Peak-Graph-Based Fast Density Peak Clustering for Image Segmentation

Junyi Guan, Sheng Li, Xiongxiong He, and Jiajia Chen

***Abstract*—Fuzzy c-means (FCM) algorithm as a traditional clustering algorithm for image segmentation cannot effectively preserve local spatial information of pixels, which leads to poor segmentation results with inconsistent regions. For the remedy, superpixel technologies are applied, but spatial information preservation highly relies on the quality of superpixels. Density peak clustering algorithm (DPC) can reconstruct spatial information of arbitrary-shaped clusters, but its high time complexity $O(n^2)$ and unrobust allocation strategy decrease its applicability for image segmentation. Herein, a fast density peak clustering method (PGDPC) based on the kNN distance matrix of data with time complexity $O(nlog(n))$ is proposed. By using the peak-graph-based allocation strategy, PGDPC is more robust in the reconstruction of spatial information of various complex-shaped clusters, so it can rapidly and accurately segment images into high-consistent segmentation regions. Experiments on synthetic datasets, real and Wireless Capsule Endoscopy (WCE) images demonstrate that PGDPC as a fast and robust clustering algorithm is applicable to image segmentation.**

***Index Terms*—image segmentation, density peak clustering, kNN, peak-graph.**

I. Introduction

IMAGE segmentation partitions image into non-overlapped and consistent regions, which is critical for image processing and computer vision. But it still faces challenges due to the multifariousness of image types, for which technologies like clustering [1], GraphCut [2], watershed transform [3], Markov random field [4], neural network [5], etc are selectable. Clustering method that automatically groups similar points into meaningful clusters is widely used in image segmentation due to its effectiveness and rapidity.

Fuzzy c-means (FCM) [6] is one of the most popular clustering algorithms for image segmentation due to its effectiveness in retaining original image information. However, FCM cannot effectively preserve the local spatial information of pixels since it only considers the spatial relationship between data points and cluster centers [7]. Numerous FCM-based methods are proposed [7], [8] to incorporate local spatial information, but the poor segmentation results with lots of low-consistent regions caused by their inefficiency in preserving local spatial information remains a problem. Lei *et al.* [9], [10] introduces a new superpixel technology to remedy this problem, yet, spatial information preservation highly relies on the quality of superpixels. Furthermore, all these FCM-based methods have a common limitation of over-dependence on the initial-setting of cluster center number. Such deficiencies of traditional FCM-based methods make room for other clustering methods for image segmentation.

Density peaks clustering (DPC) proposed by Rodriguez and Laio does well in reconstructing spatial information of data points due to its density-based allocation strategy and can easily find density peaks (hereafter peak) as cluster centers. This makes DPC a potential novel clustering method for image segmentation. Although DPC's clustering performance is remarkable, it still faces some shortcomings: 1) it requires high time complexity $O(n^2)$ to calculate the distance between data points; 2) its allocation strategy is not robust in reconstructing the spatial information of complex-shaped clusters. Even though many algorithms are proposed to improve DPC [13]–[24], they only focus on overcoming one or another shortcoming of DPC, in other words, speed-up methods do not add extra clustering accuracy, while allocation-improved methods run even slower.

With the intention to fully and extensively improve DPC, we propose a fast density peak clustering algorithm based on peak-graph (PGDPC).[1] With the kNN distance matrix of points, PGDPC can run about $O(nlog(n))$. We divide data points into peaks and non-peaks in advance and design a two-step allocation strategy: first, we directly assign non-peaks within their local areas by using DPC's allocation strategy; and then, we assign peaks along geodesic paths in a peak-graph which is constructed along allocation paths of non-peaks. The main contributions of PGDPC are: 1) PGDPC with time complexity $O(nlog(n))$ can handle large datasets rapidly; 2) our peak-graph-based allocation strategy can integrally reconstruct the spatial information of complex-shaped clusters; 3) PGDPC is found superior to some popular state-of-the-art methods when applied to real image segmentation and Wireless Capsule Endoscopy (WCE) image segmentation, which makes it an applicable alternative clustering method for image segmentation.

II. Method

A. DPC Algorithm and Its Problem Statement

1) DPC Algorithm: DPC calculates two quantities for each point i: the density $\rho_i : \rho_i = \sum_{i \neq j} exp(-(\frac{d_{ij}}{d_c})^2)$, and the

Manuscript received February 2, 2021; revised March 19, 2021; accepted April 5, 2021. Date of publication April 16, 2021; date of current version May 14, 2021. This work was supported in part by the National Science Foundation of P.R. China under Grant 61873239, and in part by Key R&D Program Projects in Zhejiang Province under Grant 2020C03074. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Le Lu *(Corresponding author: Sheng Li.)*

The authors are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: jonnyguan73@163.com; shengli@zjut.edu.cn; hxx@zjut.edu.cn; fl_katrina@163.com).

Digital Object Identifier 10.1109/LSP.2021.3072794

[1][Online]. Available: https://github.com/Guanjunyi/PGDPCforImageSegementation.git

distance $\delta_i : \delta_i = \min_{j:\rho_j>\rho_i}(d_{ij})$ from its nearest point with a higher density. Where d_{ij} is the Euclidean distance from point i to point j, while d_c is a user-specified cutoff distance. For the highest density point i, $\delta_i = \max_{i\neq j}(d_{ij})$.

By considering cluster centers as peaks characterized by a higher density ρ than their neighbors and by a relatively large distance δ from points with higher densities. Cluster centers are easily located by searching large ρ-δ points in a decision graph. After cluster centers are selected and labeled, according to DPC's allocation strategy, each non-center point is only allowed to inherit the label of its closest point with a higher density. Once each point obtains its label, clustering completes.

2) Problem Statement: As mentioned in Section I, besides high complexity, DPC's allocation strategy is not robust for complex-shaped clusters, because a complex-shaped cluster generally consists of multiple peaks. However, since peaks do not have a point with higher density in their local areas, according to DPC's allocation strategy, they will be assigned into clusters beyond their local areas, which leads to the loss of local spatial information of peaks. Nonetheless, points of different local areas are most likely not in the same cluster, so DPC may misassign peaks leading to poor clustering results. [22], [23] address this shortcoming by applying geodesic distance, however, since all points need to be assigned along the geodesic paths, these methods are more time-consuming than DPC.

Therefore, in order to overcome the allocation limitation of peaks while speeding up DPC, we propose a fast method with complexity $O(nlog(n))$, called PGDPC, which only assigns peaks along geodesic paths after constructing a peak-graph.

B. Proposed Method

PGDPC performs clustering in four steps: (i) search for peaks of dataset; (ii) assign non-peaks within local area; (iii) assign peaks in a peak-graph; (iv) select cluster centers among peaks to complete clustering.

1) Search of Peaks: For each point i, we set i and its k nearest neighbors $N_k(i) = \{i^1, i^2, \ldots i^k\}$ as its local area, where i^k indicates the kth nearest neighbor of point i. In order not to add extra complexity, we use kNN density [18] as:

$$\rho_i' = \exp\left(-\frac{1}{k}\sum_{j\in N_k(i)}(d_{ij})^2\right) \tag{1}$$

Since peaks are density maximum in their local areas, we define peak and non-peak as:

Definition 1: if $\forall j \in N_k(i)$ satisfies that $\rho_j' < \rho_i'$, then point i is a peak, otherwise, i is a non-peak.

With this definition, we can easily classify points into peaks and non-peaks.

2) Allocation of Non-Peaks Within Local Area: DPC's allocation strategy can be proved reliable for non-peaks based on the idea of the mean shift method [25]. A non-peak at least has a dense region within its local area, among which the nearest dense region is exactly where its nearest neighbor of higher density locates. According to the idea of the mean shift method that a point should shift towards a dense region in its proximity, every non-peak should assign to the same cluster as its nearest neighbor of higher density, i.e., DPC's allocation strategy.

However, peaks cannot directly follow DPC's allocation strategy, because a peak's nearest point of higher density is beyond its local area, which may cause the peak to be assigned into an unassociated cluster. Therefore, an alternative allocation strategy of peaks is necessary.

3) Allocation of Peaks in a Peak-Graph: In order to assign peaks into associated clusters, we come up with a graph-based allocation strategy. Different from SSSP-DPC [22] that assigns all points along the geodesic path in a graph of the dataset, we only assign peaks along the geodesic path in a specific peak-graph, since non-peaks can be assigned reliably within local areas. In order to fast construct the peak-graph, we propose a low-complexity method according to DPC-based allocation paths of non-peaks.

DPC-based allocation paths of non-peaks have constructed an allocation forest composed of multiple allocation trees (denoted as T) with peaks as root nodes. By adding edges that connect mutual-neighbor pairs between adjacent trees, the forest can be connected as an allocation graph.

Definition 2: if $T(i) \neq T(j)$ (i.e., the allocation tree of j), $\exists\, i \in N_k(j)$, and $\exists\, j \in N_k(i)$, then i and j are a mutual-neighbor pair between tree $T(i)$ and $T(j)$.

In the allocation graph, points connected by each edge are closely associated neighbors. Hence, we approximately suppose that all edges have equal weights of association. Based on this, the association between peaks can be evaluated by the geodesic distance in-between (i.e., the number of edges of the shortest path between peaks).

It is worth noting that in an allocation tree a non-peak's depth value (denoted as ϕ) equals the geodesic distance from the non-peak to the peak (root node). While the depth values of non-peaks can be calculated during the allocation of non-peaks: we initially set each $\phi_i = 0$, then during the allocation process, we execute $\phi_i = \phi_j + 1$, where j is i's nearest point of higher density. After the allocation is completed, ϕ_i is the depth value, and if $\phi_i = 0$, then i is a peak.

On this basis, we can fast calculate the association degree $A(i, j)$, namely geodesic distance between adjacent peaks i and j, based on the edge with minimum sum of ϕ between adjacent tree $T(i)$ and $T(j)$ as:

$$\begin{aligned} A(i,j) = \min(&\phi_{i'} + \phi_{j'}) \\ &+ 1, i' \in T(i) \cap N_k(j'), j' \in T(j) \cap N_k(i') \end{aligned} \tag{2}$$

After obtaining association degree of peaks, we construct a weighted graph $G(V, E, w)$ of peaks, called peak-graph, where a peak (node) $v \in V$ is connected to its adjacent peak $u \in V$ through an edge $e_{v,u} \in E$ with an associated cost $w_{v,u} = A(v, u)$. On the basis of peak-graph, we assign a peak to the same cluster as its nearest peak of higher density, and define δ' path as the geodesic path between them:

$$\begin{aligned} \delta_v' &= \min_{u:\rho_u'>\rho_v'} \xi(\Gamma = \{v \to u\}) \\ &= \min_{u:\rho_u'>\rho_v'} (\xi(\Gamma = \{v, p_2, p_3, , , p_t, u\}) \\ &= \min_{u:\rho_u'>\rho_v'} \sum_{i=1}^{t} w_{\Gamma(i),\Gamma(i+1)} \end{aligned} \tag{3}$$

Where $\xi(\Gamma = \{v \to u\})$ is the geodesic distance of geodesic path $\Gamma = \{v \to u\} = \{v, p_2, p_3, , , p_t, u\}$ between peak v and u, and $p_2, \ldots, p_t$ are peaks along the geodesic path. For peaks, their δ' are calculated by applying the Dijkstra algorithm [26]. If there is no path from peak v to other peak of higher density (i.e., $\delta_v' =$

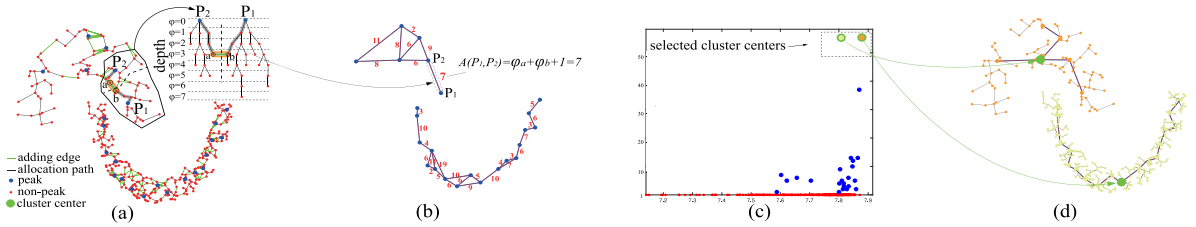

Fig. 1. The core clustering idea of PGDPC on the Jain dataset. First, data points are divided into peaks (blue point) and non-peaks (red point). Then, an allocation graph (a) is constructed by DPC-based allocation paths (black line) of non-peaks and adding edges (green line) between adjacent trees. On this basis, a peak-graph (b) is constructed, where δ' paths of peaks are located. Subsequently, two cluster centers are selected and given unique labels by intuitively observing a ρ'-δ' decision graph (c). Finally, clustering is completed (d) after all non-center points inherit their labels along their δ' paths.

$\varnothing$), we define $\delta'_v = \frac{3}{2} \times (\max_{u:\delta'_u \neq \varnothing}(\delta'_u))$. Followed, among these peaks, we will select appropriate cluster centers.

4) Cluster Center Selection: Cluster centers are considered as peaks that characterized by a higher density ρ' than their neighbors and a relatively large geodesic distance δ' from peaks with higher densities. Since non-peaks with a density smaller than their neighbors are not considered as candidate cluster centers, for non-peaks, we define their δ' paths as their DPC-based allocation paths, $\delta' = 1$. After cluster centers have been located and given unique labels, the remaining points inherit labels along the δ' paths to complete clustering. Fig. 1 illuminates the clustering process of our algorithm on the Jain [39] dataset that consists of two multi-peak clusters.

C. Complexity Discussions

PGDPC's only parameter k is the number of neighbors that usually around $1\%n$ to $2\%n$ [12], where n is the total number of points in the dataset. On the basis of the kNN distance matrix calculated by a fast kNN algorithm (cover tree [37] or kd-tree [38]) with time complexity $O(nlog(n))$. The complexity required for PGDPC's four steps in performing clustering are: step (i) search of peaks and step (ii) allocation of non-peaks together need $O(nlog(k))$; step (iii) allocation of peaks needs $O(|V|log(|V|) + |E|)$ ($|V|$ and $|E|$ represent the number of peak and edges in the $G(V, E, w)$) respectively); (iv) selection of cluster centers to complete clustering needs $O(N)$. Since k, $|V|$, and $|E|$ are all far less than n, the overall time complexity of PGDPC is about $O(nlog(n))$. PGDPC is proved to be extremely efficient with its entire operation requiring no additional distance calculation, except for kNN search.

III. Experiment

A. Experimental Set up

Data Sets: 4 different types of synthesis datasets: (Jain [39], Lineblobs [40], Agg [41], and Birch1 [42]) for clustering performance evaluation of PGDPC. The Berkeley segmentation dataset and benchmark (BSDS500) [28] for real image segmentation experiment. A set of abnormal WCE images obtained from cooperative hospitals and the KVASIR [33] dataset for the WCE image segmentation experiment. **Machine configuration**: experiments are conducted on Matlab (r2017b) on Mac-Book Pro with 2.9 GHz Intel Core i5, 8 G RAM. **Data preprocessing**: we use min-max normalization [27] to normalize input data.

B. Experiments on Synthetic Datasets

Fig. 2(a) shows PGDPC's perfect identification of various-shaped clusters on 4 synthetic datasets. It proves PGDPC's well function on the reconstruction of spatial information of various complex-shaped clusters, which is obviously superior to DPC [20]. Fig. 2(b) displays the execution speed comparison of PGDPC, FastDPeak [16] (i.e., an excellent fast DPC algorithm), k-means [11], and FCM [6]) on the Birch1 dataset with 100,000 points. The results demonstrate PGDPC's fast speed and ability to handle large datasets.

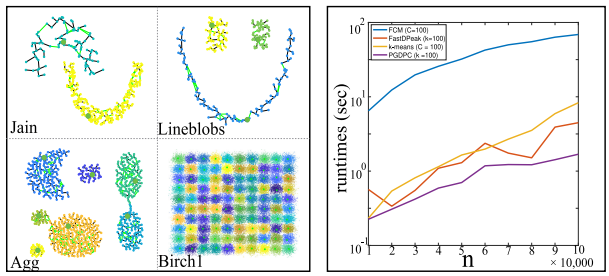

Fig. 2. Experiments of PGDPC on 4 synthesis datasets, with k = 10, 10, 10 and 100, respectively (a). Speed comparison of PGDPC, FastDpeak, FCM, and k-means on 10 different size sampling datasets of the Birch1 dataset (b).

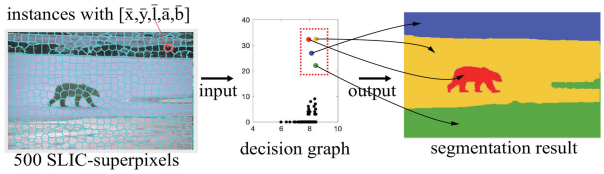

Fig. 3. Our real image segmentation framework.

C. Experiments on Real Image Segmentation

1) Real Image Segmentation Framework: Since the number of pixels in a real image are usually huge, to reduce calculation cost, we use high-efficient superpixel technology (such as SLIC [36], SNIC [43], and SH [44]) to evenly divide the image into N_{sp} superpixels in advance, here we choose the SLIC algorithm. In order to improve the consistency of adjacent superpixels, we smooth the image with Gaussian blur. As input data of PGDPC, each superpixel is represented by $[\bar{x}, \bar{y}, \bar{l}, \bar{a}, \bar{b}]'$, that is, the average value of all pixels' $[x, y, l, a, b]'$ in it, where $[x, y]$ represents coordinate information and $[l, a, b]$ represents the CIE-Lab color information. Then, by observing the decision graph, we easily obtain the number of clusters C. The segmentation task is completed after grouping N_{sp} superpixels into C clusters. We name this real image segmentation method as SLIC-PGDPC with two parameters setting as: $N_{sp} = 500$, and $k = 10$. Fig. 3 illustrates our framework of real image segmentation.

Fig. 4. Comparison of segmentation results on 3 typical real images from the BSDS500 using different algorithms.

TABLE I
AVERAGE PERFORMANCE OF SIX ALGORITHMS ON THE BSDS500 THAT INCLUDES 500 REAL IMAGES

Algorithm	FCM	FGFCM	SFFCM	MMGR-AFCF	SLIC-FCM	SLIC-PGDPC
PRI↑	0.69	0.70	0.80	0.80	0.75	**0.81**
VI↓	3.23	3.18	2.01	2.07	2.35	**1.90**
GCE↓	0.47	0.46	0.26	0.24	0.31	**0.22**
BDE↓	14.26	14.41	12.50	**12.30**	16.79	13.44

TABLE II
COMPARISON OF EXECUTION TIMES (IN SECONDS) OF SIX ALGORITHMS ON THE BSDS500 THAT INCLUDES 500 REAL IMAGES

Algorithm	FCM	FGFCM	SFFCM	MMGR-AFCF	SLIC-FCM	SLIC-PGDPC
Runtime↓	2.14s	7.23s	0.76s	0.73s	0.82s	**0.68s**

2) Results on Real Images: We demonstrate the superiority of the SLIC-PGDPC on real image segmentation with images selected from BSDS500. Fig. 4 is the segmentation results of comparing algorithms on 3 typical real images selected from BSDS500. As shown, FCM and FGFCM fail to segment the images due to the loss of spatial information; SFFCM and MMGR-AFCF obtain high-consistent areas by dividing images into few large superpixels using the MMGR superpixel algorithm in advance, the segmentation results are still not satisfying, because large superpixels always mean a high risk of errors (marked by white area); SLIC-FCM (ie., the FCM algorithm uses SLIC superpixels as input) gets high-consistentbut poor segmentation results; while SLIC-PGDPC obtains both complete and consistent segmentation areas, and perfectly recognizes the outlines of all objects. SLIC-PGDPC's good outcome is a credit to the PGDPC's reconstruction ability of spatial information, and different from SFFCM and MMGR-AFCF, SLIC-PGDPC evenly subdivides an image into so tiny superpixels that even when superpixel errors occur, they won't influence the final segmentation results. This verifies that PGDPC's excellent reconstruction ability of spatial information in image segmentation, which means, unlike FCM-based image segmentation methods, PGDPC can get rid of the high dependency of superpixels.

Table I displays the average performance (the probabilistic rand index (PRI), the variation of information (VI), the global consistency error (GCE), and the boundary displacement error (BDE) [34]) of comparing algorithms on the BSDS500 dataset, where SFFCM and MMGR-AFCF comparably get the best BDE scores, while SLIC-PGDPC provides the best results in PRI, VI, and GCE performance metrics. This experiment demonstrates that SLIC-PGDPC has excellent segmentation performance on real images.

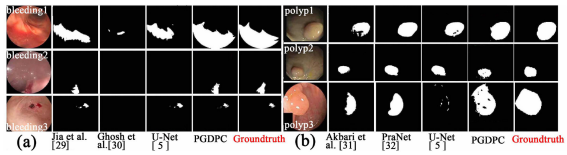

Fig. 5. Comparison of segmentation results of different algorithms on 3 typical bleeding images (a) and 3 typical polyp images (b).

TABLE III
COMPARISON OF AVERAGE IOU SCORE OF DIFFERENT ALGORITHMS ON 30 BLEEDING WCE AND 30 POLYP WCE IMAGES

	Bleeding WCE image				Polyp WCE image			
Algorithm	Jia [29]	Ghosh [30]	U-Net [5]	PGDPC	Akbari [31]	PraNet [32]	U-Net [5]	PGDPC
IOU↑	0.50	0.40	0.54	**0.55**	0.3539	0.4435	0.3690	**0.5457**

In addition, Table II displays the average computational cost of different algorithms on the BSDS500 dataset, where SLIC-PGDPC is faster than all other comparative algorithms.

D. WCE Image Segmentation

WCE image is critical for gastrointestinal (GI) tract abnormal detection. However, unlike real images, WCE image faces high color consistency and blurred outlines problems, which may lead to poor superpixel segmentation results. Herein, to achieve WCE image segmentation, we apply PGDPC to directly group pixels that represented as $[x, y, l, a, b, l', u', v']'$, where $[l,' u,' v']$ is the CIE-LUV color information that used for enhancing features. To reduce computing cost, we compress a WCE image into 100×100 pixels.

Fig. 5 shows the segmentation results of PGDPC and 5 well-trained neural network algorithms [5], [29]–[32] on 6 typical WCE images. As shown, only PGDPC successfully segments edge-blur bleeding areas and similar-color polyp areas from 6 typical abnormal WCE images. Besides, Table III displays the average IOU (Intersection over Union) scores [35] of 6 algorithms on 60 WCE images (30 bleeding images from cooperative hospitals and 30 polyp WCE images from the KVASIR dataset [33], where PGDPC obtains the highest overall IOU score. This experiment demonstrates that PGDPC is capable of WCE image segmentation.

IV. CONCLUSION

PGDPC, a fast DPC algorithm for image segmentation that performs clustering by searching for density peaks in a peak-graph is proposed. PGDPC inherits DPC's idea of finding cluster centers as density peaks, and its peak-graph-based allocation strategy vastly improves DPC's performance of spatial information reconstruction on complex structure clusters. Moreover, PGDPC has a much faster speed than DPC because it is almost free from distance calculation except for the kNN search. With these two improvements and by combining with the SLIC superpixel technology, PGDPC is enabled to deal with image segmentation rapidly and effectively. Numerous experiments have verified that SLIC-PGDPC not only can quickly segment real images but also can ensure a high-consistentin the segmentation areas. In addition, we apply PGDPC to the more challenging WCE image segmentation, experimental results prove that PGDPC also has the ability to successfully perform WCE image segmentation. Thus, we are confident of PGDPC as an applicable clustering algorithm for image segmentation.

REFERENCES

[1] J. Enguehard, P. O'Halloran, and A. Gholipour, "Semi-supervised learning with deep embedded clustering for image classification and segmentation," *IEEE Access*, vol. 7, pp. 11093–11104, 2019.

[2] Z. Liu *et al.*, "Liver CT sequence segmentation based with improved U-net and graph cut," *Expert Syst. Appl.*, vol. 126, pp. 54–63, 2019.

[3] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5221–5229.

[4] Y. Duan *et al.*, "SAR image segmentation based on convolutional-wavelet neural network and Markov random field," *Pattern Recognit.*, vol. 64, pp. 255–267, 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241

[6] J. C. Bezdek, R. Ehrlich, and W. Full., "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2-3, pp. 191–203, 1984.

[7] S. Krinidis and V. Chatzis, "A robust fuzzy local information C-means clustering algorithm," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1328–1337, May 2010.

[8] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognit.*, vol. 40, no. 3, pp. 825–838, 2007.

[9] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, and A. K. Nandi, "Superpixel-based fast fuzzy c-means clustering for color image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 9, pp. 1753–1766, Sep. 2019.

[10] T. Lei, P. Liu, X. Jia, X. Zhang, H. Meng, and A. K. Nandi, "Automatic fuzzy clustering framework for image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2078–2092, Sep. 2020.

[11] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, vol. 1, no. 14, 1967, pp. 281–297.

[12] A. Rodriguez *et al.*, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[13] B. Wu and B. M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1620–1628, Aug. 2017.

[14] B. Liang *et al.*, "Fast density clustering strategies based on the k-means algorithm," *Pattern Recognit.*, vol. 71, pp. 375–386, 2017.

[15] X. Xu, S. Ding, and T. Sun, "A fast density peaks clustering algorithm based on pre-screening," in *Proc. IEEE Int. Conf. Big Data Smart Comput.*, 2018, pp. 513–516.

[16] Y. Chen *et al.*, "Fast density peak clustering for large scale data based on kNN," *Knowl.-Based Syst.*, vol. 187, 2020, Art. no. 104824.

[17] R. Liu *et al.*, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci.*, vol. 450, pp. 200–226, 2018.

[18] J. Xie *et al.*, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors," *Inf. Sci.*, vol. 354, pp. 19–40, 2016.

[19] Y. Liu, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy," *Knowl.-Based Syst.*, vol. 133, pp. 208–220, 2017.

[20] G. Wang, Y. Wei, and P. Tse, "Clustering by defining and merging candidates of cluster centers via independence and affinity," *Neurocomputing*, vol. 315, pp. 486–495, 2018.

[21] J. Hou, A. Zhang, and N. Qi, "Density peak clustering based on relative density relationship," *Pattern Recognit.*, vol. 108, 2020, Art. no. 107554.

[22] D. U. Pizzagalli, S. F. Gonzalez, and R. Krause, "A trainable clustering algorithm based on shortest paths from density peaks," *Sci. Adv.*, vol. 5, no. 10, 2019, Art. no. eaax3770.

[23] M. Du *et al.*, "Density peaks clustering using geodesic distances," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1335–1349, 2018.

[24] S. A Seyedi *et al.*, "Dynamic graph-based label propagation for density peaks clustering," *Expert Syst. Appl.*, vol. 115, pp. 314–328, 2019.

[25] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[26] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[27] P. Franti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 29, no. 5, pp. 761–775, 2006.

[28] P. Arbeláez, M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[29] X. Jia and Q. Meng, "A study on automated segmentation of blood regions in wireless capsule endoscopy images using fully convolutional networks," in *Proc. IEEE 14th Int. Symp. Biomed. Imag.*, 2017, pp. 179–82.

[30] T. Ghosh, L. Li, and J. Chakareski, "Effective deep learning for semantic segmentation based bleeding zone detection in capsule endoscopy images," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 3034–3038.

[31] M. Akbari *et al.*, "Polyp segmentation in colonoscopy images using fully convolutional network," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2018, pp. 69–72.

[32] D.-P. Fan *et al.*, "Pranet: Parallel reverse attention network for polyp segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2020, pp. 263–273.

[33] K. Pogorelov *et al.*, "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 164–169.

[34] X. Wang, Y. Tang, S. Masnou, and L. Chen, "A global/local affinity graph for image segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1399–1411, Apr. 2015.

[35] H. Rezatofighi *et al.*, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

[36] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol, no. 11, pp. 2274–2282, Nov. 2012.

[37] A. Beygelzimer *et al.*, "Cover trees for nearest neighbor, " in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 97–104.

[38] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.

[39] A. Jain and M. Law, "Data clustering: A user's dilemma," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, 2005, pp. 1–10.

[40] L. Zelnikmanor and P. Perona, "Self-tuning spectral clustering," in *Proc. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608

[41] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, pp. 1–30, 2007.

[42] T. Zhang *et al.*, "BIRCH: A new data clustering algorithm and its applications," *Data Mining Knowl. Discov.*, vol. 1, no. 2, pp. 141–182, 1997.

[43] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4651–4660.

[44] X. Wei, Q. Yang, Y. Gong, N. Ahuja, and M. Yang, "Superpixel hierarchy," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4838–4849, Oct. 2018.