Applied Soft Computing 113 (2021) 108005

Contents lists available at ScienceDirect

Applied Soft Computing

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

CGFFCM: Cluster-weight and Group-local Feature-weight learning in Fuzzy C-Means clustering algorithm for color image segmentation

Amin Golzari Oskouei [a,b], Mahdi Hashemzadeh [a,b,*], Bahareh Asheghi [a,b], Mohammad Ali Balafar [c]

[a] Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran
[b] Artificial Intelligence and Machine Learning Research Laboratory, Azarbaijan Shahid Madani University, Tabriz, Iran
[c] Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Iran

ARTICLE INFO

Article history:
Received 24 February 2021
Received in revised form 7 October 2021
Accepted 14 October 2021
Available online 26 October 2021

Keywords:
Fuzzy C-means
Color image segmentation
Feature weighting
Cluster weighting
Group-local feature weighting
Clustering

ABSTRACT

The fuzzy c-means (FCM) algorithm is a popular method for data clustering and image segmentation. However, the main problem of this algorithm is that it is very sensitive to the initialization of primary clusters, so it may not perform well in segmenting complex images. Another problem with the FCM is the equal importance of the image features used during the segmentation process, which causes unstable performance on different images. In this paper, we propose an FCM-based color image segmentation approach, termed CGFFCM, applying an automatic cluster weighting scheme to reduce the sensitivity to the initialization, and a group-local feature weighting strategy to better image segmentation. Also, we combine the proposed clustering algorithm with the Imperialist Competitive Algorithm (ICA) to optimize the feature weighting process. In addition, we apply an efficient combination of image features to increase the segmentation quality. The performance of CGFFCM is evaluated and compared with state-of-the-art methods (such as SMKIFC (semi-supervised surrogate-assisted multi-objective kernel intuitionistic fuzzy clustering), and A-PSO-IT2IFCM (alternate particle swarm optimization based adaptive interval type-2 intuitionistic FCM clustering algorithm)) using the Berkeley benchmark dataset. The results obtained by CGFFCM are 95%, 79%, and 91%, in terms of average *Accuracy*, *NMI*, and *F-score* metrics, respectively, which all are better than the competitors. The implementation source code of CGFFCM is made publicly available at https://github.com/Amin-Golzari-Oskouei/CGFFCM.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.8364771.v1.

1. Introduction

Image segmentation is the process of segmenting a digital image into several parts (a set of pixels or super-pixels). The aim is to simplify or change the representation of an image into something more meaningful and easier to analyze. In computer vision, the clustering-based image segmentation method is used in various applications such as object detection [1], people counting [2–4], face and facial expression recognition [5], fingerprint recognition [6], fire detection [7], medical image processing [8] and other industrial and monitoring applications [9–12]. In recent decades, various image segmentation methods have been proposed that can be classified into three categories: threshold-based methods [13,14], cluster-based methods [15–21], and region-based methods [19]. The different approaches proposed in each of these methods have their advantages and limitations. Among these methods, the cluster-based one is trendy and useful because of retaining more image information, easy and fast implementation [19], and the good results they provide [18].

The fuzzy c-means (FCM) clustering algorithm [22] is one of the most popular clustering algorithms that is applied for image segmentation. In this method, a pixel can be assigned to several clusters with different degrees of membership. Fuzzy clustering methods (such as FCM) compared to hard clustering methods (such as k-means [23]) can retain more image information and

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Correspondence to: Azarbaijan Shahid Madani University, Tabriz-Azarshahr Road, Tabriz, 5375171379, Iran.
E-mail addresses: a.golzari@tabrizu.ac.ir (A. Golzari Oskouei), hashemzadeh@azaruniv.ac.ir (M. Hashemzadeh), bahareh.asheghi@tabrizu.ac.ir (B. Asheghi), balafarila@tabrizu.ac.ir (M.A. Balafar).

https://doi.org/10.1016/j.asoc.2021.108005
1568-4946/

achieve better segmentation results [24]. Also, the fuzzy membership function in this algorithm helps us to discover the complex relationships between a sample (here the pixels) and all clusters more accurately [25].

However, the main problem of the FCM algorithm is its sensitivity to the initial cluster centers, which can drop its performance in segmenting complex images with inhomogeneous regions [26]. Another problem with this algorithm is that it considers the same importance for all features [27,28], while in most of the clustering applications, especially in the color image segmentation task, usually various groups of features (such as a group of color features, a group of texture features, a group of edge features, etc.), including sub-features, are utilized [17]. As the number of features increases, some features may be less important in some images while more important in others. Hence, considering the same weight for all of the features in a group makes the result of image segmentation unsatisfactory [16,17]. To better understand this fact, see Fig. 1. This figure shows each of the RGB color channels of a sample image. As shown in this figure, the G channel is likely to be a better feature for segmenting this image. In fact, in this channel, each part of the image is well separated and does not overlap, while, in channel R, and even worse in channel B, the parts are not clearly separable and may have many overlaps. This can be true for other groups of features, so considering the same weight for all types of features and sub-features is likely to reduce the efficiency of the clustering.

Regarding the FCM's sensitivity to the initialization problem, various methods have been proposed so far. Most of them can be used for all clustering-based applications, and some methods have been proposed explicitly for the image segmentation task. Some methods, such as the fuzzy c-mean++ [29], try to eliminate dependence on random initial conditions by spreading the initial cluster representatives in the data space at the initialization step. Some methods apply a scheme to prevent the formation of low-quality clusters during the algorithm's iteration. For example, the methods presented in [28,30] start from random centers, and the desired centers are calculated automatically during algorithm iterations.

The method in [31] presents an unsupervised fuzzy-model-based image segmentation method. This method combines generalized Gaussian density and color into the fuzzy clustering algorithm and incorporates their neighboring information into a learning process to improve segmentation results. Also, this method uses membership entropy to reduce the sensitivity to primary clusters. Some other methods try to eliminate this sensitivity by combining the meta-heuristic methods with the FCM or k-means algorithm. The CSFCM method [32] improves the FCM-based image segmentation by combining it with three meta-heuristics of genetic, biogeography, and firefly. In [33], to find the suitable cluster centers and the fuzzifiers, a method called A-PSO-IT2IFCM has been introduced for image segmentation using the particle swarm optimization (PSO) algorithm. In [34], a combined algorithm based on modified k-means and ICA (Imperialist Competitive Algorithm) was developed, in which cluster centers were selected and computed appropriately. In [35], authors proposed ICAFCM (combined algorithm based on FCM and ICA) and PSOFCM (combined algorithm based on FCM and PSO) algorithms aiming to help the FCM to escape from local optima and increase the convergence speed of the ICA and PSO algorithms in the clustering process. Other similar methods were proposed in [36–38]. Most of these methods aim to find the appropriate number of clusters or primary centers. These methods often use standard clustering methods (such as FCM or k-means). Although by selecting the appropriate initial centers, the initialization sensitivity is reduced, the problem of giving equal importance to the features remains. In fact, finding appropriate initial centers is necessary but not enough. Without the proper feature weighting mechanism, the results do not improve significantly.

In recent years, Multi-Objective Evolutionary Algorithms (MOEAs) have gained more attention. Zhao et al. [20,33,39] applied it to intuitionistic fuzzy clustering optimization for the color image segmentation tasks. The purpose of this algorithm was to find the optimal cluster centers. The main problem with MOEA was its high computational cost. To solve this problem, a method, called KRVEA (Kriging-assisted reference vector guided evolutionary algorithm), was proposed in [40]. Zhao et al. [20] also used KRVEA to find the cluster centers in the intuitionistic fuzzy clustering algorithm for the color image segmentation task.

To solve the second problem of FCM (i.e., the problem of giving equal importance to various features), different feature weighting techniques [41] have been proposed, as well. Feature weighting techniques can be divided into two groups. The first group includes the algorithms that assign weights to the features globally. That is, a given feature is assigned with only one weight along with all the clusters. The methods in the second group assign local weights to the features, and thus, the same feature has different weights in different clusters. This scheme has shown a better performance than the global weighting scheme [28].

In 2014, Xing et al. [16] presented an image segmentation method based on feature-weighted FCM algorithm. They introduced a global feature weighting algorithm, namely IFWFCM. IFWFCM automatically computed feature weights in the segmenting process. In 2016, a method called "maximum-entropy-regularized weighted FCM (EWFCM)" was introduced [42]. In this method, a new objective function, based on the local feature-weight entropy regularization technique, was proposed to find the optimal weights for the features. A robust local feature weighting hard c-means (RLFWHCM) approach was proposed in [43]. In this method, the weights of features were automatically calculated per cluster during the iteration process. In 2018, a new method called weighted k-harmonic means (WKHM) clustering algorithm using feature weighting for color image segmentation was introduced [17], which applied a global feature weighting technique.

In [44], a new objective function based on a kernel metric was proposed. They used local feature weighting scheme to find those clusters that have linearly non-separable patterns or non-hyperspherical shapes. In their work, a multi-objective optimization method was proposed to be used in a feature-weighted clustering process. Two separate objective functions, taking into account the inter-cluster separation and intra-cluster compactness, were optimized simultaneously. Other feature weighting methods are introduced in [43,45].

Recently, in [28], we presented a new FCM-based clustering algorithm based on the local feature weighting and cluster weighting schemes to overcome the two problems mentioned above. In this method, cluster weighting was used to reduce the FCM's sensitivity to the selection of initial centers, and feature weighting was used to increase the accuracy of the clustering. The experiments confirmed that the local feature weighting technique (weighting the features in each cluster separately) is more useful than the global weighting methods (weighting the features by the same weights in all clusters). However, in this method, as in other methods, the difference in the importance of the sub-features of a group of features, which can be very effective in the correct segmentation of images, has not been considered.

In this paper, we propose an FCM-based color image segmentation approach, according to the following motivation, method and results, contribution, and research questions.

Motivation: As stated above, some improved versions of the FCM algorithm have been proposed in the literature, each of which has somehow mitigated the first problem (equal importance of features) or the second one (sensitivity to initialization).

Fig. 1. Three color features extracted from an RGB image. (a) Input image, (b) channel R, (c) channel G, and (d) channel B.

However, to the best of our knowledge, the difference in the importance of the sub-features of a group of features, which can be very effective in the correct segmentation of images, has not been considered yet. Also, for the image segmentation task, simultaneous treatment of the two mentioned problems of FCM has not been considered. Motivated by this knowledge and given the fact that the two mentioned problems can coexist in almost all real-world clustering-based image segmentation tasks, we propose an improved FCM-based segmentation method, termed CGFFCM (Cluster-weight and Group-local Feature-weight learning in Fuzzy C-Means clustering algorithm), to overcome the two problems simultaneously.

Method and Results: In order to cope with the two problems of FCM-based image segmentation (i.e., the equal importance to features and initialization sensitivity problems), an automatic cluster weighting scheme is designed to reduce the initialization's sensitivity, and a group-local feature weighting strategy is applied to better segmentation of the images. In the proposed group-local feature weighting technique, the local weights and group weights of features in each cluster are measured independently. To find the values of the optimal weights of each group, we combine the clustering procedure with the ICA meta-heuristic optimization method [46]; therefore, all weights, including the weights of the groups and the weights of the features within each group, are calculated automatically and optimally during the clustering process. Moreover, we utilize an efficient combination of image features, consisting of eight features from three groups of features (i.e., local homogeneity, color space, and texture), to have better segmentation results. Local homogeneity is a feature based on the local image information and refers to the image regions' uniformity. We use each of the HSV color channels to calculate local homogeneity for each pixel. Thus three features are extracted. To extract the color feature, we use the CIELAB color space. The color component of a pixel in each CIELAB channel is considered as the color feature of that pixel. Therefore, the three features L, A, and B are considered as the color features of the image. Also, to calculate the texture components, we use the Gabor filter and Gray-Level Co-occurrence Matrix (GLCM).

Most of the existing clustering methods, such as [35–38,47], which are a combination of clustering algorithms and ICA methods, have two general goals: (1) finding suitable cluster centers, and (2) find the appropriate number of clusters. Unlike the existing methods, the proposed method uses the ICA metaheuristic algorithm to find the weight coefficient of groups. Moreover, the optimal values of the cluster centers are obtained automatically during the algorithm repetition (with cluster weighting technique). In addition, the main difference between the proposed method and the existing methods is that most of the existing methods use standard FCM algorithms and metaheuristic algorithms (such as ICA, PSO, and GA). Basically no improvement is suggested for FCM, while in the proposed method, a new weighted and non-Euclidean distance function, namely a group-local feature-weight technique, and a cluster weighting technique have been added to the FCM algorithm.

The performance of CGFFCM is evaluated on the standard Berkeley segmentation dataset [48] and compared with the results of other successful clustering algorithms. The obtained results show the high efficiency of the CGFFCM against the competitors. Extensive experiments are performed to evaluate the effectiveness of each solution applied in CGFFCM.

Contribution: In summary, the contributions of this work is fourfold:

(1) To reduce the FCM's sensitivity to initial centers, a cluster weighting technique is applied. The weight of the clusters is calculated dynamically while taking the importance of the features and sub-features in each cluster into account;

(2) In the image segmentation process, the importance of the extracted features is different for each image. A group of features may be important in general, but the importance of a sub-feature in a group may be less or more than other sub-features. Hence, this paper introduces a group-local feature weighting technique. In this technique, unlike other feature weighting techniques such as local weighting (feature has different weights in different clusters) or global weighting (feature has same weights in all clusters), the groups and the sub-features in each group have different weights depending on their importance in the clusters. Thus weighting is done more effectively and efficiently, resulting in better segmentation;

(3) Group-local feature weighting and cluster weighting are performed simultaneously and automatically during the clustering process resulting in high-quality segments regardless of the initial centers;

(4) An efficient combination of image features, consisting of eight sub-features from three groups of features (i.e., local homogeneity, color space, and texture), is used to have better segmentation results. The local homogeneity is based on the local image information and refers to the image regions' uniformity. The CIELAB color space provides a perceptually uniform space, meaning that the Euclidean distance between two color points in the CIELAB color space corresponds to the perceptual difference between the two colors by the human visual system [49]. The texture components are also used together with color information and local homogeneity to achieve better segmentation results.

Research Questions: we aim to answer the following research questions in this study:

(1) Does weighting the clusters automatically reduce the sensitivity of the FCM clustering-based image segmentation method to the initialization?

(2) How much does weighting features and sub-features (in a group-local feature-weighting manner) improve the performance of the FCM clustering-based image segmentation method?

The rest of the paper is organized as follows. In Section 2, a brief overview of our previous work presented in [28], which is the basis of this work, and the ICA meta-heuristic algorithm [46] are given. In Section 3, the proposed color image segmentation approach is described in detail. In Section 4, the experimental results are presented. In Section 5, the conclusions and possible future works are discussed.

2. Preliminaries

In this section, the clustering method presented in [28], which is the basis of the proposed method (CGFFCM), and also the ICA metaheuristic algorithm, which is used as an optimizer in the feature weighting process in the proposed approach, are introduced.

2.1. FCM clustering method based on feature-weight and cluster-weight learning

The method in [28] presents a clustering algorithm that reduces the FCM's sensitivity to the initialization by using a cluster weighting scheme. It also uses a local feature weighting mechanism to increase the clustering accuracy. Also, it applies a new objective function based on the non-Euclidean distance criterion. This distance criterion is not sensitive to noise and outliers. The method tries to minimize the objective function in Eq. (1):

$$F(\boldsymbol{U},\boldsymbol{C},\boldsymbol{W},\boldsymbol{Z}) = \sum_{n=1}^{N}\sum_{k=1}^{K} {u_{nk}}^{\alpha} z_k^p d_{nk}, \tag{1}$$

where, $\mathbf{U} = [u_{nk}]$ is a membership matrix, u_{nk} represents the membership of nth data point to the kth cluster, $\mathbf{C} = [c_{km}]$ is a matrix of cluster centers, c_{km} represent the mth feature in the kth cluster, $\mathbf{W} = [w_{km}]$ is the feature weight matrix, w_{km} represents the weight of mth feature in the kth cluster, $\boldsymbol{Z} = [z_k]$ represents cluster weight vector having the length of K, z_k represents the weight of kth cluster, N is the number of data points, K refers to the number of clusters, and α is the fuzzification coefficient ($\alpha > 1$). The parameter p is within the range $0 \le p < 1$. This parameter controls the sensitivity of the weight updates to the relative differences of the cluster variances. It helps the algorithm to form balanced clusters in terms of cluster variance. See [28] for more details on this parameter. d_{nk} is a weighted and non-Euclidean distance measure and is defined as Eq. (2). d_{nk} shows the distance between the nth sample (x_{nm}) from the center of the kth cluster (c_{km}). In calculating this distance, the weighting of features in each cluster (w_{km}) is used. Therefore, to calculate the distance of the nth sample from the kth cluster, the weight of the features related to the kth cluster is considered.

$$d_{nk} = \sum_{m=1}^{M} w_{km}^{q}\left(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)\right), \tag{2}$$

where, γ_m shows the inverse variance of the mth feature of $\mathbf{X}$ dataset, $\mathbf{X} = [x_{nm}]$ is a dataset matrix, x_{nm} represents the mth feature in nth data point, and M refers to the number of features. The parameter q is in the range $q < 0$ and $q > 1$. This parameter is used for satisfying the "*feature weighting principle*" [28]. See [28] for more details on how q works. The updating equations for u_{nk}, c_{km}, w_{km}, and z_k are obtained through Eqs. (3) to (6), respectively.

$$u_{nk} = \frac{1}{\sum_{l=1}^{K}\left[\frac{z_k^p d_{nk}}{z_l^p d_{nk}}\right]^{\frac{1}{(\alpha-1)}}}, \tag{3}$$

$$c_{km} = \frac{\sum_{n=1}^{N} {u_{nk}}^{\alpha} \exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right) x_{nm}}{\sum_{n=1}^{N} {u_{nk}}^{\alpha} \exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)}, \tag{4}$$

$$w_{km} = \begin{cases} \frac{1}{h_m} & if\ Dw_{km} = 0\ and\ h_m = |\{s: Dw_{ks} = 0\}|, \\ 0 & if\quad Dw_{km} \neq 0\ but\ \exists s\ such\ that\ Dw_{ks} = 0, \\ \frac{1}{\sum_{s=1}^{M}\left[\frac{Dw_{km}}{Dw_{ks}}\right]^{\frac{1}{q-1}}} & if\quad Dw_{ks} \neq 0\ .\quad \forall 1 \le s \le M. \end{cases} \tag{5}$$

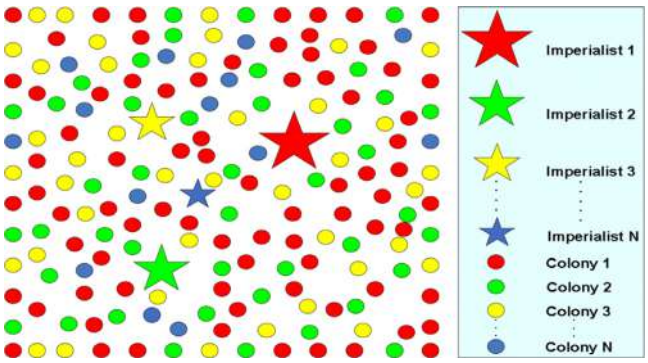

Fig. 2. An example of the initial population of empires and colonies [46].

In Eq. (5) $Dw_{km} = \sum_{n=1}^{N} {u_{nk}}^{\alpha}\left(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)\right)$.

$$z_k = \begin{cases} \frac{1}{g_k} & if\ Dz_k = 0\ and\ g_k = |\{l: Dz_l = 0\}|, \\ 0 & if\quad Dz_k \neq 0\ but\ \exists l\ such\ that\ Dz_l = 0, \\ \frac{1}{\sum_{l=1}^{K}\left[\frac{Dz_k}{Dz_l}\right]^{\frac{1}{p-1}}} & if\quad Dz_l \neq 0\ .\quad \forall 1 \le l \le \mathrm{K}. \end{cases} \tag{6}$$

In Eq. (6) $Dz_k = \sum_{n=1}^{N} {u_{nk}}^{\alpha} d_{nk}$.

2.2. ICA algorithm

ICA [46] is a population-based metaheuristic algorithm that has been used to solve a variety of optimization problems. This algorithm is inspired by the imperialist competition concept, which tries to present the social policies of imperialism to control more countries and use the resources of the colonial countries. If one empire loses its power, the other empires compete to take its possession. In the ICA, this process is simulated by individuals known as countries.

ICA starts with an initial population (i.e., the countries), and the objective function is calculated. According to the rate obtained for each country, the most powerful countries are chosen as the imperialists, and the rest are the colonies of these imperialists. Then, the imperialists start to compete with each other to seize more colonies. The best imperialist has a better chance of having more colonies. Then, each of the imperialists makes their empire with his colonies. Fig. 2 shows an example of the initial population of empires and colonies [46]. Larger empires have more colonies, and smaller empires have fewer colonies. In this figure, Imperialist 1 is the most powerful and has the largest number of colonies.

After the division of the colonies between the imperialists, these colonies approached their dependent imperialist countries. After this movement, if one of the colonies has more power than its corresponding imperialist, the imperialist and that colony will exchange. To start the competition between empires, the total value objective function for each empire must be calculated. This value depends on both the imperialist and its colonies' objective functions. Then, the competition begins, and the weakest empire loses its possession, and the powerful one tries to gain it. An empire that has lost all of its colonies will fall. Eventually, the most powerful empire will take over other empires and win the competition. This competition-based evaluation method is used as an optimization procedure.

3. Methodology

3.1. Overview of the proposed method

Fig. 3 shows a general overview of the proposed image segmentation approach. As shown in this figure, the process of segmentation begins with the feature extraction step. The extracted features are entered into the CGFFCM clustering algorithm. Along with the clustering process, the ICA is applied to calculate the parameters required in the clustering algorithm optimally. The ICA is utilized to improve the CGFFCM by automatically and optimally finding the weights of feature groups.

In the following, in Section 3.2, the extracted features from color images are introduced; in Section 3.3, the CGFFCM clustering method is described; and in Section 3.4, the way of combining the ICA algorithm with the CGFFCM clustering method is presented.

3.2. Feature extraction

By reviewing the previous research and studying the efficiency of various features used for the color image segmentation purpose [17,50,51], we propose to use an efficient combination of image features, including local homogeneity, color, and texture. In the following, each of these features is described.

(1) Local homogeneity

Local homogeneity is a feature based on the local image information and refers to the image regions' uniformity [17]. The local homogeneity of the pixel (i, j) is obtained from Eq. (7):

$$H_{ij} = 1 - E_{ij} \times \mathrm{V_{ij}}, \tag{7}$$

where, E_{ij} and $\mathrm{V_{ij}}$ indicate the discontinuity and standard deviation, respectively. To calculate the discontinuity, we first calculate the gradient using the Sobel operator and then normalize it, as Eq. (8):

$$E_{ij} = \frac{e_{ij}}{e_{max}} \quad \text{where } e_{ij} = \sqrt{G_X^2 + G_y^2}, \tag{8}$$

where, $e_{max} = max\left\{e_{ij}\right\}$, G_x, and G_y represent the image gradient in the x and y directions, respectively.

We convert the input image to the HSV color space and then, calculate the standard deviation in each color channel, according to Eq. (9):

$$V_{ij} = \frac{v_{ij}}{v_{max}}, \qquad v_{ij} = \sqrt{\frac{1}{d^2} \sum_{p=i-\left(\frac{d-1}{2}\right)}^{i+\left(\frac{d-1}{2}\right)} \sum_{q=j-\left(\frac{d-1}{2}\right)}^{j+\left(\frac{d-1}{2}\right)} \left(I_{pq} - \mu_{ij}\right)^2}, \tag{9}$$

where $v_{max} = max\left\{v_{ij}\right\}$, for an image with the size $m \times n$, $p \le m$, $1 \le i, q \le n$, and $1 \le j$. w_{ij} is a window of size $d \times d$ $(d = 5)$ with pixel center (i, j) for calculating the deviation. μ_{ij} is the mean color component of the pixel in the center of the μ_{ij} window, which is calculated using Eq. (10):

$$\mu_{ij} = \frac{1}{d^2} \sum_{p=i-\left(\frac{d-1}{2}\right)}^{i+\left(\frac{d-1}{2}\right)} \sum_{q=j-\left(\frac{d-1}{2}\right)}^{j+\left(\frac{d-1}{2}\right)} I_{pq}. \tag{10}$$

Fig. 4 shows a sample image and three local homogeneity features extracted from three color channels.

(2) Color

To extract the color feature, we use the CIELAB color space. The color component of a pixel in each CIELAB channel is considered as the color feature of that pixel. CIELAB color space provides a perceptually uniform space, meaning that the Euclidean distance between two color points in CIELAB color space corresponds to the perceptual difference between the two colors by the human visual system [49]. Thus, in various machine vision applications, the CIELAB color space has shown superior performance compared to other spaces [17,52,53]. Also, previous research results show that CIELAB is the most suitable color space for color image segmentation using FCM [16,17,54]. Fig. 5 shows a sample image and three color features extracted from CIELAB color channels.

(3) Texture

The texture is a common feature used in image segmentation. Texture components are often used together with color information to achieve better segmentation results. The combination of color and texture features in image segmentation has better results than color features alone [55]. To calculate texture components, we use the Gabor filter and Gray-Level Co-occurrence Matrix (GLCM) [56]. These two methods are very efficient and common in various applications for extracting texture features [51,57].

Analyzing an image by the Gabor filter is similar to the human visual perception system. To obtain the texture feature at the pixel level, we apply the Gabor filter to the image and extract the local energy of the filter responses, which is considered as the pixel's texture feature.

The other texture feature used in our method is the GLCM. GLCM is a set of features based on second-order statistics that are easy to implement and performs well in terms of time and complexity [56]. GLCM considers spatial correlation between image pixels [58,59]. By using GLCM, better segmentation can be achieved for images [60]. Fig. 6 shows a sample image and two features extracted from its texture components.

After extracting the above three groups of features for each pixel of the input image, an 8-dimensional feature vector, consisting of three components of local homogeneity, three color components, and two texture components, is constructed and fed into the clustering step.

3.3. CGFFCM clustering

The CGFFCM clustering algorithm is inspired by our previous work in [28]. However, in the CGFFCM, to better distinguish between different groups of features, we use a group-local feature weighting strategy to improve the clustering accuracy. In this strategy, a weight is assigned to a group of features in all clusters (group weighting), and another weight is assigned to each feature in a group of features (local weighting).

The advantage of group-local feature weighting is most evident in image segmentation, where usually different groups of features are used (e.g., the group of features extracted from the color space). A group of features may be more important than the other groups for segmenting an input image. Also, within a group, there can be significant differences between the importance of features. Therefore, assigning the same weight to all components of a group of features (for example, all color channels in CIELAB), in many cases, reduces the quality of the image segmentation.

For example, suppose that in an image, the CIELAB group of features are more important than the other groups of features. Also, within this group, the feature extracted from channel A contains more information for better segmentation of the image. By using the group-local feature weighting mechanism, the groups with more information and more important features within those groups gain more weight. In fact, to segment an image, those groups of features and sub-features that are suitable for segmenting the given image play a more effective role.

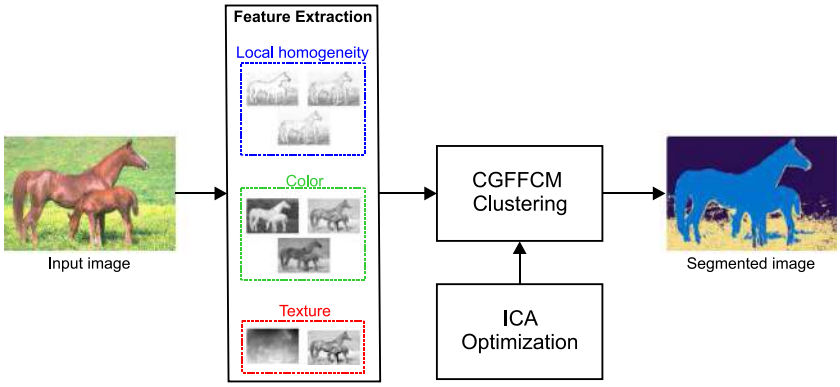

Fig. 3. Diagram of the proposed approach.

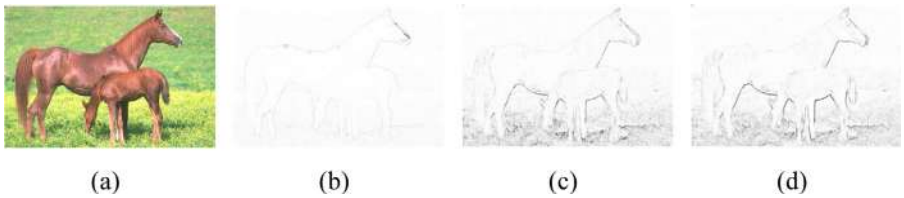

Fig. 4. Three local homogeneity features extracted from HSV color channels. (a) Input image, and the feature extracted from channel (b) H, (c) S, and (d) V.

Fig. 5. Three color features extracted from CIELAB color channels. (a) Input image, and the feature extracted from the channel (b) L, (c) A, and (d) B. (For more clarity and proper visualization, images (b) and (d) have been normalized).

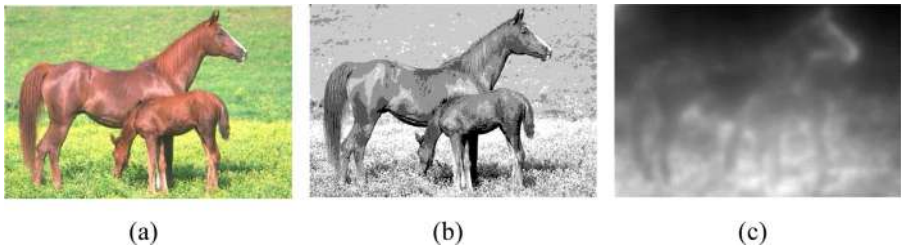

Fig. 6. Two features extracted from texture components. (a) Input image, (b) Gabor filter, and (c) GLCM.

The main difference between the CGFFCM and the method presented in [28] and other local feature weighting methods is in better identification of features that are suitable for the image segmentation task. Although both the CGFFCM and the method in [28] identify important features and give them more weight, the CGFFCM method assigns much accurate weights to the features. The experiments presented in Section 4 prove this claim.

Suppose that the three groups of features used in our approach, i.e., the local homogeneity, the CIELAB color space, and the texture features, are shown with the following names, respectively: $G(1) = \{HF\}$, $G(2) = \{LabF\}$, and $G(3) = \{TF\}$. A vector of group weight coefficients called $\boldsymbol{V} = [v_g]$ having the length of G (here, $G = 3$) is defined, where v_g represents the weight coefficient of the features in group g.

Also, Eq. (11) shows the objective function used in the CGFFCM method:

$$F(\boldsymbol{U}, \boldsymbol{C}, \boldsymbol{W}, \boldsymbol{Z}) = \sum_{n=1}^{N}\sum_{k=1}^{K} {u_{nk}}^{\alpha} z_k^p d_{nk}^{(gw)}, \tag{11}$$

where, $d_{nk}^{(gw)}$ is a new non-Euclidean distance measure based on group-local feature weighting and is defined as Eq. (12):

$$\begin{aligned} d_{nk}^{(gw)} &= \sum_{g=1}^{G} v_g \sum_{m\in\{\mathrm{HF},LabF,TF\}} w_{km}^{q}\left(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)\right) \\ &= v_1 \sum_{m\in\mathrm{HF}} w_{km}^{q}\left(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)\right) \\ &\quad + v_2 \sum_{m\in LabF} w_{km}^{q}\left(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)\right) \end{aligned}$$

$$+ v_3 \sum_{m\in TF} w_{km}^{q}(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)). \tag{12}$$

The following constraints are assumed for the objective function in Eq. (11):

$$\begin{aligned}
&u_{nk} \in [0,1]\,.\sum_{k=1}^{K} u_{nk} = 1, \qquad where\ 1 \le n \le N\ and\ 1 \le k \le K;\\
&z_k \in [0,1]\,.\sum_{k=1}^{K} z_k = 1, \qquad where\ 1 \le k \le K;\\
&v_g \in [0,1]\,.\sum_{g=1}^{G} v_g = 1, \qquad where\ 1 \le g \le G;\\
&w_{km} \in [0,1]\,.\sum_{m\in\{\mathrm{HF},LabF,TF\}} w_{km} = 1,\\
&\quad where\ 1 \le k \le K\ and\ 1 \le m \le M.
\end{aligned} \tag{13}$$

The Lagrangian of Eq. (11) with the constraints in Eq. (13) is Eq. (14):

$$\begin{aligned}
\tilde{F} = &\sum_{n=1}^{N}\sum_{k=1}^{K} {u_{nk}}^{\alpha} w_{km}^{q} z_k^{p} d_{nk}^{(gw)} - \delta\left(\sum_{k=1}^{K} u_{nk} - 1\right)\\
&- \sum_{g=1}^{G} \psi_g \left(\sum_{m\in\{\mathrm{HF},LabF,TF\}} w_{km} - 1\right) - \omega\left(\sum_{k=1}^{K} z_k - 1\right),
\end{aligned} \tag{14}$$

where, δ, ψ_g and ω stand for the parameters of the Lagrangian multiplier. By solving the Lagrange equation, the updating equations for u_{nk}, c_{km}, w_{km}, and z_k are obtained (Eqs. (15) to (18)).

$$u_{nk} = \frac{1}{\sum_{l=1}^{K}\left[\frac{z_k^{p} d_{nk}^{(gw)}}{z_l^{p} d_{nl}^{(gw)}}\right]^{\frac{1}{(\alpha-1)}}}, \tag{15}$$

$$c_{km} = \frac{\sum_{n=1}^{N} {u_{nk}}^{\alpha} \exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right) x_{nm}}{\sum_{n=1}^{N} {u_{nk}}^{\alpha} \exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right)}, \tag{16}$$

$$w_{km} = \begin{cases} \frac{1}{h_m} & if\ Dw_{km} = 0\ and\ h_m = |\{s: Dw_{ks} = 0\}| \\ 0 & if\ Dw_{km} \neq 0\ but\ \exists s\ such\ that\ Dw_{ks} = 0 \\ \frac{1}{\sum_{s\in\{\mathrm{HF},LabF,TF\}}\left[\frac{Dw_{km}}{Dw_{ks}}\right]^{\frac{1}{q-1}}} & if\quad Dw_{ks} \neq 0\,.\quad \forall 1 \le s \le M, \end{cases} \tag{17}$$

where, $Dw_{km} = \sum_{n=1}^{N} {u_{nk}}^{\alpha}(1 - exp\left(-\gamma_m \left(x_{nm} - c_{km}\right)^2\right))$.

$$z_k = \begin{cases} \frac{1}{g_k} & if\ Dz_k = 0\ and\ g_k = |\{l: Dz_l = 0\}| \\ 0 & if\quad Dz_k \neq 0\ but\ \exists l\ such\ that\ Dz_l = 0 \\ \frac{1}{\sum_{l=1}^{K}\left[\frac{Dz_k}{Dz_l}\right]^{\frac{1}{p-1}}} & if\quad Dz_l \neq 0\,.\quad \forall 1 \le l \le \mathrm{K}, \end{cases} \tag{18}$$

where, $Dz_k = \sum_{n=1}^{N} {u_{nk}}^{\alpha} d_{nk}^{(gw)}$.

Similar to the method presented in [28], to find the appropriate value of p, we apply an iteration-based approach using three parameters p_{init}, p_{step}, and p_{max}. We start the algorithm using a small value (p_{init}) for p. In each iteration, we increase p as much as p_{step} until the maximum value (p_{max}) is obtained. If an empty cluster or a cluster with one sample appears, we decrease p by p_{step} regardless of whether p equals p_{max} or not. At this point, we choose the values of u_{nk}, w_{km}, and z_{km} corresponding to the previous p. The algorithm continues until in two successive iterations, the difference between the two objective function values is less than the threshold value ε, or the number of iterations reaches the maximum (t_{max}). The pseudo-code of the algorithm is presented in Fig. 7.

3.4. Combining CGFFCM with ICA

We use the ICA algorithm to automatically find the optimum values of the weight coefficients vector ($\boldsymbol{V}$) in the CGFFCM algorithm. The algorithm of this process consists of 10 steps, as follows:

Step 1: Generate an initial population:

The initial population of the coefficient vector is generated randomly by Eq. (19).

$$population = \begin{bmatrix} \boldsymbol{V}_1 \\ \boldsymbol{V}_2 \\ \boldsymbol{V}_3 \\ \dots \\ \boldsymbol{V}_{N_{inital}} \end{bmatrix}$$

$$\boldsymbol{V}_i = Country_i = [v_1, v_2, v_3], \quad i = 1, 2, \dots, N_{inital} \tag{19}$$

where, N_{inital} indicates the number of population, $\boldsymbol{V}_i$ is a vector of length three and represents the country i. Its elements, i.e., v_1, v_2, and v_3, indicate the weight coefficients of group 1, group 2, and group 3 (i.e., three groups of features), respectively. Each of the elements is in the range [0.1, 0.8], where $\sum_{g=1}^{G} v_g = 1$. Considering the range [0.1, 0.8], all groups are assigned with a weight coefficient greater than zero. In other words, all groups of features are involved in the clustering process. If the range [0,1] or [0, 0.9] was selected, in the first case, a zero weight was assigned for two groups, and in the second case, a zero weight was given to one group. Assigning zero weight to a group means that group of features is insignificant and left out in the clustering process. Based on the experiments (presented in Section 4.7), considering the range [0.1, 0.8], the lowest weight assigned to the groups is approximately 0.1. Hence, we can conclude that a proper range in our algorithm is [0.1, 0.8]. We use this interval only to create primary countries, so by running the ICA algorithm, the group(s) may receive zero weight(s).

Step 2: Calculate the objective function value:

We run the CGFFCM algorithm (see Fig. 7) once for each $\boldsymbol{V}_i$ and compute the objective function in Eq. (11) for each $\boldsymbol{V}_i$.

Step 3: Select the imperialists:

The initial population is sorted ascendingly according to their objective function's value. Countries with the least objective function are chosen as imperialists, and the rest are chosen as the colonies of these imperialists. In our experiments, the number of empires is considered according to the initial population in the range $N_{imp} = [10, 20]$.

Step 4: Select the imperialist states:

The colonies are divided among imperialists according to the power of each imperialist. This power is calculated by Eqs. (20) to (22) for each imperialist:

$$C_n = min_i\{C_i\} - Cost_n, \tag{20}$$

$$P_n = \left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right|, \tag{21}$$

$$C_n^{norm} = \mathrm{round}\ \{P_n(N_{col})\}, \tag{22}$$

where, $Cost_n$ is the value of the nth imperial objective function, and C_n is its normalized value.

Step 5: Move the colonies towards their imperialists:

As explained in Section 2.2, after dividing the colonies among the imperialists, the colonies move closer to their corresponding imperialist countries. Fig. 8 shows an example of this movement.

Algorithm 1. CGFFCM clustering algorithm.

Input: Dataset $\boldsymbol{\chi} = \{x_n\}_{n=1}^{N}$, Initial centers $C^{(0)}$, Number of clusters K (Number of color regions), Number of features M, Secondary parameters t_{max}, $p_{max}, p_{init}, p_{step}$, ε, Exponent of feature weight q, Fuzzy degree α
Output: Membership matrix U, Cluster centers matrix C

set $t = 0$
set $p_{init} = 0$
set $z_k^{(0)} = \frac{1}{K}$, $\forall k = 1 \dots K$
set $w_{km}^{(0)} = \frac{1}{M}$, $\forall k = 1 \dots K$, $\forall m = 1 \dots M$
set empty= **FALSE** //No empty or singleton clusters yet detected
$p = p_{init}$
repeat
$t = t + 1$
Update the cluster assignments matrix $\boldsymbol{U}$ by (15)
if empty or singleton clusters have occurred at time t **then** //reduce p.
    empty=**TRUE**
    $p = p - p_{step}$
    if $p < p_{init}$ **then**
        return NULL
    end if
//Revert to the assignments and weights corresponding to the reduce p.
$u_{nk}^{(t)} = \left[\mathbf{U\_history}^{(p)}\right]_{nk}, \forall k = 1 \dots K$, $\forall n = 1 \dots N$
$z_k^{(t-1)} = \left[\boldsymbol{Z}\mathbf{\_history}^{(p)}\right]_{k}, \forall k = 1 \dots K$
$w_{km}^{(t-1)} = \left[\boldsymbol{w}\mathbf{\_history}^{(p)}\right]_{km}, \forall m = 1 \dots M$, $\forall k = 1 \dots K$
end if
Update the cluster center matrix $\boldsymbol{C}$ by (16)
if $p < p_{max}$ **and** empty=**FALSE** then //increase p.
    $\mathbf{U\_history}^{(p)} = [u_{nk}^{(t)}]$
    $\boldsymbol{W}\mathbf{\_history}^{(p)} = [w_{km}^{(t-1)}]$
    $\boldsymbol{Z}\mathbf{\_history}^{(p)} = [z_k^{(t-1)}]$
    $p = p + p_{step}$
end if
Update the feature weight matrix $\boldsymbol{W}$ by (17)
Update the cluster weight matrix $\boldsymbol{Z}$ by (18)
until $\left|F^{(t)} - F^{(t-1)}\right| < \varepsilon$ ***or*** $t \geq t_{max}$
return $\boldsymbol{U}, \boldsymbol{C}$

Fig. 7. The pseudo-code of the CGFFCM algorithm.

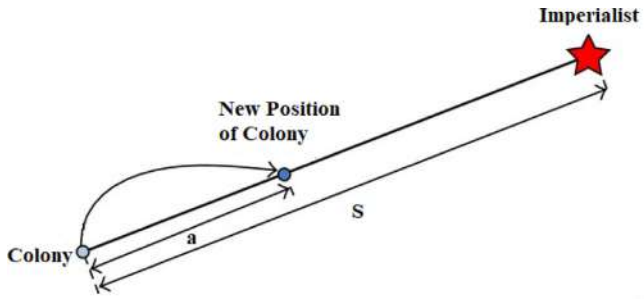

Fig. 8. An example movement of a colony towards its imperialist [46].

As shown in this figure, each colony moves a unit towards the corresponding imperialist and reaches its new position. a is a random variable with a uniform distribution (see Eq. (23)).

$$a \sim U(0, \beta \times S), \tag{23}$$

where, β (greater than 1) causes the colonies to move in different directions towards their imperialist, and S is the distance between the colony and the imperialist. In Ref. [46], it is shown that the best value to reach the global minimum fastly is $\beta = 2$. We also use this value in our experiments.

However, the colonies do not move exactly straight but deviate from the original position with θ angle (see Fig. 9). We choose this angle randomly and with a uniform distribution (see Eq. (24)).

$$\theta \in U(-\gamma . \gamma), \tag{24}$$

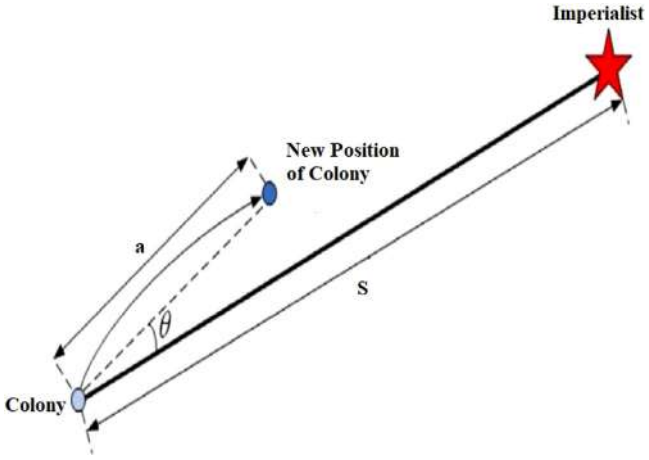

Fig. 9. An example movement of a colony towards their corresponding imperialists in a random direction [46].

where, γ is a parameter that adjusts the deviation from the original direction. Considering the radian unit for θ, a number close to $\pi/4$, is a good choice to reach the global minimum faster [46].

Step 6: Calculate the cost of all colonies in each empire:

Because the colonies have been placed in a new position, each colony's cost may have changed. Thus, we recalculate all colonies' cost (we run the algorithm in Fig. 7 for each colony again, and obtain the value of the objective function for each colony). If the

cost of a colony is less than its imperialist, the colony's position and the imperialist will exchange.

Step 7: Calculate the total cost of each empire:

The cost of each empire depends on the imperialist power and its colonies that is calculated by Eq. (25).

$$TC_n = Cost\ (imperialist_n) + \xi mean\{Cost(colones\ of\ empire_n)\}, \quad (25)$$

where, TC_n is the total cost of the nth empire, ξ is an attenuation coefficient in [0,1] to reduce the effect of colonies cost. In our experiments, the value of ξ is set in the range of $\xi = [0.05, 0.5]$. A small value for ξ causes the total power of the empire to be determined by just the imperialist, and increasing it will increase the role of the colonies in determining the total power of an empire. Inspired by [46,61], we also use the value of 0.1 in our experiments.

Step 8: Empires compete:

All empires, according to their power (total cost), try to get the colonies of the weakest empires (Eqs. (26) and (27)):

$$TC_n^{norm} = TC_n - max_j\{TC_j\}, \quad (26)$$

$$P_{P_n} = \left| \frac{TC_n^{norm}}{\sum_{j=1}^{N_{imp}} TC_j^{norm}} \right|, \quad (27)$$

where, TC_n^{norm} is the total normalized cost of the nth empire, and P_{P_n} is the possession probability of each empire.

The roulette wheel is used to select the winning empire randomly. The victorious empire will dominate the weakest colony of the weakest empire. To perform the roulette wheel, it is necessary to calculate the cumulative probability (see Eq. (28)).

$$C_{P_n} = \sum_{i=1}^{n} P_{P_i}, \quad (28)$$

A random number is then generated with a uniform distribution and compared with all C_{P_n}. Each sector with a higher probability will have more chances to be chosen; thus, the winning empire is determined. As mentioned in using the roulette wheel, it is necessary to calculate the cumulative distribution. To reduce this time-consuming step, the following approach is applied (Eqs. (29) to (31)):

$$\boldsymbol{P} = \left[P_{P_1}, P_{P_2}, \ldots, P_{P_{N_{imp}}}\right], \quad (29)$$

$$\boldsymbol{R} = \left[r_1, r_2, \ldots, r_{N_{imp}}\right] \sim U(0, 1), \quad (30)$$

$$\boldsymbol{D} = \boldsymbol{P} - \boldsymbol{R} = \left[D_1, D_2, \ldots, D_{N_{imp}}\right] = \left[P_{P_1} - r_1, P_{P_2} - r_2, \ldots, P_{P_{N_{imp}}} - r_{N_{imp}}\right], \quad (31)$$

where, $\boldsymbol{P}$ is the vector of possession probability of all empires, and $\boldsymbol{R}$ is a vector with uniformly distributed random numbers. The maximum index in $\boldsymbol{D}$ shows the winner empire that gets the colony. Once the winning empire is determined, the weakest colony of the weakest empire is given to the winning empire. So we have to eliminate one of the weak empire populations and add them to the winning population.

Step 9: Eliminate the weak empires:

If there is an empire without a colony, that empire is eliminated, and its imperialist is given to the best empire.

Step 10: Check the termination condition:

If only one empire remained, stop. Else go to *Step 5*.

Finally, after the termination of the algorithm, the resulting clusters are considered as the segments of the input image. The implementation (source code) of CGFFCM is available at https://github.com/Amin-Golzari-Oskouei/CGFFCM.

3.5. Datasets

To evaluate the performance of the proposed method and compare it with other approaches, we use the benchmark Berkeley dataset [48]available at https://github.com/BIDS/BSDS500.git. Fig. 10 shows some of the images in this dataset and their ground truths. The set of tested images is also available at https://github.com/Amin-Golzari-Oskouei/CGFFCM.

3.6. Remarks

The computational complexity of CGFFCM (Algorithm 1), independent of the ICA optimization and feature extraction steps, depends on four updating stages: updating membership matrix **U**, centers matrix **C**, weights matrix **W**, and clusters **Z**. The computational complexity of each stage is equal to NKM, where N refers to the number of data samples, K is the number of clusters, and M is the number of features. Since each stage is run separately, the overall computational complexity is equal to $t\ (NMK + NMK + NMK + NMK) = 4tNMK$ where t is the number of iterations.

4. Results and discussion

In this section, the performance of the proposed approach is evaluated. The results are compared with the following methods: the FCM [62], the robust local feature weighting hard c-means (RLFWHCM) [43], the fuzzy c-means clustering with the entropy of feature weight (EWFCM) [42], the fuzzy c-means clustering method based on feature-weight and cluster-weight Learning [28], the feature group weighted fuzzy k-harmonic means (GWFKHM) [17], the feature weighted k-harmonic means (WKHM) [17], the semi-supervised surrogate-assisted multi-objective kernel intuitionistic fuzzy clustering (SMKIFC) [20], the alternate particle swarm optimization based adaptive interval type-2 intuitionistic FCM clustering algorithm (A-PSO-IT2IFCM) [33], semi-supervised fuzzy clustering algorithm with spatial constraints [21] (SSFC-SC), kriging-assisted reference vector guided evolutionary algorithm (KRVEA) [40], feature-weighted fuzzy c-means-based method (IFWFCM) in [16], clustering based on the combination of FCM and ICA [36], and clustering based on the combination of FCM, PSO, and ICA algorithms [35]. In addition, in this section, the detailed results of the experiments are provided to demonstrate the effectiveness of the extracted features in the proposed method. Furthermore, the results of the experiments related to the effectiveness of the group-local feature weighting scheme and group weight coefficient vector are presented.

Parameter ε and the maximum number of iterations are common in the implemented methods, which are set to 10^{-5} and 200, respectively. The parameter α in fuzzy algorithms is set to 2. In the proposed algorithm and algorithm in [28], the required parameters are set as follows: $p_{step} = 0.01$, $p_{init} = 0$, $p_{max} = 0.5$, and $q = 2$. The number of color regions (the number of clusters) for each testing image (introduced in Section 4.1) is the same as the number of color regions labeled in the dataset.

4.1. Performance metrics

In the experiments, the following three metrics are used to measure the performance of algorithms.

Accuracy: This metric is used when the cluster's structure is determined in the dataset [63], which here are the colored labeled regions. *Accuracy* is defined as the number of pixels clustered correctly divided by the number of all pixels. *Accuracy* measures the similarity between the obtained segmentation result and the

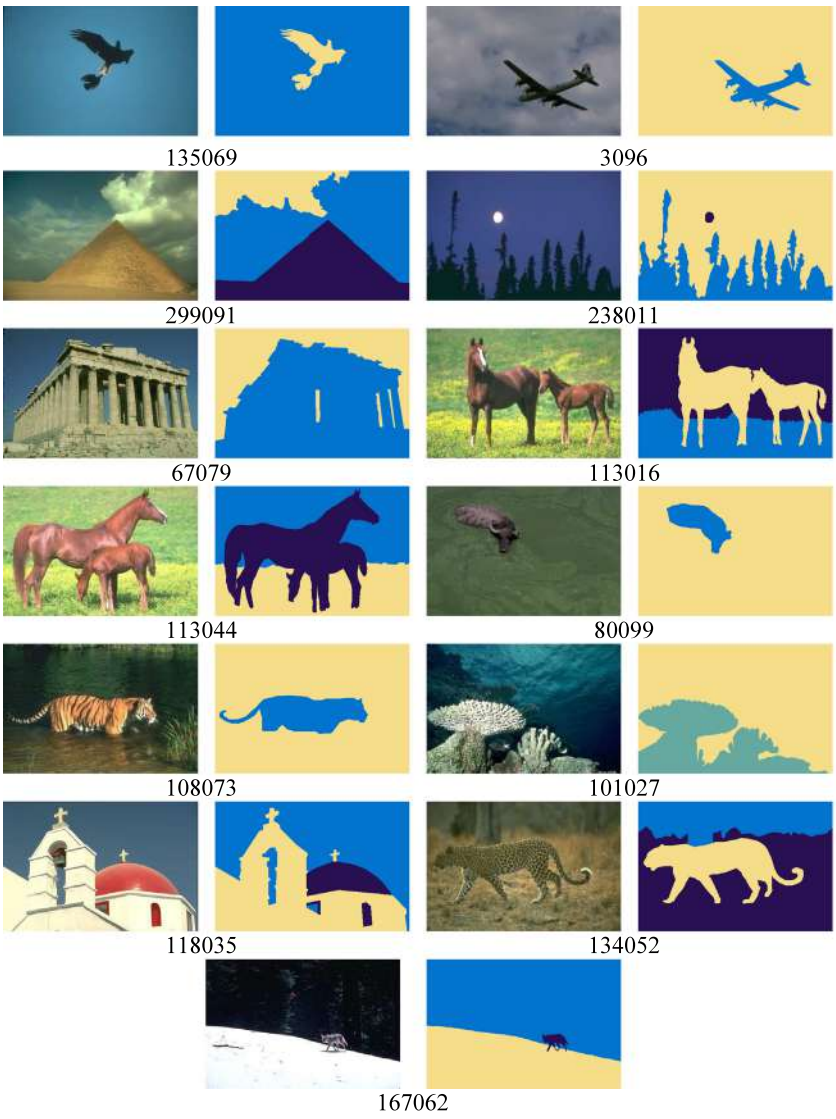

Fig. 10. Sample images from the Berkeley dataset and the corresponding ground truth images.

ground truth. This metric works well when the color regions are balanced (balanced clusters). This criterion reports incorrect results for evaluating imbalanced color regions (imbalanced clusters). This metric is calculated by Eq. (32).

$$ACC = \frac{\sum_{k=1}^{K} d_k}{N} \times 100, \tag{32}$$

where, d_k represents the number of pixels that are correctly located in the kth colored region. K is the number of clusters (regions), and N represents the total number of pixels in the image.

Normalized Mutual Information (NMI): This is a symmetric metric used to measure the common information between two clusters' (regions) distributions [64]. *NMI* measures the mutual information between the obtained segmentation result and the ground truth. This is a good measure for determining the quality of clustering. Since it is normalized, we can measure and compare it between different numbers of color regions. This criterion is defined by Eq. (33).

$$NMI\,(R, Q) = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} p\,(i,j) \log \frac{p(i,j)}{p(i)p(j)}}{\sqrt{H\,(R)\,H(Q)}} \times 100, \tag{33}$$

where, R and Q are two partitions of the input image, including I and J regions, respectively. $P\,(i)$ is the probability that a randomly selected pixel from the input data is assigned to R_i. $P\,(i,j)$ is the probability that a sample belongs to both regions R_i and Q_i. $H\,(R)$ is the entropy associated with all the probabilities $P\,(i)\,(1 \leq i \leq I)$ in partition R.

F-score: This criterion is used to evaluate the segmentation results quantitatively. *F-score* provides a way to combine both precision and recall into a single measure that captures both properties. We can have excellent precision with terrible recall, or alternately, terrible precision with excellent recall. *F-score* provides a way to express both concerns with a single score. So, this is the harmonic mean of the two fractions. Unlike the *Accuracy* criterion, *F-Score* is the most common metric used on imbalanced problems. Therefore, it is a good criterion for evaluating imbalanced color regions. This metric is calculated by Eq. (34).

$$F = \frac{2TP}{2TP + FN + FP} \times 100, \tag{34}$$

where, *TP*, *FP*, and *FN* denote the true positive, false positive, and false negative rates in the segmentation result. A higher value of the *F-score* means a better segmentation result.

Performance improvement: This criterion represents the percentage of improvement obtained by the proposed method (CGFFCM) on each image compared to the other algorithms' best result. For each performance metric (i.e., *Accuracy*, *NMI*, and *F-score*), the performance improvement criterion is calculated separately by Eq. (35).

$$Performance\ improvement = \frac{R_{our} - R_{other}}{R_{other}} \times 100, \tag{35}$$

where, R_{our} is the results of the proposed method, and R_{other} is the other algorithms' best result.

In our experiments, these metrics are expressed as a percentage. A high percentage indicates a better performance.

4.2. ICA parameter tuning

Besides the advantages that evolutionary algorithms have, they usually have several parameters that should be tuned well. For example, in the ICA algorithm, the number of empires or the number of initial populations may affect the final result. However, some parameters have less impact on the final results. In the proposed method, appropriate values are set for the effective parameters (such as the number of population and the number of empires), and for the other parameters, the value proposed in the main ICA article has been used. In this section, we discuss how to adjust the number of population and the number of empires parameters that existed in the ICA algorithm.

Number of population: We run CGFFCM for different populations in the range [10,150] by step 10. The objective function values of CGFFCM for the different populations on all tested images are shown in Fig. 11. In this figure, for each image, the values of the objective function are normalized between 0 and 1. As shown in this figure for a small number of populations, the value of the objective function is very large. As the population grows, the value of the objective function decreases. This decrease starts approximately from the population of 70 and continues until the end. The point is that for more than 100 populations, the value of the objective function does not decrease significantly. Therefore, in CGFFCM, the range between 70 and 100 is selected as the appropriate range for the population. Of course, for some images, the value of the objective function may not be the lowest in that range. Our goal is to select the range with the lowest value of the objective function for all images on average. On average, all values of the objective functions are high before this interval and do not decrease significantly after this interval, so we choose the average population value for CGFFCM. Hence, we can conclude that a proper value of $N_{initial}$ in our algorithm is equal to 85—the average value of the above-stated range between 70 and 100.

Number of empires: To find the appropriate range for the number of empires, we perform an experiment similar to the previous one. In this experiment, we calculate the value of the objective function for the number of empires in the interval [2,40]. The results are shown in Fig. 12. As the number of empires grows, the value of the objective function decreases. This decrease starts approximately from the population of 15 and continues until the end. The point is that for more than 20 populations, the value of the objective function does not decrease significantly. Therefore, in CGFFCM, the range [15,20] is selected as the appropriate range for the number of empires. On average, all values of the objective functions are high before this interval and do not decrease significantly after this interval. Hence, we can conclude that an acceptable range for *N_imp* in CGFFCM is [15,20].

4.3. Experiment 1: Results with CIELAB color space

As stated before, CIELAB color space is the most suitable color space for the clustering-based image segmentation methods [16,17]. In this experiment, the performance of CGFFCM and seven other methods, the IFWFCM, WKHM, FCM, EWFCM, RLFWHCM, [35,36], are evaluated using the only CIELAB color features. The purpose of this experiment is to evaluate the performance of CGFFCM, regardless of using the other extracted features (i.e., texture and local homogeneity). Table 1 compares the results of different methods. This table shows the *Accuracy*, *NMI*, and *F-score* rates from top to bottom for each image, respectively. Also, the percentage of improvement obtained by CGFFCM on each image compared to the other methods' best result is shown in Table 1. The results for the FCM, EWFCM, RLFWHCM, [35,36] methods have been obtained through our fair implementation, and the results for WKHM and IFWFCM have been quoted directly from the relevant publications. In Table 1, the best rates are bold-faced. Some references did not use the images and criteria to evaluate the performance of their proposed method. So, for unavailable values (unreported values), we use the symbol "-" to show that results are not available for a method. To compare the visual performance of the methods, the segmentation results of each method are shown in Fig. 13.

From Table 1 and Fig. 13, it is evident that CGFFCM performs better than other methods. It performs better for all evaluation metrics on all images (except the 108073, and 80099). Since CGFFCM is not sensitive to the selection of initial centers, it can provide more stable results compared to the other methods.

For images with balanced segmented regions, most of the compared methods have almost similar results. In such images, the *Accuracy* metric is a good metric to show the quality of the segmentation. In images 113044, 134052, 101027 and 299091 the average *Accuracy* of CGFFCM is 83.42%, while for FCM, EWFCM, RLFWHCM, [35,36] methods, the average *Accuracy* is 77.03%, 72.67%, 70.83%, 77.95% and 77.18%, respectively.

As stated before, in images with imbalanced segmented regions, the *F-score* and *NMI* metrics are appropriate metrics to show the quality of the segmentation. In images 135069, 238011, and 3096 (images with imbalanced segmented regions), the average *NMI* and *F-score* metrics of CGFFCM are 80.97% and 86.13%, respectively. For FCM, EWFCM, RLFWHCM, [35,36] methods, the average *NMI* is 52.80%, 70.27%, 58.13%, 76.01%, and 76.26, and the average *F-score* is 77.01%, 82.73%, 83.68%, 82.66%, and 82.73% respectively. These results indicate that for segmenting images with balanced/imbalanced regions, CGFFCM performs much better than the compared methods.

We observe from Fig. 13, the proposed method outperforms other methods in terms of miss-segmentation of object parts. For instance, our method successfully segments the "Moon" in image 238011 and the "Eagle" in image 135069, whereas the FCM and RLFWHCM methods miss-segment the "Moon" object. Also, the RLFWHCM method miss-segment the "Eagle" object, too. In EWFCM, the "Sky" and "Ground" objects, in image 113016, are in the same segment, while these two parts are completely different colors. These results reveal that using only the CIELAB color space feature may not improve the segmentation accuracy. In addition, such a CIELAB color space feature is not informative enough to capture essential region information, especially in complex images with low color contrast.

4.4. Experiment 2: Experiment with all extracted features

In this section, we use all of the extracted features to segment the images. We compare the results of CGFFCM with six methods: FCM, EWFCM, RLFWHCM, [28,35,36] in Table 2. This table shows

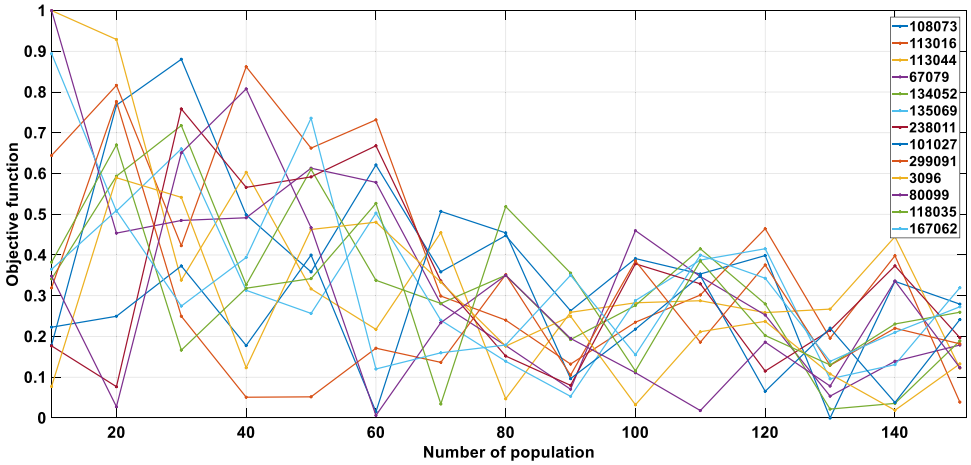

Fig. 11. Value of objective function of CGFFCM for different number of populations on all tested images.

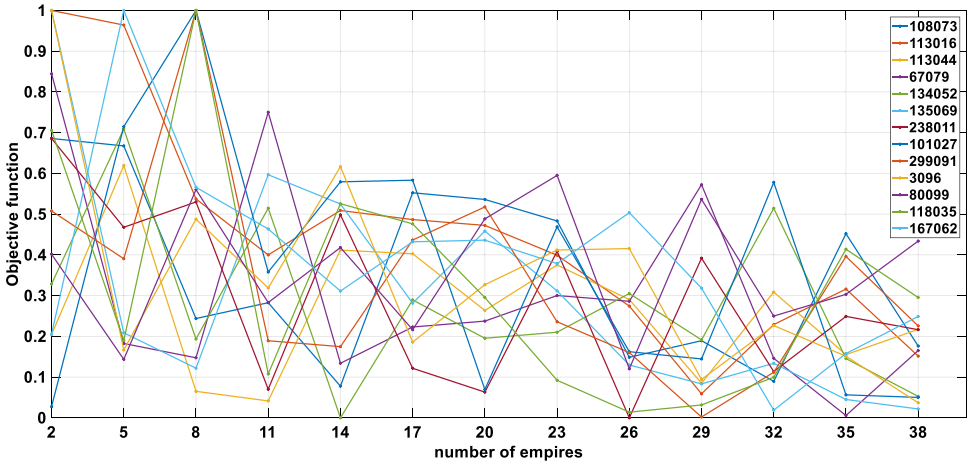

Fig. 12. Value of object function of CGFFCM for different number of empires on all tested images.

the *Accuracy*, *NMI*, and *F-score* rates from top to bottom for each image, respectively. The percentage of improvement obtained by CGFFCM on each image compared to the other algorithms' best result is also reported in Table 2. In addition, to compare the visual performance of the methods, the segmentation results of different methods are shown in Fig. 14.

According to Table 2, the feature weighting-based methods have better results than the metaheuristic-based methods. The average accuracy of EWFCM, RLFWHCM, [28], and CGFFCM is 92.19, 92.26, 92.64, and 95.02, respectively, while the average accuracy of metaheuristic-based methods in [35] and [36] is 89.25 and 88.52 respectively. Similarly, for other criteria, two methods [35,36] have worse results than the feature weighting-based methods. This is because, in these methods, the problem of giving equal importance to features remains.

Table 2 and Fig. 14 indicate that FCM cannot precisely segment some of the regions within specific images. For example, the moon in image 238011 and background images 108073, 134052, and 101027. In contrast, CGFFCM can successfully address the problems as mention earlier. Moreover, the CGFFCM has the best performance for all images except images 67079, 118035, and 167062. Similar to the CGFFCM, the other feature-weighting-based methods (i.e., EWFCM, [28], and RLFWHCM) have performed relatively well. The results show that the basic FCM algorithm has the worst performance. Comparing the results in Tables 1 and 2, it can be concluded that the combination of three groups of features applied in our approach has better segmentation results than considering only the CIELAB feature. This fact confirms the efficiency of the proposed combination of the features used in our approach.

Moreover, some important points we observe from Fig. 14, as follows:

(1) The segmentation results of all methods are visually better than their results using only the color feature (Fig. 13). As stated earlier, this is mainly because all methods use local homogeneity, color space, and texture features to capture essential image information and segment images with low color contrast and complex textured patterns. Moreover, using local homogeneity, color space, and texture features provides additional information, which can improve the segmentation accuracy.

(2) From the viewpoint of human perception, for images 108073, 113016, 113044, and 134052, the result of CGFFCM can represent segmented regions of the background correctly, while the other methods cannot. For complex images with small unbalanced color regions, such as 135069, 238011, 3096, and 80099, the segmentation results of CGFFCM on the small regions are better than the other approaches. The cluster weighting technique also makes the proposed algorithm not sensitive to non-homogeneous patterns. For instance, the textures of natural scenes in images 101027, 134052, 108073, and 299091 are non-uniform (e.g., shadows of the "Water" and "Leaf" regions). We

Table 1
The *Accuracy*, *NMI*, and *F-score* rates obtained by different methods using the CIELAB color space on each testing image.

Image	FCM	EWFCM	RLFWHCM	WKHM	IFWFCM	[35]	[36]	Ours (CGFFCM)	Performance improvement
108073	81.40	79.36	80.68	46.92	44.20	**87.55**	83.26	82.52	−5.74
	21.02	15.78	22.72	-	-	**26.81**	21.43	23.34	−12.94
	53.61	48.86	54.12	-	-	**60.97**	54.95	55.77	−8.52
113016	82.57	53.01	73.32	**84.68**	59.79	82.77	82.56	84.02	−0.77
	54.95	41.02	48.12	-	-	55.70	54.90	**58.16**	**4.41**
	82.93	54.77	75.25	-	-	83.03	82.93	**84.86**	**2.20**
113044	83.52	73.95	59.12	82.88	85.17	83.50	83.50	**88.14**	**3.48**
	61.87	47.42	49.83	-	-	62.80	62.80	**66.37**	**5.68**
	82.39	74.34	57.29	-	-	82.02	82.02	**88.00**	**6.80**
67079	88.55	88.65	93.50	-	-	88.55	88.54	**98.48**	**5.32**
	57.04	57.02	65.01	-	-	57.03	57.03	**89.16**	**37.14**
	90.81	90.90	95.07	-	-	90.80	90.81	**98.87**	**3.99**
134052	57.73	57.53	66.15	54.77	52.15	60.31	57.73	**66.48**	**0.49**
	32.42	31.26	38.90	-	-	32.31	32.42	**39.81**	**2.33**
	53.49	57.62	67.06	-	-	54.60	53.49	**67.10**	**0.05**
135069	71.59	98.41	95.71	99.39	99.39	99.28	99.29	**99.58**	**0.19**
	12.25	69.14	24.80	-	-	82.58	82.66	**88.62**	**7.21**
	82.45	99.17	97.77	-	-	99.62	99.62	**99.78**	**0.16**
238011	88.60	88.86	90.34	88.91	88.76	88.62	88.60	**91.72**	**1.52**
	68.36	68.42	67.91	-	-	67.68	68.36	**69.74**	**1.92**
	58.74	61.22	60.97	-	-	58.53	58.74	**64.40**	**5.19**
101027	87.76	87.65	95.75	-	-	88.24	87.76	**98.12**	**2.47**
	40.73	40.42	72.03	-	-	42.24	40.73	**85.12**	**18.17**
	78.86	78.74	92.73	-	-	79.18	78.86	**96.88**	**4.47**
299091	79.13	71.58	62.33	71.62	71.63	79.76	79.76	**80.96**	**1.50**
	66.71	64.90	42.86	-	-	67.40	67.40	**68.66**	**1.86**
	78.37	54.49	57.77	-	-	78.94	78.94	**80.03**	**1.38**
3096	98.85	98.62	99.11	98.06	0.9807	98.85	98.85	**99.31**	**0.20**
	77.79	73.27	81.69	-	-	77.78	77.78	**84.57**	**3.52**
	89.84	87.80	92.30	-	-	89.84	89.84	**94.23**	**2.09**
80099	99.62	99.62	98.07	99.65	99.65	99.66	**99.67**	99.66	−0.01
	90.73	90.45	64.61	-	-	91.34	**91.42**	91.26	−0.17
	96.91	96.90	82.04	-	-	97.25	**97.29**	97.23	−0.06
118035	95.23	95.32	91.87	-	-	95.22	95.22	**96.52**	**1.25**
	80.21	80.36	72.54	-	-	80.20	80.20	**83.54**	**3.95**
	95.34	95.32	86.26	-	-	95.34	95.34	**96.27**	**0.97**
167062	89.63	95.10	93.07	-	-	95.54	95.54	**95.62**	**0.08**
	79.12	85.04	82.43	-	-	85.70	85.70	**85.76**	**0.07**
	67.27	72.61	70.41	-	-	**73.37**	**73.37**	73.29	−0.10

observe that all methods have unsatisfactory segmentation performance, whereas CGFFCM performs well in segmenting noisy non-homogeneous regions.

(3) It is evident from Fig. 14 and Table 2, all feature-weighted methods (i.e., EWFCM, RLFWHCM, [28], and CGFFCM) are visually and quantitatively better than the FCM, [35,36] methods. Also, the results imply that the feature weighting schema used in other methods increases the segmentation results less than CGFFCM. This is explained by the fact that in the proposed group-local feature weighting technique, the optimal weights of groups and sub-features are obtained, which directly affects the segmentation results.

4.5. Experiment 3: The effect of group-local feature weighting scheme

To investigate the effect of the group-local feature weighting scheme used in our approach on the final image segmentation quality, we evaluate the performance of the proposed approach once with group-local feature weighting (proposed algorithm) and once without group-local feature weighting. In the case of without group-local feature weighting, we assign the same weights to all features in each cluster and each group. These weights are fixed during the algorithm run and are not updated. Fig. 15 shows the results of this experiment on different images in terms of the *Accuracy* metric. Also, to compare the visual performance of both cases, we compare the segmentation results in Fig. 16.

As shown in Figs. 15 and 16, the proposed approach has much better performance with the group-local feature weighting technique than without weighting mode. By using the group-local feature weighting, the *Accuracy* rate of the proposed approach is improved by an average of 12% on all testing images. For some images, such as 134052, 135069, and 167062, the effect of the group-local feature weighting technique is more significant than other images. As shown in Fig. 16, in the case of without group-local feature weighting, serious miss-segmentation is caused in background regions. In addition, in image 167062, the "Wolf" object is not correctly segmented. It is quite clear that in the second case, these problems are mainly solved, and segmentation results are improved. In these images, only one group of features (or sub-features) is very important, and using these features is enough to achieve better results. Incorporating all the features equally (without group-local feature weighting technique) dramatically reduces the quality of segmentation. Therefore, it can be concluded that by using the group-local feature weighting technique, the important features are better recognized, and as a result, the quality of the segmentation is better.

Fig. 13. Comparison of the performance of the proposed approach with other methods using the CIELAB color space. Results obtained by (a) FCM, (b) EWFCM, (c) RLFWHCM, (d) [35], (e) [36], and Ours (CGFFCM).

Table 2
The *Accuracy*, *NMI*, and *F-score* rates obtained by different methods using all three extracted features on each testing image.

Image	FCM	EWFCM	RLFWHCM	[28]	[35]	[36]	Ours (CGFFCM)	Performance improvement
108073	81.71	93.10	82.48	94.24	88.43	83.61	**95.40**	**1.23**
	21.65	48.88	17.06	52.31	28.61	22.10	**60.14**	**14.96**
	54.19	77.20	51.13	79.78	62.65	55.61	**84.05**	**5.35**
113016	83.03	90.16	87.05	90.28	83.10	82.94	**92.74**	**2.72**
	55.51	70.99	63.21	65.87	56.06	55.33	**72.74**	**2.46**
	83.40	90.46	86.53	90.03	83.80	83.35	**92.72**	**2.49**
113044	84.87	94.84	82.03	89.94	84.79	84.79	**96.08**	**1.30**
	63.52	80.22	62.83	69.20	63.31	63.31	**83.27**	**3.80**
	83.84	94.68	79.44	89.24	83.78	83.87	**96.06**	**1.47**
67079	88.61	88.75	98.31	95.91	88.61	88.61	**98.48**	**0.17**
	57.17	57.27	88.28	77.57	57.17	57.17	**89.16**	**0.99**
	90.87	90.99	98.74	96.90	90.86	90.87	**98.87**	**0.13**
134052	57.88	76.12	69.90	80.95	60.89	57.88	**83.09**	**2.64**
	33.28	41.47	38.13	49.80	32.75	33.28	**51.98**	**4.37**
	54.29	71.92	70.09	80.86	55.40	54.29	**82.32**	**1.80**
135069	99.29	99.23	99.50	96.12	99.32	99.32	**99.59**	**0.09**
	82.67	81.21	87.01	31.32	83.30	83.34	**88.62**	**1.85**
	99.63	99.59	99.74	97.98	99.65	99.64	**99.78**	**0.04**
238011	88.62	95.54	90.42	95.63	95.03	**94.91**	**96.06**	**0.44**
	67.69	73.31	68.84	71.49	71.94	**71.53**	**74.15**	**1.14**
	58.53	64.50	62.80	94.14	**94.78**	94.60	92.86	−2.05
101027	88.33	88.30	97.92	96.58	88.75	88.33	**98.32**	**0.40**
	42.50	42.39	84.46	76.16	43.95	42.50	**86.29**	**2.16**
	79.73	79.65	96.59	94.19	79.98	79.73	**97.20**	**0.63**
299091	79.84	80.73	72.37	74.32	79.97	78.97	**81.99**	**1.56**
	67.13	67.38	60.95	60.93	66.95	66.95	**69.83**	**3.63**
	79.01	79.79	57.68	74.11	78.24	78.24	**80.96**	**1.46**
3096	98.97	98.84	99.14	99.12	98.97	98.97	**99.46**	**0.32**
	79.42	77.11	82.08	81.26	79.42	79.42	**87.19**	**6.22**
	90.91	89.90	92.60	92.92	90.91	90.91	**95.60**	**2.88**
80099	99.66	99.68	**99.68**	98.71	99.65	99.62	**99.68**	0
	91.38	91.66	**91.66**	74.23	91.18	90.79	**91.66**	0
	97.27	97.37	**97.37**	88.63	97.17	96.97	**97.37**	0
118035	95.29	95.38	**97.04**	94.65	95.28	95.28	96.52	−0.53
	80.32	80.53	**84.71**	78.86	80.31	80.31	83.21	−1.77
	95.36	95.37	**96.67**	90.97	95.36	95.36	95.89	−0.80
167062	95.55	**97.92**	97.62	97.87	97.54	97.54	97.91	−0.01
	85.71	90.24	89.86	90.15	89.63	89.63	**90.35**	**0.12**
	73.37	79.38	80.24	79.32	79.10	79.10	**80.35**	**0.13**

4.6. Experiment 4: Effect of the group weight coefficient vector

In order to investigate the effect of the group weight coefficient vector ($\boldsymbol{V}$) on the image segmentation results, we run the proposed algorithm for each image 15 times and obtained the average weight for each group ($G(1) = \{HF\}$, $G(2) = \{LabF\}$, and $G(3) = \{TF\}$). The obtained results are compared in Fig. 17. This figure shows that on images 113044, 67079, 135069, 101027, 299091, 3096, and 80099, the weight of group 2 is higher than other groups, and on images 108073, 113016, 134052, 238011, 118035, and 167062, the weight of groups 1 and 3 is higher than group 2. Upon further investigation, we find that on images with more details and colored regions, the weight of group 1 and group 3 is high, and on images with fewer details and colored regions, the weight of group 2 is high. In fact, in highly detailed images with color regions, the HF and TF features are more important and crucial in the image segmentation process. In contrast, in images with low detail and colored regions, the CIELAB feature is much more important. Therefore, it is necessary to consider appropriate weights for each group of extracted features during the clustering process, according to the structure and details of the under-process image. This important aim is achieved in the proposed approach by automatically determining the group weight coefficients vector $\boldsymbol{V}$.

In the ICA optimization process, no specific range is applied to the weights, and the weight of a group can be even zero. However, for each image, the optimal weights obtained for each group are greater than zero. This shows that all groups have a positive effect on the final result. In all images, except 67079, the minimum weight obtained is approximately 0.1.

4.7. Experiment 5: Comparison of CGFFCM with state-of-the-art methods

In this section, the performance of CGFFCM is compared with other successful clustering-based image segmentation methods (i.e., the SMKIFC, A-PSO-IT2IFCM, SSFC-SC, KRVEA, [28], and GWFKHM). The performance of the different methods is shown in Table 3. This table shows the *Accuracy*, *NMI*, and *F-score* rates from top to bottom for each image, respectively. Also, the percentage of improvement obtained by the proposed algorithm on each image compared to the other algorithms' best result is reported in Table 3. The results of other methods, except [28], have been quoted directly from the relevant publications.

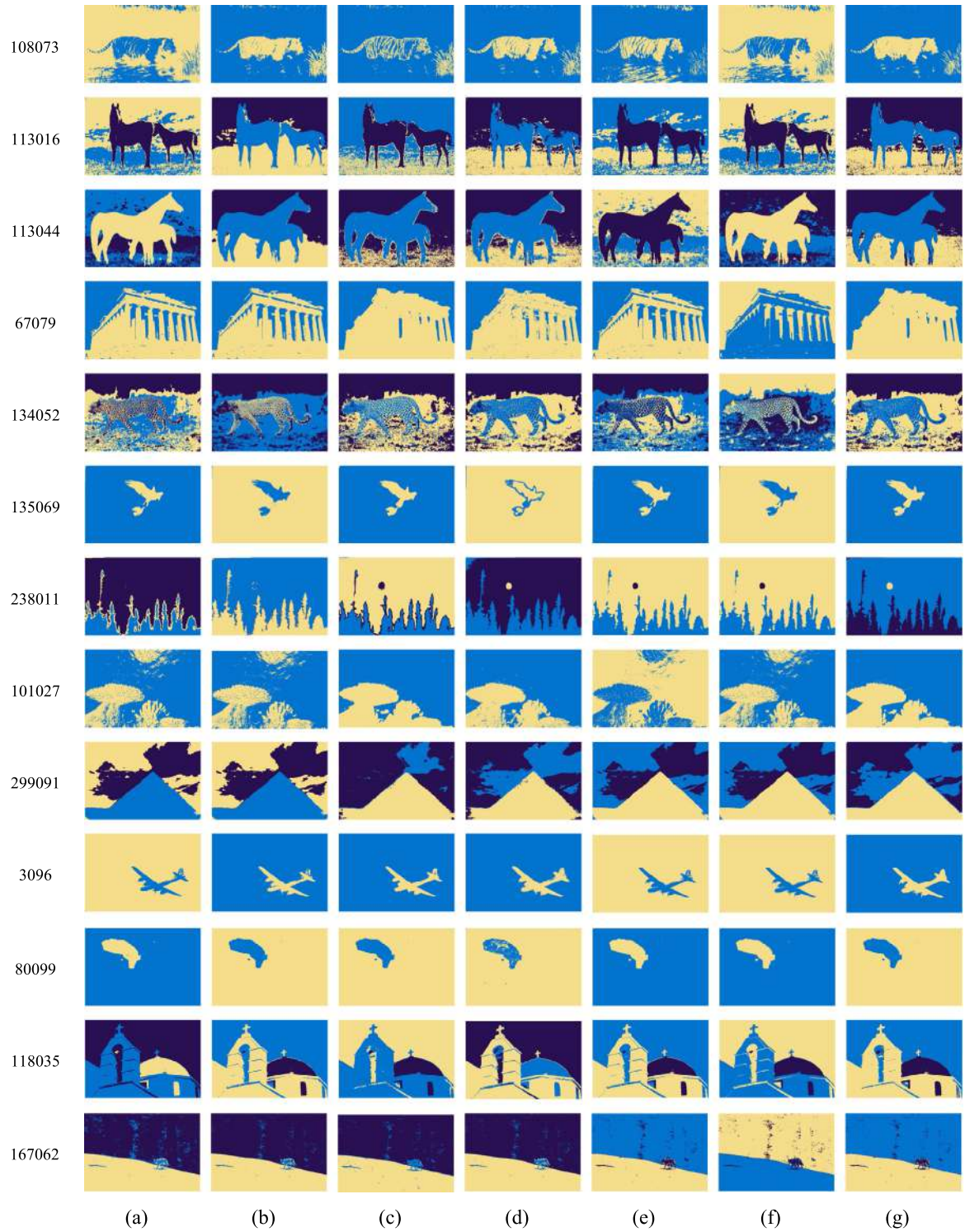

Fig. 14. Comparison of the performance of the proposed approach with other methods using all the extracted features. Results obtained by (a) FCM, (b) EWFCM, (c) RLFWHCM, (d) [28], (e) [35], (f) [36], and (e) Ours (CGFFCM).

As shown in Table 3, CGFFCM has the best performance for all images except 238011 and 167062. In these two images, the method SMKIFC performs better than CGFFCM in terms of *Accuracy* and *NMI*, but CGFFCM has a higher *F-score* than SMKIFC. In other images, CGFFCM provides better results than SMKIFC. Generally, the results show that CGFFCM has a higher performance than other successful methods in this field. After CGFFCM, the [28], SMKIFC, A-PSO-IT2IFCM, and GWFKHM have relatively good performance.

Table 4 shows the average results for CGFFCM and other compared algorithms. As shown in this table, CGFFCM has the best results. In terms of *Accuracy* criteria, after CGFFCM, the methods SMKIFC, A-PSO-IT2IFCM, KRVEA, GWFKHM, and SSFC-SC have better results, respectively. In terms of *NMI* criteria, after

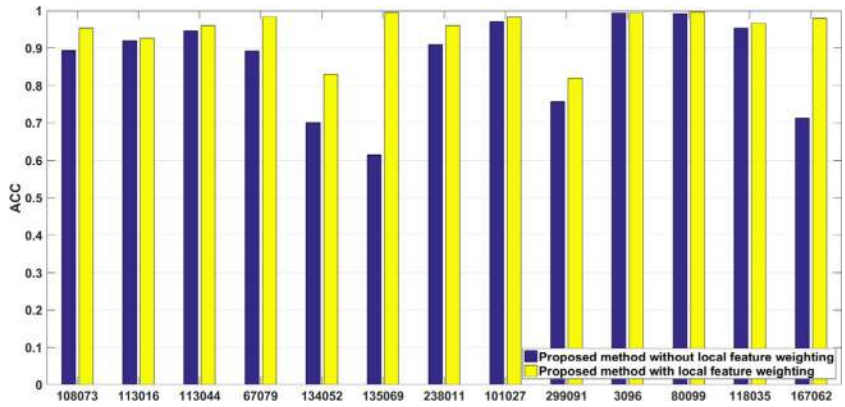

Fig. 15. Effect of the group-local feature weighting on the image segmentation quality in terms of the *Accuracy* rate on different images.

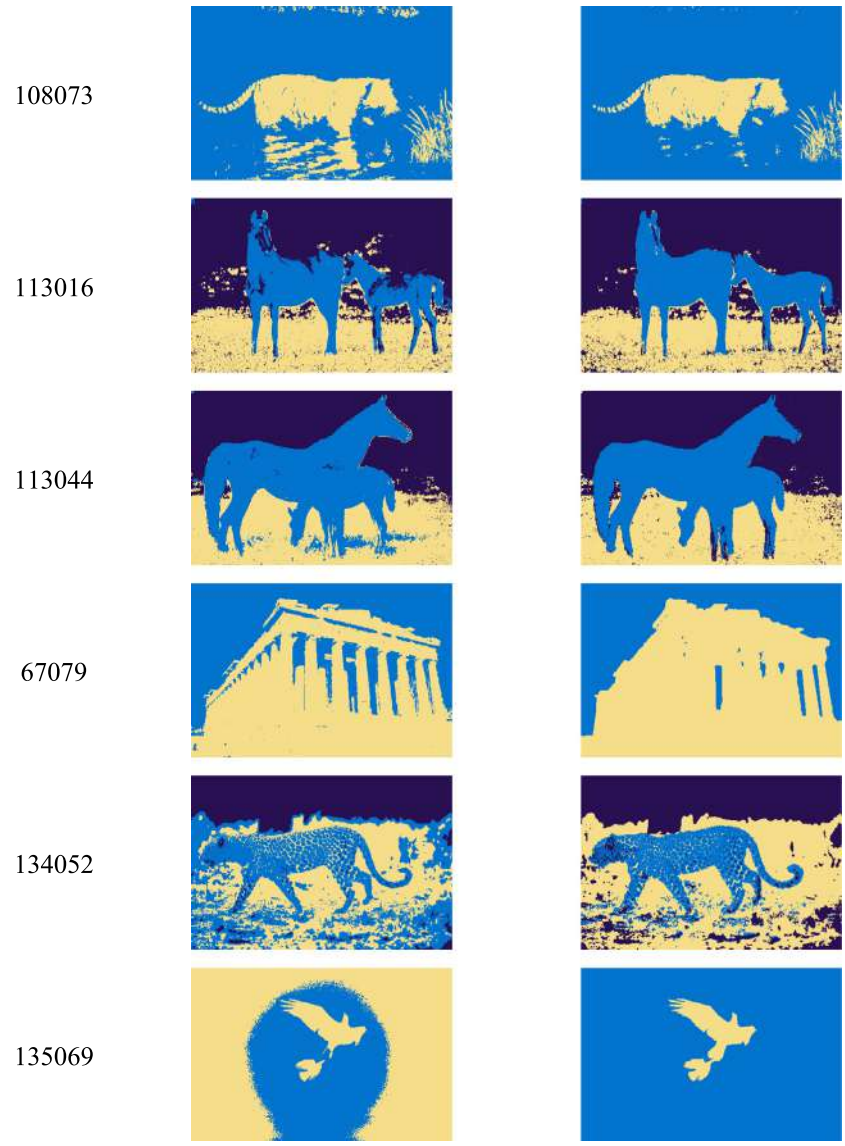

Fig. 16. Effect of the group-local feature weighting on the image segmentation quality. Results obtained by (a) proposed approach without group-local feature weighting, and (b) proposed approach with group-local feature weighting.

CGFFCM, the methods [28], A-PSO-IT2IFCM, SMKIFC, KRVEA, and SSFC-SC, have better results, respectively. Also, In terms of *F-score* criteria, after CGFFCM, methods [28], SMKIFC, KRVEA, and SSFC-SC have better results, respectively. Methods A-PSO-IT2IFCM and SMKIFC have almost high performance in terms of *Accuracy* criterion and have similar accuracy to CGFFCM. However, in terms of *NMI*, and *F-score* criteria are much lower than CGFFCM. This

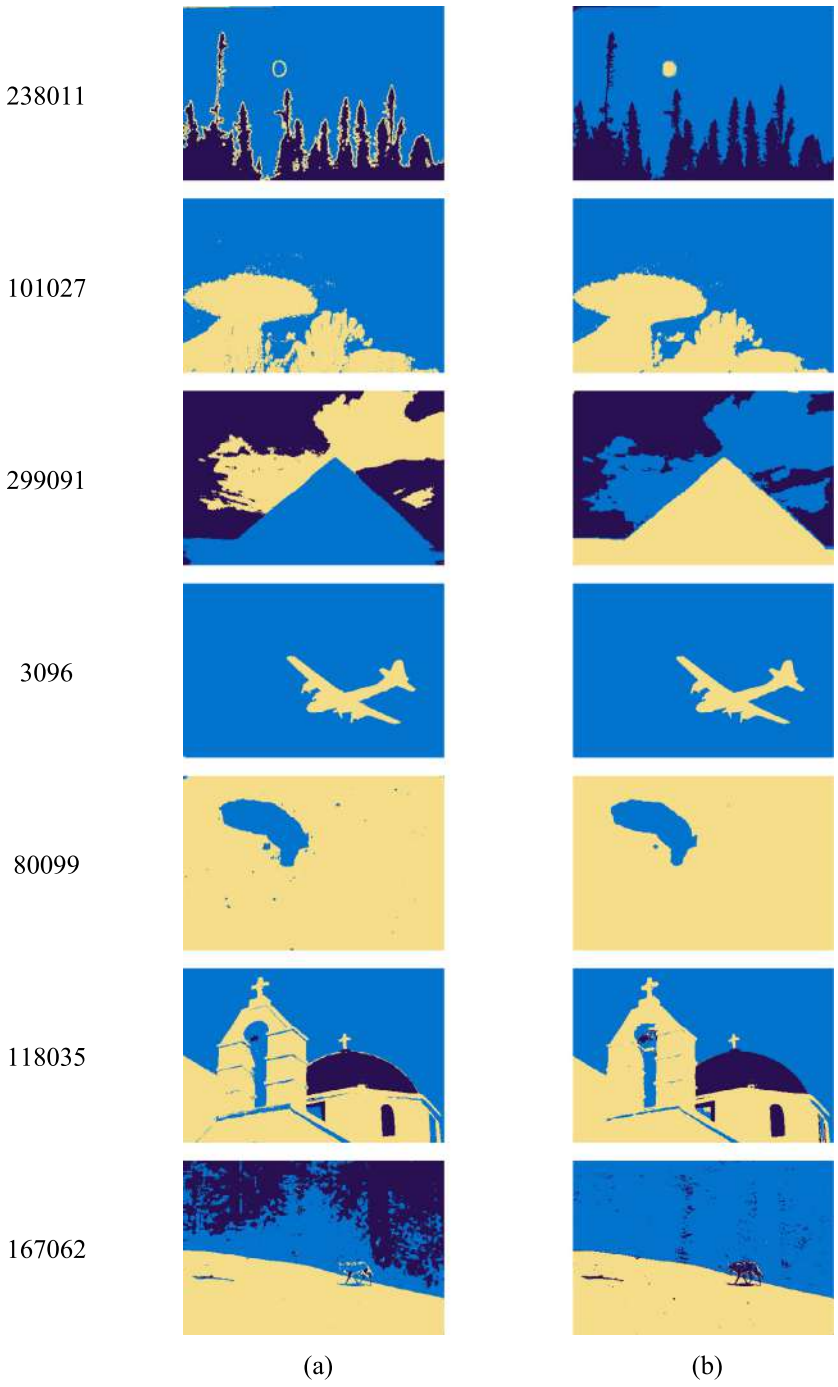

Fig. 16. (*continued*).

shows that these methods are not able to segment those images with imbalanced color regions properly.

4.8. Experiment 6: CGFFCM runtime and comparison with the baseline method

In this section, the runtime of CGFFCM is compared with the baseline algorithm presented in [28]. The running time reported for both methods is only the clustering time. The experiments are conducted on a computer with Intel corei7-4700HQ, CPU 2.40 GHz, and 8 GB RAM in MATLAB2016b environment. The runtime of both algorithms is reported in Table 5. For each image, each method is executed 10 times, and the average running time is reported. As shown in Table 5, the running time of CGFFCM for images 108073, 113044, 67079, 134052, 238011, 299091, 80099, 118035, and 167062 is less than the compared method. CGFFCM is a little slower than the method in [28] for images 113016, 135069, 101027, and 3096, but according to Table 3, CGFFCM has a much better performance than [28] in these images. The average runtime for all images in CGFFCM and the compared method is 24.86 and 25.91, respectively.

4.9. Experiment 7: Analysis of the behavior of the objective function

In this section, we aim to investigate the behavior of the objective function on all tested images. To this end, we illustrate the value of the objective function during the algorithm iterations (Fig. 18) to show how the proposed algorithm alternates between

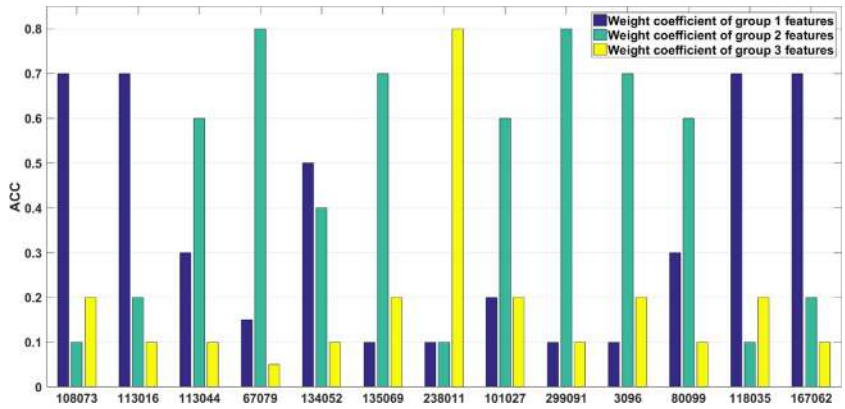

Fig. 17. Effect of the group weight coefficient vector on the image segmentation quality in terms of the *Accuracy* rate on different images.

Table 3
Comparison of the *Accuracy*, *NMI*, and *F-score* rates of the proposed approach with other state-of-the-art methods.

Image	GWFKHM	A-PSO-IT2IFCM	SMKIFC	SSFC-SC	KRVEA	[28]	Ours (CGFFCM)	Performance improvement
108073	60.47	-	-	-	-	94.24	**95.40**	**1.23**
	-	-	-	-	-	52.31	**60.14**	**14.9**
	-	-	-	-	-	79.78	**84.05**	**5.35**
113016	**93.77**	89.35	89.82	87.61	88.47	90.28	92.74	−1.09
	-	42.35	43.97	37.94	43.80	65.87	**72.74**	**10.42**
	-	-	41.62	41.40	41.32	90.03	**92.72**	**2.98**
113044	95.89	-	**96.17**	83.96	68.70	89.94	96.08	−0.09
	-	-	76.01	33.44	5.43	69.20	**83.27**	**9.55**
	-	-	84.54	37.00	34.85	89.24	**96.06**	**7.64**
67079	-	83.08	86.60	82.67	83.72	95.91	**98.48**	**2.67**
	-	45.09	47.73	44.39	45.29	77.57	**89.16**	**14.94**
	-	-	70.71	69.97	71.14	96.90	**98.87**	**2.03**
134052	68.33	-	-	-	-	80.95	**83.09**	**2.64**
	-	-	-	-	-	49.80	**51.98**	**4.37**
	-	-	-	-	-	80.86	**82.32**	**1.80**
135069	99.58	99.18	99.46	59.49	99.27	96.12	**99.59**	**0.01**
	-	81.22	86.05	37.17	82.24	31.32	**88.62**	**2.98**
	-	-	73.76	49.22	77.96	97.98	**99.78**	**1.83**
238011	85.48	95.54	**96.38**	95.40	95.65	95.63	96.06	−0.33
	-	72.83	**74.35**	72.41	73.21	71.49	74.15	−0.26
	-	-	85.73	85.92	85.29	**94.14**	92.86	−1.35
101027	-	88.90	92.66	88.57	86.02	96.58	**98.32**	**1.80**
	-	45.97	60.29	45.03	41.06	76.16	**86.29**	**13.30**
	-	**-**	76.50	69.05	71.30	94.19	**97.20**	**3.19**
299091	**93.68**	-	-	-	-	74.32	81.99	−12.47
	-	-	-	-	-	60.93	**69.83**	**14.60**
	-	-	-	-	-	74.11	**80.96**	**9.24**
3096	98.33	98.83	-	-	-	99.12	**99.46**	**0.34**
	-	74.81	-	-	-	81.26	**87.19**	**7.29**
	-	-	-	-	-	92.92	**95.60**	**2.88**
80099	95.59	-	93.40	86.33	92.92	98.71	**99.68**	**0.98**
	-	-	21.19	6.57	0.02	74.23	**91.66**	**23.48**
	-	-	36.74	35.43	33.80	88.63	**97.37**	**9.86**
118035	-	93.51	94.11	93.38	93.85	94.65	**96.52**	**1.97**
	-	76.74	77.50	69.78	76.02	78.86	**83.21**	**5.51**
	-	-	71.81	75.58	70.16	90.97	**95.89**	**5.40**
167062	-	98.56	**98.08**	97.95	95.45	97.87	97.91	−0.65
	-	90.72	**92.14**	90.05	84.86	90.15	90.35	−1.94
	-	-	59.85	58.15	57.08	79.32	**80.35**	**1.29**

Table 4
The average performance of CGFFCM and other state-of-the-art methods.

Metrics	GWFKHM	A-PSO-IT2IFCM	SMKIFC	SSFC-SC	KRVEA	[28]	Ours (CGFFCM)
Accuracy	87.90	93.36	94.07	86.11	89.33	92.64	**95.03**
NMI	-	66.21	64.35	48.53	50.21	67.26	**79.12**
F-score	-	-	66.80	57.96	60.32	88.39	**91.84**

the $\boldsymbol{U}$, $\boldsymbol{C}$, $\boldsymbol{W}$, and $\boldsymbol{Z}$ optimization steps to get a local optimum of $\boldsymbol{F}$.

As shown in Fig. 18, the value obtained for the objective function is reduced in the initial iterations markedly. Finding the suitable cluster centers in the initial iterations is the reason for such behavior. For all tested images except 3096 and 80099, the value of the objective function has been significantly reduced from first to 20th iterations, and there are, however, slight decreases from iterations 20th to the end. This shows that the proposed algorithm achieves a nearly optimal solution in the initial iterations. Also, the proposed method is optimized for images 101027, 118035, 167062, 108073, 113016, 113044, 135069, and 134052 in the 52nd iteration, for images 238011, 3096, and 67079 in the 53rd iteration, and for images 299091 and 80099 in the 58th and 59th iteration, respectively. By performing the group-local feature weighting along with the cluster weighting in a concurrent manner, the algorithm achieves an optimal solution very fast, irrespective of the initialization.

5. Conclusion

In this study, we presented a new method, termed CGFFCM (Cluster-weight and Group-local Feature-weight learning in Fuzzy C-Means), for color image segmentation based on the FCM clustering algorithm. During the color image segmentation process, we used a group-local feature weighting scheme and a cluster weighting strategy to have better results. All weights, including the group-local weights of features and the clusters' weights, were calculated automatically and simultaneously during the learning process. We also used an efficient combination of image features, including the local homogeneity, CIELAB, and texture, to improve image segmentation. The group-local feature weighting scheme caused better application of the features, and on the other hand, the weighting of clusters reduced the algorithm's sensitivity to the initialization issue. So better segmentation results were obtained for various images. A combination of the ICA meta-heuristic algorithm with the proposed clustering algorithm was used to find the optimal group weight of features.

The performance of CGFFCM was examined and confirmed by extensive experiments. The results of the experiments confirmed the positive effects of the group-local feature weighting scheme and group weight coefficient vector in the proposed approach. The Berkeley benchmark image dataset, including different colored regions and various imbalance and balance regions, was used to examine the efficiency of the proposed solutions. The performance metrics of *Accuracy*, *NMI*, *F-score* were used to compare the results with state-of-the-art clustering-based image segmentation methods. The comparisons showed that CGFFCM has significantly better performance, especially when dealing with complex imbalanced colored region images. Also, the investigations on the effectiveness of solutions used in the proposed approach confirmed the positive effects of all solutions. In addition, the proposed method performed well in terms of running time.

CGFFCM has two limitations that can be mitigated in future research. The first limitation is that it requires the number of clusters (colored regions), similar to most of the existing clustering methods. The number of clusters must be known in advance. However, in real-world applications, the number of clusters is often unknown. Thus, we are interested in the automatically determining of the number of cluster regions during the clustering process. The second limitation is that in segmenting complex images such as medical and SAR (Synthetic-Aperture Radar) images, we need to extract other types of features. The three features used in this study are often not suitable for clustering medical and SAR images. Thus, as a future direction, it is of interest to further develop the CGFFCM to segment the medical and SAR images.

Table 5
Runtime comparison of CGFFCM with baseline method in [28].

Image	[28]	Ours (CGFFCM)
108073	18.93	**18.01**
113016	**28.11**	29.18
113044	33.18	**30.10**
67079	20.56	**18.36**
134052	32.21	**29.57**
135069	**17.25**	18.28
238011	31.23	**30.93**
101027	**17.99**	18.40
299091	34.01	**32.18**
3096	**18.73**	18.24
80099	21.78	**20.19**
118035	31.12	**29.42**
167062	31.75	**30.35**

CRediT authorship contribution statement

Amin Golzari Oskouei: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mahdi Hashemzadeh:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Visualization, Writing – review & editing. **Bahareh Asheghi:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mohammad Ali Balafar:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.

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.

Appendix

Here, we present how the updating equation for the membership function u_{nk} is obtained from the Lagrange $\tilde{F}$ equation (Eq. (14)). The steps are as follows:

$$\frac{\partial \tilde{\mathrm{F}}}{\partial u_{nk}} = 0$$
$$\alpha {u_{nk}}^{\alpha-1} z_k^p d_{nk}^{(gw)} - \delta = 0$$
$$u_{nk} = \left[\frac{\delta}{\alpha z_k^p d_{nk}^{(gw)}}\right]^{\frac{1}{\alpha-1}}. \tag{A.1}$$

Now, we just have to get the value of δ. To this end, we proceed as follows:

$$\frac{\partial \tilde{\mathrm{J}}}{\partial u_{nl}} = 0 \Rightarrow \delta = \alpha {u_{nl}}^{\alpha-1} z_l^p d_{nl}^{(gw)}, \tag{A.2}$$

We conclude from Eqs. (A.1) and (A.2):

$$u_{nk} = \left[\frac{\alpha {u_{nl}}^{\alpha-1} z_l^p d_{nl}^{(gw)}}{\alpha z_k^p d_{nk}^{(gw)}}\right]^{\frac{1}{\alpha-1}}$$
$$u_{nk} = u_{nl}\left[\frac{z_l^p d_{nl}^{(gw)}}{z_k^p d_{nk}^{(gw)}}\right]^{\frac{1}{\alpha-1}}$$
$$u_{nl} = u_{nk}\left[\frac{z_k^p d_{nk}^{(gw)}}{z_l^p d_{nl}^{(gw)}}\right]^{\frac{1}{\alpha-1}} \Rightarrow \sum_{l=1}^{K} u_{nl} = \sum_{l=1}^{K} u_{nk}\left[\frac{z_k^p d_{nk}^{(gw)}}{z_l^p d_{nl}^{(gw)}}\right]^{\frac{1}{\alpha-1}}. \tag{A.3}$$

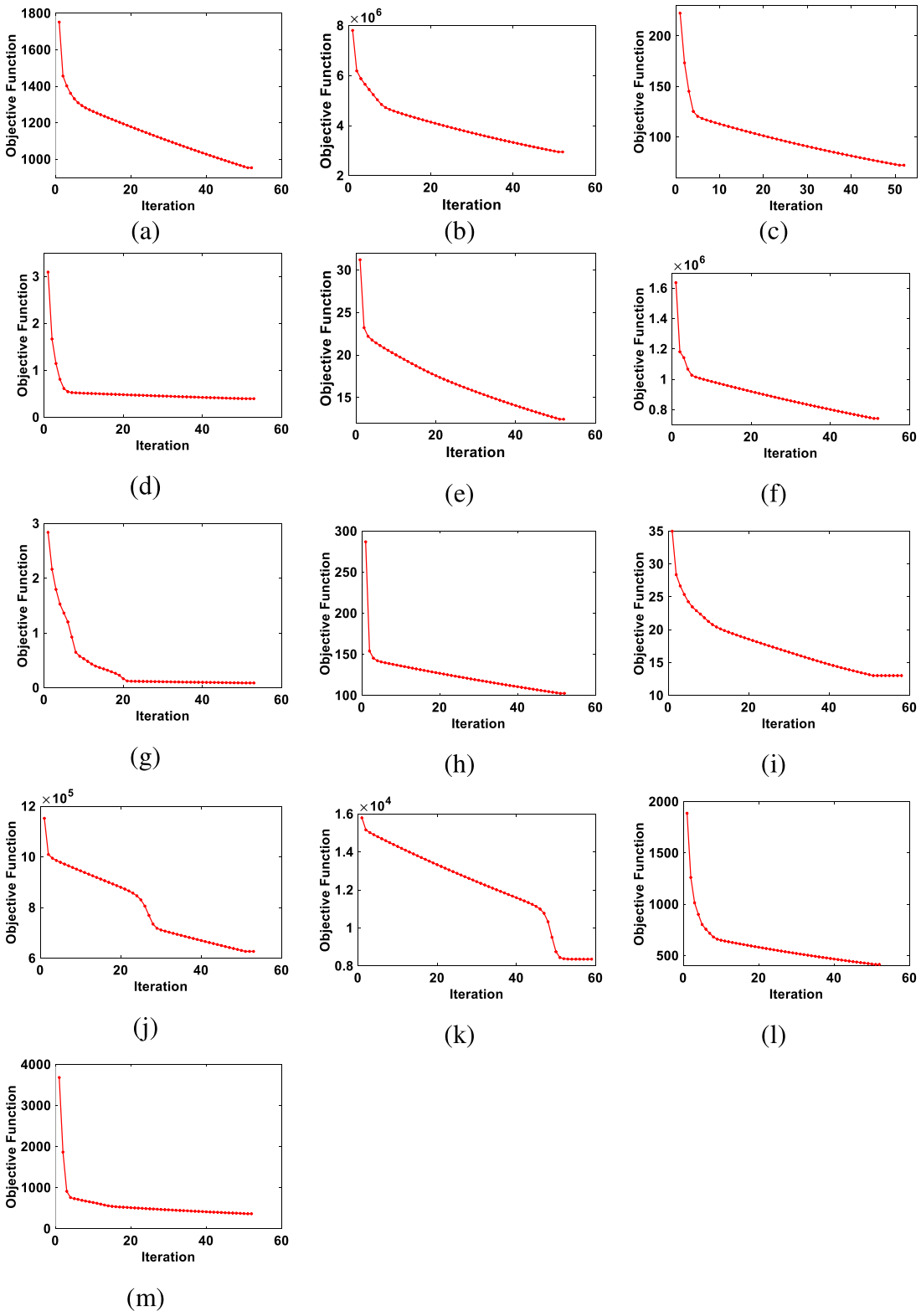

Fig. 18. The value of the objective function along the algorithm iterations on each testing image (a) 108073, (b) 113016, (c) 113044, (d) 67079, (e) 134052, (f) 135069, (g) 238011, (h) 101027, (i) 29909, (j) 3096, (k) 8099, (l) 118035, (m) 167062.

Given that $\sum_{l=1}^{K} u_{nl} = 1$, we can write the Eq. (A.3) as follows (see Eq. (A.4)):

$$1 = u_{nk} \sum_{l=1}^{K} \left[\frac{z_k^p d_{nk}^{(gw)}}{z_l^p d_{nl}^{(gw)}} \right]^{\frac{1}{\alpha-1}} \Rightarrow u_{nk} = \frac{1}{\sum_{l=1}^{K} \left[\frac{z_k^p d_{nk}^{(gw)}}{z_l^p d_{nl}^{(gw)}} \right]^{\frac{1}{\alpha-1}}}. \quad \text{(A.4)}$$

Thus, the membership function (Eq. (15)) is obtained.

Now, we describe the steps to obtain the c_{km}:

$$\frac{\partial \tilde{F}}{\partial c_{km}} = 0 \Rightarrow \sum_{n=1}^{N} \sum_{g=1}^{G} 2 u_{nk}{}^{\alpha} z_k^p v_g w_{km}^q (-\gamma_m)(x_{nm} - c_{km})$$

$$\times \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right) = 0. \tag{A.5}$$

We can express the Eq. (A.5) as follows:

$$\sum_{n=1}^{N}\sum_{g=1}^{G} 2u_{nk}{}^{\alpha} z_k^p v_g w_{km}^q (-\gamma_m) x_{nm} \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right)$$
$$- \sum_{n=1}^{N}\sum_{g=1}^{G} 2u_{nk}{}^{\alpha} z_k^p v_g w_{km}^q (-\gamma_m) c_{km} \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right)$$
$$= 0$$

$$2w_{km}^q z_k^p (-\gamma_m) \sum_{n=1}^{N}\sum_{g=1}^{G} u_{nk}{}^{\alpha} v_g x_{nm} \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right)$$
$$- 2w_{km}^q z_k^p (-\gamma_m) (c_{km}) \sum_{n=1}^{N}\sum_{g=1}^{G} u_{nk}{}^{\alpha} v_g \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right)$$
$$= 0$$

$$c_{km} = \frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right) x_{nm} \sum_{g=1}^{G} v_g}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} \left(\exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)\right) \sum_{g=1}^{G} v_g}$$
$$= \frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} \exp\left(-\gamma_m (x_{nm} - c_{km})^2\right) x_{nm}}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} \exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)}.$$

Assuming $\sum_{g=1}^{G} v_g = 1$, the *cluster center updating function* (Eq. (16)) is obtained.

Concerning w_{km}, if $Dw_{km} = 0$ and h_m is the number of features such as s for which Dw_{ks} is zero, then w_{km} is defined as $\frac{1}{h_m}$. By contrast, if $Dw_{km} \neq 0$, and if there is a feature like s for which Dw_{ks} is zero, the value of w_{km} will be zero. If there is no feature such as s for which Dw_{ks} is zero, then we can calculate the value of w_{km} as follows:

$$\frac{\partial \tilde{F}}{\partial w_{km}} = 0 \Rightarrow \sum_{n=1}^{N}\sum_{g=1}^{G} u_{nk}{}^{\alpha} z_k^p v_g q w_{km}^{q-1} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) - \psi = 0$$

$$w_{km} = \left[\frac{\psi}{q w_{km}^{q-1} z_k^p \sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}. \tag{A.6}$$

Now, we just have to get the value of ψ. To this end, we proceed as follows:

$$\frac{\partial \tilde{F}}{\partial w_{ks}} = 0$$

$$\psi = \sum_{n=1}^{N}\sum_{g=1}^{G} u_{nk}{}^{\alpha} z_k^p v_g q w_{ks}^{q-1} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right))$$

$$\psi = q w_{ks}^{q-1} z_k^p \sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g. \tag{A.7}$$

We conclude from Eqs. (A.6) and (A.7):

$$w_{km} = \left[\frac{q w_{ks}^{q-1} z_k^p \sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g}{q w_{km}^{q-1} z_k^p \sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}$$

$$w_{km} = w_{ks} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}$$

$$w_{ks} = w_{km} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}$$

$$\sum_{s\in\{HF,LabF,TF\}} w_{ks}$$
$$= \sum_{s\in\{HF,LabF,TF\}} w_{km} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}. \tag{A.8}$$

Given that $\sum_{s\in\{HF,LabF,TF\}} w_{ks} = 1$, we can express the Eq. (A.8) as follows:

$$1 = \sum_{s\in\{HF,LabF,TF\}} w_{km} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}$$

$$w_{km} = \frac{1}{\sum_{s\in\{HF,LabF,TF\}} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right)) \sum_{g=1}^{G} v_g}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right)) \sum_{g=1}^{G} v_g}\right]^{\frac{1}{q-1}}}$$
$$= \frac{1}{\sum_{s\in\{HF,LabF,TF\}} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_m (x_{nm} - c_{km})^2\right))}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} (1 - exp\left(-\gamma_s (x_{ns} - c_{ks})^2\right))}\right]^{\frac{1}{q-1}}}.$$

Assuming $\sum_{g=1}^{G} v_g = 1$, the feature weight *updating function* (Eq. (17)) is obtained.

Concerning z_k, if $Dz_k = 0$ and g_k is the number of clusters such as l for which Dz_l is zero, then z_k is defined as $\frac{1}{g_k}$. By contrast, if $Dz_k \neq 0$, and if there is a cluster like l for which Dz_l is zero, the value of z_k will be zero. If there is not a cluster such as l for which Dz_l is zero, then we can calculate the value of z_k as follows:

$$\frac{\partial \tilde{F}}{\partial z_k} = 0 \Rightarrow \sum_{n=1}^{N} u_{nk}{}^{\alpha} p z_k^{p-1} d_{nk}^{(gw)} - \omega = 0$$

$$z_k = \left[\frac{\omega}{p \sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}\right]^{\frac{1}{p-1}}. \tag{A.9}$$

Now, we just have to get the value of ω. To do this, we proceed as follows:

$$\frac{\partial \tilde{F}}{\partial z_l} = 0$$

$$\omega = \sum_{n=1}^{N} u_{nl}{}^{\alpha} p z_l^{p-1} d_{nl}^{(gw)}$$

$$\omega = p z_l^{p-1} \sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}. \tag{A.10}$$

We conclude from Eqs. (A.9) and (A.10):

$$z_k = \left[\frac{z_l^{p-1} \sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}\right]^{\frac{1}{p-1}}$$

$$z_k = z_l \left[\frac{\sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}}{\sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}\right]^{\frac{1}{p-1}}$$

$$z_l = z_k \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}{\sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}}\right]^{\frac{1}{p-1}} \Rightarrow \sum_{l=1}^{K} z_l = \sum_{l=1}^{K} z_k \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}{\sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}}\right]^{\frac{1}{p-1}}. \tag{A.11}$$

Given that $\sum_{l=1}^{K} z_l = 1$, we can express the Eq. (A.11) as follows:

$$1 = z_k \sum_{l=1}^{K} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}{\sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}}\right]^{\frac{1}{p-1}}$$

$$z_k = \frac{1}{\sum_{l=1}^{K} \left[\frac{\sum_{n=1}^{N} u_{nk}{}^{\alpha} d_{nk}^{(gw)}}{\sum_{n=1}^{N} u_{nl}{}^{\alpha} d_{nl}^{(gw)}}\right]^{\frac{1}{p-1}}}.$$

Thus, the cluster weight function (Eq. (18)) is obtained.

References

[1] F. Yang, H. Fan, P. Chu, E. Blasch, H. Ling, Clustered object detection in aerial images, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8311–8320.

[2] M. Hashemzadeh, G. Pan, Y. Wang, M. Yao, J. Wu, Combining velocity and location-specific spatial clues in trajectories for counting crowded moving objects, Int. J. Pattern Recognit. Artif. Intell. 27 (2013) 1354003.

[3] M. Hashemzadeh, G. Pan, M. Yao, Counting moving people in crowds using motion statistics of feature-points, Multimedia Tools Appl. 72 (2014) 453–487.

[4] M. Hashemzadeh, N. Farajzadeh, Combining keypoint-based and segment-based features for counting people in crowded scenes, Inform. Sci. 345 (2016) 199–216.

[5] N. Farajzadeh, M. Hashemzadeh, Exemplar-based facial expression recognition, Inform. Sci. 460–461 (2018) 318–330.

[6] R.P. Sharma, S. Dey, Two-stage quality adaptive fingerprint image enhancement using fuzzy C-means clustering based fingerprint quality analysis, Image Vis. Comput. 83–84 (2019) 1–16.

[7] M. Hashemzadeh, A. Zademehdi, Fire detection for video surveillance applications using ICA K-medoids-based color model and efficient spatio-temporal visual features, Expert Syst. Appl. 130 (2019) 60–78.

[8] M. Hashemzadeh, B. Adlpour Azar, Retinal blood vessel extraction employing effective image features and combination of supervised and unsupervised machine learning methods, Artif. Intell. Med. 95 (2019) 1–15.

[9] M. Hashemzadeh, N. Farajzadeh, A machine vision system for detecting fertile eggs in the incubation industry, Int. J. Comput. Intell. Syst. 9 (2016) 850–862.

[10] N. Farajzadeh, A. Karamiani, M. Hashemzadeh, A fast and accurate moving object tracker in active camera model, Multimedia Tools Appl. 77 (2018) 6775–6797.

[11] M. Hashemzadeh, Hiding information in videos using motion clues of feature points, Comput. Electr. Eng. 68 (2018) 14–25.

[12] M. Hashemzadeh, B. Asheghi, N. Farajzadeh, Content-aware image resizing: An improved and shadow-preserving seam carving method, Signal Process. 155 (2019) 233–246.

[13] L. He, S. Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation, Neurocomputing 240 (2017) 152–174.

[14] S. Pare, A. Kumar, V. Bajaj, G.K. Singh, A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve, Appl. Soft Comput. 47 (2016) 76–102.

[15] K.S. Tan, N.A. Mat Isa, W.H. Lim, Color image segmentation using adaptive unsupervised clustering approach, Appl. Soft Comput. 13 (2013) 2017–2036.

[16] H.-J. Xing, M.-H. Ha, Further improvements in feature-weighted fuzzy C-means, Inform. Sci. 267 (2014) 1–15.

[17] Z. Zhou, X. Zhao, S. Zhu, K-harmonic means clustering algorithm using feature weighting for color image segmentation, Multimedia Tools Appl. 77 (2018) 15139–15160.

[18] T.R. Farshi, J.H. Drake, E. Özcan, A multimodal particle swarm optimization-based approach for image segmentation, Expert Syst. Appl. 149 (2020) 113233.

[19] L. Feng, H. Li, Y. Gao, Y. Zhang, A color image segmentation method based on region salient color and fuzzy C-means algorithm, Circuits Systems Signal Process. 39 (2020) 586–610.

[20] F. Zhao, Z. Zeng, H. Liu, R. Lan, J. Fan, Semisupervised approach to surrogate-assisted multiobjective kernel intuitionistic fuzzy clustering algorithm for color image segmentation, IEEE Trans. Fuzzy Syst. 28 (2020) 1023–1034.

[21] L.H. Son, T.M. Tuan, Dental segmentation from X-ray images using semi-supervised fuzzy clustering with spatial constraints, Eng. Appl. Artif. Intell. 59 (2017) 186–195.

[22] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Springer Science & Business Media, 2013.

[23] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA., 1967, pp. 281–297.

[24] X. Zhang, M. Jian, Y. Sun, H. Wang, C. Zhang, Improving image segmentation based on patch-weighted distance and fuzzy clustering, Multimedia Tools Appl. 79 (2020) 633–657.

[25] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.

[26] S.K. Choy, K. Yuen, C. Yu, Fuzzy bit-plane-dependence image segmentation, Signal Process. 154 (2019) 30–44.

[27] B.A. Pimentel, R.M.C.R de Souza, Multivariate fuzzy C-means algorithms with weighting, Neurocomputing 174 (2016) 946–965.

[28] M. Hashemzadeh, A. Golzari Oskouei, N. Farajzadeh, New fuzzy C-means clustering method based on feature-weight and cluster-weight learning, Appl. Soft Comput. 78 (2019) 324–345.

[29] A. Stetco, X.-J. Zeng, J. Keane, Fuzzy C-means++: Fuzzy C-means with effective seeding initialization, Expert Syst. Appl. 42 (2015) 7541–7548.

[30] G. Tzortzis, A. Likas, The MinMax k-means clustering algorithm, Pattern Recognit. 47 (2014) 2505–2516.

[31] S.K. Choy, T.C. Ng, C. Yu, Unsupervised fuzzy model-based image segmentation, Signal Process. 171 (2020) 107483.

[32] H. Abdellahoum, N. Mokhtari, A. Brahimi, A. Boukra, CSFCM: An improved fuzzy C-means image segmentation algorithm using a cooperative approach, Expert Syst. Appl. 166 (2021) 114063.

[33] F. Zhao, Y. Chen, H. Liu, J. Fan, Alternate PSO-based adaptive interval type-2 intuitionistic fuzzy C-means clustering algorithm for color image segmentation, IEEE Access 7 (2019) 64028–64039.

[34] M. Babrdelbonb, S.Z.M.H.M. Hashim, N.E.N. Bazin, Data analysis by combining the modified k-means and imperialist competitive algorithm, J. Teknol. 70 (2014).

[35] H. Emami, F. Derakhshan, Integrating fuzzy K-means, particle swarm optimization, and imperialist competitive algorithm for data clustering, Arab. J. Sci. Eng. 40 (2015) 3545–3554.

[36] R. Mikaeil, S.S. Haghshenas, S.S. Haghshenas, M. Ataei, Performance prediction of circular saw machine using imperialist competitive algorithm and fuzzy clustering technique, Neural Comput. Appl. 29 (2018) 283–292.

[37] Z. Aliniya, S.A. Mirroshandel, A novel combinatorial merge-split approach for automatic clustering using imperialist competitive algorithm, Expert Syst. Appl. 117 (2019) 243–266.

[38] T. Niknam, E.T. Fard, S. Ehrampoosh, A. Rousta, A new hybrid imperialist competitive algorithm on data clustering, Sadhana 36 (2011) 293.

[39] F. Zhao, J. Fan, H. Liu, R. Lan, C.W. Chen, Noise robust multiobjective evolutionary clustering image segmentation motivated by the intuitionistic fuzzy information, IEEE Trans. Fuzzy Syst. 27 (2018) 387–401.

[40] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization, IEEE Trans. Evol. Comput. 22 (2018) 129–142.

[41] I. Niño Adan, D. Manjarres, I. Landa-Torres, E. Portillo, Feature weighting methods: A review, Expert Syst. Appl. 184 (2021) 115424.

[42] J. Zhou, L. Chen, C.L.P. Chen, Y. Zhang, H.-X. Li, Fuzzy clustering with the entropy of attribute weights, Neurocomputing 198 (2016) 125–134.

[43] X.-b. Zhi, J.-l. Fan, F. Zhao, Robust local feature weighting hard c-means clustering algorithm, Neurocomputing 134 (2014) 20–29.

[44] Z. Zhou, S. Zhu, Kernel-based multiobjective clustering algorithm with automatic attribute weighting, Soft Comput. 22 (2018) 3685–3709.

[45] M.R.P. Ferreira, F.d.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems 237 (2014) 1–46.

[46] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 4661–4667.

[47] http://archive.ics.uci.edu/ml/index.php.

[48] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 898–916.

[49] G. Wyszecki, W.S. Stiles, Color Science, Wiley, New York, 1982.

[50] F. Gamino-Sánchez, I.V. Hernández-Gutiérrez, A.J. Rosales-Silva, F.J. Gallegos-Funes, D. Mújica-Vargas, E. Ramos-Díaz, B.E. Carvajal-Gámez, J.M.V. Kinani, Block-matching fuzzy C-means clustering algorithm for segmentation of color images degraded with Gaussian noise, Eng. Appl. Artif. Intell. 73 (2018) 31–49.

[51] D. Reska, M. Kretowski, GPU-accelerated image segmentation based on level sets and multiple texture features, Multimedia Tools Appl. (2020).

[52] K. Wang, L. Li, J. Zhang, End-to-end trainable network for superpixel and image segmentation, Pattern Recognit. Lett. 140 (2020) 135–142.

[53] B. Yuan, L. Han, H. Yan, Explore double-opponency and skin color for saliency detection, Neurocomputing (2020).

[54] D.-W. Kim, K.H. Lee, D. Lee, A novel initialization scheme for the fuzzy c-means algorithm for color clustering, Pattern Recognit. Lett. 25 (2004) 227–237.

[55] K. Sakthivel, R. Nallusamy, C. Kavitha, Color image segmentation using SVM pixel classification image, world academy of science, engineering and technology, Int. J. Comput. Electr. Autom. Control Inf. Eng. 8 (2015) 1919–1925.

[56] A. Humeau-Heurtier, Texture feature extraction methods: A survey, IEEE Access 7 (2019) 8975–9000.

[57] M. Gao, H. Chen, S. Zheng, B. Fang, Feature fusion and non-negative matrix factorization based active contours for texture segmentation, Signal Process. 159 (2019) 104–118.

[58] L. Li, Q. An, An in-depth study of tool wear monitoring technique based on image segmentation and texture analysis, Measurement 79 (2016) 44–52.

[59] C. Sompong, S. Wongthanavasu, An efficient brain tumor segmentation based on cellular automata and improved tumor-cut algorithm, Expert Syst. Appl. 72 (2017) 231–244.

[60] Z. Xing, H. Jia, Multilevel color image segmentation based on GLCM and improved salp swarm algorithm, IEEE Access 7 (2019) 37672–37690.

[61] T. Niknam, E. Taherian Fard, N. Pourjafarian, A. Rousta, An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering, Eng. Appl. Artif. Intell. 24 (2011) 306–317.
[62] J.C. Bezdek, Objective function clustering, in: Pattern Recognition with Fuzzy Objective Function Algorithms, Springer, 1981, pp. 43–93.
[63] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (1980) 1–8.
[64] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.