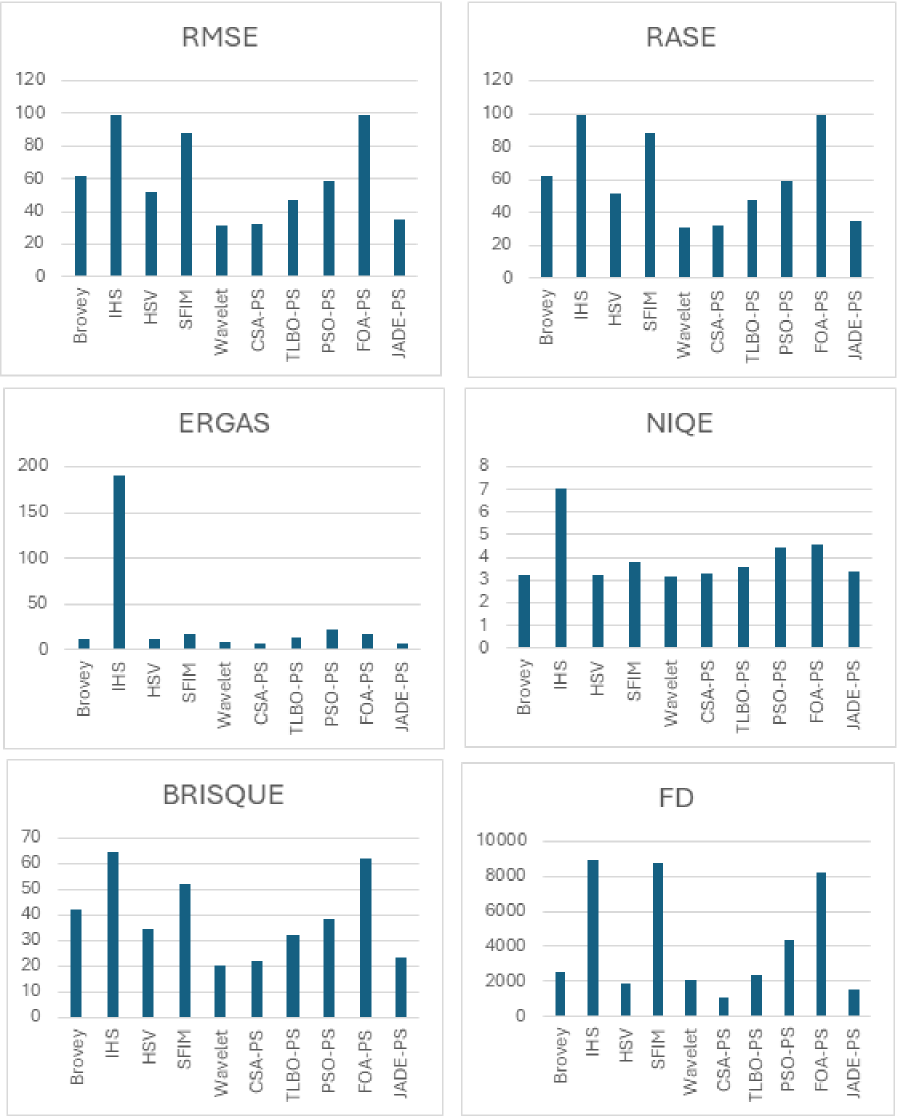

Figure 10. Average value of RMSE, RASE, ERGAS, NIQE, BRISQUE, and FD for all the methods compared.

Figures 10 and 11 also allow comparisons of the five population-based methods. It can be clearly observed that CSA-PS obtains the best results for all the indices represented. In addition, JADE-PS is the second best, with values very close to those obtained by CSA-PS for some indices. On the contrary, PSO-PS and FOA-PS are the worst methods, generating the worst or second-worst result for all the analyzed indices. FOA-PS is the worst method according to 7 indices (RMSE, RASE, NIQE, BRISQUE, FD, PSNR, SSIM), while PSO-PS is the worst method according to the other 3 indices (ERGAS, QAVE, CC). Regarding the FD value, it is observed that the result for image 4 is clearly worse than for the rest of the images. Furthermore, the images generated by the FOA-PS method have a smaller range of variation for this value than the images generated by the other methods.

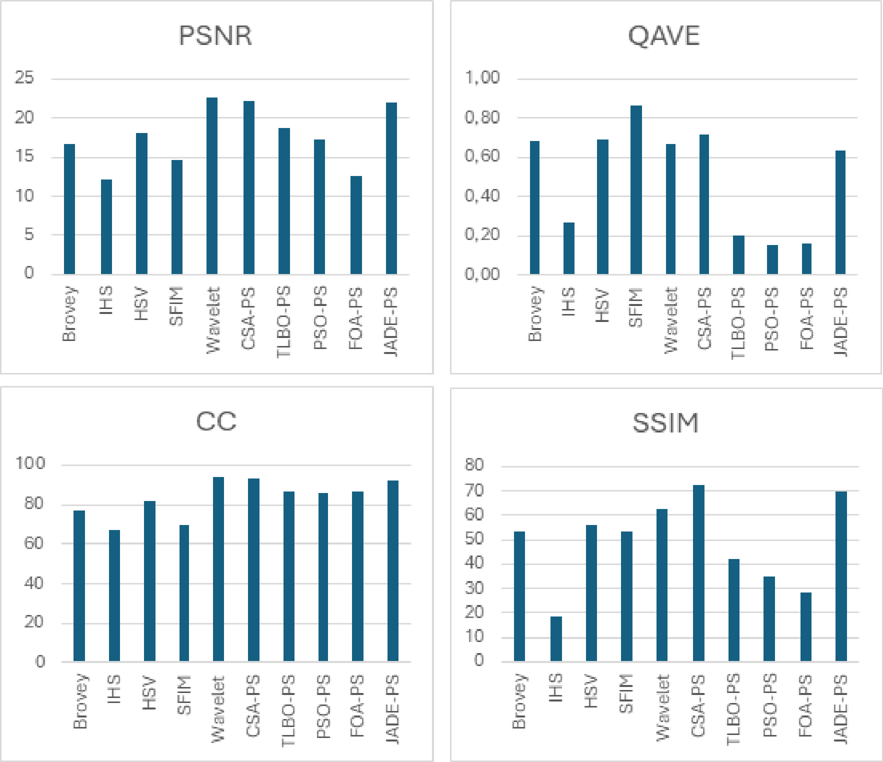

Figure 11. Average value of PSNR, QAVE, CC and SSIM for all the methods compared.

The box plots included in Figure 12 complete the comparison of the population-based methods. The boxes show the FD results obtained for each image in the 30 independent tests carried out. The figure clearly shows that CSA-PS is the method that obtains the best results. The boxes corresponding to CSA-PS are those with the lowest mean and lowest amplitude. The means obtained for JADE-PS are similar, but in some cases there is more dispersion in the results. TLBO-PS and PSO-PS are the methods that present more variability in the results, obtaining worse values in the second case. Finally, the FOA-PS results present little variability, but the average values obtained only improve those obtained by PSO-PS in some cases.

If the information in Figures 10 and 11 is used to compare the two subsets of methods, it is observed that IHS is the method that obtains the worst overall results. SFIM and FOA-PS also perform poorly for many of the indices. However, in the case of the QAVE index, SFIM is the method that obtains the best result. On the other hand, CSA-PS, JADE-PS, and wavelet obtain the best results for many of the indices, with small differences in most cases. The QAVE index seems to show a different trend from the rest of the indices, since for this index the Brovey and HSV methods obtain results quite similar to those obtained by CSA-PS, JADE-PS, and wavelet.

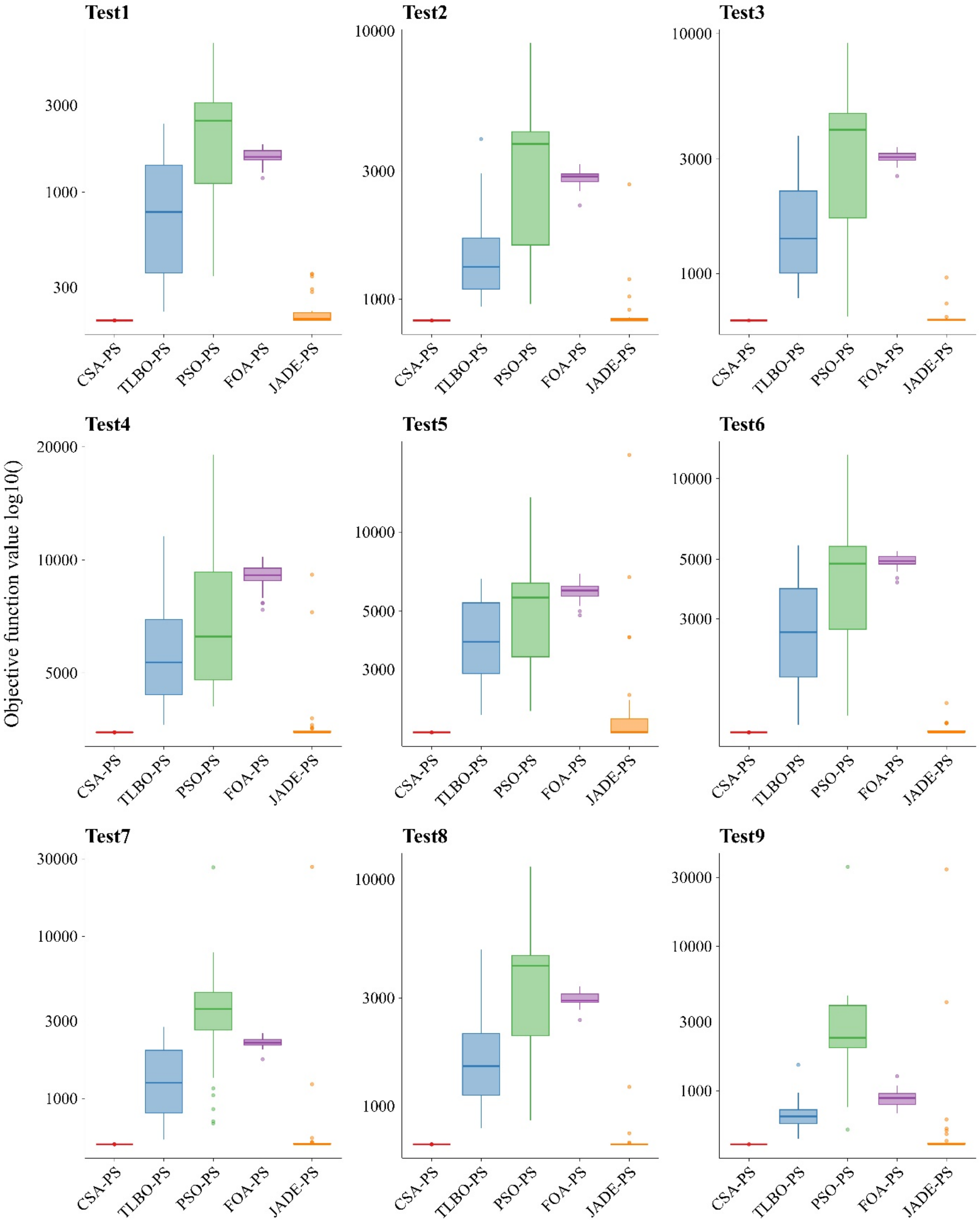

Figure 12. Boxplot of the objective function value of the population-based methods.

With the aim of making a global assessment of the methods, a ranking was created based on the quality of the results obtained by each method for the set of indices. To rank the methods, the index obtained for each image by the 10 methods was considered, and a score between 1 and 10 was assigned to each method (where 1 represents the best method and 10 the worst). Next, the average score for the set of 9 images was calculated. The results obtained are shown in Table 11. It is observed that the CSA-PS method obtains the best results for most of the indices, followed by wavelet and JADE-PS. It is also observed that there are 5 ties in the results obtained for CSA-PS and wavelet, which indicates that the overall results of both methods are quite similar. In contrast, IHS obtains the worst results for most of the indices. FOA-PS is the second-worst method according to almost all indices, and PSO-PS is the third worst.

Table 11. Ranking of the methods for each index.

	Brovey	IHS	HSV	SFIM	Wavelet	CSA-PS	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
FD	5.22	9.00	3.56	8.78	3.67	1.00	5.00	7.00	8.89	2.89
RMSE	7.00	9.22	5.56	7.33	1.89	1.89	4.56	6.11	8.78	2.67
PSNR	7.00	9.33	5.67	7.33	2.00	1.89	4.44	6.11	8.67	2.56
RASE	7.00	9.22	5.44	7.33	1.89	1.89	4.56	6.11	8.78	2.67
QAVE	3.78	7.67	3.22	1.22	4.33	2.78	7.44	8.78	8.67	5.56
CC	8.00	9.33	5.89	8.56	2.00	1.67	5.00	6.22	5.67	2.67
ERGAS	5.56	10.00	5.00	6.44	2.00	1.67	5.56	8.56	7.89	2.33
SSIM	5.22	9.22	4.78	4.89	3.33	1.33	7.22	7.78	9.11	2.11
NIQE	4.00	8.22	3.44	4.33	3.22	4.33	5.67	8.44	8.11	5.11
BRISQUE	7.00	9.22	5.56	7.33	1.89	1.89	4.56	6.11	8.78	2.67

Figure 13 shows the global ranking of the methods based on the average value of the rankings obtained for the 10 indices. This figure shows that, when the set of indices is considered globally, the best method is CSA-PS and the worst is IHS. As far as population-based methods are concerned, CSA-PS and JADE-PS are among the top 3 methods; TLBO-PS is in the middle position in the ranking, and the other two population-based methods only outperform IHS. The results of the wavelet method are between those obtained by CSA-PS and JADE-PS, with small differences between these methods.

To complete the analysis of the results, a statistical test was performed to determine if the differences observed between the CSA-PS method and the remaining methods were significant. The Wilcoxon test was applied to analyze the statistical significance of the improvement obtained by CSA-PS compared to the rest of the methods. This test is applied to the results of two methods in order to determine that there are no significant differences between both methods. The results obtained when the test was applied to each pair of methods with a significance level equal to 0.05 appear in Table 12. For each test, the table shows the value of the test statistic (*stat*), the probability value corresponding to *stat* (*sig*), the sum of positive ranks ($r+$), and the sum of negative ranks ($r-$). The *sig* values indicate that there are no significant differences with wavelet or JADE-PS for the RMSE, RASE, ERGAS, PSNR, and BRISQUE indices. For the CC index, there are no significant differences between CSA-PS and wavelet. There is no significant difference between CSA-PS and wavelet, Brovey, or HVS for the results of the QAVE index. Furthermore, there is no significant difference between CSA-PS and wavelet, Brovey, HVS, SFIM or JADE-PS for the NIQE index. This is the index for which the statistical test indicates more pairs of cases with nonsignificant differences. In the rest of the cases not listed, the differences between

CSA-PS and the other methods are significant.

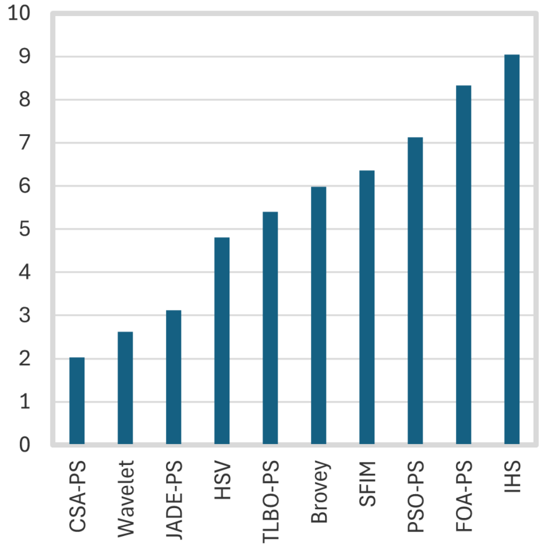

Figure 13. Ranking of the 10 methods.

In summary, the differences between CSA-PS and wavelet are not significant for 8 of the 10 indices (all except FD and SSIM), while in the case of JADE-PS they are not significant for 6 of the indices (RMSE, RASE, ERGAS, PSNR, NIQE, BRISQUE). Furthermore, for the QAVE and NIQE indices, it is observed that there are no significant differences between CSA-PS and other 3 and 5 methods, respectively. The sums of the ranks obtained in the test allow us to determine which of the two methods that present significant differences is better. In all cases except one, this information indicates that CSA-PS obtains a significantly better error than the other method compared. The only case in which CSA-PS obtains a significantly worse value is when compared to the SFIM method taking into account the QAVE index. The difference between the results of SFIM and CSA-PS with respect to the QAVE index was already pointed out in the previous discussion.

Therefore, the statistical tests support that CSA-PS obtains better quality results than many of the compared methods, although they are of the same quality as those generated by JADE-PS and wavelet for several of the analyzed indices.

Figure 14 shows the runtime values of all the methods. Upon examining this figure, it is evident that CSA-PS is faster than other population-based methods. Although the results obtained for some IQMs by JADE-PS are similar or even better than those obtained by CSA-PS, it can be observed that JADE-PS consumes much more time than CSA-PS. Therefore, CSA-PS would be the best option to use in the L0pan method defined in this article, since it can obtain higher quality results than the other population-based methods and requires less time. The figure also shows that wavelet is faster than CSA-PS. Certainly, all methods not based on populations are faster.

Table 12. Results of the Wilcoxon test that compares CSA-PS with the other methods.

		Brovey	IHS	HVS	SFIM	Wavelet	TLBO-PS	PSO-PS	FOA-PS	JADE-PS
RMSE	*stat*	-2666	-2666	-2666	-2547	-533	-2666	-2666	-2666	-1718
	sig	8	8	8	11	594	8	8	8	86
	r+	45	45	45	44	18	45	45	45	37
	r−	0	0	0	1	27	0	0	0	8
CC	*stat*	-2666	-2666	-2666	-2666	-59	-2666	-2666	-2666	-2192
	sig	8	8	8	8	953	8	8	8	28
	r+	0	0	0	0	23	0	0	0	4
	r−	45	45	45	45	22	45	45	45	41
RASE	*stat*	-2666	-2666	-2666	-2547	-178	-2666	-2666	-2666	-1599
	sig	8	8	8	11	859	8	8	8	0,11
	r+	45	45	45	44	21	45	45	45	36
	r−	0	0	0	1	24	0	0	0	9
QAVE	*stat*	-1755	-2366	-950	-2,49	-1364	-2670	-2670	-2673	-2668
	sig	79	18	342	13	173	8	8	8	8
	r+	5,5	0	14,5	43,5	11	0	0	0	0
	r−	30,5	28	30,5	1,5	34	45	45	45	45
ERGAS	*stat*	-2666	-2666	-2666	-2666	-652	-2666	-2666	-2666	-1599
	sig	8	8	8	8	515	8	8	8	0,11
	r+	45	45	45	45	28	45	45	45	26
	r−	0	0	0	0	17	0	0	0	9
SSIM	*stat*	-2666	-2666	-2666	-2073	-2666	-2666	-2666	-2666	-2192
	sig	8	8	8	38	8	8	8	8	28
	r+	0	0	0	5	0	0	0	0	4
	r−	45	45	45	40	45	45	45	45	41
PSNR	*stat*	-2666	-2666	-2666	-2547	-533	-2666	-2666	-2666	-1125
	sig	8	8	8	11	594	8	8	8	0,26
	r+	0	0	0	1	27	0	0	0	13
	r−	45	45	45	44	18	45	45	45	32
NIQE	*stat*	-652	-2310	-889	-652	-652	-2073	-2666	-2547	-1689
	sig	515	21	374	515	515	38	8	11	91
	r+	17	42	15	28	17	40	45	44	30
	r−	28	3	30	17	28	5	0	1	6
BRISQUE	*stat*	-2666	-2666	-2666	-2547	-533	-2666	-2666	-2666	-1718
	sig	8	8	8	11	594	8	8	8	86
	r+	45	45	45	44	18	45	45	45	37
	r−	0	0	0	1	27	0	0	0	8
FD	*stat*	-2666	-2666	-2666	-2666	-2666	-2666	-2666	-2666	-2666
	sig	8	8	8	8	8	8	8	8	8
	r+	45	45	45	45	45	45	45	45	45
	r−	0	0	0	0	0	0	0	0	0

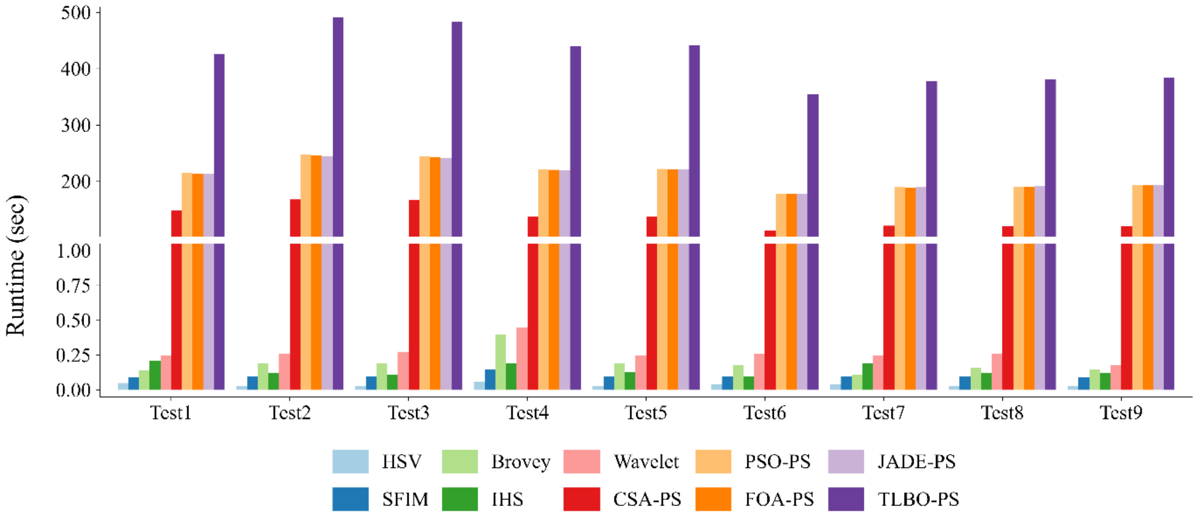

Figure 14. Runtime comparison.

It must be taken into account that population-based methods are iterative methods. In this case, 1500 iterations have been performed. Therefore, it would be possible to reduce the number of iterations of the algorithms to reduce their execution time and bring it closer to that of classical methods. This requires additional testing to determine how much the number of iterations can be reduced without significantly affecting the quality of the resulting images.

5. Conclusions

Optimizing limited on-board energy for Earth observation satellites is essential to extend their operational lifespan while minimizing costs. One strategy to conserve energy involves capturing high-resolution panchromatic (PAN) images, which require less energy than high-resolution multispectral (MSI) imaging and also reduce data transmission bandwidth. Although MSI provides detailed color information, synthesizing high-resolution multispectral images enables visually rich data suitable for analysis. The fusion technique known as pansharpening combines the high spatial resolution of PAN images with the color depth of MSI, producing pansharpened multispectral images (PSI) that maintain spatial and chromatic fidelity. High-quality pansharpening methods are necessary to avoid distortions in PSI, as traditional analytical models often introduce color and spatial inaccuracies.

This article presents an analytical model of an innovative pansharpening technique, called L0pan, designed to improve chromatic and spatial preservation. Structural parameters for the proposed method were calculated through evolutionary computing methods, including swarm-based and differential evolution techniques (CSA, TLBO, PSO, FOA, and JADE). Comparative experiments evaluated the performance of classical pansharpening methods (Brovey, HIS, HSV, SFIM, and wavelet) against the proposed approach paired with CSA, TLBO, PSO, FOA, and JADE. Among these, the CSA-based approach, CSA-PS, demonstrated superior PSI quality and computational efficiency, providing faster runtime than other population-based methods. Although the JADE-based approach achieved similar average quality for some metrics, it was slower. Notably, CSA-PS's results were comparable to those of the wavelet method, one of the classical techniques.

Based on L0-Norm principles, L0pan addresses outliers effectively, reducing errors commonly associated with traditional fusion methods, thereby creating PSI with a balanced spatial detail and chromatic accuracy vital for remote sensing. As demands for efficient, high-quality remote sensing increase, optimized pansharpening becomes critical, ensuring sustainable and reliable imaging. Research in this field prioritizes methods that maximize satellite efficiency and data quality, as seen in the development and optimization of methods like L0pan. These advancements play a crucial role in enabling Earth observation satellites to meet scientific and economic needs sustainably, supporting longer mission durations and greater image fidelity. Pansharpening algorithms like L0pan, which use mathematical frameworks to ensure data fidelity, advance the sustainability and scientific value of remote sensing imaging.

Key advantages of the proposed method include:

1. The adaptability of the proposed analytical model (Eq 15) for PSI synthesis across images with varying spectral bands.
2. Efficient solutions to the proposed model via population-based algorithms.
3. Enhanced chromatic and spatial preservation in PSI synthesis.
4. Statistically superior performance in PSI synthesis compared to widely used classical methods.

Future research will apply this model to a broader image set and examine the impact of population size and iteration count on CSA algorithm results. Adjusting these parameters is expected to reduce algorithm runtime, further enhancing the competitiveness of the proposed method.

Author contributions

M.A.G.: investigation; software; methodology. M.L.P.D.: investigation; formal analysis; validation; writing-original draft; writing-review & editing. E.B.: conceptualization; writing-original draft; writing-review & editing; supervision.

Conflict of interest

The authors declare that there are no conflicts of interest. Professor María-Luisa Pérez-Delgado is the Guest Editor for AIMS Mathematics and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

References

1. I. Amro, J. Mateos, M. Vega, R. Molina A. K. Katsaggelos, A survey of classical methods and new trends in pansharpening of multispectral images, *EURASIP J. Adv. Signal Process.,* **79** (2011), 1–22. https://doi.org/10.1109/MGRS.2021.306346
2. D. Wen, X. Huang, F. Bovolo, J. Li, X. Ke, A. Zhang, et al., Change detection from very-high-spatial-resolution optical remote sensing images: Methods, applications, and future directions, *IEEE Geosci. Remote Sens. Mag.,* **9** (2021), 68–101. https://doi.org/10.1109/MGRS.2021.3063465

3. H. Yao, R. Qin, X. Chen, Unmanned aerial vehicle for remote sensing applications—A review, *Remote Sens.,* **11** (2019), 1443. https://doi.org/10.3390/rs11121443

4. G. Licciardi, G. Vivone, M. D. Mura, R. Restaino, J. Chanussot, Multi-resolution analysis techniques and nonlinear PCA for hybrid pansharpening applications, *Multidim. Syst. Sign. Process,* **27** (2016), 807–830. https://doi.org/10.1007/s11045-015-0359-y

5. A. Azarang, H. Ghassemian, A new pansharpening method using multi resolution analysis framework and deep neural networks, In: *3rd International Conference on Pattern Recognition and Image Analysis (IPRIA),* Shahrekord, Iran: IEEE, 2017, 1–6. https://doi.org/10.1109/PRIA.2017.7983017

6. S. A. Elmasry, W. A. Awad, S. A. Abd El-hafeez, Review of different image fusion techniques: Comparative study, In: *Internet of Things—Applications and Future. Lecture Notes in Networks and Systems,* Springer Singapore, 2020, 41–51. https://doi.org/10.1007/978-981-15-3075-3_3

7. L. He, Y. Rao, J. Li, J. Chanussot, A. Plaza, J. Zhu, et al., Pansharpening via detail injection based convolutional neural networks, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.,* **12** (2019), 1188–1204. https://doi.org/10.1109/JSTARS.2019.2898574

8. K. Zhang, F. Zhang, W. Wan, H. Yu, J. Sun, J. Del Ser, E. Elyan, A. Hussain, Panchromatic and multispectral image fusion for remote sensing and earth observation: Concepts, taxonomy, literature review, evaluation methodologies and challenges ahead, *Inf. Fusion,* **93** (2023), 227–242. https://doi.org/10.1016/j.inffus.2022.12.026

9. M. A. Günen, Weighted differential evolution algorithm based pansharpening, *Int. J. Remote Sens.,* **42** (2021), 8468–8491. https://doi.org/10.1080/01431161.2021.1976874

10. P. Civicioglu, E. Besdok, Pansharpening of remote sensing images using dominant pixels, *Expert Syst. Appl.,* **242** (2024), 122783. https://doi.org/10.1016/j.eswa.2023.122783

11. P. Civicioglu, E. Besdok, Contrast stretching based pansharpening by using weighted differential evolution algorithm, *Expert Syst. Appl.,* **208** (2022), 118144. https://doi.org/10.1016/j.eswa.2022.118144

12. S. Yang, M. Wang, L. Jiao, Fusion of multispectral and panchromatic images based on support value transform and adaptive principal component analysis, *Inf. Fusion,* **13** (2012), 177–184. https://doi.org/10.1016/j.inffus.2010.09.003

13. H. R. Shahdoosti, MS and PAN image fusion by combining Brovey and wavelet methods, *arXiv preprint arXiv:170101996,* 2017. https://doi.org/10.48550/arXiv.1701.01996

14. V. Yilmaz, C. Serifoglu Yilmaz, O. Güngör, J. Shan, A genetic algorithm solution to the gram-schmidt image fusion, *Int. J. Remote Sens.,* **41** (2020), 1458–1485. https://doi.org/10.1080/01431161.2019.1667553

15. K. Amolins, Y. Zhang, P. Dare, Wavelet based image fusion techniques – An introduction, review and comparison, *ISPRS J. Photogramm. Remote Sens.,* **62** (2007), 249–263. https://doi.org/10.1016/j.isprsjprs.2007.05.009

16. G. Kaur, K. S. Saini, D. Singh, M. Kaur, A comprehensive study on computational pansharpening techniques for remote sensing images, *Arch. Computat. Methods Eng.,* **28** (2021), 1–18. https://doi.org/10.1007/s11831-021-09565-y

17. A. Chakraborty, A. K. Kar, Swarm intelligence: A review of algorithms, In: *Nature-Inspired Computing and Optimization. Modeling and Optimization in Science and Technologies,* Springer Cham, 2017, 475–494. https://doi.org/10.1007/978-3-319-50920-4_19

18. A. E. Hassanien, E. Emary, *Swarm intelligence: principles, advances, and applications*, CRC Press, 2018. https://doi.org/10.1201/9781315222455

19. J. Kennedy, R. Eberhart, Particle swarm optimization, In: *Proceedings of the IEEE International Conference on Neural Networks (ICNN'95),* Perth, WA, Australia, **4** (1995), 1942–1948. http://dx.doi.org/10.1109/ICNN.1995.488968

20. P. Civicioglu, E. Besdok, Colony-Based Search Algorithm for numerical optimization, *Appl. Soft Comput.,* **151** (2024), 111162. https://doi.org/10.1016/j.asoc.2023.111162

21. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput.-aided Des.,* **43** (2011), 303–315. https://doi.org/10.1016/j.cad.2010.12.015

22. Q. K. Pan, H. Y. Sang, J. H. Duan, L. Gao, An improved fruit fly optimization algorithm for continuous function optimization problems, *Knowl.-Based Syst.,* **62** (2014), 69–83. https://doi.org/10.1016/j.knosys.2014.02.021

23. W. Y. Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, *Knowl.-Based Syst.,* **26** (2012), 69–74. https://doi.org/10.1016/j.knosys.2011.07.001

24. J. Zhang, A. C. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*, Berlin. Heidelberg: Springer, 2009. http://dx.doi.org/10.1007/978-3-642-01527-4

25. S. Singh, H. Singh, G. Bueno, O. Deniz, S. Singh, H. Monga, P. N. Hrisheekesha, A. Pedraza, A review of image fusion: Methods, applications and performance metrics, *Digit. Signal Process.,* **137** (2023), 104020. https://doi.org/10.1016/j.dsp.2023.104020

26. J. G. Liu, Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details, *Int. J. Remote Sens.,* **21** (2000), 3461–3472. https://doi.org/10.1080/014311600750037499

27. H. M. Palancıoğlu, Histogram modification based pansharpening by using differential evolution algorithm, *Concurr. Comput.: Practic. Exp.,* **34** (2022), e7335. https://doi.org/10.1002/cpe.7335

28. M. A. Günen, U. H. Atasever, Remote sensing and monitoring of water resources: A comparative study of different indices and thresholding methods., *Sci. Total Environ.,* **926** (2024), 172117. https://doi.org/10.1016/j.scitotenv.2024.172117

29. M. L. Pérez-Delgado, M. A. Günen, A comparative study of evolutionary computation and swarm-based methods applied to color quantization, *Expert Syst. Appl.,* **231** (2023), 120666. https://doi.org/10.1016/j.eswa.2023.120666

30. M. L. Pérez-Delgado, M. E. Celebi, A comparative study of color quantization methods using various image quality assessment indices, *Multimed. Syst.,* **30** (2024), 40. https://doi.org/10.1007/s00530-023-01206-7

31. L. Wald, Quality of high resolution synthesised images: Is there a simple criterion?, In: *Third Conference: Fusion of Earth data: merging point measurements, raster maps and remotely sensed images.* SEE/URISCA, Sophia Antipolis, France, 2000, 99–103.

32. M. A. Günen, Comparison of histogram-curve fitting-based and global threshold methods for cloud detection, *Int. J. Environ. Sci. Technol.,* **21** (2024), 5823–5848. https://doi.org/10.1007/s13762-023-05379-6

33. A. Mittal, A. K. Moorthy, A. C. Bovik, Blind/referenceless image spatial quality evaluator, In: *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR),* Pacific Grove, CA, USA: IEEE, 2011, 723–727. https://doi.org/10.1109/ACSSC.2011.6190099

34. A. Mittal, R. Soundararajan, A. C. Bovik, Making a "completely blind" image quality analyzer, *IEEE Signal Process. Lett.,* **20** (2012), 209–212. https://doi.org/10.1109/LSP.2012.2227726

35. Z. Wang, A. C. Bovik, *Gradient estimates for solutions of nonlinear elliptic and parabolic equations*, Switzerland: Springer Cham, 2006. https://doi.org/10.1007/978-3-031-02238-8