Evolving Systems
https://doi.org/10.1007/s12530-022-09443-3

ORIGINAL PAPER

Chimp optimization algorithm in multilevel image thresholding and image clustering

Zubayer Kabir Eisham[1] · **Md. Monzurul Haque**[1] · **Md. Samiur Rahman**[1] · **Mirza Muntasir Nishat**[1] · **Fahim Faisal**[1] · **Mohammad Rakibul Islam**[1]

Received: 17 January 2022 / Accepted: 5 May 2022

Abstract

Multilevel image thresholding and image clustering, two extensively used image processing techniques, have sparked renewed interest in recent years due to their wide range of applications. The approach of yielding multiple threshold values for each color channel to generate clustered and segmented images appears to be quite efficient and it provides significant performance, although this method is computationally heavy. To ease this complicated process, nature inspired optimization algorithms are quite handy tools. In this paper, the performance of Chimp Optimization Algorithm (ChOA) in image clustering and segmentation has been analyzed, based on multilevel thresholding for each color channel. To evaluate the performance of ChOA in this regard, several performance metrics have been used, namely, Segment evolution function, peak signal-to-noise ratio, Variation of information, Probability Rand Index, global consistency error, Feature Similarity Index and Structural Similarity Index, Blind/Referenceless Image Spatial Quality Evaluatoe, Perception based Image Quality Evaluator, Naturalness Image Quality Evaluator. This performance has been compared with eight other well known metaheuristic algorithms: Particle Swarm Optimization Algorithm, Whale Optimization Algorithm, Salp Swarm Algorithm, Harris Hawks Optimization Algorithm, Moth Flame Optimization Algorithm, Grey Wolf Optimization Algorithm, Archimedes Optimization Algorithm, African Vulture Optimization Algorithm using two popular thresholding techniques-Kapur's entropy method and Otsu's class variance method. The results demonstrate the effectiveness and competitive performance of Chimp Optimization Algorithm.

Keywords Thresholding · Clustering · Optimization algorithm · Metaheuristic · ChOA

1 Introduction

For the past few years, image processing has been a rapidly expanding discipline with a wide range of applications. Newer and more efficient techniques are gradually being introduced and applied to various image processing application usage scenarios. Image thresholding and image clustering are two key studies in this field. Both of these studies hold significant impact in different fields such as medicine (Gao et al. 2011), agriculture (Lanthier et al. 2008), computer vision (Jolion et al. 1991), pattern recognition (Haralick and Kelly 1969), object identification (Barik and Mondal 2010), image synthesis (Reed et al. 2016), animation (Lu and Zhang 2010) and so on. These versatile range of use case scenarios and impact of these applications necessitate some organized, efficient and optimized methods.

Thresholding-based techniques have gained a lot of popularity among image segmentation methods due to their efficacy and ease of use. In these types of approaches, image

✉ Mirza Muntasir Nishat
mirzamuntasir@iut-dhaka.edu

Zubayer Kabir Eisham
zubayerkabir@iut-dhaka.edu

Md. Monzurul Haque
monzurulhaque@iut-dhaka.edu

Md. Samiur Rahman
samiurrahman20@iut-dhaka.edu

Fahim Faisal
faisaleee@iut-dhaka.edu

Mohammad Rakibul Islam
rakibultowhid@yahoo.com

[1] Department of EEE, Islamic University of Technology, Gazipur, Bangladesh

Published online: 27 May 2022

Segmentation requires thresholding measures to isolate different objects or specific parts of the image or to extract specific image information. In Bi-level thresholding, the image is divided into two classes where pixel values less than a specific threshold value are classified under one class and pixel values above the specific threshold are classified under another class. Multilevel thresholding is generally used in RGB images where the whole image is subdivided into multiple classes based on multiple threshold values. This multilevel thresholding technique is the baby step for image clustering where the image is first segmented using multiple threshold values and later on, these thresholds are used to generate different clusters. Clustering can also be done using a single threshold value for each color channel (Red, Green and Blue) but multilevel thresholding for each color channel results in more efficient clustering of images. The image pixels are classified in such a way that the pixels with similar spatial coordinates fall under the same cluster, and pixels assigned in the same cluster vary with pixels assigned to other clusters, which is achieved by using the threshold values for each color channel.

Kapur's maximum entropy method (Kapur et al. 1985) and Otsu's class variance method (Otsu Jan. 1979) are two the most popular methods for image segmentation and clustering purposes. Both of these methods require optimized values of respective objective functions which itself is a cumbersome process, also the computational complexity of these techniques arises significantly as the number of threshold increases. This is where metaheuristic optimization algorithms come into play. These optimization algorithms not only deal with the computational accuracy of above mentioned multilevel thresholding techniques, also they are capable of reducing the complexity and computational time of the problem.

Chimp Optimization Algorithm (ChOA) is a novel metaheuristic optimization algorithm proposed by M. Khishe and M.R. Mosavi in 2020 (Khishe and Mosavi 2020), where they mimicked the sexual motivation and diverse intelligence of chimps in group hunting. ChOA provides a mathematical modeling of hunting mechanisms of 4 types of chimps along with their respective hunting strategies. This algorithm has been proven to be worth solving many complex engineering problems for its excellent balance between exploration and exploitation (Nagadurga et al. 2021; Pedram Haeri Boroujeni and Pashaei 2021). In this paper, we have used ChOA for image clustering using multilevel image thresholding technique. Kapur's entropy method and Otsu's thresholding method were used for evaluating the performance of ChOA and the performance has been compared with other well known metaheuristic algorithms, namely Particle Swarm Optimization Algorithm (PSO), Whale Optimization Algorithm (WOA), Salp Swarm Optimization Algorithm, Harris Hawks Optimization Algorithm (HHO), Moth Flame Optimization Algorithm (MFO), Grey Wolf Optimization Algorithm (GWO), Archimedes Optimization Algorithm (AOA) and African Vulture Optimization Algorithm (AVOA).

2 Related works

Image segmentation has always been a challenging task for the researchers. A good number of methods have been found in the literature for image segmentation such as- edge based method (Muthukrishnan and Radha 2011), wavelet transform based method (Demirhan et al. 2015), neural network based method (Wang et al. 2010), clustering based method (Chuang et al. 2006), threshold based method (Ouadfel and Taleb-Ahmed 2016) etc. Threshold-based techniques, more specifically multilevel threshold-based techniques have been highly adapted for image segmentation applications that can extensively be found in the literature. In 1985, Kapur et al. proposed one of the most popular thresholding based image segmentation methods, named as maximum entropy method, where they optimized the maximum entropy of the image histogram to measure the homogeneity of different classes and find the optimal threshold values (Kapur et al. 1985). In Otsu (Jan. 1979) the author presented a new thresholding technique known as Otsu's method where the target image is similarly sub divided into various classes based on multiple threshold values which are found by maximizing the interclass variance.

Image clustering approaches based on thresholding techniques appear to be more effective than other approaches that can be found in the literature namely, K-means data classification method (MacQueen 1967), Fuzzy c-means (FCM) method (Bezdek et al. 1984) etc. Thresholding based approaches overcome the drawbacks of other clustering methods such as, reliance on the number of clusters, computational complexities, repetitive natures and so on. For grayscale images, this process of clustering is relatively easier by means of binarization and thresholding. In such instances, the quality of clusters improves as the number of thresholds increases in multilevel thresholding. When multilevel thresholding is used on color photos, however, the technique gets more complicated. Demirci et al. (2014) proposed an idea where they determined a single threshold value separately for each color channel (Red, Green, and Blue) using Kapur's entropy method and Otsu's method for image clustering. Authors in Rahkar Farshi et al. (2018) extended this idea of multilevel thresholding for each color channel separately using Kapur's entropy method. They showed that their approach resulted in better clustering than the work in Demirci et al. (2014) as they could increase the number of clusters by increasing the number of sub-cubes generated by means of multilevel thresholding for each color channel.

The rising problem complexity in multilevel thresholding highly motivated the use of optimization algorithms in this context for quite a long time. Authors in Sharma et al. (2018) used Firefly Algorithm (FA) in grayscale image segmentation using multiple thresholds. Similar works can be found in Pei et al. (2009) where the authors used Otsu's method by using Differential Evolution algorithm.

For RGB images, the application of optimization algorithms can be found extensively in the literature for multilevel thresholding purposes. Cuckoo Search Algorithm (CSO) were used for multilevel thresholding as a means of image segmentation using Kapur's entropy method (Brajevic et al. 2012). Authors in [22] used Crow Search Algorithm in the similar context for optimal multilevel image segmentation. Whale optimization Algorithm (WOA) has been used in underwater multilevel image segmentation using Kapur's maximum entropy technique (Yan et al. 2021). Authors in Jia et al. (2019) applied a modified Moth Flame Optimization Algorithm in multilevel color segmentation purpose on 10 test images to yield more optimal and accurate results comparing to other algorithms in this regard. In [42], authors developed an improved Biogeography-Based Optimization Algorithm for the applications of clustering optimization and applied it in medical image segmentation. Authors in Wong et al. (2011) used Particle Swarm Optimization Algorithm for image clustering using two distinct fitness functions and showed that PSO works better than conventional K-means method because of its ability to generate more compact clusters. A hybrid Firefly and Particle Swarm Optimization Algorithm was used in the similiar purpose in the works of Rahkar Farshi and Ardabili (2021) which resulted in comparatively superior performance than 4 other metaheuristic algorithms that they compared with. In Rahkar Farshi et al. (2018), the authors evaluated the performance of Particle Swarm Optimization (PSO) and Forest Optimization Algorithm (FOA) in image clustering based on multilevel thresholding for each color channel, following by the work of Demirci et al. (2014) in the similar context who did the clustering using a single threshold for each color channel.

The Chimp Optimization algorithm has gained much popularity in recent days in several fields of engineering and complex applications. The innate strategic strengths of this algorithm of not getting trapped in local minima for multi dimensional problems and avoiding slow convergence speed due to its proper balance between exploration and exploitation phase, this algorithm tends to provide better performance in most of the engineering applications as compared to other existing algorithms (Kharrich et al. 2021). In the image processing genre, where the optimization problems become compact due to their high computational complexity, specially in multilevel thresholding and threshold based clustering, ChOA can provide extensive performance in this regard. Authors in Houssein et al. (2021) used ChOA in a similar context for segmentation of thermography based breast cancer imaging. In Tianqing et al. (2021), authors used a novel two-phase approach for classifying chest X-ray images, for real time detection of COVID-19 cases. They used deep CNN in first phase and extreme learning machines in second phase, stabilized by Chimp Optimization algorithm for better result.

Some more advances and applications of Chimp Optimization Algorithm can be found in recent literatures. Authors in Kaur et al. (2021) applied sin- cosine functions to update the equations of Chimp and thus developed a novel fusion algorithm named SChoA for HLS of datapaths in digital filters and engineering applications. In Wang et al. (2021), authors developed a binary version of Chimp optimization algorithm (BChOA) to solve optimization problems. Authors in Khishe et al. (2021) proposed a weighted Chimp Optimization algorithm by demonstrating that the proposed algorithm generates better results in terms of convergence speed and avoidance of local minima as compared to other metaheuristic algorithms. In Dhiman (2021), authors proposed a hybrid algorithm named SSC (Spotted Hyena-based Chimp Optimization Algorithm) for solving engineering problems, which is a combination of sine-cosine function and attacking strategy of Spotted Hyena Optimizer (SHO). Authors in Kaidi et al. (2022) proposed a Dynamic Levy Flight Chimp optimization algorithm to mitigate the problem of local optima stagnation of Chimp Optimization algorithm. In the works of (Khishe and Mosavi 2020), authors trained an Artificial Neural Network (ANN) using Chimp Optimization algorithm for the purpose of classification of underwater acoustical dataset.

This study focuses on carrying out an comprehensive investigation on the performances of different optimization algorithms, based on image thresholding. Hence, the performances are compared in an extensive fashion with a view to executing how Chimp optimization algorihm can play a significant role in terms of optimizing the fitness function using Kapur and OTSU's method for image clustering using multilevel thresholding. Therefore, the paper at hand provides a wide-ranging perspective in terms studying the applicability and compatibility of Chimp optimization algorithm pertaining to image thresholding and image clustering which will open new windows in the domain of image processing.

3 Methodology

Thresholding can be thought of a statistical-decision theory problem, and the goal of the problem is to minimize the average error resulting from assigning pixels to two or more groups. The probability density function (PDF) of each group's intensity level and the probability that each group's occurrence in a given application are the two fundamental pillars of the solution to the thresholding problem, but predicting the nature of the PDF is difficult. Otsu's method and Kapur's method are some of the alternative of this problem, and applying these alternative solutions to multilevel thresholding and determining the the threshold value with stochastic algorithm is one of the way to achieve the threshold values.

Gray scale images, also RGB images, have intensities in the range of $\{0, 1, 2, ..., L-1\}$; thus the probability of the ith gray level is given by the equation that follows.

$$c_i = \frac{h_i}{M \times N} \tag{1}$$

In (1), M and N signifies the size of an image and h_i points to pixel number of the level i, $0 \leq i \leq (L-1)$. In the case of multi-level thresholds Th_i, Th_2 are used for clarification, and number of pixels clustered will be incremented by one. In the case of gray picture threshold, value of the pixels between two thresholds will be rounded up to the immediate highest threshold i.e. pixel value between $Th_i < t_1, t_2, ..., Th_{i+1} - 1 \leq Th_i$ will be rounded up to Th_{i+1}.

In the case of RGB image, the same concept of rounding up is applied to each color histogram.

3.1 Kapur's method

Kapr's method (Mirjalili et al. 2017) was devised from the theory of entropy, and though it was, at first, proposed for binary thresholding, however, the technique can also be applied to multilevel thresholding.

$$Fitness(Th_1, Th_2, ..., Th_n) = His_0 + His_1 + \cdots + His_n \tag{2}$$

In (2), Th_i values are thresholds value for multilevel segmentation, and His_i values denotes histogram entropy value up to the particular threshold. Each of the entropy is given by the following equations.

$$His_0 = -\sum_{i=0}^{Th_1-1} \frac{c_i}{w_0} ln \frac{c_i}{w_0}, w_0 = \sum_{i=0}^{Th_1-1} c_i \tag{3}$$

$$His_1 = -\sum_{i=Th_1}^{Th_2-1} \frac{c_i}{w_1} ln \frac{c_i}{w_1}, w_1 = \sum_{i=Th_1}^{Th_2-1} c_i \tag{4}$$

$$His_n = -\sum_{i=Th_n}^{L-1} \frac{c_i}{w_n} ln \frac{c_i}{w_n}, w_n = \sum_{i=Th}^{L-1} c_i \tag{5}$$

3.2 Otsu's method

Otsu's method (Heidari et al. 2019) was devised from class variance algorithm. In terms of optimization algorithm, the objective functions is stated below.

$$Fitness(Th_1, Th_2, ..., Th_n) = \sigma_0 + \sigma_1 + \cdots + \sigma_n \tag{6}$$

In (6), σ_i values are between-class variance values, and they are given by the next sets of equations.

$$\begin{gathered}\sigma_0 = w_0(\mu_0 - \mu_t)^2 \\ w_0 = \sum_{i=0}^{Th_1-1} c_i, \mu_0 = \sum_{i=0}^{Th_1-1} \frac{ic_i}{w_0}\end{gathered} \tag{7}$$

$$\begin{gathered}\sigma_1 = w_1(\mu_1 - \mu_t)^2 \\ w_1 = \sum_{i=Th_1-1}^{Th_2-1} c_i, \mu_1 = \sum_{Th_1-1}^{Th_2-1} \frac{ic_i}{w_1}\end{gathered} \tag{8}$$

$$\begin{gathered}\sigma_n = w_n(\mu_n - \mu_t)^2 \\ w_1 = \sum_{i=L-1}^{Th_n-1} c_i, \mu_n = \sum_{Th_n}^{L-1} \frac{ic_i}{w_n}\end{gathered} \tag{9}$$

In (7)(8)(9), μ_t is defined as follow.

$$\mu_t = \sum_{i=0}^{L-1} c_i \tag{10}$$

In (7)(8)(9) and (10), μ_i denotes the mean of groups where $0 \leq i \leq n$ and the overall image average value is denoted by μ_T

4 Image clustering

After achieving the best set of threshold values, where number of threshold value depends on the application (i.e. particular image), the set of values can be used to cluster the image. In RGB image, three sets of thresholds are used to cluster each dimension of color (i.e. RGB image where there is a dimension for each color histogram). The integration of thresholds values to color partition was done by the techniques in Demirci et al. (2014). Though the technique in (Demirci et al. 2014) was designed for single value threshold, the authors in (Rahkar Farshi et al. 2018) enhanced the technique for application in

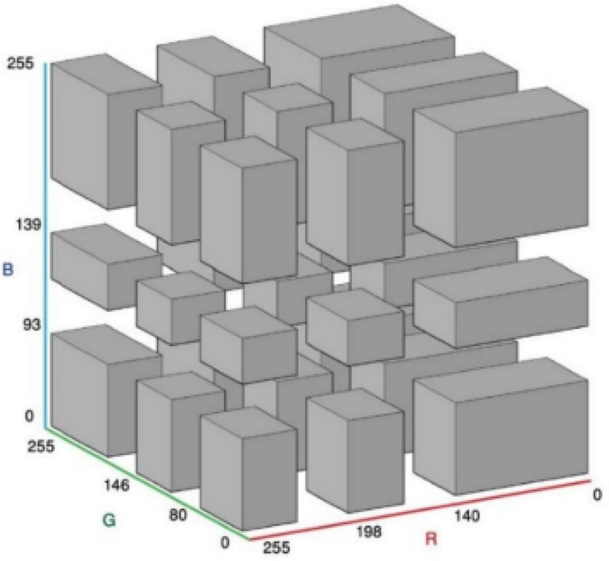


Fig. 1 3D color space with assigned clusters (Rahkar Farshi et al. 2018)

multi-level thresholds. Even after the criteria for each color channel are determined, establishing meaningful clusters with information from each color channel is a key issue. Subsets of color space are created using the threshold values computed for each channel. As a result, in Fig. 1, there are cubes instead of discreet values of the pixels. Figure 1 was taken from (Rahkar Farshi et al. 2018). This is due to the rounding up of pixel values to the nearest highest threshold(i.e. pixel value between $Th_i < t_1, t_2, ..., Th_{i+1} - 1 \leq Th_i$ will be rounded up to Th_{i+1}), hence creating cubes in RGB color space. The distribution of cubes differs from image to image, and by increasing the number of thresholds, larger cubes can be divided into smaller cubes and the maximum number of cubes is given by (11). Clusters may appear to be far apart to be related. However, image clustering with the technique above creates meaningful clusters as shown in Rahkar Farshi et al. (2018). The size of the cubes in RGB space is related to individual pixel intensity distribution of an image, thus creating unique positions of cubes in RGB color space for particular image, and considering the homogeneity of an image, though large size of cube enhances the homogeneity, smaller size of cubes can also maintain the homogeneity if the number of thresholds are increased; the reason is pixel value are limited to the range [0,255]. So, increasing the number of thresholds will create such distribution where the clustered image will have more similar distribution to the original image, hence increasing homogeneity.

$$max.\ number\ of\ clusters\ =\ (number\ of\ threshold\ +\ 1)^3 \tag{11}$$

5 Necessary parameters

5.1 Segment evolution function

To quantitatively judge the achieved thresholds values for RGB image, evolution function is used which was first proposed in Liu and Yang (1994b) and the technique is elaborated in Liu and Yang (1994a) and Borsotti et al. (1998).

$$F = \frac{1}{1000(M \times N)} \sqrt{R} \sum_{i=L-1}^{R} \frac{e_i^2}{\sqrt{A_i}} \tag{12}$$

In (12), $M \times N$ gives the number of pixels; R denotes the number of obtained regions. A_i gives the pixel number in the i)th region. Average of color error in the ith region is signified by e_i, and it is defined as the sum of the Euclidean distances between pixels of the ith region in the original color image and the attributed pixel values of the ith region in the segmented image. The F-value shrinks excessively, and to prevent that $\sqrt{R}$ term is multiplied. The number of region is inversely proportional to F, and lower the F, the better.

Related to this there are two other widely used quantitative assessment function termed as F' and Q, proposed in Borsotti et al. (1998). F' is defined as below.

$$F'(I) = \frac{1}{1000(M \times N)} \sqrt{\sum_{A=1}^{MAX} [R(A)]^{1+\frac{1}{A}}} \times \sum_{i=L-1}^{R} \frac{e_i^2}{\sqrt{A_i}} \tag{13}$$

Q is further reinforced by F'.

$$Q(I) = \frac{1}{1000(M \times N)} \sqrt{R} \sum_{i=1}^{R} \left[\frac{e_i^2}{1 + logA_i} + \left(\frac{R(A_i)}{A_i} \right)^2 \right] \tag{14}$$

R is the number of regions marked, and A_i represents pixel numbers in the *ith* region, and e_i denotes the *ith* color region. Lower values of F' and Q are desired for better clustering performance.

In Farshi et al. (2020) the authors represented these 3 quantitative assessment functions in an organized manner.

5.2 Structure Similarity Index (SSIM)

SSIM is given by the following equation (Wang et al. 2004).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{15}$$

where $\mu_x$and μ_y denote the mean intensity of the original image and that of the segmented image, respectively. σ_x^2 and σ_y^2 denote the standard deviation of the original image and that of the segmented image, respectively. σ_{xy} denotes the co-variance between the original image and the segmented image. c_1 and c_2 are constants.

According to brightness, contrast, and structural similarity, the SSIM is utilized to detect the similarity between the segmented image and the original image.The image segmentation impact improves as the SSIM value approaches one.

5.3 Peak signal-to-noise ratio (PSNR)

PSNR is given by the following equations (Aldahdooh et al. 2018).

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE^2}\right) \tag{16}$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - k(i,j)]^2 \tag{17}$$

Based on the intensity value of the image, the PSNR is used to detect the difference between the segmented image and the original image.The better the image segmentation effect, the higher the PSNR value. A segmented image with a greater PSNR value may be worse than a segmented image with a lower PSNR value due to differences in human visual acuity.

5.4 Variation of information (VOI)

VOI refers to the unrelated randomness property between different image segments, that is, how much the distribution of randomness varies from one segment to another. A lower value of VOI denotes better performance.

5.5 Probability Rand Index (PRI)

PRI denotes a connection between the segmented image and the main image. Higher value of this parameter is desired.

5.6 Global consistency error (GCE)

GCE denotes the extent to which one segmented image can be viewed or considered as a refinement of another segmented image. Lower result of this parameter is desired as it indicates better performance.

5.7 Feature Similarity Index (FSIM)

This is one of the important image segmentation parameters for image quality assessment that measures how much dependency the segmented image has with the original image. FSIM value always ranges between -1 to 1. A higher value of FSIM signifies better performance.

$$FSIM(x,y) = \frac{\sum_{x\in\Omega} S_L(X)PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{18}$$

Original image. FSIM value always ranges between – 1 and 1. A higher value of FSIM signifies better performance.

5.8 Blind/referenceless image spatial quality evaluatoe (BRISQUE)

BRISQUE uses statistical method of local normal luminance coefficients is used by BRISQUE to measure probable losses of “naturalness” in the image due to distortions. It does not compute ringing, blur, blocking, or anyother distortion specific parameters. Lower score signifies greater quality of an image (Mittal et al. Dec. 2012; MATLAB 2021).

5.9 Perception based image quality evaluator (PIQE)

PIQE is an unbiased paramter for image quality assessment without relying on subjective view. This paremeter measures distortions based on local features from visibly important spatial regions, and smaller PIQE score signifes better quality (Venkatanath et al. 2015; MATLAB 2021).

5.10 Naturalness image quality evaluator (NIQE)

Based on space domain natural scene statistic (NSS) model, NIQE measures the quality with the construction of some quality indicative statistical features, and similarly better quality is indicated by lower NIQE score. Mittal et al. (2013) MATLAB (2021)

6 Chimp Optimization Algorithm

Chimp optimization algorithm (ChOA) is a stochastic algorithm. In recent years, there has been a paradigm shift in algorithms in solving multidimensional problems. The major reason is deterministic algorithms are quite computationally expensive when it comes to finding global optimum value. Due to the randomness as an innate property of these algorithms, stochastic algorithms have high probability of convergence to global optimum in finite number of iterations, thus lessening the computational burden.

ChOA is one such algorithm, and it is a nature inspired swarm intelligence algorithm. Most of the stochastic algorithms have two fundamental phases: exploration and exploitation. In the exploration phase, search agents, initiated by the algorithm, roam around the search space (i.e. the variable space) for the best fitness value (output of the multi-dimensional function) of the objective function (multi-dimensional function). The exploration and the exploitation phase are divided in terms of iteration, and an outside function based on the iteration values determine the transition from exploration to exploitation. In the exploitation phase, the search agents circles around the best solution obtained so far in quest of finding more better solution. Search agents have cognitive abilities modeled and governed by few mathematical equations, and it is this cognitive ability along with the transition from exploration to exploitation phase makes stochastic algorithm appropriate in image thresholding application.

6.1 Inspiration Of ChOA

ChOA was inspired from the group behavior of chimps in hunting prey (Khishe and Mosavi 2020). There are four types of chimps when it comes to hunting: driver, barrier, chaser and attacker. Driver keeps track of prey's movement; while barriers try to impede the progression of the prey; meanwhile chaser move swiftly to catch the prey, and finally it's the attacker that launches the attack based on anticipating prey's movement. It's a group hunting, and role of each member is crucial in order to catch the prey. Attacker chimps are usually the ones who do the hunting. Driver, barrier, and chaser chimps are all involved in the hunting procedure on occasion. Unfortunately, there is no information regarding the best position in an abstract search space (prey). It is assumed that the first attacker (best solution available), driver, barrier, and chaser are better informed about the location of possible prey in order to mathematically imitate the chimps' behavior. As a result, four of the greatest answers so far are saved, and other chimps are pushed to update their places based on the best chimps' placements.

6.2 Mathematical modeling Of ChOA

The four individual groups search the issue space locally and globally using their unique patterns. Driving and chasing are given by the following equations.

$$\mathbf{d} = |\mathbf{c}.\mathbf{x}_{prey}(t) - \mathbf{m}.\mathbf{x}_{chimp}(t)| \tag{19}$$

$$\mathbf{x}_{chimp}(t+1) = \mathbf{x}_{prey}(t) - \mathbf{a}.\mathbf{d} \tag{20}$$

Current iteration number is defined with the symbol t, and **a**,**m** and **c** are the coefficient vectors, $\mathbf{x}_{prey}$ is prey's vector position and $\mathbf{x}_{chimp}$ gives the chimp's vector position. **a**,**m** and **c** are defined by the equation that follows.

$$\mathbf{a} = 2.\mathbf{f}.\mathbf{a}_1 - \mathbf{f} \tag{21}$$

$$\mathbf{c} = 2.\mathbf{r}_2 \tag{22}$$

$$\mathbf{m} = \mathbf{Choatic\_value} \tag{23}$$

Non-linear reduction of **f** is done, from 2.5 to 0, in both the phases. Random vectors are generated in each iteration and two random sets are assigned to $\mathbf{r}_1$and $\mathbf{r}_2$, and **m** value is generated from various chaotic map.

In exploitation phase, the chimps behavior are modeled by the equations as follows.

$$\mathbf{d}_{attacker} = |\mathbf{c}_1.\mathbf{x}_{attacker} - \mathbf{m}_1.\mathbf{x}| \tag{24}$$

$$\mathbf{d}_{barrierr} = |\mathbf{c}_2.\mathbf{x}_{barrier} - \mathbf{m}_2.\mathbf{x}| \tag{25}$$

$$\mathbf{d}_{chaser} = |\mathbf{c}_3.\mathbf{x}_{chaser} - \mathbf{m}_3.\mathbf{x}| \tag{26}$$

$$\mathbf{d}_{driver} = |\mathbf{c}_4.\mathbf{x}_{driver} - \mathbf{m}_4.\mathbf{x}| \tag{27}$$

$$\mathbf{x}_1 = \mathbf{x}_{attacker} - \mathbf{d}_{attacker}.\mathbf{a}_1 \tag{28}$$

$$\mathbf{x}_2 = \mathbf{x}_{barrier} - \mathbf{d}_{barrier}.\mathbf{a}_2 \tag{29}$$

$$\mathbf{x}_3 = \mathbf{x}_{chaser} - \mathbf{d}_{chaser}.\mathbf{a}_3 \tag{30}$$

$$\mathbf{x}_4 = \mathbf{x}_{drivere} - \mathbf{d}_{driver}.\mathbf{a}_4 \tag{31}$$

$$\mathbf{x}(t+1) = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4}{4} \tag{32}$$

$$\mathbf{x}_{chimp}(t+1) = \begin{cases} \mathbf{x}_{prey}(t) - \mathbf{a}.\mathbf{d}\ if\,\mu < 0.5 \\ \mathbf{Chaotic\_}value\ if\,\mu \geq 0.5 \end{cases} \tag{33}$$

In the above equations **c** is a random vector ranging from 0 to 2 (i.e.[0,2]), and **a** is random variable ranging from [-2**f**,2**f**]. In case of chaotic map (Saremi et al. 2014), the initial value was set to 0.7 in image thresholding optimization problem.

The original ChOA algorithm incorporate six different chaos map and two different sets of equations to update dynamic strategies. The dynamic strategy is used for determining the coefficients $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$, and $\mathbf{c}_4$.

For the sake of the simplicity of this work, only one set of dynamic strategy and one chaos map were used. This particular version of ChOA is labelled as **ChOA12**. The chaos map is known as Gauss or Mouse. It's defined as follows.

$$x_{i+1} = \begin{cases} 1 & x_i = 0 \\ \frac{1}{mod(x_i,1)} & otherwise \end{cases} \tag{34}$$

In the final stage, chimps relinquish their hunting responsibilities after acquiring meet and subsequent social drive (sex and grooming). As a result, they strive to gather meat in a chaotic manner.In tackling high-dimensional problems, ChOA is aimed to address the two issues of sluggish convergence speed and entrapment in local optima. This chaotic behavior in the last step aids chimps in overcoming the two issues of entrapment in local optima and sluggish convergence rate in high-dimensional problem solving.

ChOA allows the chimps to update their locations based on the positions of the attacker, barrier, chaser, and driving chimps and assault the prey, according to the operators who have already been provided. However, ChOA may still be at risk of becoming trapped in local minima, necessitating the use of other operators to overcome this problem. Despite the fact that the proposed driving, blocking, and chasing mechanisms appear to depict the exploration process, ChOA requires additional operators to focus on the exploration phase.

6.3 Pseducode

Algorithm 1 Algorithm for Chimp Optimization Algorithm

```
Input: Number of iteration, Objective Function, Chimp number
Output: Objective Value
    InitialisationChimp Population, f,m,a
    Calculate the initial position of each chimp
    Divide chimps randomly into independent groups
    This is repeated before stopping conditions are met
    Calculate the fitness of each chimp
    Determine $\mathbf{x}_{attacker}, \mathbf{x}_{chaser}, \mathbf{x}_{barrier}, \mathbf{x}_{diver}$
    While current iteration < Maximum Number of iteration
        for loop each chimp
        Extract the chimp's group
        Use its mathematical model to update f,m,c
        Use f, m and c to calculate a and then d
        end for loop
        for each search chimp
            if (μ< 0.5)
                if ($|a| < 1$)
                    Update the position of the current search agent by the models
                else if ($|a| > 1$)
                Select a random search agent
                end if
            else if (μ>0.5)
                Update the position of the current search
                end if
            end for loop
        Update f, m, a and c
        Update $x_{Attacker}$, $x_{Driver}, x_{Barrier}, x_{Chaser}$
        t=t+1
    end while
    return $x_{Attacker}$
```

6.4 Flowchart

See Fig. 2.

7 Result and analysis

See Figs. 3, 4, 5 and 6

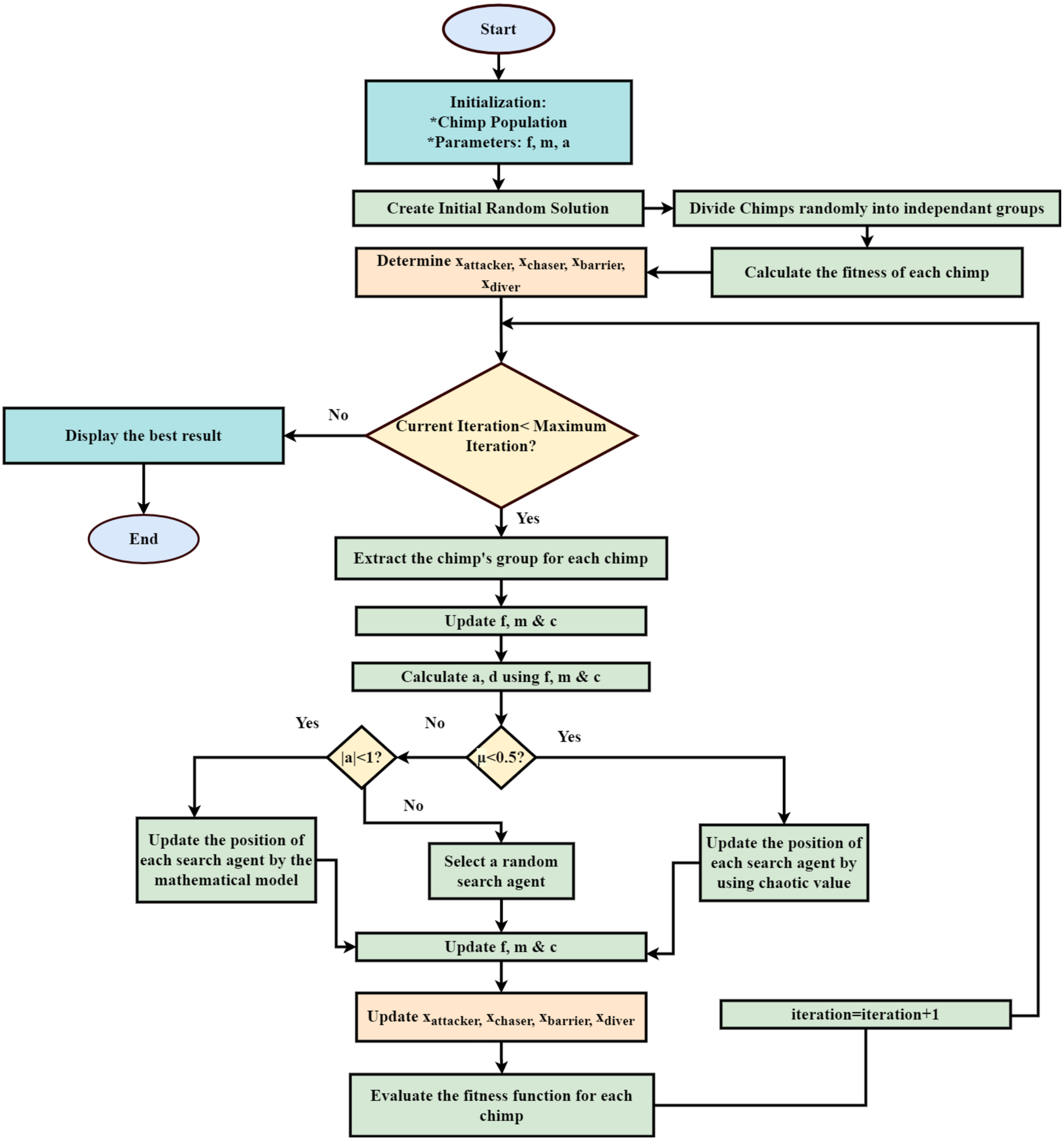
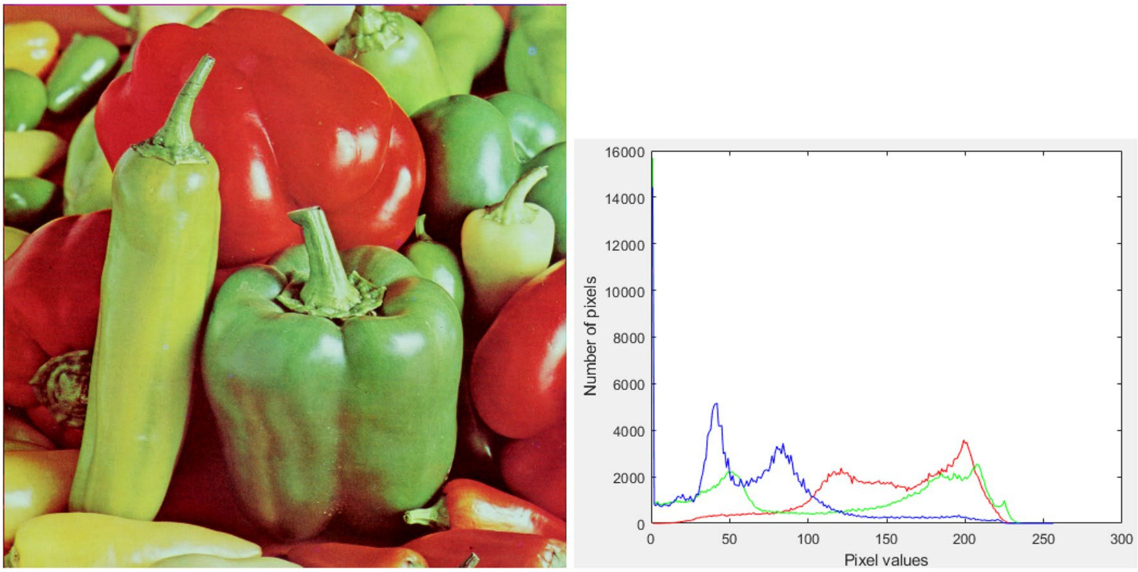

Start
Initialization:
*Chimp Population
*Parameters: f, m, a
Create Initial Random Solution
Divide Chimps randomly into independant groups
Calculate the fitness of each chimp
Determine x_attacker, x_chaser, x_barrier, x_diver
Current Iteration< Maximum Iteration?
No
Display the best result
End
Yes
Extract the chimp's group for each chimp
Update f, m & c
Calculate a, d using f, m & c
μ<0.5?
Yes
No
|a|<1?
Yes
No
Update the position of each search agent by the mathematical model
Select a random search agent
Update the position of each search agent by using chaotic value
Update f, m & c
Update x_attacker, x_chaser, x_barrier, x_diver
Evaluate the fitness function for each chimp
iteration=iteration+1

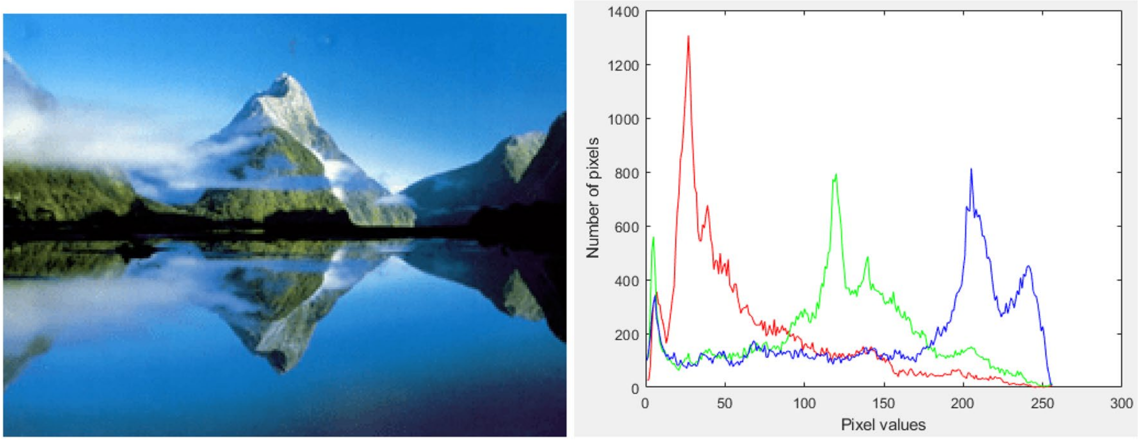
Fig. 2 Flowchart of Chimp Optimization Algorithm



(a) Image 1

(b) Histogram of image 1

Fig. 3 Original image 1



(a) Image 2

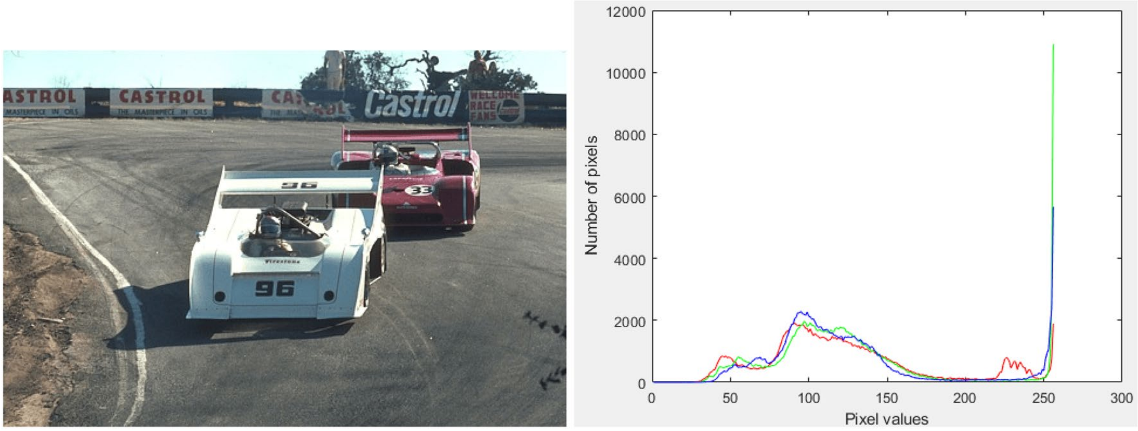
(b) Histogram of image 2

Fig. 4 Original image 2

7.1 Overall method

In this work, Chimp optimization algorithm has been applied in multilevel image thresholding and image clustering problem using Kapur's entropy method and Otsu's method. The performance of the proposed algorithm is evaluated based on multiple performance metrics. Later on, the overall performance has been compared with 8 other existing metaheuristic algorithms in the similar context to depict a conclusion on the efficiency of ChOA in this problem.

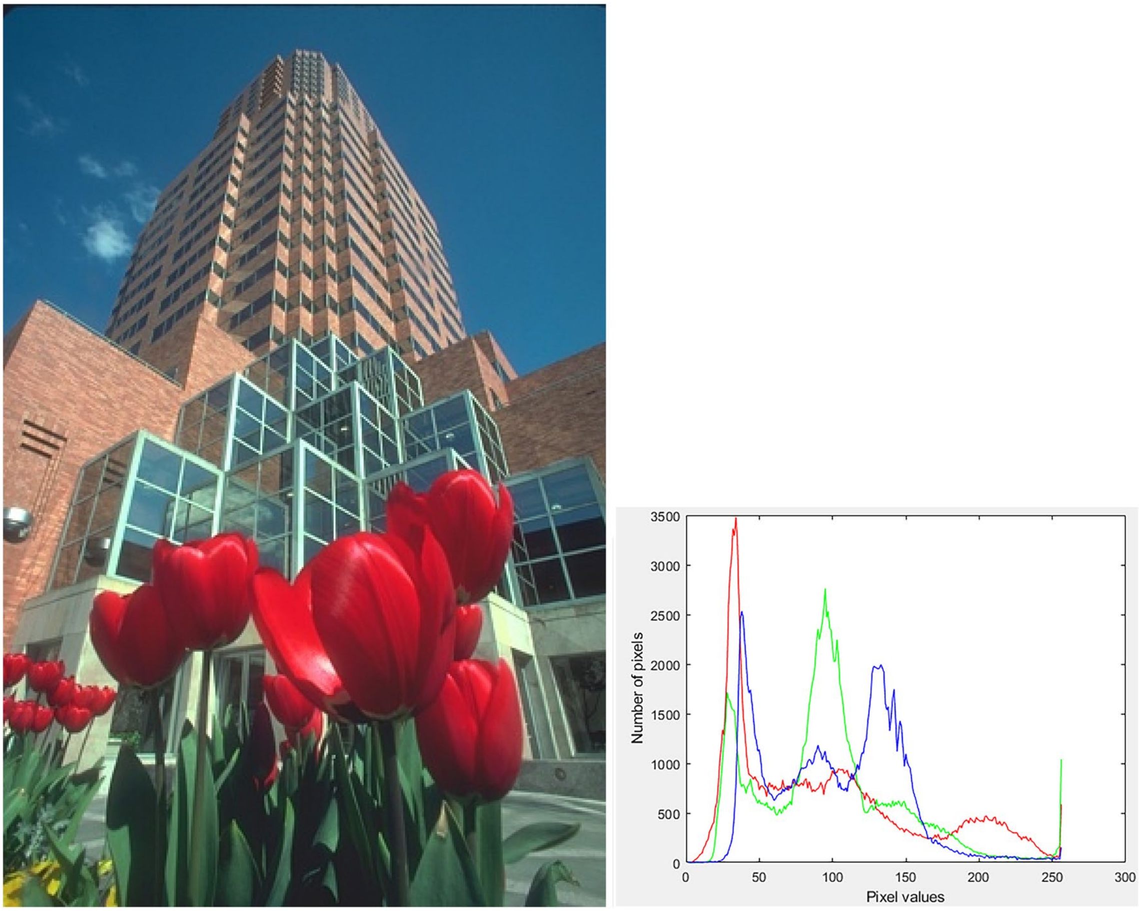
However, due to the stochastic nature of the algorithms and pixel distribution of images, it is hard to evaluate algorithms analytically. The performance of ChOA can be evaluated in terms of two broad applications, image clustering and image segmentation. The tables shows the average value after 30 tries and in each tries there were 100 iterations and in each iteration number of particles were 100. Chimp Optimization Algorithm was applied for each color channel (Red, Green and Blue) separately. Maximum entropy was evaluated for each color channel using Kapur's entropy method, similarly that has been done for Otsu's class variance method. The objective functions were maximized to generate k number of thresholds for each channel individually (k = 2, 5, 8, 10). These thresholds in 3 dimensional



(a) Image 3

(b) Histogram of image 3

Fig. 5 Original image 3



(a) Image 4

(b) Histogram of image 4

Fig. 6 Original image 4

plane were used to further generate small sub-cubes, or clusters.

Tables 1 and 2 contain the fitness values measured for each color channel separately by applying both Kapur's entropy method and Otsu's class variance method for the four test images.

From Figs. 7, 8, 9, 10, 11 and Figs. 14, 15, 16 are the clustered Images after applying all algorithms for different

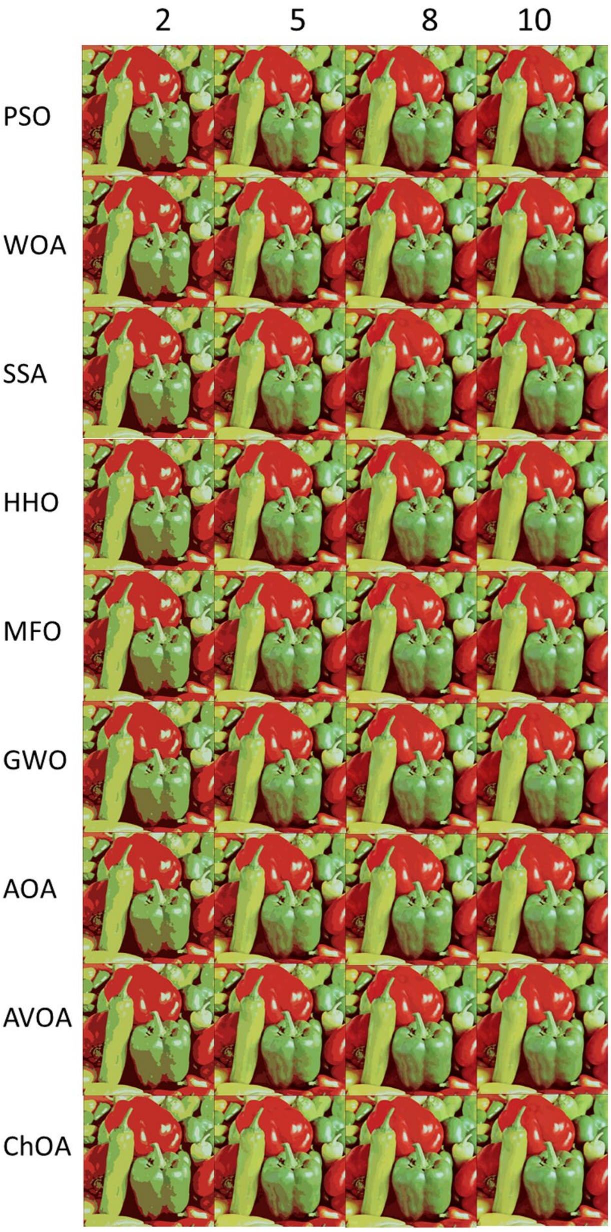


Fig. 7 Kapur's thresholding of image 1

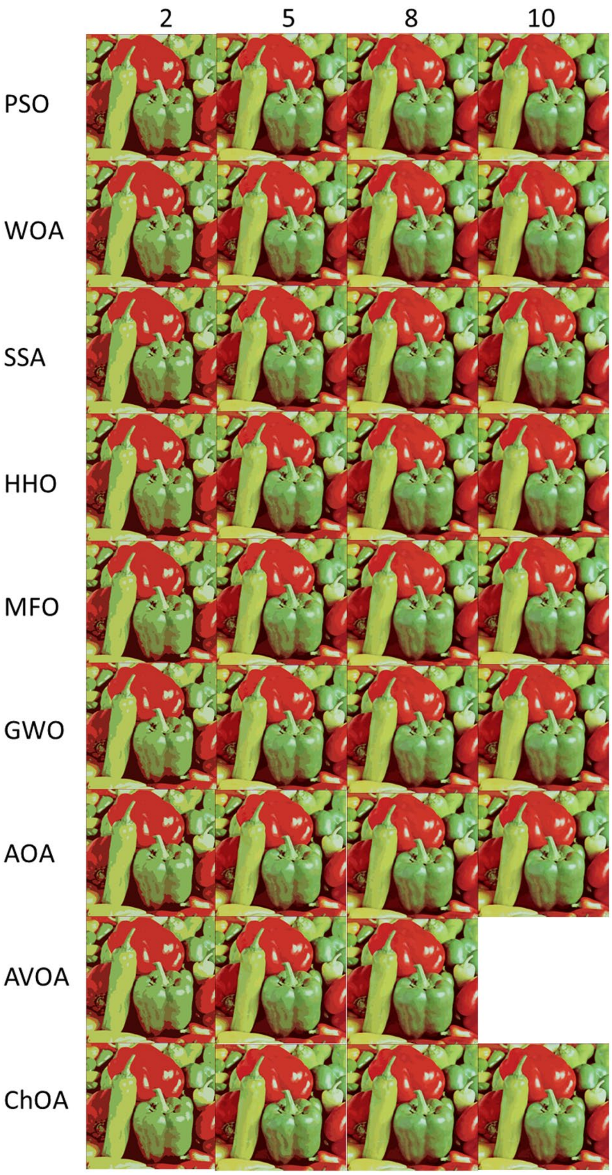


Fig. 8 Otsu's thresholding of image 1

threshold values, for both Kapur's and Otsu's method. The results signify the effectiveness of ChOA in successful multilevel thresholding and image clustering. It is also evident as expected that the clustered image quality gradually improves as the number of thresholds (k) increases. Similar strategy was repeated for 8 other metaheuristic optimization algorithms: Particle Swarm Optimization Algorithm (PSO) (Kennedy and Eberhart 1995), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis 2016), Salp Swarm Algorithm (Mirjalili et al. 2017), Harris Hawks Optimization Algorithm (HHO) (Heidari et al. 2019), Moth Flame Optimization Algorithm (MFO) (Mirjalili 2015), Grey Wolf Optimization Algorithm (GWO) (Mirjalili 2014), Archimedes Optimization Algorithm (AOA) (Hashim et al. 2021), African Vulture Optimization Algorithm (AVOA) (Abdollahzadeh et al. 2021).

Figures 12 and 13 represent the whole research flow. In Fig. 13, randomly initialized search agents in ascending order is a crucial part of the MATLAB code. Kapur's threshold values are mathematically defined in ascending order and the lower values of pixel below a particular threshold are rounded up to that threshold. The optimization algorithm doesn't inherently generate search solution set in an

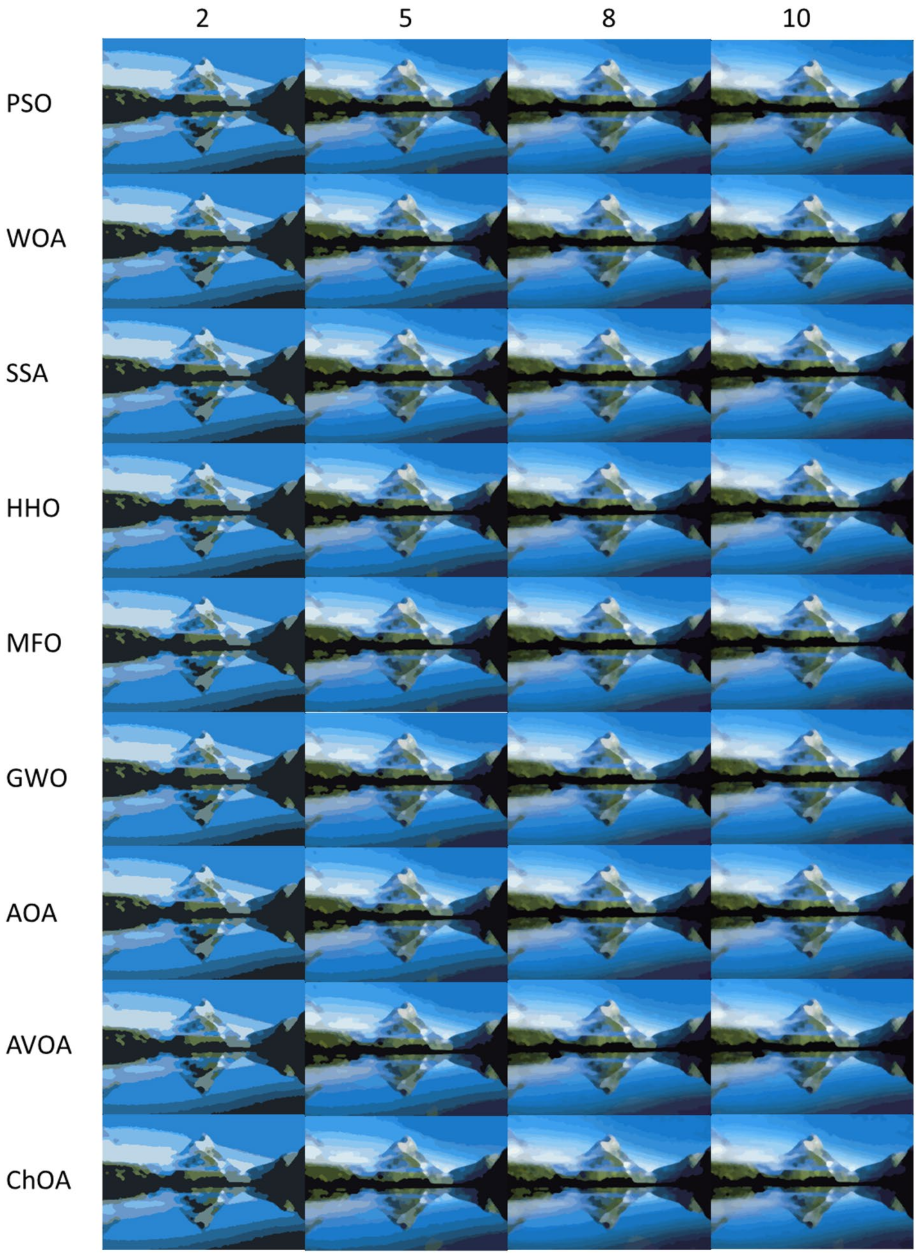


Fig. 9 Kapur's thresholding of image 2

ascending order. So, without this particular part of code the algorithm, doesn't converge to any meaningful solution, but with this part, the optimization algorithm stores the fitness value of ascending ordered threshold before first iteration; any fitness value that doesn't maintain this ascending order will give Zero fitness value. Since the optimization algorithm stores the fitness value of ascending ordered threshold, the algorithm will see any descending order threshold value as a solution set as unworthy and update the next set of solution set so that it doesn't generate other than ascending order solution set.

7.2 Benchmark images

Four test images were taken from Berkeley Segmentation Data Set 500 (BSDS500). Chimp Optimization Algorithm was applied on these images to generate clustered images by means of both Kapur's entropy method and Otsu's class variance method for different number of thresholds. The performance of ChOA was evaluated by computing some of the performance parameters. From Figs. 3, 4 and 5 represent the test RGB images.

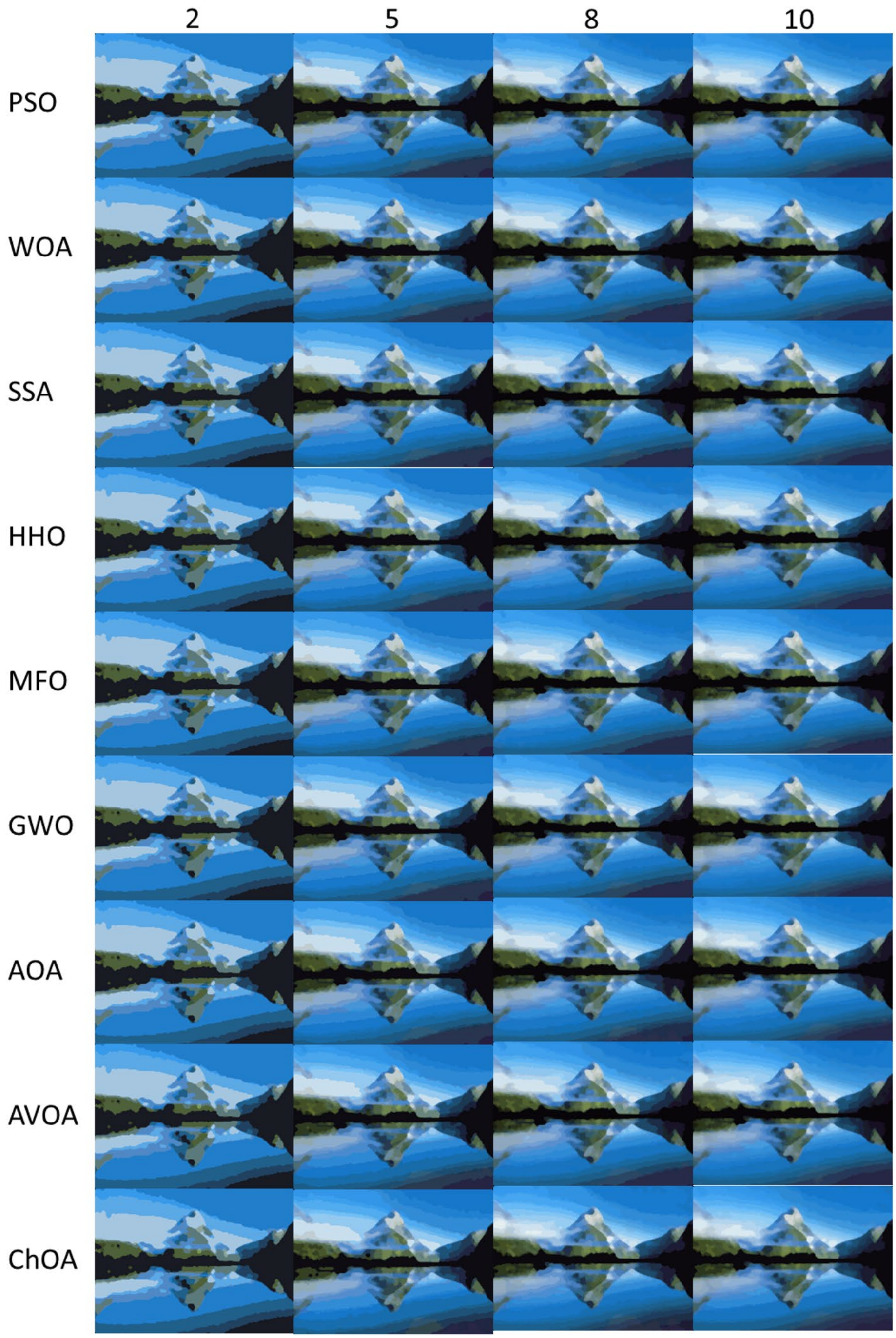


Fig. 10 Otsu's thresholding of Image 2

7.3 Experimental environment

For conducting our experiments, we used MATLAB r2020a. These were carried out in Windows 10-64 bit environment and Core i5 2.20 GHz CPU was used with 8GB RAM. Each experiment was run 30 times for each single case of threshold value and the average value of those 30 values was taken for every parameter calculation.

7.4 Image clustering

The primary parameters in image clustering are F, F', Q. Table 3 covers F values, and it is apparent that ChOA performed no less than other algorithms. Moreover, F values from using Kapur's method is better than F values from using Otsu's method. Nonetheless, in both of the method ChOA performance have significance in overall application.

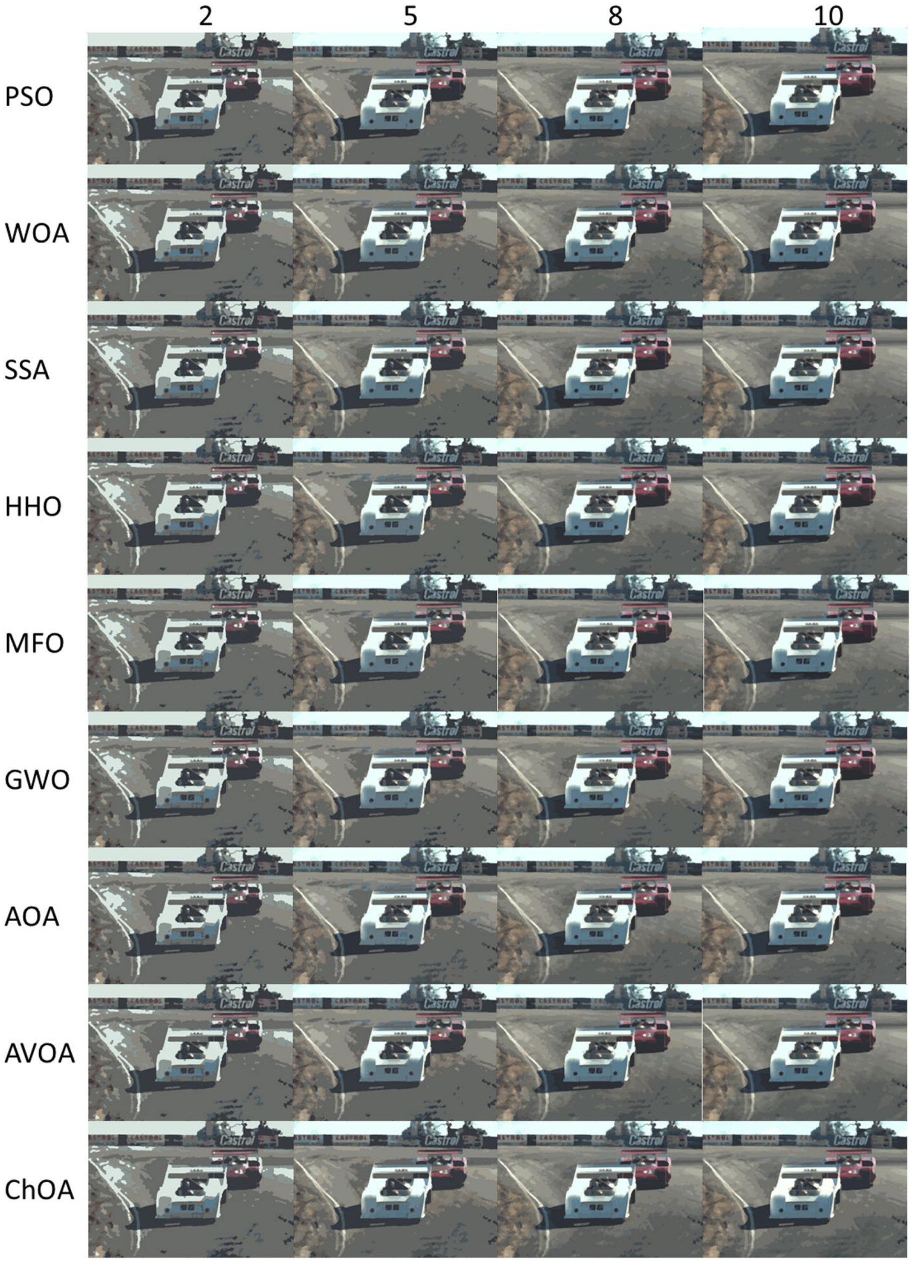


Fig. 11 Kapur's thresholding of Image 3

In Table 3, using Kapur's method for image 1, in case of $k = 5$, it performed the best.

Using Otsu's method, ChOA performed relatively better when $k = 5$ and $k = 8$, and ChOA was in second best position when $k = 10$. In case of F' value, apparent from Table 4, ChOA gave the best and the second best when $k = 8$ and $k = 5$ respectively by Kapur's method, and ChOA performed best when $k = 8$ and performed second best when $k = 5$ using Otsu's method. In case of other k values, ChOA gave competitive results.

When it comes to Q value in image 1, ChOA gave the best and the second best for $k = 8, 10$ respectively for Kapur's method; and by using Otsu's method, for $k = 5, 8, 10$, ChOA gave best result for the first two k values and third best result for the last k value. The values of the parameter Q can be found from Table 5.

Similarly, for F value in image 2, ChOA performed best when $k = 5$ using Kapur's method, and when $k = 5$ and $k = 8$ using Otsu's method. ChOA gave the best F' value when $k = 5$ using Kapur's method, and gave the best and second best F' value when $k = 5, 8$ respectively using Otsu's method. Performance in Q values were similar to that of F' values with slight exceptions: ChOA gave best when $k = 5, 8$ using Otsu's method.

In case of image 3, ChOA gave the best F value using Otsu's method when $k = 2, 8, 10$ and gave the third best

Fig. 12 Program workflow

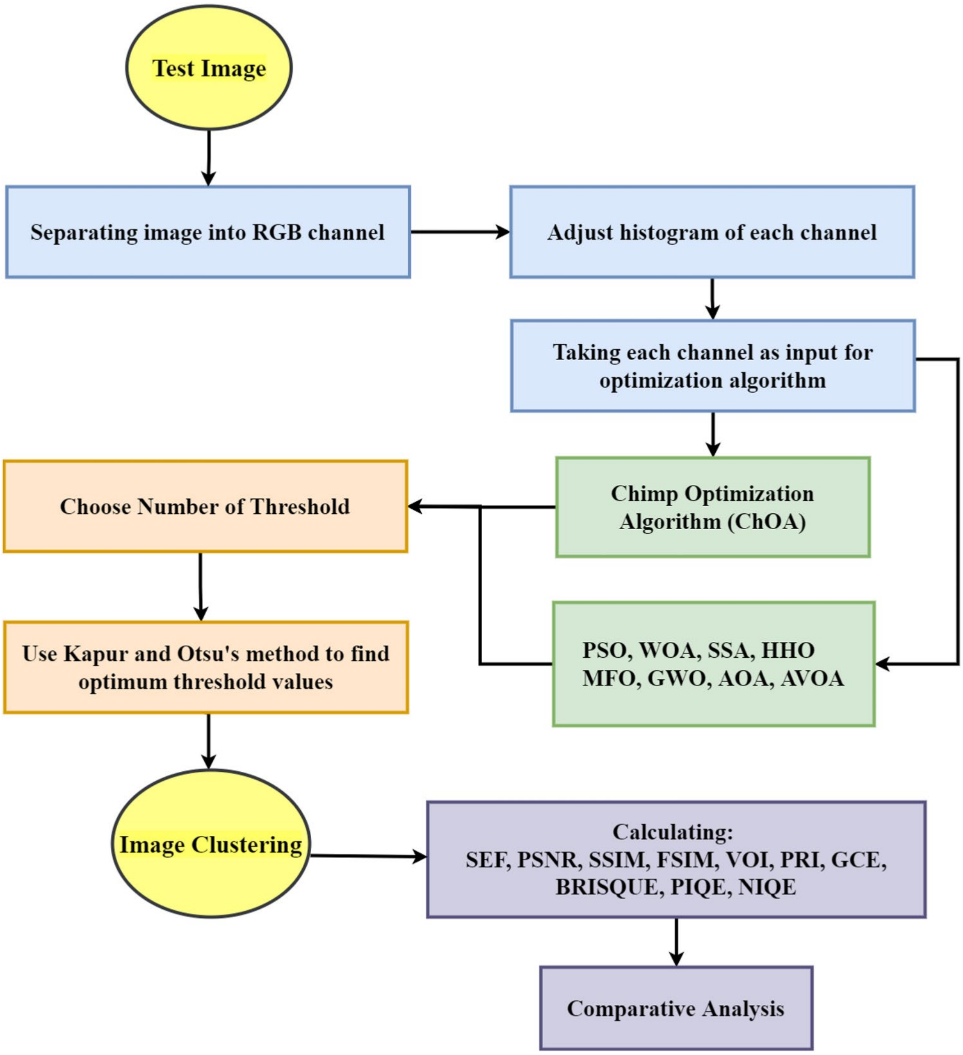
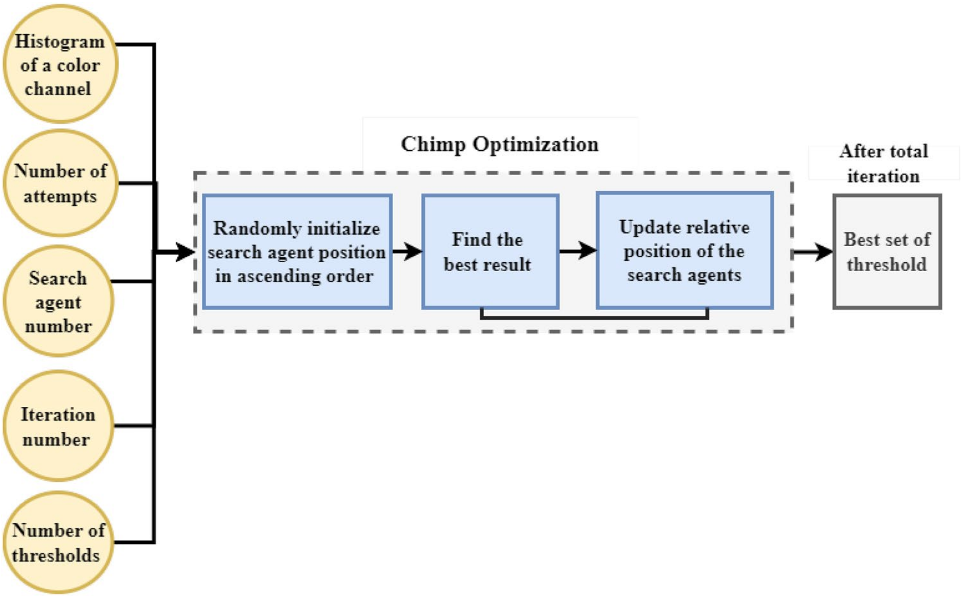

Test Image
Separating image into RGB channel
Adjust histogram of each channel
Taking each channel as input for optimization algorithm
Chimp Optimization Algorithm (ChOA)
PSO, WOA, SSA, HHO MFO, GWO, AOA, AVOA
Choose Number of Threshold
Use Kapur and Otsu's method to find optimum threshold values
Image Clustering
Calculating: SEF, PSNR, SSIM, FSIM, VOI, PRI, GCE, BRISQUE, PIQE, NIQE
Comparative Analysis

Fig. 13 Image thresholding workflow



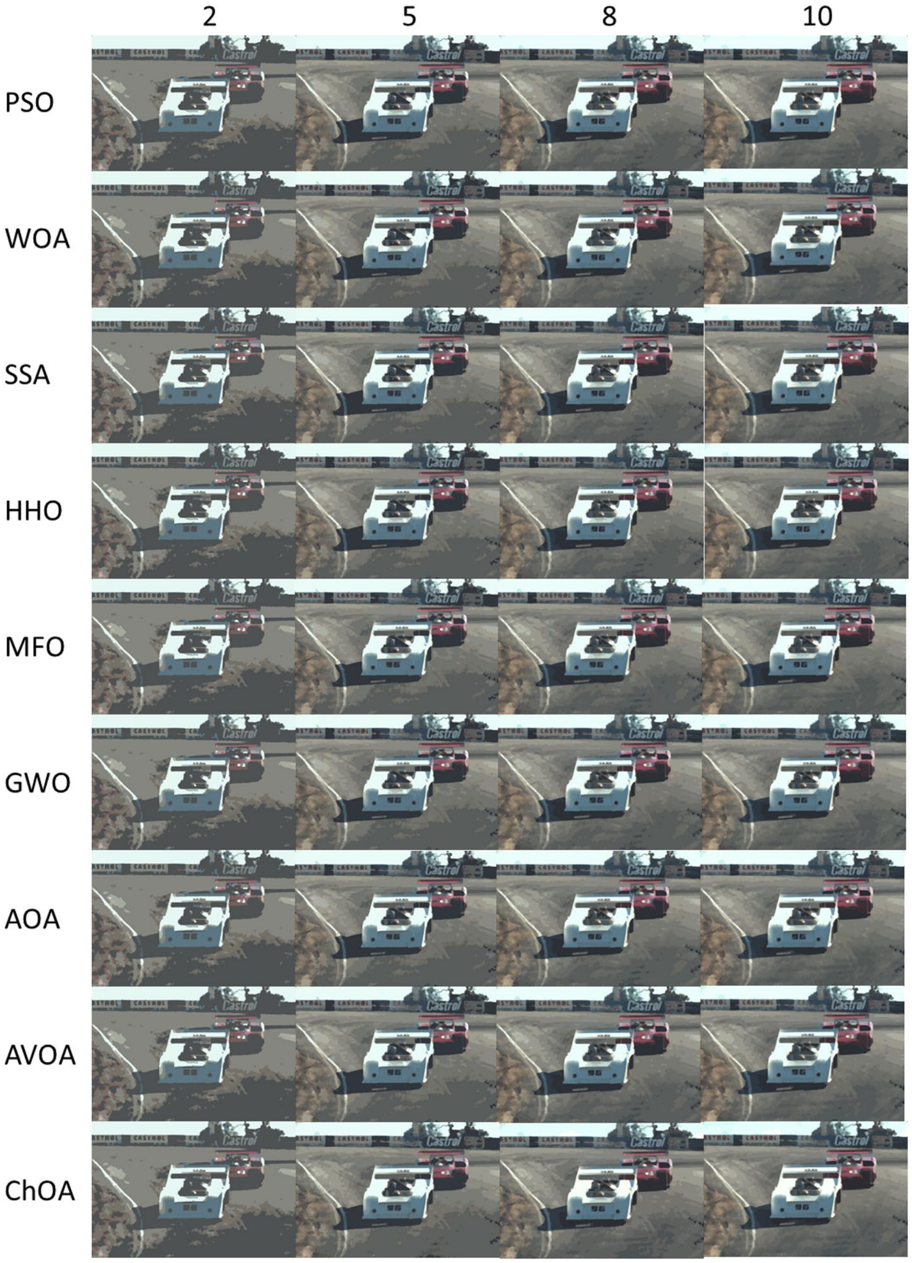


Fig. 14 Otsu's thresholding of Image 3

result when$k = 5, 10$ using Kapur's method. For F' Value, ChOA gave the third best result when $k = 5$ by using Kapur's method, and gave the best result using Otsu's method when $k = 2, 8, 10$. As for Q value using Kapur's method, ChOA gave the best value when number of threshold were 2,10 and the second best result when the number of threshold was 5. Additionally, using Otsu's method, ChOA gave the best values for $k = 2, 8, 10$.

In case of image 4, ChOA gave the best F value when $k = 5, 8, 10$ using Kapur's method and gave the best result when $k = 2, 8, 10$ using Otsu's method. For F' Value, ChOA gave the second and third best result using Kapur's method when $k = 5, 10$ respectively, and gave the best result using Otsu's method when $k = 2, 8$ and the second best results when $k = 5, 10$. As for Q value using Kapur's method, ChOA gave the best value when number of threshold was 5 and the third best result when the number of threshold were 8,10. Additionally, using Otsu's method, ChOA gave the best when $k = 2, 8$, and second best when $k = 10$.

Table 6 represents the number of clusters generated by each algorithm for each of the threshold values, both for Kapur's entropy method. It is to be mentioned that the maximum number of generated sub-cubes or clusters is determined by (11). But generally the generated clusters are lower in number since some spaces in the 3D plane are kept vacant and some are filled. ChOA tends to generate lower number

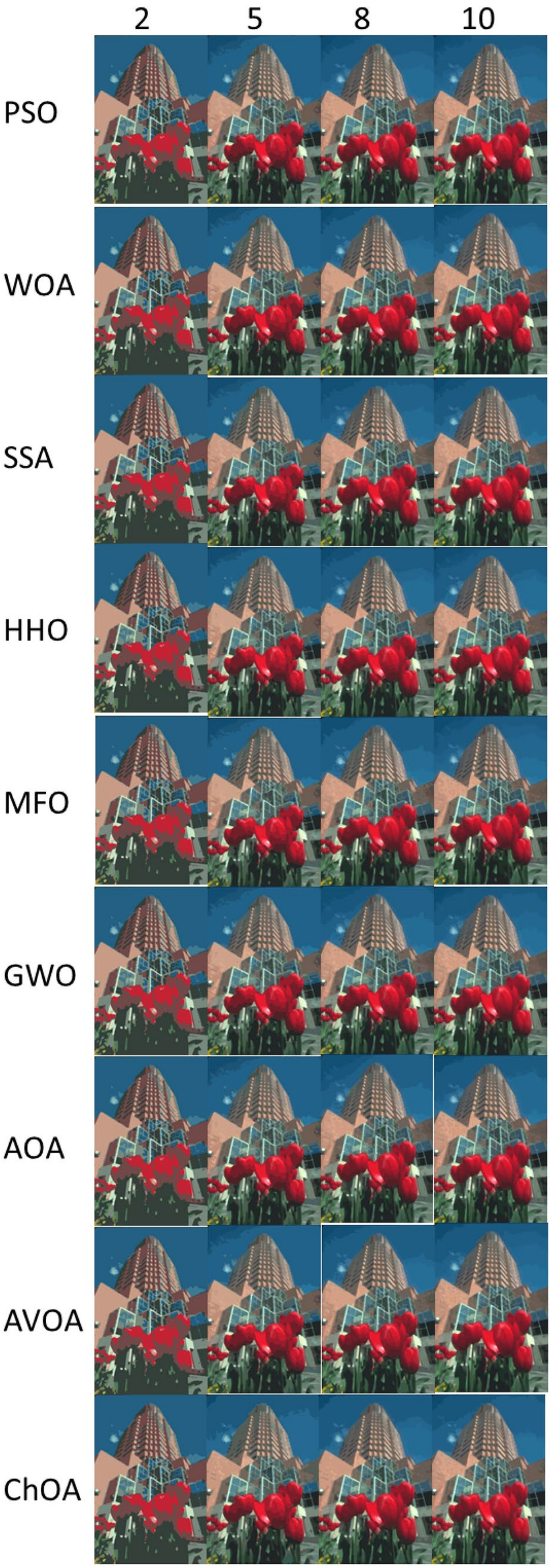


Fig. 15 Kapur's thresholding of Image 4

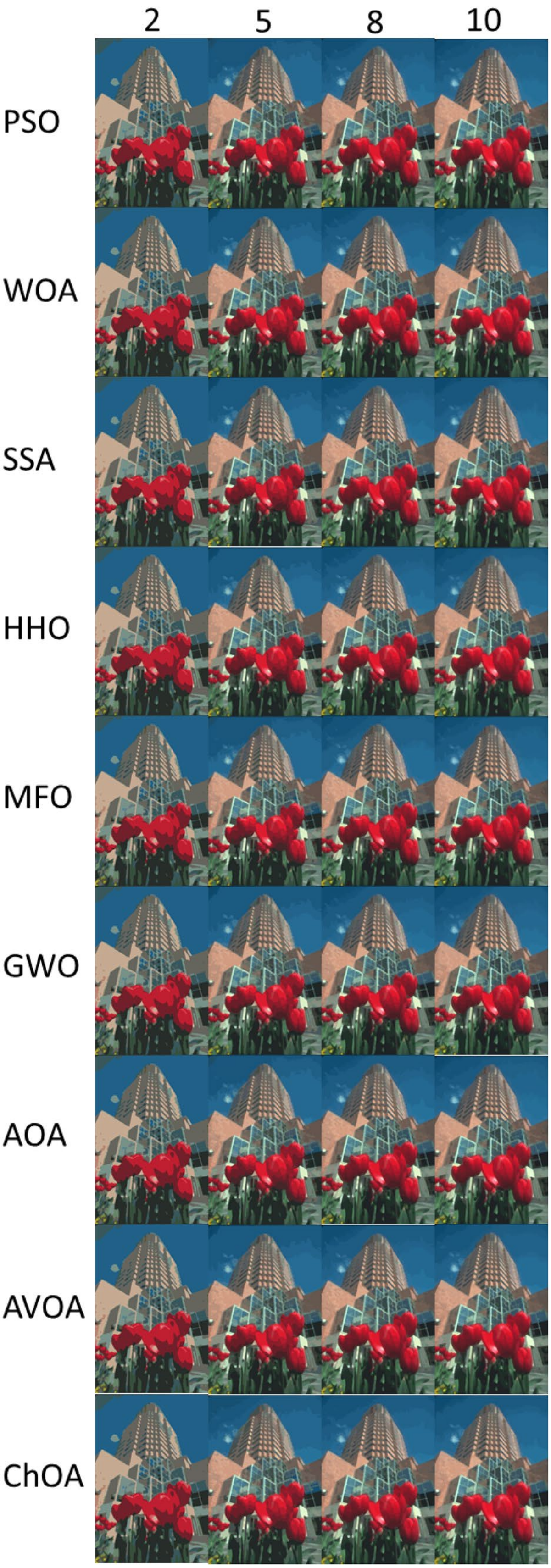
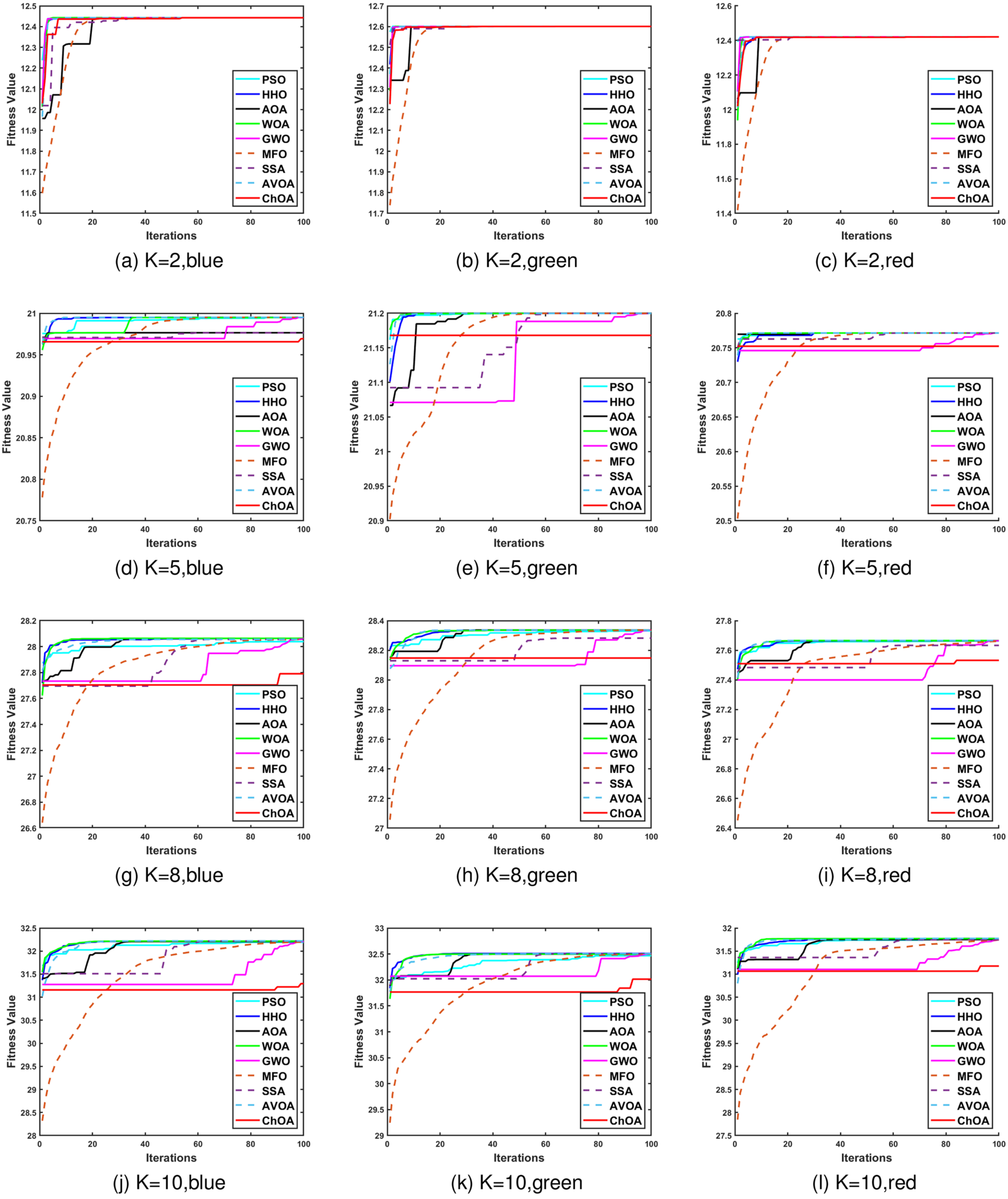


Fig. 16 Otsu's thresholding of Image 4



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

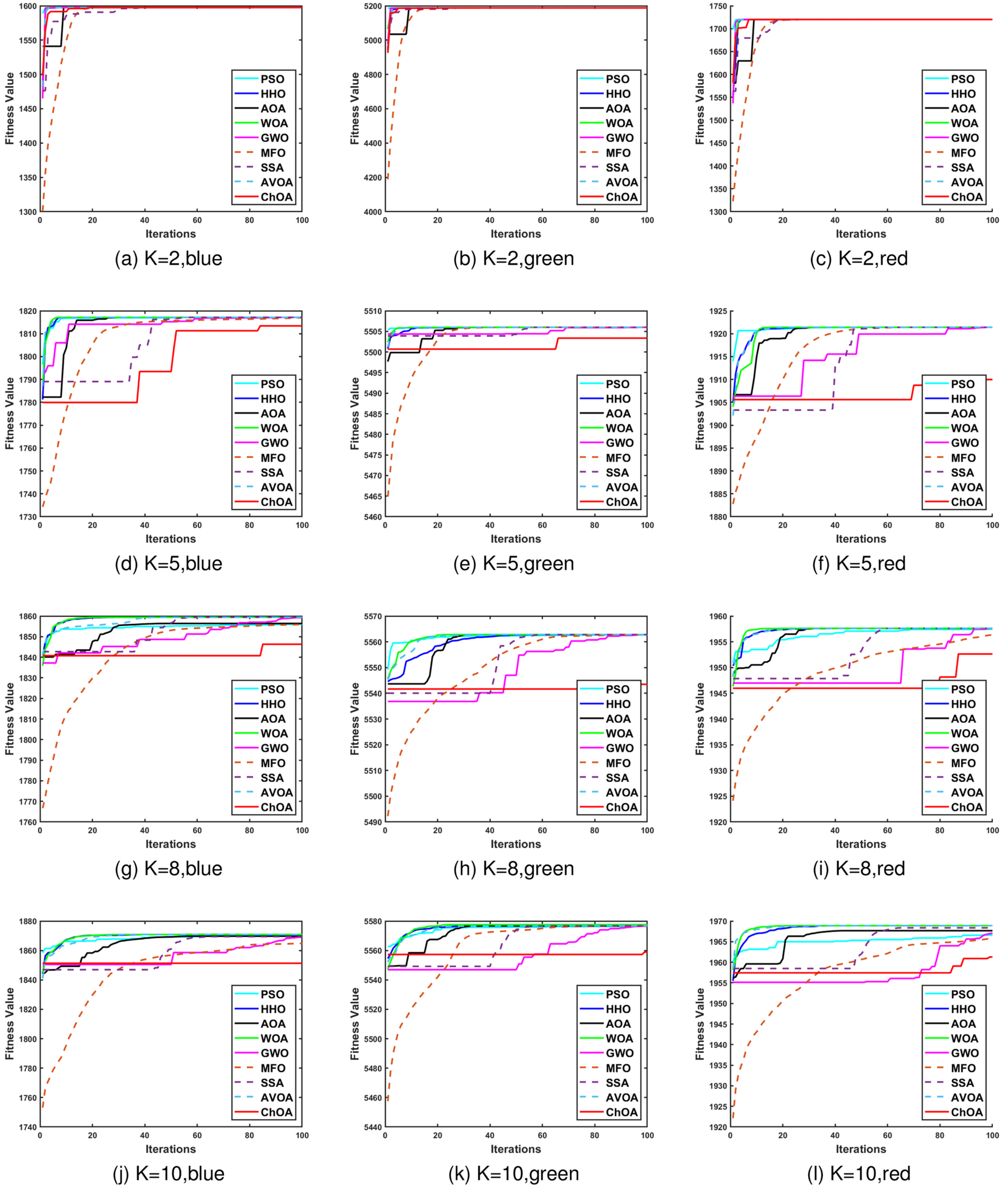
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 17 Converge curve solving Kapur's method for image 1



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

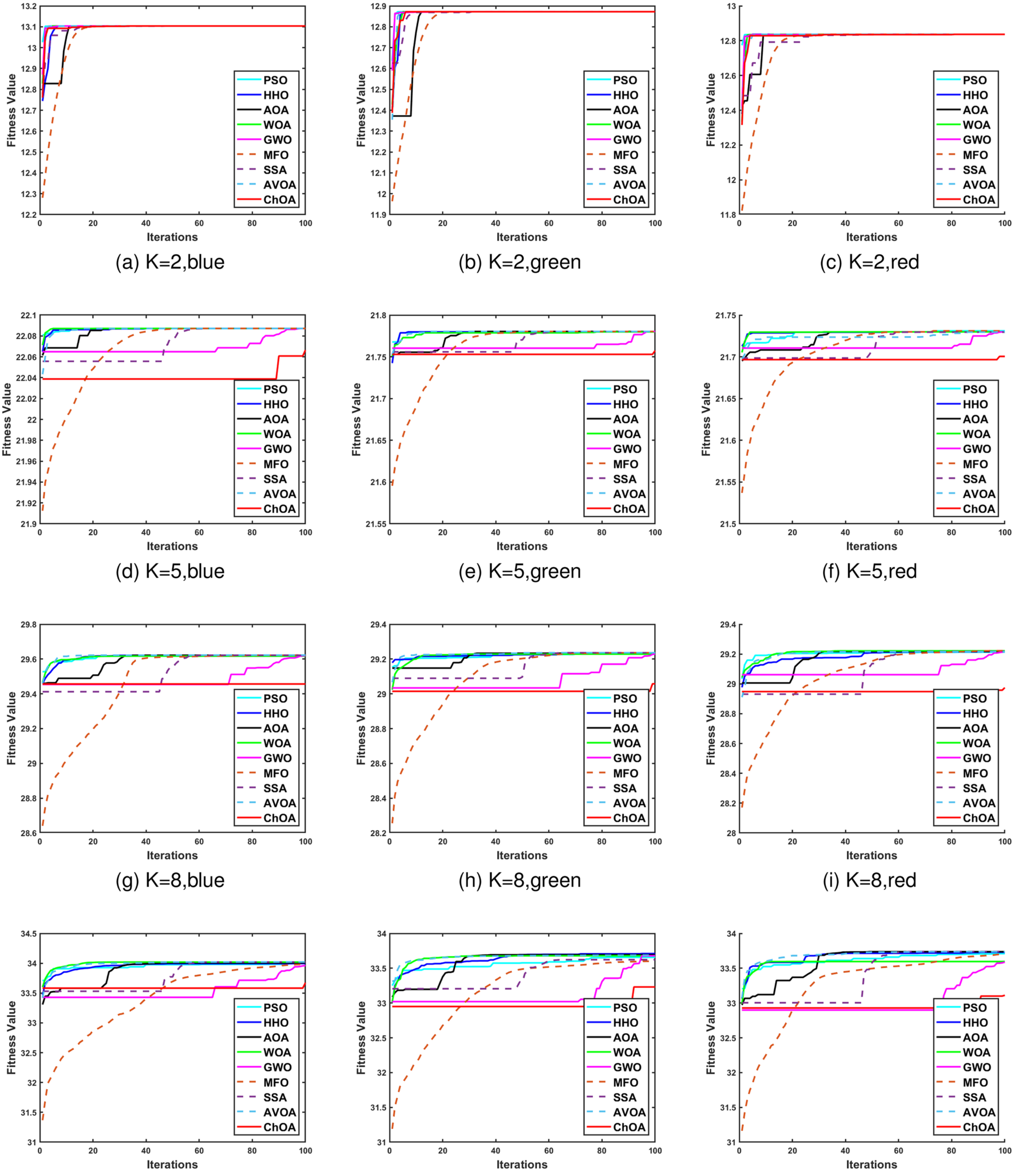
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 18 Converge curve solving Otsu's method for image 1



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

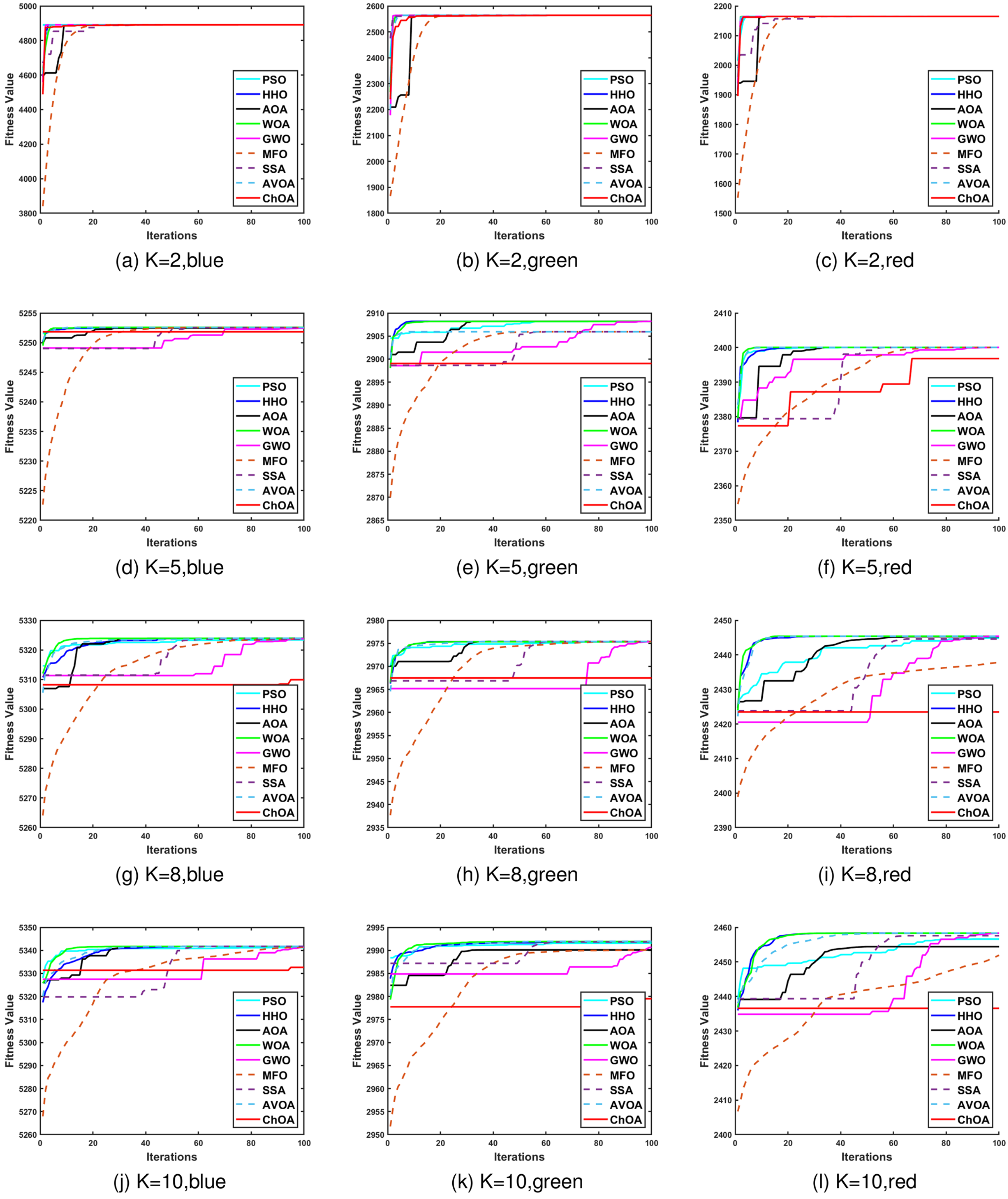
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 19 Converge curve solving Kapur's method for image 2



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

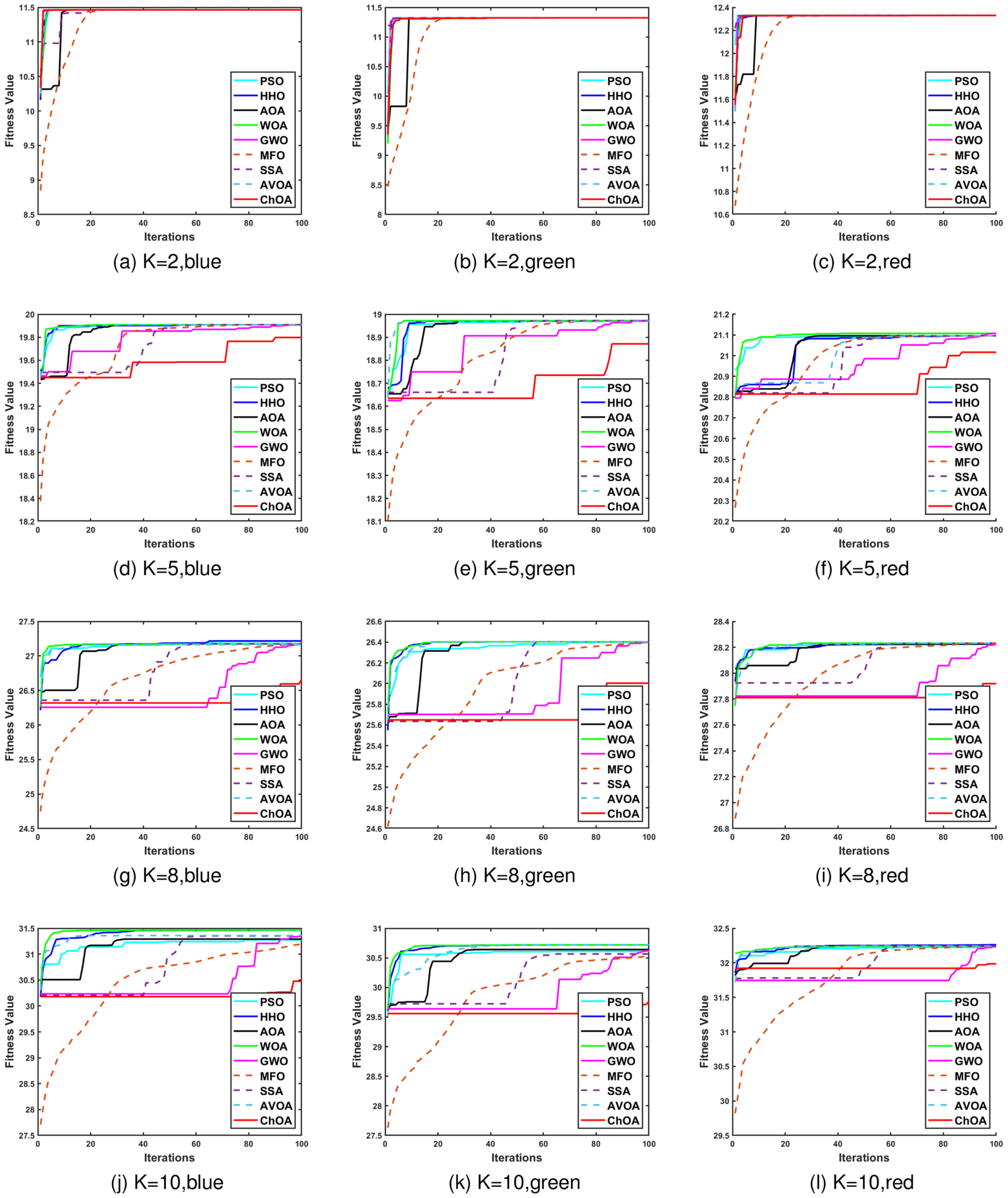
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 20 Converge curve solving Otsu's method for image 2



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

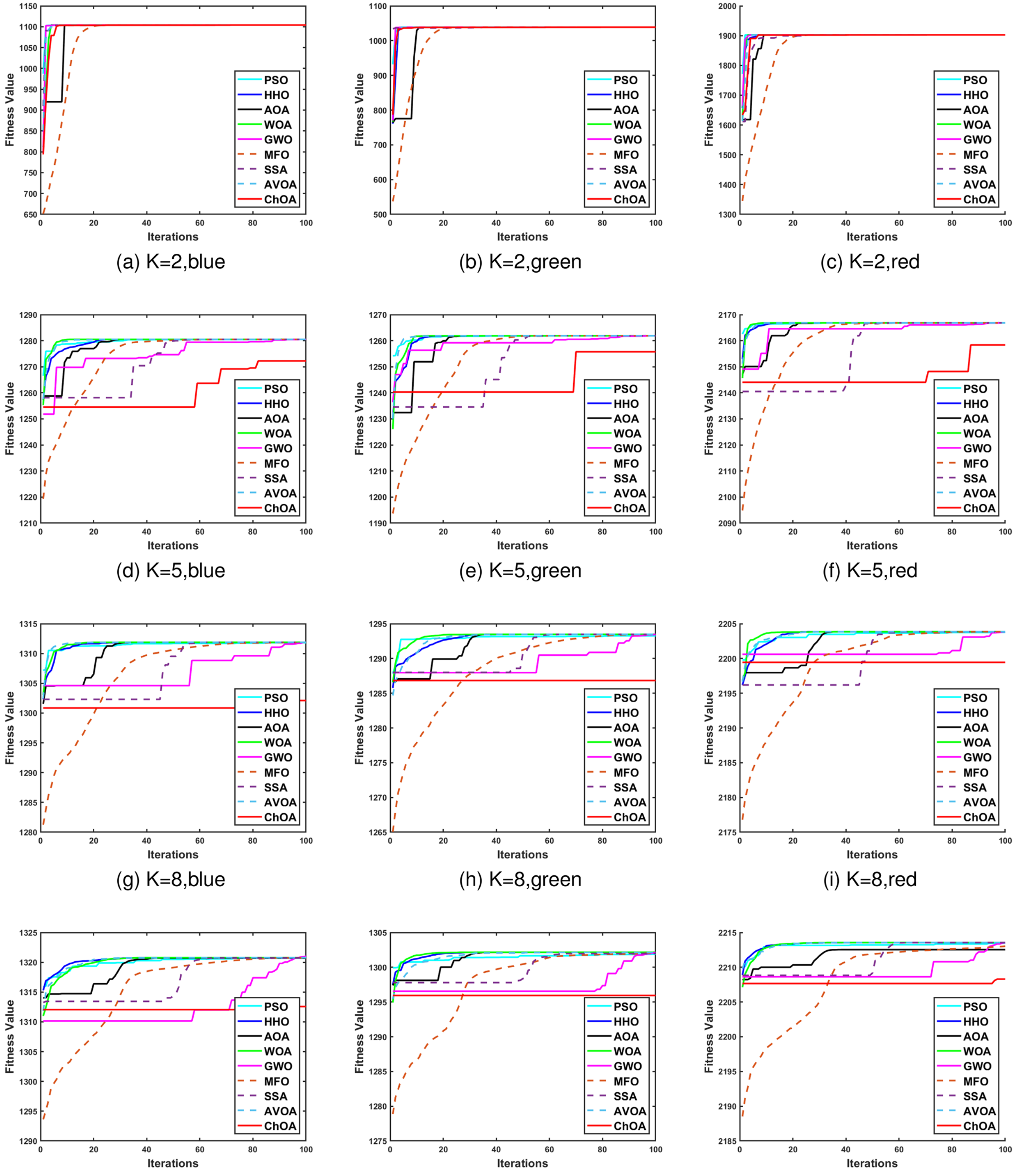
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 21 Converge curve solving Kapur's method for image 3



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

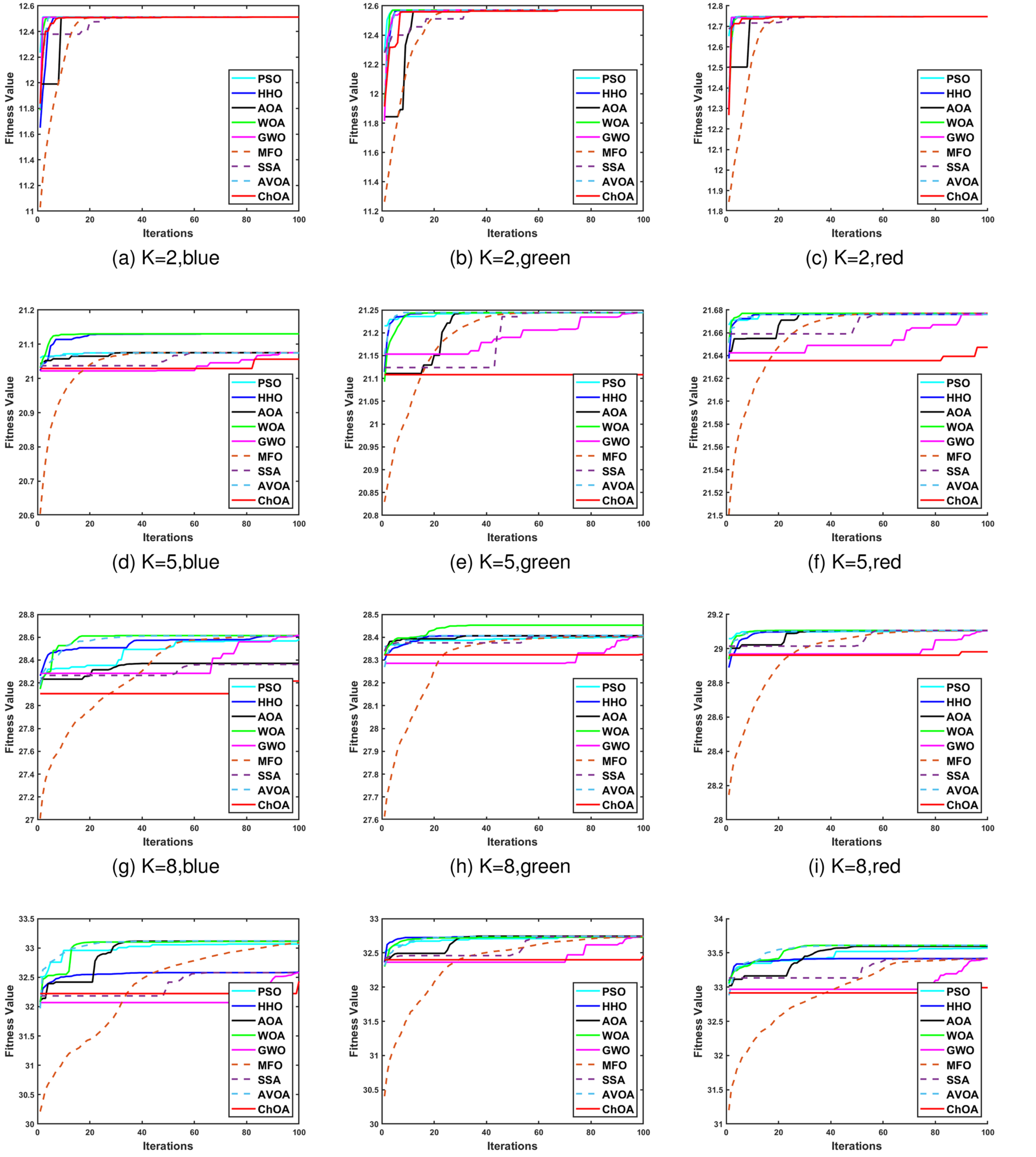
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 22 Converge curve solving Otsu's method for image 3



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

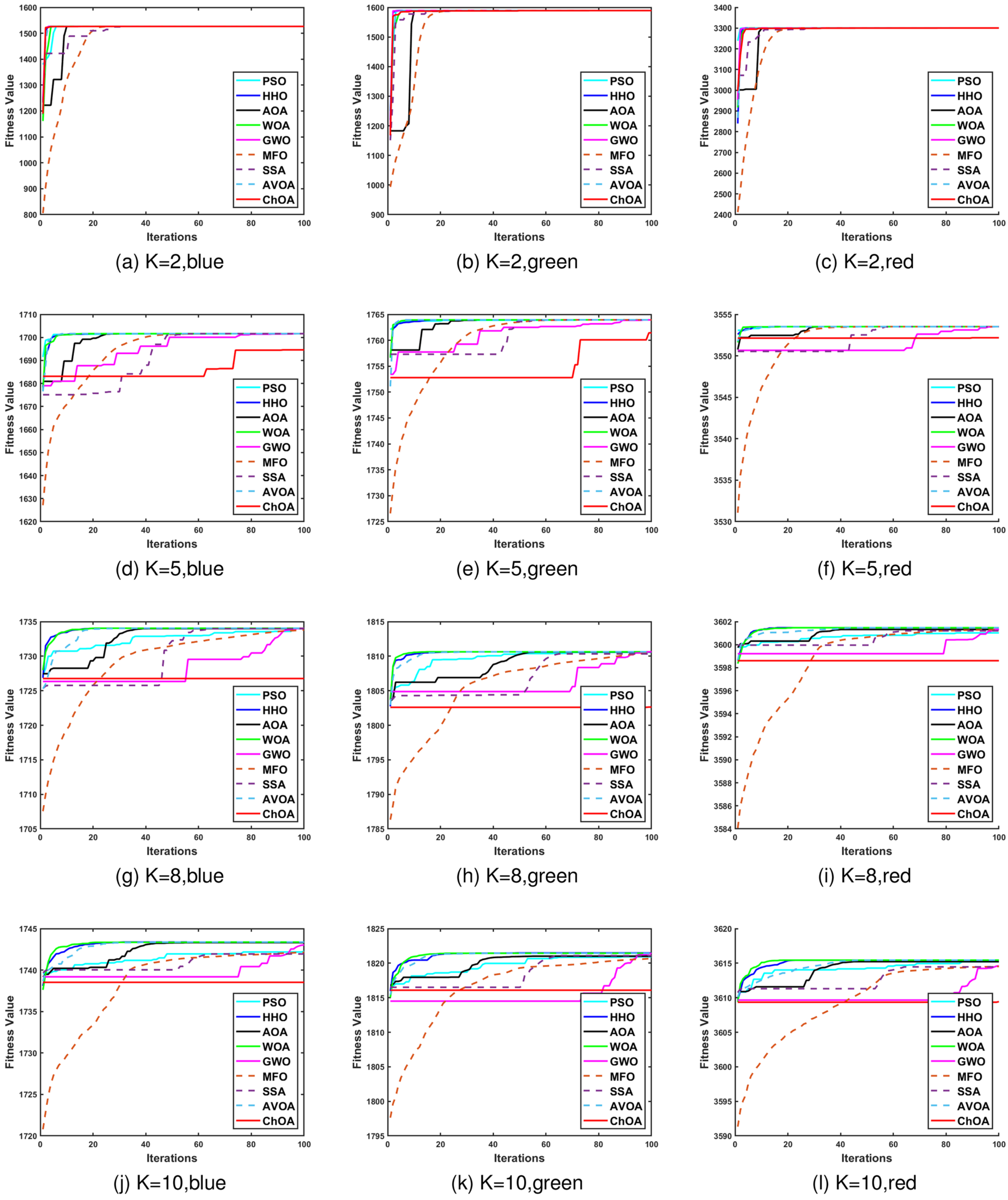
(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 23 Converge curve solving Kapur's method for image 4



(a) K=2,blue

(b) K=2,green

(c) K=2,red

(d) K=5,blue

(e) K=5,green

(f) K=5,red

(g) K=8,blue

(h) K=8,green

(i) K=8,red

(j) K=10,blue

(k) K=10,green

(l) K=10,red

Fig. 24 Converge curve solving Otsu’s method for image 4

Table 1 Fitness Values of Image 1 and Image 2

		K	Color	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	kapur	2	R	12.372857	12.372857	12.372857	12.372857	12.372857	12.372857	12.372857	12.372857	12.372754
			G	12.421712	12.421712	12.421712	12.421712	12.421712	12.421712	12.421712	12.421712	12.421395
			B	12.593963	12.593963	12.593963	12.593963	12.593963	12.593963	12.593963	12.593963	12.593887
		5	R	20.666193	20.666160	20.666193	20.666193	20.666193	20.666162	20.666193	20.666193	20.643232
			G	20.934369	20.938162	20.927835	20.928681	20.935612	20.939279	20.928788	20.9289432	20.911856
			B	21.176528	21.176385	21.176410	21.176196	21.176597	21.176538	21.176597	21.172512	21.117204
		8	R	27.474067	27.486959	27.465192	27.484303	27.467813	27.481512	27.481728	27.461675	27.288891
			G	27.969456	27.9666203	27.970492	27.964796	27.973122	27.966387	27.968874	27.963489	27.678408
			B	28.276833	28.289863	28.259285	28.289836	28.278877	28.287689	28.277100	28.290128	28.096257
		10	R	31.503162	31.524930	31.465783	31.521176	31.487059	31.509512	31.523769	31.505456	30.938372
			G	32.095035	32.111117	32.097204	32.112612	32.085780	32.085206	32.085769	32.0910865	31.382341
			B	32.419176	32.447299	32.393470	32.447251	32.416763	32.433842	32.441883	32.445479	31.876969
	otsu	2	R	1688.6068	1688.6068	1688.6068	1688.6068	1688.6068	1688.6068	1688.6068	1688.6068	1688.6040
			G	1564.1468	1564.1468	1564.1468	1564.14686	1564.14686	1564.1468	1564.1468	1564.1468	1564.1431
			B	5163.0786	5163.0786	5163.0786	5163.0786	5163.0786	5163.0786	5163.0786	5163.0786	5163.0724
		5	R	1886.0937	1886.0975	1886.0976	1886.0976	1886.0976	1886.0915	1886.0915	1886.0976	1877.6782
			G	1778.9139	1778.9164	1778.9157	1778.9164	1778.8789	1778.9164	1778.9164	1778.9164	1775.4114
			B	5478.3788	5478.3747	5478.3788	5478.3788	5478.3788	5478.3781	5478.3788	5478.3788	5474.7757
		8	R	1921.2703	1921.4733	1921.3843	1921.4792	1920.5207	1921.3944	1921.4604	1921.4810	1913.1397
			G	1817.5782	1819.7774	1818.7684	1820.0816	1816.9701	1819.3842	1818.6199	1818.2624	1800.7978
			B	5534.9632	5535.0690	5534.8814	5535.0774	5535.0317	5535.0397	5535.0472	5535.0773	5516.3147
		10	R	1930.8586	1932.6789	1931.0013	1932.6754	1928.8831	1930.4847	1932.0463	NaN	1922.1757
			G	21830.3434	1831.9524	1830.2871	1831.9658	1826.2210	1831.7611	1829.9380	NaN	1809.3954
			B	5548.7543	5549.3642	5549.0851	5549.1045	5548.5593	5548.9177	5548.7225	NaN	5525.2978
Image 2	kapur	2	R	12.769517	12.769517	12.769517	12.769517	12.769517	12.769517	12.769517	12.769517	12.769115
			G	13.056162	13.056162	13.056162	13.056162	13.056162	13.056162	13.056162	13.056162	13.056034
			B	12.813901	12.813901	12.813901	12.813901	12.813901	12.813901	12.813901	12.813901	12.813655
		5	R	21.545681	21.545115	21.545476	21.545342	21.545681	21.545162	21.545550	21.545448	21.508079
			G	21.979204	21.980376	21.973448	21.980349	21.980200	21.980315	21.979435	21.980036	21.954498
			B	21.615313	21.614305	21.6120875	21.613021	21.615179	21.613390	21.614106	21.612996	21.588597
		8	R	28.790121	28.797747	28.772019	28.795719	28.800917	28.791314	28.777562	28.778000	28.616082
			G	29.477244	29.477643	29.478742	29.478411	29.479473	29.475401	29.479927	29.479973	29.274375
			B	29.000661	28.998212	29.001691	29.001747	29.003544	28.997842	29.002927	29.001982	28.825251

Table 1 (continued)

	K	Color	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
	10	R	33.215121	33.158651	33.185556	33.189398	33.212201	33.109765	33.182616	33.191660	32.596906
		G	33.810679	33.829308	33.816052	33.832864	33.807115	33.798933	33.808557	33.834285	33.350285
		B	33.324712	33.336083	33.332417	33.341131	33.329619	33.272821	33.301099	33.311236	32.908248
otsu	2	R	2125.9467	2125.9467	2125.9467	2125.9467	2125.9467	2125.9467	2125.9467	2125.9467	2125.9326
		G	4872.6117	4872.6117	4872.6117	4872.6117	4872.6117	4872.6117	4872.6117	4872.6117	4872.6083
		B	2545.3990	2545.3990	2545.3990	2545.3990	2545.3990	2545.3990	2545.3990	2545.3990	2545.3876
	5	R	2358.0556	2358.0605	2358.0653	2358.0653	2358.0653	2358.0653	2358.0633	2358.0653	2350.7479
		G	5235.3007	5235.3022	5235.3028	5235.3028	5235.2993	5235.2990	5235.3028	5235.3028	5232.3926
		B	2886.5770	2886.8465	2886.7221	2886.8519	2886.2012	2886.8506	2886.5912	2886.7221	2878.8699
	8	R	2400.5133	2401.0892	2400.6929	2401.1075	2396.4526	2401.1010	2400.7760	2401.0872	2382.6225
		G	5305.0591	5305.5828	5305.3512	5305.5993	5305.0611	5305.4588	5305.5667	5304.9028	5292.0115
		B	2952.7949	2952.8199	2952.8482	2952.8503	2952.8455	2952.7965	2952.6386	2952.8550	2944.9897
	10	R	2413.3195	2414.3844	2413.0339	2414.4026	2410.0243	2414.3864	2413.1526	2414.4272	2392.8014
		G	5322.6833	5323.2472	5322.9468	5323.2486	5322.5828	5322.8612	5323.2275	5323.2621	5305.1676
		B	2968.2615	2968.9581	2968.3933	2968.9885	2968.3039	2968.1777	2967.9752	2968.9109	2960.8309

Table 2 Fitness values of Image 3 and Image 4

		K	Color	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 3	kapur	2	R	12.291294	12.291294	12.291294	12.291294	12.291294	12.291294	12.291294	12.291294	12.290830
			G	11.403050	11.403050	11.403050	11.403050	11.403050	11.403050	11.403050	11.403050	11.402998
			B	11.281663	11.281663	11.281663	11.281663	11.281663	11.281663	11.281663	11.281663	11.281584
		5	R	21.031903	21.030366	21.025004	21.003236	21.034637	21.042328	20.878209	20.956416	20.886269
			G	19.751055	19.794187	19.796310	19.793889	19.797222	19.796574	19.796615	19.796748	19.626686
			B	18.729892	18.887606	18.888730	18.887239	18.878486	18.888530	18.775965	18.887909	18.700802
		8	R	28.055990	28.054417	28.058352	28.055487	28.061566	28.059131	28.061638	28.048979	27.760383
			G	27.001710	27.017876	27.003858	27.0155210	27.005196	27.0119786	27.021176	27.001105	26.356683
			B	26.092771	26.241450	26.143653	26.240878	26.174391	26.238050	26.227804	26.236016	25.695290
		10	R	32.057228	32.067446	32.062962	32.070388	32.060278	32.056516	32.074887	32.075519	31.739260
			G	31.057274	31.237465	30.952461	31.190502	31.032936	31.092619	31.103508	31.207569	30.400521
			B	30.315514	30.551967	30.304227	30.547003	30.290965	30.456497	30.454080	30.561148	29.741036
	otsu	2	R	1848.8312	1848.8312	1848.8312	1848.8312	1848.8312	1848.8312	1848.8312	1848.8312	1848.8246
			G	1047.9308	1047.9308	1047.9308	1047.9308	1047.9308	1047.9308	1047.9308	1047.9308	1047.9100
			B	991.40433	991.40433	991.40433	991.40433	991.40433	991.40433	991.40433	991.40433	991.39895
		5	R	2107.6433	2107.6465	2107.6472	2107.6472	2107.6472	2107.6463	2107.6472	2107.6472	2099.5810
			G	1215.8271	1215.8289	1215.8289	1215.8289	1215.8289	1215.8275	1215.8289	1215.8289	1207.8897
			B	1209.6561	1209.6613	1208.9458	1209.6619	1209.6619	1209.6604	1209.6605	1209.6619	1198.9441
		8	R	2142.9055	2142.9742	2142.9467	2142.9795	2142.9696	2142.9503	2142.9703	2142.9788	2136.2455
			G	1245.7437	1245.7947	1245.6794	1245.7966	1245.7944	1245.7782	1245.6820	1245.7968	1237.4876
			B	1240.7066	1240.7664	1240.7666	1240.7754	1240.7624	1240.7457	1240.7478	1240.7754	1233.1521
		10	R	2151.5508	2151.7445	2151.5807	2151.7516	2151.6248	2151.6560	2151.6117	2151.7570	2146.3745
			G	1254.4264	1254.4488	1254.5314	1254.5626	1254.4166	1254.4525	1254.4607	1254.5588	1247.7149
			B	1248.6223	1248.6957	1248.7244	1248.8211	1248.6846	1248.5334	1248.4866	1248.8328	1243.0356
Image 4	kapur	2	R	12.702121	12.702121	12.702121	12.702121	12.702121	12.702121	12.702121	12.702121	12.701896
			G	12.272001	12.272001	12.272001	12.272001	12.272001	12.272001	12.272001	12.272001	12.271890
			B	12.567272	12.567272	12.567272	12.567272	12.567272	12.567272	12.567272	12.567272	12.565750
		5	R	21.621497	21.621287	21.621418	21.621207	21.621534	21.621208	21.621484	21.621254	21.592733
			G	20.301826	20.300886	20.295287	20.30018	20.301862	20.300753	20.301807	20.3011196	20.225679
			B	21.101447	21.095670	21.095169	21.098768	21.101938	21.102155	21.102100	21.096283	21.027340
		8	R	29.048428	29.050683	29.046852	29.051759	29.051878	29.049261	29.049136	29.051064	28.885892
			G	27.018084	27.019959	27.018414	27.021286	27.023012	27.016488	27.019820	27.020738	26.824234
			B	28.176694	28.215559	28.175845	28.212806	28.178609	28.189916	28.182138	28.188811	28.105777
		10	R	33.406770	33.451707	33.384690	33.371454	33.403845	33.337967	33.340673	33.392079	32.922182
			G	30.706307	30.768322	30.693262	30.760078	30.709459	30.722779	30.687167	30.730489	30.392419
			B	32.406350	32.404275	32.382758	32.399444	32.414234	32.390566	32.412764	32.409662	31.996483
	otsu	2	R	3154.5198	3154.5198	3154.5198	3154.5198	3154.5198	3154.5198	3154.5198	3154.5198	3154.4998
			G	1446.2514	1446.2514	1446.2514	1446.2514	1446.2514	1446.2514	1446.2514	1446.2514	1446.2348
			B	1497.7029	1497.7029	1497.7029	1497.7029	1497.7029	1497.7029	1497.7029	1497.7029	1497.7004
		5	R	3408.0065	3408.0065	3408.0065	3408.0065	3408.0065	3408.0047	3408.0043	3408.0065	3406.4441
			G	1579.6046	1579.6102	1579.6102	1579.6102	1579.6102	1579.6099	1579.6102	1579.6102	1569.6511
			B	1657.0532	1657.0037	1657.0264	1657.0048	1657.0588	1657.0169	1657.0203	1657.0242	1651.7061
		8	R	3456.3291	3456.4767	3456.3732	3456.4781	3456.3125	3456.4497	3456.4291	3456.4826	3452.6821
			G	1608.2817	1608.6150	1607.9583	1608.6161	1607.8449	1608.6039	1608.4153	1608.6171	1599.8342
			B	1701.7142	1701.9509	1701.7316	1701.9555	1701.5893	1701.9391	1701.8621	1701.9569	1694.5229
		10	R	3468.5617	3469.7650	3468.9095	3469.8057	3468.5095	3468.7043	3468.6594	3469.8060	3464.6132
			G	1616.1023	1616.5897	1614.7736	1616.5972	1616.0020	1616.4179	1616.3880	1616.5904	1610.9169
			B	1712.0581	1712.5404	1712.1088	1712.5434	1711.8068	1712.4504	1712.3493	1712.5299	1707.1319

Table 3 F values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	8.594E-6	8.594E-6	8.594E-6	8.594E-6	8.594E-6	8.594E-6	8.594E-6	8.594E-6	8.946E-6
		5	6.838E-5	6.885E-5	7.185E-5	7.106E-5	6.846E-5	6.832E-5	7.027E-5	7.134E-5	6.773E-5
		8	0.0001628	0.0001589	0.0001624	0.0001612	0.0001603	0.0001575	0.0001621	0.0001659	0.0001726
		10	0.0002549	0.0002626	0.0002639	0.0002651	0.0002574	0.0002562	0.0002564	0.0002679	0.0002618
	Otsu	2	2.821E-5	2.821E-5	2.821E-5	2.821E-5	2.821E-5	2.821E-5	2.821E-5	2.821E-5	3.077E-5
		5	7.914E-5	8.084E-5	8.072E-5	8.083E-5	8.031E-5	8.126E-5	8.092E-5	8.083E-5	7.391E-5
		8	0.0002145	0.0002216	0.0002223	0.0002219	0.0002014	0.0002209	0.0002288	0.0002221	0.0001864
		10	0.0003224	0.0003615	0.0003229	0.0003590	0.0002930	0.0003331	0.0003337	NaN	0.0002865
Image 2	Kapur	2	9.812E-6	9.812E-6	9.812E-6	9.812E-6	9.812E-6	9.812E-6	9.812E-6	9.812E-6	1.022E-5
		5	6.060E-5	6.212E-5	6.040E-5	5.848E-5	5.993E-5	6.242E-5	5.933E-5	5.748E-5	6.660E-5
		8	0.0001784	0.0001759	0.0001831	0.0001790	0.0001812	0.0001746	0.0001785	0.0001810	0.0001595
		10	0.0002777	0.0002720	0.0002726	0.0002703	0.0002778	0.0002710	0.0002632	0.0002665	0.0002852
	Otsu	2	1.703E-5	1.703E-5	1.703E-5	1.703E-5	1.703E-5	1.703E-5	1.703E-5	1.703E-5	1.758E-5
		5	8.314E-5	8.327E-5	8.297E-5	8.302E-5	8.258E-5	8.302E-5	8.296E-5	8.297E-5	7.886E-5
		8	0.0002138	0.0002071	0.0002084	0.0002063	0.0002048	0.0002072	0.0002084	0.0002173	0.0001995
		10	0.0003225	0.0003260	0.0003152	0.0003214	0.0003084	0.0003342	0.0003235	0.0003116	0.0003295
Image 3	Kapur	2	2.331E-6	2.331E-6	2.331E-6	2.331E-6	2.331E-6	2.331E-6	2.331E-6	2.331E-6	2.332E-6
		5	1.212E-5	1.431E-5	1.331E-5	1.366E-5	1.345E-5	1.366E-5	1.169E-5	1.331E-5	1.314E-5
		8	3.114E-5	3.189E-5	3.173E-5	3.258E-5	3.206E-5	3.398E-5	3.323E-5	3.245E-5	3.348E-5
		10	5.896E-5	6.120E-5	5.936E-5	6.088E-5	5.751E-5	5.438E-5	5.509E-5	6.215E-5	5.556E-5
	Otsu	2	3.813E-6	3.813E-6	3.813E-6	3.813E-6	3.813E-6	3.813E-6	3.813E-6	3.813E-6	3.783E-6
		5	1.322E-5	1.323E-5	1.312E-5	1.324E-5	1.324E-5	1.322E-5	1.324E-5	1.324E-5	1.375E-5
		8	4.485E-5	4.266E-5	4.466E-5	4.277E-5	4.405E-5	4.420E-5	4.432E-5	4.300E-5	3.986E-5
		10	7.468E-5	7.202E-5	7.374E-5	7.215E-5	7.313E-5	7.512E-5	7.653E-5	7.183E-5	6.522E-5
Image 4	Kapur	2	1.441E-5	1.441E-5	1.441E-5	1.441E-5	1.441E-5	1.441E-5	1.441E-5	1.441E-5	1.446E-5
		5	6.945E-5	6.612E-5	6.627E-5	6.706E-5	6.925E-5	6.968E-5	6.969E-5	6.536E-5	6.386E-5
		8	0.0001940	0.0001923	0.0001947	0.0001916	0.0001983	0.0001918	0.0001945	0.0001951	0.0001789
		10	0.0003137	0.0002969	0.0003227	0.0002825	0.0003172	0.0002951	0.0003127	0.0003072	0.0002760
	Otsu	2	1.159E-5	1.159E-5	1.159E-5	1.159E-5	1.159E-5	1.159E-5	1.159E-5	1.159E-5	1.150E-5
		5	6.098E-5	7.101E-5	6.678E-5	7.150E-5	5.969E-5	6.597E-5	6.291E-5	6.725E-5	6.973E-5
		8	0.0002204	0.0002256	0.0002202	0.0002252	0.0002241	0.0002258	0.0002272	0.0002251	0.0001944
		10	0.0003645	0.0004136	0.0003377	0.0004180	0.0003563	0.0003790	0.0003646	0.0004073	0.0003220

of clusters comparing to other algorithms for same number of threshold value.

After careful observation to the tabulated results, it becomes apparent that ChOA gave excel performance when image clustering is considered. In these four images, the pixels distribution are not deterministic and the nature of ChOA is stochastic. Taken into consideration that the nature of the pixel distribution is quite random, since ChOA performed relatively well in these four images considering the above clustering parameters, it can be concluded that using ChOA is the better choice in image clustering comparing to other algorithms. ChOA has outperformed all other algorithms in most of the cases in terms of these clustering parameters. It is further evident that ChOA has shown more superior performance in Otsu's class variance method than it had done for Kapur's entropy method, implying that it is more compatible for image clustering approaches based on Otsu's class variance method (Fig. 14).

Table 4 F' Values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	9.179E-7	9.179E-7	9.179E-7	9.179E-7	9.179E-7	9.179E-7	9.179E-7	9.179E-7	9.528E-7
		5	9.582E-6	1.081E-5	8.418E-6	8.543E-6	9.858E-6	1.089E-5	8.377E-6	8.611E-6	1.157E-5
		8	7.175E-5	6.926E-5	7.671E-5	7.100E-5	7.399E-5	6.863E-5	7.040E-5	6.323E-5	6.712E-5
		10	0.0001941	0.0001881	0.0002062	0.0001854	0.0001935	0.0001842	0.0001846	0.0001940	0.0001770
	Otsu	2	2.986E-6	2.986E-6	2.986E-6	2.986E-6	2.986E-6	2.986E-6	2.986E-6	2.986E-6	3.212E-6
		5	1.078E-5	1.147E-5	1.141E-5	1.146E-5	1.115E-5	1.182E-5	1.144E-5	1.146E-5	1.186E-5
		8	8.571E-5	9.167E-5	9.132E-5	0.0001038	7.627E-5	8.116E-5	8.581E-5	8.056E-5	6.477E-5
		10	0.0002471	0.0002232	0.0002320	0.0002363	0.0002370	0.0002830	0.0002393	NaN	0.0002086
Image 2	Kapur	2	9.833E-7	9.833E-7	9.833E-7	9.833E-7	9.833E-7	9.833E-7	9.833E-7	9.833E-7	1.024E-6
		5	6.825E-6	6.951E-6	6.610E-6	6.390E-6	6.703E-6	6.918E-6	6.586E-6	6.313E-6	7.232E-6
		8	2.425E-5	2.384E-5	2.546E-5	2.426E-5	2.622E-5	2.427E-5	2.785E-5	2.508E-5	2.233E-5
		10	5.123E-5	4.888E-5	5.377E-5	5.028E-5	5.405E-5	4.935E-5	4.729E-5	5.197E-5	5.900E-5
	Otsu	2	1.770E-6	1.770E-6	1.770E-6	1.770E-6	1.770E-6	1.770E-6	1.770E-6	1.770E-6	1.867E-6
		5	9.338E-6	9.203E-6	9.066E-6	9.112E-6	9.077E-6	9.256E-6	9.078E-6	9.066E-6	8.579E-6
		8	3.206E-5	3.952E-5	3.370E-5	3.637E-5	3.005E-5	3.798E-5	3.235E-5	3.401E-5	3.082E-5
		10	7.464E-5	7.425E-5	6.706E-5	7.086E-5	6.660E-5	8.497E-5	7.360E-5	6.991E-5	7.862E-5
Image 3	Kapur	2	2.331E-7	2.331E-7	2.331E-7	2.331E-7	2.331E-7	2.331E-7	2.331E-7	2.331E-7	2.332E-7
		5	1.312E-6	1.547E-6	1.458E-6	1.434E-6	1.525E-6	1.538E-6	1.234E-6	1.432E-6	1.368E-6
		8	3.570E-6	3.612E-6	3.746E-6	3.713E-6	3.600E-6	3.828E-6	3.854E-6	3.759E-6	4.012E-6
		10	8.179E-6	7.642E-6	8.391E-6	7.782E-6	7.726E-6	7.478E-6	7.672E-6	7.882E-6	7.810E-6
	Otsu	2	3.813E-7	3.813E-7	3.813E-7	3.813E-7	3.813E-7	3.813E-7	3.813E-7	3.813E-7	3.783E-7
		5	1.369E-6	1.372E-6	1.353E-6	1.365E-6	1.365E-6	1.374E-6	1.367E-6	1.365E-6	1.432E-6
		8	6.077E-6	6.413E-6	6.440E-6	6.290E-6	6.372E-6	6.270E-6	6.333E-6	6.290E-6	5.049E-6
		10	1.248E-5	1.229E-5	1.207E-5	1.256E-5	1.243E-5	1.250E-5	1.225E-5	1.217E-5	1.044E-5
Image 4	Kapur	2	1.441E-6	1.441E-6	1.441E-6	1.441E-6	1.441E-6	1.441E-6	1.441E-6	1.441E-6	1.446E-6
		5	7.480E-6	7.091E-6	7.099E-6	7.274E-6	7.485E-6	7.450E-6	7.546E-6	6.972E-6	7.066E-6
		8	3.459E-5	4.276E-5	3.033E-5	3.945E-5	3.369E-5	3.649E-5	3.201E-5	3.457E-5	3.603E-5
		10	0.0001097	8.978E-5	0.0001030	9.266E-5	0.0001037	0.0001005	0.0001014	0.0001003	9.412E-5
	Otsu	2	1.159E-6	1.159E-6	1.159E-6	1.159E-6	1.159E-6	1.159E-6	1.159E-6	1.159E-6	1.152E-6
		5	8.242E-6	8.529E-6	8.399E-6	8.545E-6	8.181E-6	8.367E-6	8.228E-6	8.414E-6	8.209E-6
		8	4.466E-5	3.997E-5	4.194E-5	4.043E-5	4.501E-5	4.132E-5	4.574E-5	4.010E-5	3.827E-5
		10	0.0001291	0.0001433	0.0001035	0.0001411	0.0001178	0.0001378	0.0001246	0.0001429	0.0001127

7.5 Image segmentation

The values of GCE, PRI, VOI, PSNR, FSIM, SSIM for different algorithms along with ChOA can be found from Tables 7, 8, 9, 10, 11, 12 respectively.

For the case of PRI, using Kapur's entropy, for image 1, ChOA gives the best value for K = 5. It gives competitive value while standing at third position comparing to other algorithms for K = 8, 10. For image 2, it gives the best value for K = 5. For image 3, it gives the best value for K = 2. For image 4, it gives the best value for K = 2. Using Otsu's entropy, for image 1, ChOA gives the best value for K = 5. For image 2, it does not provide best value for any case all of the values are nearly close to the other competing algorithms. For image 3, ChOA gives the best value for K = 2. For image 4, it provides the best value for K = 2 and while standing at third position, gives competitive value for K = 5.

Now in case of GCE, using Kapur's entropy, for image 1, ChOA gives the best value for K = 2, and provides good, competitive values for K = 8, 10. For image 2, it gives the best value for K = 2 and second best value for K = 8. For image 3, it provides the best values for K = 5, 8, and third

Table 5 Q Values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	3.378E-6	3.378E-6	3.378E-6	3.378E-6	3.378E-6	3.378E-6	3.378E-6	3.378E-6	3.418E-6
		5	2.025E-5	2.073E-5	2.013E-5	2.004E-5	2.035E-5	2.073E-5	1.995E-5	2.012E-5	2.066E-5
		8	7.664E-5	7.327E-5	8.287E-5	7.505E-5	8.013E-5	7.422E-5	7.444E-5	6.491E-5	7.079E-5
		10	0.0002739	0.0002524	0.0002930	0.0002429	0.0002730	0.0002532	0.0002522	0.0002639	0.0002360
	Otsu	2	7.013E-6	7.013E-6	7.013E-6	7.013E-6	7.013E-6	7.013E-6	7.013E-6	7.013E-6	7.314E-6
		5	2.306E-5	2.355E-5	2.350E-5	2.353E-5	2.328E-5	2.369E-5	2.355E-5	2.353E-5	2.303E-5
		8	8.700E-5	9.003E-5	8.976E-5	0.0001014	8.080E-5	8.149E-5	8.399E-5	8.152E-5	7.114E-5
		10	0.0003514	0.0002630	0.0003114	0.0002965	0.0003621	0.0004279	0.0003278	NaN	0.0002937
Image 2	Kapur	2	1.118E-5	1.118E-5	1.118E-5	1.118E-5	1.118E-5	1.118E-5	1.118E-5	1.118E-5	1.125E-5
		5	2.973E-5	2.978E-5	2.946E-5	2.892E-5	2.953E-5	2.967E-5	2.928E-5	2.872E-5	2.957E-5
		8	5.703E-5	5.641E-5	5.776E-5	5.693E-5	5.791E-5	5.672E-5	5.891E-5	5.753E-5	5.571E-5
		10	8.795E-5	8.746E-5	9.076E-5	8.891E-5	8.987E-5	8.844E-5	8.554E-5	8.954E-5	9.588E-5
	Otsu	2	1.388E-5	1.388E-5	1.388E-5	1.388E-5	1.388E-5	1.388E-5	1.388E-5	1.388E-5	1.392E-5
		5	3.470E-5	3.471E-5	3.479E-5	3.482E-5	3.471E-5	3.485E-5	3.473E-5	3.479E-5	3.453E-5
		8	6.966E-5	7.429E-5	7.008E-5	7.192E-5	6.749E-5	7.311E-5	6.947E-5	7.090E-5	6.664E-5
		10	0.0001120	0.0001156	0.0001049	0.0001120	0.0001043	0.0001236	0.0001113	0.0001097	0.0001188
Image 3	Kapur	2	3.295E-6	3.295E-6	3.295E-6	3.295E-6	3.295E-6	3.295E-6	3.295E-6	3.295E-6	3.290E-6
		5	6.075E-6	6.464E-6	6.348E-6	6.280E-6	6.481E-6	6.500E-6	5.947E-6	6.342E-6	6.074E-6
		8	1.188E-5	1.170E-5	1.177E-5	1.180E-5	1.193E-5	1.215E-5	1.188E-5	1.179E-5	1.187E-5
		10	1.842E-5	1.810E-5	1.853E-5	1.814E-5	1.806E-5	1.742E-5	1.764E-5	1.838E-5	1.718E-5
	Otsu	2	2.675E-6	2.675E-6	2.675E-6	2.675E-6	2.675E-6	2.675E-6	2.675E-6	2.675E-6	2.665E-6
		5	6.389E-6	6.410E-6	6.363E-6	6.402E-6	6.402E-6	6.410E-6	6.404E-6	6.402E-6	6.437E-6
		8	1.514E-5	1.543E-5	1.547E-5	1.531E-5	1.539E-5	1.536E-5	1.540E-5	1.530E-5	1.348E-5
		10	2.370E-5	2.327E-5	2.332E-5	2.356E-5	2.396E-5	2.359E-5	2.339E-5	2.312E-5	2.058E-5
Image 4	Kapur	2	8.742E-6	8.742E-6	8.742E-6	8.742E-6	8.742E-6	8.742E-6	8.742E-6	8.742E-6	8.761E-6
		5	2.481E-5	2.454E-5	2.450E-5	2.457E-5	2.475E-5	2.489E-5	2.482E-5	2.428E-5	2.394E-5
		8	5.688E-5	6.097E-5	5.483E-5	5.874E-5	5.662E-5	5.751E-5	5.554E-5	5.704E-5	5.565E-5
		10	0.0001263	0.0001048	0.0001176	0.0001105	0.0001189	0.0001193	0.0001169	0.0001159	0.0001123
	Otsu	2	8.021E-6	8.021E-6	8.021E-6	8.021E-6	8.021E-6	8.021E-6	8.021E-6	8.021E-6	7.984E-6
		5	2.593E-5	2.718E-5	2.664E-5	2.724E-5	2.574E-5	2.653E-5	2.612E-5	2.670E-5	2.657E-5
		8	6.994E-5	6.746E-5	6.766E-5	6.770E-5	6.987E-5	6.835E-5	7.110E-5	6.756E-5	6.052E-5
		10	0.0001497	0.0001517	0.0001225	0.0001476	0.0001382	0.0001578	0.0001438	0.0001514	0.0001336

best value for K = 10. For image 4, it provides second best value for K = 8 and fifth best for K = 5. Using Otsu's entropy, for image 1, ChOA gives the best value for K = 2, 8, it also provides competitive value for K = 10 by giving the third best value. For image 2, it provides the best value for K = 2, 5, 10, and third best value for K = 8. For image 3, it provides the best value for K = 5 and competitive values for other cases. For image 4, it provides best value for K = 8 and second value for K = 10, while providing competitive values for the other two cases (Fig. 15).

For the case of VOI, using Kapur's entropy, for image 1, ChOA gives the best value for K = 2, 5, and competitive values for higher thresholds. For image 2, it gives the best values for K = 2 and competitive value for K = 5. For image 3, it provides the best value for K = 5 while providing nearly the best values for the other cases. For image 4, it gives the second best value for K = 5. Using Otsu's variance, for image 1, it provides the best value for K = 2. For image 2, it provides the best values for K = 2, 5. For image 3, it provides the best value for K = 2 and competitive values for

Table 6 Number of Clusters

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	22	22	22	22	22	22	22	22	22
		5	109	110	110	110	110	110	109	110	115
		8	296	289	298	292	296	293	292	288	296
		10	460	467	468	466	466	462	460	476	457
	Otsu	2	25	25	25	25	25	25	25	25	25
		5	138	141	141	141	139	141	141	141	135
		8	341	341	341	341	337	341	345	346	320
		10	539	552	528	562	524	546	541	NaN	482
Image 2	Kapur	2	15	15	15	15	15	15	15	15	15
		5	59	59	58	58	59	59	58	58	56
		8	124	123	124	123	124	123	126	125	121
		10	179	183	182	183	180	191	182	181	191
	Otsu	2	19	19	19	19	19	19	19	19	19
		5	66	66	66	66	66	66	66	66	64
		8	141	145	143	143	141	144	143	143	138
		10	211	215	208	213	206	217	211	211	213
Image 3	Kapur	2	17	17	17	17	17	17	17	17	17
		5	52	52	53	52	54	53	48	52	51
		8	112	110	112	110	110	112	111	110	110
		10	179	167	183	170	177	171	173	172	163
	Otsu	2	16	16	16	16	16	16	16	16	16
		5	65	65	65	65	65	65	65	65	59
		8	147	149	150	149	149	148	149	149	127
		10	232	226	228	228	226	231	230	226	191
Image 4	Kapur	2	20	20	20	20	20	20	20	20	20
		5	98	96	96	97	98	98	98	96	93
		8	244	243	244	242	245	243	243	244	235
		10	379	356	383	361	378	373	383	374	357
	Otsu	2	23	23	23	23	23	23	23	23	23
		5	108	109	109	109	108	109	108	109	105
		8	286	292	279	292	282	292	289	292	253
		10	439	437	419	436	435	447	440	436	398

other cases. For image 4, it provides values close to the best value in the list for all four threshold values.

In case of image 1, ChoA gave the best result in PSNR for $k = 2, 5$ using Kapur's method; SSIM result was best in $k = 1$, and for FSIM the result was best when $k = 2, 5$ for Kapur's method and when $k = 2$ for otsus' method.

For image 2, ChOA's best result result in PSNR was when $k = 5$ using Kapur's method, and SSIM result was best when $k = 5$ using Kapur's method, and FSIM results were exactly similar to that of SSIM and PSNR (Fig. 16).

In case of image 3, PSNR was best when $k = 2$ using Kapur's method and when $k = 2, 8$ using Otsu's method, and the algorithm gave the second best PSNR when $k = 10$ using Otsu's method. For SSIM, in both Kapur's and Otsu's method, the best result was when $k = 2$. For FSIM, ChOA gave the best for $k = 2$ in Kapur's method and $k = 2$ in Otsu's method; ChOA also gave third best FSIM using Otsu's method.

For image 4, best result for PSNR was given by ChOA when $k = 2$ in both of the method.

In case of other threshold numbers, ChOA gave competitive result in the rest of the parameters.

The result achieved for different image segmentation parameters signifies Chimp Optimization Algorithm's

Table 7 GCE values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	0.7401022	0.7401022	0.7401022	0.7401022	0.7401022	0.7401022	0.7401024	0.7401022	0.739842l
		5	0.8090890	0.8087080	0.8122764	0.8115658	0.8095260	0.8076563	0.8114339	0.8126369	0.8127623
		8	0.8375948	0.8362083	0.8377495	0.8341675	0.8401173	0.8346083	0.8342278	0.8478774	0.8371196
		10	0.8377620	0.8465624	0.8346650	0.8469645	0.8383987	0.8378010	0.8424697	0.8409490	0.8404271
	Otsu	2	0.7380297	0.7380297	0.7380297	0.7380297	0.7380297	0.7380297	0.7380297	0.7380297	0.7352447
		5	0.8184662	0.8151762	0.8155834	0.8151739	0.8179646	0.8126813	0.8121230	0.8151739	0.8157110
		8	0.8364772	0.8394654	0.8378076	0.8408182	0.8351204	0.8368324	0.8374493	0.8383382	0.8293978
		10	0.8268988	0.8255083	0.8306147	0.8267042	0.8362147	0.8222586	0.8322172	NaN	0.8234572
Image 2	Kapur	2	0.7067648	0.7067648	0.7067648	0.7067648	0.7067648	0.7067648	0.7067648	0.7067648	0.6995715
		5	0.7692075	0.7666628	0.7700412	0.7652705	0.7687575	0.7683177	0.7676031	0.7664471	0.7729088
		8	0.7847124	0.7885286	0.7855112	0.78114062	0.7847000	0.7844945	0.7873781	0.7852935	0.7814643
		10	0.7861190	0.7885115	0.7892061	0.7895727	0.7879943	0.7893165	0.7932822	0.7860364	0.7897926
	Otsu	2	0.7189673	0.7189673	0.7189673	0.7189673	0.7189673	0.7189673	0.7189673	0.7189673	0.7189496
		5	0.7840605	0.7849918	0.7881583	0.7872190	0.7878117	0.7843835	0.7878127	0.7881583	0.7788456
		8	0.7935995	0.8027025	0.7979745	0.8047776	0.7987591	0.7973333	0.7934298	0.7944037	0.7936273
		10	0.7961882	0.7955455	0.7950484	0.7961843	0.7957715	0.7921153	0.7993835	0.7962816	0.7914766
Image 3	Kapur	2	0.7071777	0.7071777	0.7071777	0.7071777	0.7071777	0.7071777	0.7071777	0.7071777	0.7098366
		5	0.7169523	0.7202978	0.7270952	0.7263886	0.7254977	0.7175942	0.7281371	0.7313596	0.7152261
		8	0.7592833	0.7524897	0.7514513	0.7549964	0.7571284	0.7521763	0.7487347	0.7503311	0.7368054
		10	0.7511133	0.7568617	0.7561045	0.7544547	0.7530906	0.7652466	0.7628362	0.7554382	0.7535885
	Otsu	2	0.7068947	0.7068947	0.7068947	0.7068947	0.7068947	0.7068947	0.7068947	0.7068947	0.7069815
		5	0.7377244	0.7419418	0.7414628	0.7416817	0.7416817	0.7414860	0.7405943	0.7416817	0.7353298
		8	0.7494254	0.7521713	0.7516004	0.7519477	0.7557806	0.7490750	0.7473474	0.7522846	0.7504423
		10	0.7430081	0.7372852	0.7422038	0.7385129	0.7414544	0.7467788	0.7401779	0.7376211	0.7505356
Image 4	Kapur	2	0.7633346	0.7633346	0.7633346	0.7633346	0.7633346	0.7633346	0.7633346	0.7633346	0.7677439
		5	0.8048444	0.8010997	0.8007615	0.8044664	0.8043818	0.8063933	0.8045323	0.8004366	0.8044034
		8	0.8251521	0.8213815	0.8246816	0.8197151	0.8279487	0.8239419	0.8296359	0.8231953	0.8201649
		10	0.8194140	0.8252892	0.8124996	0.8168426	0.8160397	0.8153317	0.8122309	0.8173751	0.8202654
	Otsu	2	0.7748162	0.7748162	0.7748162	0.7748162	0.7748162	0.7748162	0.7748162	0.7748162	0.7773568
		5	0.8232756	0.8079247	0.8144996	0.8078039	0.8245431	0.8151019	0.8191737	0.8138300	0.8180743
		8	0.8242975	0.8220828	0.8236080	0.8216555	0.8264061	0.8235773	0.8256691	0.8222287	0.8201661
		10	0.8230292	0.8220396	0.828461	0.8216130	0.8220380	0.8197402	0.8223072	0.8166962	0.8173459

effectiveness in multilevel thresholding based applications. ChOA can keep up with moderate to competitive performance in most of the cases, and in many cases it outperforms all other algorithms that we have compared with. It is to mention that ChOA provided comparatively better performance in terms of these segmentation parameters while lower levels of thresholding were used. The different chaotic maps available in the ChOA can be utilized to generate better results by switching to different values of m (23). However, there is no universal metaheuristic algorithm that can outperform all other algorithms in all segmentation tasks. That is why, ChOA can be considered as quite an important one in this regard.

7.6 Image quality

To measure the quality of the segmented image, parameters BRISQUE,PIQE, and NIQE were used to evaluate, and the values of the paremeters are given in the Table 13, Table 14, Table 15 accordingly.

Table 8 PRI values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	0.9514666	0.9514666	0.9514666	0.9514666	0.9514666	0.9514666	0.9514666	0.9514666	0.9513863
		5	0.9727083	0.9726548	0.9733310	0.9732004	0.9727636	0.9725107	0.9731275	0.9733631	0.9743540
		8	0.9815287	0.9815430	0.9820300	0.9814921	0.9819127	0.9812118	0.9814995	0.9829055	0.9819686
		10	0.9833491	0.9839416	0.9833330	0.9840665	0.9834140	0.9834156	0.9836827	0.9838227	0.9838362
	Otsu	2	0.9515554	0.9515554	0.9515554	0.9515554	0.9515554	0.9515554	0.9515554	0.9515554	0.9509178
		5	0.9767826	0.9764159	0.9764649	0.9764209	0.9767403	0.9760988	0.9760339	0.9764209	0.9777553
		8	0.9835268	0.9835102	0.9835367	0.9835569	0.9832785	0.9833973	0.9836028	0.9837485	0.9824061
		10	0.9844744	0.9848700	0.9846917	0.9848447	0.9851909	0.9842573	0.9847806	NaN	0.9829497
Image 2	Kapur	2	0.9335052	0.9335052	0.9335052	0.9335052	0.9335052	0.9335052	0.9335052	0.9335052	0.9291518
		5	0.971205	0.9711415	0.9724253	0.9713595	0.9711347	0.9713116	0.9714544	0.9713961	0.9734346
		8	0.9816942	0.9818422	0.9818438	0.9812145	0.9816557	0.9816419	0.9817693	0.9817147	0.9792358
		10	0.9836498	0.9835151	0.9839184	0.9838088	0.9838458	0.9834957	0.9841798	0.9836209	0.9823904
	Otsu	2	0.9441387	0.9441387	0.9441387	0.9441387	0.9441387	0.9441387	0.9441387	0.9441387	0.9440744
		5	0.9775444	0.9775307	0.9777809	0.9776590	0.9778562	0.9773850	0.9778163	0.9777809	0.9772468
		8	0.9832419	0.9840568	0.9836675	0.9841374	0.9838575	0.9835854	0.9832675	0.9834995	0.9823950
		10	0.9853449	0.9855164	0.985329	0.9855252	0.9851840	0.9853058	0.9856893	0.9854580	0.9839688
Image 3	Kapur	2	0.9111205	0.9111205	0.9111205	0.9111205	0.9111205	0.9111205	0.9111205	0.9111205	0.9128027
		5	0.9310841	0.9320394	0.9339085	0.9370163	0.9318510	0.9293245	0.9331559	0.9351969	0.9300046
		8	0.9663346	0.9649999	0.9650456	0.9648381	0.9665265	0.9635618	0.9632104	0.9644666	0.9584711
		10	0.9680131	0.9682988	0.9683558	0.9687684	0.9679444	0.9692214	0.9686050	0.9680466	0.9674254
	Otsu	2	0.9085145	0.9085145	0.9085145	0.9085145	0.9085145	0.9085145	0.9085145	0.9085145	0.9086484
		5	0.9595404	0.9598124	0.9596954	0.9597562	0.9597562	0.9598261	0.9597341	0.9597562	0.9566119
		8	0.9702900	0.9700872	0.9701775	0.9699262	0.9705375	0.9700931	0.9697596	0.9699806	0.9693644
		10	0.9741403	0.9732780	0.9736777	0.9734191	0.9739932	0.9743388	0.9743474	0.9732717	0.9725623
Image 4	Kapur	2	0.9307807	0.9307807	0.9307807	0.9307807	0.9307807	0.9307807	0.9307807	0.9307807	0.9323508
		5	0.9671772	0.9659721	0.9655904	0.9665656	0.9671403	0.9673020	0.9670801	0.9658614	0.9654510
		8	0.9782044	0.9760988	0.9787477	0.9763018	0.9785625	0.9777054	0.9794284	0.9778260	0.9756341
		10	0.9806452	0.9804251	0.9797207	0.9791084	0.9798526	0.9793233	0.9790864	0.9800520	0.9789193
	Otsu	2	0.9457497	0.9457497	0.9457497	0.9457497	0.9457497	0.9457497	0.9457497	0.9457497	0.9466141
		5	0.9747321	0.9726784	0.9735348	0.9726290	0.9748935	0.9736714	0.9743312	0.9734442	0.9745284
		8	0.9805452	0.9809622	0.9800648	0.9809254	0.9807647	0.9811282	0.9811564	0.9809844	0.9791631
		10	0.9833113	0.9837541	0.9832399	0.9836989	0.9831646	0.9830735	0.9834571	0.9831194	0.9810584

In terms of BRISQUE values, ChOA gave such threshold values that the segmented images' BRISQUE values were best for the most test images.For $k = 2, 5, 10,$ segmented image 1 through ChOA generated threshold values using kapur's method gave the best BRISQUE scores. In case of Otsu's method, for $k = 2, 5$ best result was observed. In case of test image 2, best score was observed when $k = 2$ using kapur's method and when $k = 2, 5$ using otsu's method. For test image 3, when $k = 2, 8$ using kapur's method, best BRISQUE score was given by the segmented image, and for otsu's method best BRISQUE score was observed when$k = 2, 8, 10$. For test image 4, best BRISQUE score was when $k = 2, 5$ using otsu's method and when $k = 2$ kapur's method.

PIQE value was the lowest for test image 1 when $k = 2, 8$ using kapur's method. For test image 2, Lowest PIQE score was observed using otsu's method, when $k = 5, 8, 10$ and when $k = 2$ the result was same as other score. For test image 3,using the kapur's method best score was observed

Table 9 VOI values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	7.7414537	7.7414537	7.7414537	7.7414537	7.7414537	7.7414537	7.7414537	7.7414537	7.7392359
		5	7.6464127	7.6473644	7.6586971	7.6542860	7.6491390	7.6407346	7.6563863	7.6589395	7.6198831
		8	7.3209785	7.2953370	7.2686000	7.2812488	7.3214487	7.3031276	7.2792588	7.3684446	7.3043113
		10	7.0872033	7.1561476	7.0397532	7.1626582	7.1028853	7.0699888	7.1227961	7.1053111	7.1163711
	Otsu	2	7.7454406	7.7454406	7.7454406	7.7454406	7.7454406	7.7454406	7.7454406	7.7454406	7.7384476
		5	7.5359435	7.5208049	7.5207965	7.5196001	7.5327450	7.5109894	7.5078136	7.5196001	7.5586077
		8	7.1565769	7.1989957	7.1756813	7.2162831	7.1735594	7.1704616	7.1565963	7.1508188	7.2519053
		10	6.8640768	6.8314111	6.8897506	6.8419384	6.9388738	6.8066352	6.9008102	NaN	7.0459398
Image 2	Kapur	2	7.8948613	7.8948613	7.8948613	7.8948613	7.8948613	7.8948613	7.8948613	7.8948613	7.882368
		5	7.3635814	7.3520387	7.3433963	7.3400988	7.3620559	7.3503067	7.3505597	7.3348338	7.3475186
		8	6.8739086	6.9013682	6.8810830	6.8584328	6.8710377	6.8697184	6.8928478	6.8759233	7.0089640
		10	6.6894738	6.7292648	6.7076410	6.7193950	6.6864872	6.7205817	6.7448233	6.7156823	6.8161101
	Otsu	2	7.7596360	7.7596360	7.7596360	7.7596360	7.7596360	7.7596360	7.7596360	7.7596360	7.7571894
		5	7.2477956	7.2556980	7.2738335	7.2712954	7.2676890	7.2605806	7.2682256	7.2738335	7.2462195
		8	6.8826767	6.9289199	6.9064509	6.9353297	6.8957024	6.9073685	6.8688403	6.9018369	6.9496651
		10	6.6563287	6.6494782	6.6547930	6.6481812	6.6635680	6.6203637	6.6643406	6.6427772	6.7412509
Image 3	Kapur	2	7.2418652	7.2418652	7.2418652	7.2418652	7.2418652	7.2418652	7.2418652	7.2418652	7.2486244
		5	6.8934430	6.9236784	6.9468805	6.8870208	6.9618974	6.9548914	6.9256135	6.9443456	6.8818086
		8	6.4523966	6.4471165	6.4335926	6.4597450	6.4347797	6.4693338	6.4547459	6.4502595	6.4652110
		10	6.2466067	6.2667075	6.2403861	6.2393676	6.2480174	6.3146049	6.3083116	6.2519319	6.2919275
	Otsu	2	7.2727034	7.2727034	7.2727034	7.2727034	7.2727034	7.2727034	7.2727034	7.2727034	7.2719195
		5	6.5187494	6.5475902	6.5483656	6.5451919	6.5451919	6.5445444	6.5370799	6.5451919	6.6139727
		8	6.2166216	6.2429659	6.2219199	6.2455054	6.2499061	6.2207273	6.1996692	6.2464193	6.2654107
		10	5.9134495	5.9558019	5.9378976	5.9560140	5.9140732	5.9383446	5.8725698	5.9577600	6.0753521
Image 4	Kapur	2	8.2033327	8.2033327	8.2033327	8.2033327	8.2033327	8.2033327	8.2033327	8.2033327	8.2225951
		5	7.6615819	7.6410822	7.6319375	7.6499954	7.6485510	7.6689197	7.6510448	7.6252340	7.6301180
		8	7.1798438	7.2404103	7.1640516	7.2128221	7.1962700	7.2066466	7.2040319	7.1886314	7.2833056
		10	6.8815728	6.9812881	6.8304234	6.9186151	6.8609963	6.8567897	6.8379102	6.8771390	7.0486096
	Otsu	2	7.9352456	7.9352456	7.9352456	7.9352456	7.9352456	7.9352456	7.9352456	7.9352456	7.9571057
		5	7.4766521	7.4261800	7.4488390	7.4277320	7.4804995	7.4486894	7.4582465	7.4467283	7.4831069
		8	6.9162132	6.8438346	6.9444255	6.8400796	6.9535993	6.8530490	6.8951099	6.8449935	7.0930089
		10	6.7137583	6.6534381	6.7947227	6.6611856	6.7079663	6.6771988	6.6959007	6.6500289	6.8004151

when$k = 8$, and for otsu's method the best result was given by the segmented image through ChOA generated set of threholds when $k = 5, 10$. For image 4, best result score was given by otsu's method when $k = 2, 5$.

In the case of NIQE value for test image 1, best result was observed when $k = 2, 10$ using kapur's method, but while using otsu's method, all the algorithm gave same result for$k = 2$, and ChOA gave second best result when $k = 10$. For test image 2,ChOA gave the best result when $k = 2$ for kapur's method and when $k = 5$ for otsu's method. For image 3, best result was observed when $k = 5, 10$ using otsu's method. For image 4, when $k = 8$, ChOA gave best result using kapur's method, and when $k = 5$, ChOA gave best result using otsu's method.

7.7 Convergence curve

From Figs. 17, 18, 19, 20, 21, 22 and 23 represent the convergence curve of the all applied algorithms. ChOA shows sharp increasing rate when the threshold value is 2 and this behaviour was common for all the test image. As for higher threshold values ChOA showed almost flat line behaviour and other algorithms performed performed nicely in terms of convergence; however, ChOA gave maximum objective value even with poor convergence. When threhold value is low ChOA performed consistently, giving the best results in terms of objective function value (Fig. 24).

Table 10 PSNR values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	28.440419	28.440419	28.440419	28.440419	28.440419	28.440419	28.440419	28.440419	28.452249
		5	32.351433	32.400800	32.292423	32.281401	32.359172	32.413187	32.272944	32.319795	32.660504
		8	35.722752	35.702646	35.908669	35.757708	35.803129	35.709419	35.774554	35.846677	35.425915
		10	37.127203	37.179786	37.218245	37.130356	37.132836	37.228318	37.102577	37.210059	36.209800
	Otsu	2	28.123691	28.123691	28.123691	28.123691	28.123691	28.123691	28.123691	28.123691	28.113286
		5	33.337560	33.319674	33.317595	33.316819	33.328843	33.316702	33.325509	33.316819	33.239077
		8	36.708543	36.658679	36.675625	36.649052	36.724064	36.683214	36.709888	36.725963	35.657104
		10	38.150669	38.278731	38.151941	38.303141	38.203479	38.149190	38.212074	NaN	36.508223
Image 2	Kapur	2	25.857437	25.857437	25.857437	25.857437	25.857437	25.857437	25.857437	25.857437	25.824765
		5	31.829167	31.913280	32.007579	31.993688	31.834822	31.928337	31.920072	31.979089	32.180175
		8	36.232772	36.133439	36.302014	36.186282	36.224685	36.223779	36.173569	36.205566	34.819399
		10	37.348654	37.530843	37.530000	37.533199	37.331486	37.535321	37.520433	37.473257	35.746845
	Otsu	2	26.519673	26.519673	26.519673	26.519673	26.519673	26.519673	26.519673	26.519673	26.535550
		5	33.007137	33.002360	33.009911	33.004080	33.026311	33.003262	33.014204	33.009911	32.771950
		8	36.182207	36.227699	36.173640	36.169374	36.292544	36.176238	36.134697	36.157687	35.596727
		10	37.998159	37.977712	37.987644	37.978985	37.909767	38.035465	38.052298	37.971090	36.592628
Image 3	Kapur	2	27.238564	27.238564	27.238564	27.238564	27.238564	27.238564	27.238564	27.238564	27.271631
		5	32.200322	32.081054	32.127274	32.484471	31.933075	31.818954	32.159671	32.211432	31.907360
		8	36.913962	36.750383	36.848534	36.701875	36.966275	36.515399	36.558050	36.777986	35.962618
		10	37.954458	37.837659	38.053994	38.009902	37.942377	37.879142	37.896664	37.893732	37.921900
	Otsu	2	28.608519	28.608519	28.608519	28.608519	28.608519	28.608519	28.608519	28.608519	28.623584
		5	35.201047	35.180165	35.171299	35.177109	35.177109	35.190577	35.183305	35.177109	34.640436
		8	37.604194	37.596324	37.604888	37.569438	37.587983	37.597968	37.601087	37.566961	37.726866
		10	38.975646	38.965201	39.059186	38.949959	38.982014	38.936565	39.109834	38.933496	39.102334
Image 4	Kapur	2	25.827964	25.827964	25.827964	25.827964	25.827964	25.827964	25.827964	25.827964	25.828374
		5	32.861925	32.809276	32.804067	32.839241	32.867054	32.865091	32.867840	32.820803	32.579941
		8	36.641179	36.331353	36.695096	36.399663	36.650502	36.525197	36.655586	36.573710	35.870206
		10	38.186096	37.946014	38.303392	38.033067	38.182733	38.215559	38.276485	38.208637	37.193032
	Otsu	2	27.069915	27.069915	27.069915	27.069915	27.069915	27.069915	27.069915	27.069915	27.073884
		5	33.894314	34.078302	34.001326	34.081949	33.880391	33.989084	33.933373	34.009388	33.615425
		8	37.544554	37.664917	37.504057	37.662937	37.496286	37.638538	37.615286	37.666288	36.861737
		10	39.072244	39.045143	39.007044	38.981624	39.084399	39.118601	39.129383	38.920229	38.472067

7.8 Observation and summary

Observing all the parameters, ChOA didn't give best results consistently. For higher threshold the algorithm gave almost flat line convergence curve. When threshold is 2, the algorithm's convergence curve had a sharp increment towards the global optimum. The possible reason could be the entrapment of search agents of ChOA by multi-modal search space rising from higher threshold values. Even in the original work of ChOA, it was observed that ChOA performed relatively poor in some of the multi-modal functions. This issue can be solved by using other versions of ChOA that employs different chaotic map to diversify search agents.In Houssein et al. (2021), authors used Levy flight to improve algorithm efficiency. There can be other ways to improve ChOA.

In case of objective function, the used algorithm to implement the Kapur's function and Otsu's function can be modified to give better results. Since stochastic algorithm generates random search agents, it doesn't always generate discrete and ascending threshold values that can be evaluated by either Kapur's or Otsu's function. Kapur's and Otsu's function can only take discrete and ascending

Table 11 FSIM values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	0.840559	0.840559	0.840559	0.840559	0.840559	0.840559	0.840559	0.840559	0.840585
		5	0.920810	0.920489	0.921740	0.921506	0.920802	0.920466	0.921496	0.921822	0.922871
		8	0.956185	0.956611	0.958265	0.957139	0.957226	0.956513	0.957194	0.957968	0.955393
		10	0.967603	0.967897	0.968163	0.967937	0.967688	0.968179	0.967967	0.968330	0.964482
	Otsu	2	0.842626	0.842626	0.842626	0.842626	0.842626	0.842626	0.842626	0.842626	0.842780
		5	0.930398	0.930382	0.930389	0.930392	0.930353	0.930403	0.930381	0.930392	0.925230
		8	0.961796	0.961615	0.961596	0.961441	0.961574	0.961678	0.961959	0.962526	0.954122
		10	0.972135	0.972140	0.971904	0.972694	0.972774	0.971812	0.972286	NaN	0.962240
Image 2	Kapur	2	0.770299	0.770299	0.770299	0.770299	0.770299	0.770299	0.770299	0.770299	0.769619
		5	0.857914	0.858560	0.860277	0.859594	0.857929	0.858511	0.858847	0.859447	0.863357
		8	0.914322	0.9143504	0.916417	0.914266	0.914216	0.914366	0.915251	0.914788	0.906757
		10	0.931493	0.931609	0.932380	0.931393	0.931270	0.931313	0.931200	0.931356	0.9226307
	Otsu	2	0.774892	0.774892	0.774892	0.774892	0.774892	0.774892	0.774892	0.774892	0.774767
		5	0.874577	0.874794	0.874762	0.874809	0.874543	0.874779	0.874724	0.874762	0.871927
		8	0.914904	0.915005	0.914990	0.914877	0.916857	0.915032	0.914741	0.915427	0.913923
		10	0.935913	0.933419	0.934923	0.933248	0.935793	0.936358	0.935601	0.933554	0.928870
Image 3	Kapur	2	0.822938	0.822938	0.822938	0.822938	0.822938	0.822938	0.822938	0.822938	0.823001
		5	0.896473	0.895169	0.895499	0.899030	0.894081	0.892958	0.897964	0.897028	0.894255
		8	0.944715	0.943191	0.944140	0.942881	0.945533	0.940668	0.940966	0.943038	0.935226
		10	0.954399	0.953651	0.953879	0.954447	0.954414	0.953765	0.953443	0.952818	0.949903
	Otsu	2	0.826202	0.826202	0.826202	0.826202	0.826202	0.826202	0.826202	0.826202	0.826305
		5	0.926220	0.925754	0.925606	0.925764	0.925764	0.925781	0.925963	0.925764	0.920650
		8	0.950123	0.949262	0.949465	0.949070	0.949579	0.949730	0.949622	0.949099	0.949689
		10	0.961428	0.961404	0.961159	0.961271	0.961952	0.960784	0.962236	0.961474	0.960805
Image 4	Kapur	2	0.789832	0.789832	0.789832	0.789832	0.789832	0.789832	0.789832	0.789832	0.789598
		5	0.909151	0.908696	0.908999	0.908971	0.909365	0.908996	0.909382	0.908718	0.904972
		8	0.945327	0.943294	0.945280	0.943423	0.944913	0.944275	0.944578	0.944375	0.939712
		10	0.958685	0.957340	0.9593667	0.957810	0.958701	0.959220	0.959387	0.958811	0.952971
	Otsu	2	0.830418	0.830418	0.830418	0.830418	0.830418	0.830418	0.830418	0.830418	0.830283
		5	0.919870	0.921300	0.920723	0.921355	0.919776	0.920561	0.920125	0.920786	0.915629
		8	0.955408	0.956310	0.955097	0.956294	0.954843	0.956447	0.955812	0.956305	0.947662
		10	0.965971	0.965944	0.964779	0.965613	0.965779	0.966354	0.966469	0.965426	0.961469

set of threshold values to evaluate the objective function. So, when an stochastic algorithm generates any value that is not in ascending order the kapur's and Otsu's currently implemented method gives Zero as result, and there is no method implemented in ChOA to generate only ascending set of threshold values. As a results, many iterations are wasted in generating zero as objective value, but due to exiting cognisant component in the algorithm, the algorithm can generate ascending threshold values after many iterations. So, one improvement can be to make ChOA cognisant in generating only ascending order values.

Another issue in ChOA is that it is a continuous algorithm. So, it can generate any threshold values between [0,255]. In current method, solution set is rounded to generate discrete values. This method is inefficient because many iterations are wasted in generating different decimal values,

Table 12 SSIM Values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	0.7918196	0.7918196	0.7918196	0.7918196	0.7918196	0.7918196	0.7918196	0.7918196	0.7916691
		5	0.8515985	0.8515301	0.8521869	0.8520201	0.8516498	0.8515387	0.8519308	0.8522966	0.8542401
		8	0.8909610	0.8901224	0.8925359	0.8910390	0.8915917	0.8902823	0.8914495	0.8917443	0.8880165
		10	0.9064684	0.9061279	0.9069916	0.9060128	0.9063522	0.9068684	0.9061212	0.9070064	0.9000728
	Otsu	2	0.7943324	0.7943324	0.7943324	0.7943324	0.7943324	0.7943324	0.7943324	0.7943324	0.7943536
		5	0.8697263	0.8698344	0.8698296	0.8698380	0.8697111	0.8698738	0.8698589	0.8698380	0.8631996
		8	0.9024935	0.9032744	0.9026491	0.9031359	0.9022195	0.9031444	0.9029160	0.9037240	0.8920301
		10	0.9183362	0.9206070	0.9178337	0.9204159	0.9177207	0.9179366	0.9178194	NaN	0.9033850
Image 2	Kapur	2	0.7381052	0.7381052	0.7381052	0.7381052	0.7381052	0.7381052	0.7381052	0.7381052	0.7374410
		5	0.8316037	0.8329436	0.8336456	0.8339786	0.8317861	0.8331025	0.8328995	0.8336116	0.8368826
		8	0.8948718	0.8943560	0.8963653	0.8948223	0.8953343	0.8950658	0.895129	0.8949714	0.8803619
		10	0.9085048	0.9108405	0.9104283	0.9106738	0.9084714	0.9120375	0.9103258	0.9097330	0.8982321
	Otsu	2	0.7494828	0.7494828	0.7494828	0.7494828	0.7494828	0.7494828	0.7494828	0.7494828	0.7493831
		5	0.8545080	0.8545495	0.8544388	0.8544629	0.8542291	0.8544344	0.8544506	0.8544388	0.8476260
		8	0.8968954	0.8985209	0.8974504	0.8984323	0.8973888	0.8986785	0.8968397	0.8971918	0.890594
		10	0.9175194	0.9166383	0.9166181	0.9164011	0.9164443	0.9182397	0.9174849	0.9161740	0.9060393
Image 3	Kapur	2	0.8260656	0.8260656	0.8260656	0.8260656	0.8260656	0.8260656	0.8260656	0.8260656	0.8261581
		5	0.8902364	0.8898632	0.8903845	0.8938270	0.8887642	0.8876132	0.8907992	0.8915735	0.8883194
		8	0.9331523	0.9314820	0.9330983	0.9315839	0.9344141	0.9300816	0.9298874	0.9320946	0.9231071
		10	0.9426619	0.9410390	0.9426921	0.9423984	0.9425457	0.9411709	0.9412509	0.9406771	0.9367924
	Otsu	2	0.8316937	0.8316937	0.8316937	0.8316937	0.8316937	0.8316937	0.8316937	0.8316937	0.8317416
		5	0.9227556	0.9222944	0.9222069	0.9222840	0.9222840	0.9223567	0.9224754	0.9222840	0.9165334
		8	0.9446697	0.9440481	0.9444181	0.9438437	0.9442943	0.9444317	0.9446580	0.9438648	0.9406071
		10	0.9560552	0.9554046	0.9553085	0.9554416	0.9558450	0.9557022	0.9569164	0.9554424	0.9508221
Image 4	Kapur	2	0.7724818	0.7724818	0.7724818	0.7724818	0.7724818	0.7724818	0.7724818	0.7724818	0.7722685
		5	0.8950666	0.8947921	0.8950790	0.8952765	0.8952010	0.8951351	0.8952332	0.8947384	0.8916416
		8	0.9291307	0.9276792	0.9288186	0.9276195	0.9287985	0.9282517	0.9280903	0.9284786	0.9244445
		10	0.9440305	0.9420389	0.9449645	0.9427996	0.9441784	0.9446729	0.9451792	0.9441357	0.9367905
	Otsu	2	0.8365728	0.8365728	0.8365728	0.8365728	0.8365728	0.8365728	0.8365728	0.8365728	0.8365128
		5	0.9058635	0.9069077	0.9064874	0.9069540	0.9057876	0.9063778	0.9061099	0.9065341	0.9025555
		8	0.9412635	0.9424368	0.9408417	0.9424116	0.9405580	0.9424614	0.9419765	0.9424334	0.9323812
		10	0.9516377	0.9521257	0.9507344	0.9518093	0.9515765	0.9521749	0.9520560	0.9516390	0.9465455

but all of those are useless because after generation of such values all are being rounded up. So, in summary, the cognisant component of algorithm wastes iteration on finding precise decimal values. So, efficiency can be improved if ChOA can only generate discrete values for solution. This can be done by making a Binary version of ChOA. Binary algorithm in multi-level thresholding was implemented in Djerou et al. (2009)

The computational time of the whole program can be reduced significantly using fast recursive segmentation algorithm (Kiani et al. 2009).

Table 13 BRISQUE Values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	53.621232	53.621232	53.621232	53.621232	53.621232	53.621232	53.621232	53.621232	53.461269
		5	48.399142	48.038926	48.227661	48.227661	48.107758	48.107758	48.227661	48.227661	47.354246
		8	46.766363	47.003490	45.715101	47.003490	45.495538	46.504801	46.701251	45.093831	46.809605
		10	43.852158	44.924561	43.064319	45.114398	45.239352	44.553634	45.638469	44.918923	42.422688
	Otsu	2	53.089912	53.089912	53.089912	53.089912	53.089912	53.089912	53.089912	53.089912	53.089912
		5	47.681770	47.767186	47.748683	47.748683	47.767186	47.744132	47.748683	47.748683	46.745034
		8	44.412728	46.025417	46.188126	46.184392	44.963372	44.505551	44.209560	44.306356	46.721979
		10	42.655581	42.699412	42.757367	42.605971	42.409884	42.781425	42.502356	NaN	43.547966
Image 2	Kapur	2	53.744095	53.744095	53.744095	53.744095	53.744095	53.744095	53.744095	53.744095	53.363598
		5	49.376413	49.349576	47.473984	49.376413	49.376413	49.884907	49.376413	49.591822	50.005100
		8	44.638258	44.488551	43.981701	44.599148	44.237874	44.544333	44.541907	43.849820	44.188143
		10	42.937088	43.101586	42.620880	41.916158	41.932326	42.922646	44.068029	42.707416	43.420969
	Otsu	2	52.976313	52.976313	52.976313	52.976313	52.976313	52.976313	52.976313	52.976313	52.976313
		5	46.224601	46.237927	46.280826	46.280826	46.280826	46.224601	46.280826	46.524064	45.096439
		8	43.562668	43.952136	43.933452	43.852208	42.644095	43.772070	43.119901	43.994197	43.652669
		10	41.944597	42.342318	42.966887	42.471998	42.642062	41.676957	43.942745	42.600163	43.332645
Image 3	Kapur	2	49.115084	49.115084	49.115084	49.115084	49.115084	49.115084	49.115084	49.115084	49.115084
		5	45.142081	43.661876	48.542706	43.239438	43.654126	43.582539	48.335490	43.654126	49.220157
		8	48.118491	47.624803	48.415016	47.305845	47.965350	49.344652	48.116079	48.064481	47.266019
		10	47.750451	47.353671	47.373715	47.746143	47.323979	47.906879	48.046371	47.464224	48.618302
	Otsu	2	51.592703	51.592703	51.592703	51.592703	51.592703	51.592703	51.592703	51.592703	51.561485
		5	49.998673	50.050136	50.090989	50.090989	50.090989	50.090989	50.090989	50.090989	49.581261
		8	48.767788	48.542366	48.651871	48.455135	48.731072	48.941247	48.412035	48.455135	48.914361
		10	44.970710	46.492298	46.349024	46.104870	45.683070	46.382987	44.958708	45.918083	44.916714
Image 4	Kapur	2	51.814695	51.814695	51.814695	51.814695	51.814695	51.814695	51.814695	51.814695	51.814695
		5	45.359126	45.579581	45.437545	45.326340	45.488929	45.488929	45.488929	45.425273	46.370587
		8	42.879882	43.895046	43.089536	43.496135	42.700859	42.677284	42.698178	43.676720	42.815786
		10	42.025559	41.917361	41.932548	41.849512	42.462290	41.952950	42.009516	41.958266	42.050140
	Otsu	2	46.994321	46.994321	46.994321	46.994321	46.994321	46.994321	46.994321	46.994321	46.829640
		5	45.377122	45.308196	45.308196	45.308196	45.377122	45.261731	45.377122	45.308196	44.812601
		8	42.303135	42.467727	42.303986	42.467727	42.043004	42.286388	42.296964	42.467727	42.630137
		10	41.618813	41.798283	41.460606	41.809775	42.306550	42.066605	42.309135	41.202211	41.650446

8 Conclusion

This work details the efficiency and performance of the Chimp Optimization Algorithm in image clustering, which is accomplished using multilevel thresholding. The results denote effectiveness of ChOA in the following application. ChOA, like most other metaheuristic algorithms, provides appropriate threshold values for each color channel of an RGB image by maximizing the Kapur's entropy function and Otsu's class variance function to generate image clusters. It appeared as a comparatively good algorithm in terms of its performance in image clustering, but this algorithm has

Table 14 PIQE Values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	84.069403	84.069403	84.069403	84.069403	84.069403	84.069403	84.069403	84.069403	83.970519
		5	79.271978	79.361424	79.194809	79.194809	79.471779	79.471779	79.194809	79.194809	79.307642
		8	76.692983	77.139599	77.028845	77.139599	76.358350	76.262309	77.031903	76.421931	75.694658
		10	73.913909	75.227705	72.047464	74.561676	74.519471	75.244528	75.796042	74.161480	75.581571
	Otsu	2	84.594611	84.594611	84.594611	84.594611	84.594611	84.594611	84.594611	84.594611	84.594611
		5	78.854402	79.440858	79.462360	79.462360	79.440858	79.162742	79.462360	79.462360	79.840960
		8	75.922614	75.787572	75.431440	75.706442	75.555729	74.362837	74.227601	74.173920	77.038554
		10	72.606767	71.875219	73.657577	73.063446	71.855961	73.147550	71.624828	NaN	73.250089
Image 2	Kapur	2	82.976454	82.976454	82.976454	82.976454	82.976454	82.976454	82.976454	82.976454	84.025259
		5	82.289071	82.371697	82.383127	82.289071	82.289071	83.123606	82.289071	82.996455	82.485613
		8	78.445970	79.615300	77.518025	78.120845	78.036568	78.089644	80.542365	79.792383	76.790092
		10	75.723473	76.082870	75.265179	75.463102	78.047920	73.992252	77.214337	76.339110	75.441784
	Otsu	2	85.190165	85.190165	85.190165	85.190165	85.190165	85.190165	85.190165	85.190165	85.190165
		5	83.898181	83.878921	83.091099	83.091099	83.091099	83.898181	83.091099	80.619413	79.080217
		8	77.918011	76.371845	76.267562	76.594999	81.147527	76.892436	78.087414	75.127219	74.974852
		10	77.422758	76.339474	76.970718	78.599004	78.033361	75.725783	78.888245	78.068736	74.501565
Image 3	Kapur	2	83.298082	83.298082	83.298082	83.298082	83.298082	83.298082	83.298082	83.298082	83.298082
		5	80.076455	79.961155	79.651000	79.083577	79.946206	79.677668	79.975990	79.946206	80.599417
		8	79.218795	79.203657	78.950823	79.383494	78.894796	77.853477	79.216928	79.164179	78.630666
		10	77.510763	79.055467	77.848400	77.585597	78.637234	77.755724	77.304581	78.229627	77.787952
	Otsu	2	82.801871	82.801871	82.801871	82.801871	82.801871	82.801871	82.801871	82.801871	82.795508
		5	78.813608	79.009382	78.821801	78.821801	78.821801	78.821801	78.821801	78.821801	80.166419
		8	76.423727	77.501118	77.460440	77.230554	77.881480	76.528798	76.274886	77.230554	79.249595
		10	77.083338	76.635097	77.466857	76.998507	75.154058	76.045796	78.017429	76.760354	78.758977
Image 4	Kapur	2	78.100314	78.100314	78.100314	78.100314	78.100314	78.100314	78.100314	78.100314	78.100314
		5	75.909814	76.042515	76.085644	76.134706	76.201580	76.201580	76.201580	76.741091	79.221426
		8	69.894840	69.974942	71.949848	67.677496	70.562302	71.018802	71.388941	70.037700	70.255357
		10	68.706228	65.667785	66.289437	67.162749	67.578108	67.306149	67.196939	67.061353	66.216111
	Otsu	2	82.161905	82.161905	82.161905	82.161905	82.161905	82.161905	82.161905	82.161905	81.657816
		5	74.781423	71.692465	71.692465	71.692465	74.781423	71.694488	74.781423	71.692465	70.391586
		8	67.428402	67.348063	66.820822	67.348063	68.891963	66.055139	67.256012	67.348063	70.289545
		10	63.888869	65.545047	66.062033	65.750524	64.710628	66.505048	65.774101	65.975582	66.809983

showed limitations in terms of converging in higher value of threshold. Even with such limitation ChOA didn't deviate from the best result given by other algorithms. So, some adjustment can improve this potential algorithm to outperform all algorithms in this application. One such improvement can be done in improving the convergence of ChOA in image threhsolding and improving the exploration of ChOA such that search agents don't get trapped in local optimum point in image thresholding application. Keeping this as a springboard of subsequent research, Chimp Optimization Algorithm can be regarded as a vital tool in image processing tasks that requires image clustering and multilevel thresholding based techniques, which may extensively used in fields like: Medical imaging, disease detection, Satelite imaging, underwater image segmentation, agricultar sector and so on. The strength of this algorithm is its ability to

Table 15 NIQE Values

		K	PSO	WOA	SSA	HHO	MFO	GWO	AOA	AVOA	ChOA
Image 1	Kapur	2	12.690589	12.690589	12.690589	12.690589	12.690589	12.690589	12.690589	12.690589	12.605161
		5	8.431372	8.474302	8.416693	8.416693	8.487638	8.487638	8.416693	8.416693	9.554108
		8	7.599817	7.414079	7.344207	7.414079	7.375476	7.832244	7.784895	7.541709	7.537691
		10	7.298654	7.180083	7.531204	7.306799	7.195615	7.322891	7.022125	7.324398	6.380568
	Otsu	2	11.015396	11.015396	11.015396	11.015396	11.015396	11.015396	11.015396	11.015396	11.015396
		5	9.277213	9.158389	9.156650	9.156650	9.158389	9.155717	9.156650	9.156650	9.412136
		8	8.018206	7.897808	8.075584	7.914684	7.787260	7.338613	7.702348	7.584185	7.938624
		10	7.232905	7.588708	7.669821	7.373784	7.921728	7.880181	7.573669	NaN	7.2928538
Image 2	Kapur	2	16.600981	16.600981	16.600981	16.600981	16.600981	16.600981	16.600981	16.600981	17.076540
		5	15.568347	15.318708	14.927598	15.568347	15.568347	15.119358	15.568347	15.147642	14.670107
		8	13.624276	13.819721	13.612558	13.309971	13.012649	12.875449	13.353471	13.650584	13.858738
		10	12.582878	11.967544	12.896179	12.926069	12.418457	11.903346	12.379177	12.320520	11.980013
	Otsu	2	18.246331	18.246331	18.246331	18.246331	18.246331	18.246331	18.246331	18.246331	18.246331
		5	15.362327	15.584403	15.153128	15.153128	15.153128	15.362327	15.153128	15.154227	15.116049
		8	12.685410	12.721154	12.823890	13.016140	13.025352	12.630501	12.055146	12.744823	12.697156
		10	11.710033	12.358268	12.377702	12.49515	11.812346	11.855017	12.037243	12.294211	11.945242
Image 3	Kapur	2	13.221289	13.221289	13.221289	13.221289	13.221289	13.221289	13.221289	13.221289	13.221289
		5	10.835557	11.945607	9.679653	10.58388	11.953600	11.947049	10.69437	11.953600	10.956103
		8	10.827368	10.772011	10.041925	10.551097	10.232425	10.099727	10.908308	10.924787	11.166871
		10	9.679320	10.127578	9.884964	10.065292	9.875765	9.709259	9.463496	9.998562	10.754164
	Otsu	2	13.467866	13.467866	13.467866	13.467866	13.467866	13.467866	13.467866	13.467866	13.552009
		5	10.593847	10.546547	10.587649	10.587649	10.587649	10.587649	10.58764	10.587649	10.215918
		8	10.723455	10.246826	10.132994	10.217441	9.8482032	10.838086	10.590114	10.217441	10.079855
		10	10.475210	10.424366	10.441787	10.437724	10.394407	10.574333	9.953972	10.315368	9.878908
Image 4	Kapur	2	12.670875	12.670875	12.670875	12.670875	12.670875	12.670875	12.670875	12.670875	12.670875
		5	9.134000	8.967988	8.727195	9.793276	8.970005	8.970005	8.970005	9.554190	9.995601
		8	8.951120	8.671288	8.527100	8.794746	9.034496	8.508955	8.466784	8.688792	8.507614
		10	8.254413	8.221178	8.637974	8.323878	8.009598	8.856883	7.358278	8.113561	7.898697
	Otsu	2	10.096202	10.096202	10.096202	10.096202	10.096202	10.096202	10.096202	10.096202	10.088656
		5	9.966926	9.459786	9.459786	9.459786	9.966926	9.466435	9.966926	9.459786	9.317724
		8	8.569605	8.289457	8.265960	8.289457	8.748086	8.168667	8.202904	8.289457	8.465271
		10	8.270119	7.834744	7.897771	8.048213	8.340841	7.517954	8.270765	8.084830	7.570792

provide competitive performance in terms of image quality and image clustering metrics, specially in lower level thresholding. The main weakness is its inability of fast convergence for higher level of thresholding applications, although the results denote significant competitiveness.

Funding The authors did not receive any funding for this study.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this article.

References

Abdollahzadeh B, Gharehchopogh FS, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 158:107408. ISSN 0360–8352. https://doi.org/10.1016/j.cie.2021.107408

Aldahdooh A, Masala E, Van Wallendael G, Barkowsky M (2018) Framework for reproducible objective video quality research with case study on PSNR implementations. Dig Signal Process 77:195–206

Barik D, Mondal M (2010) Object identification for computer vision using image segmentation. In: 2010 2nd international conference on education technology and computer, pp V2-170-V2-172. https://doi.org/10.1109/ICETC.2010.5529412

Bezdek JC, Ehrlich R, Full W (1984) FCM: the fuzzy c-means clustering algorithm. Comput Geosci 10:191–203

Biogeography-Based Optimization Algorithm and its application to clustering optimization and medical image segmentation. In: IEEE Access 7:28810–28825, 2019. https://doi.org/10.1109/ACCESS.2019.2901849.67, ISSN 0965-9978

Borsotti M, Campadelli P, Schettini R (1998) Quantitative evaluation of color image segmentation results. Pattern Recognit Lett 19(8):741–747. https://doi.org/10.1016/S0167-8655(98)00052-X (**ISSN 0167-8655**)

Borsotti M, Campadelli P, Schettini R (1998) Quantitative evaluation of color image segmentation results. Pattern Recognit Lett 19(8):741–747 (**ISSN 0167-8655**)

Brajevic I , Tuba M, Bacanin N (2012) Multilevel image thresholding selection based on the Cuckoo Search Algorithm. Pankaj Upadhyay, Jitender Kumar Chhabra

Chuang KS, Tzeng HL, Chen S, Wu J, Chen TJ (2006) Fuzzy c-means clustering with spatial information for image segmentation. Comput Med Imaging Graphic 30(1):9–15

Demirci R, Güvenç ve U, Kahraman H (2014) ”GÖRÜNTüLERİN RENK UZAYI YARDIMIYLA AYRIŞTIRILMASI”, İleri Teknoloji Bilimleri Dergisi, c. 3, sayı. 1, ss. 1-8, Ağu

Demirhan A, Törü M, Güler I (2015) Segmentation of tumor and edema along with healthy tissues of brain using wavelets and neural networks. IEEE J Biomed Health Inf 19:1451–1458

Dhiman Gaurav (2021) SSC: a hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. Knowl Based Syst 222. https://doi.org/10.1016/j.knosys.2021.106926 (**ISSN 0950–7051**)

Djerou L, Khelil N, Dehimi HE, Batouche M (2009) Automatic multilevel thresholding using binary particle swarm optimization for image segmentation. In: International conference of soft computing and pattern recognition 2009, pp 66–71. https://doi.org/10.1109/SoCPaR.2009.25

Farshi T, Drake JH, özcan E (2020) A multimodal particle swarm optimization-based approach for image segmentation. Expert Syst Appl 149:113233 (**ISSN 0957-4174**)

Gao H, Dou L, Chen W, Xie G (2011) The applications of image segmentation techniques in medical CT images. In: Proceedings of the 30th Chinese control conference, pp 3296–3299

Haralick RM, Kelly GL (1969) Pattern recognition with measurement space and spatial clustering for multiple images. Proc IEEE 57(4):654–665. https://doi.org/10.1109/PROC.1969.7020

Hashim FA, Hussain K, Houssein EH et al (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51:1531–1551

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872 (**ISSN 0167-739X**)

Houssein Essam H, Emam Marwa M, Ali Abdelmgeid A (2021) An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2021.115651 (**ISSN 0957–4174**)

Jia H, Ma J, Song W (2019) Multilevel thresholding segmentation for color image using modified moth-flame optimization. IEEE Access 7:44097–44134. https://doi.org/10.1109/ACCESS.2019.2908718

Jolion J-M, Meer P, Bataouche S (1991) Robust clustering with applications in computer vision. IEEE Trans Pattern Anal Mach Intell 13(8):791–802

Kaidi W, Khishe M, Mohammadi M (2022) Optimization dynamic levy flight chimp, systems knowledge-based. ISSN 235235:107625. https://doi.org/10.1016/j.knosys.2021.107625 (**ISSN 0950-7051**)

Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, and Image Processing 29(3):273–285. https://doi.org/10.1016/0734-189X(85)90125-2 (**ISSN 0734-189X**)

Kapur’s entropy based optimal multilevel image segmentation using Crow Search Algorithm. Appl Soft Comput 97(Part B):105522, 2020 ISSN 1568-4946

Kaur M, Kaur R, Singh N et al (2021) SChoA: a newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications. Eng Comput. https://doi.org/10.1007/s00366-020-01233-2

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95—International Conference on Neural Networks, pp 1942–1948, vol 4. https://doi.org/10.1109/ICNN.1995.488968

Kharrich M, Mohammed OH, Kamel S, Aljohani M, Akherraz M, Mosaad MI (2021) Optimal design of microgrid using chimp optimization algorithm. In: 2021 IEEE international conference on automation/XXIV congress of the Chilean Association of Automatic Control (ICA-ACCA), pp 1–5. https://doi.org/10.1109/ICAACCA51523.2021.9465336

Khishe M, Mosavi MR (2020) Classification of underwater acoustical dataset using neural network trained by Chimp Optimization Algorithm. Appl Acoust. https://doi.org/10.1016/j.apacoust.2019.107005 (**ISSN 0003-682X**)

Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338 (**ISSN 0957-4174**)

Khishe M, Nezhadshahbodaghi M, Mosavi MR, Martín D (2021) A weighted Chimp Optimization Algorithm. IEEE Access 9:158508–158539. https://doi.org/10.1109/ACCESS.2021.3130933

Kiani H, Safabakhsh R, Khadangi E (2009) Fast recursive segmentation algorithm based on Kapur’s entropy. In: 2009 2nd international conference on computer, control and communication, pp 1–6. https://doi.org/10.1109/IC4.2009.4909269

Lanthier Y, Bannari A, Haboudane D, Miller JR, Tremblay N (2008) Hyperspectral data segmentation and classification in precision agriculture: a multi-scale analysis. In: IGARSS 2008–2008 IEEE international geoscience and remote sensing symposium, pp II-585-II-588. https://doi.org/10.1109/IGARSS.2008.4779060

Liu J, Yang Y-H (1994a) Multiresolution color image segmentation. IEEE Trans Pattern Anal Mach Intell 16:689–700

Liu J, Yang Y-H (1994b) Multiresolution color image segmentation. IEEE Trans Pattern Anal Mach Intell 16(7):689–700. https://doi.org/10.1109/34.297949

Lu X, Zhang M (2010) The animation and comics content retrieval model based on analysis of clustered group. In: International conference on biomedical engineering and computer science 2010, pp 1–4. https://doi.org/10.1109/ICBECS.2010.5462355

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. University of California Press, Oakland, pp 281–297

MATLAB (2021) 9.10.0.1602886 (R2021a). Natick, Massachusetts: The MathWorks Inc

Mirjalili S (2014) Seyed Mohammad Mirjalili, Andrew Lewis, Grey Wolf optimizer. Adv Eng Softw 69:46–61 (**ISSN 0965-9978**)

Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006 (**ISSN 0950-7051**)

Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili Seyed Mohammad (2017) Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191 (**ISSN 0965-9978**)

Mirjalili S, Lewis A (2016) The whale optimization algorithm, advances in engineering software, volume 95, p 51-X **(Zhang, D. Wang and H. Chen, Improved)**

Mittal A, Moorthy AK, Bovik AC (2012) No-reference image quality assessment in the spatial domain. IEEE Trans Image Process 21(12):4695–4708. https://doi.org/10.1109/TIP.2012.2214050

Mittal A, Soundararajan R, Bovik AC (2013) Making a “Completely Blind. Image Quality Analyzer”. IEEE Signal Process Lett 20(3):209–212. https://doi.org/10.1109/LSP.2012.2227726

Muthukrishnan R, Radha M (2011) Edge detection techniques for image segmentation. Int J Comput Sci Inf Technol 3(6):259

Nagadurga T, Narasimham PVRL, Vakula VS, Devarapalli R, Márquez FPG (2021) Enhancing global maximum power point of solar photovoltaic strings under partial shading conditions using chimp optimization algorithm. Energies 14:4086. https://doi.org/10.3390/en14144086

Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66. https://doi.org/10.1109/TSMC.1979.4310076

Ouadfel S, Taleb-Ahmed A (2016) Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study. Expert Syst Appl 55:566–584

Pedram HBS, Pashaei E (2021) Data clustering using chimp optimization algorithm. In: 2021 11th international conference on computer engineering and knowledge (ICCKE), pp 296–301. https://doi.org/10.1109/ICCKE54056.2021.9721483

Pei Z, Zhao Y, Liu Z (2009) Image segmentation based on differential evolution algorithm. In: International conference on image analysis and signal processing 2009, pp 48–51. https://doi.org/10.1109/IASP.2009.5054643

Rahkar Farshi TK, Ardabili A (2021) A hybrid firefly and particle swarm optimization algorithm applied to multilevel image thresholding. Multim Syst 27:125–142

Rahkar Farshi T, Demirci R, Feizi-Derakhshi MR (2018) Image clustering with optimization algorithms and color space. Entropy (Basel) 20(4):296. https://doi.org/10.3390/e20040296 (**PMID: 33265387; PMCID: PMC7512815**)

Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis. In: International conference on machine learning, pp 1060–1069. PMLR

Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25:1077–1097. https://doi.org/10.1007/s00521-014-1597-x

Sharma A, Chaturvedi R, Dwivedi U, Kumar S, Reddy S (2018) Firefly algorithm based Effective gray scale image segmentation using multilevel thresholding and Entropy function. Int J Pure Appl Math 118

Tianqing H, Khishe M, Mohammadi M, Parvizi G-R, Taher SH, Karim TA (2021) Rashid real-time, COVID-19 diagnosis from X-ray images using deep CNN and extreme learning machines stabilized by chimp optimization algorithm. Biomed Signal Process Control 68:102764. ISSN 1746-8094. https://doi.org/10.1016/j.bspc.2021.102764

Venkatanath N, Praneeth D, Maruthi Chandrasekhar Bh, Channappayya SS, Medasani SS (2015) Blind image quality evaluation using perception based features. In: 2015 twenty first national conference on communications (NCC), pp 1–6. https://doi.org/10.1109/NCC.2015.7084843.

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612. https://doi.org/10.1109/TIP.2003.819861

Wang Z, Ma Y, Cheng F, Yang L (2010) Review of pulse-coupled neural networks. Image Vis Comput 28(1):5–13

Wang J, Khishe M, Kaveh M et al (2021) Binary Chimp Optimization Algorithm (BChOA): a new binary meta-heuristic for solving optimization problems. Cogn Comput 13:1297–1316. https://doi.org/10.1007/s12559-021-09933-7

Wong MT, He X, Yeh W (2011) Image clustering using Particle Swarm Optimization. In: IEEE congress of evolutionary computation (CEC) 2011, pp 262–268. https://doi.org/10.1109/CEC.2011.5949627

Yan Z, Zhang J, Yang Z, Tang J (2021) Kapur’s entropy for underwater multilevel thresholding image segmentation based on whale optimization algorithm. In: IEEE access, vol 9, pp 41294–41319. https://doi.org/10.1109/ACCESS.2020.3005452

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.