



Automatic image segmentation based on label propagation

Ivar Vargas Belizario  | Oscar Cuadros Linares  | João do Espirito Santo Batista Neto

Institute of Mathematics and Computer Science
(ICMC), University of São Paulo, São Carlos, Brazil

Correspondence

Ivar Vargas Belizario, Institute of Mathematics and
Computer Science (ICMC), University of São Paulo,
São Carlos, Brazil. Email: ivar@usp.br

Funding information

FAPESP, Grant/Award Numbers: 2018/06074-0,
2021/00360-3; Coordenação de Aperfeiçoamento de
Pessoal de Nível Superior, Grant/Award Number:
Finance Code 001

Abstract

This article introduces an automatic approach for the segmentation of coloured natural scene images based on graphs and the propagation of labels originally designed for communities detection in complex networks. Images are initially pre-segmented with super-pixels, followed by feature extraction using colour information of each super-pixels. The resulting graph consists of vertices which represent super-pixels, whereas the edge weights are a measure of similarity between super-pixels. The resulting segmentation corresponds to the propagation of labels among the vertices. In this article, three strategies for propagating labels have been formulated: (i) iterative propagation (ILP), (ii) recursive propagation (RLP) and (iii) a weighted recursive propagation (WRLP). The experiments have shown that the proposed methods, when compared to other state-of-the-art methods, produce better results in terms of segmentation quality and processing time.

1 | INTRODUCTION

Automatic image segmentation is still considered an open problem, for its importance in tasks such as image classification, object detection, and tracking. Moreover, segmentation is often a subjective task, since the regions of interest or objects in images are defined by the perception of different individuals. Specifically, the automatic image segmentation of natural scenes (e.g. landscapes, cities, roads, and remote sensing images) aims to identify and separate objects of interest (foreground) from the rest of the image (background).

During the last years, several methods for automatic image segmentation have been proposed. These can be based on simple thresholding, region growing and edge detection or more sophisticated methods such as level-set, graph clustering, machine learning, and many hybrid approaches [1–3]. What normally defines the best approach to segmentation is the application. Region growing and level set methods, for example, are often used when the object of interest is known and usually demand user intervention during the process. Threshold and graph clustering methods, on the other hand, are usually used for automatic image segmentation and are capable of determining the foreground and background regions.

For graph-based segmentation approaches, it is crucial to build a graph in which the image is properly represented, that is, all image information necessary for a robust segmentation must

be kept in the graph model. Usually, graph vertices are represented by individual pixels, and edge weights are obtained by measuring the proximity (similarity metrics) among pixels. Segmentation is then achieved by applying graph clustering methods.

If every pixel of an image with n pixels is represented as a vertex in the graph, the number of edges can be $n(n-1)/2$ or even higher. Since the computational complexity of graph clustering methods is related to the graph cardinality, finding the best partition of a graph is an NP-complete problem [4]. Hence, it is crucial to build graphs from images where both the number of vertices and edges are minimum, while guaranteeing segmentation quality. To reduce the computational cost, many works have proposed a pre-segmentation task that segments the image into small regions of homogeneous properties, the so-called super-pixels [5–8]. A graph is then built, in which a vertex represents a super-pixel instead of a single pixel. By using super-pixels, the graph cardinality can be drastically reduced, in some cases up to 90% [5].

There is a vast range of graph clustering methods. We highlight: Normalized Cut [4], Fast Greedy [9] Label Propagation [10], Louvain [11] and Efficient Graph-Based Image Segmentation [12]. Most of these methods were originally proposed for clustering graphs built from various data types, not necessarily images. Many do not take into account the domain information. For example, the label propagation method, was originally

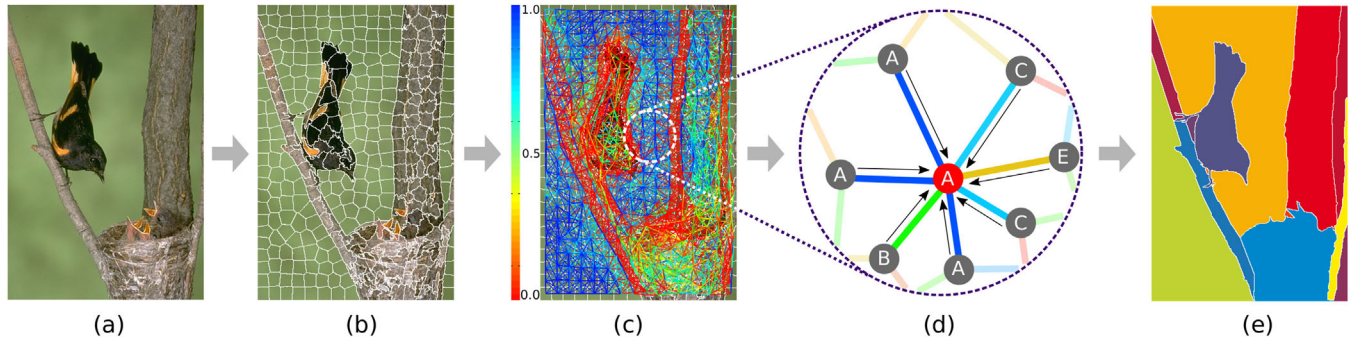


FIGURE 1 Illustration of our method. (a) Input image; (b) super-pixel extraction; (c) graph building. Blue edges indicate high similarity among vertices, whereas red edges represent low similarity; (d) label propagation. The red vertex is assigned the most frequent label in its neighbourhood, (e) segmentation result, that is, regions containing super-pixels with the same label

proposed as a clustering strategy for huge graphs, with more than a million vertices [13]. However, when applied to images, over-segmentation is likely to occur. Besides, it is a non-deterministic approach. Nevertheless, the computational complexity of this method is nearly linear [5], one of the fastest and most appealing methods for very large graphs.

The low computational cost is a potential feature of the label propagation method to generate robust segmentation, as long as the intrinsic features of the image are taken into account. This is the main motivation of this work. We propose a graph-based approach to image segmentation that addresses two challenges: reduce the high graph cardinality and produce high-quality image segmentation using the label propagation strategy.

The main idea of our proposal is illustrated in Figure 1. Given a natural scene image, we extract super-pixels. Next, a graph is created, where super-pixels are mapped onto vertices and the edges represent the similarity among super-pixels. Finally, segmentation is achieved with label propagation clustering techniques.

The main contributions of this work are the following:

- A strategy to build graphs from images in such manner that simplifies the propagation of labels during the clustering process;
- A novel similarity measure to compute edge weights from super-pixels;
- Three deterministic image segmentation methods based on the label propagation strategy.

Applications of the proposed methods are not limited to natural scenes images. Non-coloured medical images such as ultrasounds [14] and X-rays [15, 16] may also be a target. The proposed methods could also be used for the segmentation of 3D magnetic resonance medical images [17]. They can be applied in pre-segmentation tasks for semantic segmentation of natural scenes [18, 19]. Another application would be the detection of communities in complex networks that model transport and propagation of epidemics or other types of networks where the spatial location is a core factor.

The remainder of this paper is organised as follows: in Section 2, we describe some traditional and state-of-the-art related

work; in Section 3, we detail concepts that underpin our proposal; in Section 4, we present three label propagation-based segmentation methods; Section 5 presents the experimental results; and finally, in Section 6 we present conclusions.

2 | RELATED WORK

Jianbo Shi and Malik [4] proposed Normalized Cut (NCut), one of the first graph clustering-based methods to automatic image segmentation. The image is represented as a weighted undirected graph, where each vertex represents a pixel and edge weights are given by the Gaussian similarity between pixels. Aware of the graph partitioning NP-complete nature, the authors proposed a method where the eigenvectors of the Laplacian matrix are used to compute a division of the graph. Finding the eigenvectors of the Laplacian matrix has a computational complexity of $O(n^3)$, where n is the number of vertices. To improve segmentation results and further decrease computational cost, several NCut-based methods followed [20–23]. Fowlkes et al. [20] employ linear algebra techniques, such as Lanczos and Nystrom, to reduce the cost of computing the eigenvectors. Cour et al. [21] proposed a multi-scale segmentation approach and claimed that the technique had almost linear computational cost $\sim O(n)$, n being the number of pixels [21].

Casaca et al. [24] proposed a methodology based on cartoon-texture concepts, image reduction, and scaling strategies with the bi-cubic interpolation to segment images. The authors claimed, in addition to an improvement in segmentation quality, a processing time six times faster than the original proposal NCut [4].

Felzenszwalb and Huttenlocher [12] introduced a method for automatic image segmentation called Efficient Graph-Based Image Segmentation (EGBIS). It creates regions of vertices by joining components of the graph, where the vertices represent location and colour information of the pixels. Edge weights are computed by measuring the distance among vertices. The computational cost is approximately linear $\sim O(m)$, where m is the number of edges. However, the number of clusters is not determined automatically. Compared to NCut, the computational cost is low, as there is no need to compute eigenvectors.

A variation of the EGBIS method was presented by Cheng et al. [25], in which the vertices are represented by super-pixels. The authors highlight three reasons for using super-pixels instead of single pixels: (i) super-pixels are units that contain features such as shape and texture; (ii) the number of super-pixels is drastically inferior than the number of pixels, yielding a more compact representation with reduced computational cost; and (iii) the super-pixel adds more coherence and robustness when compared with raw pixels alone.

Cuadros et al. [5, 6] proposed an automatic segmentation method, which includes a combination of a colour based version of the Speed-up Turbo Pixels (SUTP) method [26] for super-pixel extraction and a method for finding communities (clusters) in large graphs, the Fast Greedy (FG) algorithm [9]. They use super-pixels to represent the vertices in the graph and colour descriptors and texture (Local Binary Patterns [27, 28]) super-pixels features. The authors found that the most accurate descriptor for segmenting natural scene images is the colour average of super-pixels.

The aforementioned method, Fast Greedy, requires a measure called Modularity, which assesses the “quality” of a division of a graph into communities. Ideally, a good division is the one in which there is a high number of edges between vertices of the same community as well as a low density of edges between members of different communities. The FG method applies a greedy strategy to approximate an optimal division of the graph in which the Modularity value is maximised. One of the main advantages of FG is the low computational cost, which is of the order $O(m d(\log n))$, where m is the number of edges, n is the number of vertices, and d is the depth of the dendrogram. For sparse and hierarchical graphs with $m \sim n$ and $d \sim \log n$ the complexity is of linear order $\sim O(n \log^2 n)$. Another important feature to highlight is that FG automatically determines the number of clusters in the graph. The major drawback, however, is its non-deterministic nature especially for image applications.

Cuadros et al. [5] also evaluated the community detection algorithm Label Propagation (LP) [10] with an iterative propagation of vertex labels. The low computational cost of LP, which is of the order $O(m)$ for each iteration, made it possible to segment very large graphs. However, according to the authors, the resulting segmentation were not as accurate as the one achieved with the FG algorithm the main reason being the LP over-segmentation problem. Like Fast Greedy, the LP approach, as implemented, does not provide deterministic segmentation. This is due to the random propagation mechanism of labels at each iteration.

Mourchid et al. [29] proposed to use the community detection algorithm, known as the Louvain method (LV) [11] for automatic image segmentation. The LV method is a greedy agglomerative hierarchical algorithm proposed by Blondel et al. [11] for community detection in large graphs. The LV method is also based on the concept of Modularity, and consists of two iterative phases: (i) Modularity optimisation: initially, each vertex is labelled as a community. Next, vertices are moved to a neighbouring community. The Modularity is evaluated at each step, the vertex remains in the new community if there is a positive gain in the Modularity. (ii) Aggregation: a new network is created where the vertices are the communities created during the

ALGORITHM 1 LP Algorithm

Input : $G(V, E)$: graph.
Output: k components.

```

1  $t = 0$ ; // in  $t=0$ 
2  $C_x(t) \leftarrow \text{Label}(V)$ ; // different labels
3 while True do
4    $t = t + 1$ ; // in each  $t$  iteration
5   foreach  $x \in X$  do
6      $C_x(t) \leftarrow f(\cdot)$ ; // Eq(1) or Eq(2)
7   if isConverged( $C_x$ ) then
8     break
9 return MakeComponents( $C$ )
```

first phase. Both (i) and (ii) are processed until no more positive gain in Modularity is achieved. The computational cost of the LV method is linear $O(n)$ for graphs with $V \approx E$, where V is the number of vertices and E the number of edges in the graph. For dense graphs, it is approximately $O(n \log n)$. However, in image segmentation domain, the method also exhibits over-segmentation.

Nguyen et al. [8] combined the LV method with super-pixels. The authors also proposed a strategy to solve the over-segmentation problem by merging a pair of regions according to some colour similarity metric over the original LV output.

3 | CONCEPTS

In this section, we present the main concepts that underpin this work. The graph clustering method Label Propagation is described, followed by the pre-segmentation super-pixels methods. Metrics for quantitative assessment of the proposed method are also introduced.

3.1 | Label propagation (LP)

LP [10] was originally proposed as a method for detecting communities in complex networks modelled as a graph. It starts by assigning a distinct label to each vertex. Next, k clusters are obtained iteratively by propagating the vertex label towards its neighbouring vertices. Given a graph $G = (V, E)$, the label propagation algorithm (Algorithm 1) performs the following steps:

- (i) Initially at $t = 0$, different labels are assigned to each vertex $C_x(t) = x$, where $x \in V$. Set $t = 1$.
- (ii) At each iteration t , arrange the graph vertices randomly $X = \text{random}(x_1, \dots, x_n), x_i \in V$.
- (iii) For each vertex $x \in X$ update its label according to the highest frequency of its n neighbouring labels. The highest frequency of labels can be determined synchronously (Equation 1) or asynchronously (Equation 2).

$$C_x(t) = f(C_{x_1}(t-1), \dots, C_{x_n}(t-1)), \quad (1)$$

$$C_x(t) = f(C_{x_{i1}}(t), \dots, C_{x_{im}}(t), C_{x_{i(m+1)}}(t-1), \dots, C_{x_{in}}(t-1)), \quad (2)$$

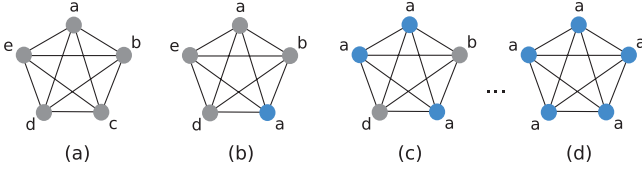


FIGURE 2 Iterative label propagation: (a) Initial graph with distinct labels; (b,c,d) iterative propagation of label *a* [10]

where (t) represents the current iteration, $(t - 1)$ the previous iteration, $f(\cdot)$ the function for determining the highest frequency of labels in the neighbourhood of vertex x .

- (iv) If each vertex x has the same label as most of its neighbours, terminate; otherwise, go to (ii).
- (v) The k communities are obtained by clustering the vertices that share the same labels.

Figure 2 illustrates how iterative propagation works for the *a* label.

3.2 | Super-pixels

A super-pixel is a set of pixels which share a common property and can be used as a pre-segmentation tool. Super-pixel methods are a relatively recent proposal, characterised by their ability to precisely adjust to the boundaries of the processed image. Hence, a good super-pixel method should seek quality (ability to self-adjust to image edges) at a reasonably running time. The major importance of super-pixel methods is associated with the reduction in the computational cost of the underlying image analysis tasks in which they are employed. Formed of a set of pixels, they can convey more information (statistical measures such as mean and variance) and, hence, be used as powerful descriptors when compared with a single pixel [30].

3.2.1 | Simple not iterative clustering (SNIC)

SNIC [31] is presented as an improvement of the SLIC super-pixel [30] both in quality and computational cost. To group pixels into a super-pixels, SNIC employs a priority queue according to the distance between the pixels and the super-pixel centroids. As a result, the method does not require iterations like its counterpart SLIC does. The steps of the SNIC algorithm (Algorithm 2) are described below:

- (i) Let K be the number of super-pixels to extract. Given an image with N pixels, set the pixel labels $L[\cdot] = -1$ to indicate that a pixel belongs to a given super-pixel. A regular grid with initial cells of dimension $S \times S$ is created, where $S = \sqrt{N/K}$.
- (ii) Initialise the super-pixel centroids, $C[\cdot] = \{x, c\}$, where x is the spatial location of the centroid, $c = \{l, a, b\}$ are the colour intensities in the CIELAB colour space for the pixels located at x .

ALGORITHM 2 SNIC Algorithm

Input : I : image.
Output: I' image segmented with super-pixels.

```

1  $L_i = -1, i \in I$ ; // set labels
2  $R \leftarrow \text{makeRegularGrid}(I)$ 
3  $C[\cdot] \leftarrow \text{makeCentroids}(R)$ ; // make centroids
4  $Q.\text{push}(x_i, c_i, k_i, 0), i \in C$ ; // insert centroids
5 while  $Q$  is not empty do
6    $i \leftarrow Q.\text{pop}()$ ; // pop pixel
7   if  $L_i == -1$  then
8      $L_i \leftarrow k_i$ ; // set label
9     foreach  $j \in N(i)$  do
10       $d \leftarrow d(j, k_i)$ ; // compute distance
11       $Q.\text{push}(x_j, c_j, k_i, d)$ ; // insert pixel j
12 return  $I'$ 

```

- (iii) Given a priority queue Q , with respect to distance d , insert the C_i centroids in the form $e = \{x, c, k, d\}$, where $k = i$ is an index to identify the super-pixel, and $d = 0$ which establishes the distance to centroid i . The distance d is initially set to 0 for all centroids.
- (iv) While the priority queue Q is not empty, pop an element (pixel) $e_i = \{x_i, c_i, k_i, d_i\}$ from Q . If $L[x_i] = -1$, then assign a label to $L[x_i] = k_i$. Update the $C[k_i]$ centroid with the just inserted pixel $\{x_i, c_i\}$.
- (v) For each x_j neighbour of x_i create $e_j = \{x_j, c_j, k_i, d_{(j, k_i)}\}$. If $L[x_j] = -1$, then push e_j into Q , with $d_{(j, k_i)}$ given by the following equation:

$$d_{(j, k_i)} = \sqrt{\frac{\|x_j, x_{k_i}\|_2^2}{s} + \frac{\|c_j, c_{k_i}\|_2^2}{m}}, \quad (3)$$

where s and m are normalising factors for space distance and colour distance, respectively.

Being initially a squared region, the super-pixel side s has a fixed value of $\sqrt{N/K}$. The m parameter is a compression factor that is provided by the user. The higher the m , the more compact the super-pixels, at the expenses of less adherence to the edges of the image, and vice versa.

3.3 | Evaluation of image segmentation

An effective way to quantitatively evaluate segmentation is to compare the segmented images S_T of a given method with a reference segmented image S_G or ground-truth. In this work, ground-truth data consists of a set of manual segmented images, subjected to users' own opinions. We describe three measures commonly used to quantitatively assess segmentation: the first is based on the coverage of regions; the second, on the distances among the object contours and the third, a measure based on probabilities.

3.3.1 | Covering

Given S_T as the segmentation result produced by a method and given S_G as the reference segmentation, the coverage of regions

(Covering) [32, 33] quantifies the coverage of the regions of S_T in relation to the regions of S_G . The Covering is defined by Equation (4)

$$Covering(S_T \rightarrow S_G) = \frac{1}{A} \sum_{R \in S_G} |R| \max_{R' \in S_T} \{\mathcal{O}(R, R')\}, \quad (4)$$

$$\mathcal{O}(R, R') = \frac{|R \cap R'|}{|R \cup R'|} = \frac{|R \cap R'|}{|R| + |R'| - |R \cap R'|}, \quad (5)$$

where A is the number of pixels in the image. R and R' are regions of S_G and S_T , respectively. $\mathcal{O}(R, R')$, also known as the Jaccard Index [34, 35], represents the overlap between the R and R' regions. $|R|$ and $|R'|$ denote the number of pixels in the R and R' regions, respectively. Equation (4) returns values in the range $0 \leq Covering(S_T \rightarrow S_G) \leq 1$. Values close to 1 indicate high similarity between S_T and S_G regions, which means good segmentation quality.

3.3.2 | Boundary-based measures

The quantitative assessment for boundary-based segmentation was initially proposed by Qian Huang and Dom [36]. Later, Martin et al. [37] extended this idea to estimate the segmentation accuracy by calculating minimum distances between the pairs of points of 2 sets of boundaries: (i) BT boundaries of some segmentation method S_T and (ii) BG boundaries of the ground truth reference segmentation S_G .

To reduce the computational cost, Estrada and Jepson [38] and Dogra et al. [39] proposed a convolutional process with circular windows to traverse the boundaries of BT and BG and, thus, compute distances according to selected metrics: Precision (P), Recall (R) and BF1-Score.

Precision (Equation 6) quantifies the similarity of the points of the boundaries BT in relation to the points of the boundaries BG, according to a maximum distance θ . Similar to Precision, the measure Recall (Equation 7) quantifies the similarity between the points of the boundaries BG in relation to the boundaries BT. The metric BF1-Score is also defined from P and R (Equation 8) [40].

$$P = \frac{1}{|BT|} \sum_{p \in BT} [Matched(p, BG) \leq \theta], \quad (6)$$

$$R = \frac{1}{|BG|} \sum_{p \in BG} [Matched(p, BT) \leq \theta], \quad (7)$$

$$BF1 - Score = 2 \times \frac{P \times R}{R + P}. \quad (8)$$

In P , $Matched(\cdot)$ traverses the p points of BT in search of points near to the BG boundaries, following a maximum distance θ . If $[\cdot]$ is true, the return value is 1, otherwise 0. In R , $Matched(\cdot)$ traverses the p points of BG searching for points near the BT boundaries. $|BT|$ and $|BG|$ are the number of points in boundaries BT and BG, respectively.

3.3.3 | Probabilistic rand index (PRI)

The PRI metric [41, 42] computes the probability that a pair of pixels (i, j) of the segmentation obtained by a S_T method have consistent labels in the set of k reference S_{G_k} segmentation (ground-truth). Thus, the PRI metric is defined in Equation (9)

$$PRI(S_T, \{S_{G_k}\}) = \frac{1}{A} \sum_{i < j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})], \quad (9)$$

where c_{ij} is the probability that pixels i and j have the same label in segmentation S_T , p_{ij} corresponds to the probability that pixels i and j share the same label in the set of reference segments S_{G_k} , and A is the total number of pixel pairs. The PRI presents values in the range of $[0, 1]$; high values indicate high similarity between the segments S_T and S_{G_k} .

4 | THE PROPOSED METHOD

The major steps of the proposed method are depicted in Figure 1. We start by computing super-pixels from a coloured natural scene image. Then, a similarity graph is created, where the vertices represent the super-pixels and edges the similarity weights between two vertices. Segmentation is finally achieved by means of label propagation based-algorithms. Each of these steps is detailed in the following subsections.

4.1 | Super-pixels extraction

Let N be the total number of pixels in the image and S the size of an initial square-shaped super-pixel. Thus, one can extract $R = N/S^2$ super-pixels from the original image. Since every super-pixel represents a graph node, the higher the R , the greater the number of vertices of the graph, leading to more accurate segmentation. Clearly, the higher the S , the smaller the amount of vertices and processing time. However, this will impact the quality of the segmentation. Determining the optimal value of S will be crucial.

We employ the SNIC super-pixel method, which provides good quality at low computational cost [31]. The output of this process is an over-segmented image with R regions or super-pixels.

4.2 | Graph building

An undirected and weighted graph $G = (V, E, W)$ is created with V vertices, E edges and W weights. With every graph node modelled as a super-pixel, $V = R$. The tasks that lead to graph G are described below:

- (i) Vertex creation: V vertices can be best represented by feature vectors, each describing the content of one super-pixel. As we are dealing with colour images, we created two colour

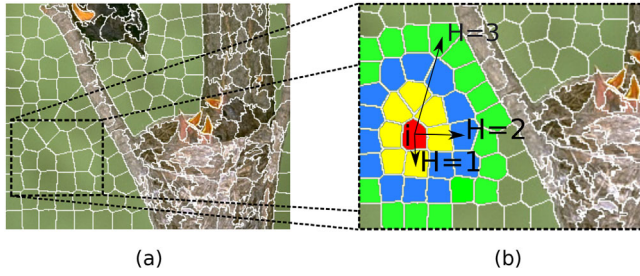


FIGURE 3 Multilevel neighbourhood strategy. (a) A subset of super-pixels; (b) edge creation for vertex i (red super-pixel) and its j neighbours for the first three levels ($H = 3$) depicted in yellow, blue and green colours

descriptors in the CIELAB colour space, namely: (i) ME, the mean pixel intensity of each colour channel (three features) and (ii) CM, the first three colour statistical moments (mean, variance and skewness) per channel, yielding nine features in total. Features are normalised with zero mean and unit variance.

- (ii) Edge creation: the definition of an edge weight follows a multilevel neighbourhood criterion H , as illustrated in Figure 3. Assume i is a super-pixel and, for a given H value, there exists a set of j neighbouring super-pixels. Recall that i and j are actually feature vectors, hence the edge weights W_{ij} can be computed by a weight function $F(\cdot)$, defined by Equation (10)

$$W_{ij} = \begin{cases} F(i, j), & \text{If } j \in H \text{ and } F(i, j) \geq T \\ \emptyset, & \text{else} \end{cases}, \quad (10)$$

where H delimits the neighbourhood reach among super-pixel i and neighbours j , T is a threshold that governs the creation of the edge weight W_{ij} . Edge weights can be interpreted as follows: W_{ij} with values close to 1 indicate similar super-pixels, whereas values close to 0 represent low similarity. In this paper, we propose three similarity measures as the weight function $F(i, j)$: Gaussian (GAU), Cosine (COS) and Tanimoto (TAN) defined, respectively, by Equations (11)–(13)

$$GAU(i, j) = e^{\frac{-d(i, j)^2}{2\sigma^2}}, \quad (11)$$

$$COS(i, j) = \begin{cases} c = \frac{i \cdot j}{\|i\| \|j\|}, & \text{If } c > 0 \\ 0, & \text{else} \end{cases}, \quad (12)$$

$$TAN(i, j) = \begin{cases} t = \frac{i \cdot j}{\|i\|^2 + \|j\|^2 - i \cdot j}, & \text{If } t > 0 \\ 0, & \text{else} \end{cases}, \quad (13)$$

where σ is a constant of value 0.5. As proposed, COS and TAN functions are guaranteed to produce values in the range $[0, 1]$. Algorithm (3) details the graph creation process.

ALGORITHM 3 Graph Building

Input : I : image; R super-pixels; T : threshold; H : level of neighbourhood; F : proximity function.

Output: $G = (V, E, W)$: weighted undirected graph.

```

1 Function Graph( $I, T, R, H, F$ )
2    $V \leftarrow \text{Description}(I, R)$ 
3    $E \leftarrow []$ ;  $W \leftarrow []$ ;  $C[\cdot] = \emptyset$ 
4   foreach  $i \in R$  do
5      $C[i] \leftarrow i$ ; // vertex  $i$  visited by  $i$ 
6      $h \leftarrow 0$ 
7      $A \leftarrow \text{Queue}(\text{FN}[i])$ ; // first neighbours of  $i$ 
8     do
9       foreach  $j \in A$  do
10         $C[j] \leftarrow i$ ; // vertex  $j$  visited by  $i$ 
11         $d \leftarrow F(V_i, V_j)$ 
12        if  $d \geq T$  then
13           $E_{ij} \leftarrow (i, j)$ ; // make edge
14           $W_{ij} \leftarrow d$ ; // save weight
15           $B \leftarrow \text{Queue}()$ 
16          foreach  $j \in A$  do
17            foreach  $t \in \text{FN}[j]$  do
18              // if  $t$  was not visited by  $i$ 
19              if  $C[t] \neq i$  then
20                 $B.\text{push}(t)$ 
21             $A \leftarrow B$ 
22             $h \leftarrow h + 1$ 
23   while  $h < H$  and  $A.\text{size}() > 0$ ;

```

4.3 | Label propagation segmentation

In this section, we describe three label propagation strategies for automatic image segmentation. The novelty of our proposal is the use of image domain information, as opposed to the label propagation method [10], originally proposed for the identification of communities in the context of complex networks.

The propagation of labels follows the spatial location of the super-pixels in the image: from top to bottom, from left to right. Vector $X = \text{SuperpixelsLocation}(v_1, \dots, v_n)$ contains the spatial location of all super-pixels extracted from the image. We describe the three variations of our label propagation method in next sections.

4.3.1 | Iterative label propagation (ILP)

ILP is a deterministic strategy for automatic segmentation of images, which propagates labels **iteratively** given an arbitrary neighbourhood range. The steps are described below:

- (i) Assign different labels to each vertex $L_i^{(t)} = i, i \in V$.
- (ii) For each iteration (t), asynchronously update the label of each vertex $i \in X$ looking at the frequency of its neighbouring vertex. This is given by Equation (14)

$$f^{(t)}(i, L) = \underset{j \in N(i)}{\operatorname{argmax}} \left(\sum L_j^{(t)}, \dots, \sum L_j^{(t-1)} \right), \quad (14)$$

ALGORITHM 4 Iterative Label Propagation

Input : I : image; T : threshold; S : super-pixel side length;
 H : level of neighbourhood; F : proximity function.

Output: k segments.

```

1  $R \leftarrow \text{Superpixels}(I, S)$ ; // SNIC
2  $\{G = (V, E, W)\} \leftarrow \text{Graph}(I, R, T, H, F)$ ; // Alg (3)
3  $X \leftarrow \text{SuperpixelsLocation}(v_1, \dots, v_n)$ 
4  $L_i^{(t=0)} \leftarrow i, i \in X$ ; // different labels
5  $t \leftarrow 1; c \leftarrow 1$ 
6 while  $c > 0$  do
7    $c \leftarrow 0$ 
8   foreach  $i \in X$  do
9      $\alpha \leftarrow L_i^{(t-1)}$ 
10     $L_i^{(t)} = f^{(t)}(i, L)$ ; // Eq (14)
11    if  $\alpha \neq L_i^{(t)}$  then
12       $c = c + 1$ 
13     $t = t + 1$ 
14 return  $\text{MakeSegmentes}(L)$ 

```

where argmax returns the most frequent label found in the neighbourhood of i . $N(i)$ gives all i neighbours.

- (iii) After each iteration, the number of label updates c is evaluated. If $c > 0$, continue to the next iteration $t = t + 1$ and go to (ii). Otherwise, finish. Algorithm (4) details the ILP method.

4.3.2 | Recursive label propagation (RLP)

RLP is a deterministic strategy for automatic segmentation which propagates labels recursively. The steps are described below:

- (i) Assign different labels to the vertices $L_i = i, i \in V$.
- (ii) For each vertex $i \in X$, recursively update the labels of the neighbouring vertices according to Equation (15)

$$f(i, L) = \begin{cases} \alpha = L_i \\ \beta = \text{argmax}_{j \in N(i)} (\sum L_j) \\ \text{If } \alpha \neq \beta \\ L_i = \beta \\ \delta_{j \in N(i)} (f(j, L) \text{ If } L_j \neq \beta) \end{cases} \quad (15)$$

where α is the current label of vertex i , β the new label assigned by argmax . When a change of labels occurs, $\alpha \neq \beta$, and label $L_i = \beta$ is finally updated. Labels of the j neighbours of i are recursively updated following $\delta_{j \in N(i)}(\cdot)$. Algorithm (5) details the RLP method.

4.3.3 | Weighted recursive label propagation (WRLP)

WRLP is similar to RLP method, but considers the edge weights in the propagation of labels. The steps are described below:

ALGORITHM 5 Recursive Label Propagation

Input : I : image; T : threshold; S : super-pixel side length;
 H : level of neighbourhood; F : proximity function.

Output: k segments.

```

1  $R \leftarrow \text{Superpixels}(I, S)$ ; // SNIC
2  $\{G = (V, E, W)\} \leftarrow \text{Graph}(I, R, T, H, F)$ ; // Alg (3)
3  $X \leftarrow \text{SuperpixelsLocation}(v_1, \dots, v_n)$ 
4  $L_i \leftarrow i, i \in X$ ; // different labels
5 foreach  $i \in X$  do
6    $f(i, L)$ ; // Eq (15)
7 return  $S \leftarrow \text{MakeSegmentes}(L)$ 

```

ALGORITHM 6 Weighted Recursive Label Propagation

Input : I : image; T : threshold; S : super-pixel side length;
 H : level of neighbourhood; F : proximity function.

Output: k segments.

```

1  $R \leftarrow \text{Superpixels}(I, S)$ ; // SNIC
2  $\{G = (V, E, W)\} \leftarrow \text{Graph}(I, R, T, H, F)$ ; // Alg (3)
3  $X \leftarrow \text{SuperpixelsLocation}(v_1, \dots, v_n)$ 
4  $L_i = i \in X$ ; // different labels
5 foreach  $i \in X$  do
6    $f(i, L, W)$ ; // Eq (16)
7 return  $\text{MakeSegmentes}(L)$ 

```

- (i) Assign different labels to the vertices $L_i = i, i \in V$.
- (ii) For each vertex $i \in X$, recursively update the labels of the neighbouring vertices, considering the edge weights. This is calculated by Equation (16):

$$f(i, L, W) = \begin{cases} \alpha = L_i \\ \beta = \text{argmax}_{j \in N(i)} \left(\sum L_j (1 + W_{ij}) \right) \\ \text{If } \alpha \neq \beta \\ L_i = \beta \\ \delta_{j \in N(i)} (f(j, L, W) \text{ If } L_j \neq \beta) \end{cases}, \quad (16)$$

with argmax returning the label with the highest frequency in the neighbourhood of vertex i . Unlike the previous strategies, argmax includes edge weights W_{ij} in the propagation process. Algorithm (6) details the WRLP method.

For the three proposed methods, the final segmentation is obtained by clustering k components of the graph, where each component is formed by vertices that share the same label. The components represent image regions.

4.4 | Computational complexity

The iterative propagation method (ILP) does not require many iterations to converge, therefore, it can be inferred that the computational complexity of the method is of linear order $O(m)$ for dense graphs, and of a linear order $O(n)$ for sparse graphs with $V \sim E$, where n and m are the number of vertices and edges of the graph, respectively.

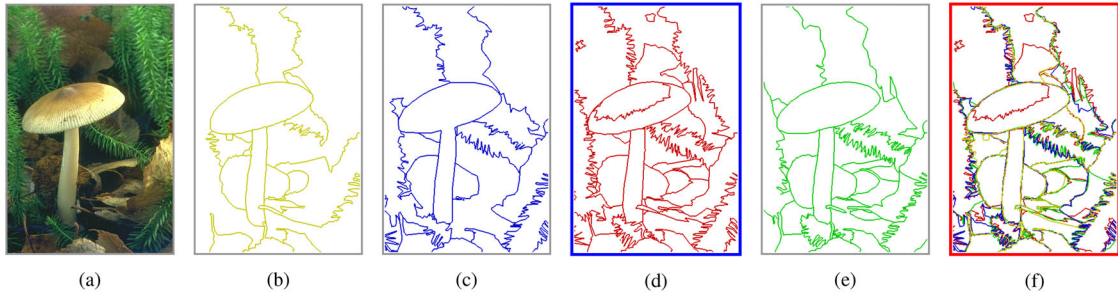


FIGURE 4 Selection of manual reference segmentation. (a) Original image; (b–e) boundaries of four manual segmented images; (f) combined boundaries of the four manual segmentation. Selected manual reference image shown in blue frame (d)

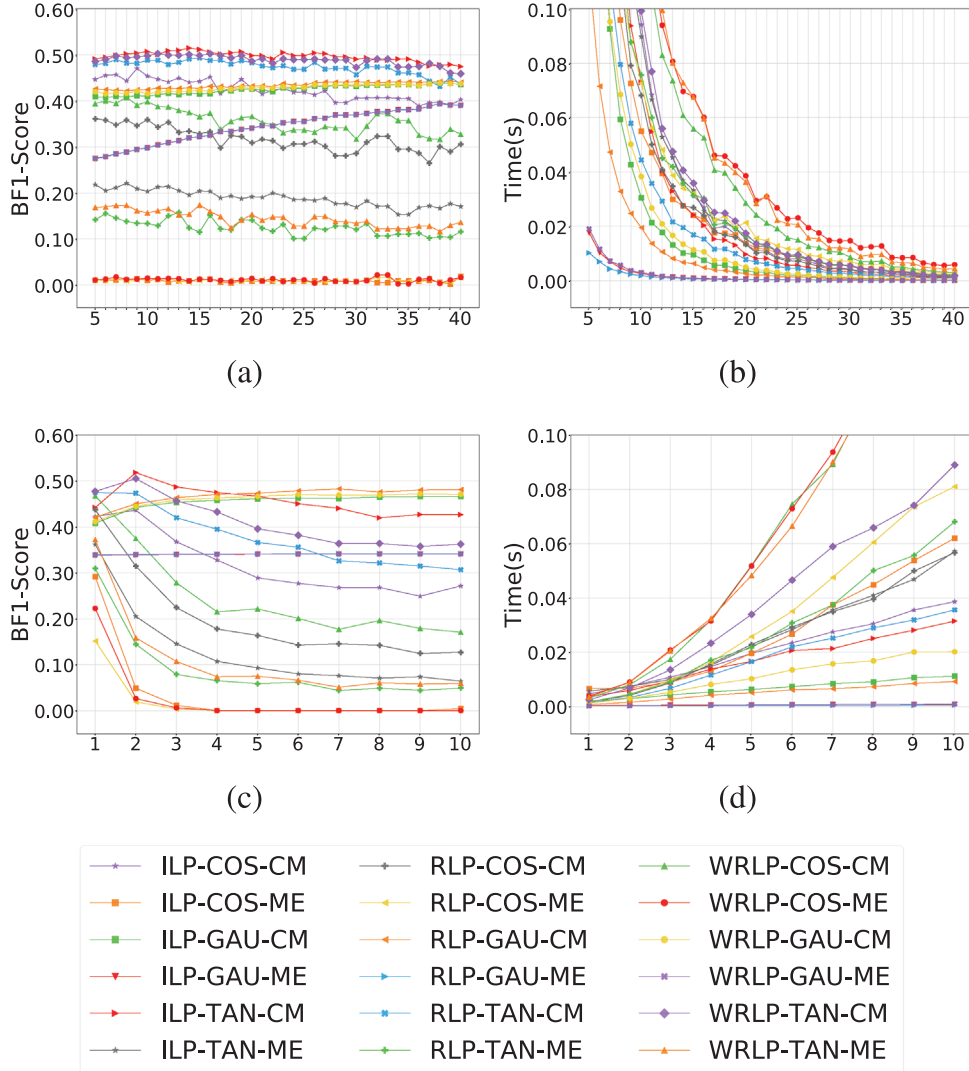


FIGURE 5 Quantitative results for S and H parameter values. (a,b) Segmentation results and processing time for S; (c,d) segmentation results and processing time for H

Regarding the recursive (RLP) and weighted recursive (WRLP) propagation methods, for low neighbourhood levels, the computational complexity is $O(m)$ since it has only one internal recursion call in the neighbourhood of each vertex. In addition, the label propagation converges very quickly.

5 | RESULTS AND DISCUSSIONS

We used the Berkeley University dataset BSDS500 [42] to evaluate the three proposed segmentation methods. It contains 500 natural scenes images and a number of human-made segmented

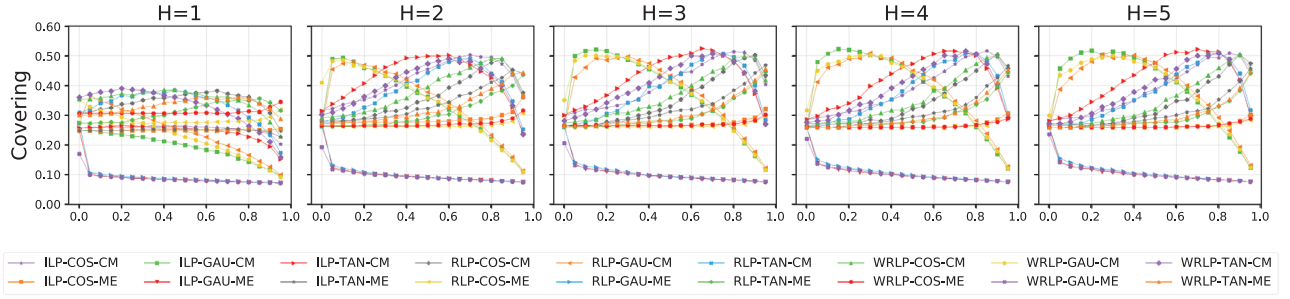


FIGURE 6 Covering results for 20 distinct threshold values (X axis) for the 100 images in the BSDS500-Validation set. From left to right, neighbourhood level $H = [1, 2, 3, 4, 5]$

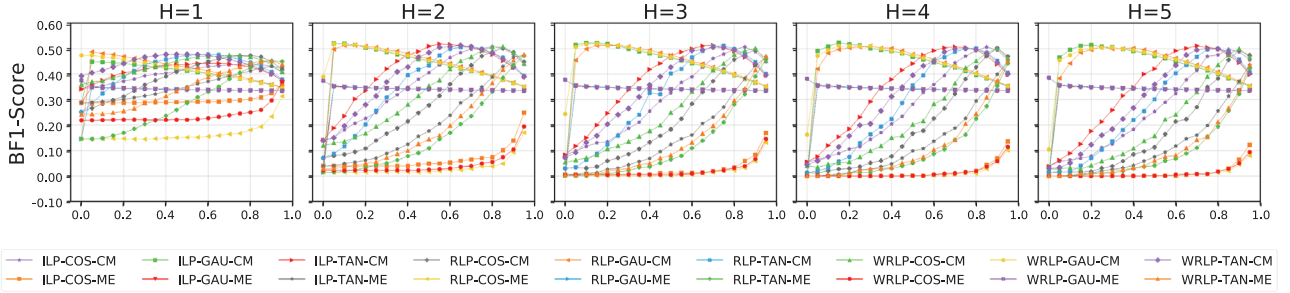


FIGURE 7 BF1-Score results for 20 distinct threshold values (X axis) for the 100 images in the BSDS500-Validation set. From left to right, neighbourhood level $H = [1, 2, 3, 4, 5]$

samples, for each image (5 – 10). The dataset consists of images for training images (200 samples), testing (200), and validation (100). Each image is 481×321 pixels, yielding graphs with a maximum of 154,401 vertices. It is worth mentioning that the manual segmented samples provided vary widely within a single image, as they are subject to the perception of each person.

Bearing in mind the high non-uniformity of the manual segmented samples for an arbitrary image in the database, we devised a strategy to select a single reference sample, so as to perform quantitative evaluation of the proposed methods. This process is illustrated in Figure 4. We first add all contours of each manual segmentation (Figure 4(b–f)) to produce a fused contour image (Figure 4(f)). We then apply Equation (8) to select the reference ground-truth contour for evaluation purposes. In this scenario, the manual segmented samples correspond to the BT boundaries and the fused boundary image, the BG term as explained in Section 3.3.

5.1 | Assessment of parameters

The proposed methods ILP, RLP, and WRLP rely on a few parameters that affect the segmentation outcome. They are: neighbourhood level (H), threshold (T) and initial length of the super-pixel side (S). The first experiment aims to show the performance of the proposed methods, as we change the aforementioned parameter values. We employ the similarity functions GAU, TAN, and COS to quantitatively evaluate the performance so as to define the “optimal” parameter setup. We employed the descriptors ME and CM, as proposed in Section 4.

We start by analysing the super-pixel size S parameter, for fixed values of T and H . Figures 5(a) and 5(b) illustrate the results for $S = [5, 6, \dots, 39, 40]$, $T = 0.7$ and $H = 3$. The best result was obtained with ILP-TAN-CM setup. We also observe that the longer the super-pixel side, the lower the segmentation quality and shorter running times. This is expected as the

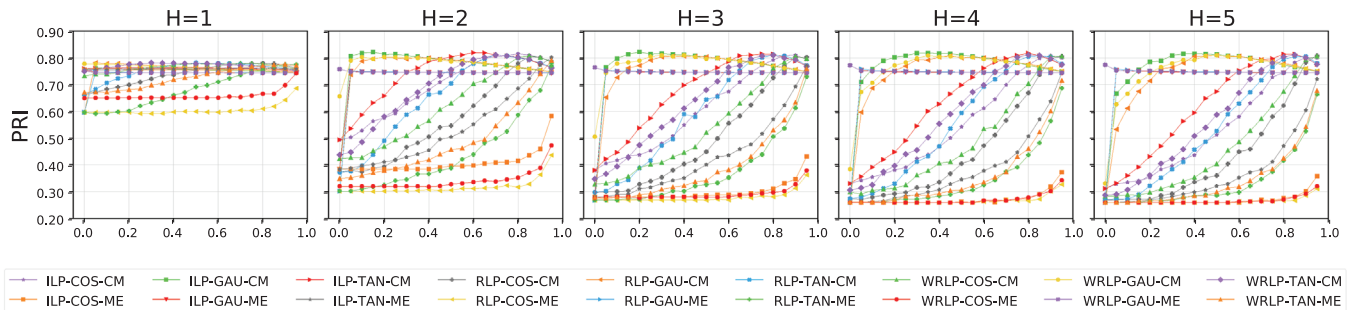


FIGURE 8 PRI results for 20 distinct threshold values (X axis) for the 100 images in the BSDS500-Validation set. From left to right, neighbourhood level $H = [1, 2, 3, 4, 5]$

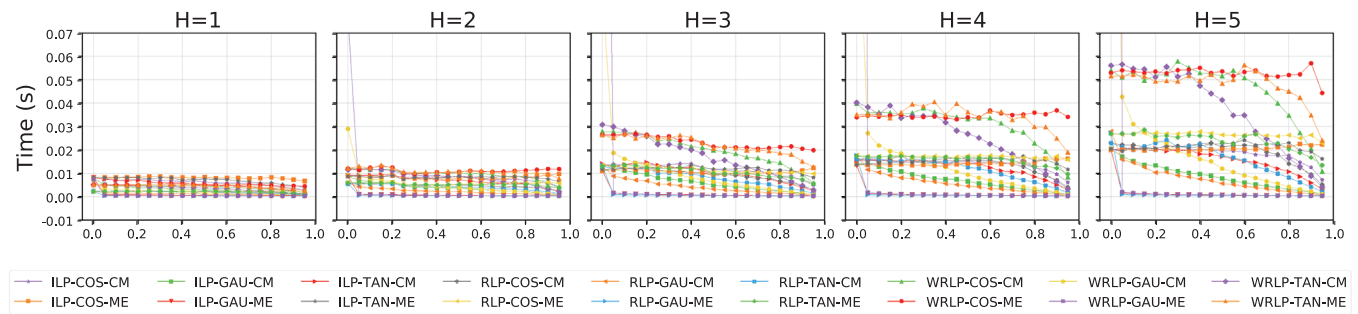


FIGURE 9 Time results for 20 distinct threshold values (X axis) for the 100 images in the BSDS500-Validation set. From left to right, neighbourhood level $H = [1, 2, 3, 4, 5]$

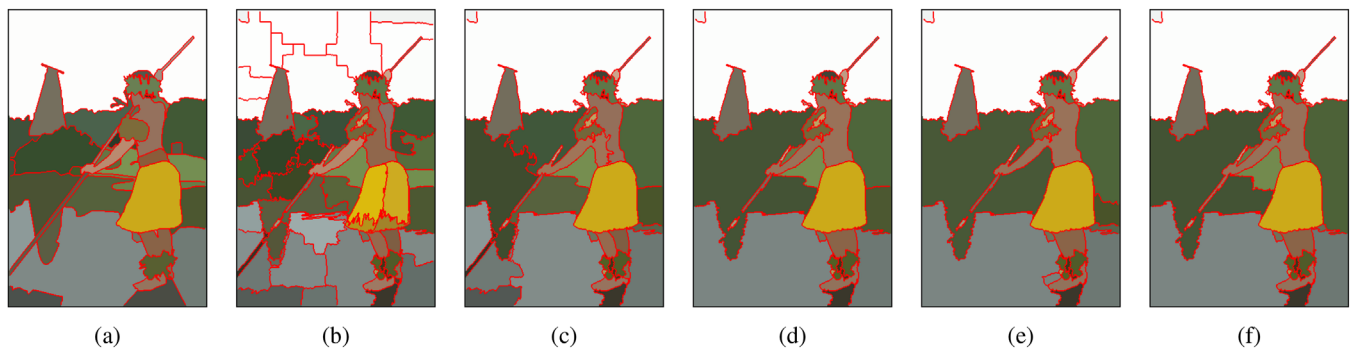


FIGURE 10 Segmentation results for distinct values of T and neighbourhood level H . (a) Ground truth; (b–f) segmentation obtained with the ILP-TAN-CM method with $H = [1, 2, 3, 4, 5]$ and fixed $T = 0.75$

underlying graph has a reduced number of vertices. The best segmentation results were obtained for super-pixel side S in the range $\{14–27\}$.

Based on this outcome, we devise a strategy to automatically determine the best value of S . We first compute the number of squared super-pixels as $N_{sp} = \sqrt{N_p}$ where N_p is the number

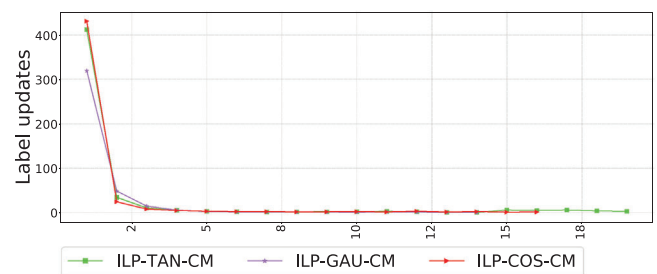


FIGURE 12 Number of iterations (X axis) versus the average of label updates (Y axis)

TABLE 1 Best nine settings for ILP, RLP and WRLP automatic segmentation methods with specific values of S , H and T

Method	S	H	T
ILP-GAU-CM	19	[2,3]	[0.05–0.25]
ILP-TAN-CM	19	[2,3]	[0.70–0.90]
ILP-COS-CM	19	[2,3]	[0.80–0.90]
RLP-GAU-CM	19	[2,3]	[0.10–0.35]
RLP-TAN-CM	19	[2,3]	[0.65–0.85]
RLP-COS-CM	19	[2,3]	[0.75–0.90]
WRLP-GAU-CM	19	[2,3]	[0.10–0.30]
WRLP-TAN-CM	19	[2,3]	[0.65–0.85]
WRLP-COS-CM	19	[2,3]	[0.80–0.90]

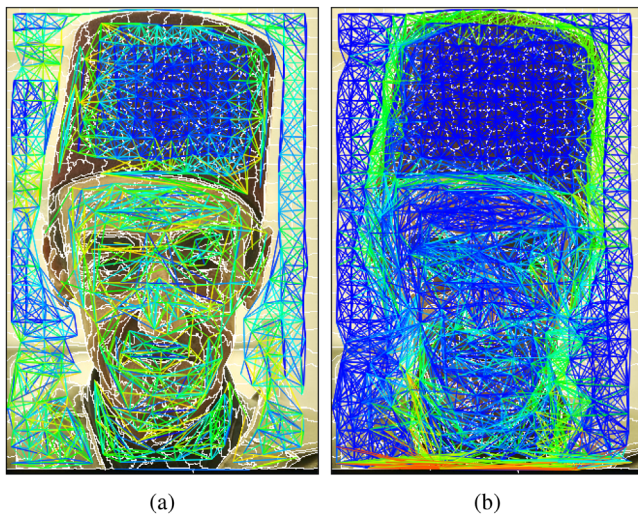


FIGURE 11 Two examples of graph construction. (a) $H = 2$, GAU, $T = 0.25$ and CM descriptor. (b) $H = 2$, GAU, $T = 0.01$ and ME descriptor

TABLE 2 Comparison with other methods. In each column, the two best results are highlighted. Some methods do not use the multilevel strategy to create edges [†]

Method	H	T	Covering	BF1-score	PRI	Time
			Mean (std)	Mean (std)	Mean (std)	Mean (s)
EGBIS-P	1 [†]	—	0.4844 (± 0.1598)	0.4717 (± 0.1428)	0.8143 (± 0.1181)	0.2580
EGBIS-SP	1 [†]	—	0.4734 (± 0.1164)	0.4728 (± 0.1248)	0.7995 (± 0.1293)	0.0004
LPCI-SP	3	0.25	0.3970 (± 0.1122)	0.4684 (± 0.1288)	0.7909 (± 0.1523)	0.0110
LV-SP	3	0.25	0.4706 (± 0.1081)	0.5061 (± 0.1356)	0.7980 (± 0.1398)	0.0715
SUTP-FG	5 [†]	6.00	0.4918 (± 0.1724)	0.4881 (± 0.1600)	0.7513 (± 0.1232)	6.8881
RLP-GAU-CM	2	0.15	0.5129 (± 0.1558)	0.5092 (± 0.1371)	0.8092 (± 0.1173)	0.0053
RLP-TAN-CM	2	0.70	0.5234 (± 0.1448)	0.5023 (± 0.1405)	0.8200 (± 0.1257)	0.0050
WRLP-GAU-CM	2	0.10	0.5291 (± 0.1545)	0.5195 (± 0.1371)	0.8218 (± 0.1241)	0.0120
WRLP-TAN-CM	2	0.70	0.5214 (± 0.1542)	0.5042 (± 0.1405)	0.8231 (± 0.1339)	0.0103
ILP-GAU-CM	2	0.10	0.5152 (± 0.1441)	0.5091 (± 0.1353)	0.8200 (± 0.1242)	0.0094
ILP-TAN-CM	2	0.60	0.5297 (± 0.1303)	0.5069 (± 0.1410)	0.8183 (± 0.1307)	0.0106
ILP*-GAU-CM	2	0.10	0.5063 (± 0.1442)	0.5051 (± 0.1322)	0.8188 (± 0.1243)	0.0049
ILP*-TAN-CM	2	0.60	0.5218 (± 0.1294)	0.5018 (± 0.1382)	0.8171 (± 0.1310)	0.0048
RLP-GAU-CM	3	0.25	0.5288 (± 0.1726)	0.5069 (± 0.1380)	0.8064 (± 0.1287)	0.0090
RLP-TAN-CM	3	0.80	0.5263 (± 0.1463)	0.4932 (± 0.1382)	0.8210 (± 0.1354)	0.0069
WRLP-GAU-CM	3	0.25	0.5233 (± 0.1658)	0.5025 (± 0.1429)	0.8120 (± 0.1310)	0.0154
WRLP-TAN-CM	3	0.80	0.5152 (± 0.1446)	0.4837 (± 0.1366)	0.8214 (± 0.1353)	0.0137
ILP-GAU-CM	3	0.20	0.5491 (± 0.1526)	0.5128 (± 0.1421)	0.8307 (± 0.1115)	0.0127
ILP-TAN-CM	3	0.75	0.5319 (± 0.1430)	0.5025 (± 0.1373)	0.8257 (± 0.1309)	0.0123
ILP*-GAU-CM	3	0.20	0.5411 (± 0.1514)	0.5069 (± 0.1395)	0.8307 (± 0.1100)	0.0069
ILP*-TAN-CM	3	0.75	0.5246 (± 0.1400)	0.4954 (± 0.1382)	0.8230 (± 0.1315)	0.0062



FIGURE 13 Qualitative results. From left to right: original image, ground truth, EGBIS-P, ILP-GAU-CM (our) and ILP-TAN-CM (our)

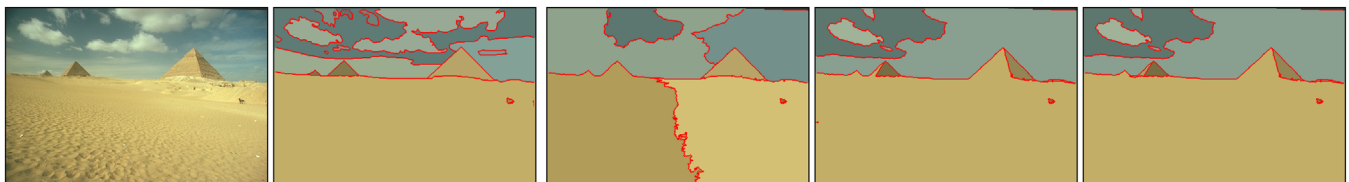


FIGURE 14 Qualitative results. From left to right: original image, ground truth, LV-SP, ILP-GAU-CM (our) and ILP-TAN-CM (our)

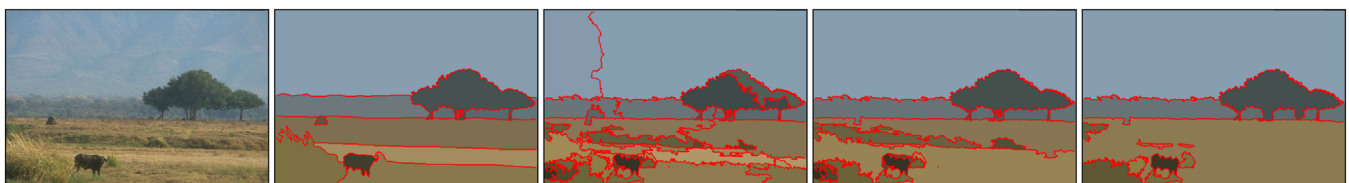


FIGURE 15 Qualitative results. From left to right: original image, ground truth, LPCI-SP, ILP-GAU-CM (our) and ILP-TAN-CM (our)

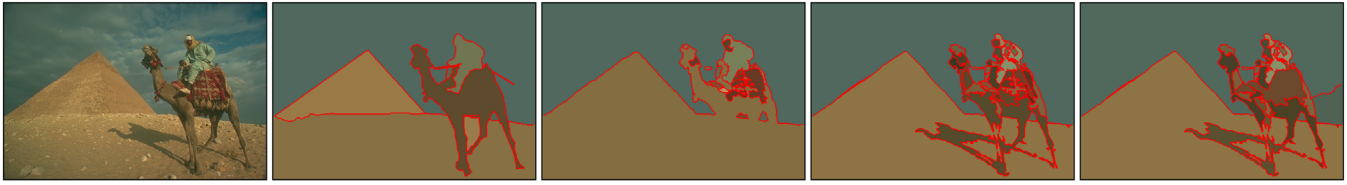


FIGURE 16 Qualitative results. From left to right: original image, ground truth, FG-SUTP, ILP-GAU-CM (**our**) and ILP-TAN-CM (**our**)

TABLE 3 Comparison with other methods. In each column the two best results are highlighted. Some methods do not use the multilevel strategy to create edges [†]

Method	H	T	Covering	BF1-score	PRI	Time
			Mean (std)	Mean (std)	Mean (std)	Mean (s)
EGBIS-P	1 [†]	—	0.7515 (± 0.1710)	0.0888 (± 0.0694)	0.7688 (± 0.1462)	2.8006
EGBIS-SP	1 [†]	—	0.6331 (± 0.1655)	0.0825 (± 0.0460)	0.6742 (± 0.1533)	0.0013
LPCI-SP	3	0.25	0.2491 (± 0.1599)	0.0907 (± 0.0569)	0.4456 (± 0.1676)	0.1092
LV-SP	3	0.25	0.4205 (± 0.0878)	0.1464 (± 0.1047)	0.5245 (± 0.1207)	0.2843
SUTP-FG	5 [†]	6.00	0.6782 (± 0.1877)	0.1029 (± 0.1612)	0.6902 (± 0.1697)	13.0435
RLP-GAU-CM	2	0.15	0.7365 (± 0.1708)	0.0823 (± 0.1286)	0.7551 (± 0.1795)	0.0196
RLP-TAN-CM	2	0.70	0.7591 (± 0.1999)	0.1733 (± 0.1301)	0.7561 (± 0.2021)	0.0208
WRLP-GAU-CM	2	0.10	0.7149 (± 0.1705)	0.0797 (± 0.1206)	0.7379 (± 0.1719)	0.0407
WRLP-TAN-CM	2	0.70	0.7092 (± 0.2047)	0.1915 (± 0.1213)	0.7166 (± 0.2021)	0.0406
ILP-GAU-CM	2	0.10	0.6910 (± 0.1682)	0.0942 (± 0.1052)	0.7332 (± 0.1639)	0.0534
ILP-TAN-CM	2	0.60	0.6894 (± 0.1956)	0.1838 (± 0.1177)	0.7070 (± 0.1979)	0.0635
ILP*-GAU-CM	2	0.10	0.6834 (± 0.1663)	0.0918 (± 0.1038)	0.7256 (± 0.1650)	0.0197
ILP*-TAN-CM	2	0.60	0.6770 (± 0.1926)	0.1838 (± 0.1131)	0.6945 (± 0.1959)	0.0196
RLP-GAU-CM	3	0.25	0.7721 (± 0.1579)	0.0848 (± 0.1263)	0.7951 (± 0.1645)	0.0339
RLP-TAN-CM	3	0.80	0.7902 (± 0.1794)	0.1872 (± 0.1201)	0.7765 (± 0.1953)	0.0337
WRLP-GAU-CM	3	0.25	0.7877 (± 0.1497)	0.0907 (± 0.1217)	0.8129 (± 0.1544)	0.0720
WRLP-TAN-CM	3	0.80	0.7435 (± 0.2006)	0.1946 (± 0.1206)	0.7359 (± 0.2088)	0.0681
ILP-GAU-CM	3	0.20	0.7873 (± 0.1457)	0.0916 (± 0.1104)	0.8270 (± 0.1349)	0.0761
ILP-TAN-CM	3	0.75	0.7184 (± 0.2037)	0.1927 (± 0.1204)	0.7264 (± 0.2107)	0.0826
ILP*-GAU-CM	3	0.20	0.7833 (± 0.1441)	0.0966 (± 0.1098)	0.8224 (± 0.1372)	0.0345
ILP*-TAN-CM	3	0.75	0.7005 (± 0.2081)	0.1855 (± 0.1160)	0.7107 (± 0.2134)	0.0314

of image pixels. The estimated super-pixel size is then given by $S^2 = \lfloor N_{SP} \rfloor$.

All BSDS500 images, for example, have 154,401 pixels and, hence, a total of $N_{SP} = \sqrt{154,401}$ super-pixels, each with side $S = 19$. We can verify that this value is within the range $\{14-27\}$.

We then analyse the influence of the neighbourhood parameter H for fixed values of S and T . Figures 5(c) and 5(d) illustrate the results for $H = [1, 2, \dots, 9, 10]$, $T = 0.7$ and $S = 19$. The best result was also obtained with ILP-TAN-CM setup. We can see that the higher the neighbourhood level, the lower the segmentation quality and the longer the running time. This is expected, as larger neighbourhoods demand graphs with more vertices. The best segmentation results were obtained for neighbourhood values $H = [1, 2, 3]$.

A more comprehensive experiment was performed to evaluate the proposed methods while varying the threshold parameter T and the neighbourhood level H , simultaneously. We defined the threshold in the range $(0.0 \leq T \leq 0.95)$, with a 0.05 increment, yielding 20 values in total. The values for the neighbourhood level were $H = [1, 2, 3, 4, 5]$. The side of the super-pixel side was set to $S = 19$.

In this experiment, we evaluate the three proposed methods (IPL, RLP and WRLP) using 2 descriptors (ME and CM) and 3 similarity functions (GAU, TAN and COS), hence a total of 18 combinations of methods. Each combination then processes 100 samples from the BSDS500-validation dataset. This procedure is carried out for 20 distinct values of threshold T and 5 values of neighbourhood parameter H . In total, 360,000 segmentation operations are processed to generate



FIGURE 17 Qualitative results. From left to right: ground truth, LPCI-SP, LV-SP, FG-SUTP and ILP-GAU-CM (our)

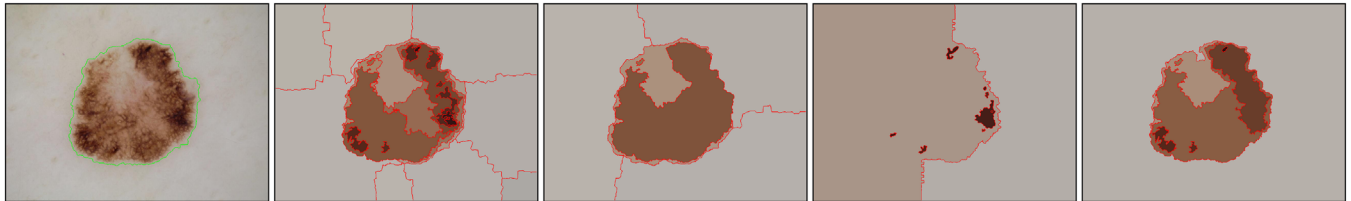


FIGURE 18 Qualitative results. From left to right: ground truth, LPCI-SP, LV-SP, FG-SUTP and ILP-GAU-CM (our)

our statistical analysis. The quantitative results of segmentation (coverage, metric BF1-Score and PRI) and time performance are shown in Figures 6–9, respectively.

Figures 6–9 show that the best results were obtained for values of $H = \{2, 3, 4, 5\}$. In terms of segmentation quality, differences among those values are not significant. However, as H increases, so does the computational cost. For this reason, we adopted values $H = \{2, 3\}$ in our further analysis.

As for the descriptors, CM produced the best results. Thus, only nine combinations were selected. Finally, with $H = \{2, 3\}$, for each of the nine combinations, we identified the threshold ranges for the best segmentation results.

The nine combinations (method + metric + descriptors) selected along with their parameter range are summarised in Table 1. All nine setups exhibited very similar running times. While analysing the wide variation in threshold parameter T ,

we observed that: values between $T=0$ and 0.5 are adequate when the GAU similarity measure is used; values between $T = 0.5$ and 1.0 are suitable for COS and TAN similarity measures. We also observe that GAU consistently gives better results than TAN which, in turn, produces better results than COS. Figure 10 illustrates the best segmentation results for this experiment. For qualitative analysis, Figure 11 shows the graphs created for descriptors CM and ME.

We also provide an analysis of the number of iterations and the label updates that occur in the iterative proposed method ILP. This analysis was carried out with configurations ILP-COS-CM, ILP-TAN-CM and ILP-GAU-CM, for 20 distinct values of the threshold parameter T and neighbourhood $H = 3$. The quantitative results and running times are given in Figures 6–9. The behaviour of the label updates against the number of iterations is illustrated in Figure 12.

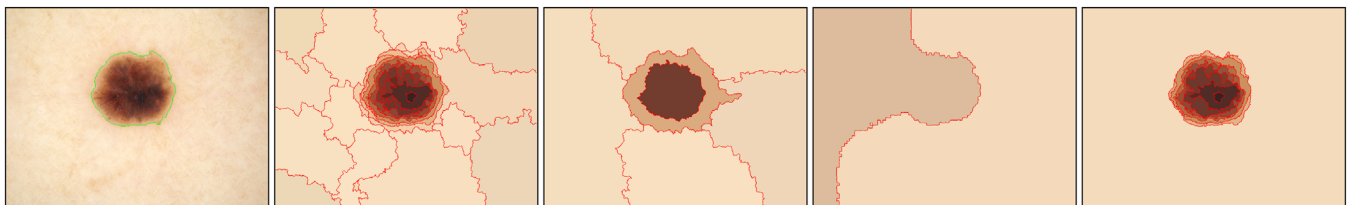


FIGURE 19 Qualitative results. From left to right: ground truth, LPCI-SP, LV-SP, FG-SUTP and ILP-GAU-CM (our)

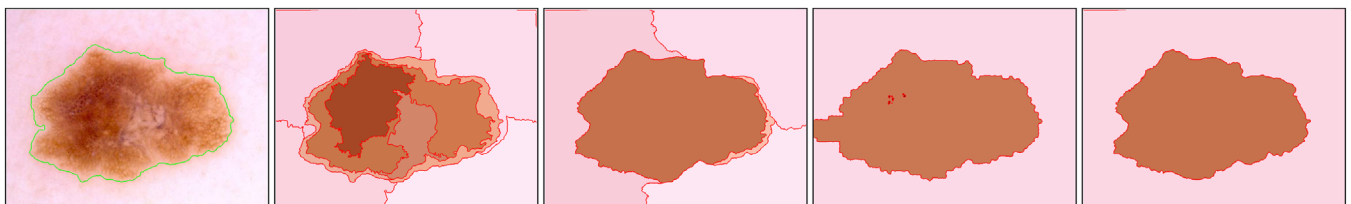


FIGURE 20 Qualitative results. From left to right: ground truth, LPCI-SP, LV-SP, FG-SUTP and ILP-GAU-CM (our)

It is clear that the majority of label updates occurs during the first three iterations. As the number of iterations increases, no more than 10 exchanges occur. Such small changes in labels have no major impact in segmentation, but may represent a considerable gain in processing time, if a cut-off at iteration three is established. Therefore, we can define the ILP* method, which consists of the ILP method with only three iterations.

5.2 | Comparison with related methods

We also compare the proposed ILP, RLP, and WRLP methods with other segmentation methods based on graphs and complex networks. The chosen methods are: EGBIS [12], SUTP-FG [5, 6], LV [29] and LPCI [5]. We also included the ILP* method, which uses only three iterations.

The values for parameters H and T used in this comparison are shown in Table 2. These were selected according to our previous experimental evaluation, whose results were shown in Table 1. We employed the descriptor CM and the similarity functions GAU and TAN. As for the other methods, we adopted the parameter setup suggested by the authors.

The original EGBIS-P method [12] is a pixel-based segmentation method, with no super-pixel support. For fairness, we implemented a super-pixel version named EGBIS-SP. The SNIC method was employed to generate super-pixels with initial lengths equal $S = 19$, thus yielding a total of $k = N/S^2$ super-pixels (N is the number of pixels in the image). The vertices were created using CM to describe super-pixels. The edges were created with the neighbourhood level $H = 1$ and Euclidean distance as a weight function. As in the original EGBIS-P, threshold was not included.

The SUTP-FG method [5, 6] was run with the following configuration (as proposed by the authors): SUTP [26] super-pixel method, with initial side equal 10 and neighbourhood radius $R = 5$. Super-pixels are described by a three-dimensional colour descriptor based on the CIELAB average value of all pixels belonging to a super pixel. The graph construction relies on a weight function based on the Euclidean distance, and edges are created for distances shorter than or equal 6.

We also implemented a variation of the LV method [29] named LV-SP that includes super-pixels with the following configuration: the SNIC method for super-pixels extraction and initial side equals $S = 19$, descriptor CM, neighbourhood level $H = 3$, weight function GAU and threshold $T = 0.25$.

The LPCI-SP was executed with a similar setup presented by Cuadros et al. [5]. We used the SNIC method to extract super-pixels with initial side length $S = 19$, descriptor CM, similarity function GAU and threshold parameter $T = 0.25$.

Quantitative and qualitative results are illustrated in Table 2 and Figures 13–16, respectively. Both results show that the proposed methods ILP, ILP*, RLP, and WRLP have better performance than the traditional methods. Among the various combinations of similarity functions and super-pixel descriptors, the best were ILP-GAU-CM and ILP-TAN-CM, with $H = 3$. As

for segmentation quality and processing times, we can see that the other combinations of our methods also have similar performance.

5.3 | Application to medical images

We also evaluated our proposed methods on a set of coloured medical images (ISIC2016 [43]). This dataset was originally proposed as a classification skin lesion challenge. It consists of two training (900) and test (379) subsets. Each skin lesion has a manual segmentation image (ground truth). In this evaluation, we selected 50 images from the test subset.

The same set of parameters used for the segmentation of natural scenes (BSDS500-validation) were employed, as described in the Section 5.2.

The quantitative results for the 50 images of the ISIC2016 dataset are presented in Table 3. The superiority of the proposed methods for neighbourhood levels $H = 3$ can be confirmed, by observing the metrics of Covering and PRI. However, the low values of the BF1-Score is a result of the reduced amount of contour information presented in the ISIC2016 dataset.

The qualitative evaluation depicted in Figures 17–20 for the ILP-GAU-CM method corresponds to the highest values of the PRI metric shown in Table 3.

The results presented for the segmentation of coloured skin lesions, also shows the feasibility of the proposed methods for other medical images modalities, not necessarily coloured-based ones, such as 3D magnetic resonances data. In this scenario, needed adaptations to the current method would be: implementation of the super-pixel technique for super-voxels, implementing descriptors for super-voxels, determining the neighbourhood level in three dimensions and the implementation of the methods of label propagation in 3D space.

6 | CONCLUSIONS



In this article, we presented three fast and accurate methods for automatic image segmentation, based on a re-formulation of the label propagation method used to identify communities in complex networks. Our methods incorporate three deterministic propagation strategies that take into account information from the image domain. The label propagation is performed on an undirected weighted graph that is a representation of the image. Edges are created using a multilevel neighbourhood strategy among vertices. Several experiments carried out with the BSDS500 and ISIC2016 datasets have shown that the proposed methods are fast and competitive in terms of segmentation quality when compared with related methods.

Among the proposed methods, the ILP-GAU-CM is the one with the best results for both datasets. Thus, the results obtained may suggest a possible extension of the ILP-GAU-CM method for medical images. In the case of magnetic resonance images, the methodology could be adapted for the 3D nature of the data.

ACKNOWLEDGEMENTS

This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors also would like to thank FAPESP for the financial support, Grant Numbers: 2018/06074-0 and 2021/00360-3.

ORCID

Ivar Vargas Belizario  <https://orcid.org/0000-0001-5970-2283>
Oscar Cuadros Linares  <https://orcid.org/0000-0003-1454-399X>

REFERENCES

- Sonka, M., et al.: Image Processing, Analysis, and Machine Vision. Cengage Learning, Boston (2014)
- Garcia-Lamont, F., et al.: Segmentation of images by color features: A survey. *Neurocomputing* 292, 1–27 (2018). <http://www.sciencedirect.com/science/article/pii/S0925231218302364>
- Zaitoun, N.M., Aqel, M.J.: Survey on image segmentation techniques. *Procedia Comput. Sci.* 65, 797–806 (2015). <http://www.sciencedirect.com/science/article/pii/S1877050915028574>
- Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(8), 888–905 (2000)
- Cuadros, O., et al.: Segmentation of large images with complex networks. In: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 24–31. IEEE, Piscataway (2012)
- Linares, O.A.C., et al.: Segmentation of large images based on super-pixels and community detection in graphs. *IET Image Proc.* 11(12), 1219–1228 (2017)
- Cigla, C., Alatan, A.A.: Efficient graph-based image segmentation via speeded-up turbo pixels. In: 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 3013–3016. IEEE, Piscataway (2010)
- Nguyen, T., et al.: A new image segmentation approach based on the Louvain algorithm. In: 2018 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1–6. IEEE, Piscataway (2018)
- Clauset, A., et al.: Finding community structure in very large networks. *Phys. Rev. E* 70, 066111 (2004)
- Raghavan, U.N., et al.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* 76, 036106 (2007)
- Blondel, V.D., et al.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* 10, 10008 (2008)
- Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vision* 59(2), 167–181 (2004)
- Šubelj, L., Bajec, M.: Unfolding communities in large complex networks: Combining defensive and offensive label propagation for core extraction. *Phys. Rev. E* 83, 036103 (2011). <https://link.aps.org/doi/10.1103/PhysRevE.83.036103>
- Ilesanmi, A.E., et al.: Multiscale superpixel method for segmentation of breast ultrasound. *Comput. Biol. Med.* 125, 103879 (2020). <http://www.sciencedirect.com/science/article/pii/S0010482520302328>
- Poap, D., Woźniak, M.: Lung segmentation on X-ray images with neural validation. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7. IEEE, Piscataway (2017)
- Poap, D., et al.: Chest radiographs segmentation by the use of nature-inspired algorithm for lung disease detection. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2298–2303. IEEE, Piscataway (2018)
- Wang, Y., et al.: Image segmentation of brain mri based on ltrdp and superpixels of improved slic. *Brain Sci.* 10(2), 116 (2020). <https://doi.org/10.3390/brainsci10020116>
- Wang, Q., et al.: A joint convolutional neural networks and context transfer for street scenes labeling. *IEEE Trans. Intell. Transp. Syst.* 19(5), 1457–1470 (2018)
- Wang, Q., et al.: Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. *IEEE Trans. Image Process.* 28(9), 4376–4386 (2019)
- Fowlkes, C., et al.: Spectral grouping using the nystrom method. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(2), 214–225 (2004)
- Cour, T., et al.: Spectral segmentation with multiscale graph decomposition. In: *Computer Vision and Pattern Recognition, 2005 (CVPR 2005)*, pp. 1124–1131. IEEE, Piscataway (2005)
- Tao, W., et al.: Color image segmentation based on mean shift and normalized cuts. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 37(5), 1382–1389 (2007)
- Cong, L., et al.: Image segmentation algorithm based on superpixel clustering. *IET Image Proc.* 12(11), 2030–2035 (2018)
- Casaca, W., et al.: Spectral image segmentation using image decomposition and inner product-based metric. *J. Math. Imaging Vision* 45(3), 227–238 (2013)
- Cheng, M.M., et al.: Hfs: Hierarchical feature selection for efficient image segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision – ECCV 2016*, pp. 867–882. Springer International Publishing, Cham (2016)
- Levinshstein, A., et al.: Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.* 31(12), 2290–2297 (2009)
- Ojala, T., et al.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* 29(1), 51–59 (1996). <http://www.sciencedirect.com/science/article/pii/0031320395000674>
- Qian, X., et al.: PLBP: An effective local binary patterns texture descriptor with pyramid representation. *Pattern Recognit.* 44(10), 2502–2515 (2011). <http://www.sciencedirect.com/science/article/pii/S0031320311001336>
- Mourchid, Y., et al.: A new image segmentation approach using community detection algorithms. In: 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 648–653. IEEE, Piscataway (2015)
- Achanta, R., et al.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(11), 2274–2282 (2012)
- Achanta, R., Süsstrunk, S.: Superpixels and polygons using simple non-iterative clustering. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4895–4904. IEEE, Piscataway (2017)
- Crevier, D.: Image segmentation algorithm development using ground truth image data sets. *Comput. Vis. Image Underst.* 112(2), 143–159 (2008). <https://doi.org/10.1016/j.cviu.2008.02.002>
- Arbelaez, P., et al.: From contours to regions: An empirical evaluation. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2294–2301. IEEE, Piscataway (2009)
- Ge, F., et al.: Image-segmentation evaluation from the perspective of salient object extraction. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1146–1153. IEEE, Piscataway (2006)
- Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: Rajpoot, N.M., Bhalerao, A.H. (eds.) *British Machine Vision Conference 2007*, pp. 1–10. British Machine Vision Association, Durham (2007) Available from: <http://dblp.uni-trier.de/db/conf/bmvc/bmvc2007.html#MalisiewiczE07>
- Huang, Q., Dom, B.: Quantitative methods of evaluating image segmentation. In: *Proceedings of International Conference on Image Processing*, vol. 3, pp. 53–56. (1995)
- Martin, D.R., et al.: Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(5), 530–549 (2004)
- Estrada, F.J., Jepson, A.D.: Benchmarking image segmentation algorithms. *Int. J. Comput. Vision* 85(2), 167–181 (2009). <https://doi.org/10.1007/s11263-009-0251-z>
- Dogra, D.P., et al.: Evaluation of segmentation techniques using region area and boundary matching information. *J. Visual Commun. Image Represent.* 23(1), 150–160 (2012). <http://www.sciencedirect.com/science/article/pii/S1047320311001210>

40. Fernandez-Moral, E., et al.: A new metric for evaluating semantic segmentation: Leveraging global and contour accuracy. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1051–1056. IEEE, Piscataway (2018)
41. Unnikrishnan, R., Hebert, M.: Measures of similarity. In: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05), vol. 1, pp. 394–394. IEEE, Piscataway (2005)
42. Arbeláez, P., et al.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(5), 898–916 (2011)
43. Gutman, D., et al.: Skin lesion analysis toward melanoma detection: A challenge at the International Symposium on Biomedical Imaging (ISBI) 2016,

hosted by the International Skin Imaging Collaboration (ISIC). eprint arXiv:1605.01397 (2016)

How to cite this article: Belizario IV, Linares OC, Neto JdESB. Automatic image segmentation based on label propagation. *IET Image Process.* 2021;15:2532–2547.

<https://doi.org/10.1049/ipr2.12242>