IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 19, 2021, accepted March 10, 2021, date of publication March 31, 2021, date of current version June 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070044

Kernel-Based Intuitionistic Fuzzy Clustering Image Segmentation Based on Grey Wolf Optimizer With Differential Mutation

XIANGXIAO LEI[1,2] AND HONGLIN OUYANG[2]
[1]School of Electronic Information Engineering, Changsha Social Work College, Changsha 410004, China
[2]College of Electrical and Information Engineering, Hunan University, Changsha 410002, China

Corresponding author: Xiangxiao Lei (xianglei1248@qq.com)

ABSTRACT Conventional fuzzy clustering algorithms present several disadvantages with respect to image segmentation, including a tendency to arrive at local optima and a relatively high sensitivity to noise and initial cluster centers. To address these issues, we herein propose a kernel-based intuitionistic fuzzy clustering approach combining an improved grey wolf optimizer with a kernel-based intuitionistic fuzzy C-means clustering (IGWO-KIFCM) algorithm capable of carrying out differential mutations for image segmentation. The proposed method extracts spatial information from images and then applies a kernel-based intuitionistic fuzzy clustering objective function to improve the robustness of the algorithm against noise. To cope with the initial sensitivity and local optima issues, we develop an improved grey wolf optimizer based on differential mutation for the global optimization of the cluster centers. A comparative optimization assessment using six classic functions reveals that the improved grey wolf optimizer algorithm outperforms both the grey wolf optimizer and mean grey wolf optimizer algorithms in terms of searching ability and does not easily run into local optima. Moreover, the IGWO-KIFCM algorithm surpasses several other algorithms with respect to clustering performance across multiple datasets, and achieves good results in segmenting images with various types of noises.

INDEX TERMS Differential mutation, grey wolf optimization, image segmentation, intuitionistic fuzzy clustering, spatial information.

I. INTRODUCTION

The fuzzy C-means (FCM) clustering algorithm is an unsupervised clustering analyzer based on the use of fuzzy sets. Because it can retain a high amount of information from an original image, it has been widely applied in areas such as fruit image recognition [1] and medical image segmentation [2]. Whereas conventional fuzzy sets consider only differences in set element membership, intuitionistic fuzzy sets (IFSs) apply hesitation and non-membership, thereby improving the ability of fuzzy sets in describing and processing complex and uncertain knowledge to reveal more accurately the fuzzy essence of objects. IFSs and the FCM algorithm can be combined to obtain an intuitionistic FCM clustering algorithm (IFCM) [3] that can process fuzzy information more effectively than conventional FCM algorithms. When the IFCM algorithm is applied to image segmentation, the spatial information of the image is not considered, given that IFCM algorithms are sensitive to noise and wild values [4]. The kernel-based IFCM clustering algorithm (KIFCM) has been proposed as a tool for overcoming the sensitivity of the IFCM to image noise [5]. In [6], an adaptive FCM algorithm based on local noise detecting and image spatial information is shown to improve the noise sensitivity of the traditional fuzzy clustering algorithm during image segmentation. However, the KIFCM algorithm is vulnerable to noise when applied to image segmentation due to insufficient consideration of the spatial information of the image, such that it cannot segment images with complex textures or backgrounds. Similar to the traditional FCM algorithm, the KIFCM algorithm is sensitive to the initial value of the clustering center and prone to falling into local optima rather than global optima.

In recent years, the swarm intelligence optimization algorithm has been widely applied to the clustering problem. This algorithm can effectively overcome the drawbacks of

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

the traditional clustering algorithm, which is sensitive to the initial value of the clustering center and easily falls into the local optimum. The grey wolf optimizer (GWO) has a strong ability to seek local optima and a fast convergence, but is rather weak in terms of searching for global optima when solving certain complex problems. To overcome these issues, many improved GWO (IGWO) algorithms have been proposed. For instance, optimization can be strengthened by incorporating crossover and mutation operators into GWO [7], whereas Levy flight can be adapted to GWO to cope with population diversity [8]. Despite significant improvements on GWO, it remains prone to falling into local optima, a slow recovery rate, and imbalance between global exploration and local development capabilities.

Therefore, to further improve the robustness of the KIFCM algorithm to image noise, we designed a strategy to extract robust spatial information from images with the goal of establishing a kernel-based IFC objective function that incorporates robust spatial information from images. To address the disadvantages of the KIFCM algorithm in terms of sensitivity to initial cluster centers and the tendency to run into local optima, we drew inspiration from [9] in developing a strategy to update wolf pack locations based on dynamic random differential mutation for the optimization of cluster centers. By applying these methods, the proposed algorithm is expected to expand the global optimization range, improve local optimization precision, increase the probability of the algorithm jumping out of the local extreme value, and carry out more accurate image segmentation.

II. RELATED ALGORITHMS

A. GWO ALGORITHM

The GWO algorithm simulates the stringent hierarchical system of a grey wolf pack and its collective hunting habits. It is possible to divide a grey wolf population by status into four categories from high to low, namely, α, β, γ, and ε. In the solution space, these can be mapped onto the optimal, second-best, third-best, and candidate solutions, respectively. In particular, α, β, and γ are leaders in terms of hunting behavior, which occurs in three stages—stalking and approaching prey; chasing and closing around the prey; and, finally, attacking and killing. The closing around behavior can be described mathematically as follows.

$$D = \left|C \otimes X_p(t) - X(t)\right| \tag{1}$$
$$X(t+1) = X_p(t) - A \otimes D \tag{2}$$
$$A = (2 \times r_1 - 1) \times a \tag{3}$$
$$C = 2 \times r_2 \tag{4}$$
$$a = 2 - 2t/MaxDT, \tag{5}$$

where D is the distance between the prey and the wolf; $X_p(t)$ and $X(t)$ are the location vectors of the prey and wolf, respectively; A and C are coefficient vectors; a is the rate of linear convergence, which decreases linearly from two to zero as the number of iterations increases; r_1 and r_2 are random variables in the interval [0,1]; and $MaxDT$ is the maximum number of iterations. The respective hunting behaviors can then be described using the following mathematical expressions:

$$D_\alpha = |C_1 \otimes X_\alpha(t) - X(t)| \tag{6}$$
$$D_\beta = \left|C_2 \otimes X_\beta(t) - X(t)\right| \tag{7}$$
$$D_\gamma = \left|C_3 \otimes X_\gamma(t) - X(t)\right| \tag{8}$$
$$X_1 = X_\alpha(t) - A_1 \otimes D_\alpha \tag{9}$$
$$X_2 = X_\beta(t) - A_2 \otimes D_\beta \tag{10}$$
$$X_3 = X_\gamma(t) - A_3 \otimes D_\gamma \tag{11}$$
$$X(t+1) = (X_1 + X_2 + X_3)/3, \tag{12}$$

where D_α, D_β, and D_γ are the distances between wolf ε and wolves α, β, and γ, respectively. Equation (12) indicates the current location of wolf ε.

B. KIFCM

It has been shown in [5] that the KIFCM algorithm can be updated as follows:

$$J = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\|\varphi(x_i - \varphi(v_j))\right\|^2 + \sum_{i=1}^{N} \pi_i^* e^{1-\pi^*}, \tag{13}$$

where $\pi_i^* = \frac{1}{c}\sum_{k=1}^{c} \pi_{ki}$ is the degree of hesitation concerning whether the i-th element belongs to the k-th cluster center and ϕ is a non-linear mapping function from the lower-dimensional input space to the higher-dimensional characteristic space.

Based on the definition of a kernel by Mercer, we obtain the following:

$$\begin{aligned} d^2(x_i, v_j) &= \left\|\varphi(x_i) - \varphi(v_j)\right\|^2 \\ &= K(x_i, x_i) + K(v_j, v_j) - 2K(x_i, v_j). \end{aligned} \tag{14}$$

Equation (13) can therefore utilize any kernel function that satisfies the requirements set by Mercer. As Gaussian kernels are superior at suppressing image data noise, they can be used to calculate the distance between each data point and the cluster center, which is $K(x, x) = 1$. Thus, (14) becomes

$$d^2(x_i, v_j) = \left\|\varphi(x_i) - \varphi(v_j)\right\|^2 = 2 - 2K(x_i, v_j). \tag{15}$$

Substituting (15) into (13) produces the following KIFCM objective function:

$$J = 2\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{*m}(1 - K(x_i, v_j)) + \sum_{i=1}^{N} \pi_i^* e^{1-\pi^*}. \tag{16}$$

The optimal solution for (16) can then be calculated to obtain the iteration expressions for obtaining membership $\boldsymbol{u}_{ij}^*$ and cluster centers $\boldsymbol{v}_j^*$ as follows.

$$u_{ij}^* = \left(\frac{\left(1 - K(x_i, v_j)\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{C}\left(1 - K(x_i, v_j)\right)^{\frac{1}{m-1}}}\right)^{-1} + \pi_{ij}^* \tag{17}$$

$$v_j^* = \frac{\sum_{j=1}^{N} u_{ij}^* x_j}{\sum_{j=1}^{N} u_{ij}^*} \tag{18}$$

$$\pi_{ij}^* = \begin{cases} \dfrac{1-2m_0}{m_0} u_{ij}(k), & 0 \le u_{ij}(k) \le m_0 \\ \dfrac{(1-2m_0)+(2m_0-1)u_{ij}(k)}{1-m_0}, & m_0 < u_{ij}(k) < 1. \end{cases} \tag{19}$$

Finally, the iteration termination condition can be set as $|U_{new} - U_{old}| < \eta$ to obtain the final matrix of cluster centers and memberships.

III. KIFCM ALGORITHM GUIDED BY GWO WITH DIFFERENTIAL MUTATION

A. EXTRACTION OF ROBUST SPATIAL INFORMATION

Gaussian noise and salt-and-pepper noise are two common types of noises in nature. Gaussian noise is, in essence, normally distributed noise, and images contaminated by Gaussian noise experience a change in greyscale value conforming to a Gaussian distribution. To eliminate the influence of Gaussian noise on image segmentation, the proposed algorithm uses the mean of pixels within a neighborhood window to represent the information for a specific pixel. Salt-and-pepper noise is noise containing only two greyscale values. Therefore, images contaminated by salt-and-pepper noise will experience a change in the greyscale values of a certain number of pixels to either 0 or 255. The most effective way to eliminate salt-and-pepper noise is via a median filter. However, as there will inevitably be pixels whose initial greyscale values are either 0 or 255, to ensure that authentic image information is retained, it is useful to develop strategies that guarantee that these pixels are not removed by the filter algorithm. With this goal in mind, we propose a new method to extract spatial information from images to overcome the effect of noise on image segmentation.

To construct the greyscale filter, we first define $X = \{x_1, x_2, x_3, \cdots, x_n\}$, an image with n pixels in which the greyscale value of pixel j is x_j. To account for salt-and-pepper noise, we divide the pixels into three greyscale categories, namely $x_j = 0$, $x_j = 255$, and $0 < x_j < 255$. To acquire new spatial information from the image pixels, we choose a neighborhood window with a diameter of ρ for overall zigzag sliding and define S_j^5 as the 5×5 neighborhood window with pixel j at its center [10]. The number of pixels with the greyscale values $x_j = 0$ and 255 within the neighborhood are denoted by ζ and η, respectively.

When $\zeta + \eta = 25$, there is a relatively high likelihood that the greyscale value of pixel j is either 0 or 255. Under a voting model, the greyscale value with the most presence would be assigned as the spatial value of j: therefore, if $\zeta > \eta$, we define $x_{jnew} = 0$; otherwise, if $\zeta < \eta$, $x_{jnew} = 255$.

On the other hand, when $\zeta + \eta \neq 25$, the mean $\bar{s}$ of the greyscale values of all pixels for which $x_j \neq 0$ and $x_j \neq 255$ are calculated as the spatial information of pixel j, and we define

$$x_{jnew} = \bar{s}. \tag{20}$$

If the greyscale values of pixel j, ζ, and η within the neighborhood meet any of the following three conditions, the spatial information of the pixels are updated accordingly:

1. $x_j = 0$ and $\zeta + \eta \ge 2$: This suggests a large likelihood that the greyscale value of pixel j equals one, in which case the spatial information of j is updated as follows:

$$x_{jnew} = \bar{s}\left[1 - \left(\frac{\zeta}{25-\eta}\right)^3\right]. \tag{21}$$

2. $x_j = 255$ and $\zeta + \eta \ge 2$: This suggests a large likelihood that the greyscale value of pixel j equals 255, in which case the spatial information of j is updated as follows:

$$x_{jnew} = 255\left(\frac{\eta}{25-\zeta}\right)^3 + \bar{s}\left[1 - \left(\frac{\eta}{25-\zeta}\right)^3\right]. \tag{22}$$

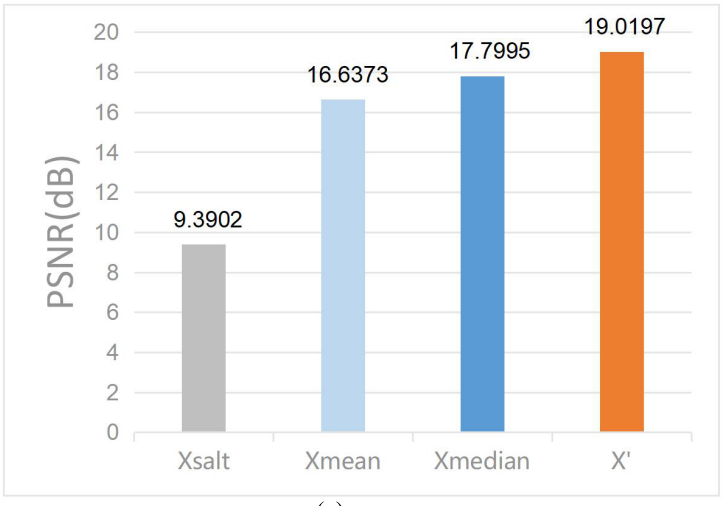
3. $x_j \neq 0$, $x_j \neq 255$, and $\zeta + \eta \ge 2$: This case corresponds to the highest likelihood that there is salt-and-pepper noise in other pixels in the neighborhood, in which case j is updated as follows [10]:

$$\begin{aligned} x_{jnew} = {} & \frac{\zeta}{25}\bar{s}\left[1 - \frac{\zeta}{25-\eta}\right] \\ & + \frac{\eta}{25}\left\{255\left(\frac{\eta}{25-\zeta}\right)^3 + \bar{s}\left[1 - \left(\frac{\eta}{25-\zeta}\right)^3\right]\right\} \\ & + \frac{25-\zeta-\eta}{25}\bar{s}. \end{aligned} \tag{23}$$

Equation (20) both eliminates the influence of Gaussian noise using a mean filter and removes the effect of salt-and-pepper noise contamination. Equations (21) and (22) attempt to retain the effect of the original pixels with greyscale values of 0 or 255 on the spatial information to effectively avoid any loss of image information, whereas (23) takes the contamination of other pixels in the neighborhood by salt-and-pepper noise into account.

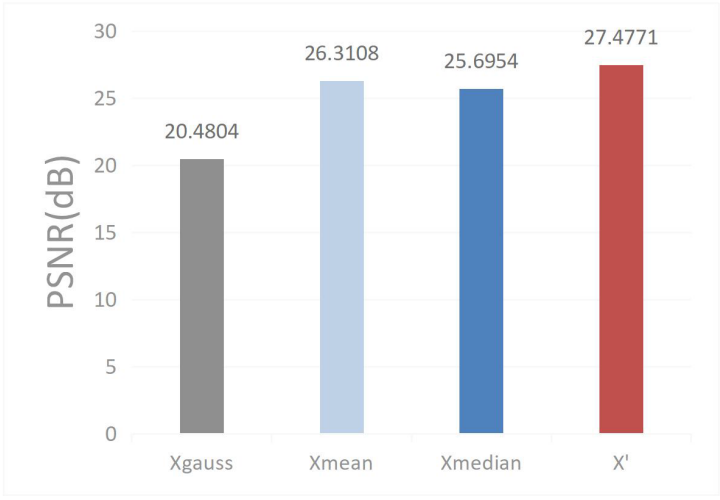
To examine the effectiveness of the spatial information obtained using this method, we added 40% salt-and-pepper noise and Gaussian noise at a 0.009 variance to the Lenna image. Fig. 1 shows a comparison of the noise resistance performance using different spatial information. The greyscale values of the image contaminated by salt-and-pepper noise in Fig. 1(a) and Gaussian noise in Fig. 1(b) are X_{salt} and X_{Gauss}, respectively. Figs. 1(c) and (d) show representations of the pixels within the neighborhood window following application of the mean filter and processing of mean and median spatial information; namely, Fig 1(c) compares the peak signal-to-noise ratio (PSNR) of X' with those of X_{salt}, X_{mean}, and X_{median}; and Fig. 1(d) compares the PSNR values of X_{Gauss}, X_{mean}, X_{median}, and X' [11]. The results in the figures confirm that the robust spatial information obtained using the proposed filter can effectively suppress the effect of salt-and-pepper and Gaussian noises.

(a)



(c)

(b)



(d)

FIGURE 1. Comparison of noise resistance performance using different spatial information: (a) Salt-and-pepper noise, (b) Gaussian noise image X_{gauss}, (c) Performance comparison of PSNR, (d) Performance comparison of PSNR.

Based on the spatial information x_{jnew} obtained using the method described above and the application of the KIFCM algorithm, it is possible to construct a KIFCM objective function incorporating the filter-derived spatial information as follows:

$$J = 2\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{*m}(1 - K(x_i, v_j)) + 2\tau\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{*m}(1 - K(x_{inew}, v_j)) + \sum_{i=1}^{N}\pi_i^* e^{1-\pi^*}, \tag{24}$$

where $\tau = 40$ is the weight of the spatial information. By applying Lagrange multiplier optimization to (24), it is possible to obtain the membership u_{ij}^*:

$$u_{ij}^* = \left(\frac{\left(1 - K(x_i, v_j)\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{C}\left(1 - K(x_i, v_j)\right)^{\frac{1}{m-1}}}\right)^{-1} + \tau\left(\frac{\left(1 - K(x_{inew}, v_j)\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{C}\left(1 - K(x_{inew}, v_j)\right)^{\frac{1}{m-1}}}\right)^{-1} + \pi_{ij}^*. \tag{25}$$

B. WOLF PACK POSITION UPDATE STRATEGIES BASED ON DYNAMIC RANDOM DIFFERENTIAL MUTATION

It can be inferred from (12) that changes in the location of wolf ε will be coupled with changes in the locations of wolves α, β, and γ. When α, β, and γ are all at their local optimal locations, the other wolves in the pack are also likely to be close to local optima, which will affect the search for the global GWO optima. By incorporating a typical differential mutation evolution algorithm [10], it is possible to add a dynamic random differential mutation mechanism to the GWO algorithm. This mechanism first randomly selects three grey wolves and then rescales the differential components of two of them. The rescaled differential components are then combined with those of the third wolf to produce a differential

disturbance that improves the algorithm's ability to seek global optima. The proposed method adopts the following differential mutation mechanism:

$$X_i(t+1) = X_{i1}(t) + \varphi(X_{i2}(t) - X_{i3}(t)), \tag{26}$$

where X_{i1}, X_{i2}, and X_{i3} are the location vectors of any three wolves in the pack, with $i1$, $i2$, $i3 \in$ [1,N] that do not repeat, and a random scaling factor of $\varphi = 0.5 + 0.5rand1$.

C. GREEDY MECHANISM

Although (12) and (26) can be used to update the locations of the grey wolves, there is no guarantee that the updated location vectors will be better fitted than the original vectors. Therefore, we adopt a greedy mechanism to compare the degrees of fitness of the new and original solutions and retain the solution with the better fitness:

$$X_i(t) = \begin{cases} X_i(t), & fit(X_i'(t)) < fit(X_i(t)) \\ X_i'(t), & fit(X_i(t)) < fit(X_i'(t)). \end{cases} \tag{27}$$

D. IGWO PSEUDOCODE

The dynamic random differential mutation procedure is applied using the following IGWO algorithm:

Input: Wolf pack scale, N; maximum number of iterations, MaxDT

Output: X_α

1. Initialize locations of N wolves
2. Calculate the degree of fitness of each grey wolf; select wolves α, β, and γ and record their location vectors, t = 0
3. while t < MaxDT
4. for i =1 to N do
5. if rand1 < 1-t/MaxDT
6. execute dynamic random differential mutation using (26) and update locations of grey wolves
7. else update locations of grey wolves using (6) to (12)
8. end if
9. end for
10. calculate the degree of fitness of each grey wolf
11. execute greedy mechanism, update wolves α, β, and γ and their location vectors
12. t = t + 1
13. end while
14. return X_α

P = 1 – t/MaxDT is the selection probability of the random difference mutation strategy. The value of 1 – t/MaxDT in the early search stages is large. The random difference mutation strategy predominantly improves the exploration ability. In the late search stages, the value of P is small and the GWO search method predominantly improves the mining capacity. By IGWO pseudocode, compared with the GWO update method ((6)–(12)), the random difference mutation strategy only uses (26) to update the position. The update expression is simple, and the calculation minimal, so the computational complexity is small.

TABLE 1. Classic functions.

Formulation	Range	F/min
$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	[-100,100]	0
$f_4(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	[-5.12,5.12]	0
$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	[-32,32]	0
$f6(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0

E. PERFORMANCE ANALYSIS OF IGWO

To test the optimization performance of the IGWO, we employed the six classic test functions f_1 to f_6 listed in Table 1 as objects for comparing the optimization results of the proposed algorithm with those of GWO [11] and mean GWO (MGWO) [12]. Specifically, f_1 to f_3 are unimodal functions with variable dimensions, whereas f_4 to f_6 are multimodal functions with variable dimensions.

All of the assessments were carried out using an Intel® Core™ i5-3230M computer with a Windows 10 operating system, 2.60 GHz CPU, and 4 GB RAM. The code was implemented using the MATLAB 2014R programming language. The size of the population was set to z = 20; 20 independent executions with a maximum number of 500 iterations were carried out. The test results are listed in Table 2.

The results listed in Table 2 indicate that the IGWO algorithm obtained much higher accuracy in seeking the optimum than either the GWO or MGWO algorithm on both the unimodal (f_1 to f_3) and multimodal functions (f_4 to f_6).

To visually represent the convergence performance of GWO, MGWO, and IGWO, Fig. 2 shows the convergence curves of the three algorithms on the test functions f_1, f_2, and f_5. It is apparent that IGWO achieves a higher speed of convergence than either the GWO or MGWO algorithm.

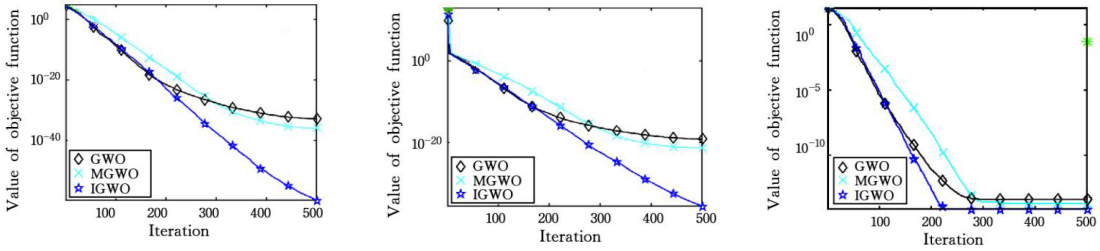


FIGURE 2. Optimization convergence curves of three algorithms on three test functions.

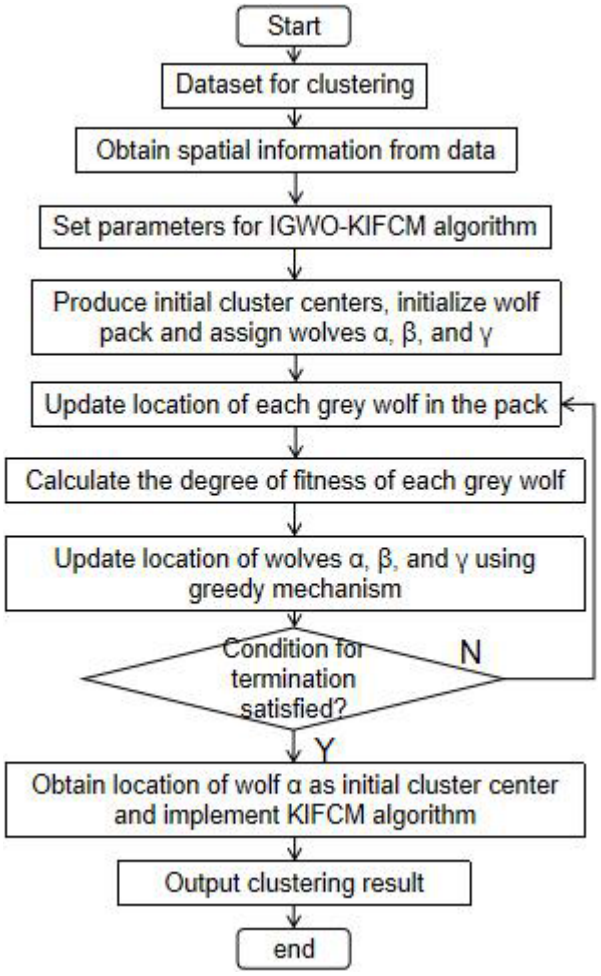


FIGURE 3. Flow chart of IGWO-KIFCM algorithm.

F. IGWO-KIFCM ALGORITHM FLOW

The process flow of the IGWO-KIFCM algorithm is shown in Fig. 3.

G. IGWO-KIFCM TIME COMPLEXITY ANALYSIS

The IFCM clustering algorithm is divided in such a manner that the sum of all distances D from each data object P to its corresponding cluster center is the smallest. Here, D is calculated as

$$D = \min\left(\sum_{i=1}^{k}\sum_{p\in C_i} dist\,(p, o_i)\right) \tag{28}$$

TABLE 2. Comparison of algorithm optimization performance using classic functions.

Function	Statistics	GWO	MGWO	IGWO
$f1$	Mean	2.715×10^{-31}	6.832×10^{-39}	5.684×10^{-53}
	Std. Dev.	3.812×10^{-31}	2.691×10^{-38}	1.896×10^{-52}
$f2$	Mean	7.869×10^{-21}	6.932×10^{-26}	3.136×10^{-34}
	Std. Dev.	6.767×10^{-21}	6.854×10^{-26}	6.547×10^{-34}
$f3$	Mean	2.839×10^{-9}	1.653×10^{-11}	5.666×10^{-16}
	Std. Dev.	1.233×10^{-8}	6.722×10^{-11}	4.192×10^{-16}
$f4$	Mean	2.353	3.218×10^{-1}	2.368×10^{-5}
	Std. Dev.	1.288	1.173×10^{-1}	1.149×10^{-5}
$f5$	Mean	7.756×10^{-13}	2.294×10^{-14}	7.923×10^{-15}
	Std. Dev.	2.412×10^{-13}	4.258×10^{-15}	1.230×10^{-15}
$f6$	Mean	2.101×10^{-3}	1.478×10^{-3}	2.663×10^{-7}
	Std. Dev.	6.231×10^{-3}	4.838×10^{-3}	7.405×10^{-7}

where O_i is the center of cluster C_i and $dist$ is the distance function. When $k = 1$, the time complexity of the algorithm is $O(n^2)$. After introducing the kernel function, when the i-th initial center point is selected where $i \in [1, k]$, the algorithm running time t is computed as

$$t = (n + 1)\,(i - 1) \tag{29}$$

The running time of the i-th initial center point is

$$T_i = mt = m\,(n + 1)\,(i - 1) \tag{30}$$

The time complexity of the KIFCM algorithm $T\,(n)$ is

$$T\,(n) = O\left(\sum_{i=1}^{k} T_i\right) = O\left(mk^2 n\right) \tag{31}$$

When $m = 2$, the KIFCM algorithm is expressed as $T\,(n) = O\left(k^2 \cdot n\right)$.

The flowchart in Fig. 3 shows that both the initialization and the robust information extraction in the program are single line statements with a time complexity of $O(j)$. From the pseudocode of the IGWO, it can be observed that, in the *for*-loop, the outer layer traverses all the data, while the inner layer finds the optimal clustering center. The IGWO algorithm is, thus, a two-layer loop with a complexity of $O(k^2)$. Therefore, the time complexity of the IGWO-KIFCM algorithm is $T\,(n) = O\left(jk^2 \cdot n\right)$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EFFECT OF NEIGHBORHOOD WINDOW SIZE ON CLUSTERING ACCURACY

From Section III.A, it can be seen that neighborhood window diameter ρ has a direct impact on the noise processing effect. The smaller the ρ value, the lower the amount of spatial

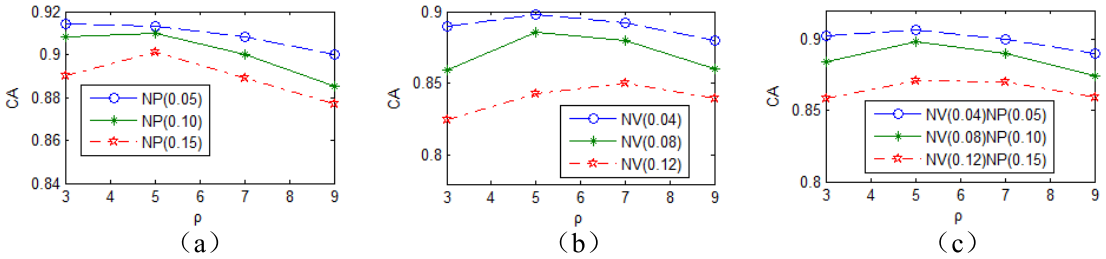


FIGURE 4. Clustering accuracy of algorithms with different ρ under different types of noise.

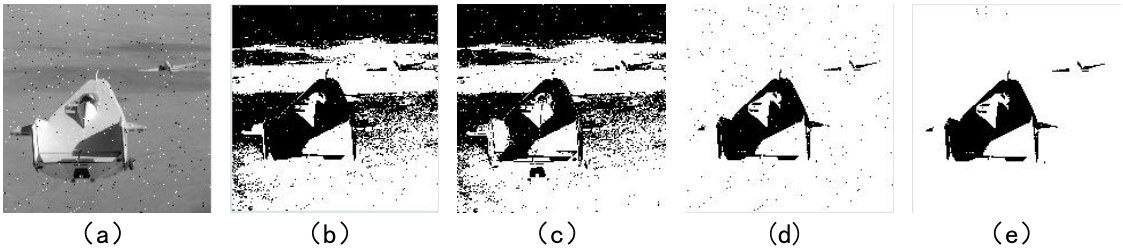


FIGURE 5. Segmentation of lifting body image (salt-and-pepper noise at 0.01 density): (a) Gaussian noise image, (b) IFCM, (c) KIFCM, (d) IGWO-KIFCM algorithm.

TABLE 3. Comparison of clustering performance of different algorithms.

Algorithms	Mean	Std.Dev.	Time for clustering/s	Clustering accuracy%
IIFCM	61.012	1.143×10^{-4}	0.0136	90.35
NIFCM	60.843	2.172×10^{-3}	0.8184	91.56
BFWCOM	60.732	4.973×10^{-5}	0.649	92.86
IGWO-KIFCM	60.576	5.198×10^{-7}	0.5433	93.13

information involved in the calculation, and the clearer the image details. The optimal range of ρ values is found by adding salt and pepper, Gaussian, and combined salt and pepper and Gaussian noises to the lifting body image to test ρ. Let ρ be 3, 5, 7, and 9 for testing. Fig. 4 shows the corresponding broken-line diagram of the change of image clustering accuracy with ρ under different noise levels. It can be seen from the figure that the optimal values of ρ are 5 and 7.

B. CLUSTERING EXPERIMENT OF IRIS DATASET

To investigate the clustering ability of the proposed algorithm, we compared its clustering optimization performance on the Iris dataset with the performance of the IIFCM [13], NIFCM [14], and BFWCOM [15] algorithms. The following conditions were set: maximum number of iterations = 200, $m = 2$, iteration error = 0.00005, and Gaussian kernel $\sigma = 0.4$. Each algorithm was run independently 20 times. Table 3 lists the results produced by each algorithm, where it can be seen that the IGWO-KIFCM algorithm has good clustering performance when processing Iris datasets and has some practical value.

C. CLUSTERING EXPERIMENTS WITH ADDITIONAL UCI DATASETS

To further verify the effectiveness of the IGWO-KIFCM algorithm in clustering real datasets, five UCI datasets other than the IRIS dataset were selected for the experiments, and the experimental results are presented in Table 4.

TABLE 4. Rand index RI (%) for different clustering algorithms on UCI datasets.

Dataset	IIFCM	NIFCM	BFWCOM	IGWO-KIFCM
Ecoli	76.82	77.43	82.17	84.40
Zoo	86.23	86.95	90.31	90.65
Automobile	74.07	73.78	74.98	75.24
Seeds	88.92	88.89	89.83	89.96
Glass	71.68	71.91	71.84	72.59

The Rand index (RI) was chosen as the index for evaluating the classification accuracy of the algorithm's clustering results.

$$RI = \frac{f_{aa} + f_{bb}}{n(n-1)} \times 2 \tag{32}$$

Here, f_{aa} is the total number of samples that belong to the same class in the original dataset and remain in the same class in the clustering results, f_{bb} is the total number of samples that belong to different classes in the original dataset and remain in different classes in the clustering results, and n is the number of samples. It can be observed in (32) that $RI \in [0,1]$, and the larger the value, the better the clustering effect.

As shown in Table 4, in comparison with several other algorithms, the IGWO-KIFCM algorithm has the highest Rand index on all the above five UCI datasets. This indicates that it has good clustering performance when processing real datasets and has some practical value.

D. IMAGE SEGMENTATION EXPERIMENT

To evaluate the effect of visual segmentation by the proposed algorithm, we performed image segmentation with the

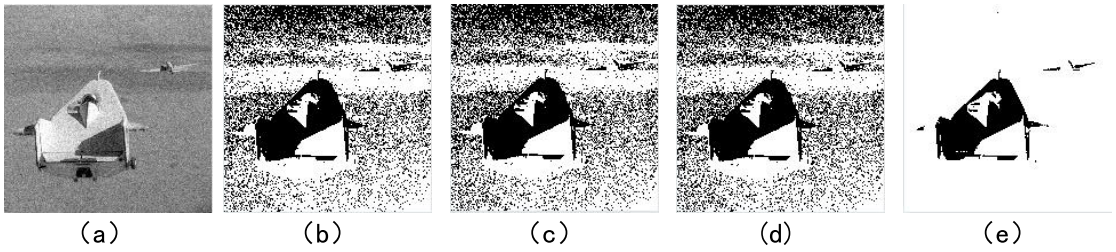


FIGURE 6. Segmentation of lifting body image (Gaussian noise at 0.003 variance): (a)Gaussian noise image, (b) IFCM, (c) KIFCM, (d) IGWO-KIFCM algorithm.

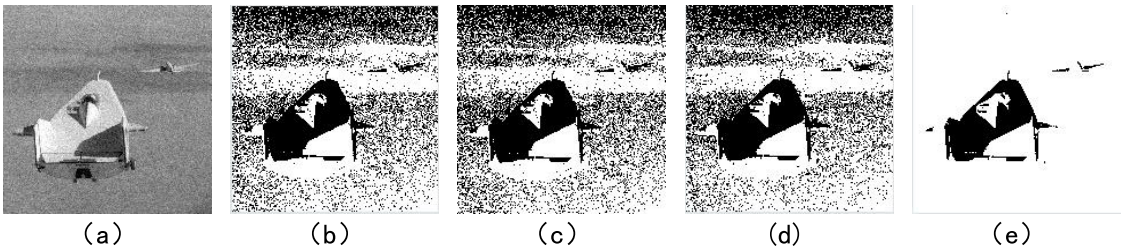


FIGURE 7. Segmentation of lifting body image (Gaussian noise at 0.003 variance + salt-and-pepper noise at 0.01 density): (a)Mixed noise image, (b) IFCM, (c) KIFCM, (d) IGWO-KIFCM algorithm.

TABLE 5. Comparison of segmentation accuracy under different noises.

Image	Noise level	IIFCM	NIFCM	BFWCOM	IGWO-KIFCM
Cameraman	salt &peper1%	87.59	89.72	90.41	92.16
lenna	salt &peper1%	87.23	89.89	90.56	92.43
liftingbody	salt &peper1%	86.97	89.30	90.47	91.37
mri	salt &peper1%	87.86	88.93	89.78	91.02
pout	salt &peper1%	85.13	87.38	88.86	90.06
liftingbody	Gaussian(δg=3)	84.67	87.05	88.62	89.78
mri	Gaussian(δg=3)	85.18	86.76	88.35	89.39
pout	Gaussian(δg=3)	84.20	86.78	87.91	89.24
liftingbody	Gaussian(δg=3)+ salt &peper1%	83.98	86.59	88.14	89.31
mri	Gaussian(6g=3)+ salt &peper1%	83.76	86.27	88.06	88.63
pout	Gaussian(δg=3)+ salt &peper1%	84.01	86.57	87.89	88.12

number of clusters set to two and the maximum number of iterations set to 100. Figs. 5, 6, and 7 show the results of segmentation of images with Gaussian, salt-and-pepper, and mixed noise, respectively. The results demonstrate that the algorithm was able to overcome the impact of all three types of noise and to successfully distinguish the object from the background in each case.

Table 5 shows the accuracy of image segmentation under different noises. It can be seen from this table that the proposed algorithm is superior to the IIFCM, NIFCM, and BFWCOM algorithms in terms of noise suppression. This is because the proposed algorithm fully considers the characteristics of Gaussian and salt-and-pepper noises when obtaining image spatial information, and then designs a targeted robust information extraction strategy that can effectively suppress the influence of Gaussian, salt-and-pepper, and mixed noises during digital image segmentation.

V. CONCLUSION

To overcome the disadvantages of conventional IFCM algorithms in terms of sensitivity to noise and initial cluster centers, in this study we developed a kernel-based intuitionistic fuzzy clustering image segmentation algorithm using an improved grey wolf optimizer with differential mutation (IGWO-KIFCM). After robust image spatial information is input to the objective function of the KIFCM algorithm, the IGWO applies dynamic random differential mutation to enhance the ability of the algorithm to search for a global optimum. The results of clustering optimization on six functions using the Iris dataset demonstrated that the IGWO is capable of effectively coping with the problems of KIFCM clustering optimization. The proposed algorithm also achieved good results in segmenting images with various types of noise.

Future work will focus on optimizing the efficiency of the IGWO-KIFCM algorithm as well as conducting effective mining of image spatial information in a larger area to further improve its noise suppression ability.

REFERENCES

[1] J. Xiong, R. Lin, Z. Liu, Z. He, Z. G. Yang, and R. Bu, "Visual technology of picking robot to detect litchi at nighttime under natural environment," *Trans. Chin. Soc. Agricult. Machinery*, vol. 48, no. 11, pp. 28–34, Nov. 2017, doi: 10.6041/j.issn.1000-1298.2017.11.004.

[2] F. Fei, L. Pei-Xue, L. Li, and C. Yu-Jie, "Study of FCM fusing improved gravitational search algorithm in medical image segmentation," *Comput. Sci.*, vol. 45, no. 6A, pp. 252–254, Dec. 2018.

[3] C. Huang, K. Lin, M. Wu, K. Hung, G. Liu, and C. Jen, "Intuitionistic fuzzy C-means clustering algorithm with neighborhood attraction in segmenting medical image," *Soft Comput.*, vol. 19, no. 2, pp. 459–470, Feb. 2015.

[4] D. Kumar, H. Verma, A. Mehra, and R. K. Agrawal, "A modified intuitionistic fuzzy C-means clustering approach to segment human brain MRI image," *Multimedia Tools Appl.*, vol. 78, no. 10, pp. 12663–12687, May 2019, doi: 10.1007/s11042-018-5954-0.

[5] Y. Xiaodong, L. Ying-Jie, S. Ya-Fei, Y. Shao-Hua, and S. Xiao-Yong, "Intuitionistic fuzzy C-means clustering algorithm based on kernelled distance," *Acta Electro. Sin.*, vol. 44, no. 10, pp. 2530–2534, Oct. 2016.
[6] F. Guo, X. Wang, and J. Shen, "Adaptive fuzzy C-means algorithm based on local noise detecting for image segmentation," *IET Image Process.*, vol. 10, no. 4, pp. 272–279, Apr. 2016, doi: 10.1049/iet-ipr.2015.0236.
[7] T. Jayabarathi, T. Raghunathan, B. R. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy*, vol. 111, pp. 630–641, Sep. 2016, doi: 10.1016/j.energy.2016.05.105.
[8] Z. Xin-Ming, J. Yun, L. Shang-Wang, L. Gui-Qi, D. Zhi, and L. Yan, "Hybrid coyote optimization algorithm with grey wolf optimizer and its application to clustering optimization," *Acta Automat. Sin.*, pp. 1–17, Aug. 2020.
[9] Z. Xinming, W. Xia, and K. Qiang, "Improved grey wolf optimizer and its application to high-dimensional function and FCM optimization," *Control Decis.*, vol. 34, no. 10, pp. 2073–2084, Aug. 2019.
[10] F. Zhao, W. Sun, H. Lui, and Z. Zeng, "Intuitionistic fuzzy clustering image segmentation based on flower pollination optimization with nearest neighbor searching," *J. Electr. Inf. Technol.*, vol. 42, no. 4, pp. 1005–1012, 2020.
[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[12] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, May 2016, Art. no. 7950348, doi: 10.1155/2016/7950348.
[13] H. Verma, R. K. Agrawal, and A. Sharan, "An improved intuitionistic fuzzy c-means clustering algorithm incorporating local information for brain image segmentation," *Appl. Soft Comput.*, vol. 46, pp. 543–557, Sep. 2016, doi: 10.1016/j.asoc.2015.12.022.
[14] R. Lan and Y. Lin, "Suppressed non-local spatial intuitionistic fuzzy C-means image segmentation algorithm," *J. Electron. Inf.*, vol. 41, no. 6, pp. 1472–1479, Jun. 2019, doi: 10.11999/JEIT180651.
[15] W. Zhang, T. Huang, and J. Chen, "A robust bias-correction fuzzy weighted C-Ordered-Means clustering algorithm," *Math. Problems Eng.*, vol. 2019, Jun. 2019, Art. no. 5984649, doi: 10.1155/2019/5984649.

XIANGXIAO LEI graduated from Central South University, in 2005. He is currently pursuing the Doctor of Engineering degree. He worked with the Changsha Social Work College. He is also an Associate Professor. His research interests include intelligent control and digital image processing.

HONGLIN OUYANG received the bachelor's degree from Hunan University, in 2005, where he is currently pursuing the Doctor of Engineering degree. He is also a Professor with Hunan University. His research interests include the power electronics and control engineering fields.