Expert Systems With Applications 166 (2021) 114063

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

CSFCM: An improved fuzzy C-Means image segmentation algorithm using a cooperative approach

Hamza Abdellahoum *, Nassim Mokhtari, Abderrahmane Brahimi, Abdelmadjid Boukra

LSI Laboratory, Department of Computer Science, University of Sciences and Technologies Houari Boumedienne, BP 32 El Alia 16111, Bab Ezzouar, Algiers, Algeria

ARTICLE INFO

Keywords:
Classification
Segmentation
Fuzzy c-means
Genetic algorithm
Biogeography based algorithm
Firefly algorithm

ABSTRACT

Fuzzy c-means (FCM) is one of the most widely used classification algorithms specially in image segmentation. Like any algorithm, FCM has some drawbacks such as the choice of the number of clusters and the cluster's center initialization. In this work, we propose new approaches to deal with these two drawbacks. We propose for the first problem two approaches. The first proposed approach exploits neural networks and the Xie and Beni index, while the second one exploits the histogram. Concerning the second problem, we propose a new metaheuristics cooperation approach using the Genetic Algorithm (GA), Biogeography Based Algorithm(BBO), and Firefly Algorithm (FA). This cooperation is managed by a multi-agent system allowing to determine automatically the fittest metaheuristics parameters. Finally, we propose to use a histogram-based version of FCM to reduce the execution time of the algorithm. Experimental results show that our proposed approach improves the performance of the basic FCM algorithm and outperforms other methods proposed in the literature.

1. Introduction

Image processing is in the intersection of several fields such as signal processing, mathematics and computer science. The phenomenal progress of computer science and especially artificial intelligence has greatly benefited image processing and analysis. One of the most important tasks of image processing is segmentation. Many segmentation methods were developed. We can cite thresholding based methods, contouring based methods, edge-based methods, semantic segmentation where each set of pixels is associated with a category, panoptic segmentation that is the result of the hybridization between semantic and instance segmentation, in whose each pixel has a label and object instances are uniquely segmented, classification methods etc. The classification method encompasses two families: supervised classification and unsupervised classification. In this work, we have opted for the segmentation because it is more general than the semantic and panoptic segmentation which is specialized in a particular field. For each domain, we have to redo the training. In this paper, we are interested in a technique of the second family (unsupervised classification) called Fuzzy c-means (FCM), one of the most widely used technique (Aiguo & Yunjie, 2019; Miao, Zhou, & Huang, 2020; Ren et al., 2019; Xia, Lin, & Li-Hua, 2019; Xiong, Tang, Chen, Hu, & Chen, 2020; Xu et al., 2020).

The fuzzy logic was proposed by Zadeh (1965) in 1965. Bezdek proposes in 1981 to merge Fuzzy logic with K-means (MacQueen, 1967) to produce FCM (Bezdek, 1981). The major drawbacks of FCM are sensitivity to noise, the dependence of cluster number choice and random clusters centers initialization. It is worth mentioning that the number of clusters is set manually by the user. In this paper, we aim to solve two major problems. The first one is the choice of the number of the cluster, the second one is to find the near-optimal initial clusters centers. Most of the methods use spatial image information that will be expensive in execution time, therefore we propose to use the histogram instead of the spatial information, which reduces the execution time.

Several works have been proposed to find the best initial cluster centers using metaheuristics to ensure a good quality of segmentation. Pant, Chinta, and Tripathy (2019) carried out a comparative study between FCM and one of the derivatives called IFCM (intuitionistic fuzzy C-means). The two methods are hybridized with fuzzy FA and FA. Three different images were used in the experimentation, and two validation indices were used to judge the quality of the result. The results prove that the hybridization of IFCM with fuzzy FA exceeded FCM, IFCM, FCM-FA, and IFCM-FA. In Kumar, Fred, Kumar and Varghese (2019), authors have proposed coupling FCM with FA, in order to find the near-optimal cluster centers. They Applied the NLTD

* Corresponding author.
E-mail addresses: habdellahoum@usthb.dz (H. Abdellahoum), nassim.mokhta@gmail.com (N. Mokhtari), abdou.brahimi3@gmail.com (A. Brahimi), aboukra@usthb.dz (A. Boukra).

https://doi.org/10.1016/j.eswa.2020.114063
Received 2 May 2020; Received in revised form 24 September 2020; Accepted 25 September 2020
Available online 28 September 2020

filter (Kumar, Fred, Kumar and Varghese, 2018) before segmentation to denoise the image. Experimental tests were performed on CT/MR (abdomen datasets) comparing Firefly-FCM with FCM-Cuckoo, FCM-ABC, and FCM-SA. The results showed that the proposed method outperformed the others. Haghdoost, Abadi, and Abedini (2015) uses Particle Swarm Optimization (PSO) to eliminate FCM faults such as low convergence rate, local optimum, and initialization sensitivity. Wang, Fang, Li and Wang (2017) used PSO to find the near-optimal cluster centers. On the other hand, Chen and Fang (2008) preferred to use another variant of PSO, named FPSO (Fuzzy PSO), to obtain the near-optimal cluster centers initialization, and evaluate it with a new fitness function that combines fuzzy cluster validity indices. A new Fuzzy C-Means algorithm based on PSO and Mahalanobis distance are developed by Liu, Yih, Lin, and Liu (2009) to improve the limits of the Gath-Geva (GG) and Gustafson–Kessel (GK) algorithms. Experimental results show that the performance of the proposed approach named PSO-FCM-M is better than those of the FCM, GG, GK algorithms. The authors of Kumar, Reddy and Rao (2018) were interested in detecting changes in Sarimages. They hybridize the artificial Bee colony metaheuristics (ABC) with FCM, for efficient image smoothing to make a decisive image classification. A comparative study between the different swarm optimization algorithms coupled with FCM (GA, PSO, ABC, GSA (Gravitational Search Algorithm), BAT, GWO (gray Wolf optimization)) was carried out by Singh, Laishram, and Roy (2019). From the precedent experiments, we note that the ABC algorithm consumes more CPU time, but gives better results. ABC algorithm was used by Balasubramani and Marcus (2013) to perform a hybridization with FCM called ABC-FCM where ABC algorithm is used to optimize the fuzzy clustering process and to detect tumors. The results of the experiments are validated by the size of the tumor. A comparison with the standard FCM and the watershed method was made denoting the superiority of ABC-FCM. Ouadfel and Meshoul (2012) proposed a hybridization of ABC and DE named MOABC. The proposed method was evaluated by applying experimental tests on 6 images often used in segmentation. MOABC was compared to MABC-FCM, ABC-FCM, PSO-FCM and FCM, the results show that MOABC and MABC have almost the same results surpassing all the others. Dolphin swarm optimization (DSA) has been exploited by Qiao and Yang (2020) to solve some FCM problems. In this work, the authors used a variant of FCM named Kernel FCM where a metaheuristic was used to find the near-optimal cluster centers. The experimental study consists of comparing the proposed method (KFCMDSA) with DSA-FCM, WOA-FCM, AGA-FCM, APSO-FCM, WSA-FCM and CSA-FCM. The results show that KFCMDSA exceeds other works. A new kidney inspired metaheuristic named FCM-KA has been used by Nayak, Vakula, Dash, and Naik (2019) in order to improve the quality of FCM by selecting the near-optimal cluster centers, to obtain a segmentation with high quality. Experimental tests were performed returning promising results by the proposed method which confirms its robustness. Kumar, Dwivedi and Jangam (2019) proposes a hybrid technique using a min–max classifier, FCM, and a Bat optimization algorithm to solve the problem of unknown cluster numbers, initialization of cluster centers and local convergence of the FCM algorithm. From experimentation, it is observed that the proposed hybrid technique is more efficient than the original FCM. Zhang, Jiang, Zhou, Xue, and Chen (2017) proposed to hybridize six variants of BBO with FCM. Experiments show that the variant named EBO-FCM gives the best cluster centers initialization. One of the recent works for the initiation of cluster centers by metaheuristics was proposed in Wang, Zhang, Xiao and Li (2017). In this work, the authors use fruit Fly with FCM to find the near-optimal center initialization to ensure a good segmentation. The evaluation of his proposal was carried out on three datasets (iris, glass, seeds), and compared with two basic methods FCM and KFCM. Experimental results show that the proposed approach exceeds the other two methods. AFSA (artificial fish swarm algorithm) inspired by the collective and social behavior of natural fish are used by Chu, Zhu, Shi, and Song (2010) and He, Belacel, Hamam, and Bouslimani (2009) to solve the issue of the local optimum problem of the FCM algorithm and choose a good initialization of the clusters centers. Other hybridizations of the AFSA are presented to resolve other segmentation problems like multi-threshold segmentation problem (Jiang, Mastorakis, Yuan, & Lagunas, 2009). In Padmavathi, Eswaramurthy, and Revathi (2018) the authors tried to solve the problem of FCM local optima, through the use of FSSO (fuzzy spider swarm optimization) which has been modified and adapted to this problem. Experiments are conducted over five datasets such as Iris flower, Glass, CMC, Vowel, and Wine revealing encouraging results for solving the clustering problem.

Other works have been proposed to improve FCM in terms of execution time and noise sensitivity, concerning the execution time, Samina, Hammad, and Humayun (2010) exploited the image histogram instead of the image. This allows us to accelerate the segmentation process concerning noise sensitivity improvement, many works were proposed. Yang, Zheng, and Lin (2001) proposed to use Possibilistic Fuzzy C-Means (PFCM). The main idea of this method is based on the integration of the penalty term in the objective function of FCM. This method was inspired by the Neighborhood Expectation-Maximization (NEM) algorithm. The author aims to reduce noise during segmentation. Ahmed, Yamany, Mohamed, Farag, and Moriarty (2002) proposed the Bias-Corrected Fuzzy C-Means (BCFCM) method which takes into account points in the pixel's neighborhood. This new information reduces impulse noise (BCFCM has been tested on MRI images). Without forgetting the well-known Spatial Fuzzy C-Means method for this problem suggested by Chuang, Tzeng, Chen, Wu, and Chen (2006). The authors used spatial information in their approach. Unlike other variants, SFCM does not change the FCM objective function, but it changes the formula giving the membership degree. Other methods for improving the performance of FCM exist. They modify the objective function by introducing spatial information and using neighborhood or using other distances instead of Euclidean distance (Chen & Zhang, 2004; Gustafson & Kessel, 1978; Krinidis & Chatzis, 2010; Pham, 2002; Pham & Prince, 1999). MMTDFCM is a version of FCM based on MMTD (Measure of Medium Truth Degree) (Hong, Xiao, & Zhu, 2006, 2007) was proposed by Zhou, Yang, & Shaobai, to improve FCM in presence of noise (Zhou, Yang, & Zhang, 2014).

Other works were proposed to solve the problem of choosing the number of clusters. There is no method determining the right value of the number of clusters. The choice of this value is either made by the user, or it is estimated approximately (Gordon, 1999). Works done in this problem are classified into two classes: global methods and local methods. In the global methods, the quality of partitioning, given a number c of partitions, is measured using a criterion or an index and the optimal number of partitions is obtained by comparing the values of this index for different numbers of partitions c. In this class of methods, the algorithm is executed several times to find the fittest value of the used index increasing, however that it is time-consuming. Among the methods of this class, we find the Calinski and Harabasz method (Calinski & Harabasz, 1974) which gives good results. Milligan and Cooper (1985) made a comparative study between 30 methods determining the value of c. They found that the method of Calinski and Harabasz surpassed all other methods. Another proposal, called GAP, is developed by Tibshirani, Walther, and Hastie (2001). The efficiency of this method is based on its generalization. Indeed, it can be applied to all partitioning techniques and all distance measurements. The Xie-Beni (XB) validation method (Xie & Beni, 1991) which has been used in the evaluation of the segmentation quality of the FCM technique, exploits the degrees of belonging of each pixel to all cluster centers. The second class of works named local methods tests whether a pair of partitions should be merged so they are better suited for hierarchical partitioning algorithms. Among the best-known works in this class, we find Duda and Hart's method (Duda & Hart, 1973). The authors use the null hypothesis stating that the ith cluster is homogeneous, they compare it to the hypothesis stating that the cluster can be divided into

2 clusters. The test is based on the comparison of the inter-cluster sum of the ith cluster with the inter-cluster sum of the subdivided ith cluster. A modified Duda and Hart approach has been proposed by Beale (1969) where the authors use F-statistics rather than a null hypothesis test. Several works have been done to solve the problem of fixing the fittest degree of fuzziness value noted m. Pal and Bezdek (1995) proposed to fix the m value in the interval [1.5; 2.5]. Ozkan and Turksen (2007) have shown that a good degree of fuzziness m belongs to [1.4; 2.6]. Usually, we use the value 2 for the degree of fuzziness. This choice is based on empirical results. However, there are theoretical studies (Jing, Deng, & Yu, 2014; Yu, Cheng, & Huang, 2005) defining the value of the degree of fuzziness, but they are limited and do not cover all cases.

In this paper, we are interested in solving two major problems: the determination of the number of clusters and the initialization of the cluster centers. In this work, we have tested our proposal on the grayscale images dataset. For the number of clusters, we propose two different methods. The first one based on the use of the histogram information, and the second using a neural network. For the initialization of the cluster centers, we propose a new cooperative approach that manipulates three metaheuristics with a new system.

The rest of this document is organized as follows. Section 2 presents an overview of the background (FCM, FA, GA, BBO, and cooperation). Section 3 describes the proposed approach. Experimental results and comparisons are presented in Section 4, followed by a conclusion in Section 5.

2. Backgrounds

2.1. Fuzzy C-means

Introduced by Dunn in 1973 and improved by Bezdek in 1981, fuzzy c-means (FCM) is an unsupervised classification algorithm based on fuzzy logic. Compared to classical partitioning methods such as k-means where each data points belongs to a single partition, in FCM a point belongs to all classes with a degree of membership, varying between 0 and 1. The clusters are represented by their centers.

The centers are calculated as follows:

$$c_i = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m} \tag{1}$$

where N represents the total number of data points and x_j refers to the jth data points, m (m > 1) is the fuzzifier (degree of fuzziness) and u_{ij} is the membership degree of the jth pixel that belongs to the cluster i. u_{ij} is given by formula (2).

$$u_{ij} = \frac{1}{\sum_{l=1}^{nc} \frac{d_{ij}^{\frac{2}{m-1}}}{d_{lj}}} \tag{2}$$

where d_{ij} represents Euclidian distance between the jth pixel and the cluster i, d_{lj} is the Euclidian distance between the jth pixel and cluster l and c is the number of clusters. The optimal clusters centers and membership degrees are calculated by minimizing the objective function (3):

$$j = \sum_{j=1}^{N} \sum_{i=1}^{nc} u_{ij}^m * d^2(x_j, c_i) \tag{3}$$

Under constraints (4):

$$\begin{cases} \forall j \in [1, N], \forall i \in [1, nc] : u_{ij} \in [0, 1] \\ \forall j \in [1, N] : \sum_{i=1}^{nc} u_{ij} = 1 \end{cases} \tag{4}$$

where N represents the data points, nc the number of clusters, u_{ij} is the membership degree of the jth pixel that belongs to the cluster i, $d^2(x_j, c_i)$ the distance between jth pixel and ith cluster, and m represents the degree of fuzziness ($1 < m \le \infty$).

Algorithm 1 FCM

C : number of cluster.
m : the degree of fuzziness (m>1).
ϵ : the error.
Initialize randomly the centers of clusters $c_i^{(0)}$
K=1
repeat
Calculate the membership $u_{ij}^{(k)}$ using the centers $c_i^{(k-1)}$:
$u_{ij} = \frac{1}{\sum_{l=1}^{c} \frac{d_{ij}^{\frac{2}{m-1}}}{d_{lj}}}$
Update the centers $c_i^{(k)}$ using $u_{ij}^{(k)}$
$c_i = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m}$
until $C_i^{(k)} - C_i^{(k+1)} < \epsilon$
Return $c_i^{(k)}$

2.1.1. Validation index

In this section, we describe the main used validation index. These indexes are used to measure the segmentation quality.

Peak-value signal-to-noise rate (PSNR)

The PSNR (Hore & Ziou, 2010) is a measure to be maximized.

$$PSNR = 10 * log \frac{x_{max}^2}{(\frac{1}{M*N}) \sum_{i=1}^{N} \sum_{j=1}^{N} [x(i,j) - y(i,j)]^2} \tag{5}$$

where x_{max} represents the maximal gray level value present in the image, M and N represent the image size, x(i,j) and y(i,j) represent respectively, the gray level values of the pixel (i,j) in the original image (to be segmented) and the processed image (image after segmentation) respectively.

Xie and Beni index (XB)

Xie and Beni's index, XB, given in formula (6), measures the ratio of the total variation within clusters to the separation of clusters.

$$XB = \frac{\sum_{(i=1)}^{c} \sum_{(k=1)}^{n} (u_{ik})^m \|x_k - C_i\|^2}{nMin\|C_k - C_i\|^2} \tag{6}$$

where u_{ik} represent the membership degree, n is the total number of data points, x_k refers to the kth data Point, m is the fuzzifier and C_i refers to the ith centers.

Subarea coefficient index (SC)

SC index given in formula (7), measures the ratio of the sum of compactness to the separation of the clusters:

$$sc = \sum_{i=1}^{c} \frac{\sum_{k=1}^{n} (u_{ik}^m) \|x_k - C_i\|^2}{N_i \sum_{j=1}^{c} \|C_j - C_i\|^2} \tag{7}$$

where C_i is the center of cluster i, N_i is the number of objects in the cluster I and m is the fuzzifier.

Classification entropy index (CE)

CE index (Hall & Kanade, 2005) (also named Partition entropy index (PE)) given in formula (8), measures the fuzziness of the cluster partition:

$$CE = -\frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{N} u_{ij} log(u_{ij}) \tag{8}$$

where u_{ij} represents the membership degree, N is the total number of data point and C is the number of centers.

Partition coefficient index (PC)

PC index (Bezdek, 1973) given in formula (9), measures the amount of overlapping between clusters.

$$PC = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^2 \quad (9)$$

where u_{ij} represents the membership degree of data j in cluster i, N is the total number of data point and C is the number of centers.

2.2. Firefly algorithm

FA is a metaheuristic developed by Xin-She Yang in 2008 at the University of Cambridge (Yang, 2008). It is inspired by the flashing behavior of fireflies. This metaheuristic is classified as a population-based method, where each firefly represents a solution. FA is based on three rules that are:

1. Fireflies are unisex, so one firefly is attracted to another without taking into consideration their gender.
2. Attractiveness is proportional to brightness, and both are inversely proportional to the distance (decreases as distance increases). For any pair of fireflies, the less bright one will be attracted to the brighter one, and so will move towards it. If there is no firefly brighter, it moves randomly.
3. The brightness of a firefly (solution) is modeled by the objective function.

The movement of a firefly i towards a brighter firefly j is defined as follows (10):

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha_t \tau_i^t \quad (10)$$

where α_t is a random factor, x_i^t the position of firefly i at time t and τ_i^t the vector of random numbers given by a Gaussian distribution at time t. α_t is given by the following formula (11):

$$\alpha_t = \alpha_0 \delta^t \qquad with \qquad 0 \le \delta \le 1 \quad (11)$$

where α_0 is the initial random factor, and δ is the cooling factor (decrease parameter). β (The variation in attractiveness) is defined by formula (12):

$$\beta = \beta_0 e^{-\gamma r^2} \quad (12)$$

where γ represents the absorption coefficient, β_0 attractivity at r = 0 and r is the Cartesian distance between two fireflies.

Algorithm 2 FA Algorithm

```
Randomly generate a population of n fireflies x_i =(i=1,..,n).
The light intensity Ii of each firefly xi is given by f(xi) where f is the
objective function
Define the absorption coefficient γ
WHILE (tMax_Generation) DO
    FOR i=1 TO n DO
       FOR j=1 TO n DO
          IF I_i < I_j THEN
           Moving firefly i to firefly j
          ENDIF
Vary the attractiveness according to the distance r.
Evaluate the new solution and update the light intensity
       ENDFOR
    ENDFOR
Determining the best solution
ENDWHILE
```

2.3. Genetic algorithm

GA was introduced around 1975 by Holland (1975). They are inspired by natural biological evolution. GA apply natural biological evolution to a population of individuals. Each individual is assigned a quality score obtained via an objective function. Genetic algorithms are based on three principles that are:

- Selection: it consists of selecting two parents in order to make a cross between them.
- Crossing: it consists of combining two parents to form one or two children.
- Mutation: it introduces a diversification in the research, by mutating certain characteristics of an individual.

There are several versions of the genetic algorithm. We present the population replacement version (Sevaux, 2004).

Algorithm 3 GA Algorithm

```
Randomly generate a population P of n solutions
repeat
  p'=∅
  repeat
    Selection of 2 solutions x and x' of P
    Crossing between the two parents x and x' to form two children
    y and y'.
    Mutate y and y' under certain conditions
    Add y and y' in P'
  until (|P'|=n)
  P= P'.
until the shutdown criteria are met
```

2.4. Biogeography based algorithm

Biogeography is the geographical distribution of biological organisms' study. It is the fruit of Alfred Wallace and Charles Darwin works in the 19th century. These works were mathematically modeled by Mac Arthur and Edward Wilson in 1960 (MacArthur & Wilson, 1967). They were interested in the distribution of species over habitats. A habitat represents a living space of species. The mathematical model developed models the migration of species from one habitat to another. The quality of the habitat is measured by a variable called HSI (Habitat suitability index). The characteristics of the habitat called SIV (suitability index variable) influences the quality of the habitat. The mathematical model takes into account the following facts:

- A habitat with high HSI contains a high number of species while a habitat with low HSI has a reduced number of species.
- The rate of emigration is high in high HSI habitats while the rate of immigration is low.
- Habitats with low HSI have a high immigration rate and a low emigration rate.
- The migration process tends to improve bad Habitats.

In 2008 (Simon, 2008) introduced a metaheuristic inspired by this model. This inspiration is based on the following analogies:

- The habitat corresponds to the solution.
- The HSI corresponds to the quality of the solution.
- The SIV corresponds to the components of the solution.
- Each solution has an emigration rate and an immigration rate.
- A good solution sends some of its characteristics (SIV) to a bad solution (emigration).
- A bad solution receives certain characteristics (SIV) from good solutions (immigration).

The immigration rate of a solution with k species is given by the following formula (13):

$$\lambda_k = I(1 - \frac{k}{N}) \tag{13}$$

The emigration rate of a solution with k species is given by the following formula (14):

$$\mu_k = E(\frac{k}{N}) \tag{14}$$

where I is the maximum immigration rate. E is the maximum emigration rate. N is the maximum number of species.

The migration process is described by Algorithm 4

Algorithm 4 Migration Algorithm

select H_i with probability $\alpha\lambda_i$
IF H_i is selected **Then**
FOR j=1 **To** n**Do**
select H_j with probability $\alpha\mu_j$
IF H_j is selected **Then**
then replace the SIV in H_i with the SIV of H_j
ENDIF
ENDFOR
ENDIF

Migration aims to intensify research. Diversification is ensured by the mutation process which consists in randomly modifying certain components of certain individuals. The mutation rate is given by the formula (15)

$$m_k = \alpha(\frac{1 - P_k}{p_{max}}) \tag{15}$$

where m_k is the mutation rate of a solution having k species, α is a user defined parameter. P_k is the probability to have k species. $Pmax$ is the probability to have max species.

P_k is given by the formula (16) (Daoudi, Boukra, & Ahmed-Nacer, 2011):

$$P_k = \begin{cases} \dfrac{\lambda_0 \ldots \lambda_{i-1}}{\mu_1\mu_2\ldots\mu_k(1+\sum_{i=1}^{n}\frac{\lambda_0\ldots\lambda_{i-1}}{\mu_1\mu_2\ldots\mu_i})} & 1 \le k \le n \\ \dfrac{1}{(1+\sum_{i=1}^{n}\frac{\lambda_0\ldots\lambda_{i-1}}{\mu_1\mu_2\ldots\mu_i})} & k = 0 \end{cases} \tag{16}$$

The mutation process is given by Algorithm 5

Algorithm 5 mutation Algorithm

FOR j= 1 **To** m **Do**
use l_i an μ_i to compute P_i
select SIV in $H_i(j)$with probability αP_i
IF $H_i(j)$ is selected
then replace $H_i(j)$ with random SI/V
ENDIF
ENDFOR

The optimization process of BBO consists in evolving a population of individuals using the migration and mutation processes. BBO algorithm is given in Algorithm 6

2.5. Cooperation of metaheuristics

Cooperation of metaheuristics is a paradigm allowing to benefit from the strength of different metaheuristics. It consists to solve a problem using several metaheuristics following a certain policy.

2.5.1. Levels of cooperation

There are two levels of cooperation which are(see Fig. 1):

Algorithm 6 BBO Algorithm

Randomly generate a population P of n solutions
WHILE the stop criterion is not met **DO**
Evaluate the HSI of each solution
Calculate the number of species S, the rate of immigration λ and emigration μ for each solution
FOR i=1 **TO** n **DO** Use λ_i to decide, in a probabilistic way, to migrate towards a solution i
IF rand(0,1) < λ_i **THEN**
FOR j=1 **TO** n **DO**
Use μ_j to decide probabilistically to emigrate from the solution j
IF rand(0,1)< μ_i **THEN**
Replace a randomly chosen variable in solution i with the variable in solution j
ENDIF
ENDFOR
ENDIF
ENDFOR
Mutation: mutating individuals
ENDWHILE

- **Low-Level**: In this level, the methods are organized in a way where one method encompasses another such as the use of local search in evolutionary algorithms.
- **High-Level**: In this level, methods operate independently, they can communicate while maintaining their integrity .

Within these levels there are two modes:

- **Relay mode**: Methods run sequentially one after the other and the result of one method is the starting point for the other.
- **Co-evolutionary mode**: In this mode, the methods are executed in a parallel manner.

We can combine the two levels with the two modes to obtain four classes of cooperation (see Fig. 2).:

- Low-level relays.
- High-level relays.
- Low-level co-evolution.
- Co-evolution high level.

3. Proposed method

In this section, we present the proposed approaches for the resolution of the FCM problems already mentioned in the introduction (number of clusters determination and initializing the cluster centers).

3.1. Number of clusters determination

The number of clusters is of great importance in segmentation because it influences the segmentation results. The right choice of the number of clusters ensures a good segmentation quality. We propose two approaches to determine the number of clusters. The first one uses a neural network and the second is based on the image histogram.

3.1.1. First approach using neural network (DNCNN)

The quality of neural network learning depends on the combination of hyper-parameters that make up the network (number of hidden layers, number of neurons per layer, activation function...etc.), we choose experimentally the best combination of hyper-parameters (the one that maximizes precision). The different values considered for each hyper-parameter are shown in Table 1:

The best architecture obtained is composed of 4 hidden layers whose characteristics are given in Table 2, an output layer with 9 neurons

Fig. 1. Different levels of cooperation.

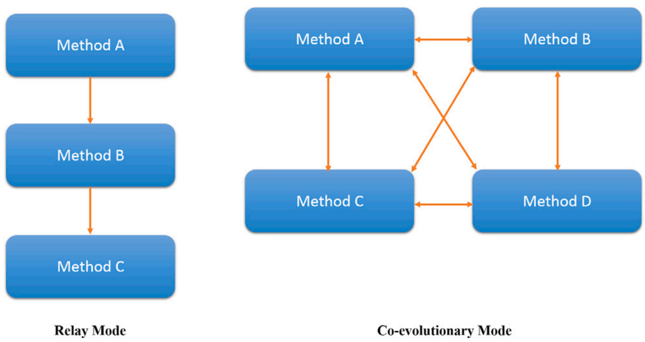


Fig. 2. Different types of cooperation.

Table 1
Neural network parameter.

Hyper-parameter	Values
Number of hidden layers	[2;5]
Number of neurons per layer	[100; 300]
Activation function	Sigmoid; Relu; Softmax

Table 2
Number of neurons per layer.

N° hidden layer	Number of neurons	Activation function
1	200	Relu
2	250	Relu
3	300	Relu

Table 3
Accuracy rate.

	Accuracy rate
Using the training package	96.15%
Using the test set	76.41%

(corresponding to the size of the output vector, where each table cell contains a value of XB) and "Softmax" as an activation function.

In aim to evaluate the quality of the Neural Network retained, we construct a Caltech 256 (Griffin, Holub, & Perona, 2006) dataset to experiment it. The accuracy rate obtained by this network is (see Table 3):

The training set is composed of 75% of the Caltech 256 dataset. The remaining 25% of the dataset represent the test set.

The main drawback of this approach is the fact that the neural output number is a fixed parameter that can be less than the number of clusters. For this reason, we propose the second approach based on the histogram. The advantage of this proposal is that the histogram of the image is used instead of the image. Histograms of images are not as varied as images.

3.1.2. Second approach using histogram information (DNCH)

To solve the problem of clusters number determination avoiding the precedent drawback of the neural network, we propose a method that

Fig. 3. Lena image.

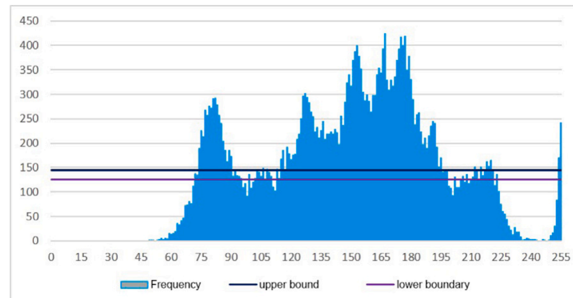


Fig. 4. Histogram of the "Lena" image.

exploits the gray level image histogram H. In this approach, we consider the intersection of the image histogram with a horizontal line having for equation y = b, b being a value to be set. We use a vector V of 256 cells. Each cell of V corresponds to a gray level and has the following value.

$$\begin{cases} V(i) = 1 & \text{if } H(i) = b, \\ V(i) = 0 & \text{otherwise.} \end{cases} \tag{17}$$

We propose to use the number of series of consecutive '1' in V as a number of clusters. The presence of '0' in the middle of a series of '1' can disturb this method (a single cluster will be considered in this case as two clusters). The value of the minimum number of consecutive 0 separating two clusters is set experimentally to 2. In this method, it is difficult to find the fittest value of b (in the equation y = b) giving the best number of clusters. For this we propose to use two curves of equations y = lb (low boundary) and y = hb (high boundary) where:

$$lb = [max(0, m - var)] \tag{18}$$

$$hb = [min(maxval, m - var)] \tag{19}$$

maxval: The highest value in the histogram. m: The average of the histogram values.

$$var = \sqrt{\frac{d(\frac{nbp}{nbG})}{nbG}} \tag{20}$$

- nbG represents the number of values greater than or equal to m
- nbP represents the number of values less than or equal to m and greater than 0.
- d is given by the following formula (21)

  $$d = \frac{\sum_i^n (\sqrt{(x_i - me)^2})^2}{n} \; with \; x > 0 \tag{21}$$

  where $x_i \in x / x = H^{(-1)}(lb) \; or \; x = H^{(-1)}(lb)$
- me represents the mean of all gray level frequencies in the histogram (see Figs. 3 and 4).

3.2. Initialization of cluster centers

In this section, we present the proposed cooperative approach resolving the cluster centers initialization problem

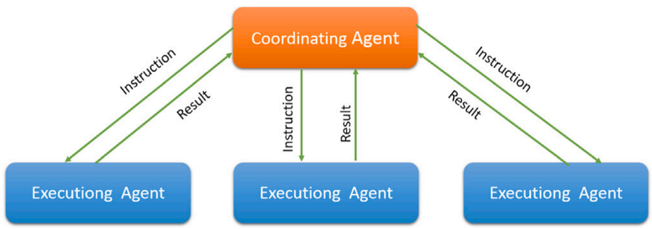


Fig. 5. Proposed cooperation.

Table 4
Messages meaning.

Message	Meaning
−1	Informs the executing agent that it has consumed its life time.
0	Corresponds to an intensification. It is sent when the result obtained by the agent is the best of the three agents.
1	Corresponds to a diversification, sent when result obtained by the agent is second best (better than one result, but worse than another)
2	Corresponds to a random generation of the agent parameters, sent when the result obtained by the agent is the worst of the three.

Table 5
The classification of parameters.

Metaheuristics	Diversification	Intensification
BBO	-Maximum likelihood of mutation -Mutation Coefficient F (in the case of a mutation DE)	Number of iterations
GA	Probability of mutation	Number of iterations
FA	$\alpha_0, \lambda, \delta$	-Number of iterations -β_0

3.2.1. Proposed cooperative system (CSFCM)

Metaheuristics are generic algorithms resolving Np-hard optimizing problems. It is known that there is no metaheuristic better than the others for any problem. In aim to benefit from the advantages of different metaheuristics, we propose to use metaheuristics in a cooperative way. The proposed cooperation takes the form of a multi-agent system, using two types of agents, coordinating agent and executing agent (Fig. 5).

A. Coordinating agent

The role of this agent is to coordinate the work of the execution agents (in our case three agents), sending them instructions depending on the latest results. Initially, the coordinating agent calculates the number of clusters (using one of the two precedent proposed methods) to be used for the image segmentation. Then it communicates the clusters number to the executing agents and launches them in a parallel way. During this phase, each execution agent is given a certain life time. If the executing agent improves the best obtained result in terms of Xie-Beni index, the Coordinating Agent resets its life time to the initial value, otherwise it decreases them. The coordinating agent sends a notification to the executing Agent if its time life becomes null (dead agent). At each iteration, the coordinating agent compares the results of alive executing agents and sends them messages depending on their state. Table 4 summarizes sent messages.

The coordinating agent continues running as long as there is an alive agent. Algorithm 7.

B. Executing agent

The role of an Executing Agent is to execute one of the three metaheuristics (BBO, GA, FA) and to transmit the result to the coordinating agent. Initially, the Executing Agent waits for the reception of the number of clusters (NbCluster) to use. Then, it randomly generates the parameters used by its metaheuristic, as well as the parameters that are specific to FCM, (i.e. the number of clusters to be used for segmentation in the range [max (2,Nbcluster-1); Nbcluster+1])

Algorithm 7 Coordinating Agent

Initialize the life time of each agent
Nbr-agent-alive = 3.
Calculate the number of clusters to be used using one of the proposed methods
Launch the three executing agents
Communicate the number of clusters to be used to the three agents
repeat
  Expect result
  IF (Xie-Beni(result) < Xie-Beni(best-result))**THEN**
  reset the agent's life time
  best-result=result
  ELSE
  decrement the agent's life time
  ENDIF
  IFlife time= 0 **THEN**
  Nb-agents-alive = Nb-agents-alive - 1
  ELSE
  Send new instruction
  ENDIF
until Nb-agents-alive= 0.
Return best result obtained

The execution agent reacts to the coordinating agent's message as follows:
Message = '−1': the coordinating agent terminates its execution,
Message = '0': the coordinating agent proceeds to an intensification by changing some parameters according to Table 5 (increase intensification parameters by 15% and decrease diversification parameters by 15%)
Message = "1": the coordinating agent proceeds to diversification by changing some parameters according to Table 5 (increase diversification parameters by 15% and decrease intensification parameters by 15%)
Message = "2": all parameters of this executing agent are randomly initialized. This happens in the case where an execution agent, that gives the best result, is unable to improve its own result. This allows to possibly, send later, '0' message to one of the other metaheuristics

The percentage of intensification/diversification increase or decrease (15%) was determined experimentally.

The parameters responsible of intensification and diversification for each metaheuristic are given in Table 5:

Algorithm 8 Executing Agent

Wait for NbCluster reception
Randomly generate the metaheuristic parameters of the agent.
Randomly generate the number of clusters to be used in the range of [max (2,nbcluster-1); nbcluster+1].
WHILE true **DO**
wait formessage
Switch(message):
Case -1: End the execution of the agent;
Case 0: Intensify;
Case 1: Diversification;
Case 2: Go to 2 /* random generation */
ENDSWITCH
execute the metaheuristic
Send the obtained result to the Coordinating agent.
ENDWHILE

In the precedent sections, we have proposed methods to estimate the number of clusters of an image and a near-optimal cluster center initialization. The following section describes the FCM algorithm based on histogram using results found by the precedent proposed approach.

3.2.2. FCM segmentation algorithm based on histogram

A. Data structure

For an image with N regions, a solution is represented by a table of the size N where the ith element represents the value of the center of the ith class, respecting the following condition (22):

$$Min \leq Solution[i] \leq Max, \forall i \in [1;N] \tag{22}$$

where Min represents the low gray level value available in the histogram, and Max is the high gray level value available in the histogram.

B. Population sort

The execution of the genetic algorithm, as well as BBO, requires a sorted population at each time. In order to meet these requirements, and preserve the performance of our algorithms, we choose to represent the population using a binary tree, sorted according to the quality of the solutions, which allows to have a sorted population with a low sorting cost. This choice avoids to sort the population at each iteration.

C. Population generation

In order to generate the different initial populations for the different metaheuristics, we opt for two different approaches:

- First approach: Random initialization of the solutions composing the population
- Second approach: Use BBO or the genetic algorithm to generate a population as diverse as possible.

In the case of the first approach. The components of each solution are generated according to the following formula (23):

$$Solution[i] = Random(Min, Max) \tag{23}$$

In the case of the second approach, metaheuristics do not seek to improve the quality of the solution, but rather to maximize diversity within the population. The generation of the population by metaheuristics (second approach), offers, generally, faster convergence towards a good solution quality.

Algorithm 9 FCM algorithm based on histogram

C : number of cluster.
m : the degree of fuzziness (m>1).
ϵ : the error.
Initialize randomly the centers of clusters $c_i^{(0)}$
K=1
repeat
Calculate the membership $u_{ij}^{(k)}$ using the centers $c_i^{(k-1)}$:
$u_{ij} = \frac{1}{\sum_{l=1}^{c} \frac{d(j-c_i^{k-1})}{d(j-c_l^{k-1})}^{\frac{2}{m-1}}}$
Update the centers $c_i^{(k)}$ using $u_{ij}^{(k)}$
$c_i^{(K)} = \frac{\sum_{j=0}^{255} f_j (u_{ij}^k)^m j}{\sum_{j=0}^{255} f_j (u_{ij}^k)^m}$
K=K+1
until $C_i^{(k)} - C_i^{(k+1)} < \epsilon$
Return (The optimal centers of clusters)

4. Experimentation

In this section, we will focus on the evaluation of the influence of the three proposed approaches on the FCM segmentation quality. We start firstly with tests on cluster number detection and then on segmentation. Hence we are going to carry out comparative tests of our approach (improved FCM) with other works that have tried to improve the FCM method

Table 6
Results obtained by the two proposed methods detecting the number of clusters applied to the noise-free images set.

Number of clusters found	Neural network based method (DNCNN)		Histogram based method (DNCH)	
	Number of image	Rates	Number of image	Rates
Dif = 0	31	0.287	99	0.917
Dif ≤ 1	63	0.583	102	0.944
Dif > 1	45	0.461	6	0.055

Table 7
Results obtained by the two methods proposed for the detection of the number of clusters applied to the set of images containing a salt & pepper noise of 10%.

Number of clusters found	Neural network based method (DNCNN)		Histogram based method (DNCH)	
	Number of image	Rates	Number of image	Rates
Dif = 0	25	0.231	36	0.333
Dif ≤ 1	74	0.685	49	0.454
Dif > 1	34	0.314	59	0.546

Table 8
Results obtained by the two methods proposed for the detection of the number of clusters applied to the set of images containing a salt & Gaussian of 10%.

Number of clusters found	Neural network based method (DNCNN)		Histogram based method (DNCH)	
	Number of image	Rates	Number of image	Rates
Dif = 0	31	0.287	101	0.935
Dif ≤ 1	63	0.583	104	0.963
Dif > 1	45	0.416	4	0.037

4.1. Cluster number detection experimentation

In this section, we compare the two proposed methods (histogram-based method DNCH and neural network-based method DNCNN). To carry out these tests, we generate a set of 108 images, with a known number of clusters varying between 2 and 10. We generate then from these images, a set of images with Gaussian noise and salt and pepper noise with 10% of noise.

To compare the proposed Neural network based method with the proposed Histogram based method we calculate Dif = |number of image clusters– number of image clusters found by the method|

Noise-free synthetic images test

From Table 6, we notice that, for Noise-free synthetic images, the Histogram based method offers better results than the Neural network-based method in all cases. Therefore, the histogram based method is better for Noise-free images

Synthetic image test with 10% salt & pepper noise

From Table 7, we notice that the histogram based method offers better results for the first category (Dif = 0) whereas the Neural network-based method gives better results in the second category. Then, globally, the Neural network-based method is more interesting for images with salt & pepper noise.

Synthetic image test with 10% Gaussian noise

From Table 8, we notice that, for Gaussian noise synthetic images, the histogram based method offers better results than the Neural network-based method in all cases. Therefore, the histogram based method is better for Gaussian noise images.

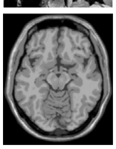
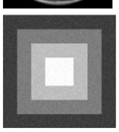
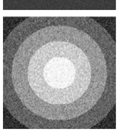
In what follows, we illustrate a summary table containing the results of the two methods. We have grouped all the images (noiseless, Gaussian noise and salt & pepper noise) in a single set of 324 images.

From the results obtained in Table 9, we notice that the use of the histogram-based method is more interesting, then we chose to use in

Table 9
Results obtained by the two methods proposed for the detection of the number of clusters applied to the set gathering all the images (noisy and non-noised).

Number of clusters found	Neural network based method (DNCNN)		Histogram based method (DNCH)	
	Number of image	Rates	Number of image	Rates
Dif = 0	87	0.269	236	0.728
Dif ≤ 1	200	0.617	255	0.787
Dif > 1	124	0.382	69	0.212

Table 10
Results obtained by the histogram method.

Image	DNCH	original	
	5	5 or 4	(Pant,2019)
	5	5	(Sepas-Moghaddam, Yazdani, & Shahabi, 2014)
	5	5	(Sepas-Moghaddam et al., 2014)
	4	4	(BrainWeb, 0000, n.d.)
	4	4	
	4	5	

what follows DNCH instead of DNCNN for the estimation of the number of cluster.

Most methods use XB validation indexes to approach the number of clusters experimentally, these methods generally suffer from the following problem: they are usually exhaustive methods over a given interval. The advantage of our method is time-saving because once the network is set up (simulation of the XB behavior) it avoids repeating the experimental studies that determine the number of clusters (time-consuming).

For further validation, we apply the DNCH over six images with a known number of clusters. The results are given in Table 10.

From Table 10 we note that the number of clusters found is equal to the number expected except for images with a high rate of noise. But the difference is not significant. We opt in what follows to use the histogram-based method to determine the number of clusters.

4.2. Proposed segmentation method experimentation

In aim to validate the proposed approach, we undertook a series of experimentations using a fixed number of clusters and variable cluster number methods. In the first one, we determine the best values of the coordinating agent parameters. In the second one, we compare the proposed approach with FCM and MMTDFCM over images without noise, with salt &pepper noise and Gaussian noise. In the third one, we compare the proposed approach to four known bio-inspired methods using five images. Finally, for further validation, we use a brain web dataset to compare our approach with FCM and FPSOFCM.

4.2.1. Parameter settings

We use in the parameter settings, XB as an index to determine the best parameter values.

Determining the rate of intensification/diversification

As said in Section 3.2.1, the coordinating agent increases or decreases the intensification/diversification of an executing agent by increasing or decreasing certain parameters by a certain rate. We will define in the following experiment the best value of this rate we test different rate values (5%, 10%, 15%, 20% and 25%) to determine the best value.

The rest of the parameters are set as follows:

- Initial life time per agent = 30.
- Life time before the random reset of the parameters of a metaheuristic = 5.
- Population size = 100.
- Number of iteration = 100.

The results are presented in Table 11.

In Table 11, we notice that the Value 15% is the most redundant best value. We retain then 15% as the rate of intensification/diversification

Determining the life time per agent

We test different life time (5, 10 15, 20 and 25) to determine the best value. The rest of the parameters are set as follows:

- Rate of intensification/diversification = 15
- The Life time before the random reset of the parameters of a metaheuristic = 5.
- Population size = 100.
- Number of initial iteration = 100.

The results are presented in Table 12

From Table 12, we notice that the most redundant value for the best life time is 15, then we retain this value for the rest of experiments

Determining the life time before random reset of the metaheuristic parameters

We test different Life time before random metaheuristic parameters reset (2, 3, 4 and 5). The rest of the parameters are set as follows:

- Rate of intensification/diversification = 15
- The Life time per agent = 15.
- Population size = 100.
- Number of initial iteration = 100.

The results are given in Table 13

In Table 13, we notice that the Value 4 is the most redundant best value. We retain then 4 as Life time before random metaheuristic parameters reset parameter.

Determining the population size

Here we will attempt to determine the population size used by metaheuristics that is most appropriate to the problem. The values considered for the size of the populations of metaheuristics are 25, 50, 75 and 100. The other parameters are set as follows:

- Rate of intensification/diversification = 15
- The Life time per agent = 15.
- Life time before random metaheuristic parameters reset parameter = 4
- The number of initial iteration = 100.

Table 14 summarizes experimental results

After the tests, we retain the population size value equal to 25

Table 11
Determining the rate of intensification/diversification.

Image	5%		10%		15%		20%		25%		Best
	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	
Image 1	0.06927	49.85	0.06927	82.81	0.06927	65.57	0.06927	84.59	0.06927	79.62	5%
Image 2	0.07372	144.03	0.07372	111.12	0.07372	110.34	0.07372	142.76	0.07372	171.65	15%
Image 3	0.01784	59.07	0.01784	64.2	0.01784	51.79	0.01784	67.17	0.01784	69.15	15%
Image 4	0.06493	89.18	0.06493	88.79	0.06493	74.41	0.06493	88.04	0.06493	96.21	15%
Image 5	0.06574	251.71	0.06574	183.7	0.06574	226.14	0.06574	302.05	0.06574	327.73	10%
Image 6	0.08313	132.12	0.08313	145.04	0.08313	120.53	0.08313	156.94	0.08313	175.85	15%
Image 7	0.03375	66.77	0.03375	75.15	0.03375	65.96	0.03375	64.64	0.03375	76.19	20%
Image 8	0.09214	64.66	0.09214	81.7	0.09214	86.69	0.09214	73.7	0.09214	80.23	5%
Image 9	0.05999	249.23	0.05999	219.22	0.05999	339.41	0.05999	213.42	0.05999	307.44	20%
Image 10	0.06115	173.68	0.06115	184.74	0.06115	186.52	0.06115	189.36	0.06115	298.29	5%
Image 11	0.06191	960	0.06191	1144	0.06191	1140	0.06191	1200	0.06191	1440	5%
Image 12	0.07	240	0.07	240	0.07	300	0.07	240	0.07	420	5%
Image 13	0.06015	300	0.06015	360	0.06015	240	0.06015	480	0.06015	360	15%
Image 14	0.0704	420	0.0704	438	0.0704	300	0.0704	360	0.0704	720	15%
Image 15	0.17876	360	0.17876	360	0.17876	300	0.17876	300	0.17876	600	15%
Image 16	0.064	240	0.064	185	0.064	180	0.064	240	0.064	180	15%
Image 17	0.42	360	0.42	382	0.42	320	0.42	387	0.42	360	15%
Image 18	0.002	420	0.002	420	0.002	369	0.002	432	0.002	600	15%
Image 19	0.104	480	0.104	360	0.104	480	0.104	840	0.104	360	10%
Image 20	0.064	378	0.064	300	0.064	240	0.064	540	0.064	300	15%

Table 12
Determining the life time per agent.

Image	5		10		15		20		25		Best
	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	
Image1	0.07524	15.04	0.077	30.46	0.07225	33.76	0.06927	49.94	0.06927	62.59	20
Image2	0.07763	21.31	0.07372	66.73	0.07372	66.23	0.07372	69.41	0.07372	70.21	15
Image3	0.01916	38.71	0.01916	55.48	0.01784	59.11	0.01784	74.92	0.01784	84.08	15
Image4	0.06586	52.82	0.06539	75.37	0.06493	114.77	0.06493	131.3	0.06493	161.05	15
Image5	0.07022	43.960	0.06574	92.9	0.06574	111.66	0.06574	165.24	0.06574	183.34	10
Image6	0.08441	31	0.08569	40.7	0.08313	69.18	0.08313	79.4	0.08313	93.91	15
Image7	0.03927	14.94	0.03927	24.53	0.03375	34.3	0.03375	42.22	0.03375	55.51	15
Image8	0.09243	19.25	0.09214	44.1	0.09214	58.99	0.09214	90.98	0.09214	95.21	10
Image9	0.06044	106.88	0.05999	244.36	0.05999	286.36	0.06089	233.47	0.05999	287.5	10
Image10	0.06965	48.66	0.06115	66.5	0.06115	95.9	0.06115	111.48	0.06115	132.31	10
Image11	0.06191	60.59	0.06191	182.35	0.06191	403.02	0.06801	416.37	0.07191	427.98	5
Image12	0.0718	53.18	0.0718	55.86	0.0703	67.32	0.0703	75.35	0.0703	82.09	15
Image13	0.07115	61.2	0.07015	76.39	0.07015	76.21	0.07015	81.3	0.07015	98.56	15
Image14	0.0709	60	0.0709	101.23	0.0704	120.2	0.0705	130.57	0.0704	135.97	15
Image15	0.1752	43.58	0.1752	67.1	0.1750	68.38	0.1751	81.25	0.1751	100.25	15
Image16	0.064	41.25	0.064	54.39	0.064	58.97	0.07	60.24	0.07	71.25	5
Image17	0.525	57.21	0.513	59.84	0.510	62.35	0.510	71.021	0.510	84.56	15
Image18	0.003	112.52	0.002	114.01	0.002	121.39	0.002	128.32	0.002	132.57	10
Image19	0.109	243.1	0.107	245.23	0.104	249.78	0.104	300.01	0.104	302.74	15
Image20	0.0981	201.21	0.985	208.04	0.981	119.35	0.0967	200.2	0.0967	208.98	20

Table 13
Determining the Life time before random reset of the metaheuristic parameters.

Image	2		3		4		5		Best
	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	
Image1	0.06927	40.27	0.07524	39.99	0.06927	36.06	0.07225	39.7	4
Image2	0.07372	36.27	0.07372	38.23	0.07372	36.08	0.07372	45.18	4
Image3	0.01784	123.82	0.01784	122.57	0.01784	104.86	0.01784	139.1	4
Image4	0.06493	232.93	0.06493	230.59	0.06493	223.54	0.06493	228.98	4
Image5	0.06574	41.06	0.06574	55.24	0.06574	52.36	0.06574	62.14	2
Image6	0.08313	46.44	0.08313	53.27	0.08313	61.93	0.08313	58.4	2
Image7	0.03375	40.78	0.03927	37.22	0.03375	28.41	0.03375	36.82	4
Image8	0.09214	30.65	0.09214	27.7	0.09214	30.11	0.09214	32.19	3
Image9	0.05999	185.77	0.05999	170.28	0.06044	293.75	0.060.44	282.71	3
Image10	0.06115	75.93	0.06115	100.22	0.06115	91.26	0.0654	92.66	2
Image11	0.297	480	0.06191	600	0.06191	540	0.06191	480	5
Image12	0.07	124	0.07	120	0.07	90	0.07	90	4
Image13	0.01025	133.2	0.01025	231	0.01025	279	0.01025	189	2
Image14	0.0724	185	0.0724	130	0.0724	75	0.0724	122	4
Image15	0.17	180	0.17	110	0.17	65	0.17	74	4
Image16	0.0647	120	0.0647	95	0.0647	94	0.0647	75	5
Image17	0.42	180	0.42	127	0.42	80	0.42	120	4
Image18	0.0022	250	0.0022	230	0.0022	210	0.0022	248	4
Image19	0.104	243	0.104	260	0.104	240	0.104	240	4
Image20	0.0967	210	0.0967	166.2	0.0967	118.8	0.0967	202.8	4

Table 14
Determining the population size.

Image	25		50		75		100		Best
	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	
Image1	0.06927	3.27	0.06927	8.92	0.06927	21.86	0.06927	36.5	25
Image2	0.07372	3.1	0.07372	11.45	0.07372	17.43	0.07372	37.78	25
Image3	0.01784	3.21	0.01784	9.5	0.01784	18.53	0.01784	25.93	25
Image4	0.06493	11.23	0.06493	21.06	0.06493	35.31	0.06493	51.67	25
Image5	0.06574	4.55	0.06574	15.49	0.06574	36.31	0.06574	58.54	25
Image6	0.08313	5.19	0.08313	18.86	0.08313	29.97	0.08313	65.01	25
Image7	0.03375	14.68	0.03375	44.27	0.03375	106.9	0.03375	145.74	25
Image8	0.09214	24.9	0.09214	53.95	0.09214	88.18	0.09214	169.37	25
Image9	0.05999	185.77	0.05999	170.28	0.05999	293.75	0.05999	282.71	50
Image10	0.06115	75.93	0.06115	100.22	0.06115	91.26	0.06115	92.66	25

Table 15
Determining the number of iterations for our cooperative system.

Image	25		50		75		100		Best
	XB	Time (s)	XB	Time (s)	XB	Time (s)	XB	Time (s)	
Image1	0.06927	3.7	0.06927	6.83	0.06927	10.78	0.06927	14.39	50
Image2	0.07372	4.25	0.07372	7.93	0.07372	10.93	0.07372	14.14	50
Image3	0.01784	3.71	0.01784	7.14	0.01784	11.82	0.01784	124.68	50
Image4	0.06493	6.76	0.06493	10.83	0.06493	14.37	0.06493	20.00	50
Image5	0.06574	2.34	0.06574	4.89	0.06574	7.20	0.06574	11.10	50
Image6	0.08313	3.05	0.08313	5.08	0.08313	9.22	0.08313	13.51	50
Image7	0.03375	8.04	0.03375	14.64	0.03375	22.07	0.03375	30.72	50
Image8	0.09214	16.10	0.09214	25.9	0.09214	31.78	0.09214	42.94	50
Image9	0.05999	10.33	0.05999	18.15	0.05999	27.70	0.05999	39.23	50
Image10	0.06115	3.52	0.06115	7.20	0.06115	14.96	0.06115	14.71	50

Table 16
Coordinating agent parameters.

Parameter	Value
Rate of intensification/diversification	15
Agent time life	15
Agent time life before the random reset of the parameters	4
Population size	25
Number of iterations	50

Determining the number of iterations

The number of iterations of the metaheuristics is a very important parameter. The following tests determine the most appropriate number of iterations of the metaheuristics for the problem. Values considered for the number of iterations are 50, 100, 150 and 200. The rest of the parameters are set as follows:

- Rate of intensification/diversification = 15%.
- The Life time per agent = 15.
- Life time before random metaheuristic parameters reset parameter = 4
- Population size = 25

Table 15 summarizes experimental results.

After tests we find that the iteration number 50 is sufficient to converge to the best solution. This value will be used for the rest of this paper. Table 16 summarizes all experimentally determined parameters

In what follows, we will conduct three types of tests to assess the quality of our approach. We start with a comparative test with MMTDFCM and FCM, then a comparison that encompasses a series of works that have used bio-inspired approaches.

We are going to carry out these comparisons in order to show the effectiveness of our proposal compared to the methods that improve FCM with classical techniques (modification of the objective function, hybridization with other segmentation techniques), as well as to methods that improve FCM with metaheuristics.

4.2.2. Comparison of CSFCM with MMTDFCM and FCM

FCM Parameter: - Fuzzifier parameter m = 2

Table 17
The artificial image and images obtained by applying noise to it.

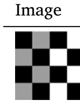
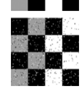
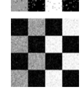
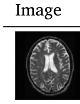
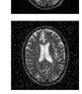
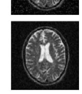
Image	Name	Number of cluster
	Artificial image	3
	Artificial image with salt & pepper noise	3
	Artificial image with Gaussian noise	3

Table 18
The MR image as well as images obtained by applying noise to it.

Image	Name	Number of cluster
	MR	4
	MR with salt & pepper noise	4
	MR with Gaussian noise	4

- Epsilon (ϵ) = $10^{(-8)}$ In this test, we compare our approach with FCM and MMTDFCM using the two proposed methods to determine the number of clusters (fixed number of clusters and a variable number of clusters). We perform these tests over 9 images represented in Tables 17–19

Tables 20, 21, 22 summarize the results of the experiments.

From Tables 20–22, we note that the proposed approach archives better results than FCM and MMTDFCM. It can be also noted that generally the version using a variable number of clusters offers better results under the VPC and VPE measure,

Table 19
The ROI image and the images obtained by applying noise to it.

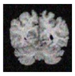
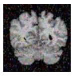
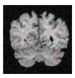
Image	Name	Number of cluster
	ROI	4
	ROI with salt & pepper noise	4
	ROI with Gaussian noise	4

Table 20
Results obtained for the tests done on the images in Table 17.

Image	Method	PSNR	PC	CE
Artificial image	FCM	23.7811	0.7329	0.4118
	MMTDFCM	23.7101	0.9582	0.0628
	CSFCMNCF	23.7101	0.9999	2.96E−04
	CSFCMVCN	113.1642	0.9999	2.96E−04
Artificial image with salt & pepper noise	FCM	15.7015	0.7526	0.3525
	MMTDFCM	22.4539	0.8967	0.1763
	CSFCMNCF	102.4505	0.9995	0.0017
	CSFCMVNC	102.4505	0.9995	0.0017
Artificial image with Gaussian noise	FCM	19.8219	0.7269	0.4392
	MMTDFCM	23.521	0.8655	0.2596
	CSFCMNCF	53.9379	0.9281	0.1452
	CSFCMVNC	40.0474	0.9313	0.127

Table 21
Results obtained for the tests done on the images in Table 18.

Image	Method	PSNR	PC	CE
MR image	FCM	11.4656	0.7902	0.4878
	MMTDFCM	20.9565	0.8424	0.2938
	CSFCMNCF	61.8352	0.9028	0.1941
	CSFCMVCN	55.6945	0.9214	0.1524
MR image with salt & pepper noise	FCM	11.2788	0.7291	0.518
	MMTDFCM	20.0788	0.7841	0.4151
	CSFCMNCF	58.9303	0.8832	0.2358
	CSFCMVNC	53.1264	0.907	0.1804
MR image with Gaussian noise	FCM	8.8638	0.5839	0.7885
	MMTDFCM	11.5931	0.4733	0.8957
	CSFCMNCF	55.8658	0.8039	0.3748
	CSFCMVNC	41.3576	0.8683	0.2212

Table 22
Results obtained for the tests done on the images in Table 19.

Image	Method	PSNR	PC	CE
ROI image	FCM	11.4767	0.8421	0.3182
	MMTDFCM	16.6042	0.7784	0.3948
	CSFCMNCF	61.1781	0.8741	0.2426
	CSFCMVCN	61.1781	0.8741	0.2426
ROI image with salt & pepper noise	FCM	10.7082	0.7678	0.4648
	MMTDFCM	15.5713	0.6677	0.6067
	CSFCMNCF	58.8062	0.8384	0.3163
	CSFCMVNC	45.1044	0.9222	0.1381
ROI image with Gaussian noise	FCM	11.8689	0.6147	0.688
	MMTDFCM	15.8405	0.6786	0.5782
	CSFCMNCF	59.0302	0.8422	0.3095
	CSFCMVNC	45.7044	0.9281	0.1294

4.2.3. *Comparison of CSFCM with basic bio-inspired methods*

In this experiment, we compare the proposed approach with four bio-inspired approaches and with FCM, using five images(see Table 23). We use XB, PC,SC and CE indices as metrics.

From Tables 24–28 it can be seen that the results obtained by the two versions of the proposed cooperation (the one using a fixed number of clusters, and using a variable number of clusters) exceed the results obtained by FCM as well as for the hybridization of FCM with other metaheuristics (BBO, ABC, AFSA, and PSO) in terms of quality.

In this paper, we have proposed a cooperative system that exploits three metaheuristics is the benefit of this advantage in order to initialize the cluster centers with the most optimal values, and that will provide

Table 23
Set of images to be used in the tests.

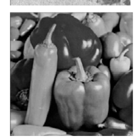
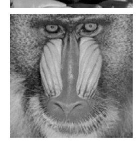
Image	Name	Number of cluster
	Lean	5
	Cameramen	4
	Air plane	5
	Pepper	4
	Baboon	4

Table 24
Cameramen image.

Method	XB	PC	SC	CE
FCM	0.1209	0.7512	0.0351	0.3897
CSFCMFCN	0.1475	**0.7996**	**0.01687**	0.3937
CSFCMVCN	**0.0312**	**0.9204**	0.0605	**0.1422**
BBO-FCM	0.0891	0.7748	0.0264	0.3626
AFSA-FCM	0.1102	0.7652	0.0284	0.3951
ABC-FCM	0.08721	0.75741	0.02014	0.353
PSO-FCM	0.08716	0.7354	0.0258	0.36251

Table 25
Baboon image.

Method	XB	PC	SC	CE
FCM	0.1056	0.7062	0.0482	0.5897
CSFCMFCN	0.0946	**0.7618**	**0.03**	**0.4726**
CSFCMVCN	**0.0909**	**0.8199**	0.1836	**0.2939**
BBO-FCM	0.0956	0.7763	0.026	0.4894
AFSA-FCM	0.1003	0.743	0.0377	0.5689
ABC-FCM	0.09592	0.76102	0.0299	0.5321
PSO-FCM	0.09598	0.7509	0.0301	0.5371

Table 26
Lena image.

Method	XB	PC	SC	CE
FCM	24.043	0.5316	0.0684	0.985
CSFCMFCN	0.0844	**0.7638**	**0.0185**	0.4723
CSFCMVCN	0.0785	**0.7642**	**0.0287**	0.4508
BBO-FCM	**0.0546**	0.2368	0.2735	0.301
AFSA-FCM	0.3532	0.3345	0.2694	0.4546
ABC-FCM	0.8996	0.0608	0.567	0.3544
PSO-FCM	0.2829	0.3925	0.941	0.49

Table 27
Air plane image.

Method	XB	PC	SC	CE
FCM	0.16107	0.72371	0.0521	0.41483
CSFCMFCN	0.1604	0.7851	0.0274	0.3933
CSFCMVCN	**0.0308**	**0.9303**	0.0866	**0.1255**
BBO-FCM	0.081	0.748001	**0.0337**	0.40021
AFSA-FCM	0.10125	0.73569	0.0421	0.3501
ABC-FCM	0.0841	0.747087	0.0385	0.3152
PSO-FCM	0.0832	0.746689	0.0388	0.31901

Table 28
Pepper image.

Method	XB	PC	SC	CE
FCM	0.1101	0.7423	0.0423	0.5103
CSFCMFCN	0.0647	**0.77712**	0.0154	0.4201
CSFCMVCN	**0.0547**	**0.8885**	**0.109**	**0.1955**
BBO-FCM	0.08201	0.77509	0.02014	0.4315
AFSA-FCM	0.08321	0.7521	0.03851	0.501
ABC-FCM	0.08251	0.7321	0.0258	0.49221
PSO-FCM	0.08263	0.7351	0.0298	0.49234

Table 29
Results obtained by our proposal as well as FPSOFCM and FCM applied to the brain web dataset.

Volume	Method				
	Metric	FCM	FPSOFCM	CSFCMFCN	CSFCMVCN
	PC	0,886496	0,890695	0,89388956	0,92439962
0%	CE	0,221893	0,213521	0,21293391	0,13923622
	XB	0,046393	0,044498	0,03931838	0,0320088
	PC	0,861299	0,866553	0,89287604	0,93176204
20%	CE	0,269972	0,259591	0,21416994	0,12077025
	XB	0,06042	0,057582	0,04071238	0,02640442
	PC	0,874644	0,878394	0,88722126	0,93110184
40%	CE	0,24508	0,237761	0,2208721	0,12168895
	XB	0,046779	0,045211	0,05149612	0,02667555

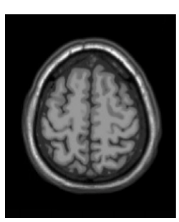


Fig. 6. Original image.

a good segmentation. The results (clusters) returned will be used in the following image processing steps after the segmentation.

For further validation, we use brain web dataset ("BrainWeb", n.d.) to compare the proposed approach (CSFCM) with FCM, and PSOFCM using both variants of cluster detection (fixed and variable) the results are shown in Table 29. For fixed method we have c = 4

According to Table 29, it can be seen that both versions of the proposed cooperation (fixed at 4 clusters, and using a variable number of clusters) offer better results in comparison with the standard version of FCM as well as the FPSOFCM version. We note also that the version using a variable number of clusters gives better results than the version using a fixed number of clusters. Finally, we present in Figs. 6–8 a visual comparison of the fixed cluster and variable cluster methods using Brain web dataset image with 0% of noise:

5. Conclusion

In this paper, we propose to solve two problems related to the use of FCM for image segmentation. First, we propose two approaches to solve the problem of cluster number detection: DNCH which uses

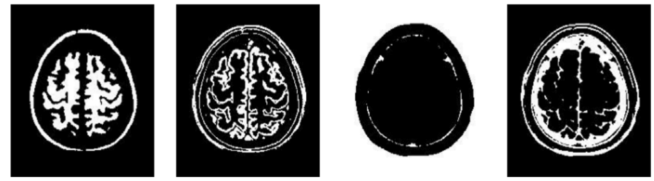


Fig. 7. Result of the segmentation obtained by our proposal using clusters fixed to 4.

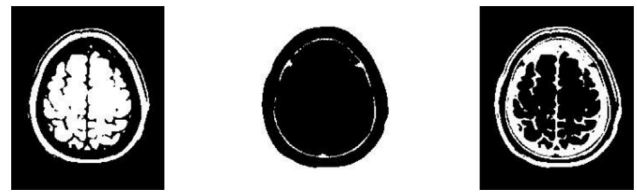


Fig. 8. Result of the segmentation obtained by our proposal using a variable number of clusters.

histogram information, and DNCN which exploits neural networks. An experimental study is conducted to evaluate the two proposed methods. The experiments are applied on a synthetic dataset composed of three parts (sharp images, images with Gaussian noise, and images with salt & pepper noise), and tested on 3 rules (Dif = 0, Dif ≤ 1 and Dif > 1). The returned results show that DNCH offers better results than DNCN. For further validation, we tested the DNCH method on six images with a known number of clusters. From the results, we concluded that the number of clusters found is equal to the expected number, except for the last image which was noisy with a high noise level (but the difference is not significant). The second problem is the initialization of cluster centers. For this problem, we propose a new cooperative system CSFCM that uses three metaheuristics (BBO, GA, and FA). Before launching the experimental study, we perform some tests to set the parameters of our system (intensification/diversification, lifetime per agent, a lifetime before the random reset of the parameters of a metaheuristic, number of iterations and population size). After selecting the parameters, we compare our approach with other works. Firstly, we compare our approach (using a fixed number of clusters, and a variable number of clusters) with MMTDFCM and FCM on 9 images (3 sharp images, 3 images with Gaussian noise, and 3 images with salt & pepper noise) and evaluate the quality with three metrics (CE, PC, and PSNR) the results show that our approach offers the best results in all cases. Another comparative study is conducted where we compare our approach with four bio-inspired methods hybridized with FCM (BBO-FCM, AFSA-FCM, ABC-FCM, PSO-FCM) and FCM. The experiment is performed on 5 images with a known number of clusters and evaluated with four metrics (XB, PC, CE, SC). The experimental study confirms the efficiency of our approach (using a fixed number of clusters, and a variable number of clusters) where we surpass FCM and the four bio-inspired methods. For further validation, we use a dataset from the Brain web to compare our approach with FCM and FPSOFCM. The proposal with the variable number of clusters gives the best results for all cases. The proposal with the fixed number of clusters exceeds FCM and FPSOFCM in most cases.

CRediT authorship contribution statement

Hamza Abdellahoum: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Drafting the manuscript, Revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. **Nassim Mokhtari:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Approval of the version of the manuscript to be published. **Abderrahmane Brahimi:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Approval of the version of the manuscript to be published. **Abdelmadjid Boukra:** Conception and design of study, Analysis and/or interpretation of data, Drafting the manuscript, Revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahmed, M. N., Yamany, S. M., Mohamed, N., Farag, A. A., & Moriarty, T. (2002). A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Transition of Medical Imaging*, *21*(3), 193–199. http://dx.doi.org/10.1109/42.996338.

Aiguo, C., & Yunjie, z. (2019). Image segmentation based on a robust fuzzy C means algorithm. *Journal of Medical Imaging and Health Informatics*, *9*(7), 1464–1468. http://dx.doi.org/10.1166/jmihi.2019.2745.

Balasubramani, k., & Marcus, K. (2013). Artificial bee colony algorithm to improve brain MR image segmentation. *International Journal on Computer Science and Engineering*, *5*(1), 31–36.

Beale, E. M. L. (1969). *Cluster analysis*. London, England: Scientific Control Systems.

Bezdek, J. C. (1973). Cluster validity with fuzzy sets. *Journal of Cybernetics*, *3*(3), 58–73. http://dx.doi.org/10.1080/01969727308546047.

Bezdek, J. (1981). *Pattern recognition with fuzzy objective function algorithms*. NORWELL, MA, USA: Plenum, http://dx.doi.org/10.1007/978-1-4757-0450-1.

(n.d.). BrainWeb: Simulated Brain Database. Retrieved from http://www.bic.mni.mcgill.ca/brainweb/.

Calinski, R. B., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*, 1–27.

Chen, w., & Fang, k. (2008). A hybridized clustering approach using particle swarm optimization for image segmentation. In *IEEE international conference on audio, language and image processing* (pp. 1365–1368). http://dx.doi.org/10.1109/icalip.2008.4590208.

Chen, S., & Zhang, D. (2004). Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *34*, 1907–1916.

Chu, X., Zhu, Y., Shi, J., & Song, J. (2010). Method of image segmentation based on fuzzy c-means clustering algorithm and artificial fish swarm algorithm. In *IEEE international conference on intelligent computing and integrated systems* (pp. 254–257). http://dx.doi.org/10.1109/iciss.2010.5657199.

Chuang, K. S., Tzeng, H. L., Chen, S., Wu, J., & Chen, T. J. (2006). Fuzzy C-means clustering with spatial information for image segmentation. *Journal of Computerized Medical Imaging and Graphics*, *30*, 9–15.

Daoudi, M., Boukra, A., & Ahmed-Nacer, M. (2011). A biogeography inspired approach for security audit trail analysis. *Journal of Intelligent Computing*, *2*(4), 157–168.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York, USA: Wiley.

Gordon, A. D. (1999). *Classification* (2nd ed.). London: Chapman and Hall-CRC.

Griffin, G., Holub, A. D., & Perona, P. (2006). Caltech. Retrieved from http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

Gustafson, D. E., & Kessel, W. C. (1978). Fuzzy clustering with a fuzzy covariance matrix. In *Decision and control including the 17th symposium on adaptive processes, 1978 IEEE conference, Vol. 17* (pp. 761–766).

Haghdoost, M., Abadi, A. H., & Abedini, S. J. M. (2015). A new method of fuzzy clustering by using the combination of the firefly algorithm and the particle swarm optimization algorithm. *WALIA Journal*, *31*(S3), 245–252.

Hall, L. O., & Kanade, P. M. (2005). Swarm based fuzzy clustering with partition validity. In *The 14th IEEE international conference on fuzzy systems* (pp. 991–995). http://dx.doi.org/10.1109/fuzzy.2005.1452529.

He, S., Belacel, N., Hamam, H., & Bouslimani, Y. (2009). Fuzzy clustering with improved artificial fish swarm algorithm. In *International joint conference on computational sciences and optimization* (pp. 317–321). http://dx.doi.org/10.1109/cso.2009.367.

Holland, j. (1975). *Adaptation in natural and artificial systems*. CAMBRIDGE, MA, USA: MIT Press.

Hong, L., Xiao, X.-A., & Zhu, W.-J. (2006). Measure of medium truth scale and its application (I). *Chinese Journal of Computers*, *29*(12), 2186–2193.

Hong, L., Xiao, X. A., & Zhu, W. J. (2007). Measure of medium truth scale and its application (II). *Chinese Journal of Computers*, *30*(9), 1551–1558.

Hore, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. In *2010 20th international conference on pattern recognition*. http://dx.doi.org/10.1109/icpr.2010.579.

Jiang, M., Mastorakis, N. E., Yuan, D., & Lagunas, M. A. (2009). Image segmentation with improved artificial fish swarm algorithm. In *Proceedings of the european computing conference, Vol. 28* (pp. 133–138). http://dx.doi.org/10.1007/978-0-387-85437-3_12.

Jing, L., Deng, D., & Yu, J. (2014). Weighting exponent selection of fuzzy C-means via Jacobian matrix. In *KSEM: International conference on knowledge science, engineering and management, Vol. 8793* (pp. 115–126). http://dx.doi.org/10.1007/978-3-319-12096-6_11.

Krinidis, S., & Chatzis, V. (2010). A robust fuzzy local information c-means clustering algorithm. *IEEE Transactions on Image Processing*, *19*, 1328–1337.

Kumar, R., Dwivedi, R., & Jangam, E. (2019). Hybrid fuzzy C-means using bat optimization and maxi-min distance classifier. In *International conference on advances in computing and data sciences, Vol. 1046* (pp. 68–79). http://dx.doi.org/10.1007/978-981-13-9942-8_7.

Kumar, S. N., Fred, A. L., Kumar, H. A., & Varghese, P. S. (2018). Nonlinear tensor diffusion filter based marker-controlled watershed segmentation for CT/MR images. In *Proceedings of international conference on computational intelligence and data engineering, Vol. 9* (pp. 317–331). http://dx.doi.org/10.1007/978-981-10-6319-0_27.

Kumar, S. N., Fred, A. L., Kumar, H. A., & Varghese, P. S. (2019). Firefly optimization based improved fuzzy clustering for CT/MR image segmentation. In J. Hemanth, & V. Balas (Eds.), *Nature inspired optimization techniques for image processing applications, Vol. 150* (pp. 1–28). Cham: Intelligent Systems Reference Library, Springer, http://dx.doi.org/10.1007/978-3-319-96002-9_1.

Kumar, J. T., Reddy, Y. M., & Rao, B. P. (2018). Change detection in sarimages based on artificial bee colony optimization with fuzzy C - means clustering. *International Journal of Recent Technology and Engineering (IJRTE)*, *7*(4), 156–160.

Liu, H.-C., Yih, J.-M., Lin, W.-C., & Liu, T.-S. (2009). Fuzzy C-means algorithm based on PSO and mahalanobis distance. *International Journal of Innovative Computing*, *5*(12), 5033–5040.

MacArthur, R., & Wilson, E. (1967). *The theory of biogeography*. Princeton, NJ: Princeton Univ. Press.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley symposium on mathematics. statistics and probability, Vol. 1* (pp. 281–297). University of California Press.

Miao, J., Zhou, X., & Huang, T.-Z. (2020). Local segmentation of images using an improved fuzzy C-means clustering algorithm based on self-adaptive dictionary learning. *Applied Soft Computing*, *91*, http://dx.doi.org/10.1016/j.asoc.2020.106200.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of groupe in a data set using sum of squares clustering. *Biometrics*, *44*, 23–34.

Nayak, J., Vakula, K., Dash, P. B., & Naik, B. (2019). Kidney-inspired algorithm and fuzzy clustering for biomedical data analysis. In *Big data analytics for intelligent healthcare management* (pp. 269–281). http://dx.doi.org/10.1016/b978-0-12-818146-1.00011-8.

Ouadfel, S., & Meshoul, S. (2012). Handling fuzzy image clustering with a modified ABC algorithm. *International Journal of Intelligent Systems and Applications*, *4*(12), 65–74. http://dx.doi.org/10.5815/ijisa.2012.12.09.

Ozkan, I., & Turksen, I. (2007). Upper and lower values for the level of fuzziness in fcm. *Information Sciences*, *177*, 5143–5152.

Padmavathi, P., Eswaramurthy, V., & Revathi, J. (2018). Fuzzy social spider optimization algorithm for fuzzy clustering analysis. In *2018 international conference on current trends towards converging technologies (ICCTCT)*. http://dx.doi.org/10.1109/icctct.2018.8550985.

Pal, N., & Bezdek, J. (1995). On cluster validity for the fuzzy c-mean model. *IEEE Transactions on Fuzzy Systems*, *3*, 370–379.

Pant, A., Chinta, S. S., & Tripathy, B. K. (2019). Comparative analysis of hybridized c-means and fuzzy firefly algorithms with application to image segmentation. In *Proceedings of the 2nd international conference on data engineering and communication technology, Vol. 828* (pp. 315–323). http://dx.doi.org/10.1007/978-981-13-1610-4_32.

Pham, D. (2002). Fuzzy clustering with spatial constraints. In *Proceedings of the international conference on image processing, Vol. 2*, New York, USA (pp. II-65–II-68). 2002.

Pham, D. L., & Prince, J. L. (1999). An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. *Pattern Recognition Letters*, *20*, 57–68.

Qiao, W., & Yang, Z. (2020). An improved dolphin swarm algorithm based on kernel fuzzy C-means in the application of solving the optimal problems of large-scale function. *IEEE Access*, *8*, 2073–2089. http://dx.doi.org/10.1109/access.2019.2958456.

Ren, T., Wang, H., Feng, H., Xu, C., Liu, G., & Ding, P. (2019). Study on the improved fuzzy clustering algorithm and its application in brain image segmentation. *Applied Soft Computing*, *81*, http://dx.doi.org/10.1016/j.asoc.2019.105503.

Samina, N., Hammad, M., & Humayun, I. (2010). Image segmentation using fuzzy clustering : A survey. In *Proceedings - 2010 6th international conference on emerging technologies, ICET 2010* (pp. 181–186). http://dx.doi.org/10.1109/ICET.2010.5638492.

Sepas-Moghaddam, A., Yazdani, D., & Shahabi, J. (2014). A novel hybrid image segmentation method. *Progress in Artificial Intelligence*, *3*(1), 39–49. http://dx.doi.org/10.1007/s13748-014-0044-7.

Sevaux, M. (2004). *Métaheuristiques, stratégies pour l'optimisation de la production de biens et de services, sous la direction du professeur Christian Tahon*. Paris: CNRS, https://tel.archives-ouvertes.fr/tel-00011623/document.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713. http://dx.doi.org/10.1109/tevc.2008.919004.

Singh, T. I., Laishram, R., & Roy, S. (2019). Comparative study of combination of swarm intelligence and fuzzy c means clustering for medical image segmentation. In A. Luhach, K. Hawari, I. Mihai, P. A. Hsiung, & Mishra R. (Eds.), *Smart computational strategies: Theoretical and practical aspects* (pp. 69–80). Singapore: Springer.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set using the gap statistic. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *63*, 411–423.

Wang, J., Fang, H., Li, B., & Wang, H. (2017). Evaluate clustering performance and computational efficiency for PSO based fuzzy clustering methods in processing big imbalanced. In *2017 IEEE international conference on communications (ICC)*. http://dx.doi.org/10.1109/ICC.2017.7996802.

Wang, Q., Zhang, Y., Xiao, Y., & Li, J. (2017). Kernel-based fuzzy c-means clustering based on fruit fly optimization algorithm. In *International conference on grey systems and intelligent services (GSIS)* (pp. 251–256). http://dx.doi.org/10.1109/gsis.2017.8077713.

Xia, P., Lin, Y., & Li-Hua, Z. (2019). An improved PSO-fcm algorithm for image segmentation. In *IOP conf. ser.: Earth environ. sci.* (pp. 1–7). http://dx.doi.org/10.1088/1755-1315/267/4/042081.

Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(8), 841–847. http://dx.doi.org/10.1109/34.85677.

Xiong, L., Tang, G., Chen, Y., Hu, Y., & Chen, R. (2020). Color disease spot image segmentation algorithm based on chaotic particle swarm optimization and FCM. *The Journal of Supercomputing*, http://dx.doi.org/10.1007/s11227-020-03171-8.

Xu, H., Houb, R., Liuc, L., Yong Caic, j., Chend, j., & Liue, j. (2020). The image segmentation algorithm of colorimetric sensor array based on fuzzy c-means clustering. *Journal of Intelligent & Fuzzy Systems*, *38*(4), 3605–3613. http://dx.doi.org/10.3233/JIFS-179583.

Yang, X. S. (2008). *Nature-inspired metaheuristic algorithms*. CAMBRIDGE, UK: Luniver Press.

Yang, Y., Zheng, Ch., & Lin, P. (2001). Fuzzy C-means clustering algorithm with a novel penalty term for image aegmentation. *Opto-Electronics Review*, *13*(4), 309–315.

Yu, Y., Cheng, Q., & Huang, H. (2005). Analysis of the weighting exponent in the fcm. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *34*, 634–639.

Zadeh, A. L. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353. http://dx.doi.org/10.1016/s0019-9958(65)90241-x.

Zhang, M., Jiang, W., Zhou, X., Xue, Y., & Chen, S. (2017). A hybrid biogeography-based optimization and fuzzy c-means algorithm for image segmentation. *Soft Computing*, *23*(6), 2033–2046. http://dx.doi.org/10.1007/s00500-017-2916-9.

Zhou, N., Yang, T., & Zhang, S. (2014). An improved FCM medical image segmentation algorithm based on MMTD. *Computational and Mathematical Methods in Medicine*, 1–8. http://dx.doi.org/10.1155/2014/690349.