IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 9, 2022, accepted March 5, 2022, date of publication March 10, 2022, date of current version March 17, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3158324

Exploratory Analysis of Different Metaheuristic Optimization Methods for Medical Image Enhancement

MUHTAHIR O. OLOYEDE[1], (Member, IEEE), ADEIZA J. ONUMANYI[2], (Member, IEEE), HABEEB BELLO-SALAU[3], (Member, IEEE), KARIM DJOUANI[1,4], (Member, IEEE), AND ANISH KURIEN[1], (Member, IEEE)

[1]F'SATI/Department of Electrical Engineering, Tshwane University of Technology, Pretoria 0183, South Africa
[2]Advanced Internet of Things, Next Generation Enterprises and Institutions, Council for Scientific and Industrial Research (CSIR), Pretoria 0001, South Africa
[3]Department of Computer Engineering, Ahmadu Bello University, Zaria 810211, Nigeria
[4]Laboratoire Images, Signaux et Systemes Intelligents (LiSSi), Université de Paris-Est Créteil (UPEC), 94000 Créteil, France

Corresponding author: Muhtahir O. Oloyede (tahir.oloyede@gmail.com)

This work was supported in part by the National Research Foundation of South Africa under Grant 90604.

ABSTRACT Metaheuristic optimization algorithms (MOAs) are popularly deployed for medical image enhancement (MIE) purposes. However, with an ever-increasing rate of newer MOAs being proposed in the literature, the question arises as to whether there exist any significant advantage(s) among these different MOAs, particularly as it pertains to MIE. In this paper, we explore this question by analyzing nine well-known MOAs for MIE, namely the artificial bee colony, cuckoo search, differential evolution, firefly, genetic algorithm, particle swarm optimization (PSO), covariance matrix adaptive evolutionary strategy (CMAES), whale optimization algorithm (WOA), and the grey wolf optimization (GWO) algorithms. First, instead of measuring an MOA's performance based on the number of generations, we adopted the fitness computation rate (FCR), which enables MOAs to be compared in a fairer sense. Secondly, we used a combination of a well-known transformation function and a robust evaluation function as our objective function in the MOAs considered in our study. Then, medical images were obtained from the Medpix database with representative samples selected from across the different parts of the body for MIE evaluation purposes. Within the constraints of the datasets used, the results indicate that, while the GWO and WOA algorithms performed slightly better empirically than the other methods over an average of 1000 Monte Carlo trials, there was little/no statistical significant difference between the other methods. The timing performance also demonstrates that there was no significant difference in the real-time processing speeds of the various MOAs, particularly when evaluated under the same FCR. As a consequence, preliminary findings from our study suggest that employing a range of past and current MOAs or proposing newer MOAs for MIE may not necessarily guarantee substantial comparative enhancement benefits. This might suggest that under high FCR levels, any MOA can be utilized for MIE.

INDEX TERMS Comparison, images, metaheuristic, optimization, performance.

I. INTRODUCTION

Medical images are important tools for detecting and diagnosing different medical conditions and ailments [1], [2]. However, the quality of medical images can often be degraded during the capture procedure due to factors such as noise interference, poor illumination, and artifacts. This may lead to the misdiagnosis of medical conditions, which can further exacerbate life threatening situations [3]. Consequently, it is imperative that effective approaches be developed for enhancing the quality of medical images [4].

There are different image enhancement approaches that aim to transform an input image towards obtaining a better, more detailed, and less noisy output image [5]–[12]. However, these image enhancement methods often produce either poorly or overly enhanced images, thus warranting the need for more effective techniques [13], [14]. To this effect, one early image enhancement method is the histogram

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

equalization (HE) technique, which operates by adjusting an image's contrast by either decreasing or increasing the global contrast of the image. Such an approach is often plagued by the problem of over-enhancement, particularly when low occurring intensities are transformed and combined with neighboring high occurring intensities [9]. As a solution, the bi-histogram equalization (BHE) was introduced for its ability to maintain the quality of an original image. Nevertheless, BHE becomes limited when the pixel distribution of an image is asymmetrical, which often occurs under real-life conditions. Other existing techniques include the logarithm transformation, gamma correction, and adaptive histogram equalization (AHE) methods, which were proposed to have less computational complexities. However, they become less effective under complex lighting differences [15]. Other approaches have been proposed such as the linear contrast stretching (LCS) method, which makes use of the concept of linear transformation wherein the gray level's dynamic range is increased. However, the need to configure manually the threshold value of the LCS often limits its performance by making it susceptible to wrong threshold values, thus leading to poor output images [16].

In terms of medical image enhancement (MIE), the authors in [3] presented an enhancement technique comprising of two modules. The fuzzy dissimilarity histogram (FDH) is conducted in the first module, which uses the fuzzy neighborhood dissimilarity metric to evaluate each pixel via fuzzy rules. The contrast and entropy parameters were used to obtain the fuzzy dissimilarity clip limit using a fuzzy inference system (FIS). Then, the enhanced image was derived can be obtained by equalizing the FDH. Contextual intensity transformation (CIT) is applied in the second module to the FDH-based equalized output to generate a final enhanced medical image. Some studies were carried out using images from the BRATS 2015 dataset and satisfactory results were obtained using the peak signal-to-noise ratio (PSNR), entropy, and the contrast ratio. In [17], a morphological transformation process was conducted on medical images to improve their contrast. A disk-shaped mask was used in the bottom-hat and top-hat transformation process, which improves the method's performance. The enhanced medical images were further generated by an iterative exfoliation process. The experimental results obtained demonstrated that the contrast of medical images were improved. In another article [18], a feedback graph attention convolutional neural network approach was presented that considers the global structure of an image by building a graph network from image sub-regions. The model comprises of three main parts, which include the parallel graph similarity branch, a feedback mechanism that refines the low-level representations using a recurrent structure, and an elimination of the artifacts to generate higher quality images. The results obtained showed satisfactory performance when compared to other approaches. In [4], an adaptive MIE technique was presented, wherein training samples were clustered into neighborhoods. The low-resolution space was then sectioned by exchanging the dictionary atoms with the cluster patch centers for computing K-nearest patches. These were then projected to map the low resolution patches upon the high resolution area. The authors conducted experiments using two medical datasets and showed that satisfactory results were obtained. Furthermore, in [19], a contrast optimization approach based on genetic algorithm was proposed by introducing a kernel that detects the edges of an image. The results obtained from their experiments demonstrated that the proposed approach was effective.

Owing to the subjective nature of the image enhancement process, several metaheuristic optimization algorithms (MOA) have found great application for image enhancement and with remarkable results to show [20]. The use of MOAs for MIE purposes has been demonstrated several times in the literature to outperform the existing methods mentioned above [11], [20]. Consequently, the literature has literally exploded with a long list of different MOAs, with many being deployed for MIE functions [21]. However, the question arises as to whether there exist any significant difference in both the qualitative and quantitative performances of these different MOAs. Hence, this paper investigates the performance of some selected well-known MOAs for MIE, namely, the artificial bee colony (ABC), cuckoo search optimization (CSO), differential evolution (DE), firefly, genetic algorithm (GA), particle swarm optimization (PSO), covariance matrix adaptive evolutionary strategy (CMAES), whale optimization algorithm (WOA), and the grey wolf optimization (GWO) algorithms. Firstly, our concern was to ensure that all MOAs were compared fairly, and to this effect we adopted the fitness computation rate (FCR) instead of the number of generations. By using the FCR, we can ensure that the objective function is computed the same number of times for all MOAs, thus guaranteeing that the computational resources of the MIE machine are consumed almost equally by all MOAs. Next, we deployed the transformation function proposed in [20] coupled with a robust evaluation function used in [11] to serve as the objective function for MIE purposes. Finally, a number of performance parameters were used to analyze the different MOAs in the qualitative and quantitative sense. As a result of our findings, the current article's contributions can be described as follows:

1) We explore whether there exists any significant advantage(s) in using different MOAs for MIE. To this effect, nine state-of-the-art MOAs were considered, which were selected across both evolutionary and swarm-based classes. This distinguishes the current article as having researched a considerable number of MOAs only for the purpose of MIE.
2) By using the FCR as a metric of evaluation instead of the number of generations, we discovered that, although the GWO and WOA algorithms yielded slightly better empirical results than the other methods, nevertheless, there was little/no significant difference in both qualitative and quantitative measures among the different MOAs.

3) Physical timing performances were measured across the different MOAs with little/no statistical significant difference to show. Thus, this confirmed that using a fixed FCR translates to the same timing performance across most MOAs.

The remainder of the paper is structured as follows: Section II presents the method used in this study, which includes the transformation and evaluation functions used, a description of the different MOAs, the datasets and performance evaluation metrics utilized. Section III presents the results and discussion from the different experiments carried out and conclusions are drawn in Section V.

II. METHOD

In order to produce an enhanced output image, most image enhancement techniques typically use an effective transformation function to map the intensity values of an original input image [22]. Such a transformation function will often be characterized by parameters whose values are required to be optimally determined in order to improve performance. To this effect, metaheuristic-based approaches are normally applied to optimize such parameters of a transformation function via an objective function (typically called an evaluation function) [20]. Consequently, we first describe the specific transformation function deployed in our study. Then we discuss the evaluation function used in each MOA. We describe each MOA, focusing specifically on the modifications made to ensure that the FCR is accurately measured. Then the performance evaluation metrics used and the medical image database considered in our study are described.

A. TRANSFORMATION FUNCTION

The transformation function proposed in [20] was used in our study. Following its formulation, the function has remained highly effective and thus widely applied in the literature. The function is stated as follows:

$$T(f(i,j)) = g(i,j) = k\left(\frac{M}{\sigma(i,j)+b}\right) \times [f(i,j) - c \times m(i,j)] + m(i,j)^{a} \quad (1)$$

where T denotes the transformation function, $f(i,j)$ denotes the input image with pixels located at pixel (i,j), and $g(i,j)$ refers to the output image with modified gray-level intensities at each pixel (i,j). The parameters a, b, c, and k play the following roles: Parameter a represents the brightening bias on the output image, which allows control of the amount of smoothening required in the output image. Parameter b ensures that a zero-standard deviation value in the local neighborhood pixels does not have much whitening effect on the final output image. Parameter c allows a fraction of the mean to be subtracted from the original pixels of the input image, and it also controls the amount of darkening effects produced in the output image. The parameter k creates an unbiased balance between pixels existing in the mid-range boundaries of the gray scale. This prevents the pixels from becoming either too dark or too bright during the enhancement process. The statistics $m(i,j)$ and $\sigma(i,j)$ denote the gray-level mean and standard deviation computed for pixels within a neighbourhood (window) centered at (i,j) and having $n \times n$ pixels. The global mean M is computed as

$$M = \sum\nolimits_{i=0}^{H-1} \sum\nolimits_{j=0}^{V-1} f(i,j), \quad (2)$$

where H and V denote the row (horizontal) and column (vertical) size of the image.

B. EVALUATION FUNCTION

An evaluation function is a mathematical expression that computes the quality of an enhanced image $g(i,j)$ with an aim to replace dependencies on subjective human assessments. It is used to determine the optimal parameter values of a transformation function T towards improving the performance of an MIE technique. In this study, we used the evaluation function proposed in [11]. It was selected for the following reasons: First, it integrates four different key metrics, thus making it robust compared to other methods. Second, the metrics were normalized, which presents a linear function that assigns specific boundaries to pixels in the dark and bright regions. Lastly, it is easily adaptable for use in different MOAs.

Further, the metrics considered in designing the evaluation function include the number of edge pixels, entropic measure, the number of foreground pixels, and the peak signal to noise ratio (PSNR). The steps involved in computing the function are as follows:

1) The number of edge pixels N_g in the enhanced image is computed using a Sobel threshold S_f, which is automatically computed from the original image $f(i,j)$ using a Sobel edge detector.
2) The Sobel threshold S_f is then used in the Sobel edge detector to obtain the edge intensities $E_g(i,j)$ of the enhanced image. It is noted that in addition to being invariant, S_f was considered for computing $E_g(i,j)$ in order to ensure a fair comparison between the original image and the different instances of the enhanced image. Thus, the number of edge pixels N_g in the enhanced image is obtained as:

$$N_g = \sum_{i=1}^{H} \sum_{j=1}^{V} E_g(i,j) \quad (3)$$

3) The variance $\vartheta_g(i,j)$ of $g(i,j)$, and the variance $\vartheta_f(i,j)$ of $f(i,j)$ are computed within a neighborhood (window) having $n \times n$ pixels. And a threshold value η_f is automatically computed for $\vartheta_f(i,j)$ using Otsu's threshold algorithm.
4) Then, a binary function $D_g(i,j)$ is used to classify the pixels belonging to the foreground objects in the enhanced image as follows:

$$D_g(i,j) = \begin{cases} 1 & if\ \vartheta_g(i,j) \geq \eta_f \\ 0 & if\ otherwise \end{cases} \quad (4)$$

for $i = 1, 2, \ldots, H$ and $j = 1, 2, \ldots, V$.

5) Thereafter, the number of pixels ϕ_g belonging to the foreground objects in $g(i, j)$ is computed as

$$\phi_g = \sum_{i=1}^{H} \sum_{j=1}^{V} D_g(i, j) \tag{5}$$

6) The entropic measure β_g of $g(i, j)$ is also given as

$$\beta_g = \begin{cases} -\sum_m \Omega_m \log(\Omega_m) & \text{for } \Omega_m \neq 0 \\ 0 & \text{for } \Omega_m = 0 \end{cases} \tag{6}$$

where Ω_m is the frequency of pixels having gray levels in the histogram bin $m = 1, \ldots, 256$.

7) The PSNR ρ_g of $g(i, j)$ is obtained as

$$\rho_g = 10 \log_{10} \left[\frac{(L-1)^2}{MSE} \right] \tag{7}$$

where L is the maximum pixel intensity value in $g(i, j)$ and the mean squared error MSE is given as

$$MSE = \frac{1}{H \times V} \sum_{i=1}^{H} \sum_{j=1}^{V} |f(i, j) - g(i, j)|^2 \tag{8}$$

8) Based on the parameters computed in (3) – (7), the evaluation function $\mathcal{E}$ is stated as

$$\mathcal{E} = 1 - \exp\left(-\frac{\rho_g}{100}\right) + \frac{N_g + \phi_g}{H \times V} + \frac{\beta_g}{8} \tag{9}$$

where $\mathcal{E}$ is a linear combination of the normalized values of the different metrics described in (3) – (7). By normalizing each metric in (8), each parameter are bounded between 0 and 1. Thus, based on this linear combination, the evaluation function is described by a defined scale bounded between a minimum value of zero and maximum value of four. A minimum value of zero represents an entirely dark enhanced image, while a maximum value of four represents an entirely bright enhanced image.

C. METAHEURISTIC OPTIMIZATION ALGORITHMS

This section discusses the rationale for selecting the algorithms explored in this study. The FCR concept is discussed, as well as the various MOAs and their associated input/output parameters.

1) CHOICE OF MOAs

Some of the earliest MOAs are the evolutionary techniques, with the GA making its debut in 1975 [23]. Since then, a plethora of MOAs have been proposed in the literature. To this end, with an inclusion of the GA, we look at three generations of MOAs starting from the 1990s to the present, with a ten-year gap between each generation. The CMAES algorithm was chosen in part because it has been one of the top performers at the Genetic and Evolutionary Computation Conference (GECCO) between 2009 to 2018 [24], a conference which showcases the most recent high-quality advances in genetic and evolutionary computation. We also considered algorithms with less than five occasions where the objective function is computed within a single iteration of the algorithm. Thus, in the order of the year of development and the number of times the fitness function ($\mathcal{F}$) is computed in a single iteration of a single solution, the following significant algorithms were chosen and listed in this format (Algorithm (year, $\mathcal{F}$)): GA (1975,4), DE (1995,2), PSO (1995,2), CMAES (1996,2), ABC (2005,4), Firefly (2007,2), CSO (2009,3), GWO (2014,1), and WOA (2016,1). Nonetheless, with an almost endless number of MOAs being suggested and deployed in the literature, we do not claim to have researched all MOAs, which is nearly impossible to achieve; thus, we consider this article as only a stepping stone to further empirical research in this and other areas of metaheuristic research.

2) FITNESS COMPUTATION RATE

The FCR is defined here as the number of times the fitness (i.e objective) function is computed during the execution of an MOA. In our study, such a fitness/objective function is termed the evaluation function, and both terms are used interchangeably.

In many MOA-based articles, the number of generations is often used as the termination criteria for an MOA. However, in recent publications, such as in [25]–[28], it has been argued that the number of generations does not guarantee fair comparison between different MOAs because an MOA may compute the fitness function more times than other MOAs. Consequently, it is fairer to use the FCR since it ensures that the number fitness evaluations consumed is kept constant while executing the different MOAs. However, because the FCR is often difficult to compute mathematically, it is ideal to measure it algorithmically during runtime [26], [27], which is what we have done for each MOA considered in our study as presented in their different pseudocodes (see Algorithms 1 - 9).

3) COMMON AND SPECIFIC INPUT AND OUTPUT PARAMETERS

The different MOAs considered in our study are governed by different input parameters, some of which are common across all MOAs. Thus, we define these parameters in Table 1 including their respective values obtained after a manual fine-tuning process. These parameter values are thus the best performing values, which were used for the MIE comparative exercise.

For the common inputs, there were a total of four different parameters to be optimized in our study (i.e. $D = 4$, where D is the total number of parameters) namely, the a, b, c, and k parameters of the transformation function, which have been discussed in Section II-A. In our pseudocodes, these parameters are denoted as $X = \{x_1, x_2, \ldots, x_D\}$, where for example, parameter a is x_1, b is x_2, c is x_3, and k is x_4. Each parameter is constrained within certain bounds, and the following bounds were used: $2 \leq a \leq 2.5$; $0.3 \leq b \leq 0.5$; $0 \leq c \leq 3$; and

TABLE 1. Input and output parameters used across the different MOAs.

Common inputs	• Define the parameters X to be optimized, • Set the constraints of each parameter, • Evaluation function (i.e. objective function) $\mathcal{E}$, • Population size (number of nests) P, • Set maximum fitness computation rate FCR_{max}
ABC	• Abandonment (Trial) limit = 12 • Acceleration Coefficient = (lower bound = -1, upper bound = 1) • Number of Onlooker Bees = 10
CSO	• Probability of discovering alien eggs $P_a = 0.2$.
DE	• Crossover probability = 0.5 • Scaling factor = (lower bound = 0.2, upper bound = 0.8)
Firefly	• Mutation coefficient = 0.2 • Attraction coefficient base value = 2 • Mutation coefficient damping ratio = 0.98 • Light absorption coefficient = 1
GA	• Selection rate = 0.8 • Selection operator = Roullette wheel • Chromosome length = 16 • Mutation rate = 0.2 • Crossover type = Single point crossover • Elitism rate = 0.5 • Crossover rate = 0.8
PSO	• Inertia weight (adaptive) = (0.1,1.1) • Self adjustment weight = 1.5 • Acceleration coefficients = Uniformly distributed (0,1) random vector • Social adjustment weight = 1.5
CMAES	• All working parameters of the CMAES, save the population size, are determined within the method, therefore it is essentially user-parameter free.
WOA	• There are no additional parameters for the WOA except the common inputs
GWO	• Except for the common inputs, the GWO technique has no other user defined parameters.
Outputs	• Best fitness value $\mathcal{E}_{max}$, • Optimal parameters X_{opt}, for all D parameters

$3 \leq k \leq 4$, which were obtained from extensive experiments conducted in [11]. Other common inputs already well known in the literature are the evaluation function $\mathcal{E}$, population size P, and the maximum fitness computation rate FCR_{max}. All other specific parameters related to the different MOAs and their respective values used in our study are highlighted in Table 1.

4) ARTIFICIAL BEE COLONY

The ABC algorithm is modeled based on the intelligent foraging behavior of the honey bee swarm [29]. It follows the process by which sets of honey bees (i.e swarm) accomplish their tasks successfully through social cooperation. Technically, the ABC algorithm employs three types of bees in the swarm, namely: employed (recruitment) bees, onlooker bees, and scout bees. The concept of a swarm corresponds to the total population P of solutions, whereas each bee refers to a single solution X_p^{pr}, i.e. the p^{th} solution in the population.

The ABC algorithm initializes by generating a random population of solutions, then image transformation is performed per p^{th} solution X_p^{pr} and the fitness of each transformed image is computed. The best solutions are then obtained and the FCR is verified to determine whether the process is terminated or not. In the recruitment stage, new solutions X_p^{new} are generated via:

$$X_p^{new} = X_p^{pr} + \phi_p \times (X_p^{pr} + X_k^{pr}) \tag{10}$$

where ϕ_p refers to the acceleration coefficient, which is a uniform random number generated between -1 and 1 per p^{th} solution, X_p^{pr} is the p^{th} previous solution obtained during the initialization phase, and X_k^{pr} is a k^{th} randomly selected solution, where $p \neq k$. After all employed bees have been evaluated, the algorithm proceeds to the onlooker bees stage. In this stage, solutions are improved via a probabilistic selection process based on the roulette wheel mechanism described as

$$Pr_p = \frac{E_p}{\sum_i E_i} \tag{11}$$

where E_p is the fitness value of the p^{th} solution in the population. Essentially, it is obvious from (11) that the better the p^{th} solution, the higher the probability of the solution being selected.

In the third (scout bees) stage, the algorithm determines a new p^{th} solution as follows:

$$X_p^{new} = L_d + \phi_{p,d} \times (U_d - L_d) \tag{12}$$

where $\phi_{p,d}$ is acceleration coefficient generated via a uniform random number within the bounds [0, 1], L_d and U_d are the lower and upper boundaries of the d^{th} dimension. The pseudocode of the ABC algorithm tailored for MIE is summarized in Algorithm 1.

5) CUCKOO SEARCH OPTIMIZATION

The CSO algorithm is discussed following its presentation in [30], and interested readers can access further details in the same reference. The algorithm mimics the biological hatching process of the cuckoo bird. Biologically, some cuckoo bird species typically lay their egg(s) in a foreign bird's nest in order to be hatched by the foreign bird. In translating this concept into an algorithm, the cuckoo's egg is denoted as a solution where all the eggs in the nest of the foreign bird constitute a population of solutions (i.e. many eggs). Thus, the algorithm iterates by replacing the not-so-good solutions in the population with better solutions. To achieve this, it uses a $L\dot{e}vy$ flight model to update its solution as follows [30]:

$$X_p^{new} = X_p^{pr} + \alpha \otimes L\dot{e}vy(\lambda) \tag{13}$$

where X_p^{new} denotes the p^{th} new solution in the population P; the flight function $L\dot{e}vy$ is computed based on the Mantegna's algorithm using a parameter λ, which denotes the $L\dot{e}vy$ walk

Algorithm 1 The ABC Algorithm for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
for $p = 1$ to $\mathcal{P}$ **do**
Generate random parameter values X_p^{pr},
Perform image transformation using (2) based on X_p^{pr}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} == X_p^{pr}$)
Terminate process if $FCR == FCR_{max}$
end for
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
for $p = 1$ to P **do**
Phase 1: Recruited bees stage
Get new solution X_p^{new} using (10)
Perform image transformation using (2) based on X_p^{new}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
Terminate process if $FCR == FCR_{max}$
end for
Phase 2: Onlooker bees stage
for $p = 1$ to P **do**
Select a p^{th} solution using (11)
Improve the p^{th} solution X_p^{new} using (10)
Perform image transformation using (2) based on X_p^{new}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
Terminate process if $FCR == FCR_{max}$
end for
Phase 3: Scout bees stage
for $p = 1$ to P **do**
Obtain a new p^{th} solution X_p^{new} using (12)
Perform image transformation using (2) based on X_p^{new}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
Terminate process if $FCR == FCR_{max}$
end for
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

parameter, α is the step size related to the scale of the problem of interest, and $\otimes$ symbol means entry wise multiplication. The *Lèvy* flight function provides a random walk with a random step length being drawn from a *Lèvy* distribution $L\grave{e}vy \sim u = v^{-\lambda}$, $(1 < \lambda \leq 3)$.

The CSO algorithm for MIE is presented in Algorithm 2, with an aim to show how the FCR is computed within the algorithm. Similar to most MOAs, an initial population of solutions is randomly generated in the initialization phase (see Algorithm 1) wherein each p^{th} solution X_p^{pr} is generated randomly within the parameter constraints. The algorithm then iterates following the FCR, which is continuously checked to terminate the algorithm whenever $FCR == FCR_{max}$. Since the MIE task is a maximization problem (i.e. to obtain an enhanced image with the largest fitness value), thus, the solution X_p^{pr} that yields the largest $\mathcal{E}$ values is saved as the best solutions. Then, new improved solutions are obtained via (13). To accomplish this, in the first phase (i.e. the get cuckoo stage), new solutions are obtained using the *Lèvy* flight function based on Mantegna's algorithm (see [30] for details). Thereafter, poorer solutions are expunged and replaced in the second phase (i.e. the abandon nest stage) based on the P_a parameter. Then, the algorithm continues to iterate until $FCR == FCR_{max}$, after which the best solutions and fitness values are obtained.

Algorithm 2 The CSO Algorithm for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
for $p = 1$ to $\mathcal{P}$ **do**
Generate random parameter values X_p^{pr},
Perform image transformation using (2) based on X_p^{pr}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} == X_p^{pr}$)
Terminate process if $FCR == FCR_{max}$
end for
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
for $p = 1$ to P **do**
Phase 1: Get Cuckoo stage
Get new parameters X_p^{new} randomly by *Lèvy* flight using Mantegna's algorithm
Perform image transformation using (2) based on X_p^{new}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
Terminate process if $FCR == FCR_{max}$
end for
Phase 2: Abandon nest stage
for $p = 1$ to P **do**
Empty a fraction of the worst solutions based on P_a using biased/selective random walks
Update the new solutions X_p^{new} using Eq. 13
Perform image transformation using (2) based on X_p^{new}
Compute fitness $\mathcal{E}_p$ of transformed image using (9)
$FCR++$ ***Increment FCR counter***
If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
Terminate process if $FCR == FCR_{max}$
end for
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

6) DIFFERENTIAL EVOLUTION

The DE algorithm proposed in [31] works using a population of candidate solutions, called agents. These agents explore the search space towards an optimal solution by combining the positions of existing agents within the population. Similar to other MOAs, it adopts an initialization phase where starting solutions are generated randomly. The algorithm then selects three agents (i.e. solutions) a, b, and c randomly from the population, which are different from the solution being evaluated. Proceeding to the mutation stage, the algorithm obtains new solutions as follows:

$$X_p^{new} = a_p + \beta \times (b_p - c_p) \tag{14}$$

where β is a scaling factor obtained using a uniform random generator constrained between a lower and upper bound value, whose values are stated in Table 1.

Algorithm 3 The DE Algorithm for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
for $p = 1$ to $\mathcal{P}$ **do**
  Generate random parameter values X_p^{pr},
  Perform image transformation using (2) based on X_p^{pr}
  Compute fitness $\mathcal{E}_p$ of transformed image using (9)
  $FCR++$ ***Increment FCR counter***
  If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
  Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
  Terminate process if $FCR == FCR_{max}$
end for
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
  for $p = 1$ to P **do**
    Phase 1: Mutation stage
    Select three solutions a, b, and c randomly from the population
    Get new solution X_p^{new} using (14)
    Phase 2: Crossover stage
    Obtain a uniform random number $r_p \sim \mathcal{U}(0, 1)$
    if $r_p < P_c$ **then**
      Maintain new solution X_p^{new}
    else
      Maintain previous solution
    end if
    Perform image transformation using (2) based on X_p^{new}
    Compute fitness $\mathcal{E}_p$ of transformed image using (9)
    $FCR++$ ***Increment FCR counter***
    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
    Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Terminate process if $FCR == FCR_{max}$
  end for
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

Then, in the second (crossover) stage, the algorithm updates the p^{th} solution using (14) on the condition that $r_p < P_c$, else it simply maintains the previous solution, where r_p is a uniformly distributed random number $r_p \sim \mathcal{U}(0, 1)$ and P_c is the crossover probability. Once the updates of all solutions are completed, the fitness values of the entire population are computed, and the best values are saved if they are better than the previous values. We summarize the pseudocode of the DE algorithm for MIE in Algorithm 3.

Algorithm 4 The Firefly Algorithm for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
for $p = 1$ to $\mathcal{P}$ **do**
  Generate random parameter values X_p^{pr},
  Perform image transformation using (2) based on X_p^{pr}
  Compute fitness $\mathcal{E}_p$ of transformed image using (9)
  $FCR++$ ***Increment FCR counter***
  If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
  Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
  Terminate process if $FCR == FCR_{max}$
end for
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
  for $p = 1$ to P **do**
    for $k = 1$ to P **do**
      Compute the distance $r_{p,k}$ using (16)
      Get new solution X_p^{new} using (15)
      Perform image transformation using (2) based on X_p^{new}
      Compute fitness $\mathcal{E}_p$ of transformed image using (9)
      $FCR++$ ***Increment FCR counter***
      If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
      Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
      Terminate process if $FCR == FCR_{max}$
    end for
  end for
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

7) FIREFLY ALGORITHM

The firefly algorithm works by modeling the flashing behavior of fireflies. It was proposed in [32] based on the concept that fireflies (i.e. solutions) will be attracted within the population based on an attractiveness property, which is proportional to their brightness. Thus, less brighter fireflies will migrate towards brighter ones. Essentially, the algorithm updates any pair of fireflies X_p^{pr} and X_k^{pr} as follows:

$$X_p^{new} = X_p^{pr} + \beta \times exp[-\gamma \times r_{p,k}^2] \times (X_p^{pr} - X_k^{pr}) + \alpha \times \epsilon \tag{15}$$

where X_p^{pr} is the previous p^{th} solution and X_k^{pr} is a distinct k^{th} solution selected from the population, β is the attraction coefficient, γ is the light absorption coefficient, α is the mutation coefficient that controls the step size, and ϵ is a uniformly distributed random parameter drawn between 0 and 1.

First, an initial population of solutions is generated similar to the other MOAs considered in our study. The distance $r_{p,k}$ between any pair of fireflies p and k (i.e. between two solutions) is then computed as

$$r_{p,k} = \frac{norm(X_p^{pr} - X_k^{pr})}{d_{max}} \quad (16)$$

where $norm(\cdot)$ is the Euclidean norm and d_{max} is farthest normalize distance between any pair of solutions in the population, computed as

$$d_{max} = norm(U - L) \quad (17)$$

where U and L are the largest and smallest values across all parameters to be optimized. Then, new solutions X_p^{new} are obtained using (15). Afterwards, the image is transformed based on the new solution and then the fitness of the enhanced image is computed and updated accordingly. The algorithm terminates when $FCR == FCR_{max}$. A summary of the entire procedure is presented in Algorithm 4.

8) GENETIC ALGORITHM

The GA is modeled after the natural selection process and it is intended to produce high-quality solutions to optimization problems [23]. Similar to other MOAs, it begins with the initialization phase where candidate solutions are generated randomly and their respective fitness values are computed.

In the first phase (i.e. the selection phase), the fittest solutions are selected as the parent indices based on the roulette wheel mechanism. The selected parents (i.e. candidate solutions) are then subjected to the crossover operator in the crossover stage in order to produce new offsprings (i.e. new solutions) X_p^{new}. These new solutions are then used in the transformation function to generate an enhanced image, whose fitness value is computed using (9). The best fitness is saved and the best solutions are processed in the third stage (i.e. the mutation phase).

In the mutation phase, a number of solutions are randomly selected and their values are changed to form new solutions as follows

$$X_p^{new} = X_p^{pr} + \sigma \times randn(D) \quad (18)$$

where $randn(\cdot)$ is a normal random number generator, D is the dimensionality, and σ is obtained as

$$\sigma = 0.1 \times (U - L) \quad (19)$$

where U and L are the lower and upper constraints of the different parameters to be optimized. Then, the fitness of the newly mutated solutions are computed and the best values are saved, while the iteration proceeds until $FCR == FCR_{max}$, which serves to terminate the algorithm. The entire algorithmic process is documented in Algorithm 5.

Algorithm 5 The Genetic Algorithm for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
for $p = 1$ to $\mathcal{P}$ **do**
  Generate random parameter values X_p^{pr},
  Perform image transformation using (2) based on X_p^{pr}
  Compute fitness $\mathcal{E}_p$ of transformed image using (9)
  $FCR++$ ***Increment FCR counter***
  If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
  Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
  Terminate process if $FCR == FCR_{max}$
end for
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
  for $p = 1$ to P **do**
    Stage 1: Selection phase
    Performance selection using the roulette wheel mechanism
    Stage 2: Crossover phase
    Apply crossover based on selected parents to obtain new solution X_p^{new}
    Perform image transformation using (2) based on X_p^{new}
    Compute fitness $\mathcal{E}_p$ of transformed image using (9)
    $FCR++$ ***Increment FCR counter***
    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
    Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Terminate process if $FCR == FCR_{max}$
    Stage 3: Mutation phase
    Select new parents randomly
    Apply mutation using (18) to obtain new solution X_p^{new}
    Perform image transformation using (2) based on X_p^{new}
    Compute fitness $\mathcal{E}_p$ of transformed image using (9)
    $FCR++$ ***Increment FCR counter***
    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
    Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Terminate process if $FCR == FCR_{max}$
  end for
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

9) PARTICLE SWARM OPTIMIZATION

The PSO algorithm models the movement of particles (i.e. candidate solutions) over a search space towards converging to an optimal solution. The evolution of each solution (or particle) in the population (i.e. a swarm) is governed by its local best position and guided towards better solutions via a velocity parameter. Essentially, the algorithm updates a particle's velocity using

$$V_p^{new} = \omega \times V_p^{pr} + c1 \times rand(D) \times (X_p^{best} - X_p^{pr}) + c2 \times rand(D) \times (G^{best} - X_p^{pr}) \quad (20)$$

where V_p^{pr} is the previous velocity of the p^{th} solution, $c1$ and $c2$ are the personal and global learning coefficients, ω is the inertial weight, X_p^{best} is the best solution of the p^{th} particle, X_p^{pr} is the previous solution of the p^{th} particle, and G^{best} is the global best solution across the entire population. Once a particle's velocity has been updated, the algorithm proceeds to obtain a new solution as follows

$$X_p^{new} = X_p^{pr} + V_p^{new} \quad (21)$$

Then, the new solutions X_p^{new} are applied in the transformation function of (1) to obtain an enhanced image. Thereafter, the fitness of the enhanced image is computed using (9) and compared with previous fitness values. If the new fitness value is larger than the previous, then the solutions are saved. The algorithm proceeds iteratively until $FCR == FCR_{max}$, which serves as the termination criteria. A summary of the pseudocode is presented in Algorithm 6.

Algorithm 6 The Particle Swarm Optimization for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
for $p = 1$ to P **do**
  Generate random parameter values X_p^{pr},
  Perform image transformation using (2) based on X_p^{pr}
  Compute fitness $\mathcal{E}_p$ of transformed image using (9)
  $FCR++$ ***Increment FCR counter***
  If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
  Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
  Terminate process if $FCR == FCR_{max}$
end for
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
  for $p = 1$ to P **do**
    Update velocity of particle using (20)
    Obtain new solution X_p^{new} using (21)
    Perform image transformation using (2) based on X_p^{new}
    Compute fitness $\mathcal{E}_p$ of transformed image using (9)
    $FCR++$ ***Increment FCR counter***
    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
    Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Terminate process if $FCR == FCR_{max}$
  end for
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

10) COVARIANCE MATRIX ADAPTIVE EVOLUTIONARY STRATEGY

The covariance matrix adaptation evolution strategy (CMAES) is a type of numerical optimization method that falls under the umbrella of evolutionary algorithms and evolutionary computation. In the CMAES algorithm proposed in [33], new candidate solutions are sampled from a multivariate normal distribution in $\mathbb{R}^D$. Recombination is equivalent to choosing a new mean value for the distribution. Mutation is the addition of a random vector, a perturbation with zero mean. A covariance matrix C represents the pairwise relationships between the variables in the distribution. The covariance matrix adaptation (CMA) approach is used to update the distribution's covariance matrix. This is especially important if the fitness function is ill-conditioned.

The adaptation of the covariance matrix is analogous to learning a second order model of the underlying objective function in classical optimization, comparable to the approximation of the inverse Hessian matrix in the quasi-Newton technique. In contrast to most conventional approaches, the CMAES method makes less assumptions about the underlying objective function. The approach does not need derivatives because it just employs a ranking (or, equivalently, sorting) of candidate solutions.

Algorithm 7 The CMAES for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
Initialize the CMAES parameters M, σ, ω, $p_c = 0$, $p_\sigma = 0$, and C
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
  for $p = 1$ to P **do**
    Sample new solutions X_p^{new} using (23)
    Perform image transformation using (2) based on X_p^{new}
    Compute fitness $\mathcal{E}_p$ of transformed image using (9)
    $FCR++$ ***Increment FCR counter***
    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
    Save best solutions (i.e. $X^{best} == X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Terminate process if $FCR == FCR_{max}$
  end for
  Sort the population of solutions X_p^{sort} = sort(X_p^{new})
  Save present mean as $M^{pr} = M$
  Update the mean to obtain new mean values M^{new} using (25)
  Update p_σ
  Update p_c
  Update C
  Update σ
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

The main loop of the algorithm is divided into three sections: 1) sampling of new solutions, 2) re-ordering (sorting) of sampled solutions based on fitness, and 3) updating of internal state variables based on the re-ordered samples. The algorithm's pseudocode is given in Algorithm 7. The sequence of the update assignments matters: M must be updated first, p_σ and p_c must be updated before C, and σ must be updated last. The update equations for the five state variables are presented below. First, we note that the search space dimension D and the iteration step p are given, which are determined by the number of solutions that are calculated. Then, the following are the five state variables:

1) M_p in $\mathbb{R}^D$, the mean of the distribution and the current preferred solution to the optimization problem,
2) $\sigma_p > 0$, the step-size
3) C, a symmetric and positive-definite $n \times n$ covariance matrix with the initialize matrix given as an identity matrix $C_0 = I$ and
4) p_σ in $\mathbb{R}^D$, p_c in $\mathbb{R}^D$, two evolution paths, initially set to the zero vector.

The iteration starts with sampling $P > 1$ candidate solutions $X_p^{new} \in \mathbb{R}^D$ from a multivariate normal distribution $\mathcal{N}(M_p, \sigma_p^2 C_p)$, i.e. for $p = 1, \ldots, P$, such that X_p^{new} is obtained as

$$X_p^{new} \sim \mathcal{N}(M_p, \sigma_p^2 C_p) \tag{22}$$

$$\sim M_p + \sigma_p \times \mathcal{N}(0, C_p) \tag{23}$$

After generating new solutions across the entire population, then the solutions are sorted in descending order of fitness values. The previous mean values M are saved as M^{pr} in order to be used later when updating the other state variables. A new set of mean values M^{new} are computed for the next round of iterations as:

$$M^{new} = \sum_{p=1}^{\mu} w_p X_{p:P}^{sort} \tag{24}$$

$$= M^{pr} + \sum_{p=1}^{\mu} w_p (X_{p:P}^{sort} - M^{pr}) \tag{25}$$

where X^{sort} is the sorted population of solutions based on the fitness values of each solution, the positive (recombination) weights $w_1 \geq w_2 \geq \cdots \geq w_\mu > 0$ sum to one, and M^{pr} is the previous mean value per solution p. Typically, $\mu \leq P/2$ and the weights are chosen such that $\mu_w := 1/\sum_{p=1}^{\mu} w_p^2 \approx P/4$. The only feedback used from the fitness function is an ordering of the sampled candidate solutions due to the indices $p : P$. However, for the sake of brevity, the updates for the evolution paths p_σ, p_c, and the covariance matrix C and step-size σ can be found in [33].

11) WHALE OPTIMIZATION ALGORITHM

The whale optimization algorithm (WOA) is a relatively newer optimization algorithm proposed in [34] that mimics the natural hunting mechanism of humpback whales. Humpback whales can detect the presence of prey and encircle it. Because the optimum design's position in the search space is unknown at the outset, the WOA algorithm for MIE presented in Algorithm 8 assumes that the current best candidate solution is the target prey or is near to it. When the best search agent is determined, the other search agents will attempt to update their locations in relation to the best search agent. The following equations illustrate this behavior:

$$D = |C \bullet X_p^{pr} - X^{best}| \tag{26}$$

$$X_p^{new} = X_p^{pr} - A \bullet D \tag{27}$$

where A and C are coefficient vectors, X_p^{pr} is the position vector of the prey, and X^{best} indicates the position vector of a whale (i.e the best solution thus far), and "$\bullet$" represents the dot product. The vectors A and C are calculated as follows:

$$A = 2a \bullet r_1 - a \tag{28}$$

$$C = 2r_2 \tag{29}$$

where components of vector a are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in [0,1]. The WOA algorithm begins with a collection of random solutions. In each iteration, search agents update their locations with relation to either a randomly selected search agent or the best solution achieved thus far. In order to facilitate exploration and exploitation, the a parameter is reduced from 2 to 0. The optimal solution is chosen when $A < 1$ for updating the position of the search agents using (27), whereas when $A > 1$, a search agent X_{rand} is picked randomly from the population of solutions, and the solution is updated using (31).

Algorithm 8 The Whale Optimization Algorithm for MIE

Require: See Table 1

*****Initialization phase*****

$FCR \leftarrow 0$ ***Initialize FCR counter***

$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***

*****Main loop begins here*****

while $FCR < FCR_{max}$ **do**

  Randomly initialize P_a

  for $p = 1$ to P **do**

    if $P_a < 0.5$ **then**

      if $|A| < 1$ **then**

        Update the position of current search agent to obtain X_p^{new} using (27)

      else if $|A| \geq 1$ **then**

        Select a random search agent (X_{rand})

        Update the position of current search agent to obtain X_p^{new} using (31)

      end if

    else if $P_a \geq 0.5$ **then**

      Update the position of the current search agent to obtain X_p^{new} using (32)

    end if

    Perform image transformation using (2) based on X_p^{new}

    Compute fitness $\mathcal{E}_p$ of transformed image using (9)

    $FCR++$ ***Increment FCR counter***

    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$

    Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$

    Terminate process if $FCR == FCR_{max}$

  end for

  Update a, A, C

end while

Save best value and terminate process if $FCR == FCR_{max}$

Post process and visualize the results

$$D = |C \bullet X_{rand} - X^{best}| \tag{30}$$

$$X_p^{new} = X_{rand} - A \bullet D \tag{31}$$

In the case where $P_a \geq 0.5$, the solution is updated using

$$X_p^{new} = D' e^{bt} cos(2\pi t) + X^{best} \tag{32}$$

where $D' = |X^{best} - X_p^{pr}|$ and it indicates the distance of the p^{th} whale to the prey (best solution obtained so far), b is a constant (often $b = 1$) for defining the shape of the logarithmic spiral, and t is a random number in [−1,1]. Finally, the WOA algorithm is ended when the termination requirement is met i.e. $FCR == FCR_{max}$. Further details regarding the WOA can be found in [34].

12) GREY WOLF OPTIMIZATION ALGORITHM

The GWO algorithm proposed in [35] simulates the natural leadership structure and hunting mechanism of grey wolves. For replicating the leadership structure, four sorts of grey wolves are used: alpha, beta, delta, and omega. In addition, three key hunting processes are incorporated to optimize performance: seeking for prey, surrounding (i.e. encircling) prey, and attacking prey. When building GWO, the fittest solution is designated as the alpha (X_α) to quantitatively describe the social hierarchy of wolves. Similarly, the second X_β and third X_δ best solutions are designated as beta and delta, respectively. The remaining possible solutions are all considered to be omega (X_ω). The GWO algorithm's hunting (optimization) is directed by X_α, X_β, and X_δ.

Grey wolves, as previously said, encircle victims during the hunt. The following equations are presented to quantitatively model their encircling behavior:

$$D = |C \bullet X_p^{pr} - X^{best}| \tag{33}$$

$$X_p^{new} = X_p^{pr} - A \bullet D \tag{34}$$

where A and C are the coefficient vectors, X_p^{pr} is the position vector of the prey, and X^{best} indicates the position vector of a wolf (i.e the best solution thus far). The vectors A and C are calculated as follows:

$$A = 2a \bullet r_1 - a \tag{35}$$

$$C = 2r_2 \tag{36}$$

where components of vector a are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in [0,1]. The value of a can be decreased as follows:

$$a = 2 - \frac{2 * FCR}{FCR_{max}} \tag{37}$$

The grey wolves have the capacity to locate and surround prey, thus, in the hunting phase, the alpha generally leads the hunt. The beta and delta may also join in hunting on occasion. However, in an abstract search space, we have no notion where the optimal solution is (i.e. prey). To mathematically mimic grey wolf hunting behavior, it is assumed that the alpha (best candidate solution), beta, and delta have superior knowledge of prospective prey locations. As a result, the first three best solutions acquired so far are kept and the other search agents (i.e. the omegas) are required to update their locations in accordance with the best search agent's position. In this regard, the following formulae are used:

$$D_\alpha = |C_1.X_\alpha - X_p^{pr}| \tag{38}$$

$$D_\beta = |C_2.X_\beta - X_p^{pr}| \tag{39}$$

$$D_\delta = |C_3.X_\delta - X_p^{pr}| \tag{40}$$

$$X_1 = X_\alpha - A_1.(D_\alpha) \tag{41}$$

$$X_2 = X_\beta - A_2.(D_\beta) \tag{42}$$

$$X_3 = X_\delta - A_3.(D_\delta) \tag{43}$$

$$X_p^{new} = \frac{X_1 + X_2 + X_3}{3} \tag{44}$$

In the attacking prey phase, when the target (i.e. prey) stops moving, the grey wolves conclude the hunt by attacking it. To mathematically depict approaching the prey, the value of a is reduced. It is worth noting that the fluctuation range of A is similarly reduced by a. In other words, A is a random value in the interval $[-2a, 2a]$ where a is decreased from 2 to 0 during the duration of repetitions (i.e. iterations of the algorithm). When the random values of A are selected in $[-1, 1]$, the future position of a search agent can be anywhere between its present position and the position of the prey.

To summarize the algorithmic process as presented in Algorithm 9, the search process begins with the GWO algorithm producing a random population of grey wolves (potential solutions). Alpha, beta, and delta wolves determine the likely position of the prey during the duration of iterations. Each potential solution changes its position with relation to the prey. The value of a is reduced from 2 to 0 to emphasize exploration and exploitation, respectively. Candidate solutions tend to diverge from the prey when $A > 1$ and converge towards the prey when $A < 1$. Finally, the GWO algorithm is ended when the termination criteria is satisfied (i.e $FCR == FCR_{max}$).

Algorithm 9 The Grey Wolf Algorithm for MIE

Require: See Table 1
*****Initialization phase*****
$FCR \leftarrow 0$ ***Initialize FCR counter***
$\mathcal{E}_{max} \leftarrow -\inf$ ***Initialize best fitness value***
Initialize $X_\alpha = -\inf, X_\beta = -\inf, X_\sigma = -\inf$
Generate initial random solutions X_p^{new} for $p = 1, 2, \ldots, P$
*****Main loop begins here*****
while $FCR < FCR_{max}$ **do**
  for $p = 1$ to P **do**
    Perform image transformation based on X_p^{new} using (2)
    Compute fitness $\mathcal{E}_p$ of transformed image using (9)
    $FCR++$ ***Increment FCR counter***
    If $\mathcal{E}_p \geq \mathcal{E}_{max}$, then $\mathcal{E}_{max} = \mathcal{E}_p$, else keep $\mathcal{E}_{max}$
    Save best solutions (i.e. $X^{best} = X_p^{new}$) if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Terminate process if $FCR == FCR_{max}$
    Update $X_\alpha = X_p^{new}$ if $\mathcal{E}_p \geq \mathcal{E}_{max}$
    Update $X_\beta = X_p^{new}$ if $\mathcal{E}_p < \mathcal{E}_{max}$ and $\mathcal{E}_p \geq X_\beta$
    Update $X_\delta = X_p^{new}$ if $\mathcal{E}_p < \mathcal{E}_{max}$ and $\mathcal{E}_p < X_\beta$ and $\mathcal{E}_p > X_\delta$
  end for
  Update a using (37)
  Update A using (35)
  Update C using (36)
  Compute X_1, X_2, and X_3 using (41),(42), and (43), respectively.
  Update new solution X_p^{new} using (44)
end while
Save best value and terminate process if $FCR == FCR_{max}$
Post process and visualize the results

D. DATASET

Medical images selected from the MedPix database were used in our study. The MedPix database is a free open online database containing medical images, clinical topics, teaching cases, and textual metadata. It consists of about 12,000 patient case scenarios, 9000 topics, and about 59,000 images. The images within the database are grouped into categories such as the brain and neuro, cardiovascular, chest, pulmonary, eye and orbit, and abdomen [36]. In this study, we have randomly selected images as representative samples from across the human body, which are classed under three use cases described as follows:

1) Use case 1: This image comprises an oral contrast-enhanced axial computed tomographic (CT) scan of a distended appendix as shown in Figure 1(a). The appendix can be seen in the area identified by the blue dotted arrow in Figure 1(a), which depicts a distended appendix having thickened enhancing walls and peri-appendiceal inflammatory changes with fat stranding. This image (of Figure 1(a)) was obtained freely from the MedPix database and can be searched for and downloaded using the code name "synpic28644".
2) Use case 2: This image presents the CT image of a brain presenting the case of an arteriovenous malformation of the right occipital lobe. The original image is shown in Figure 3(a) with the region of interest enclosed by the dotted yellow box wherein the vascular mass can be seen. It is noted that this image (of Figure 3(a)) can be downloaded freely from the MedPix database using the code name "synpic51882".
3) Use case 3: This use case presents a lateral digital subtraction angiography view of the external carotid artery, showing the filling of the right external carotid artery branches obtained after the injection of some contrast medium. The original image from this process is shown in Figure 5(a), which can be searched for and downloaded from the MedPix database using the code name "synpic15935".

E. PERFORMANCE METRICS
In order to adequately assess the effectiveness of any MIE technique, it is essential to define the quantitative measures used in the evaluation process. To this effect, the metrics considered include the number of edges, number of pixels in the fore-ground, entropic measure, and the peak signal to noise ratio, which can be obtained as previously stated in (3), (5), (6), and (7), respectively. Also, the absolute mean brightness error (AMBE) ξ value, which is used to measure the brightness level of an enhanced image has been considered as a metric and can be computed as

$$\xi = |\delta(f(i,j)) - \delta(g(i,j))| \tag{45}$$

Equation (45) evaluates the difference in the mean brightness error value between the original and enhanced medical images. Furthermore, the mean brightness of the original and enhanced images can be computed as follows:

$$\delta(f(i,j)) = \frac{1}{HV}\sum_i\sum_j \delta(f(i,j)) \tag{46}$$

$$\delta(g(i,j)) = \frac{1}{HV}\sum_i\sum_j \delta(g(i,j)) \tag{47}$$

where $\delta(f(i,j))$ represents the mean brightness of the original image, and $\delta(g(i,j))$ denotes the mean brightness of the enhanced image.

III. RESULTS AND DISCUSSION
The MOAs considered in our study were evaluated using different medical images selected from different regions of the body. The qualitative and quantitative results obtained are presented and discussed in this section. By qualitative results, we imply the visual assessment of the enhanced images from the viewpoint of a radiologist or any human observer. Such an assessment is based upon the salient differences between the different enhanced output images and the original input image.

In terms of quantitative outcomes, we begin by discussing the fitness evolution trend of each method per input image. The fitness trends are provided in terms of the best, worst, and mean fitness values derived as a function of the FCR from 1000 independent trials (i.e. Monte Carlo experiments). In addition, the values of each measure produced by each method are displayed and analyzed in relation to the corresponding enhanced images. For each approach, we then show the final optimized parameter values that were utilized in the transformation function to generate the various enhanced images. The physical timing performance of each method and a thorough statistical significance tests of the results per use-case are discussed in the last part of this section.

A. USE-CASE 1: IMAGE OF ACUTE APPENDICITIS
We applied each algorithm to use case 1, where the area of interest to a radiologist is depicted by the magnified region in Figure 1(a), which depicts a distended appendix having thickened enhancing walls and periappendiceal inflammatory changes with fat stranding. Following the application of the different algorithms to the image of Figure 1(a), we present the enhanced output images per MOA in Figures 1(b) - 1(j), respectively.

1) QUALITATIVE ANALYSIS
It can be seen that the enhanced images in Figures 1(b) - 1(j) demonstrate an improved contrast and brightness level as against the original image. Visually, the original image presents a blurrier and less detailed CT scan. Such improvement in the contrast level in Figures 1(b) - 1(j) implies that the boundaries of the different organs were better delineated in the enhanced images than in the original image.

Specifically, the zoomed area of each image shows that the thickened walls of the distended appendix were better enhanced with an improved intensity and brightness around the boundaries of the wall protruding to the caecum. This ensures that the radiologist is better informed about the degree of distension around the appendix as compared to using only the original image. Furthermore, the oral contrast liquid used during the diagnostic procedure as seen by the whitish dispersion in the original image are more vivid in the enhanced images. Additionally, the details of the fat stranding in the appendix region are better pronounced in the enhanced images than in the original image, thus allowing a radiologist to provide an improved diagnosis.

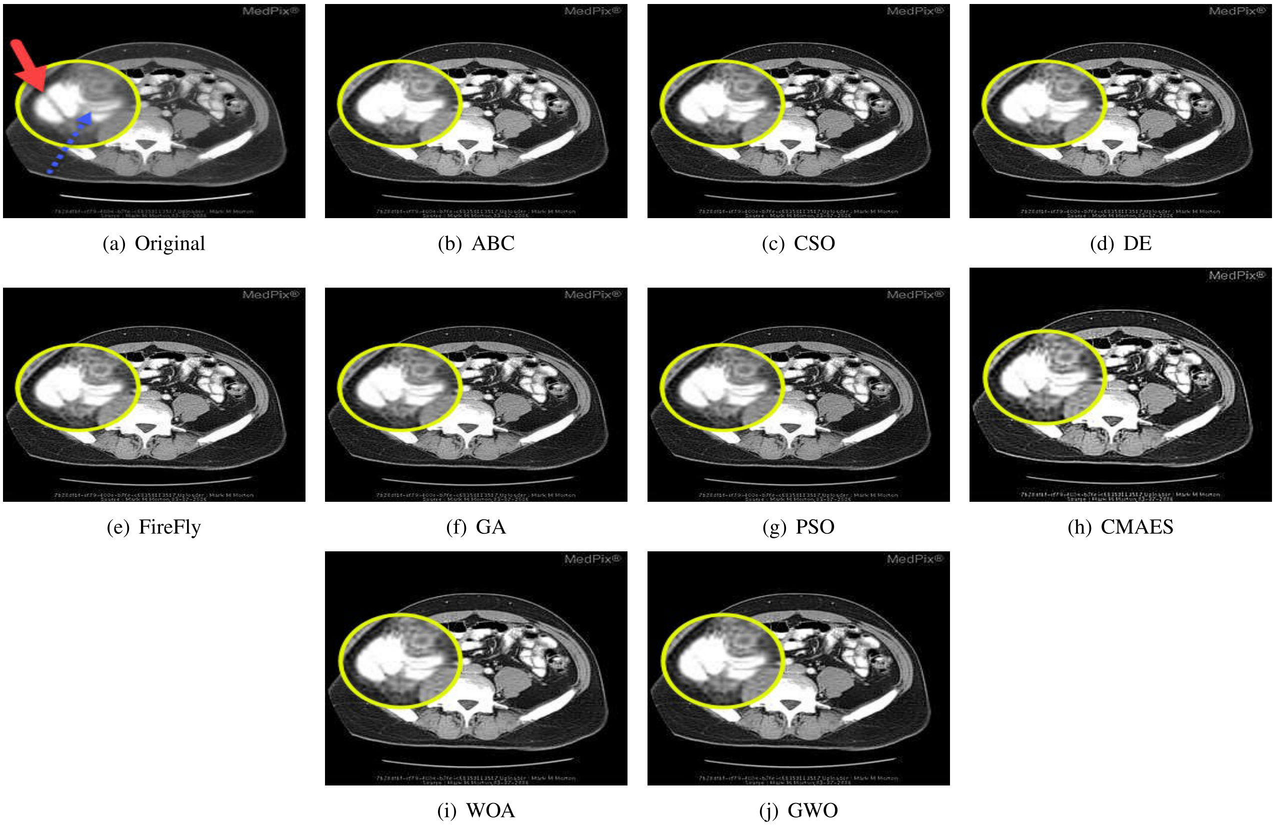


FIGURE 1. Use case 1: Original and enhanced CT images of an acute appendicitis.

Looking closely at the different enhanced images, it can be seen that the ABC algorithm produced an overtly brighter image than the other methods, which caused the region of interest to be more obscured than expected. At first glance, it may be difficult to visually determine the best enhanced image, however, by enlarging the images and visually parsing through them in a slideshow manner, we came to the following opined conclusions:

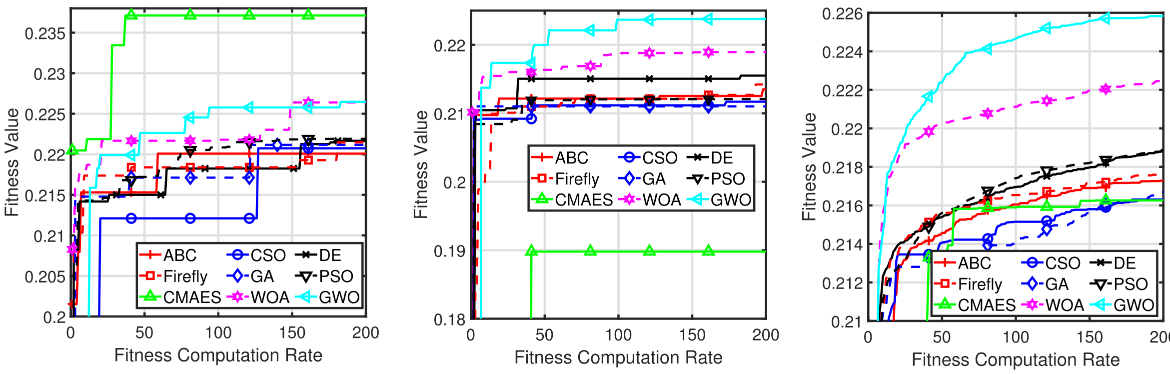
1) The ABC algorithm produced a slightly blurrier image than the other algorithms, with a generally lesser feel of texture and provision of details. Thus, the ABC algorithm performed least in this use case.
2) The enhanced images produced by the DE and Firefly algorithm were visually indistinguishable, thus implying that both algorithms achieved similar performance levels.
3) It is generally difficult to distinguish visually between the performances of the different MOAs as it concerns the enhancement of the present use case image.

Following the above observations, we shall next examine and analyse the quantitative results per MOA towards arriving at some substantial conclusions.

2) QUANTITATIVE ANALYSIS

a: FITNESS EVOLUTION PERFORMANCE

Figure 2 presents the fitness evolution graphs of the different MOAs as a function of the FCR. There are three different results presented in this regard, including the best, worst, and mean fitness evolution performance results shown in Figures 2(a) - 2(c), respectively. We note that Monte Carlo (MC) simulation consisting of 1000 different MC trials were used to arrive at the different results of Figure 2. In particular, Figure 2(a) represents the best fitness evolution results obtained per algorithm from within a set of 1000 different MC trials. It is worth noting that the different enhanced images presented in Figure 1 are the outcomes of this best fitness value graphed in Figure 2(a). On the other hand, the output images produced by the worst and mean fitness performance processes of each algorithm are not presented here since the evolution trends of Figures 2(b) and 2(c) are sufficient to describe their different performance levels. The worst performance graph of Figure 2(b) corresponds to the lowest fitness values obtained after conducting 1000 MC trials per algorithm. Whereas, we computed the mean results of Figure 2(c) by averaging the fitness evolution graphs over 1000 MC trials.



(a) Best (out of 1000 trials) (b) Worst (out of 1000 trials) (c) Mean (over 1000 trials)

FIGURE 2. Fitness value evolution for use case 1.

Generally, it can be seen from Figure 2 that the performance rate of each algorithm improves with an increase in the FCR. Such an increase in the performance of each algorithm is expected since only better solutions (i.e better enhanced images) are retained whenever the fitness function is computed. In particular, it can be seen in Figure 2(a) that after computing the fitness function 200 times, the different algorithms can be rated as follows: the CMAES algorithm achieved the highest fitness value, followed closely by the GWO and WOA, then the PSO followed by the DE, Firefly, GA, CSO, and the ABC algorithm. We can also see in Figure 2(a) that there exists a small difference (≈0.015) in the fitness values of the seemingly best algorithm (i.e the CMAES) and the least performing algorithm (i.e the ABC). Such close fitness values typically correspond to an almost indistinguishable visual difference between the enhanced images of the different algorithms, thus supporting the qualitative conclusions drawn in section III-A1.

Based on an average of 1000 different MC trials, Figure 2(c) shows that the GWO algorithm provides the best performance followed by the WOA algorithm. Essentially, the figure reveals that there exists only a marginal difference (< 0.008) in the fitness values of the best and least performing algorithm. This thus leads to the conclusion that there may be no significant difference in using one MOA over another in this use case.

b: OTHER PERFORMANCE METRICS

Although the fitness evolution performance graphs of Figure 2 have provided an overall assessment of the different algorithms, nevertheless, it is worthwhile to further analyse other specific outcomes regarding the quality of the enhanced images. Consequently, a few notable metrics were measured and presented in Table 2. It can be seen in Table 2 that the CMAES algorithm achieved the best results (see bold values in Table 2) in terms of its detailed variance and number of edges, whereas the GA and WOA achieved the best results in terms of the peak SNR and entropy, respectively.

TABLE 2. Use case 1: Image enhancement performance metrics.

	Detailed Variance	Number of Edges	Peak SNR	Entropy	AMBE
ABC	2342	4270	16.9299	4.8367	**0.0235**
CSO	2502	4613	18.5260	4.8423	0.0068
DE	2835	4772	17.9725	4.8486	0.0124
Firefly	2820	4767	17.9864	4.8441	0.0107
GA	2527	4634	**18.7130**	4.8376	0.0034
PSO	2700	4721	18.6540	4.8337	0.0010
CMAES	**6554**	**5794**	15.7351	4.9157	0.0206
WOA	3105	4736	17.9981	**4.9698**	0.01331
GWO	3001	4701	18.4936	4.9561	0.0057

Interestingly, the AMBE result of the ABC algorithm (i.e. AMBE = 0.0235) in Table 2 indicates that the ABC algorithm contributed the largest brightness value compared to the other algorithms. Such a high AMBE value corroborates the visual conclusions drawn in section III-A1, which states that the ABC algorithm produced an overtly brighter enhanced image. This high AMBE value ultimately caused the ABC algorithm to over brighten the region of interest as compared to the other algorithms (see Figure 1(b)), thus diminishing its overall visual performance.

Again, we cannot over emphasize the fact that there was really no significant difference in the individual metric values achieved by the different algorithms. For example, there was no significant difference in the supposed better performance of the DE algorithm over the Firefly algorithm (see close values of both algorithms in Table 2). These close values of both algorithms typically corroborates our visual conclusions drawn in section III-A1, which affirms that there was no significant difference between the DE and Firefly algorithms.

Additionally, our conclusions about the insignificant differences in the performance levels of the different algorithms can be further emphasized based on the number-of-edges metric (see third column in Table 2). In this case, although the DE and Firefly algorithms may have enhanced more edges than the ABC algorithm, it should nevertheless be noted that these edges could as well have been enhanced in the wrong regions of interest. For example, the dividing line seen clearly

TABLE 3. Use case 1: Parameter values.

	a	b	c	k
ABC	2.2904	0.3359	0.6866	3.4870
CSO	2.0000	0.3007	0.7985	3.7621
DE	2.0171	0.3000	0.7989	4.0000
Firefly	2.0490	0.3000	0.7988	3.9989
GA	2.0000	0.3061	0.8085	3.8941
PSO	2.0287	0.3000	0.8186	3.9997
CMAES	2.2752	0.1911	0.8429	3.9542
WOA	2.0000	0.3000	0.7994	4.0000
GWO	2.0000	0.3000	0.8131	4.0000

in the caecum region of Figure 1(a) (see red line) becomes difficult to discern in all the enhanced images due to excessive brightening of the pixel edges. Thus, it is possible that introducing better results in one dimension of an enhanced image does not necessarily imply a better overall enhanced image. Consequently, we suggest that one single MOA may not necessarily provide the best enhanced image with regards to all constituting metrics of assessment.

We present in Table 3 the final optimized parameter values of the transformation function as estimated by the different MOAs. These values are reported here so that the final enhanced images shown in Figures 1(b) - 1(g) can be independently reproduced for validation purposes. These values are also presented as they provide some plausible explanation for the different outcomes obtained per algorithm. For example, we can explain why the ABC algorithm produced an overtly bright image since it converged to a larger a parameter value, which controls the brightness level of the enhancement process.

B. USE-CASE 2: IMAGE OF RIGHT OCCIPITAL LOBE

Each algorithm was applied to use case 2 and the corresponding enhanced images outputted per algorithm are presented in Figures 3(b) - 3(j).

1) QUALITATIVE ANALYSIS

We observe from Figures 3(b) - 3(j) that the output CT images are better enhanced compared to the original image in Figure 3(a). Specifically, the finer details within the vascular mass bounded by the yellow-dotted box are well delineated in the enhanced images. Thus, this reveals to the radiologist a more vivid display of the degree of growth of the vascular mass as compared to using the original image for diagnostic purposes.

In terms of the performance of the different MOAs, it can be seen from closer examination of the enhanced images that the GA algorithm introduced the largest degree of brightness. Whereas, the CMAES algorithm produced the darkest enhanced image. Such a dispersion in the brightness level of the different enhanced images did not affect the region of interest, neither was any other region of the CT image affected. Thus, such negligible differences in the contrast and brightness levels of Figures 3(b) - 3(g) makes it qualitatively difficult to determine the best enhanced image with regards to the region of interest.

2) QUANTITATIVE ANALYSIS

Here, an attempt is made to quantitatively appraise the performance levels of the different MOAs based on the following specific metrics:

a: FITNESS EVOLUTION PERFORMANCE

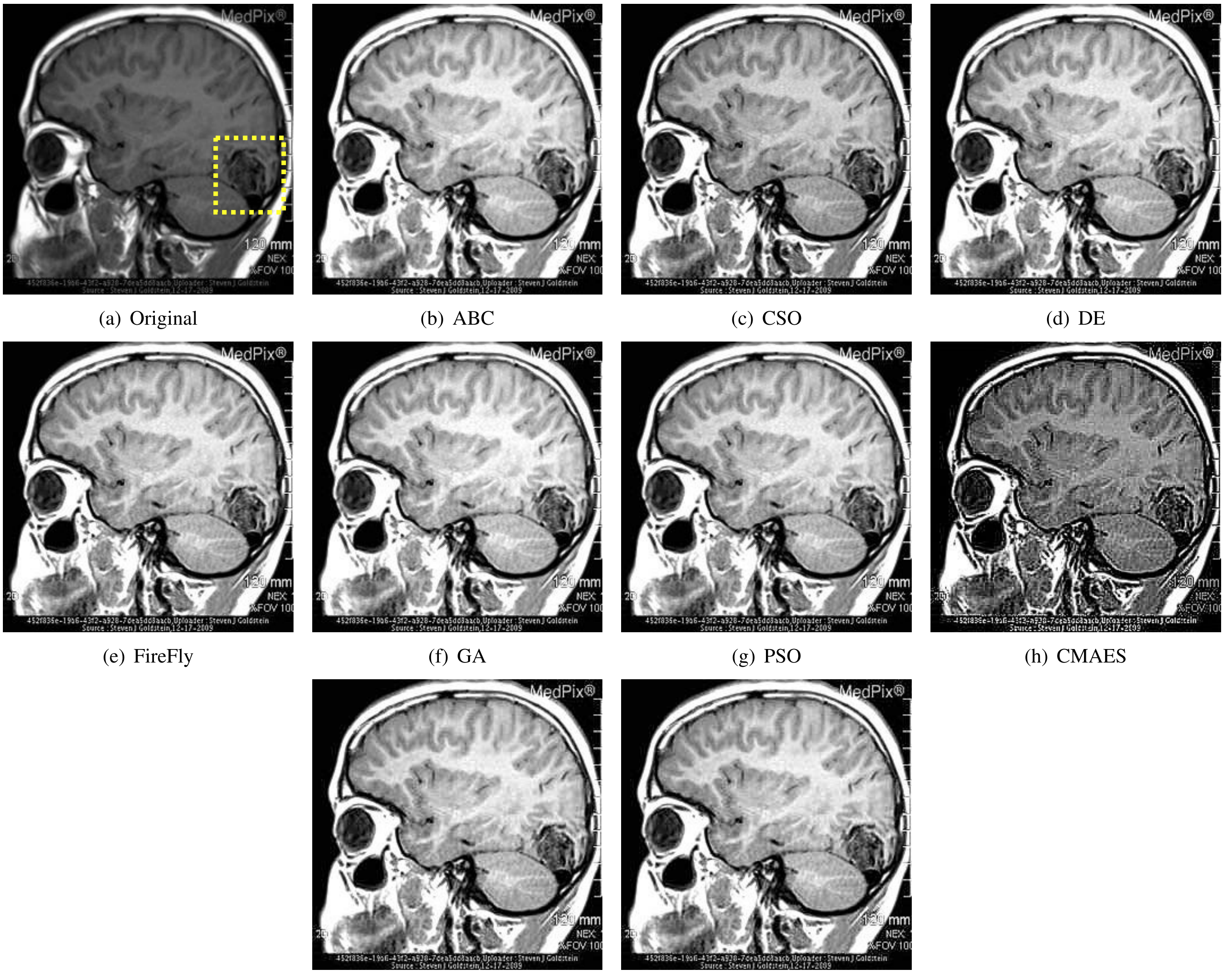
First, it is worth mentioning that the enhanced images of Figures 3(b) - 3(j) correspond to the outcome of the best fitness evolution per MOA out of 1000 MC trials graphed in Figure 4(a). From Figure 4(a), it is noted that the different MOAs converged to very close fitness values after computing the fitness function 200 times. Such close fitness values (<0.045) between the best (i.e the CMAES algorithm) and the least (i.e the CSO algorithm) affirms why it was qualitatively difficult to determine the best enhanced image in Section III-B1. From the best performance graph of Figure 4(a), it is noted that the CMAES algorithm suffices as the best performing algorithm towards the end range of the FCR. This implies that if we had shortened the FCR (probably to obtain output images faster), the WOA algorithm instead of the CMAES algorithm would have yielded the best enhanced image within such a simulation period.

Nevertheless, we note that such a high fitness value of the CMAES algorithm may not always be guaranteed in all trials, particularly as noted in the worst performance curve of Figure 4(b). Note that the curve of Figure 4(b) corresponds to the worst output result obtained out of 1000 different MC trials. Thus, in this single trial, it can be seen that the WOA algorithm yielded the highest fitness value and closely followed by the PSO, DE, and GA algorithms. Since such extreme cases (i.e best and worst case of Figures 4(a) and 4(b), respectively) may not provide the best assessment across 1000 different MC trials, thus, we provide the mean performance curves computed over 1000 MC trials as shown in Figure 4(c). Such a reliable fitness evolution statistics of Figure 4(c) reveals that the GWO algorithm generally yielded the highest fitness value (i.e best enhanced image), whereas the CSO and CMAES algorithms sufficed as the least performers.

b: OTHER PERFORMANCE METRICS

We further examined the enhanced images of Figures 3(b) - 3(j) based on the performance metrics that constitute the overall fitness value. These results are documented in Table 4. Interestingly, the GA algorithm yielded the largest AMBE value, which is the metric that quantifies the degree of brightness of an enhanced image. Such a large AMBE value of the GA algorithm accurately affirms the conclusion drawn in Section III-B1, which states that the GA algorithm yielded the brightest enhanced image. Similarly, we note that the CMAES algorithm produced the least AMBE value, which affirms that it yielded the darkest enhanced image (see Figure 3(h)).

Although the CMAES algorithm may have yielded the largest value for the detailed variance, number of edges,

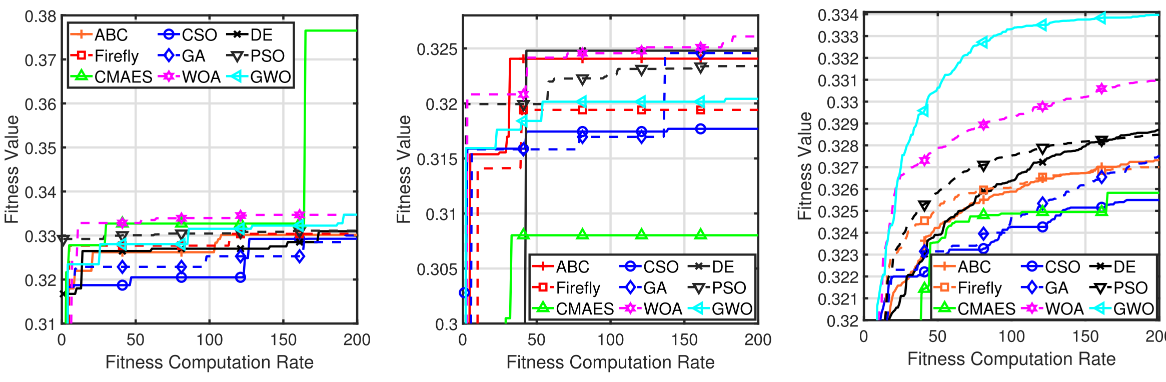


(a) Original (b) ABC (c) CSO (d) DE

(e) FireFly (f) GA (g) PSO (h) CMAES

(i) WOA (j) GWO

FIGURE 3. Use case 2: Original and enhanced CT images of the right occipital lobe of a brain.



(a) Best (out of 1000 trials) (b) Worst (out of 1000 trials) (c) Mean (over 1000 trials)

FIGURE 4. Fitness value evolution for use case 2.

and the peak SNR, nevertheless, going by the fitness value, it was not ranked as the best performing algorithm going by the mean fitness graph of Fig 4(c). This is due to the fact that, as a statistical method, such MOAs cannot be evaluated by a single trial of the MIE procedure, and so numerous trials are frequently necessary to pick the optimum

TABLE 4. Use case 2: Performance metrics.

	Detailed Variance	Number of Edges	Peak SNR	Entropy	AMBE
ABC	17760	6277	6.1329	6.8932	0.3365
CSO	18369	6642	7.9720	6.8699	0.2539
DE	19095	6860	6.4912	6.9193	0.3146
Firefly	19063	6852	6.2980	6.9203	0.3182
GA	19292	6892	5.4277	6.8928	**0.3598**
PSO	19200	6879	6.3743	6.9174	0.3217
CMAES	**28674**	**10225**	**8.8040**	6.6268	0.1046
WOA	19591	6923	6.2936	6.9853	0.3260
GWO	19559	6910	6.3527	**6.9876**	0.3231

TABLE 5. Use case 2: Parameter values.

	a	b	c	k
ABC	2.2715	0.3419	0.3357	3.1753
CSO	2.0000	0.3065	0.6201	4.0000
DE	2.0701	0.3000	0.5680	4.0000
Firefly	2.3377	0.3000	0.5472	4.0000
GA	2.4976	0.3000	0.4867	3.8862
PSO	2.0019	0.3000	0.5668	3.9999
CMAES	2.4105	0.0888	0.8952	3.1768
WOA	2.0000	0.3000	0.5640	4.0000
GWO	2.0030	0.3000	0.5663	4.0000

performance. We also noticed that the metric values acquired across the multiple MOAs were not significantly different, which resulted in significantly near fitness scores across all methods. As a result, such near fitness values confirm the difficulties in visually differentiating the best improved image in Figure 3. Finally, Table 4 shows that no one MOA attained the greatest value across all measures, showing that it is nearly impossible to claim any MOA as the top performer across all potential use cases.

The optimized parameters obtained by the different MOAs are reported in Table 5 for the sake of repeatability. We remark that these are the parameter values used to generate the enhanced images shown in Figures 3(b) - 3(j). For instance, we noticed that the GA algorithm's large a value (see Table 5) explains why it produced the brightest image (in Figure 3) when compared to the other methods.

C. USE-CASE 3: IMAGE OF RIGHT COMMON CAROTID ARTERY INJECTION

This use case was processed using the different MOAs and the respective enhanced images by the different algorithms are presented in Figures 5(b) - 5(j).

1) QUALITATIVE ANALYSIS

A radiologist's major concern in analyzing the various images of Figure 5 is to identify whether there is any indication of opacification of the internal carotid artery (ICA) or its branches. Although there was no indication of opacification because the contrast medium can be seen to be clearly disseminated over the ICA, any viewer can perceive that the original image has a blurrier sight than the enhanced images. As a result, several of the arteries in Figure 5(a) are made nearly invisible to the human eye when compared to the enhanced images in Figures 5(b) - 5(j). To that end, there is no doubt that the enhanced images from the various MOAs provide a radiologist with a more detailed perspective than viewing the original image.

It may be challenging to determine the best improved image based just on visual examination. However, when it comes to the worst performance, it is clear that the contrast of Figure 5(b) generated by the ABC algorithm is the blurriest. It displays a lot of weak arterial lines at the image's border, resulting in fewer characteristics for analysis. To summarize, while the original image was definitely enhanced across the multiple MOAs, there is very little to separate visually the performance of the different methods.

2) QUANTITATIVE ANALYSIS

The quantitative performance of the different MOAs are analysed based on the following metrics:

a: FITNESS EVOLUTION PERFORMANCE

Figures 6(a), 6(b), and 6(c) show the findings of the best, worst, and mean fitness values as a function of the FCR per MOA. We examine Figure 6(a) because the enhanced images in Figure 5 are derived from these data after computing the fitness function 200 times. Figure 6(a) shows that remarkably near fitness values were obtained across the different MOAs, justifying the indistinguishable visual performance of the various enhanced images. Nonetheless, the CMAES, DE, and PSO algorithms are shown to have produced the best fitness values, thus resulting in the better looking enhanced images of these algorithms as shown in Figures 5(h), 5(d), and 5(g), respectively.

Figure 6(b) presents the lowest fitness performance of the different algorithms obtained out of 1000 MC trials. At an FCR of 200, we found that the GWO algorithm produced the largest fitness value, followed closely by the WOA, ABC, and Firefly algorithms. This means that determining which MOA would perform best based on a single run of the algorithm is often challenging since randomized results may suffice during the initial iterations of the different algorithms. However, after an average of 1000 MC trials, we can observe in Figure 6(c) that the GWO and PSO algorithms surpassed the DE and other algorithms in terms of their fitness values. Despite these findings, we infer that the fitness values obtained by the algorithms are not significantly different, which explains the difficulty in visually choosing the best improved image.

b: OTHER PERFORMANCE METRICS

Table 6 shows the various performance measures that make up the fitness values. Based on the detailed variance values and the number of edges, an interesting finding is made, revealing that the ABC algorithm yielded the lowest values. These values explain why the ABC algorithm produced the most blurry enhanced image. Again, the ABC algorithm's lower visual performance is supported by its lowest entropy

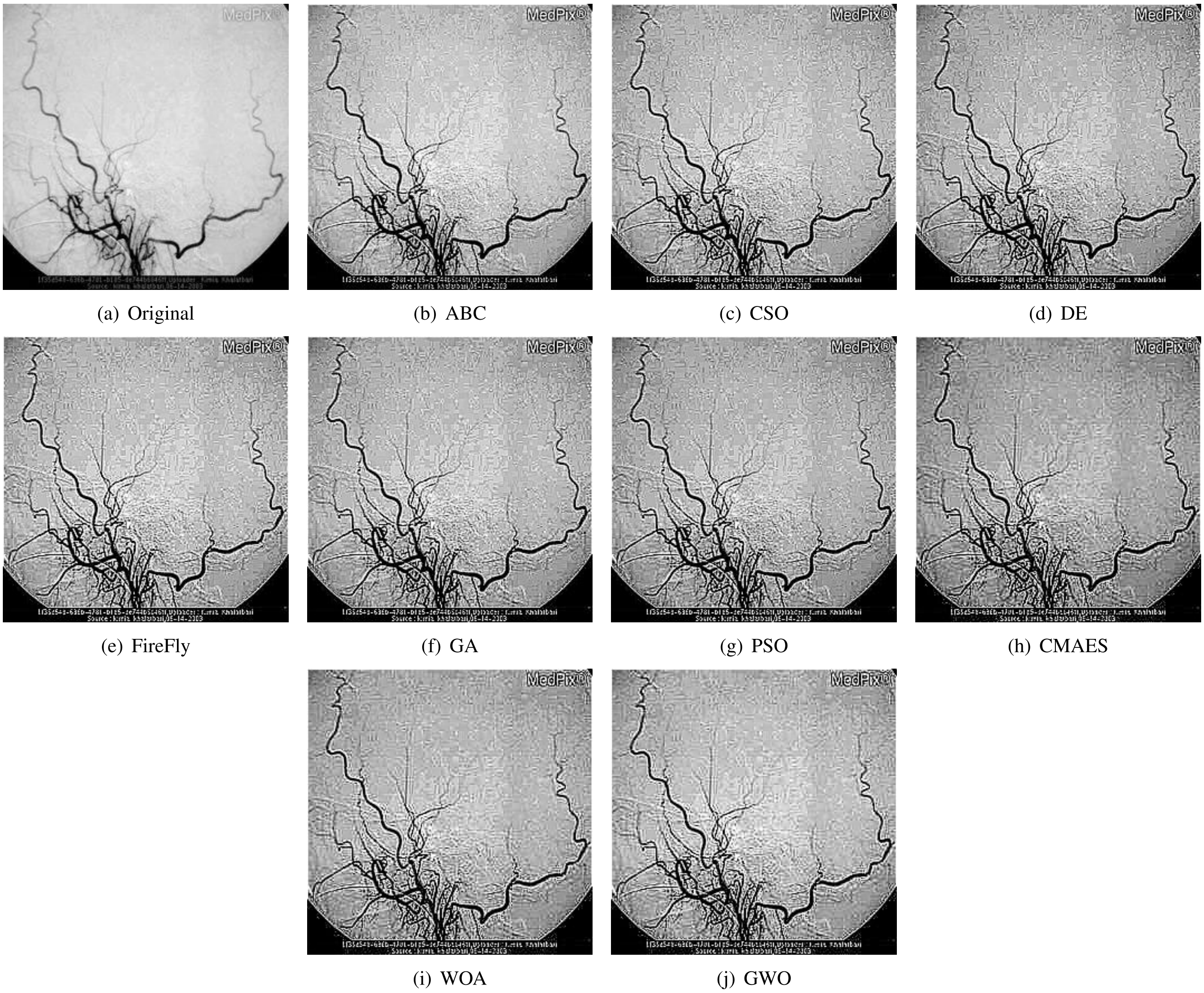


FIGURE 5. Use case 3: Original and enhanced DSA images of the right carotid artery injection of a brain.

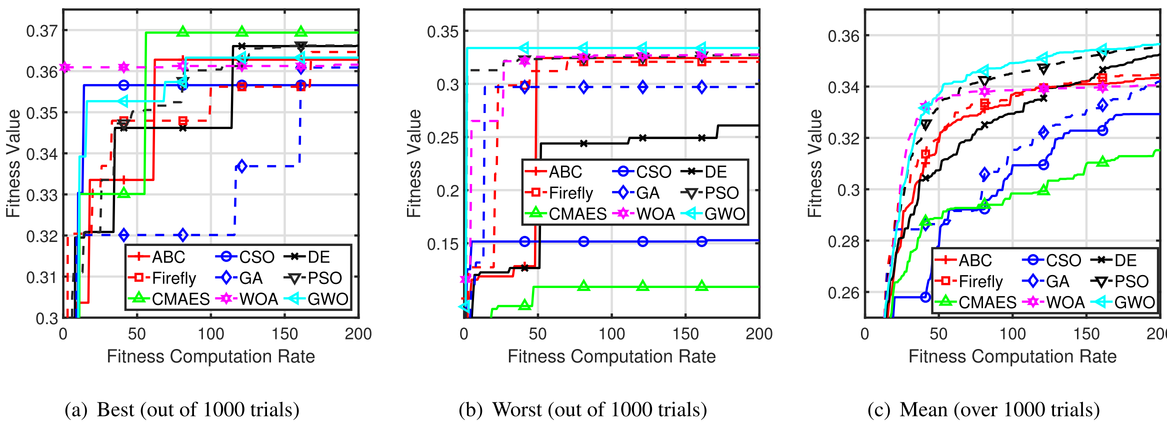


FIGURE 6. Fitness value evolution for use case 3.

value, which explains the absence of adequate information (i.e. blurriness) in its enhanced image. The CMAES, DE, Firefly, and PSO algorithms, on the other hand, provided the highest performance values across all measures, which confirms their somewhat improved visual performance in Figures 5(h), 5(d), 5(e) and 5(h), respectively.

TABLE 6. Use case 3: Performance metrics.

	Detailed Variance	Number of Edges	Peak SNR	Entropy	AMBE
ABC	15681	7737	**13.4275**	6.8089	0.0725
CSO	20595	10234	10.3458	6.8611	0.1001
DE	22270	11179	9.3603	6.9204	0.1306
Firefly	22168	11112	9.7023	6.8668	0.0955
GA	21286	10630	9.9537	6.8943	0.1136
PSO	22264	11174	9.3127	6.9270	0.1356
CMAES	**23106**	**11433**	8.2093	**6.9734**	**0.1838**
WOA	21723	10590	9.3645	6.9106	0.1333
GWO	21976	10748	9.4878	6.9065	0.1067

TABLE 7. Use case 3: Parameter values.

	a	b	c	k
ABC	2.0855	0.3679	0.9843	3.2695
CSO	2.0285	0.3207	0.9950	3.9289
DE	2.3377	0.3000	0.9948	4.0000
Firefly	2.1469	0.3000	0.9929	3.9866
GA	2.2115	0.3000	0.9940	3.8051
PSO	2.3381	0.3001	0.9955	4.0000
CMAES	2.6789	0.2631	0.9978	3.7781
WOA	2.2417	0.3000	0.9970	3.9477
GWO	2.5000	0.3000	0.9891	4.0000

Table 7 presents the optimum parameter values of the transformation function as computed by the different MOAs and utilized to create the enhanced images of Figures 5(b) - 5(j). These parameters are provided so that the various improved images of Figure 5 may be recreated and confirmed independently by any interested researcher. In particular, both the DE and PSO algorithms converged to similar parameter values, which explains why they produced comparable improved images with identical output performance metrics.

D. CPU TIMING PERFORMANCE

The timing performance of the different algorithms were compared and reported per use case in Table 8. For this purpose, all simulations were conducted using a PC running an Intel(R) Core i5-7500 CPU processor @ 3.40 GHz with an installed memory (RAM) of 16GB. To ensure that the timing experiments were relatively valid, the following measures were taken:

1) All algorithms were encoded and simulated in MATLAB 2017a.
2) All use-case images were resized to the same size (i.e. 256 × 256)
3) The population size and FCR were fixed for all MOAs.
4) The start time for measuring the timing performance of each algorithm was taken at the initialization of each algorithm and was ended strictly after the main loop was completed.
5) It was ensured that only the MATLAB software was kept running as the only foreground process during each simulation period in order to ensure that no extra processing time was incurred by the PC.

TABLE 8. CPU processing time (in seconds) across the different use cases.

	Use case 1	Use case 2	Use case 3
ABC	5.0951	5.6703	5.7862
CSO	4.9027	5.5576	6.0604
DE	4.9256	5.6555	5.9871
Firefly	4.9647	5.6744	5.8872
GA	4.8783	5.5673	5.3434
PSO	4.9709	5.7516	5.9027
CMAES	4.8492	5.7127	5.7383
WOA	4.2577	**5.0048**	**5.0924**
GWO	**4.2176**	5.0910	5.138

6) Finally, the timing results reported in Table 8 were obtained over an average of 1000 MC trials per use-case.

In terms of the run-time of each MOA, we observed that the GA algorithm converged the quickest in the first and third use case, whereas the CSO algorithm came quickest in the second use case. Despite these algorithms being the fastest, we note that there was less than an average difference of 0.1 - 0.3 seconds in the computing times of the different algorithms. This implies that when the FCR is kept fixed and equal for all MOAs, then there exists little or no significant difference in their respective real time computational speeds.

IV. STATISTICAL SIGNIFICANCE TESTS BASED ON THE USE-CASES

We performed statistical significance tests on the mean outcomes over 1000 independent trials. Instead of merely using the best results, we used the mean outcomes, which ensures that a more rigorous analysis is achieved. To this end, we ran the Brown-Forsythe and Welch ANOVA tests on the mean outcomes per use-case, as well as a posthoc comparison of the various MOAs using Games-Howell's multiple comparison tests. Our collected findings are therefore provided in Tables 10 - 13 for the different use-cases. The Games-Howell's test was chosen because it is appropriate for comparing two or more sample populations and, unlike the Tukey's test, it does not presuppose homogeneity of variances or equal sample sizes, which better suits our test results.

The first column of Tables 10 - 13 mentions the paired algorithms compared with, while the second column shows the difference in the means of both algorithms. A positive or negative mean difference in the second column indicates that the mean of the first listed algorithm is bigger or smaller than the mean of the second algorithm, respectively. The confidence interval of the mean difference between the matched algorithms is shown in the third column. The fourth column indicates whether the estimated p-value is larger or less than the threshold p-value (i.e. 0.05), and the last column offers the statistical significance conclusion based on the threshold p-value. Table 9 shows the symbols used to indicate the range of p-values and their related meanings.

Table 10 for use-case 1 shows that, with the exception of the CMAES, WOA, and GWO algorithms, there was no

TABLE 9. The p-value range, corresponding symbol and interpretation.

Symbol	Range	Interpretation
ns	$P > 0.05$	not significant
*	$0.01 < P \leq 0.05$	weakly significant
**	$0.001 < P \leq 0.01$	significant
***	$0.0001 < P \leq 0.001$	very significant
****	$P \leq 0.0001$	extremely significant

TABLE 10. Use-case 1: Statistical significance test compared between the different MOAs.

Games-Howell's multiple comparisons test	Mean Diff.	95.00% CI of diff.	Below threshold?	Summary
ABC vs. CSO	0.001187	-0.002448 to 0.004821	No	ns
ABC vs. DE	-0.001401	-0.004770 to 0.001968	No	ns
ABC vs. Firefly	-0.0008345	-0.004070 to 0.002401	No	ns
ABC vs. GA	0.001215	-0.002401 to 0.004831	No	ns
ABC vs. PSO	-0.00132	-0.004705 to 0.002065	No	ns
ABC vs. CMAES	0.004701	2.503e-005 to 0.009376	Yes	*
ABC vs. WOA	-0.005211	-0.008779 to -0.001643	Yes	***
ABC vs. GWO	-0.00836	-0.01169 to -0.005033	Yes	****
CSO vs. DE	-0.002587	-0.006086 to 0.0009112	No	ns
CSO vs. Firefly	-0.002021	-0.005391 to 0.001349	No	ns
CSO vs. GA	0.0000285	-0.003708 to 0.003765	No	ns
CSO vs. PSO	-0.002507	-0.006021 to 0.001008	No	ns
CSO vs. CMAES	0.003514	-0.001254 to 0.008282	No	ns
CSO vs. WOA	-0.006398	-0.01009 to -0.002707	Yes	****
CSO vs. GWO	-0.009547	-0.01301 to -0.006088	Yes	****
DE vs. Firefly	0.0005665	-0.002515 to 0.003648	No	ns
DE vs. GA	0.002616	-0.0008629 to 0.006095	No	ns
DE vs. PSO	0.000081	-0.003157 to 0.003319	No	ns
DE vs. CMAES	0.006101	0.001530 to 0.01067	Yes	**
DE vs. WOA	-0.00381	-0.007240 to -0.0003802	Yes	*
DE vs. GWO	-0.00696	-0.01014 to -0.003781	Yes	****
Firefly vs. GA	0.00205	-0.001300 to 0.005399	No	ns
Firefly vs. PSO	-0.0004855	-0.003584 to 0.002613	No	ns
Firefly vs. CMAES	0.005535	0.001060 to 0.01001	Yes	**
Firefly vs. WOA	-0.004376	-0.007675 to -0.001078	Yes	**
Firefly vs. GWO	-0.007526	-0.01056 to -0.004490	Yes	****
GA vs. PSO	-0.002535	-0.006029 to 0.0009593	No	ns
GA vs. CMAES	0.003485	-0.001269 to 0.008240	No	ns
GA vs. WOA	-0.006426	-0.01010 to -0.002754	Yes	****
GA vs. GWO	-0.009576	-0.01301 to -0.006137	Yes	****
PSO vs. CMAES	0.00602	0.001437 to 0.01060	Yes	**
PSO vs. WOA	-0.003891	-0.007336 to -0.0004456	Yes	*
PSO vs. GWO	-0.007041	-0.01024 to -0.003845	Yes	****
CMAES vs. WOA	-0.009911	-0.01463 to -0.005193	Yes	****
CMAES vs. GWO	-0.01306	-0.01760 to -0.008519	Yes	****
WOA vs. GWO	-0.00315	-0.006539 to 0.0002396	No	ns

statistical significance between the different MOAs. In this situation, the GWO method is acknowledged to have obtained the best mean outcomes and is thus thought to be significantly distinct from all other MOAs. Nonetheless, no significant difference was found between the GWO and WOA, indicating that both algorithms function equally. As a result, only the GWO and WOA algorithms produced a highly significant difference in empirical performance when compared to the other approaches.

Table 11 shows that, with an exception of the GWO, WOA, and CMAES algorithms, there was no significant difference between the other MOAs investigated in our study for use-case 2. We notice that such a difference in mean values might be either positive or negative; hence, the GWO algorithm obtained positive differences when compared to the others, suggesting that it performed better than the others.

The findings for use-case 3 shown in Table 12 indicate that there was no significant statistical difference between the other algorithms, save for the GWO and WOA algorithms. Nevertheless, it should be noted that these results

TABLE 11. Use-case 2: Statistical significance test compared between the different MOAs.

Games-Howell's multiple comparisons test	Mean Diff.	95.00% CI of diff.	Below threshold?	Summary
ABC vs. CSO	0.000369	-0.004655 to 0.005393	No	ns
ABC vs. DE	-0.00054	-0.006116 to 0.005036	No	ns
ABC vs. Firefly	-0.000464	-0.005964 to 0.005036	No	ns
ABC vs. GA	0.0000735	-0.005040 to 0.005187	No	ns
ABC vs. PSO	-0.001702	-0.007296 to 0.003891	No	ns
ABC vs. CMAES	0.005679	-0.0008162 to 0.01217	No	ns
ABC vs. WOA	-0.004164	-0.009544 to 0.001216	No	ns
ABC vs. GWO	-0.00717	-0.01255 to -0.001792	Yes	**
CSO vs. DE	-0.000909	-0.005718 to 0.003900	No	ns
CSO vs. Firefly	-0.000833	-0.005553 to 0.003887	No	ns
CSO vs. GA	-0.0002955	-0.004555 to 0.003964	No	ns
CSO vs. PSO	-0.002071	-0.006900 to 0.002757	No	ns
CSO vs. CMAES	0.00531	-0.0005445 to 0.01116	No	ns
CSO vs. WOA	-0.004533	-0.009111 to 4.487e-005	No	ns
CSO vs. GWO	-0.007539	-0.01211 to -0.002963	Yes	****
DE vs. Firefly	0.000076	-0.005229 to 0.005381	No	ns
DE vs. GA	0.0006135	-0.004289 to 0.005516	No	ns
DE vs. PSO	-0.001162	-0.006564 to 0.004239	No	ns
DE vs. CMAES	0.006219	-0.0001127 to 0.01255	No	ns
DE vs. WOA	-0.003624	-0.008803 to 0.001555	No	ns
DE vs. GWO	-0.00663	-0.01181 to -0.001452	Yes	**
Firefly vs. GA	0.0005375	-0.004278 to 0.005353	No	ns
Firefly vs. PSO	-0.001238	-0.006561 to 0.004084	No	ns
Firefly vs. CMAES	0.006143	-0.0001225 to 0.01241	No	ns
Firefly vs. WOA	-0.0037	-0.008798 to 0.001398	No	ns
Firefly vs. GWO	-0.006706	-0.01180 to -0.001610	Yes	**
GA vs. PSO	-0.001776	-0.006698 to 0.003146	No	ns
GA vs. CMAES	0.005605	-0.0003256 to 0.01154	No	ns
GA vs. WOA	-0.004237	-0.008914 to 0.0004389	No	ns
GA vs. GWO	-0.007244	-0.01192 to -0.002569	Yes	****
PSO vs. CMAES	0.007381	0.001035 to 0.01373	Yes	**
PSO vs. WOA	-0.002462	-0.007659 to 0.002736	No	ns
PSO vs. GWO	-0.005468	-0.01066 to -0.0002716	Yes	*
CMAES vs. WOA	-0.009843	-0.01600 to -0.003683	Yes	****
CMAES vs. GWO	-0.01285	-0.01901 to -0.006690	Yes	****
WOA vs. GWO	-0.003006	-0.007971 to 0.001959	No	ns

TABLE 12. Use-case 3: Statistical significance test compared between the different MOAs.

Games-Howell's multiple comparisons test	Mean Diff.	95.00% CI of diff.	Below threshold?	Summary
ABC vs. CSO	0.02476	0.01196 to 0.03757	Yes	****
ABC vs. DE	0.001053	-0.01196 to 0.01407	No	ns
ABC vs. Firefly	-0.00351	-0.01621 to 0.009195	No	ns
ABC vs. GA	0.01466	0.002145 to 0.02717	Yes	**
ABC vs. PSO	-0.01148	-0.02447 to 0.001500	No	ns
ABC vs. CMAES	0.02913	0.01738 to 0.04089	Yes	****
ABC vs. WOA	-0.00622	-0.01872 to 0.006284	No	ns
ABC vs. GWO	-0.01315	-0.02683 to 0.0005362	No	ns
CSO vs. DE	-0.02371	-0.03638 to -0.01103	Yes	****
CSO vs. Firefly	-0.02827	-0.04063 to -0.01591	Yes	****
CSO vs. GA	-0.01011	-0.02226 to 0.002053	No	ns
CSO vs. PSO	-0.03625	-0.04889 to -0.02360	Yes	****
CSO vs. CMAES	0.004369	-0.007009 to 0.01575	No	ns
CSO vs. WOA	-0.03098	-0.04313 to -0.01883	Yes	****
CSO vs. GWO	-0.03791	-0.05127 to -0.02455	Yes	****
DE vs. Firefly	-0.00456	-0.01714 to 0.008010	No	ns
DE vs. GA	0.0136	0.001226 to 0.02598	Yes	*
DE vs. PSO	-0.01254	-0.02539 to 0.0003189	No	ns
DE vs. CMAES	0.02808	0.01647 to 0.03969	Yes	****
DE vs. WOA	-0.00727	-0.01964 to 0.005097	No	ns
DE vs. GWO	-0.0142	-0.02776 to -0.0006384	Yes	*
Firefly vs. GA	0.01817	0.006114 to 0.03022	Yes	***
Firefly vs. PSO	-0.00798	-0.02052 to 0.004569	No	ns
Firefly vs. CMAES	0.03264	0.02138 to 0.04390	Yes	****
Firefly vs. WOA	-0.00271	-0.01475 to 0.009334	No	ns
Firefly vs. GWO	-0.00964	-0.02291 to 0.003629	No	ns
GA vs. PSO	-0.02614	-0.03849 to -0.01379	Yes	****
GA vs. CMAES	0.01447	0.003430 to 0.02552	Yes	**
GA vs. WOA	-0.02087	-0.03271 to -0.009037	Yes	****
GA vs. GWO	-0.0278	-0.04089 to -0.01472	Yes	****
PSO vs. CMAES	0.04062	0.02903 to 0.05220	Yes	****
PSO vs. WOA	0.005267	-0.007070 to 0.01760	No	ns
PSO vs. GWO	-0.00166	-0.01520 to 0.01187	No	ns
CMAES vs. WOA	-0.03535	-0.04638 to -0.02432	Yes	****
CMAES vs. GWO	-0.04228	-0.05464 to -0.02992	Yes	****
WOA vs. GWO	-0.00693	-0.02000 to 0.006142	No	ns

were computed throughout the complete FCR range of values (i.e. 1 - 200), thus demonstrating that the algorithms performed similarly in half of the comparison list (see Table 12).

TABLE 13. Statistical significance test between the run-times of the different MOAs.

Dunnett's T3 multiple comparisons test	Mean Diff.	95.00% CI of diff.	Below threshold?	Summary
ABC vs. CSO	0.0103	-3.640 to 3.661	No	ns
ABC vs. DE	-0.00553	-2.723 to 2.712	No	ns
ABC vs. Firefly	0.008433	-2.508 to 2.525	No	ns
ABC vs. GA	0.2542	-1.857 to 2.365	No	ns
ABC vs. PSO	-0.02453	-2.598 to 2.548	No	ns
ABC vs. CMAES	0.0838	-2.509 to 2.677	No	ns
ABC vs. WOA	0.7322	-1.705 to 3.170	No	ns
ABC vs. GWO	0.7017	-1.932 to 3.336	No	ns
CSO vs. DE	-0.01583	-3.303 to 3.272	No	ns
CSO vs. Firefly	-0.00187	-3.125 to 3.121	No	ns
CSO vs. GA	0.2439	-3.354 to 3.842	No	ns
CSO vs. PSO	-0.03483	-3.204 to 3.134	No	ns
CSO vs. CMAES	0.0735	-3.112 to 3.259	No	ns
CSO vs. WOA	0.7219	-2.338 to 3.782	No	ns
CSO vs. GWO	0.6914	-2.527 to 3.910	No	ns
DE vs. Firefly	0.01397	-2.992 to 3.020	No	ns
DE vs. GA	0.2597	-3.170 to 3.690	No	ns
DE vs. PSO	-0.019	-3.072 to 3.034	No	ns
DE vs. CMAES	0.08933	-2.981 to 3.159	No	ns
DE vs. WOA	0.7378	-2.202 to 3.678	No	ns
DE vs. GWO	0.7072	-2.398 to 3.812	No	ns
Firefly vs. GA	0.2458	-2.225 to 2.716	No	ns
Firefly vs. PSO	-0.03297	-2.908 to 2.842	No	ns
Firefly vs. CMAES	0.07537	-2.818 to 2.969	No	ns
Firefly vs. WOA	0.7238	-2.031 to 3.479	No	ns
Firefly vs. GWO	0.6932	-2.237 to 3.623	No	ns
GA vs. PSO	-0.2787	-2.807 to 2.249	No	ns
GA vs. CMAES	-0.1704	-2.719 to 2.378	No	ns
GA vs. WOA	0.478	-1.912 to 2.868	No	ns
GA vs. GWO	0.4475	-2.143 to 3.037	No	ns
PSO vs. CMAES	0.1083	-2.834 to 3.051	No	ns
PSO vs. WOA	0.7568	-2.050 to 3.563	No	ns
PSO vs. GWO	0.7262	-2.253 to 3.705	No	ns
CMAES vs. WOA	0.6484	-2.176 to 3.473	No	ns
CMAES vs. GWO	0.6179	-2.378 to 3.614	No	ns
WOA vs. GWO	-0.03057	-2.893 to 2.832	No	ns

Essentially, it should be highlighted that, on average, across 1000 separate trials, the GWO and WOA algorithms performed significantly better than the other approaches. It also shows that there was no substantial difference between the GWO and WOA algorithms, meaning that either algorithms can be employed. It also shows that there was no statistically significant difference in the performance of the ABC, CSO, DE, firefly, GA, and PSO algorithms. As a result, any algorithm from this collection of approaches can be put for usage.

Finally, we performed statistical significance tests between the run-times of the various algorithms, and the results are shown in Table 13. Because the number of samples per technique was fewer than 50 (see Table 8), the Dunnett's T3 posthoc test was utilized, showing itself as a superior strategy than the Games-Howell's test in this circumstance. It is clear that there was no substantial variation in the run-time of the different algorithms, which strongly supports the notion that adopting equal FCR across all approaches ensures fairness in comparison. As a result, when it comes to selecting an MOA for MIE on a run-time basis, any MOA can be chosen without sacrificing timing performance.

V. CONCLUSION

This article has investigated whether there exist any significant difference in the performance of different metaheuristic optimization algorithms (MOAs) as applied for medical image enhancement (MIE). To this effect, well-known MOAs were investigated for MIE based on some selected medical images obtained from the Medpix database. Fairness was achieved by terminating the different algorithms based on the fitness computation rate (FCR), which ensures that the objective function is consumed equally by all MOAs during their different algorithmic processes. From an empirical standpoint, and based on an average of 1000 Monte Carlo independent trials, the GWO and WOA algorithms achieved slightly better results than the other methods. However, there was no substantial difference in performance among the other MOAs, including the ABC, DE, CSO, GA, Firefly, and PSO algorithms. Furthermore, there was no significant difference in the performance of the GWO and WOA algorithms, thus suggesting that any of both algorithms can be used for MIE purposes. In terms of qualitative (i.e. visual) evaluation, there was little to differentiate the performance of the various algorithms. Furthermore, statistical analyses showed that there was no statistically significant difference in the computing times of the various methods. As a result, there may be no discernible difference in adopting one MOA over another for MIE applications. Nonetheless, the aim of this study merely seeks to inspire new and wider investigations based on a larger corpus of images, as well as based on the analysis of several other MOAs, with the goal to either substantiate or dispute the conclusions found herein.

ACKNOWLEDGMENT

The authors would like to thank the Tshwane University of Technology for their support. Opinions, findings, and conclusions or recommendations expressed in any publication generated by the NRF supported research are those of the author(s) alone, and the NRF accepts no liability whatsoever in this regard.

REFERENCES

[1] F. Zerka, V. Urovi, A. Vaidyanathan, S. Barakat, R. T. H. Leijenaar, S. Walsh, H. Gabrani-Juma, B. Miraglio, H. C. Woodruff, M. Dumontier, and P. Lambin, "Blockchain for privacy preserving and trustworthy distributed machine learning in multicentric medical imaging (C-DistriM)," *IEEE Access*, vol. 8, pp. 183939–183951, 2020.

[2] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016.

[3] B. Subramani and M. Veluchamy, "Fuzzy contextual inference system for medical image enhancement," *Measurement, J. Int. Meas. Confederation*, vol. 148, Dec. 2019, Art. no. 106967.

[4] L. Jiang, S. Ye, X. Yang, X. Ma, L. Lu, A. Ahmad, and G. Jeon, "An adaptive anchored neighborhood regression method for medical image enhancement," *Multimedia Tools Appl.*, vol. 79, nos. 15–16, pp. 10533–10550, Apr. 2020.

[5] H. Singh, A. Kumar, L. K. Balyan, and H.-N. Lee, "Fractional-order integration based fusion model for piecewise gamma correction along with textural improvement for satellite images," *IEEE Access*, vol. 7, pp. 37192–37210, 2019.

[6] M. O. Oloyede, G. P. Hancke, and H. C. Myburgh, "A review on face recognition systems: Recent approaches and challenges," *Multimedia Tools Appl.*, vol. 79, nos. 37–38, pp. 27891–27922, Oct. 2020.

[7] M. O. Oloyede and G. P. Hancke, "Unimodal and multimodal biometric sensing systems: A review," *IEEE Access*, vol. 4, pp. 7532–7555, 2016.

[8] M. O. Oloyede, G. P. Hancke, and N. Kapileswar, "Evaluating the effect of occlusion in face recognition systems," in *Proc. IEEE AFRICON, Sci., Technol. Innov. Africa*, Sep. 2017, pp. 1547–1551.

[9] M. O. Oloyede, G. P. Hancke, and H. C. Myburgh, "Improving face recognition systems using a new image enhancement technique, hybrid features and the convolutional neural network," *IEEE Access*, vol. 6, pp. 75181–75191, 2018.

[10] N. Salem, H. Malik, and A. Shams, "Medical image enhancement based on histogram algorithms," *Proc. Comput. Sci.*, vol. 163, pp. 300–311, Jan. 2019.

[11] M. Oloyede, G. Hancke, H. Myburgh, and A. Onumanyi, "A new evaluation function for face image enhancement in unconstrained environments using metaheuristic algorithms," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, pp. 1–18, Dec. 2019.

[12] H. Bello-Salau, A. M. Aibinu, A. J. Onumanyi, E. N. Onwuka, J. J. Dukiya, and H. Ohize, "New road anomaly detection and characterization algorithm for autonomous vehicles," *Appl. Comput. Informat.*, vol. 16, nos. 1–2, pp. 223–239, May 2018.

[13] H. Singh, A. Kumar, and L. K. Balyan, "A sine-cosine optimizer-based gamma corrected adaptive fractional differential masking for satellite image enhancement," in *Advances in Intelligent Systems and Computing*, vol. 741. Singapore: Springer, Aug. 2019, pp. 633–645.

[14] H. Bello-Salau, A. J. Onumanyi, R. F. Adebiyi, E. A. Adedokun, and G. P. Hancke, "Performance analysis of machine learning classifiers for pothole road anomaly segmentation," in *Proc. IEEE 30th Int. Symp. Ind. Electron. (ISIE)*, Kyoto, Japan, Jun. 2021, pp. 1–6.

[15] R. Kapoor, R. Gupta, L. H. Son, R. Kumar, and S. Jha, "Fog removal in images using improved dark channel prior and contrast limited adaptive histogram equalization," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 23281–23307, Aug. 2019.

[16] H. L. Tong, M. F. A. Fauzi, S. C. Haw, H. Ng, and T. T. V. Yap, "Automatic classification and retrieval of brain hemorrhages," in *Computational Science and Technology* (Lecture Notes in Electrical Engineering), vol. 481. Singapore: Springer, Aug. 2019, pp. 1–11.

[17] R. Firoz, M. S. Ali, M. N. U. Khan, M. K. Hossain, M. K. Islam, and M. Shahinuzzaman, "Medical image enhancement using morphological transformation," *J. Data Anal. Inf. Process.*, vol. 4, no. 1, pp. 1–12, 2016.

[18] X. Hu, Y. Yan, W. Ren, H. Li, Y. Zhao, A. Bayat, and B. Menze, "Feedback graph attention convolutional network for medical image enhancement," 2020, *arXiv:2006.13863*.

[19] S. Chakraborty, A. Raman, S. Sen, K. Mali, S. Chatterjee, and H. Hachimi, "Contrast optimization using elitist Metaheuristic optimization and gradient approximation for biomedical image enhancement," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 712–717.

[20] C. Munteanu and A. Rosa, "Gray-scale image enhancement as an automatic process driven by evolution," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 1292–1298, Apr. 2004.

[21] N. Sahebjamnia, A. M. Fathollahi-Fard, and M. Hajiaghaei-Keshteli, "Sustainable tire closed-loop supply chain network design: Hybrid metaheuristic algorithms for large-scale networks," *J. Cleaner Prod.*, vol. 196, pp. 273–296, Sep. 2018.

[22] L.-W. Wang, Z.-S. Liu, W.-C. Siu, and D. P.-K. Lun, "Deep lightening network for low-light image enhancement," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[23] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.

[24] U. Škvorc, T. Eftimov, and P. Korošec, "GECCO black-box optimization competitions: Progress from 2009 to 2018," in *Proc. Genet. Evol. Comput. Conf. Companion*, Jul. 2019, pp. 275–276.

[25] N. Veček, M. Mernik, B. Filipič, and M. Črepinšek, "Parameter tuning with chess rating system (CRS-tuning) for meta-heuristic algorithms," *Inf. Sci.*, vol. 372, pp. 446–469, Dec. 2016.

[26] M. Črepinšek, S.-H. Liu, and M. Mernik, "Replication and comparison of computational experiments in applied evolutionary computing: Common pitfalls and guidelines to avoid them," *Appl. Soft Comput.*, vol. 19, pp. 161–170, Jun. 2014.

[27] M. Črepinšek, S.-H. Liu, L. Mernik, and M. Mernik, "Is a comparison of results meaningful from the inexact replications of computational experiments?" *Soft Comput.*, vol. 20, no. 1, pp. 223–235, Jan. 2016.

[28] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart, "Designing and reporting on computational experiments with heuristic methods," *J. Heuristics*, vol. 1, no. 1, pp. 9–32, Sep. 1995.

[29] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes Univ., Kayseri, Turkey, Tech. Rep. TR06, Oct. 2005.

[30] X. S. Yang and S. Deb, "Cuckoo search via Lèvy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, 2009, pp. 210–214.

[31] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

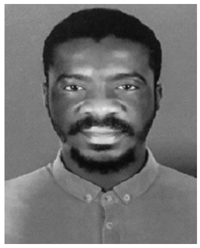
[32] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. Frome, U.K.: Luniver Press, Sep. 2010.

[33] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.

[34] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[35] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[36] J. J. Lau, S. Gayen, A. B. Abacha, and D. Demner-Fushman, "A dataset of clinically generated visual questions and answers about radiology images," *Sci. Data*, vol. 5, no. 1, pp. 1–10, Dec. 2018.



MUHTAHIR O. OLOYEDE (Member, IEEE) received the B.Eng. degree in electrical engineering from the University of Ilorin, Nigeria, the M.Sc. degree in information systems from the University of East London, U.K., in 2011, and the Ph.D. degree in computer engineering from the University of Pretoria, in 2019. He is currently a Postdoctoral Research Fellow with the Department of Electrical Engineering, Tshwane University of Technology. His research interests include application of biometric sensing systems, wireless sensor networks, image processing, computer vision, and artificial intelligence.

ADEIZA J. ONUMANYI (Member, IEEE) received the B.Eng. degree in electrical and electronics engineering from Abubakar Tafawa Balewa University, Bauchi, Nigeria, in 2005, and the M.Eng. and Ph.D. degrees in communication engineering from the Federal University of Technology (FUT) Minna, Nigeria, in 2010 and 2014, respectively. He is a currently a Researcher at South Africa's Council for Scientific and Industrial Research (CSIR), Pretoria. He has several research articles published in peer-reviewed journals and at IEEE flagship conferences. Between 2010 and 2021, he lectured and conducted research with the Department of Telecommunication Engineering, FUT Minna, where he was involved in securing several grants, serving on several organizing committees for various conferences, including IEEE conferences, reviewing several articles for high impact journals, and participating in various technical workshops. His research interests include spectrum sensing in cognitive radio, wireless sensor networks, smart transactive microgrids, DC nanogrids, radar systems, image processing, cyber-physical systems, and low powered wireless area networks.

HABEEB BELLO-SALAU (Member, IEEE) received the B.Tech. degree in electronic/electrical engineering from the Ladoke Akintola University of Technology, Ogbomoso, Nigeria, in 2009, the M.Sc. degree in communication engineering from International Islamic University Malaysia, Kuala Lumpur, in 2012, and the Ph.D. degree in communication engineering from the Federal University of Technology Minna, Nigeria, in 2017. He is currently working with the Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria. He has authored and coauthored more than 20 different research articles in peer reviewed journals and over 30 conference papers. His research interests include digital signal and image processing, vehicle *ad-hoc* networks, artificial intelligence, cognitive radio, and wireless sensor networks.

KARIM DJOUANI (Member, IEEE) is currently a Scientist and a Technical Group Supervisor of pattern recognition, machine learning, soft computing, networking systems, and robotics. In January 2011, he was appointed as a Full Professor with the French South African Institute of Technology (F'SATI), Tshwane University of Technology (TUT), Pretoria, South Africa. He is currently a Full Professor with University Paris-Est Créteil (UPEC). Since January 2014, he has been a recipient of the DST/NRF SARChI Chair in Enabled Environment for Assistive Living with TUT, where he was seconded by the French Ministry of Higher Education to F'SATI, from July 2008 to December 2010. Until July 2008, he was also the Manager of national and European projects with the LISSI Laboratory. He has authored/coauthored more than 300 articles in archival journals, conference proceedings, and 18 chapters in edited books and two books and seven patents. His research interests include development of novel and highly efficient algorithms for reasoning systems with uncertainty and optimization, networked control systems, wireless *ad-hoc* networks, wireless and mobile communication, wireless sensor networks, and robotics. He is a member of IEEE Communication, Computer, Robotics and Automation and Artificial Intelligence Societies and several French National Research Task Groups. He also acts as an expert and an advisor for different public and private institutions (ANR-France, NRF-South Africa, UNU, EU, Algerian Telecommunication operators, and different SMMEs and large companies). Since 2016, he has been acted as an Advisor for digitalization strategy and RnD for telecommunication operators.

ANISH KURIEN (Member, IEEE) received the D.Tech. degree in electrical engineering with specialization in telecommunication technology from the Tshwane University of Technology and the Ph.D. degree in computing (through co-tutelle) from the University of Paris-Est, France, in 2012. He is currently a Professor with the Department of Electrical Engineering, Tshwane University of Technology. He has been the Node Director of the French South African Institute of Technology (F'SATI), since 2010, a French-South African bilateral program that has been in existence, since 1997. He has over 21 years of teaching and research experience. He has been supervised and co-supervised over 39 postgraduate students, since 2006. He has contributed to over a 100 publications in journals, peer-reviewed conference proceedings, book chapters, one book, and a patent. His current research interests include the domains of wireless communications, radio resource management, and mobile network optimization.