Robust Deep Kernel-Based Fuzzy Clustering With Spatial Information for Image Segmentation

Lujia Lei (✉ leilujia@stu.xupt.edu.cn)

Xi'an University of Posts and Telecommunications <https://orcid.org/0000-0001-8529-6952>

Chengmao Wu

Xi'an University of Posts and Telecommunications

Xiaoping Tian

Xi'an University of Posts and Telecommunications

Research

Keywords: image segmentation, fuzzy clustering, auto-encoder, deep learning, kernel function, neighborhood information, robustness

Posted Date: August 24th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-805598/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Applied Intelligence on April 11th, 2022. See the published version at <https://doi.org/10.1007/s10489-022-03255-3>.

RESEARCH

Robust deep kernel-based fuzzy clustering with spatial information for image segmentation

Lujia Lei^{1*}, Chengmao Wu² and Xiaoping Tian³

*Correspondence:

leilujia@stu.xupt.edu.cn

¹School of Electronics and Engineering, Xi'an University of Posts & Telecommunications, 710121 Xi'an, China
Full list of author information is available at the end of the article

Abstract

Clustering algorithms with deep neural network have attracted wide attention of scholars. A deep fuzzy K-means clustering algorithm model with adaptive loss function and entropy regularization (DFKM) is proposed by combining automatic encoder and clustering algorithm. Although it introduces adaptive loss function and entropy regularization to improve the robustness of the model, its segmentation effect is not ideal for high noise; At the same time, its model does not use a convolutional auto-encoder, which is not suitable for high-dimensional images. Therefore, on the basis of DFKM, this paper focus on image segmentation, combine neighborhood median and mean information of current pixel, introduce neighborhood information of membership degree, and extend Euclidean distance to kernel space by using kernel function, propose a dual-neighborhood information constrained deep fuzzy clustering based on kernel function (KDFKMS). A large number of experimental results show that compared with DFKM and classical image segmentation algorithms, this algorithm has stronger anti-noise robustness.

Keywords: image segmentation; fuzzy clustering; auto-encoder; deep learning; kernel function; neighborhood information; robustness

1 Introduction

Among many fuzzy clustering algorithms, fuzzy C-means (FCM) clustering algorithm (1984) [1] is the most classical one. Yang et al. (2008) [2] extended FCM to kernel space and proposed kernel fuzzy c-means clustering algorithm (KFCM), KFCM overcomes the difficulty of FCM to deal with non-linear separable samples. However, neither FCM nor KFCM algorithms consider the spatial neighborhood information of pixels, which leads to poor clustering results of noisy images. Therefore, Ahmed et al. (2002) [3] introduced the local spatial restriction term into the objective function of FCM, and proposed an FCM algorithm based on spatial information (FCM-S). However, the algorithm needs to calculate the neighborhood term of each pixel in each iteration, which leads to a long running time. Cai et al. (2007) [4] proposed the fast generalized fuzzy C-means clustering (FGFCM) algorithm, which further improved the segmentation speed of the algorithm. To further reduce the complexity of FCM-S, Chen and Zhang (2004) [5] proposed to replace FCM-S neighborhood pixels with mean filter and median filter respectively, and obtained FCM-S1 and FCM-S2 algorithms. At the same time, they used kernel induced distance instead of Euclidean distance, and gave the kernel extension algorithms (2004) [6] KFCM-S, KFCM-S1 and KFCM-S2. Krinidis and Chatzis (2010) [7] proposed a fuzzy clustering algorithm based on local information (FLICM), which introduced a new fuzzy factor to balance the noise effect and detail information of the image. Hou Lili (2016) [8] added membership information to the spatial distance, improved the fuzzy factor of FLICM, and further made the segmentation performance of the FLICM algorithm better. Later, Gong Maoguo et al. (2011) [9] extended the FLICM algorithm to the kernel space, and proposed a C-means clustering algorithm based on the kernel function to weigh the fuzzy weighting factor (KWFLICM).

In recent years, with the rise of deep learning, clustering methods with deep neural networks have attracted great attention. These methods are designed to use, for example, Deep Belief Network (DBN) (2009) [10], Deep Automatic Encoder (DAE) (1998) [11] and Convolutional Neural Network (CNN) (2018) [12] to develop and process data with more complex structure. First of all, some of these models separate the clustering process from the deep network training process, that is, apply the clustering method to the feature representation of the hidden layer of the training deep network. For example, Mojtaba Yeganejou et al. (2018) [13] proposed the deep clustering model, which

combines the convolutional neural network (CNN) and FCM algorithm, has a good performance on the MNIST data set. Zhao Ziyuan et al. (2020) [14] combined deep belief network (DBN) and fuzzy C-means (FCM), and proposed an unsupervised deep fuzzy C-means clustering network (UDFCMN) to process lung CT images. In the network, the pre-processed image is first encoded as a multi-layer hidden layer vector to extract deep features, then the FCM algorithm is used to cluster to generate initial cluster labels, and the network parameters are fine-tuned by using the labels through backpropagation. Caron et al. (2018) [15] proposed a clustering method called Deepcluster, which also directly uses the features obtained by the convolutional neural network (CNN) for clustering, and uses the clustering results as pseudo-labels to update the convolutional network. So that deep feature learning and clustering can be performed in the network at the same time. Secondly, some feature learning processes and clustering processes are combined in a unified framework. For example, Xie et al. (2016) [16] first proposed Deep Embedded Cluster (DEC), which embeds the conventional K-means algorithm into the auto-encoder network model for feature mining and sample clustering, and used Kullback-Leibler divergence loss function to fine-tune the parameters of the model. Because DEC model uses stacked auto-encoder instead of convolutional auto-encoder, it is not suitable for high-dimensional images, so Li and Qiao et al. (2018) [17] proposed a deep embedded clustering model (DBC) based on convolutional auto-encoder. Yang et al. (2016) [18] proposed an unsupervised learning recursive framework (JULE) which combines deep representation and image clustering. In this framework, the image clustering process and the deep representation are carried out in the forward propagation and backward propagation of convolution network, respectively. At the same time, the clustering results can provide supervision information for the deep representation. However, these methods all use stochastic gradient descent (SGD), which leads to relatively slow convergence. And these methods alternate between continuous gradient descent updating and clustering. In contrast, Maziarmoradi Fard et al. (2020) [19] proposed a deep K-mean algorithm, which embeds the K-mean algorithm into the autoencoder. In the model, the deep representation and clustering parameters are jointly learned by the gradient update. Zhang and Li et al. (2020) [20] combined autoencoder and fuzzy k-means algorithm to form a deep fuzzy k-means clustering algorithm (DFKM) with adaptive loss and entropy regularization, which proposed a new gradient update to execute simultaneously deep feature extraction and clustering have good performance in data set and image segmentation. However, most of these deep clustering algorithm models do not consider the anti-noise robustness of high-dimensional data. Therefore, inspired by [7] and [20], this paper introduces neighborhood pixel space and membership information into the deep fuzzy clustering algorithm to improve the anti-noise robustness of the original algorithm, and uses the kernel space distance measure to replace the Euclidean distance space measure, proposes a deep fuzzy clustering robust image segmentation algorithm based on the dual-neighborhood pixel spatial information of kernel function. Firstly, the local spatial information restriction term is introduced into the objective function of DFKM. However, due to the high complexity of each iteration of the algorithm, mean filtering and median filtering image pixels are used to replace local spatial information pixels, and similar effects can be achieved. At the same time, the kernel function is used to extend the Euclidean space to the kernel space. Because the deep robust fuzzy clustering algorithm embedded in a single neighborhood spatial information has certain limitations, we combine the pixel median and mean neighborhood information, and introduce neighborhood membership information, propose the final algorithm (KDFKMS). A large number of image segmentation experiments show that the proposed algorithm not only improves the segmentation performance of the algorithm, but also enhances the anti-noise robustness.

2 Methods

2.1 Kernel-based fuzzy clustering

At present, kernel method is widely used in nonlinear classification in pattern recognition. In order to understand kernel method accurately, we need to understand kernel function first. Kernel function is defined as follows.

Definition 1. (Kernel function): Suppose that $\mathbf{H}$ is a feature space and a mapping $\Phi : \mathbf{x} \rightarrow \mathbf{H}$. If $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$, K is called kernel function. Kernel function between $\Phi(x)$ and $\Phi(y)$ is expressed as:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle \quad (1)$$

where, $\langle \Phi(x), \Phi(y) \rangle$ represents inner product operation of x, y mapping on the feature space, then

$$\begin{aligned} \|\Phi(x) - \Phi(y)\|^2 &= (\Phi(x) - \Phi(y))^T (\Phi(x) - \Phi(y)) \\ &= K(x, x) - 2K(x, y) + K(y, y) \end{aligned} \quad (2)$$

At present, Gaussian kernel is one of the most widely used kernel function, which is defined as follows.

$$K(x, y) = \exp \left(-\frac{\|x - y\|^2}{2\sigma^2} \right) \quad (3)$$

Where x and y are the i -th and j -th samples in the entire data set, respectively; σ^2 is the Gaussian kernel function scale parameter, which can be used to indicate the difference between all samples from the entire data set, and is defined as follows.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d})^2} \quad (4)$$

Where, $d_i = \|x_i - \bar{x}\|$ represents the gray distance difference between the sample point x_i and the center $\bar{x} = \sum_{i=1}^n x_i / n$ of the sample data set $\mathbf{X} = \{x_1, x_2, \dots, x_i, \dots, x_n\}$, and $\bar{d}$ is the average of the distances d_i of all the sample points x_i , that is, $\bar{d} = \sum_{i=1}^n d_i / n$. From Eq.(3), we know that $K(x, x) = 1, K(y, y) = 1$. Therefore, for Gaussian kernel function, Eq.(2) can be updated as follows.

$$\|\Phi(x) - \Phi(y)\|^2 = 2 - 2K(x, y) \quad (5)$$

Yang et al.(2008) [2] introduced Gaussian kernel function into fuzzy clustering and proposed KFCM, which is to map the original space samples to high-dimensional feature space through Gaussian kernel function. Based on the above description, the fuzzy clustering algorithm with Gaussian kernel function is used to optimize the objective function as follows.

$$\min J(U, V) = 2 \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m (1 - K(x_i, v_j)) \quad (6)$$

The membership function and cluster center updating formula corresponding to the above algorithm are as follows.

$$v_j = \frac{\sum_{i=1}^n u_{ij}^m K(x_i, v_j) x_j}{\sum_{i=1}^n u_{ij}^m K(x_i, v_j)} \quad (7)$$

$$u_{ij} = \frac{(1 - K(x_i, v_j))^{-1/(m-1)}}{\sum_{j=1}^c (1 - K(x_i, v_j))^{-1/(m-1)}} \quad (8)$$

Here, m represents the fuzzy index; n is the total number of samples; c is the number of clusters; x_i represents the i -th sample. The kernel fuzzy C-means clustering algorithm makes the linearly inseparable samples linearly separable in the high-dimensional feature space, which overcomes the lack of FCM for non-linear separable samples to a certain extent.

2.2 Robust fuzzy clustering with spatial information

Ahmed et al. (2002) [3] proposed a fuzzy C-means clustering algorithm (FCM-S) combining spatial information by introducing local spatial constraints. This algorithm enhances the segmentation performance of noisy images to a certain extent, which is defined as follows.

$$\min J(U, V) = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \|x_i - v_j\|^2 + \frac{\alpha}{N_R} \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \sum_{p \in N_i} \|x_p - v_j\|^2 \quad (9)$$

Where, N_i represents the set of pixels in the local window centered on x_i ; N_R represents the number of pixels in the neighborhood window; α is the coefficient of the control neighborhood item, which is used to adjust the influence of neighborhood information on the center point. The algorithm has achieved good results in reducing the influence of noise on the image. Later, scholars Chen and Zhang (2004) [5] proposed FCM-S1 and FCM-S2 that use mean filtering and median filtering image pixels instead of FCM-S neighborhood pixels to reduce the complexity of the FCM-S.

To further enhance the robustness of FCM-S algorithm to image noise, Chen and Zhang(2004) [6] used the kernel induced distance instead of Euclidean distance, and proposed a kernel fuzzy C-means clustering algorithm (KFCM-S) based on the constraints of the neighborhood space. The optimization model of KFCM algorithm can be expressed as follows.

$$\min J(U, V) = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m (1 - K(x_i, v_j)) + \beta \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m (1 - K(\bar{x}_i, v_j)) \quad (10)$$

At the same time, the $\bar{x}_i$ in KFCM-S is replaced with mean filter pixels and median filter pixels to get KFCM-S1 and KFCM-S2 algorithms.

2.3 Robust fuzzy clustering with local membership

Lchihashi et al.(2001) [21] proposed a fuzzy C-means clustering algorithm (KLFCM) based on KL divergence. This algorithm does not require a fuzzy factor of membership, and the regularization term about KL divergence is added to the objective function of FCM. The

objective function is defined as follows.

$$\min J(U, V) = \sum_{i=1}^n \sum_{j=1}^c u_{ij} \|x_i - v_j\|_2^2 + \lambda \sum_{i=1}^n \sum_{j=1}^c u_{ij} \log \frac{u_{ij}}{\phi_j} \quad (11)$$

In which λ represents the trade-off factor for controlling the fuzziness of the regularization term and satisfies $\lambda > 0$; ϕ_j is used to control the clustering scale, which represents the prior probability of class j and satisfies $\sum_{j=1}^c \phi_j = 1$, can be expressed as:

$$\phi_j = \sum_{i=1}^n u_{ij} / n \quad (12)$$

KLFCM improves the segmentation performance of the original algorithm. In order to improve the anti-noise robustness of the algorithm, Gharieb et al. (2014) [22] proposed a robust fuzzy clustering algorithm based on KL divergence constraint with local membership degree (LMKLFCM), and its objective function is defined as follows.

$$\min J(U, V) = \sum_{i=1}^n \sum_{j=1}^c u_{ij} \|x_i - v_j\|_2^2 + \lambda \sum_{i=1}^n \sum_{j=1}^c u_{ij} \log \frac{u_{ij}}{\pi_{ij}} \quad (13)$$

Among them, λ is the scale parameter that controls the degree of fuzzy clustering; π_{ij} is the membership degree of neighborhood mean, which is calculated by averaging the membership degrees in neighborhood windows around the membership degree u_{ij} of the i -th pixel, and can be expressed as:

$$\pi_{ij} = \frac{1}{\partial_{i'}} \sum_{i' \in \partial_i} u_{i'j} \quad (14)$$

In which ∂_i is a group of pixels in the neighborhood window of the i -th pixel, and $\partial_{i'}$ is the cardinal number of ∂_i . Obviously, all the pixels in the window are linearly weighted and averaged according to the weight $1/\partial_{i'}$, which makes the membership u_{ij} of the pixel in the center of the neighborhood constrained by the local membership π_{ij} , so that the boundary area of the image changes more smoothly, thus achieving the anti-noise effect.

Since the improved algorithm is added to the neighborhood mean membership degree, the image segmentation effect has been significantly improved, but it cannot handle more complex nonlinear neighborhood membership data well. Zhao Quanhua (2019) [23] et al. The replacement in Eq.(13) is expressed as follows:

$$u_{\partial_{ij}} = \sum_{i' \in \partial_i} \lambda_{i'i} u_{i'j} \quad (15)$$

Where ∂_i is the neighborhood pixel set of pixel i , and $\lambda_{i'i}$ is defined as follows:

$$\lambda_{i'i} = \frac{\exp \left(- (2\theta^2)^{-1} (u_{ij} - u_{i'j})^2 \right)}{\sum_{i''} \exp \left(- (2\theta^2)^{-1} (u_{ij} - u_{i''j})^2 \right)} \quad (16)$$

Where θ is a constant in the formula.

3 Proposed deep fuzzy clustering algorithm model

In this article, we use the framework of a combination of convolutional autoencoders and fuzzy clustering algorithms. Auto-encoder is a special instance of deep neural network. After training the network, the data can be reduced to a low-dimensional vector, and then try to reconstruct the input according to this vector. The feature of the auto-encoder is that it can learn deep feature representations in a completely unsupervised manner. The basic framework of this article is shown in Fig.1, where Conv is the convolution operation, Tanh is the activation function, and DeConv represents the deconvolution operation. It can be seen from the figure that firstly, after the original image is preprocessed, rgb values of the image pixel neighborhood information is introduced into the convolutional auto-encoder network. By training the network, the hidden layer features (HL feature) are extracted in the $M/2$ -th layer, then introduce them into the objective function of the fuzzy clustering algorithm.

Hidden layer features are added simultaneously to gradient and weight update iterations as regularization items to optimize the network. Finally, the segmented image and the real image are subjected to performance analysis to draw conclusions.

4 Deep kernel fuzzy clustering with dual-neighborhood information

Because the adjacent pixels of an image have the characteristics of correlation with each other, the neighborhood information of pixels can improve the effective segmentation of noisy images. Firstly, the neighborhood information of pixel membership degree is introduced, and then the Euclidean distance space measure is extended to the kernel space distance measure, a robust image segmentation algorithm based on the neighborhood information of pixels based on kernel function is proposed. Combining kernel function and neighborhood pixel spatial information restriction term, the objective function and constraints are defined as follows.

$$\begin{aligned}
 \min J(U, V) = & \frac{1}{2} \sum_{i=1}^N \|h_i^{(M)} - x_i\|_2^2 + \gamma u_{ij} \log u_{ij} + \frac{\lambda_2}{2} \sum_{m=1}^M \|W^{(m)}\|_2^2 + \|b^{(m)}\|_2^2 \\
 & + \frac{\lambda_1}{2} \sum_{i=1}^N \sum_{j=1}^K d_{ij} u_{ij} \left\| \Phi \left(h_i^{(M/2)} \right) - \Phi \left(v_j \right) \right\|_2^2 \\
 & + \alpha \sum_{i=1}^N \sum_{j=1}^K \tilde{d}_{ij}^{(M/2)} u_{ij} \left\| \Phi \left(\tilde{h}_i^{(M/2)} \right) - \Phi \left(v_j \right) \right\|_2^2 \\
 \text{S.t. } & \sum_{j=1}^K u_{ij} = 1, u_{ij} \in [0, 1], 0 < \sum_{i=1}^N u_{ij} < 1, 1 \leq i \leq N, 1 \leq j \leq K
 \end{aligned} \tag{17}$$

Where, N is the number of cluster samples; K is the number of clusters; v_j is the center of cluster j of low-dimensional feature space sample data, $v_j \in \mathbb{R}^{z'}$, $z' = z(M/2)$; α is the control parameter of the spatial information penalty item, γ is the trade-off parameter that controls the distribution of u_{ij} ; λ_1 and λ_2 are the trade-off parameters; x_i represents the input of the first layer network, and $h_i^{(M/2)}$ represents the characteristic data output by the $M/2$ -th layer network. $h_i^{(m)}$ is described as:

$$h_i^{(m)} = f \left(W^{(m)} h_i^{(m-1)} + b^{(m)} \right) \in \mathbb{R}^{Z_m} \tag{18}$$

Where, $m = 1, 2, \dots, M$; and Z_m is the number of nodes of the M -th layer of neural network; $f(\cdot)$ is the activation function of the network layer, W and b are the weight and bias matrix of the corresponding layer. Because the neighborhood information penalty items in the objective function of the algorithm in this paper are all filtered by the mean or median of the features trained by the network, the updating iteration of the weight and bias matrix of the corresponding layer is still the method proposed by Zhang and Li et al. (2020) [20], so it is iterated by the random gradient descent method, and the iteration formula is as follows:

$$W^{(m)} = W^{(m)} - \mu \nabla_{W^{(m)}} F \quad (19)$$

$$b^{(m)} = b^{(m)} - \mu \nabla_{b^{(m)}} F \quad (20)$$

where, $\nabla_{W^{(m)}} F$ and $\nabla_{b^{(m)}} F$ are the descent gradients of the weight and bias, namely:

$$\nabla_{W^{(m)}} F = \delta_i^{(m)} h_i^{(m-1)T} + \lambda_2 W^{(m)} \quad (21)$$

$$\nabla_{b^{(m)}} F = \delta_i^{(m)} + \lambda_2 b^{(m)} \quad (22)$$

Where,

$$\delta_i^{(m)} = \begin{cases} \left(W^{(m+1)} \delta_i^{(m+1)} + \lambda_1 \Lambda_i^{(m)} \right) \odot f' \left(z_i^{(m)} \right) & m \neq M \\ \left(h_i^{(M)} - h_i^{(0)} \right) \odot f' \left(z_i^{(M)} \right) & m = M \end{cases} \quad (23)$$

Where $\Lambda_i^{(m)}$ is defined as:

$$\Lambda_i^{(m)} = \begin{cases} \left(h_i^{(m)} 1^T - C \right) \alpha_i & m = M/2 \\ 0 & m \neq M/2 \end{cases} \quad (24)$$

Among them, the j -th element of α_i is $d_{ij} u_{ij}$. In Eq.(17), d_{ij} is described as follows:

$$d_{ij} = (1 + \sigma) \frac{\left\| \Phi \left(h_i^{(M/2)} \right) - \Phi \left(v_j \right) \right\|_2 + 2\sigma}{2 \left(\left\| \Phi \left(h_i^{(M/2)} \right) - \Phi \left(v_j \right) \right\|_2 + \sigma \right)^2} \quad (25)$$

Where, σ is a trade-off parameter used to control the degree of outliers of various types. $\tilde{d}_{ij}$ means filtering or median filtering for d_{ij} .

$$\tilde{d}_{ij} = (1 + \sigma) \frac{\left\| \Phi \left(\tilde{h}_i^{(M/2)} \right) - \Phi \left(v_j \right) \right\|_2 + 2\sigma}{2 \left(\left\| \Phi \left(\tilde{h}_i^{(M/2)} \right) - \Phi \left(v_j \right) \right\|_2 + \sigma \right)^2} \quad (26)$$

Where, $\tilde{h}_i^{(M/2)}$ represents the neighborhood mean or median value of the feature data output from the $M/2$ -th layer network. From Eq.(5), it can be seen that the objective function can be rewritten as follows.

$$\begin{aligned} \min J(U, V) = & \frac{1}{2} \sum_{i=1}^N \left\| h_i^{(M)} - x_i \right\|_2^2 + \lambda_1 \sum_{i=1}^N \sum_{j=1}^K d_{ij} u_{ij} \left[1 - K \left(h_i^{(M/2)}, v_j \right) \right] + \gamma u_{ij} \log u_{ij} \\ & + 2\alpha \sum_{i=1}^N \sum_{j=1}^K \tilde{d}_{ij} u_{ij} \left[1 - K \left(\tilde{h}_i^{(M/2)}, v_j \right) \right] + \frac{\lambda_2}{2} \sum_{m=1}^M \left\| W^{(m)} \right\|_2^2 + \left\| b^{(m)} \right\|_2^2 \end{aligned} \quad (27)$$

The objective function constrained optimization problem is solved by the Lagrangian multiplier method, and the Lagrangian function is:

$$L = \frac{1}{2} \sum_{i=1}^N \|h_i^{(M)} - x_i\|_2^2 + \sum_{i=1}^N \lambda_i \left(1 - \sum_{j=1}^K u_{ij}\right) + \gamma u_{ij} \log u_{ij} + \lambda_1 \sum_{i=1}^N \sum_{j=1}^K d_{ij} u_{ij} \left[1 - K\left(h_i^{(M/2)}, v_j\right)\right] + \frac{\lambda_2}{2} \sum_{m=1}^M \|W^{(m)}\|_2^2 + \|b^{(m)}\|_2^2 + 2\alpha \sum_{i=1}^N \sum_{j=1}^K \tilde{d}_{ij} u_{ij} \left[1 - K\left(\tilde{h}_i^{(M/2)}, v_j\right)\right] \quad (28)$$

Where, λ is the introduced Lagrange multiplier, find partial derivatives of the above formulas about u_{ij} and λ_i respectively, and make its value 0, which can be obtained:

$$\frac{\partial L}{\partial u_{ij}} = \lambda_1 d_{ij} \left[1 - K\left(h_i^{(M/2)}, v_j\right)\right] + 2\alpha \tilde{d}_{ij} \left[1 - K\left(\tilde{h}_i^{(M/2)}, v_j\right)\right] + \gamma (\log u_{ij} + 1) - \lambda_i = 0 \quad (29)$$

$$\frac{\partial L}{\partial \lambda_i} = 1 - \sum_{j=1}^K u_{ij} = 0 \quad (30)$$

Combining Eq.(29) with Eq.(30), the expression of u_{ij} is obtained as Eq.(31).

$$u_{ij} = \frac{\exp(-\gamma^{-1}(A_{ij} + B_{ij}))}{\sum_{j=1}^K \exp(-\gamma^{-1}(A_{ij} + B_{ij}))} \quad (31)$$

where, $A_{ij} = \lambda_1 d_{ij} \left(1 - K\left(h_i^{(M/2)}, v_j\right)\right)$, $B_{ij} = 2\alpha \tilde{d}_{ij} \left(1 - K\left(\tilde{h}_i^{(M/2)}, v_j\right)\right)$. According to Eq.(28), the partial derivative of the cluster center v_j can be obtained as follow.

$$\begin{aligned} \frac{\partial L}{\partial v_j} = & \lambda_1 \sum_{i=1}^N d_{ij} u_{ij} \left(h_i^{(M/2)} - v_j\right) K\left(h_i^{(M/2)}, v_j\right) \\ & + 2\alpha \sum_{i=1}^N \tilde{d}_{ij} u_{ij} \left(\tilde{h}_i^{(M/2)} - v_j\right) K\left(\tilde{h}_i^{(M/2)}, v_j\right) \end{aligned} \quad (32)$$

Making the value of it 0, the cluster center v_j can be obtained as:

$$v_j = \frac{\lambda_1 \sum_{i=1}^N C_{ij} h_i^{(M/2)} + 2\alpha \sum_{i=1}^N D_{ij} \tilde{h}_i^{(M/2)}}{\lambda_1 \sum_{i=1}^N C_{ij} + 2\alpha \sum_{i=1}^N D_{ij}} \quad (33)$$

Where, $C_{ij} = d_{ij} u_{ij} K\left(h_i^{(M/2)}, v_j\right)$, $D_{ij} = \tilde{d}_{ij} u_{ij} K\left(\tilde{h}_i^{(M/2)}, v_j\right)$.

Because the deep fuzzy clustering algorithm embedded with single neighborhood spatial information has some limitations. So, in order to meet the segmentation requirements of different noisy images, mean filtering neighborhood information and median filtering neighborhood information are simultaneously embedded into the objective function of the fuzzy clustering algorithm to improve the algorithm's segmentation performance. Although the segmentation effect has been improved, it is sensitive to strong noise and outliers. Therefore, on the basis of the improved objective function, the neighborhood information of membership degree is introduced, which makes the membership degree within classes tend to be consistent, thus achieving the purpose of improving the noise resistance and segmentation

accuracy of the algorithm. The improved objective function and constraints are defined as follows.

$$\begin{aligned}
 \min J(U, V) = & \frac{\lambda_1}{2} \sum_{i=1}^N \sum_{j=1}^K d_{ij} u_{ij} \left\| \Phi \left(h_i^{(M/2)} \right) - \Phi(v_j) \right\|_2^2 + \frac{\lambda_2}{2} \sum_{m=1}^M \left\| W^{(m)} \right\|_2^2 + \left\| b^{(m)} \right\|_2^2 \\
 & + \alpha \sum_{i=1}^N \sum_{j=1}^K \bar{d}_{ij} u_{ij} \left\| \Phi \left(\bar{h}_i^{(M/2)} \right) - \Phi(v_j) \right\|_2^2 + \frac{1}{2} \sum_{i=1}^N \left\| h_i^{(M)} - x_i \right\|_2^2 \\
 & + \beta \sum_{i=1}^N \sum_{j=1}^K \hat{d}_{ij} u_{ij} \left\| \Phi \left(\hat{h}_i^{(M/2)} \right) - \Phi(v_j) \right\|_2^2 + \gamma u_{ij} \log \frac{u_{ij}}{u_{\partial_{ij}}} \\
 \text{s.t. } & \sum_{j=1}^K u_{ij} = 1, u_{ij} \in [0, 1], 0 < \sum_{i=1}^N u_{ij} < 1, \sum_{j=1}^K u_{\partial_{ij}} = 1, 1 \leq i \leq N, 1 \leq j \leq K
 \end{aligned} \tag{34}$$

where, α and β are the control parameters of spatial information penalty, $\bar{d}_{ij}$ and $\hat{d}_{ij}$ are the mean and median neighborhood filtering information of d_{ij} , respectively. $u_{\partial_{ij}}$ is defined as follows.

$$u_{\partial_{ij}} = \sum_{i' \in \partial_i} \lambda_{ii'} u_{i'j} \tag{35}$$

Where, θ is a constant and $\lambda_{ii'}$ is defined as:

$$\lambda_{ii'} = \frac{\exp \left(- (2\theta^2)^{-1} (u_{ij} - u_{i'j})^2 \right)}{\sum_{i''} \exp \left(- (2\theta^2)^{-1} (u_{ij} - u_{i''j})^2 \right)} \tag{36}$$

Using Lagrange multiplier method to solve the optimization model corresponding to the above objective function, we have Eq.(37) and Eq.(38).

$$u_{ij} = \frac{u_{\partial_{ij}} \exp \left(-\gamma^{-1} (A_{ij} + B_{ij} + C_{ij}) \right)}{\sum_{j=1}^K u_{\partial_{i,j}} \exp \left(-\gamma^{-1} (A_{ij} + B_{ij} + C_{ij}) \right)} \tag{37}$$

Where, $A_{ij} = \lambda_1 d_{ij} \left(1 - K \left(h_i^{(M/2)}, v_j \right) \right)$, $B_{ij} = 2\alpha \bar{d}_{ij} \left(1 - K \left(\bar{h}_i^{(M/2)}, v_j \right) \right)$, $C_{ij} = 2\beta \hat{d}_{ij} \left(1 - K \left(\hat{h}_i^{(M/2)}, v_j \right) \right)$.

$$v_j = \frac{\lambda_1 \sum_{i=1}^N D_{ij} h_i^{(M/2)} + 2\alpha \sum_{i=1}^N E_{ij} \bar{h}_i^{(M/2)} + 2\beta \sum_{i=1}^N F_{ij} \hat{h}_i^{(M/2)}}{\lambda_1 \sum_{i=1}^N D_{ij} + 2\alpha \sum_{i=1}^N E_{ij} + 2\beta \sum_{i=1}^N F_{ij}} \tag{38}$$

Where, $D_{ij} = d_{ij} u_{ij} K \left(h_i^{(M/2)}, v_j \right)$, $E_{ij} = \bar{d}_{ij} u_{ij} K \left(\bar{h}_i^{(M/2)}, v_j \right)$, $F_{ij} = \hat{d}_{ij} u_{ij} K \left(\hat{h}_i^{(M/2)}, v_j \right)$. The outline of the proposed KDFKMS is shown in Algorithm 1.

5 Experimental results and discussion

In this paper, many clustering algorithms studied in the literature mentioned above are compared and analyzed comprehensively. We mainly introduces kernel function and neighborhood information into deep clustering algorithm. Therefore, we compare the KDFKMS algorithm with several classic fuzzy clustering algorithms that integrate image local spatial information, which include KFCM_S1, KFCM_S2, FGFCM, RPFCM, ILKFCM, FLICM, KFLICM and KWFLICM. In addition, we also compared the proposed algorithm with DFKM to analyze the effectiveness and robustness of the improved algorithm for segmentation of noisy images.

5.1 Evaluation Index of Performance

To objectively and quantitatively evaluate the effectiveness and robustness performance of algorithm. In this paper, accuracy (ACC), precision (PR), specificity (SP), sensitivity (SE), misclassification error (ME) and modified peak signal-to-noise ratio ($PSNR$) are used as quantitative evaluation indexes.

Algorithm 1 KDFKMS

Input: Image $X = \{x_1, x_2, \dots, x_n\}$, set cluster number K , convergence end threshold ε , maximum iteration number T_{max} , parameter $\lambda_1, \lambda_2, \alpha, \beta, \sigma, \gamma, \delta, \theta$ and maximum iteration number L of network training.

Output: The final membership values corresponding to all centers.

- 1: Initialize membership matrix $u_{ij}^{(1)}$ and set iteration times $t = 1$; Initialize the first layer network input node $h^{(0)} = X$; Initialize the weight matrix $W^{(m)}$ and the offset vector $b^{(m)}$, where $m = 1, 2, \dots, M/2, \dots, M$.
- 2: Train the neural network, update the weight matrix and bias vector by Eq.(19) and (20), and obtain the network deep feature data by Eq.(18), and take the middle layer $h_i^{M/2}$ of the automatic encoder network.
- 3: Median filtering and mean filtering are carried out on the middle layer data $h_i^{M/2}$, and the neighborhood window size of pixels is 3×3 .
- 4: using Eq.(37) to update the membership degree u_{ij}^{t+1}
- 5: using Eq.(38) to update the cluster center v_j^t
- 6: Update iteration times: $t + 1 \rightarrow t$
- 7: Stop if $\|v^{t+1} - v^t\| < \varepsilon$, otherwise set counter $t + 1 \rightarrow t$ and go to Step 4.

ME is defined as:

$$ME = \left(1 - \sum_{j=1}^c |A_j \cap B_j| / |B_j| \right) \times 100\% \quad (39)$$

In which, A_j is the number of pixels belonging to j -th class in the image obtained by algorithm, and c represents the number of pixels belonging to j -th classes in the ground truth image. c is the total number of image segmentation categories, and $m \times n$ is the size of the image. The smaller ME is, the segmentation result is closer to the ground truth image, and better the performance of clustering algorithm. On the contrary, the poorer performance of fuzzy clustering segmentation algorithm.

The peak signal-to-noise ratio is an important indicator for evaluating the quality of an image or signal. In the research work, the modified peak signal-to-noise ratio proposed by Guo et al. (2012) [24] is more suitable for evaluating the anti-noise ability of image segmentation algorithms, and is defined as follows:

$$PSNR = 10 \log \left(255^2 / MSE \right) \quad (40)$$

Where, $MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|I_{ij} - K_{ij}\|^2$, A is the mean square error between the ideal segmentation result of noiseless image and the actual segmentation result of noiseless image, I_{ij} is the pixel value of the noise less image at (i, j) , K_{ij} is the pixel value at (i, j) obtained by segmentation algorithm for image corrupted by noise, and $m \times n$ is the size of the image. The larger the value of $PSNR$, the better the segmentation performance and robustness of the algorithm; on the contrary, the worse the segmentation performance and robustness of the algorithm.

For ACC , PR , SE and SP , the larger the value, the better the segmentation performance of the algorithm. See [25]-[26] for its definition.

5.2 Test Results and Analysis

In the proposed algorithm, after a large number of experimental tests, the parameters are set to

$\lambda_1 = 0.1, \lambda_2 = 0.01, \alpha = 0.5, \beta = 0.5, \sigma = 0.1, \gamma = 0.01, \delta = 3.8, \theta = 3.8, L = 10^4$. In the experiment, the maximum iteration times T and the algorithm end threshold ε of all algorithms are set to 100 and 10^{-3} , respectively. Fuzzy weighted index of all algorithms $m = 2$. The restriction parameters of KFCM.S1 and KFCM.S2 are both set to 3.8 as suggested by Ahmed et al., setting $\lambda_s = 3, \lambda_g = 6$ in FGFCM and FGKFCM as suggested by Cai et al. It should be pointed out that for the local spatial information of pixels, the neighborhood window cannot be too large, otherwise, the segmentation results will become blurred, so the size of neighbor window is setting as 3×3 .

5.2.1 Segmentation Performance on Synthetic image

To begin with, we evaluate these algorithms with a background-free synthetic four-cluster image shown in Fig.2(a). Different levels of Gaussian noise, Salt and pepper noise and Rician noise are added to this image.

Fig.2 shows the segmentation results of different algorithms in Gaussian noisy synthetic image with normalized variance of 0.05, and segmentation results of the noisy synthetic image with Salt-and-pepper noise with intensity of 15% and Rician noisy image with standard deviation of 42 are given in Fig.3 and Fig.4 respectively. Table 1 shows the ACC , $PSNR$, ME , SE , SP , and PR indexes of the noisy images at different levels of each algorithm.

For the convenience later, DFKM, FGFCM, ILKFCM, RPFM, KFCM_S1, KFCM_S2, FLICM, KFLICM, KWFLICM and KDFKMS algorithms are denoted as AM1-AM12, respectively.

Figure 2 0.1 Gaussian-noised synthetic image. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Figure 3 15% Salt-and-pepper-noised synthetic image. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

As can be seen from Fig.2(c)-(k), the DFKM, FGFCM, ILKFCM, RPFM, KFCM_S1, KFCM_S2, FLICM, KFLICM and KWFLICM are affected by Gaussian noise to different extent. Among them, DFKM and RPFM have the worst effects under the influence of Gaussian noise, because the DFKM algorithm does not use any of the images spatial information, so a large amount of noise remains in the segmentation result. The FGFCM, ILKFCM, KFCM_S1, KFCM_S2, FLICM, KFLICM and KWFLICM is second, moreover, it is found from Fig.2(l) that the proposed KDFKMS algorithm can eliminate almost all the noise. For the Salt and pepper noisy image, it can be seen from Fig.3(f) that the RPFM has the worst effect and divides the four clusters of errors into three clusters; the DFKM, FGFCM, KFCM_S1 and FLICM algorithms segmentation results have more noise; although ILKFCM, KFCM_S2, KFLICM and KWFLICM can remove most of the noise, but the visual segmentation effect is slightly worse than the KDFKMS algorithm. With respect to Rician noisy images, it can be seen from fig. 4(d) and fig.4(g) that FGFCM and KFCM_S1 algorithms divide four types of errors into three types, Fig.4(f) shows that the segmentation results of RPFM have more noise points, while ILKFCM, KFCM_S2, FLICM and KFLICM have a few noise points, among which KWFLICM and KDFKMS have better segmentation performance, while KDFKMS has a little less error points. Furthermore, from Table 1 that KDFKMS can obtain a smaller value of ME , and a higher value of ACC , $PSNR$, SP , SE and PR , which indicates that the improved algorithm can achieve better segmentation accuracy and anti-noise performance. To test the robustness performance of each algorithm, different levels of Gaussian noise, Salt and pepper noise and Rician noise are added to Fig.2(a), the varying curves of ACC , ME , and $PSNR$ of different segmentation algorithms are shown as follows.

Figure 4 72 Rician-noised synthetic image. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM-S1; (h)KFCM-S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Table 1 Comparative the Index values of different segmentation algorithms

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
0.1GS	PSNR	14.1146	17.8086	17.7361	10.9620	17.3868	17.3912	17.9680	18.7719	20.4572	33.9952
	ME	9.4686	9.4256	8.6853	32.0642	11.0087	9.0042	10.8203	7.9566	4.7772	0.1157
	ACC	0.9527	0.9468	0.9505	0.8397	0.9389	0.9489	0.9459	0.9541	0.9700	0.9994
	SE	0.9053	0.9179	0.9253	0.6794	0.9021	0.9221	0.8918	0.9326	0.9644	0.9988
	SP	0.9684	0.9564	0.9589	0.8931	0.9512	0.9578	0.9639	0.9613	0.9719	0.9996
	PR	0.9053	0.8754	0.8824	0.6794	0.8603	0.8794	0.8918	0.8894	0.9197	0.9988
15%S&P	PSNR	12.9472	12.2155	21.3544	11.3526	16.4694	21.5794	14.5186	21.2803	22.3184	27.7567
	ME	14.4111	16.7823	2.8885	35.0073	11.6928	1.8078	18.3538	3.1975	1.7995	0.4065
	ACC	0.9279	0.9100	0.9795	0.8250	0.9355	0.9849	0.9082	0.9779	0.9849	0.9980
	SE	0.8559	0.8443	0.9833	0.6499	0.8952	0.9941	0.8165	0.9802	0.9942	0.9959
	SP	0.9520	0.9319	0.9782	0.8833	0.9489	0.9818	0.9388	0.9772	0.9819	0.9986
	PR	0.8559	0.8052	0.9377	0.6499	0.8537	0.9480	0.8165	0.9348	0.9481	0.9959
72Rician	PSNR	14.1993	15.2099	17.5486	10.1767	13.9481	16.9453	20.0107	18.9931	21.4370	35.6662
	ME	8.7828	25.5866	5.1491	32.9566	36.7721	7.8293	3.7709	4.1179	2.1631	0.0876
	ACC	0.9561	0.8660	0.9682	0.8352	0.8101	0.9548	0.9811	0.9733	0.9831	0.9996
	SE	0.9122	0.7563	0.9607	0.6704	0.6444	0.9339	0.9623	0.9710	0.9905	0.9991
	SP	0.9707	0.9026	0.9707	0.8901	0.8653	0.9618	0.9874	0.9741	0.9806	0.9997
	PR	0.9122	0.7212	0.9161	0.6704	0.6146	0.8906	0.9623	0.9260	0.9446	0.9991

* 0.1 GS denotes Gaussian noise with normalized variance of 0.1; 15% S&P denotes Salt and pepper noise with intensity of 15%; 72 Rician denotes Rician noise with standard deviation of 72.

Figure 5 Change curve of ACC,ME and PSNR of Gaussian noise image

From Fig.5, with the increase of Gaussian noise, the *PSNR* and *ACC* of each algorithm decline, and the *ME* index increases. Form Fig.6, when the intensity of Salt and pepper noise is in the range of 5%-20%, as the noise increases, the *PSNR* of the *FGFCM* does not decrease but increases, and the *ME* does not increase but decreases. The reason is that *FGFCM* is too sensitive to Salt and pepper noise,

Figure 6 Change curve of ACC,ME and PSNR of Salt and pepper noise image

Figure 7 Change curve of ACC,ME and PSNR of Rician noise image

which leads to the segmented image retaining a large number of amplified Salt and pepper noise points. And from Fig.7, the ACC and ME of KFCM.S1, KFCM.S2, FLICM and DFKM algorithms jump, which is due to the misclassification, that is, the four categories are divided into three categories, so the indexes become unstable.

However, the algorithm in this paper always obtains the highest ACC and $PSNR$ and the highest ME . Each index changes steadily with the influence of noise. Therefore, the improved algorithm has stronger robustness and the ability to suppress noise interference.

As shown in Fig.8(a), we used the synthetic four-cluster image with background to quantify the performance of the segmentation algorithms. Fig.8 shows the segmentation results of the algorithm on Gaussian noisy images with normalized variance of 0.05.

Fig.9 shows the segmentation result of the algorithm on the noisy image of Salt and pepper with intensity of 10%. Fig.10 shows the segmentation result of the algorithm on the Rician noisy image with standard deviation of 42. The testing indexes are shown in Table 2. It can be seen from fig.8(k) that the segmentation effect of KWFLICM algorithm is better than FGFCM, ILKFCM, RPFM, KFCM.S1, KFCM.S2, FLICM and KFLICM, but there are still a large number of noisy pixel blocks, and DFKM has relatively few noise points; It can be seen from fig.8(l) that the segmentation result of KDFKMS has almost no noise points, and retains the details and contour information of the image, which is closer to the original segmentation image and achieves better segmentation performance.

For Salt and pepper noise images in Fig.9, the segmentation results of DFKM, FGFCM, ILKFCM, RPFM, KFCM.S1, FLICM, KFLICM and KWFLICM all contain more noise points. KFCM.S2 introduces pixel neighborhood median information, which enhances the algorithm's ability to suppress Salt and pepper noise to a certain extent, but its ability is limited, and the segmentation result still has a small amount of noise; However, the algorithm in this paper makes full use of the spatial neighborhood information of the image, which can suppress the noise and retain the details and texture information in the image.

From Fig.10, the segmentation results of FGFCM, ILKFCM, RPFM, KFCM.S1 and KFCM.S2 still contain more Rician noise, which can not effectively segment image objects. Although DFKM, FLICM, KFLICM and KWFLICM can segment the target well, there are still many noise points in the background area; But proposed algorithm can effectively segment the target and suppress the background noise, so this algorithm has certain advantages for Rician noisy image segmentation.

Test indexes of different algorithms are given in table 2. the data show that KDFKMS has smaller ME , and the test indexes of ACC , SE , SP , PR and $PSNR$ are larger than those of other algorithms. From various test indexes, we can further objectively judge the superior segmentation performance and robustness of the improved algorithm.

Similarly, different levels of Gaussian noise, Salt and pepper noise and Rician noise are added to Fig.8(a), the varying curves of SE , SP and PR of different segmentation algorithms are shown as follows.

It can be seen from Fig.11 that compared with other algorithms, the proposed algorithm obtains the highest SE , SP , and PR . Each index decreases steadily with the change of noise intensity. From Fig.12, the algorithm in this paper and the KFCM.S2 have stronger

Figure 8 0.05 Gaussian-noised synthetic image. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Figure 9 10% Salt-and-pepper-noised synthetic image. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

anti-noise robustness. It can be seen from Fig.13 that the SP , SE and PR of the algorithm in this paper and the DFKM are higher than other algorithms, these two algorithms have stronger anti-noise robustness against Rician noise, but with the increase of noise, each index of DFKM drops faster, which shows that DFKM is more sensitive to strong noise than proposed algorithm.

Next, in order to further evaluate the performance of the algorithm, mixed noises include Gaussian noise with normalized variance of 0.05 and Salt and pepper noise with intensity of 10%; Gaussian noise with normalized variance of 0.1 and Rician noise with standard deviation of 32, Speckle noise with normalized variance of 0.3 and Salt and pepper noise with intensity of 10% are added to three-cluster synthetic image in Fig.14(a). The segmentation results of the algorithm in different mixed noise images are shown in Fig.14, Fig.15 and Fig.16, respectively. Table 3 shows testing indexes. It can be seen from Fig.14 that KWFLICM and the proposed algorithm perform well in segmentation results, and the segmentation results contain fewer noise points; DFKM and RPFCM have the worst segmentation results; FGFCM, ILKFCM, KFCM_S1, KFCM_S2, FLICM and KFLICM suppress the influence of noise to a certain extent, but the segmentation results still have more noise points. Fig. 15 shows that DFKM and RPFCM algorithms are very poor in segmentation results, followed by FGFCM, ILKFCM, KFCM_S1, KFCM_S2 and KFLICM. The FLICM and KWFLICM have good segmentation results, and the KDFKMS algorithm has fewer noise points. It can be seen from Fig.16 that the proposed algorithm is slightly inferior to KWFLICM in visual effect, but superior to other algorithms in processing images with 0.3 Speckle noise and 10% Salt and pepper noise. According to table 3, the index of this algorithm is slightly inferior to KWFLICM algorithm except ME and SE index, and most indexes of noisy images are better than other algorithms. Therefore, from the objective point of view of the test data, it further shows that KDFKMS algorithm has better segmentation performance and anti-noise interference ability.

Figure 10 42 Rician noise synthetic image. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM-S1; (h)KFCM-S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Table 2 Comparative the Index values of different segmentation algorithms

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
0.05 GS	PSNR	23.8767	22.6340	22.1348	18.4297	21.4076	21.7279	22.3549	24.8199	27.8807	37.6240
	ME	4.3450	11.1525	14.3300	26.8500	16.2575	17.6275	11.9250	9.3525	4.5275	0.0750
	ACC	0.9783	0.9293	0.9134	0.8658	0.9038	0.8969	0.9404	0.9383	0.9624	0.9996
	SE	0.9566	0.8885	0.8567	0.7315	0.8374	0.8237	0.8808	0.9065	0.9547	0.9992
	SP	0.9855	0.9429	0.9323	0.9105	0.9259	0.9213	0.9603	0.9489	0.9650	0.9998
10%S&P	PR	0.9566	0.8384	0.8084	0.7315	0.7902	0.7773	0.8808	0.8554	0.9009	0.9992
	PSNR	9.5112	14.4170	18.9927	17.0156	20.2571	30.1249	18.4317	23.0980	24.0669	31.9409
	ME	32.0600	17.8775	17.4100	17.7975	19.9725	1.1125	30.1650	13.5475	11.0125	0.4025
	ACC	0.8397	0.8957	0.8980	0.9110	0.8852	0.9795	0.8492	0.9173	0.9300	0.9980
	SE	0.6794	0.8212	0.8259	0.8220	0.8003	0.9889	0.6984	0.8645	0.8899	0.9960
42 Rician	SP	0.8931	0.9205	0.9221	0.9407	0.9135	0.9764	0.8994	0.9349	0.9434	0.9987
	PR	0.6794	0.7750	0.7794	0.8220	0.7552	0.9332	0.6984	0.8158	0.8397	0.9960
	PSNR	30.8280	16.6100	21.3798	20.1104	22.2305	22.6271	23.7062	29.1432	32.7420	40.8528
	ME	0.8850	10.0475	7.9925	27.1625	12.8200	16.5575	7.2050	3.5025	1.3375	0.0275
	ACC	0.9956	0.9448	0.9451	0.8642	0.9210	0.9023	0.9640	0.9676	0.9784	0.9999
42 Rician	SE	0.9911	0.9194	0.9201	0.7284	0.8718	0.8344	0.9280	0.9650	0.9866	0.9997
	SP	0.9970	0.9532	0.9535	0.9095	0.9374	0.9249	0.9760	0.9684	0.9756	0.9999
	PR	0.9911	0.8676	0.8682	0.7284	0.8227	0.7874	0.9280	0.9106	0.9310	0.9997

* 0.1 GS denotes Gaussian noise with normalized variance of 0.1; 15% S&P denotes Salt and pepper noise with intensity of 15%; 72 Rician denotes Rician noise with standard deviation of 72.

Figure 11 Change curve of SE,SP and PR of Gaussian noise image

5.2.2 Segmentation Performance on real-world images

Next, we use real-world images from the Berkeley segmentation dataset and benchmark (BSDS500) (2011) [27] to further evaluate the segmentation effectiveness and anti-noise robustness of the proposed algorithm and the others algorithm. Mixed noise, Gaussian noise,

Salt and pepper noise, Rician noise and Speckle noise with different noise levels are added to the image, and the segmentation indexes of each algorithm are recorded and analyzed. Fig.17 and Fig.18 show the segmentation results of mixed noise include Gaussian noise with normalized variance of 0.1 and Salt and pepper noise with intensity of 4% images, test indexes are given in table 4. Fig.19 and Fig.20 show the segmentation results of Gaussian noise images with normalized variance of 0.1 and 0.03 respectively, and the indexes are given in Table 5. The segmentation result of Salt and pepper noise with noise intensity of 30% images are given in Figs.21 and 22, and the indexes are given in Table 6. The segmentation result of Rician noise with standard deviation of 82 images are given in Figs.23 and 24, and the

Figure 15 Synthetic image with 0.1 Gaussian noise & 32 Rician noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM; (k) KWFLICM; (l) KDFKMS.

Figure 16 Synthetic image with 0.3 Speckle noise & 10% Salt-and-pepper noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM; (k) KWFLICM; (l) KDFKMS.

indexes are given in Table 7. Fig. 25 and Fig. 26 show the segmentation results of Speckle noise images with normalized variance of 0.2 and 0.3 respectively, and the indexes are given in Table 8.

Table 3 Comparative the Index values of different segmentation algorithms.

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
0.05GS & 10%S&P	<i>PSNR</i>	14.7549	21.8467	22.0925	13.6893	21.3400	21.9224	22.8110	23.6952	24.1372	30.1358
	<i>ME</i>	22.5510	3.8971	2.8519	27.1118	4.7302	3.5568	5.0797	1.5869	1.2070	0.8163
	<i>ACC</i>	0.8497	0.9704	0.9773	0.8193	0.9648	0.9726	0.9661	0.9858	0.9883	0.9946
	<i>SE</i>	0.7745	0.9711	0.9816	0.7289	0.9628	0.9745	0.9492	0.9942	0.9980	0.9918
	<i>SP</i>	0.8872	0.9700	0.9752	0.8644	0.9658	0.9717	0.9746	0.9816	0.9835	0.9959
	<i>PR</i>	0.7745	0.9418	0.9520	0.7289	0.9337	0.9451	0.9492	0.9642	0.9679	0.9918
0.1GS & 32Rician	<i>PSNR</i>	14.0772	21.6137	20.5752	14.1079	21.3625	20.3296	25.1472	21.8529	23.6261	26.6320
	<i>ME</i>	26.1856	4.1153	5.5725	27.9709	4.4800	6.0883	2.6855	3.7704	1.7136	1.9409
	<i>ACC</i>	0.8254	0.9689	0.9592	0.8135	0.9665	0.9558	0.9821	0.9712	0.9849	0.9871
	<i>SE</i>	0.7381	0.9689	0.9544	0.7203	0.9653	0.9492	0.9731	0.9724	0.9930	0.9806
	<i>SP</i>	0.8691	0.9689	0.9616	0.8601	0.9671	0.9590	0.9866	0.9706	0.9809	0.9903
	<i>PR</i>	0.7381	0.9397	0.9256	0.7203	0.9362	0.9206	0.9731	0.9430	0.9630	0.9806
0.3Speckle & 10%S&P	<i>PSNR</i>	12.6312	20.9956	18.5395	13.4649	20.8402	18.7326	23.7214	20.3706	23.2540	25.6069
	<i>ME</i>	31.5521	4.1092	6.6315	27.8595	4.4128	7.4585	3.4103	4.6326	1.7578	1.7960
	<i>ACC</i>	0.7897	0.9690	0.9521	0.8143	0.9669	0.9466	0.9773	0.9655	0.9846	0.9880
	<i>SE</i>	0.6845	0.9690	0.9438	0.7214	0.9660	0.9355	0.9659	0.9638	0.9925	0.9820
	<i>SP</i>	0.8422	0.9689	0.9563	0.8607	0.9674	0.9522	0.9829	0.9663	0.9807	0.9910
	<i>PR</i>	0.6845	0.9398	0.9153	0.7214	0.9368	0.9073	0.9659	0.9347	0.9626	0.9820

* 0.05GS denotes Gaussian noise with normalized variance of 0.05; 32Rician denotes Rician noise with standard deviation of 32; 0.3Speckle denotes Speckle noise with normalized variance of 0.3

Figure 17 0.1 Gaussian-noised & 4% Salt-and-pepper noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM; (k) KWFLICM; (l) KDFKMS.

Figure 18 0.1 Gaussian-noised & 4% Salt-and-pepper noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM; (k) KWFLICM; (l) KDFKMS.

Table 4 Comparative the Index values of different segmentation algorithms on mixed noise image.

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
#118035 (0.1GS,4%S&P)	<i>ACC</i>	0.8006	0.8434	0.8106	0.7477	0.8478	0.8368	0.8736	0.8738	0.9057	0.9641
	<i>SE</i>	0.7008	0.7755	0.7263	0.6215	0.7820	0.7656	0.8104	0.8211	0.8689	0.9462
	<i>SP</i>	0.8504	0.8774	0.8528	0.8107	0.8806	0.8724	0.9052	0.9002	0.9241	0.9731
	<i>PR</i>	0.7008	0.7598	0.7115	0.6215	0.7661	0.7501	0.8104	0.8045	0.8512	0.9462
	<i>PSNR</i>	6.9648	6.7677	5.8954	5.4632	6.9126	6.6272	7.6045	7.7920	9.1893	14.1243
	<i>ME</i>	29.9163	22.8600	27.7848	37.8514	22.2103	23.8483	18.9623	18.2972	13.5219	5.3782
#253036 (0.1GS,4%S&P)	<i>ACC</i>	0.8577	0.9432	0.9484	0.7868	0.9257	0.9120	0.9544	0.9534	0.9697	0.9885
	<i>SE</i>	0.8577	0.9484	0.9536	0.7868	0.9309	0.9171	0.9544	0.9586	0.9749	0.9885
	<i>SP</i>	0.8577	0.9380	0.9433	0.7868	0.9206	0.9068	0.9544	0.9482	0.9645	0.9885
	<i>PR</i>	0.8577	0.9387	0.9438	0.7868	0.9214	0.9077	0.9544	0.9487	0.9649	0.9885
	<i>PSNR</i>	8.6236	12.8736	13.3368	6.8092	11.6065	10.8164	13.8594	13.8261	16.0038	21.6870
	<i>ME</i>	13.7292	5.1599	4.6379	20.8490	6.9080	8.2862	4.1120	4.1438	2.5097	0.6781

* 0.1GS denotes Gaussian noise with normalized variance of 0.1; 4%S&P denotes Salt and pepper noise with intensity of 4%

Figure 19 0.1 Gaussian noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS. **Figure 20** Real-world image with 0.03 Gaussian noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Table 5 Comparative the Index values of different segmentation algorithms on Gaussian noise image.

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
#296059 (0.1GS)	<i>ACC</i>	0.7001	0.7388	0.6816	0.6588	0.7441	0.7323	0.7189	0.7288	0.7231	0.8412
	<i>SE</i>	0.5502	0.6185	0.5328	0.4882	0.6265	0.6088	0.5784	0.6036	0.5950	0.7618
	<i>SP</i>	0.7751	0.7989	0.7561	0.7441	0.8029	0.7940	0.7892	0.7914	0.7871	0.8809
	<i>PR</i>	0.5502	0.6059	0.5220	0.4882	0.6138	0.5964	0.5784	0.5913	0.5829	0.7618
	<i>PSNR</i>	6.8474	9.7083	8.4585	7.4788	9.7047	9.3708	9.6883	9.5924	9.6050	12.0123
#245051 (0.03GS)	<i>ME</i>	44.9816	39.0380	47.6040	51.1752	38.2329	40.0101	42.1623	40.5295	41.3896	23.8230
	<i>ACC</i>	0.8176	0.6979	0.6779	0.7431	0.8096	0.8079	0.6976	0.7533	0.7101	0.9221
	<i>SE</i>	0.7265	0.5572	0.5273	0.6147	0.7247	0.7223	0.5463	0.6403	0.5755	0.8832
	<i>SP</i>	0.8632	0.7682	0.7533	0.8074	0.8520	0.8508	0.7732	0.8098	0.7774	0.9416
	<i>PR</i>	0.7265	0.5459	0.5166	0.6147	0.7100	0.7076	0.5463	0.6273	0.5639	0.8832
	<i>PSNR</i>	10.5085	8.4563	8.0584	9.2564	10.6525	10.5282	8.5789	9.3547	8.5603	14.0185
	<i>ME</i>	27.3541	45.2180	48.2089	38.5295	28.4648	28.7103	45.3656	36.9058	43.3812	11.6793

* 0.1GS denotes Gaussian noise with normalized variance of 0.1

Figure 21 Real-world image with 30% Salt-and-pepper noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Table 6 Comparative the Index values of different segmentation algorithms on Salt-and-pepper noise image

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
#35008 (30%S&P)	ACC	0.8116	0.7634	0.7610	0.7130	0.6682	0.6785	0.8709	0.8252	0.7829	0.9301
	SE	0.8116	0.7686	0.7662	0.7130	0.6734	0.6836	0.8709	0.8304	0.7881	0.9301
	SP	0.8116	0.7582	0.7558	0.7130	0.6630	0.6733	0.8709	0.8200	0.7778	0.9301
	PR	0.8116	0.7607	0.7583	0.7130	0.6665	0.6766	0.8709	0.8219	0.7800	0.9301
	PSNR	7.2790	6.3526	6.3083	5.4576	4.8573	4.9959	8.9695	7.7008	6.7357	11.6024
	ME	18.7110	23.1598	23.3975	28.4603	32.6792	31.6526	12.6780	16.9792	21.2045	6.9145
#372019 (30%S&P)	ACC	0.7088	0.7228	0.7259	0.6511	0.7099	0.7273	0.7028	0.7402	0.7496	0.8966
	SE	0.5632	0.5946	0.5992	0.4767	0.5753	0.6014	0.5542	0.6207	0.6348	0.8450
	SP	0.7816	0.7869	0.7893	0.7383	0.7773	0.7903	0.7771	0.8000	0.8070	0.9225
	PR	0.5632	0.5825	0.5871	0.4767	0.5636	0.5892	0.5542	0.6081	0.6219	0.8450
	PSNR	7.0158	8.7388	8.6617	6.7148	8.1529	8.5599	8.8970	9.0414	9.3601	10.8897
	ME	43.6759	41.3119	40.8449	52.3339	43.2426	40.6325	44.5820	38.6999	37.2893	15.5038

* 30%S&P denotes Salt-and-pepper noise with intensity of 30%

Table 7 Comparative the Index values of different segmentation algorithms on Rician noise image

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
#3063 (82)	ACC	0.7399	0.9182	0.9241	0.7713	0.9104	0.8956	0.9403	0.9293	0.9425	0.9492
	SE	0.6098	0.8876	0.8965	0.6570	0.8759	0.8537	0.9105	0.9044	0.9137	0.9342
	SP	0.8049	0.9334	0.9379	0.8285	0.9276	0.9165	0.9553	0.9418	0.9567	0.9569
	PR	0.6098	0.8696	0.8783	0.6570	0.8581	0.8364	0.9105	0.8860	0.9137	0.9152
	PSNR	9.1103	14.5268	14.8315	10.2915	14.2329	13.6372	16.3910	15.1318	16.3453	16.5001
	ME	39.0211	12.2726	11.3879	34.2984	13.4455	15.6631	8.9488	10.5971	8.6282	7.6139
#86000 (82)	ACC	0.7311	0.7594	0.7131	0.6980	0.7123	0.7128	0.6680	0.7063	0.7126	0.8444
	SE	0.4622	0.5344	0.4417	0.3959	0.4401	0.4412	0.3360	0.4281	0.4406	0.6889
	SP	0.8207	0.8344	0.8036	0.7986	0.8030	0.8034	0.7787	0.7990	0.8032	0.8963
	PR	0.4622	0.5183	0.4284	0.3959	0.4268	0.4279	0.3360	0.4152	0.4274	0.6889
	PSNR	6.1769	5.1986	6.3516	5.8835	6.1657	6.1133	6.3490	6.3708	6.5619	9.1853
	ME	53.7775	47.5735	56.2140	60.4057	56.3818	56.2691	66.3979	57.5754	56.3235	31.1125

* The contents in parentheses denote the standard deviation of Rician noise

Figure 23 Real-world image with 82 Rician noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Figure 24 Real-world image with 82 Rician noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFCM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Figure 25 Real-world image with 0.2 Speckle noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Figure 26 Real-world image with 0.3 Speckle noise. (a)Original image; (b)Noisy image; (c)DFKM; (d)FGFCM; (e)ILKFCM; (f)RPFM; (g)KFCM_S1; (h)KFCM_S2; (i)FLICM; (j)KFLICM;(k) KWFLICM;(l) KDFKMS.

Table 8 Comparative the Index values of different segmentation algorithms on Speckle noise image

Noisy image	Index	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10
#36046 (0.2)	ACC	0.7588	0.8620	0.8615	0.7000	0.8337	0.7902	0.8906	0.8598	0.8980	0.9373
	SE	0.6382	0.8034	0.8026	0.5499	0.7610	0.6956	0.8359	0.8000	0.8573	0.9059
	SP	0.8191	0.8913	0.8910	0.7750	0.8701	0.8374	0.9179	0.8896	0.9183	0.9530
	PR	0.6382	0.7871	0.7863	0.5499	0.7455	0.6815	0.8359	0.7838	0.8399	0.9059
	PSNR	9.3451	12.6341	12.5839	8.8850	11.8845	10.8970	13.8675	12.5492	13.8706	16.1852
	ME	36.1824	20.6689	20.7479	45.0062	24.9156	31.4519	16.4112	21.0109	15.2790	9.4054
#344010 (0.3)	ACC	0.7956	0.9101	0.8708	0.7573	0.9035	0.8728	0.9357	0.9084	0.9142	0.9510
	SE	0.6934	0.8756	0.8165	0.6360	0.8656	0.8196	0.9035	0.8729	0.8817	0.9265
	SP	0.8467	0.9274	0.8979	0.8180	0.9224	0.8994	0.9517	0.9261	0.9305	0.9633
	PR	0.6934	0.8578	0.8000	0.6360	0.8480	0.8029	0.9035	0.8552	0.8638	0.9265
	PSNR	10.6776	14.8522	13.2346	9.9415	14.5482	13.3118	16.1733	14.7805	15.1223	17.4006
	ME	30.6604	12.7648	18.6696	36.3987	13.7668	18.3658	9.6508	13.0304	12.1573	7.3458

* The contents in parentheses denote the normalized variance of Speckle noise

Figs.17-26 show that the proposed algorithm in this paper has better segmentation performance than other algorithms on Gaussian noise images, Salt and pepper noise images, mixed noise images, Speckle noise images, and Rician noise images. It contains less misclassified pixels caused by noise, and can well preserve image details. Therefore, compared with other algorithms, the improved algorithm has more segmentation advantages. At the same time, the segmentation indexes provided in Table 4-8 further illustrate this point.

5.3 Complexity Analysis and Test of the Proposed Algorithm

The algorithm in this paper is mainly divided into two parts: network training and clustering algorithm. Because this algorithm takes the network layer data after training and then goes through clustering algorithm, so the computational complexity of the algorithm is mainly

composed of clustering algorithm. The complexity of clustering algorithm mainly consists of two parts, one part comes from the calculation of local spatial information of each pixel, and the other part comes from the iteration of the algorithm. In order to compare the efficiency of the algorithm proposed in this paper with that of the algorithm, the computational complexity of different algorithms is analyzed as follows. Suppose the image has N pixels, K is the number of clusters, T is the number of iteration steps, w is the length of local window, then the computational complexity of KFCM.S1 or KFCM.S2 algorithm mainly consists of two parts, one part comes from the computational complexity $O(N \times w^2)$ of the mean (or median) information of each pixel, and the other part is the algorithm iteration complexity $O(N \times K \times T)$; the computational complexity of FGFCM is mainly composed of two parts, namely the computational complexity $O(N \times w^2)$ of nonlinear weighting and filtered image and the complexity of algorithm iteration $O(Q \times K \times T)$. It is well-known that the computational complexity of the FLICM is $O(N \times K \times w^2 \times T)$; for the KFLICM, have the same computational complexity $O(N \times K \times w^2 \times T)$; the computational complexity of the KWFLICM is $O(N \times [(2w^2) + 5w^2] \times K \times T)$. The computational complexity of the DFKM is $O(N \times K \times T)$; the computational complexity of the ILKFCM is $O(N \times [2w^4 + w^2] \times K \times T)$. The computational complexity of the algorithm in this paper is mainly composed of three parts, which are the computational complexity of the mean information of all pixels, the computational complexity of the median information of all pixels, and the computational complexity of the algorithm iteration, which is $O(N \times K \times w^2 \times T + N \times w^2 + N \times w^2)$.

6 Conclusion

Deep fuzzy clustering algorithm has become a research hotspot in the field of computer vision in recent years. Although deep fuzzy clustering algorithm has achieved good results in data set classification, but few people focus on image segmentation and consider its anti-noise robustness. In this paper, the neighborhood information of membership degree is introduced into the entropy regularization penalty of the deep fuzzy K-means algorithm, and the dual neighborhood information of pixels is introduced into the objective function of the deep fuzzy K-means algorithm, and its Euclidean distance is extended to the kernel space. The results of image segmentation test show that the proposed algorithm has better segmentation performance and anti-noise robustness. Although the research has achieved some results, there are still some challenges in many aspects. For example, after the introduction of neural networks, the time overhead is large. In addition, it needs further improvement in the trade-off algorithm's anti-noise performance and preservation of image details.

Abbreviations

KDFKMS: a dual-neighborhood information constrained deep fuzzy clustering based on kernel function; FCM: fuzzy C-means clustering algorithm; DFKM: A deep fuzzy K-means clustering algorithm model with adaptive loss function and entropy regularization; KFCM: kernel fuzzy c-means clustering algorithm; PSNR: peak signal-to-noise ratio; ME: misclassification error; SE: sensitivity.

Acknowledgements

None.

Authors' contributions

Lujia Lei: Acquisition of data, Analysis and interpretation of data, Writing - original draft, Writing - review & editing. Chengmao Wu: Conception and design of study, Review & editing. Xiaoping Tian: Review & editing.

Funding

This work was supported in part by the National Natural Science Foundation of China (61671377), the Shaanxi Natural Science Foundation of China (2021JM-459). The authors are thankful to the anonymous reviewers for their constructive suggestions to improve the overall quality of this paper.

The funders had no roles in the design of the study, the collection, analysis, and interpretation of data, and the writing of the manuscript.

Availability of data and materials

The data sets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Text for this section.

Author details

¹School of Electronics and Engineering, Xi'an University of Posts & Telecommunications, 710121 Xi'an, China. ²School of Electronics and Engineering, Xi'an University of Posts & Telecommunications, 710121 Xi'an, China. ³School of Electronics and Engineering, Xi'an University of Posts & Telecommunications, 710121 Xi'an, China.

References

- Bezdek, J. C., Ehrlich, R., & Full, W. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2–3), 191–203. (1984) [https://doi.org/10.1016/0098-3004\(84\)90020-7](https://doi.org/10.1016/0098-3004(84)90020-7)
- Yang, M., & Tsai, H. A Gaussian kernel-based fuzzy c-means algorithm with a spatial bias correction[J]. *Pattern Recognition Letters*, 29:1713–1725. (2008) <https://doi.org/10.1016/j.patrec.2008.04.016>

3. M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," in IEEE Transactions on Medical Imaging, 21(3):193-199.(2002) <https://doi.org/10.1109/42.996338>.
4. W. Cai, S. Chen, and D. Zhang. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation, Pattern Recognition, 40 (3) :825-838.(2007) <https://doi.org/10.1016/j.patcog.2006.07.011>
5. S. Chen, D. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Transactions on Systems, 8(4): 1907-1916.(2004) <https://doi.org/10.1109/tsmcb.2004.831165>
6. S. Chen, D. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),34:1907-1916.(2004) <https://doi.org/10.1109/tsmcb.2004.831165>
7. S. Krinidis, V. Chatzis, A robust fuzzy local information C-means clustering algorithm, IEEE Transactions on image processing, 19(5):1328-1337. (2010) <https://doi.org/10.1109/TIP.2010.2040763>
8. L. Hou, (2016) Study on image segmentation based on spatial information constraint clustering algorithm. Dissertation, East China Normal University.
9. M. Gong, Z. Zhou, J. Ma. Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering, IEEE Transactions on Image Processing, 21 (4) :2141-2151.(2011) <https://doi.org/10.1109/TIP.2011.2170702>
10. Hinton, G. E., Osindero, S. & Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets.?Neural Computation, 18, 1527-1554.
11. Bengio, Y. Learning deep architectures for ai. Foundations and trends R in Machine Learning 2(1):1-127.(2009) <https://doi.org/10.1561/22000000006>
12. Lcun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278-2324.(1998)
13. Yeganejou, M., & Dick, S. Classification via Deep Fuzzy c-Means Clustering.?2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1-6.(2018)<https://doi.org/10.1109/FUZZ-IEEE.2018.8491461>
14. Zijuan Zhao, Juanjuan Zhao, Kai Song, Akbar Hussain, Qianqian Du, Yunyun Dong, Jihua Liu, Xiaotang Yang, Joint DBN and Fuzzy C-Means unsupervised deep clustering for lung cancer patient stratification, Engineering Applications of Artificial Intelligence, 91:0952-1976,(2020) <https://doi.org/10.1016/j.engappai.2020.103571>
15. Caron, M., Bojanowski, P., Joulin, A. & Douze, M. Deep Clustering for Unsupervised Learning of Visual Features. Proceedings of the European Conference on Computer Vision (ECCV), September, 132-149.(2018).
16. J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in International Conference on Machine Learning, pp. 478-487.(2016).
17. Li F., Hong Q., Bo Z. Discriminatively boosted image clustering with fully convolutional auto-encoders Pattern Recognit., 83, pp.7161-173. (2018) <https://doi.org/10.1016/j.patcog.2018.05.019>
18. Yang, J., Parikh, D., Batra, D., Joint unsupervised learning of deep representations and image clusters. In: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 5147-5156.(2016)
19. Maziar Moradi Fard, Thibaut Thonet, Eric Gaussier, Deep k-Means: Jointly clustering with k-Means and learning representations, Pattern Recognition Letters, 138:185-192, ISSN 0167-8655.(2020)<https://doi.org/10.1016/j.patrec.2020.07.028>
20. R. Zhang, X. Li, H. Zhang and F. Nie, "Deep Fuzzy K-Means With Adaptive Loss and Entropy Regularization," in IEEE Transactions on Fuzzy Systems, vol. 28(11): 2814-2824.(2020)<https://doi.org/10.1109/TFUZZ.2019.2945232>.
21. Kiyotaka, H.I., & Honda, K. Fuzzy c-Means Clustering with Regularization by KL Information. (2001)
22. Gharieb, R.R., & Gendy, G. Fuzzy C-means with a local membership kl distance for medical image segmentation.?2014 Cairo International Biomedical Engineering Conference (CIBEC), 47-50.(2014).<https://doi.org/10.13176/11.605>
23. Zhao Quanhua, Jia Shuhan, Gao Jun, Gao Xin. Fuzzy clustering image segmentation combined with membership space constraint[J]. Science of Surveying and Mapping, 44(05):164-170.(2019)
24. Guo, Yanhui & Sengur, Abdulkadir. A Novel Color Image Segmentation Approach Based on Neutrosophic Set and Modified Fuzzy c-Means. Circuits, Systems, and Signal Processing. 32(4):1699 -1723. (2012).<https://doi.org/10.1007/s00034-012-9531-x>.
25. M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, T. Moriarty, A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data, IEEE Trans. Med. Imaging 21(3):193-199.(2002)
26. H. H. Chang, A. H. Zhuang, D. J. Valentino, et al., Performance measure characterization for evaluating neuroimage sSegmentation algorithms, NeuroImage 47(1):122-135.(2009)
27. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation" IEEE Trans. Pattern Anal. Mach. Intell., vol. 33(5):898-916.(2011) <https://doi.org/10.1109/TPAMI.2010.161>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Summary.docx](#)