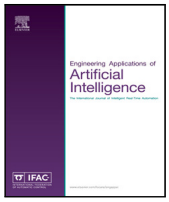
Engineering Applications of Artificial Intelligence 127 (2024) 107212

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Global superpixel-merging via set maximum coverage

Xubing Yang [a,*], Zhengxiao Zhang [a], Li Zhang [a], Xijian Fan [a], Qiaolin Ye [a], Liyong Fu [b,*]

[a] College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China
[b] Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, Beijing, 100091, China

ARTICLE INFO

Keywords:
Superpixel
Image segmentation
Maximum coverage
Region-merging
Hierarchical decomposition

ABSTRACT

Due to better boundary adherence and low computational cost, the superpixel segmentation algorithm SLIC (simple linear iterative clustering) has been widely applied in vision-based applications. However, limit to unavoidable over-segmentation problem, one has to consider region-merging to reconstruct entire objects from the segmented superpixels (or called regions). The existing region-merging methods are generated from data clustering, and avoidably suffer from error-merging, slow convergence speed, or easily dropping in LOCAL optimal problems, especially for high-resolution RS (remote sensing) images. In this paper, instead of data clustering, we propose a fast GLOBAL method based on Set Maximum Coverage, termed as MaxCov-merging. Theoretically, the existence of the maximum coverage is proved by using Bayes optimal decision principle. To speed up MaxCov-merging, some heuristic strategies are also provided. Finally, extensive verification and comparison are carried on the public and our collected high-resolution images. Compared with the state-of-the-art methods, the comparison shows the superiority of our MaxCov in terms of the performance of globality, ease of use and fast region-merging speed.

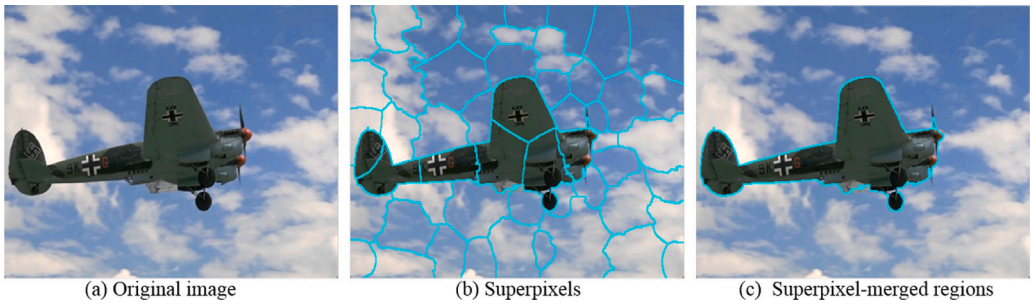
1. Introduction

Superpixel techniques, e.g., SLIC, have been widely used in remote sensing, image process and computer vision (Stutz et al., 2018; Wu et al., 2021; Yuan et al., 2021; Li et al., 2021; Derksen et al., 2019; Zhang et al., 2020; Martins et al., 2021; Zhou and Yun, 2020), which aim to partition image pixels into a group of meaningful atomic regions named superpixels. Although they are easy to use, better boundary adherence, and low computational cost, the problem of over-segmentation, under-segmentation or them both (Lei et al., 2019) always occurs in generating superpixels. Here over-segmentation means that in human vision, some large-size image objects are unavoidably split into multiple patches (regions); while under-segmentation means that there have some superpixels, which pixels are from multiple distinct image objects. In general, the quality of a segmentation is determined by the initial superpixel parameters, e.g., the number of superpixels k in SLIC. In practical view, one prefers to over-segmentation, e.g., adopting a large k for the SLIC segmentation, to achieve a better boundary adherence for the most of image objects. In this view, the under-segmentation problem seems able to be relieved, e.g., by increasing k, but over-segmentation is unavoidable. To clarify, Fig. 1 shows an example. At first glance, there have two objects in Panel (a) of Fig. 1: an aeroplane in the foreground and a blue sky with white clouds in the background. After SLIC segmentation, both objects are over-segmented to multiple superpixels, as shown in Panel (b). The difference between them is that: each foreground superpixel gains a better segmentation boundary; while the background superpixels are not, e.g., some clouds are under-segmented. In order to reconstruct image objects, we need a method to merge the segmented patches/regions together, and to form complete objects. Panel (c) of Fig. 1 shows the desired region-merging result, in which the merged aeroplane object is well-segmented from the merged background.

As for region-merging, the existing methods are commonly based on data clustering, including affinity propagation (AP) (Zhou, 2015; Wu et al., 2016), DBSCAN (Density-Based Spatial Clustering of Application with Noise) (Akyilmaz and Leloglu, 2016; Hadavand et al., 2019), and hierarchical clustering (Hu et al., 2017). Concretely, the former two directly use AP or density-based DBSCAN to cluster the segmented superpixels, and return the clustering results as the final region-merging. Due to data-clustering, they both just focus on the closest pair of *local* superpixels in each iteration, regardless of whether they are from the same object or not. To resist this local-merging, the latter one (Hu et al., 2017) followed the accumulated hierarchical clustering (Roux, 2018) and proposed a so-called *global*-merging by using evolution analysis. However, the limitations like slow convergence speed and local optimal solutions unavoidably affect the performance of clustering-based algorithms.

* Corresponding authors.
E-mail addresses: xbyang@njfu.edu.cn (X. Yang), xxz@njfu.edu.cn (Z. Zhang), zhangli@njfu.edu.cn (L. Zhang), xijian.fan@njfu.edu.cn (X. Fan), yqlcom@njfu.edu.cn (Q. Ye), fuly@ifrit.ac.cn (L. Fu).

https://doi.org/10.1016/j.engappai.2023.107212
Received 15 October 2022; Received in revised form 11 September 2023; Accepted 25 September 2023



(a) Original image (b) Superpixels (c) Superpixel-merged regions

Fig. 1. Illustration for superpixel segmentation and region-merging. Panels (a) and (b) show an image and its SLIC superpixels. Panel (c) is the desired region-merging result from the perspective of image segmentation and object detection.

The description of abbreviation

DBSCAN	Density-Based Spatial Clustering of Applications with Noise
AP	Affinity Propagation data clustering
HC	Hierarchical Clustering
HC-merging	Hybird simplification for image hierarchies in region-merging
SLIC	Simple Line Iterative Clustering
MaxCov	A global region-merging method based set Maximum Coverage
GT	Ground Truth
BR	Boundary Recall, an indicator for object boundary adherence
CUE	Corrected Under- segmentation Error
NBR	Not-Boundary-Rate, an indicator for region-merging
BSDS	Berkeley Segmentation Data Set

On the other hand, the goal of region-merging is different from superpixel segmentation. For example, in SLIC a segmented superpixel only needs to consider the local pixels within a $2s \times 2s$ image region, where $s = \sqrt{N/k}$ and N is the total number of pixels. But for region-merging, it is opposite. The goal of region-merging is to form a complete object by merging similar superpixels together. In human vision, a common sense for the merging is that the superpixels from the same object should have a higher similarity than those from different objects. Thus, the merging method should be designed in *global* view to capturing the integrity of image objects. However, the existing methods failed to capture this integrity due to the following facts: (1) each merging/clustering step is performed on the most similar pair of superpixels, even if they are from different objects; (2) Once the error-merging happens, e.g., in hierarchical clustering, it is difficult to be corrected in subsequent iterations. In addition, due to the complexity of a application itself, the color values of a given object may not be always fixed (pure color), but gradually vary in a certain range, or terminologically, are drawn from an unknown unimodal distribution. That is, the distance between a pair of homogeneous superpixels, e.g., drawn from the same object, should be closer, but not necessarily be the closest. In order to enhance the concept of closest distance utilized in data clustering, our objective is to introduce a novel metric for measuring "closer" similarity, and then develop a GLOBAL region-merging method on the collected closer pairs. Because this method is built upon Set Maximum Coverage, we name it MaxCov merging in this work.

We highlight our contributions as follows.

- In contrast to local region-merging, a global method named MaxCov merging is proposed;
- The MaxCov merging problem can be converted into calculating a maximum coverage set to ensure globality, rather than clustering data locally as used in existing methods;
- Theoretically, the existence of the global coverage is proved, based on the theory of set maximum cover;
- We also provide some heuristic strategies, e.g. asymptotic approximation, and redundant region elimination, to speed-up MaxCov region-merging.

The rest of this paper is organized as below. In Section 2, we briefly review the related work. Our method will be detailed in Section 3, including modeling and solution, geometrical interpretation, and related theoretical proofs. To verify the performance of our proposed, experimental simulation is arranged in Section 4. We conclude the whole paper in Section 5.

2. Preliminaries

Existing superpixel segmentation/generation methods are commonly divided into two categories (Stutz et al., 2018): graph-based and gradient-based ones. The former, e.g. normalized cuts (Shi and Malik, 2000) or spectral clustering (Chen et al., 2017), treats each pixel as a node and constructs edge weights between pairs of nodes in the graph. Superpixels are then created by minimizing the defined cost function over the graph; The latter usually originates from data clustering, including mean shift (Wang et al., 2022), watershed (Vincent and Soille, 1991), turbopixel (Levinshtein et al., 2009), and SLIC (Achanta et al., 2012). Then the superpixels are generated from the refined clusters by gradient decent methods.

In region-merging, in addition to the above-mentioned, there also have several non-SLIC superpixel-merging methods in the literature. For example, Xiang et al. aimed to eliminate the mixed ALFCE (adaptive superpixel generation approach) superpixels of SAR (synthetic aperture radar) images and provided a method based on statistical region merging (SRM) (Xiang et al., 2021; Nock and Nielsen, 2004). By relaxing regular-shape constraints in superpixel segmentation, Lei et al. (2019) proposed an adaptive morphological reconstruction (AMR) on irregular watershed superpixels and also proposed two AMR methods for region-merging, named AMR_WT (watershed transformation) and AMR_SC (spectral clustering), respectively. As addressed in Stutz et al. (2018), there have more than 28 superpixel algorithms, here we only name a few. Among these methods, due to the superiority in boundary adherence, low computational complexity and ease of operation, SLIC has been the most popular in various vision-based applications. Next our discussion is narrowed on SLIC and we just review the most related work in this section.

2.1. DBSCAN or AP clustering-based merging

To bridge the gap between superpixels and image objects, DBSPA (density-based super-pixel aggregation) (Hadavand et al., 2019) aims to group superpixels together by DBSCAN, consisting of four steps: SLIC segmentation, DBSCAN clustering on superpixels, non-core areas segmentation and random forest classification. Instead of DBSCAN, SLICAP (SLIC AP) (Zhou, 2015) uses affinity propagation (AP) to cluster SLIC superpixels. The authors defined a set of weighted distances in CIELab space to pass the superpixel similarities to AP algorithm. Then the final merging is obtained by the "examplars", i.e. the cluster centers of the AP.

The main shortcomings lie in three-fold: (1) The merging seems very casual, and the final result is over-dependent on the selected clustering methods; (2) It is difficult to control, either the number of final regions or the processing of the merging; (3) They have no consideration on the integrity of objects.

2.2. Evolution analysis method (Hu et al., 2017)

Different from the above, Hu et al. suggested a evolution analysis method (Hu et al., 2017) to simplify region-based image hierarchies in local and global view, named hybrid simplification (shortly, HC-merging). Here the "hybrid" means intra- or inter-region homogeneity and heterogeneity measured by the probability functions, which are defined in (1) and (2), respectively.

$$P_U(i) = \frac{\upsilon_i - \upsilon_{\min}}{\upsilon_{\max} - \upsilon_{\min}} \tag{1}$$

$$P_O(i) = \frac{MI(i) - MI_{\min}}{MI_{\max} - MI_{\min}} \tag{2}$$

where P_U and P_O denote probability functions of under- and over-segmentation, and i is the index of the section in the hierarchy. The symbols υ and MI denote global variance and region autocorrelation metric, respectively. The subscript "min" or "max" stands for the global minimum (at the 0-section) or maximum (top section). Then the hybrid simplification can be determined by the following function,

$$e(t) = P(\omega_U) \sum_{i=1}^{t} p_U(i) + P(\omega_O) \sum_{i=t+1}^{N} p_O(i) \tag{3}$$

where $P(\omega_U)$ and $P(\omega_O)$ are prior probabilities of under- and over-segmentation errors. Correspondingly, p_U and p_O denote the density functions.

Intuitively, it is efficient for those images which have clear hierarchical structures. However, it is unknown whether HC-merging is efficient for remote sensing images due to the complexity of RS scenes. Moreover, using an absolute metric, e.g. global variance, to identify redundant regions is also unsuitable, because it may deepen the depth of the hierarchy in the process of tree-building, if these redundancies have different differences (or distances) from the global minimum. In other words, the more complex the scene is, the worse the situation becomes. Additionally, originated from hierarchical clustering, it may result in error-merging problem because each step of region-merging is still executed on the most similar pair.

To explain our opinions, we give an example on a satellite RS image, as shown in Fig. 2. To fit the boundary of the object "road" well, the initial k for SLIC is set to $k = 2000$. After segmentation, the results are obtained by our MaxCov-, DBSCAN-, AP-, and HC-based region-merging methods, as shown in Panels (c) to (f), respectively. Obviously, the results of DBSCAN and AP seem not yet complete, in which so many superpixels of the object "farmland" can continue to be merged, as shown in Panels (d) and (e); Correspondingly, both results in the rightmost column, i.e., Panels (c) and (f), seem acceptable because most objects are got merged. In contrast to HC-merging, MaxCov seems so clear that many image details are still retained. For example, the superpixels located in the area of the "village" are not merged any more because they are different from their neighboring superpixels. However, due to the merging on the most similar pair, an error-merging problem appears in HC-merging, as shown in the corner of right-bottom of Panel (f), where some superpixels of the "road" and "farmland" are falsely merged together. Similar phenomena can also be seen in the objects "village" and "farmland".

3. OUR MaxCov merging

In image processing and pattern recognition, an image is usually defined as a qualitative model (Nikou et al., 2010; Sanjay-Gopal and Hebert, 1998),

$$P(\mathbf{z}) = \sum_{i=1}^{K} P(\omega_i) p(\mathbf{z}|\omega_i; \boldsymbol{\theta}_i) \tag{4}$$

where K is the number of image objects. $P(\omega_i)$ and $p(\mathbf{z}|\omega_i)$ denote the prior probability and density function (PDF) of the ith object, respectively. The variable $\mathbf{z}$ denotes a random pixel, but $\boldsymbol{\theta}_i$ is a density parameter. In computation, the PDF of a given object can usually be approximated by a Gaussian function, in this case $\boldsymbol{\theta}_i$ is used as the density parameter, e.g., formed by the mean $\mathbf{u}_i$ and the covariance $\boldsymbol{\Sigma}_i$ of the pixel distribution of the ith object.

From (4), for a given pixel $\mathbf{z}$, which object it should belong to is determined by the maximum posteriori probability. Reflecting in region-merging problem, the decision can also be made by the prior probability, which will be summarized in Lemma 1 for easy-reading.

Lemma 1. *Let ω_j be the jth object. For $\forall \mathbf{z} \in \omega_j$, its posteriori probability is proportionate to its prior one. That is,*

$$P(\omega_j|\mathbf{z}) \propto P(\omega_j) \cdot p(\mathbf{z}|\omega_j) \tag{5}$$

where the proportional notation $a \propto b$ means that there exists a positive real number $\epsilon > 0$ such that $a = \epsilon b$.

The proof and related interpretation are provided in Appendix A.1. Note that the posteriori probability is obtained by the maximum Bayes decision, which guides us for the object-region maximization problem, and how to use a similarity metric.

- *Object-region maximization*

Lemma 1 states that the maximization of $P(\omega_j|\mathbf{z})$ is equivalent to maximize $P(\omega_j) \cdot p(\mathbf{z}|\omega_j)$. If all objects have been well segmented by SLIC, then $p(\mathbf{z}|\omega_j)$ is actually confirmed because in this case, the distribution of the ith object can be accurately estimated, e.g., by using the aforementioned Gaussian function. As for $P(\omega_j)$, it is usually approximated by the frequency of the pixel $\mathbf{z}$, for $\forall \mathbf{z} \in \omega_j$, in which ω_j can be explained as a confidence interval of $\mathbf{z}$. Obviously, this frequency is obtained in global view, e.g., using the ratio between the number of occurrences of the pixel $\mathbf{z}$ and total image pixels. Thus the occurrence of the pixel $\mathbf{z}$ does not depend on the localization of the superpixels, which means that $\mathbf{z}$ may appear in multiple superpixels or even in multiple objects, as shown the pixel of "farmland" in Fig. 2. If continuously varying $\mathbf{z}$ in a confidence interval of an image object, geometrically, it would form a connected region to cover the entire object, i.e., object-region maximization.

- *Usage of similarity metric*

The similarity used for region-merging can be interpreted by pixel distribution, as shown in Fig. 3. Suppose we are given two objects, denoted by $p(\mathbf{z}|\omega_i)$ and $p(\mathbf{z}|\omega_j)$, respectively. If a pixel $\mathbf{z}$ appears in the magenta overlapping area, it is difficult to determine which object it should belong to. Whatever the answer is, however, at least we know that it is unreasonable if $\mathbf{z}$ is merged to the most similar object as the data clustering methods do. In computer vision, it should be better if the decision is made in global view, which will be detailed in next subsections.

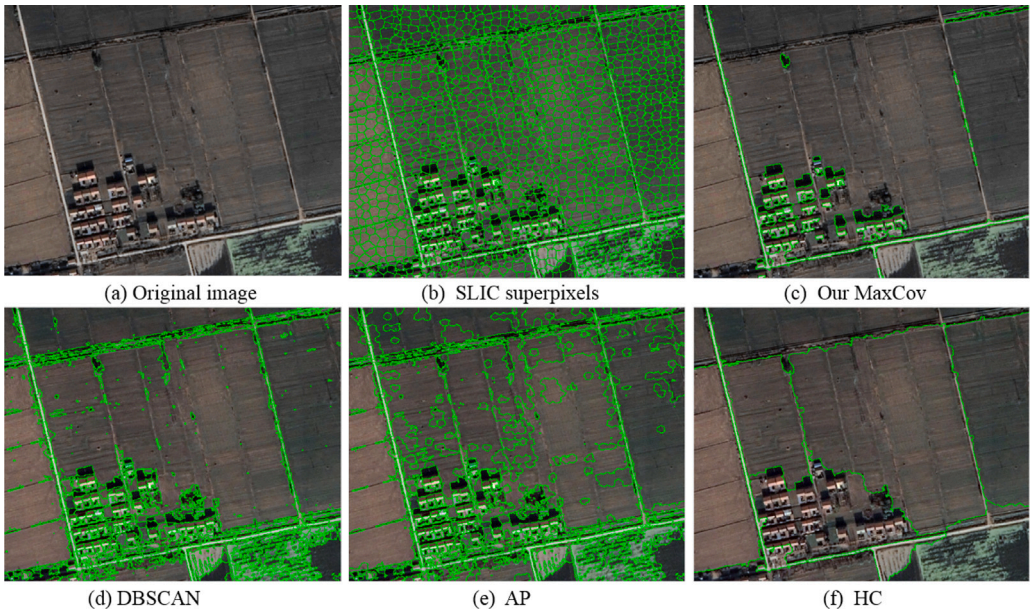


Fig. 2. Region-merging on a given RS image (a) and the SLIC superpixels (b). Panels (c) to (f) show the results of region-merging, obtained by our MaxCov, DBSCAN, AP, and HC, respectively.

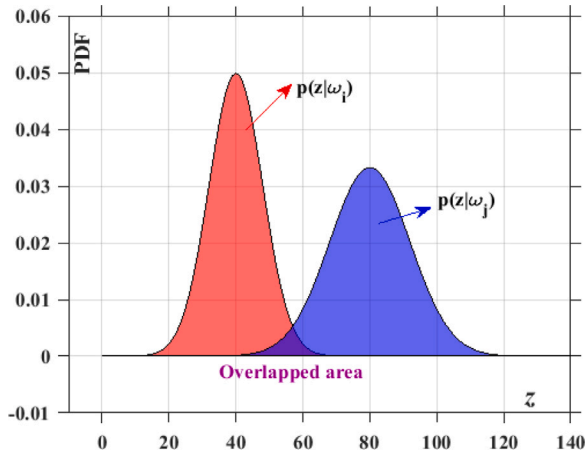


Fig. 3. Interpretation for the similarity between regions.

3.1. Problem description and conversion

A superpixel in a 2D RGB image is usually defined as a 5-dimensional vector. Let R_i be the ith superpixel. Correspondingly, the bold $\mathbf{R}_i$ denotes its pixel-pattern vector, i.e., $\mathbf{R}_i = (x_i, y_i, r_i, g_i, b_i)^T$ or $\mathbf{R}_i = (\mathbf{l}_i^T, \mathbf{p}_i^T)^T$. Here $\mathbf{l}_i = (x_i, y_i)^T$ is used for pixel location and $\mathbf{p}_i = (r_i, g_i, b_i)^T$, for color values. If without ambiguity, R_i is also used to represent a set of pixels, drawn from the ith superpixel region. Let the set $\{R_1, \ldots, R_k\}$ be the segmented superpixel regions, such that $I = \cup_{i=1}^{k} R_i$ and $R_i \cap R_j = \varnothing$ $(\forall i, j \in \{1, \ldots, k\}, i \neq j)$, where I is a full set, corresponding to the given image. The operator "$\cup$" and "$\cap$" stand for set union and intersect, respectively. According to set calculation, this set forms a *coverage* for I, as illustrated in Fig. 4. This coverage is shown as the union of the regions numbered "1" to "53", with the initial $k = 50$. Hereafter, the number of initial or final superpixels for SLIC segmentation is denoted by the same symbol k, though they may have a slight difference.

As shown in Fig. 4, for a given image I and k final segmented regions, there has a total of 2^k coverages, corresponding to the power set of the superpixels $\{R_1, \ldots, R_k\}$. Among them, we attempt to find such a coverage C, in which each element is able to cover an entire object if the corresponding object-region is connected. This coverage C for K objects can thus be defined in (6).

$$C = \{S_1, S_2, \ldots, S_K\}, \quad S_i = \cup_{j=1}^{i_K} R_{i_j}. \tag{6}$$

where $\{i_1, \ldots, i_K\}$ is a sorted subset of the index set $\{1, \ldots, k\}$, $i = 1 \sim K$. Worked as a coverage, C must satisfy (7).

$$\begin{aligned} &I = \cup_{i=1}^{K} S_i, \\ &S_i \cap S_j = \varnothing, \quad \forall i, j \in \{1, \ldots, K\}, i \neq j \end{aligned} \tag{7}$$

According to (6), if the merged region S_j can cover the jth object, $j \in \{1 \ldots K\}$, then the coverage C formed by all S_j is our desired maximum coverage. To approach this goal, first of all, we need to find an efficient strategy for searching such a coverage, instead of the searching in 2^k coverages.

3.2. Maximum coverage

Inspired by bottom-up strategy in hierarchic clustering (Hu et al., 2017), we provide a bi-direction global method: bottom-up for virtual-merging and up-bottom for real-merging, as illustrated in Fig. 5. For convenience, we give some definitions.

Definition 1 (*Base Region or Unit*)**.** A region is formed by only one superpixel, called base region or unit.

Definition 2 (*l-layer Region*)**.** A l-layer region is a set, in which each element contains l units.

In order to efficiently manage so many regions, we rewrite the aforesaid R_i as $R_i^{(l)}$, and sort the elements of l-layer in ascending order by unit number, as shown in Fig. 5.

In bottom-up step, each virtual-merging in the l-layer is just carried on two elements (regions): one in the $(l-1)$-layer and the other in the 1-layer, if they are similar in both spatial proximity and color similarity. Repeat virtual-merging from bottom to top until an empty layer appears, shown as $\varnothing$ in Fig. 5. Actually, this step is to obtain the maximum object region, which can contain as many units as possible, e.g., the element "4,5, …,47,49" in the second top layer in Fig. 5. Considering easy explanation, the region with yellow boundaries is also

Fig. 4. Original image and the numbered region partitions.

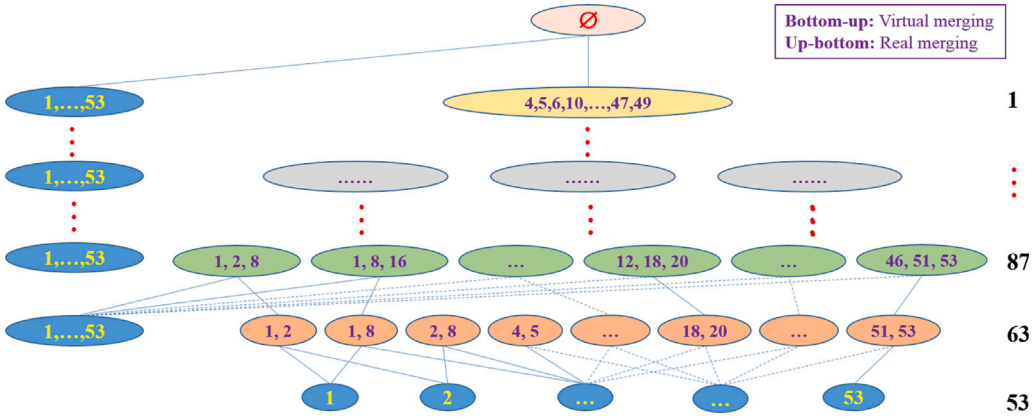


Fig. 5. Explanation for maximal region generation. The bold number in the rightmost column is the number of elements in current layer.

visualized in Fig. 6. Note that the "virtual" just means that a pair of candidate regions satisfies the necessary condition of region-merging. Whether they will be merged or not depends on the step of up-bottom real-merging.

Up-bottom is a decision-making step for the final regions. As shown in Fig. 5, suppose that the last maximal region(s) appears in the $(t-1)$-layer, here t means the " top" layer, corresponding to the empty set $\varnothing$. Next, backtrack from the $(t-1)$-layer to the bottom layer and remove redundant virtual-merging, then the remaining regions are used to form real-merging. Concretely, the real-merging consists of two steps. Firstly, we select as many disjoint elements as possible in the current layer, and use them to form a real-merging set. Secondly, two kinds of elements should be removed from the virtual-merging: (1) the remaining elements in the current layer; and (2) the elements in the next layer if they intersect to the real-merging set. A strategy named longest-and-shortest is provided for real-merging if there have at least two disjoint elements in the current layer. This means that the first two disjoint elements are selected directly by the longest distance and added to the real-merging set; Then the next element is selected in turn by the shortest distance, if it is also disjoint to the elements in the current real-merging set.

Compared with the existing methods, the advantages of our MaxCov lie in 4-fold: (1) The proposed region-merging is designed in global view, which significantly differs from the local clustering-based ones; (2) It is more immune to local optimum, though the region-merging is still performed on the pairs of the similar regions, but NOT on the most similar pairs; (3) The region-merging can be early terminated as long as the last maximum region or $\varnothing$ appears; while the clustering-based methods, e.g. hierarchical clustering, do not stop until all data are grouped into the last ONE cluster; (4) It is also more economical than power set generation in terms of computational cost or memory storage. As shown in Fig. 5, there are merely 63 or 87 elements in the 2th and 3th layer of the virtual-merging, rather than 1378 or 23,436 elements obtained by the combinatorial number C_{53}^{2} or C_{53}^{3}, respectively, when computing power set.

For convenience, the above discussion and computation are summarized in Theorem 1 and Algorithm 1, respectively.

Theorem 1 (*The Existence of Maximum Coverage*)**.** *For any given units, there must exist a maximum coverage.*

The proof is provided in Appendix A.2. In favor of generating higher-layer element, as illustrated in Fig. 5, we need a metric to measure similarity for a candidate pair of regions. In general, it can be defined as

$$\|\mathbf{R}_i^{(l)} - \mathbf{R}_j^{(1)}\| < \epsilon, \tag{8}$$

where $\mathbf{R}_i^{(l)}$ denotes the ith element in the l-layer.

Instead of global searching, we can find the candidate base regions for $\mathbf{R}_i^{(l)}$ in its neighborhood, as defined in (9). Substitute $\mathbf{R}_i^{(l)} = ((\mathbf{l}_i^{(l)})^T, (\mathbf{p}_i^{(l)})^T)^T$ into (8) and expand it, we have

$$\|\mathbf{p}_i^{(l)} - \mathbf{p}_j^{(1)}\| < \epsilon \quad \wedge \quad \mathbf{l}_j^{(1)} \in Ne(\mathbf{l}_i^{(l)}). \tag{9}$$

where $Ne(\mathbf{l}_i^{(l)})$ denotes the neighborhood of $\mathbf{l}_i^{(l)}$, which is easily obtained by SLIC segmentation, e.g. the adjacency graph matrix of superpixels (Hu et al., 2017; Akyilmaz and Leloglu, 2016), as shown in Fig. 6. For example, the neighborhood of the base region "1", $Ne(\mathbf{l}_1^{(1)})$, is composed of its adjacent regions "2", "7" and "8". Thus, the spatial proximity for these base regions can be accurately measured by this adjacency graph; Whether they will be merged or not is also determined by color similarity. Visually, these color values may be quite different, e.g., the color of "1" and "7". As defined in (9), this color proximity strongly depends on the given positive real value ϵ, which will be detailed in the next subsection. Here we summarize the above steps in Algorithm 1.

In view of image understanding, region-merging is very subjective. Our MaxCov-merging is easy to use because it just needs an appropriate

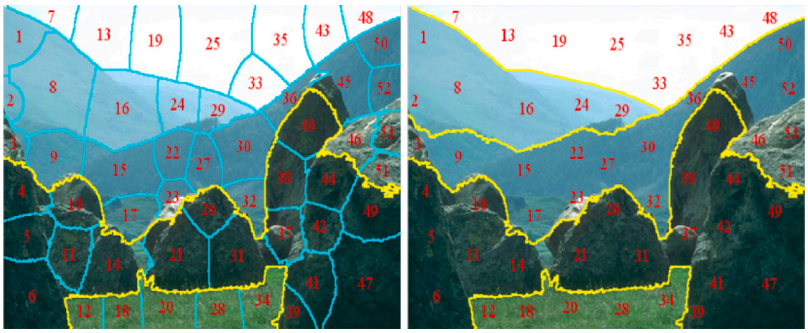


Fig. 6. Illustration for the maximal object-region and final region-merging.

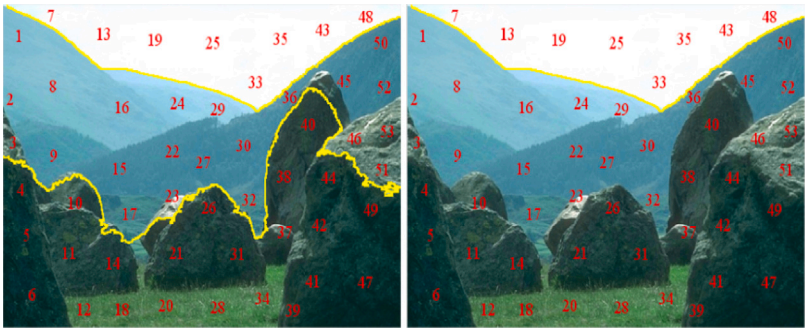


Fig. 7. Another two cases of the final region-merging.

Algorithm 1: MaxCov Algorithm

Input : Given k base regions $\{R_i^{(1)}\}_{i=1}^k$ and ϵ;
Output: Maximum coverage Ω.

1 *Initialization*. Set $\Omega = \varnothing$, and compute region adjacency matrix $\mathbf{M}$ and $\{\mathbf{R}_i^{(1)}\}_{i=1}^k$; Let $t = 0$;
2 *Bottom-up for virtual-merging*. Do virtual-merging by (9) w.r.t. M and ϵ, until $\varnothing$ appears. Set $t = top - 1$;
3 *Up-bottom for real-merging*
4 **repeat**
5 Obtain disjoint maximum virtual-merging regions Ω_t in the t-layer;
6 Update Ω by $\Omega = \Omega \cup \Omega_t$;
7 Remove those elements from the $(t-1)$-layer if they intersect with Ω;
8 Set $t \leftarrow (t-1)$;
9 **until** $t <= 1$;
10 Return Ω ;

ϵ. Then it will be convergent to the desired human subject automatically. For example, if given another two ϵ's, one can obtain different results of the final region-merging, as shown in Fig. 7. These results are all based on object-area maximization, just with different image understandings.

3.3. Heuristic strategies

For ease of use, in this subsection we provide two heuristic strategies: parameter selection for ϵ and speed-up strategy for MaxCov-merging.

3.3.1. Parameter selection for ϵ

As described in (9), the similarity between a pair of regions, e.g., $\mathbf{R}_i^{(l)}$ and $\mathbf{R}_j^{(1)}$, is measured by the proximity of color values and spacial coordinates. Since spacial proximity can be obtained easily and accurately, the virtual-merging between $\mathbf{R}_i^{(l)}$ and $\mathbf{R}_j^{(1)}$ to be executed or not, naturally, is determined by ϵ. In order to explore ϵ, Fig. 8 shows an example on 260 pairwise distances in ascend between the superpixels and their neighbors, legend with red lines. In this case, the necessary condition of region-merging is subject to $\epsilon \in [0, 235]$, here the upper bound 235 is the longest distance of the color values of the superpixel pairs. Interestingly, there always have some jump points (or called knee points), in which each of them can be used as a candidate of ϵ. In computation, it can be estimated by function approximation, polynomial regression (Hu et al., 2017) or supervised methods (Tong et al., 2020). For example, the result shown in Fig. 6 is obtained with the letting $\epsilon = 56$, a jump point (the exact value 55.4) on the red distance curve of Fig. 8. If redraw such curve after this region-merging, we show it with blue lines in Fig. 8, in which the superpixels being merged together share the same color value. If viewing the blue curve as an approximation to the red one, we are motivated to consider a speed-up version.

3.3.2. Speed-up strategy for MaxCov

A larger ϵ means that there have more base regions satisfying (9). It is no doubt that it will deepen the layer/depth of virtual-merging, and thus require more memory storage and computation capacity. Instead, for a smaller ϵ, although the merging is easy to be implemented, an entire object-region is usually unobtainable. Intuitively, in this case only some sub-regions of the same object may satisfy the condition (9), and then can be merged; While the rest cannot. This reminds us to develop an asymptotic method for the MaxCov-merging, inspired by interval (or set) cover theory. Concretely, for a pre-defined ϵ, if we can find a ordered real sequence $\{\epsilon_i\}$, satisfying $0 = \epsilon_0 < \epsilon_1 < \cdots < \epsilon_n = \epsilon$, then MaxCov-merging on $[0, \epsilon]$ can be asymptotically approximated by executing MaxCov-merging n times respectively on the nested small intervals $[0, \epsilon_i]$, $i = 1 \sim n$, where the "asymptotically" means that the previous output of the MaxCov-merging, e.g., on $[0, \epsilon_{i-1}]$, will

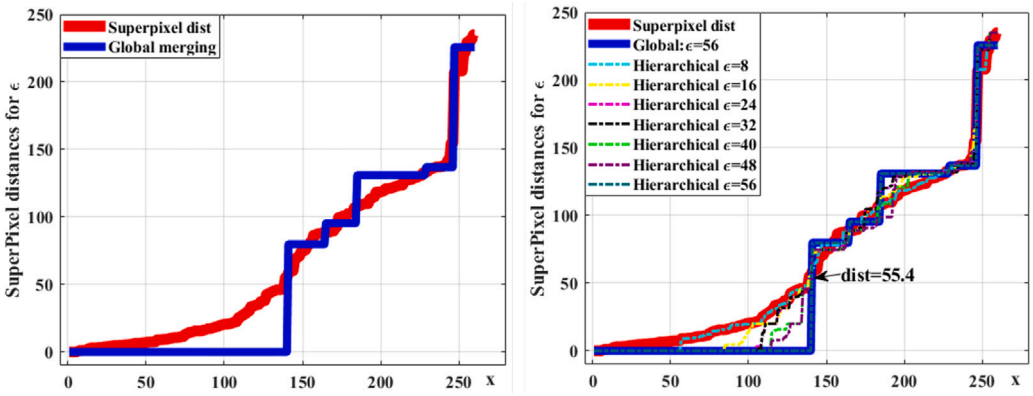


Fig. 8. Illustration for selecting ϵ. Left: Global region-merging; Right: hierarchical approximation to global-merging. Here the abscissa "x" denotes the index of the paired superpixel distances ordered in ascend.

be used as the next input of MaxCov-merging, e.g., on $[0, \epsilon_i]$, where $[0, \epsilon_{i-1}] \subset [0, \epsilon_i]$. Thus, the interval $[0, \epsilon_i]$ for the ith MaxCov-merging can be decomposed into two disjoint sub-intervals: $[0, \epsilon_{i-1}]$ and $(\epsilon_{i-1}, \epsilon_i]$. Because the merging on $[0, \epsilon_{i-1}]$ has been completed in previous steps, the merging on $[0, \epsilon_i]$ is just carried on $(\epsilon_{i-1}, \epsilon_i]$. Thus, a lot of memory and computation time can be saved. For convenience, the above is concluded in the following theorem.

Theorem 2 (*Asymptotic Approximation Theorem*)**.** *For any given positive ϵ, there must exist a sequence of nested intervals convergent to the closed interval* $[0, \epsilon]$.

The proof is provided in Appendix A.3. An explanation for the asymptotic method is that, for the ith MaxCov on $[0, \epsilon_i]$, a pair of regions can be merged if it satisfies (9), i.e., $\|\mathbf{p}_i^{(l)} - \mathbf{p}_j^{(1)}\| < \epsilon_i$. Naturally, it also satisfies $\|\mathbf{p}_i^{(l)} - \mathbf{p}_j^{(1)}\| < \epsilon$ and the merged pair should appear in the global virtual-merging. Fig. 8 also shows an example for the asymptotic method, with the setting $\epsilon = 56$ and the real sequence $\{\epsilon_j | \epsilon_j = j * 8, j = 1 \sim 7\}$, shown as the curve with the legend "Hierarchical $\epsilon = j * 8$". When ϵ_j closes to the ϵ, the obtained asymptote also gradually approaches the blue solid global curve. In contrast, it is interest that the region-merging time of the asymptotic method is merely 0.0287 s, achieving 24 times faster than 0.68 s of the global MaxCov. For convenience, the steps of the asymptotic method are also summarized in Algorithm 2, entitled "Asymptotic MaxCov".

Algorithm 2: Asymptotic MaxCov

Input : Given k base regions $\{R_i^{(1)}\}_{i=1}^k$ and ϵ;
Output: Maximum coverage Ω.

1 *Initialization*. Initialize Ω, $\mathbf{M}$ and $\{\mathbf{R}_i^{(1)}\}_{i=1}^k$; Obtain a real sequence $\{\epsilon_j\}_{j=1}^n$ w.r.t ϵ, and let $j = 1$;
2 **repeat**
3 Call Algorithm 1 for ϵ_j-MaxCov and return Ω_j ;
4 Update Ω, $\mathbf{M}$ and $\{\mathbf{R}_i^{(1)}\}_{i=1}^{k_j}$ by the obtained Ω_j ;
5 Update $j \leftarrow (j + 1)$;
6 **until** $j <= n$;
7 Return Ω ;

3.3.3. Preprocessing for redundant regions

Generally, people prefer a large k for superpixel segmentation to decrease under-segmentation, and at the same time obtain better boundary adherence. Thus redundant superpixels (base regions) will be generated, especially in some large-area image objects, e.g., ground, farmland or sky. They are similar to each other in both color values and spatial locations. Furthermore, they almost have no contribution to region-merging, but on the contrary, sharply deepen the depth of the virtual-merging. Therefore, it is necessary to provide a preprocessing step to remove redundancy before region-merging.

Instead of *global* (absolute) variance used in Hu et al. (2017), here we introduce a *local* (relative) one, aiming at simultaneously removing redundant regions from multiple objects. For a given base region, saying $\mathbf{R}_i^{(1)}$, the local variance is defined in (10).

$$v_i^2 = \frac{1}{|Ne(\mathbf{R}_i^{(1)})|} \sum_{j \in Ne(\mathbf{R}_i^{(1)})} \left\| \mathbf{R}_j^{(1)} - \mathbf{R}_i^{(1)} \right\|^2 \tag{10}$$

where $Ne(\mathbf{R}_i^{(1)})$ denotes the neighbors of $\mathbf{R}_i^{(1)}$.

As shown in Fig. 9, redundancies have been filtered to form some larger-size regions, each of which containing a red ellipse. Then these obtained large-size regions can also be treated as base regions and directly used for the MaxCov-merging. Intuitively, this preprocessing step is very efficient for dropping the number of base regions, and thus sharply decreasing the depth of the virtual-merging. In this example, with the help of preprocessing, the number of base regions drops from the initial 500 to the remaining 308. More extensive experiments will be provided in next section.

4. Experimental simulation

To evaluate the effectiveness of the proposed method, the experiments are divided into two parts. The first one is to show its performance on multi-scale RS images. Due to high-resolution and no ground-truth in a real application, the results are shown in visualization. The second focuses on the quality evaluation on the labeled public images. We take four clustering based methods as baselines, i.e., DBSCAN-, AP-, HC-based region-merging and mean shift clustering (Wang et al., 2022). The data used in this section consists of our collected images and public image database, such as Berkeley Segmentation Data Set (BSD500) (Arbelaez et al., 2011).[1] The base regions (superpixels) for all images are obtained by SLIC segmentation. The comparison in this section is conducted on a Dell PC, with a 2.83 GHz Intel Core 2 Quad CPU (4G RAM), running Matlab 2017b in a Windows 7 operating system.

4.1. Experiments on RS images

As aforementioned, a significant difference from the existing methods is that our MaxCov is proposed in global view. To verify this globality, the experiment is carried on three types of RS images, as shown in Fig. 10.

[1] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html#bsds500.

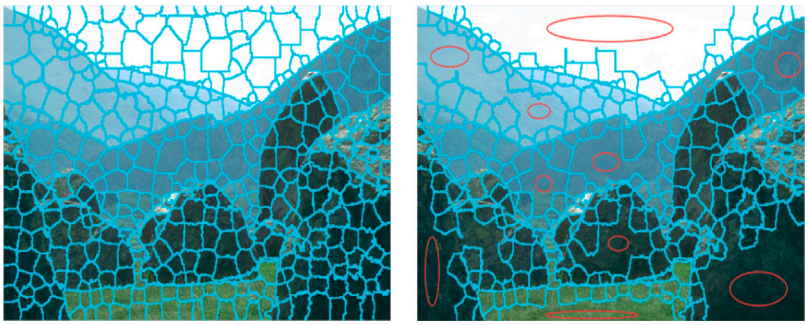


Fig. 9. An example for the preprocessing. The preprocessing result of initial 500 SLIC superpixels (left) is shown in the right panel, in which a redundancy in the same object has been merged into a large-size region, marked with a red ellipse.

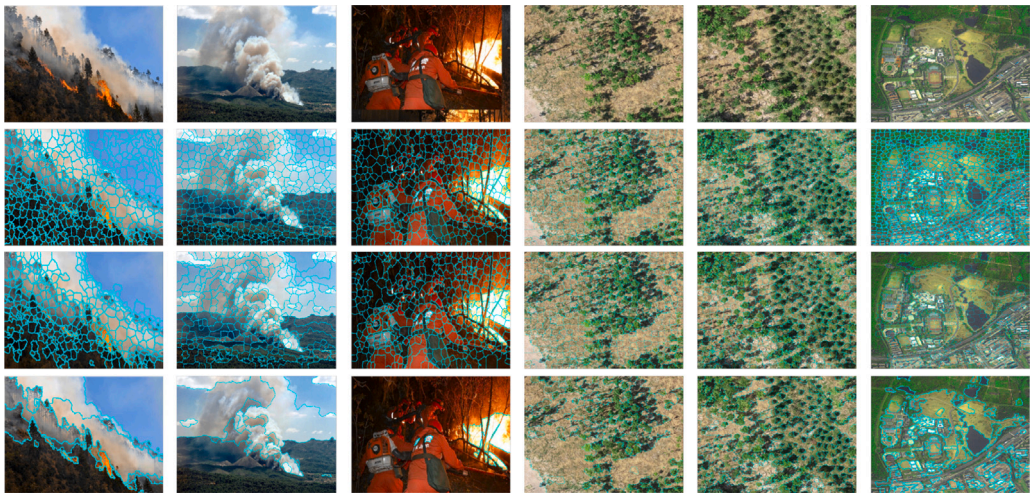


Fig. 10. The visualization for the proposed MaxCov method on multi-scale RS images. From top to bottom, the figures are original images, SLIC segmentations, the preprocessing, and the final results of MaxCov region-merging, respectively.

The selected RS images are composed of three groups, i.e., including three ground-monitoring forest fire/smoke images (Qian et al., 2023), two UAV-based images for forest resources inventory, and one satellite-based Landsat imagery. From left to right, we name them by using RS1 ∼ RS6 respectively. After SLIC segmentation, the preprocessing step is adopted to filter redundant regions. Then MaxCov-merging is executed on the preprocessed regions. At last, the final region-merging results are shown in the bottom line. At the first glance, the base regions in the first group have been well-merged, and the connected object regions or sub-regions, e.g., the regions of the objects blue-sky, orange-fire-flame and pale-white-smoke, can be separated from each other. While for the last two, it is different. Due to the complexity of real applications themselves, sometimes it is difficult or even impossible to obtain all object regions simultaneously just by a uniform scale parameter ϵ. In this case, it seems better if one can provide more information for a desire object in the unsupervised or supervised manner, e.g., providing a more accurate ϵ, adding a preprocessing step like image stretching, or collecting supervision information like land-cover classification (Tong et al., 2020). In this paper, we follow the unsupervised learning, as the baselines do. Here those objects, such as tree canopies or shadows in the second group, and the lakes or ponds in the third one, are chosen as the desired objects. As for the speed-up strategies used for MaxCov, we report the executing time in Table 1, with or without these heuristics. The result shows that the MaxCov-merging using the proposed speed-up strategy ("with") runs dozens of times faster than the one not using this strategy ("without"), e.g., the average time (Aver) in Table 1.

Due to the expensive time-consuming, especially for the AP and HC clustering-based methods, here we just verify the performance of our proposed MaxCov on high-resolution RS images. In next subsection, we will provide an extensive comparison on multiple quantitative indicators.

4.2. Experiments on BSD500

Arguably, the most important property of SLIC is boundary adherence, commonly measured by *boundary recall (BR)* and *corrected under segmentation error (CUE)* (Michael et al., 2013; Yan and Sun, 2016; Guan et al., 2022). Both *BR* and *CUE* are defined as:

$$\begin{aligned} BR &= \frac{\sum_{\mathbf{p}\in B(g)} |\{\mathbf{p}|\ \|\mathbf{p}-\mathbf{q}\| \le 2, \mathbf{q}\in B(s)\}|}{|B(g)|} \\ CUE &= \frac{\sum_j |s_j - g_{\max}(s_j)|}{\sum_i |g_i|} \end{aligned} \tag{11}$$

where $s = \cup_j s_j$ and $g = \cup_i g_i$, denoting the union of disjoint regions, obtained from final-merging or ground truth segmentation, respectively. Here $g_{\max}(s_j) = \arg\max_i |s_j \cap g_i|$. $B(\cdot)$ denotes the boundary of a region, which is also a pixel set, composed of the pixels located at the boundary, e.g., $\mathbf{p}$ or $\mathbf{q}$ in (11). $|\cdot|$ denotes the cardinality of a set, i.e., the number of elements if the given set is finite.

For the SLIC segmentation, if increasing k, then boundary adherence tends to be better. While for region-merging, we need another indicator, named *Not-Boundary-Rate (NBR)*, to measure object boundary adherence. It is defined in (12):

$$NBR = \frac{|B(s) \cap \overline{B(g)}|}{|\overline{B(g)}|} \tag{12}$$

where $\overline{B}$ denotes the complementary set of B.

Table 1
The comparison on the executing time between the methods "with" or "without" heuristic strategies (in seconds).

Image	RS1	RS2	RS3	RS4	RS5	RS6	**Aver**
Size	512 × 288	677 × 1024	514 × 900	2400 × 2800	2400 × 2800	2560 × 2560	
With	1.7	4.9	3.7	7.1	7.8	15.1	6.7
Without	134.3	149.2	38.9	75.4	38.1	425.4	143.6

Fig. 11. Explanation for NBR on the final region-merging results, obtained by our MaxCov, HC, DBSCAN, AP and mean shift, respectively. In the colored boundaries, the red one is for region-merging, and the green one with a triple pixel-width is for the ground truth.

NBR measures what fraction of boundaries of final merged regions fall out of the boundaries of the given ground truth. That is, for a good region-merging method, it should be able to achieve a high *BR* and a low *NBR* simultaneously. Geometrically, this means that very few true object boundaries are missed. Fig. 11 demonstrates an explanation for the indicators *BR* and *NBR*, in which two kinds of colored boundaries are used: the green one with a triple pixel-width is for the ground truth, and the red boundary with a normal pixel-width is for the region-merging. The figures from left to right show the final region-merging results of MaxCov and baselines. The last three results show that the red boundaries but not overlapping with the green ones are not object boundaries, which can be measured by *NBR*, according to the definition (12). Intuitively, both MaxCov and HC outperform other methods in object boundary adherence, where the object boundaries obtained from the first two methods adhere to the ground truth very well. Actually, these *BR*s are very similar, achieving 85.585, 85.585, 87.097, 85.887 and 86.452 (in percentage, %), respectively. However, reflecting in *NBR* they are quite different. For example, in this case, their *NBR*s are 2.360%, 2.360%, 10.076%, 8.690% and 7.850%, respectively, which means that the *NBR* of the first two is significantly lower than that of the rest methods. Although without considering *CUE*, it seems effective to measure object boundary adherence by the indicators *BR* and *NBR*. Next, the performance of region-merging will be quantitatively measured by these indicators.

For visualization, the selected images from the BSDS database can be divided into two groups. The first group is composed of simple images, containing one or two objects, as shown the first four rows in Fig. 12; The second is more complex, in which each image has at least two more objects. Here the number of object regions is determined by the given ground truth (GT). For easy-reading, a boundary of GT, obtained from the first cell of MAT file of BSDS database, is shown in green lines with the triple pixel-width. In terms of parameter setting, there is merely one parameter ϵ for MaxCov-merging, which can be adaptively determined by superpixel distances, as explained in Fig. 8. For convenience of readers, we also provide our MATLAB codes and high-resolution figures, available at https://github.com/xbyang1000/Region-merging. For the HC-based region-merging,[2] the parameter "Num_of_region" is predefined by GT, and the remaining parameters are default, including the criterion, shape and compactness (Hu et al., 2017). With the suggestion of Peter Kovesi,[3] the parameter $minPt$ for DBSCAN is set to 1. Similarly, the parameters of AP-merging, e.g., the weights w_l, w_a, w_b, and radius $colorradius$, are selected from the set $\{3, 10, 10, 20\}$, following the suggestion of Ref. Zhou (2015). For the method Ref. Wang et al. (2022), the radius h to the mean shift is suggested to tune in the range of (0,30), then both SLIC segmentation and data clustering are carried on the color and position features. Different from Ref. Wang et al. (2022), here we do not consider other auxiliary information as used for gas ash microscopic image segmentation, e.g., textures and the transformed features, for fairness. In this work, SLIC superpixels are solved by the MATLAB function "superpixels", with the initial k at 500 or 1000 for a better boundary adherence to the ground truth, and then used as the input of all region-merging. The final results are shown in Fig. 12 for visualization, and in Table 2 for numeric values quantified by the indicators *BR* (%), *CUE* (%), *NBR* (%), and the executing time (shortly *Time*, in seconds). Here "*Time*" refers to the total time of parameter selection and region-merging, excluding the time of SLIC segmentation and visualization.

As shown in Fig. 12, both our MaxCov and HC-based methods have better object boundary adherences than the other two, where most objects can be well-merged. Especially, the globality of MaxCov can be found in the images "3096", "8068", "86 016", "353 013", and "100 007", where the superpixels of the objects like "sky", "goose", "bird and branch", "parterre" and "snowed or iced ground" are merged well. Furthermore, some superpixels not belonging to the ground truth, e.g., the plate for bonsai in "353 013" and the boat logos in "384 022", are also preserved in the final region-merging; On the contrary, there also exist those superpixels, which are highlighted by GT but missed by region-merging methods, e.g., the legs of the dragonfly in "35 070", the tile contours of the parterre in "86 016", and the wooden stakes in "196 027". That is, in contrast to the human annotated ground truth, a unideal final result is not necessarily caused by region-merging methods, but by SLIC. The reason is that region-merging is just a post-processing step of superpixel segmentation, which performance is essentially determined by SLIC superpixels. For example, on "86 016" the highest *BR* and the lowest *CUE* are merely 19.263% and 59.909%, which means that many pixels of GT boundaries are missed by SLIC segmentation, as shown the visual evidence in Fig. 12. Reflecting in the post-processing step, this missing would avoidably affect and thus result in a bad region-merging. Therefore, the responsibility between SLIC and the next region-merging can be distinguished by the indicators *BR* and *CUE*. On the other hand, among the region-merging methods, if simultaneously achieving a higher *BR* and a lower *CUE*, it means that SLIC superpixels are efficient towards GT. In this situation, the lower *NBR* (the error of region-merging) achieves, the better quality of region-merging becomes. Table 2 shows that DBSCAN achieves an average *BR* at 71.643% and *CUE* at 9.384%. However, it also achieves the highest average *NBR* at 16.074%, which declares its failure in the final region-merging. As shown in Fig. 12, there are so many final regions in the

[2] C++ code with MATLAB interfaces, available at https://github.com/zwhoo/Scale-Sets-Image-Analysis-Toolkit.

[3] Available at https://www.peterkovesi.com/projects/segmentation/.

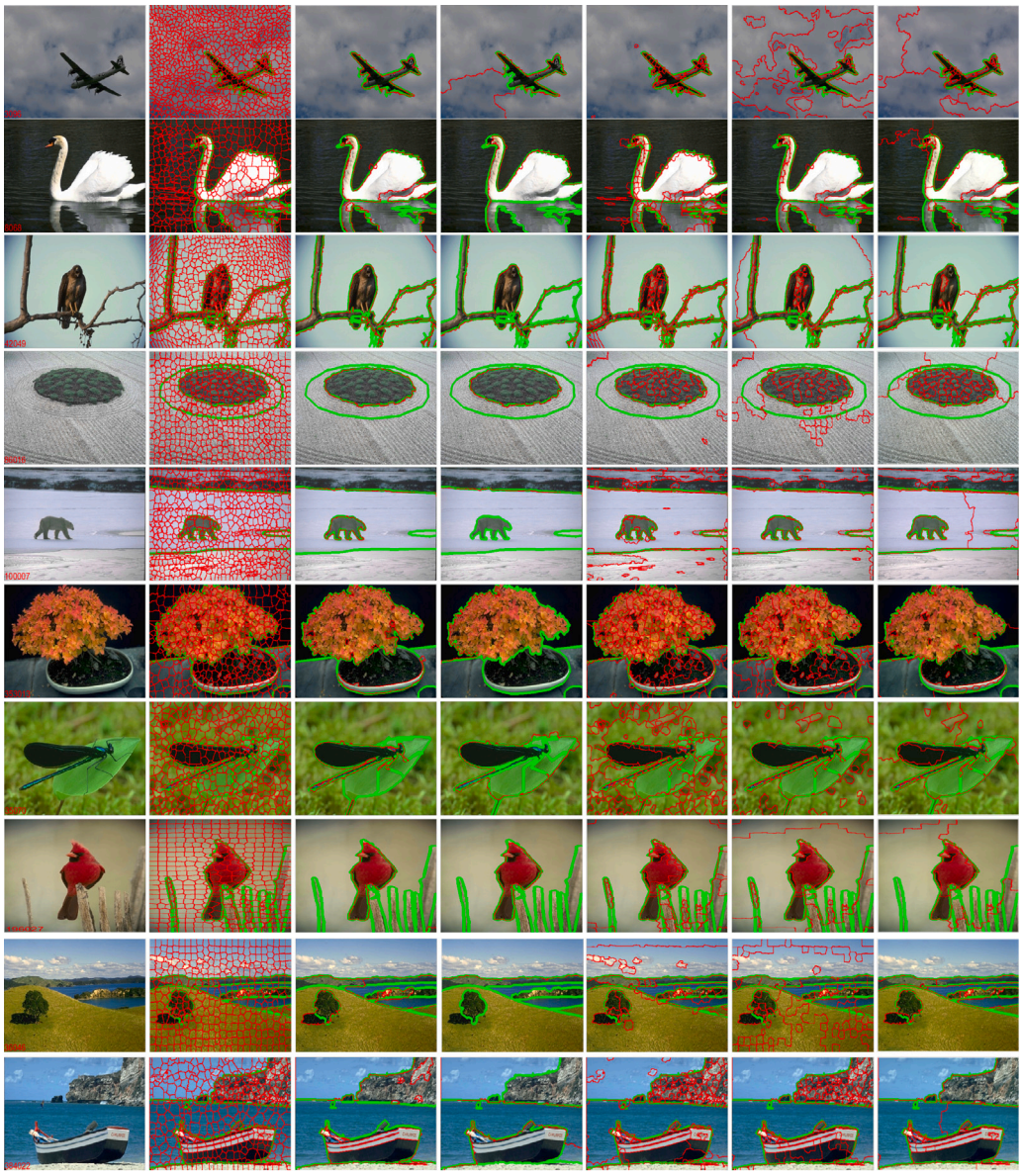


Fig. 12. Visualization for the region-merging on BSDS images. The leftmost two columns are original images and SLIC superpixels. The rest five are region-merging results respectively obtained by MaxCov-, HC-, DBSCAN-, AP-based and Mean Shift methods, in which the ground truth is shown in triple-pixel-width green lines, while the boundary of final region-region is shown in red lines.

DBSCAN- and AP-merging needed to continue merging. In order to show the performance of object boundary adherence more intuitively, a scoring indicator is constructed on the boundary indicators *BR* and *NBR*, i.e., $Score = \beta(1 - NBR) + (1 - \beta)BR$, where $0 \leq \beta \leq 1$. For example, letting $\beta = 0.8$ to pay more attention on the adherence of object boundary than that of superpixel boundary. Table 2 states that both MaxCov and HC significantly outperform the other two methods. Concretely, although HC achieves a lower average *NBR* than MaxCov, e.g., 2.362 vs. 4.570, it fails to capture more object boundaries than MaxCov. It is worth mentioning that the *NBR* of MaxCov is slightly decreased by 2.208%, but the *BR* is significantly increased by 20.102%. As for mean shift based method, although there are one or two cases that look good, e.g., the result of "36 046" or "384 022", the overall performance is still poor. In term of region-merging time, DBSCAN-clustering based method wins the best, achieving an average time at 1.612 s; Then HC- and our MaxCov-merging methods follow, achieving the time at 2.129 and 2.641 s, respectively. While for AP-clustering, it needs 29.537 s, which is dozens of times slower than other methods, including the time spent on the parameter-tuning for w_l, w_a, w_b, and $colorradius$, respectively. Finally, we also conduct the experiment on entire BSD500 dataset, and report the averaged results in Table 3. It also shows that MaxCov can achieve the highest score of all methods.

The conclusions are summarized as follows: (1) There still has a large gap between the human-annotated ground-truth and computer-aided region-merging methods. An evidence is from the tile contours in "86 016", which is easily captured by human vision, but missed by SLIC segmentation; (2) A unideal result is not necessarily caused by region-merging, e.g., by the difference between computer-vision and human-vision; (3) The error-merging problem may appear when a pair of candidate regions has a high similarity in both color and spacial proximities; (4) As shown in Fig. 10, it would be better if using multiple scale parameters for region-merging, as the explanation for the

Table 2
Numeric results of the comparison between our MaxCov and baseline methods.

Image ID	MaxCov	HC	DBSCAN	AP	Mean-shift
	BR(%)/*CUE*(%) *NBR*(%)/*Time* (s)				
3096	68.977/0.979 2.450/1.827	62.367/5.756 3.461/1.322	81.343/0.829 4.630/2.694	73.667/0.968 16.024/16.625	75.484/0.824 10.454/3.458
8068	52.564/5.893 6.132/1.782	27.107/24.686 3.033/1.072	72.893/3.031 14.783/1.590	58.761/12.174 8.155/13.136	61.234/5.688 18.548/8.450
42 049	68.216/18.107 7.268/1.110	21.635/44.895 2.480/1.348	78.548/10.196 15.918/1.444	70.708/42.613 12.834/14.360	74.335/22.587 13.554/6.453
86 016	15.161/59.909 1.925/1.180	15.161/59.909 1.925/1.573	16.968/59.220 7.944/1.127	19.263/56.943 11.596/21.284	20.744/58.332 10.744/9.271
100 007	49.631/27.541 2.340/1.565	25.769/30.811 1.080/1.599	87.946/1.264 16.622/1.284	85.240/8.236 10.264/21.630	88.369/8.632 18.547/5.655
353 013	58.704/43.311 6.967/1.689	44.844/45.077 3.818/1.606	80.984/2.556 29.340/1.130	79.707/25.894 27.084/25.527	74.326/32.548 10.250/12.231
35 070	49.719/16.103 3.771/1.600	36.758/16.405 2.577/1.084	79.053/5.087 27.026/1.092	72.512/12.850 14.139/42.018	70.654/9.564 15.899/8.563
196 027	22.786/18.197 1.945/2.544	22.786/18.197 1.945/1.729	54.681/7.705 9.915/2.536	49.540/18.045 8.529/31.574	45.322/12.438 9.587/9.420
36 046	53.926/16.943 4.323/8.604	20.563/22.752 1.089/4.553	79.301/2.282 12.641/1.034	67.726/19.519 17.467/46.594	55.365/15.471 3.647/9.655
384 022	54.854/19.267 8.581/4.512	16.262/55.078 2.209/5.400	84.709/1.670 21.946/2.187	81.019/13.056 20.193/62.623	58.658/16.847 12.480/22.365
Average	49.454/22.625 4.570/2.641	29.325/30.357 2.362/2.129	71.643/9.384 16.074/1.612	65.814/20.030 14.624/29.537	62.449/18.293 11.471/9.556
Score	**86.234**	83.976	81.467	81.460	82.433

Table 3
Averaged results on entire BSD500 dataset when the number of superpixel is 500.

	MaxCov	HC	DBSCAN	AP	Mean shift
BR (%)	48.659	25.594	76.349	68.114	65.560
CUE (%)	22.569	32.548	11.548	22.364	20.867
NBR (%)	6.251	4.584	19.565	15.483	18.785
Time (s)	3.684	2.561	1.855	38.457	10.240
Score:	**84.731**	81.480	79.618	81.236	78.084

subsequence $\{\epsilon_i\}$ in Section 3, especially in high-resolution RS images with complex objects. Actually, it is difficult and also unnecessary to determine a set of accurate $\{\epsilon_i\}$, because that, for a specified application, users just care about one or two interest objects, e.g. the fire flame in a forest fire video-monitoring system. For ease of use, the scale parameters $\{\epsilon_i\}$ can also be set to a monotone sequence with a fixed step.

5. Conclusion

In this paper, a global region-merging method, termed as MaxCov, is proposed. To capture the integrity of a given image object, MaxCov aims to do region-merging between similar pairs of regions, but not the most similar ones as the clustering-based methods do. It consists of two steps: bottom-up virtual-merging and up-bottom real-merging. The former is used to collect candidate region pairs; While the latter is for decision-making in global view. The similarity of MaxCov is measured by color values and the adjacency of regions. Inspired by the set covering principle, the region-merging problem is converted into computing set maximum coverage. The related theoretical evidences, e.g., the existence of maximum coverage and the asymptotic behavior of MaxCov, are also provided. Visual and quantitative comparisons show the effectiveness of our MaxCov in terms of *NBR*, *BR* and *CUE*.

It is worthy-mentioning that, for fairness, this work originates from unsupervised learning, but it is easily extended into supervised learning cases, e.g., using a supervised ϵ for the desired object, or a supervised strategy for redundant region elimination. Intuitively, this should be capable of improving the quality and time of region-merging. Some new methods for superpixel generation and region-merging, e.g., Refs. Ma et al. (2022) and Xu et al. (2023), are also worth paying attention to. Due to space limitation, here we do not provide more verifications. The result on RS images reminds us that this method may be efficient for pixel annotation, e.g., annotating the pixels of forestry objects "fire", "smoke", and "tree canopy" in batch (Zhu et al., 2023; Rong et al., 2023). Additionally, what the result will be if we utilize the rapidity of DBSCAN-clustering, and control the early-stopping problem in DBSCAN-merging, e.g., taking the DBSCAN's output as MaxCov's input. Our future work also includes the extension to other superpixel segmentation/generation techniques.

CRediT authorship contribution statement

Xubing Yang: Conceptualization, Methodology, Paper writing. **Zhengxiao Zhang:** Experiment and programming. **Li Zhang:** Investigation and data visualization, Investigation. **Xijian Fan:** Experimental comparison, Validation, Supervision. **Qiaolin Ye:** Algorithm examination. **Liyong Fu:** Writing – review & editing.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

This research was supported in part by National Key Research and Development Program of China under Grant. 2022YFD2201005-03, and in part by the National Science Foundation of China under Grants 62072246 and 32371877.

Appendix

A.1. Interpretation for pixel posteriori probability

Proof. Let $P(\omega_j|\mathbf{z})$ and $p(\omega_j|\mathbf{z})$ be the posteriori probability and density of the j-object, respectively, we have

$$p(\omega_j|\mathbf{z}) = p(\omega_j)P(\omega_j|\mathbf{z}) \tag{13}$$

From (4), for a given pixel $\mathbf{z}$, which object it should belong to is determined by the maximum posteriori probability $P(\omega_i|\mathbf{z}), i = 1, 2, \dots, K$. That is, if $\mathbf{z} \in \omega_j$, $P(\omega_j|\mathbf{z})$ should be the maximum value in K posteriori probabilities, i.e.,

$$j = \arg\max_{1\le i\le K} P(\omega_i|\mathbf{z}), \quad P(\omega_i|\mathbf{z}) = \frac{P(\omega_i)p(\mathbf{z}|\omega_i;\boldsymbol{\theta}_i)}{P(\mathbf{z})} \tag{14}$$

where $P(\mathbf{z})$ is defined in (4), and $\boldsymbol{\theta}_i$ is the ith parameter to be determined.

After image segmentation, these parameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K$ can be viewed as confirmed constants because they can be estimated from the pixels of the segmented objects, e.g., by MLE (Maximum Likelihood Estimation) or EM (Expectation Maximization) algorithm (Nguyen et al., 2020). Thus we can rewrite $p(\mathbf{z}|\omega_i;\boldsymbol{\theta}_i)$ as $p(\mathbf{z}|\omega_i)$ for the simplicity. Suppose the given image has been well-segmented, e.g., by SLIC technology, this means that the boundary of each region adheres to the corresponding object or sub-object well. In this case, K (the number of objects) is fixed, though it may have slight difference from human-vision. If $\mathbf{z}$ is varied in the pixel domain of the given image, thus it must belong to the one of K objects, and in this case we have $P(\mathbf{z}) = 1$. Substitute it into (14) and replace $p(\mathbf{z}|\omega_i;\boldsymbol{\theta}_i)$ with $p(\mathbf{z}|\omega_i)$, thus (5) holds, i.e., we have $P(\omega_j|\mathbf{z}) \propto P(\omega_j)p(\mathbf{z}|\omega_j)$, where the operator $\propto$ denotes positive proportion. ■

A.2. The proof to Theorem 1

Proof. In addition to the previous notations, it is obvious that if K equals to k, Theorem 1 holds because in this case, there has no over-segmentation problem, and the union of superpixels is the desired coverage. Next the proof is given when $K < k$. For easy reading, we divide the proof into two cases.

- *Case 1: All object-regions are connected.*

Given K objects, the maximum coverage is defined as

$$Cov = \{R_{i_1}^{(l_1)}, \dots, R_{i_K}^{(l_K)}\} \tag{15}$$

where $\sum_{i=1}^{K} l_i = k$.

Since it is a coverage, Cov should satisfy (16) and (17).

$$\begin{aligned} &R_{i_j}^{(l_j)} \cap R_{i_m}^{(l_k)} = \varnothing \\ &\forall j, m \in \{1, \dots, K\}, j \neq m \end{aligned} \tag{16}$$

and

$$\cup_{j=1}^{K} R_{i_j}^{(l_j)} = I \tag{17}$$

where $R_{i_j}^{(l_j)}$ denotes the i_jth real-merging element in the l_j-layer. Both l_j and i_j are index indicators, i.e. $l_j \in \{1, \dots, t-1\}$, and $i_j \in \{1, \dots, l_j^{(max)}\}$, where $l_j^{(max)}$ denotes the number of elements in the l_j-layer. Next we prove the existence for the Cov.

As aforementioned, the union of all base regions can also be viewed as a coverage, i.e.,

$$I = \cup_{i=1}^{k} R_i^{(1)}. \tag{18}$$

Thus the current coverage Cov is an ordered set of the base regions, denoted by S,

$$\begin{aligned} S = \Big\{ &R_{j_1}^{(1)}, \dots, R_{jl_1}^{(1)}, R_{j,(l_1+1)}^{(1)}, \dots, R_{j,(l_1+l_2)}^{(1)}, \\ &\cdots, R_{j,(l_1+\cdots+l_{K-1}+1)}^{(1)}, \dots, R_{j,(l_1+\cdots+l_{K-1}+l_K)}^{(1)} \Big\} \end{aligned} \tag{19}$$

Without loss of generality, assume $R_{i_1}^{(l_1)}$ is formed by the first l_1 units in S, $R_{i_2}^{(l_2)}$ is by the next l_2 units, $\cdots$, and $R_{i_K}^{(l_K)}$ is by the last l_K ones. Because they can be merged sequentially, we will obtain a strictly monotonically decreasing sequence, denoted by $S, S_1, S_2, \dots, S_n$, where $|S| > |S_1| > |S_2| > \cdots > |S_n|$. By at most $(k - K)$ times merging, the sequences will be convergent to Cov.

- *Case 2: Some object-regions are disconnected.*

Suppose there are m disconnected object-regions. For the ith one, let i_K be the number of the connected sub-object-regions, $i = 1, 2, \dots, m$. Thus we have a total of $(K - m + \sum_{i=1}^{m} i_K)$ connected object regions or sub-object regions.

In this case, since all object regions can be viewed as connected, e.g., letting $K = K - m + \sum_{i=1}^{m} i_K$. Then according to *Case 1*, we can also obtain a sequence convergent to the K objects, in which each disconnected object region can be described by the union of the corresponding multiple connected sub-objects.

In summary, no matter what the object-region is, connected or disconnected, there always exist a sequence convergent to Cov, i.e. the existence for the maximum coverage. ■

A.3. The proof to Theorem 2

Proof. In addition to the previous notations. For the given positive value ϵ, it is easy to know that there must exist a monotonic real sequence $\{\epsilon_i\}$, satisfying $0 \le \epsilon_1 \le \cdots \le \epsilon_i \cdots \le \epsilon$. For example, let $\epsilon_i = i * \epsilon/n$, where n is the number of the equal division of the interval $[0, \epsilon]$. Then a sequence of nested intervals, $(I_j)_{j\in\mathbb{N}}$, $I_j = [0, \epsilon_j]$, can be obtained, which satisfies the following conditions:

$$\begin{aligned} &(1).\ \forall i \in \mathbb{N} : I_i \subseteq I_{i+1} \\ &(2).\ \forall \varepsilon > 0,\ \exists N \in \mathbb{N} :\ |I_N| = \varepsilon \end{aligned} \tag{20}$$

Note that the sequence $\{\epsilon_i\}$ is also bounded by ϵ. Therefore, there must exist such an $N \in \mathbb{N}$, such that, when $n > N$, $\{\epsilon_n\} \to \epsilon$ holds. In this case, the nested intervals will be convergent to the interval $[0, \epsilon]$. ■

References

Achanta, R., Shaji, A., Smith, K., et al., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2274–2282.

Akyilmaz, E., Leloglu, U., 2016. Segmentation of SAR images using similarity ratios for generating and clustering superpixels. Electron. Lett. 52 (8), 654–656.

Arbelaez, P., Maire, M., Fowlkes, C., et al., 2011. Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (5), 898–916.

Chen, J., Li, Z., Huang, B., 2017. Linear spectral clustering superpixel. IEEE Trans. Image Process. 26 (7), 3317–3330.

Derksen, D., Inglada, J., Michel, J., 2019. Scaling up SLIC superpixels using a tile-based approach. IEEE Trans. Geosci. Remote Sens. 57 (5), 3073–3085.

Guan, Z., Miao, X., Mu, Y., et al., 2022. Forest fire segmentation from aerial imagery data using an improved instance segmentation model. Remote Sens. 14 (13), 3159.

Hadavand, A., Saadatseresht, M., Homayouni, S., et al., 2019. A novel density-based super-pixel aggregation for automatic segmentation of remote sensing images in urban areas. Earth Observ. Geomat. Eng. 84–91.

Hu, Z., Li, Q., Zhang, Q., et al., 2017. Unsupervised simplification of image hierarchies via evolution analysis in scale-sets framework. IEEE Trans. Image Process. 26 (5), 2394–2407.

Lei, T., Jia, X., Liu, T., et al., 2019. Adaptive morphological reconstruction for seeded image segmentation. IEEE Trans. Image Process. 28 (11), 5510–5523.

Levinshtein, A., Stere, A., Kutulakos, K., et al., 2009. Turbopixels: Fast superpixels using geometric flows. IEEE Trans. Pattern Anal. Mach. Intell. 31 (12), 2290–2297.

Li, H., Feng, R., Wang, L., et al., 2021. Superpixel-based reweighted low-rank and total variation sparse unmixing for hyperspectral remote sensing imagery. IEEE Trans. Geosci. Remote Sens. 59 (1), 629–647.

Ma, F., Zhang, F., Xiang, D., et al., 2022. Fast task-specific region merging for SAR image segmentation. IEEE Trans. Geosci. Remote Sens. 60, 5222316.1–5222316.16.

Martins, J., Menezes, G., Gonçalves, W., et al., 2021. Machine learning and SLIC for tree canopies segmentation in urban areas. Ecol. Inform. 66, 101465.

Michael, V.D.B., Boix, X., Roig, G., et al., 2013. SEEDS: Superpixels extracted via energy-driven sampling. Int. J. Comput. Vis. 111 (3), 298–314.

Nguyen, T., Nguyen, D., Chamroukhi, F., et al., 2020. Approximation by finite mixtures of continuous density functions that vanish at infinity. Cogent Math. Stat. 7 (1), 1750861.

Nikou, C., Likas, A., Galatsanos, N., 2010. A Bayesian framework for image segmentation with spatially varying mixtures. IEEE Trans. Image Process. 19 (9), 2278–2289.

Nock, R., Nielsen, F., 2004. Statistical region merging. IEEE Trans. Pattern Anal. Mach. Intell. 26 (11), 1452–1458.

Qian, J., Lin, J., Bai, D., et al., 2023. Omni-dimensional dynamic convolution meets bottleneck transformer: a novel improved high accuracy forest fire smoke detection model. Forests 14 (4), 838.

Rong, Q., Hu, C., Hu, X., et al., 2023. Picking point recognition for ripe tomatoes using semantic segmentation and morphological processing. Comput. Electron. Agric. 210, 107923.

Roux, M., 2018. A comparative study of divisive and agglomerative hierarchical clustering algorithms. J. Classification 35 (2), 345–366.

Sanjay-Gopal, S., Hebert, T., 1998. Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm. IEEE Trans. Image Process. 7 (7), 1014–1028.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), T888–905.

Stutz, D., Hermans, A., Leibe, B., 2018. Superpixels: An evaluation of the state-of-the-art. Comput. Vis. Image Underst. 166, 1–27.

Tong, X., Xia, G., Lu, Q., et al., 2020. Land-cover classification with high-resolution remote sensing images using transferable deep models. Remote Sens. Environ. 237, 111322.

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. 13 (6), 583–598.

Wang, P., Ji, J., Zhang, K., et al., 2022. Gas ash microscopic image segmentation with SLIC and mean shift. In: The 34th Chinese Control and Decision Conference (CCDC), Hefei, China. pp. 2762–2767. http://dx.doi.org/10.1109/CCDC55256.2022.10034284.

Wu, X., Shi, Z., Zou, Z., 2021. A geographic information-driven method and a new large scale dataset for remote sensing cloud/snow detection. ISPRS J. Photogramm. Remote Sens. 74, 87–104.

Wu, H., Wu, Y., Zhang, S., et al., 2016. Cartoon image segmentation based on improved SLIC superpixels and adaptive region propagation merging. In: IEEE International Conference on Signal and Image Processing (ICSIP). pp. 277–281.

Xiang, D., Zhang, F., Zhang, W., et al., 2021. Fast pixel-superpixel region merging for SAR image segmentation. IEEE Trans. Geosci. Remote Sens. 59 (11), 9319–9335.

Xu, Y., Gao, X., Zhang, C., et al., 2023. High quality superpixel generation through regional decomposition. IEEE Trans. Circuits Syst. Video Technol. 33 (4), 1802–1815.

Yan, Y., Sun, J., 2016. An improved-SLIC algorithm based on regional re-division. Laser J. 37 (8), 129–133, (Chinese paper with English abstract).

Yuan, X., Shi, J., Gu, L., 2021. A review of deep learning methods for semantic segmentation of remote sensing imagery. Expert Syst. Appl. 169, 114417.

Zhang, Y., Liu, K., Dong, Y., et al., 2020. Semisupervised classification based on SLIC segmentation for hyperspectral image. IEEE Geosci. Remote Sens. Lett. 17 (8), 1440–1444.

Zhou, B., 2015. Image segmentation using SLIC superpixels and affinity propagation clustering. Int. J. Sci. Res. 4 (4), 1525–1529.

Zhou, Y., Yun, T., 2020. Individual tree crown segmentation based on aerial image using superpixel and topological features. J. Appl. Remote Sens. 14 (2), 022210.

Zhu, Y., Li, D., Fan, J., et al., 2023. A reinterpretation of the gap fraction of tree crowns from the perspectives of computer graphics and porous media theory. Front. Plant Sci. 14, 1109443.