Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-08391-6

Learning contextual superpixel similarity for consistent image segmentation

Mahaman Sani Chaibou[1,2,3] · **Pierre-Henri Conze**[3,4] · **Karim Kalti**[1,5] · **Mohamed Ali Mahjoub**[1] · **Basel Solaiman**[3]

Received: 4 February 2019 / Revised: 10 August 2019 / Accepted: 13 October 2019 /
Published online: 25 November 2019

Abstract
This paper addresses the problem of image segmentation by iterative region aggregations starting from an initial superpixel decomposition. Classical approaches for this task compute superpixel similarity using distance measures between superpixel descriptor vectors. This usually poses the well-known problem of the semantic gap and fails to properly aggregate visually non-homogeneous superpixels that belong to the same high-level object. This work proposes to use random forests to learn the merging probability between adjacent superpixels in order to overcome the aforementioned issues. Compared to existing works, this approach learns the fusion rules without explicit similarity measure computation. We also introduce a new superpixel context descriptor to strengthen the learned characteristics towards better similarity prediction. Image segmentation is then achieved by iteratively merging the most similar superpixel pairs selected using a similarity weighting objective function. Experimental results of our approach on four datasets including DAVIS 2017 and ISIC 2018 show its potential compared to state-of-the-art approaches.

Keywords Context description · Superpixels similarity · Machine learning · Random forests · Image segmentation · Region-growing

1 Introduction

Image segmentation is a fundamental task in many pattern recognition and computer vision applications such as object detection, content-based image retrieval and medical image

✉ Mahaman Sani Chaibou
sallaoudt@gmail.com

[1] Université de Sousse, Ecole Nationale d'Ingénieurs de Sousse, LATIS - Laboratory of Advanced Technology and Intelligent Systems, 4023, Sousse, Tunisie

[2] Institut Supérieur d'Informatique et des Techniques de Communication, Université de Sousse, 4011, Hammam Sousse, Tunisie

[3] IMT Atlantique, Technopôle Brest-Iroise, CS 83818, 29238, Brest Cedex 03, France

[4] LaTIM UMR 1101, Inserm, IBRBS, 22 rue Camille Desmoulins, 29238, Brest, France

[5] Faculté des sciences de Monatir, Université de Monastir, 5019, Monastir, Tunisie

analysis. Segmentation is the process that consists of partitioning an image into homogeneous regions of pixels with similar characteristics and spatially accurate boundaries [22].

Region-growing is a popular image segmentation technique that operates by merging regions with similar pixels on their borders in an iterative fashion. At each iteration, all pixels that border the growing region are examined and the most similar are aggregated to that region. Initial regions may be pixels or regions produced by dedicated over-segmentation techniques, in which case they are called superpixels [34].

This paper addresses the problem of segmenting an image starting from an initial superpixel decomposition. In the majority of classical approaches for this task, the computation of the superpixel similarity is performed using distance measurements between superpixel feature vectors. This generally poses the problem of the semantic gap and thus does not allow to correctly aggregate visually non-homogeneous superpixels belonging to the same high-level object. In this work we propose to use random forests to learn the probability of fusion between adjacent superpixels in order to overcome the aforementioned problems. The core contribution is the use of random forests (RF) to learn how to merge neighboring superpixels. RF were introduced in the machine learning community by [2, 8]. In the computer vision community their popularity arose mainly from [24, 30]. They are now widely used for various supervised classification tasks from image classification to video segmentation [6, 13, 48].

In the latter topic, videos can contain single or multiple objects of interest to be tracked throughout the frames of the sequence. Occlusions, deformations, or interactions with one another make segmentation and tracking tasks very challenging. The effectiveness of a segmentation algorithm is shown by the way it addresses these challenges. Several tasks can be build on top of the segmentation results, such as saliency detection [11, 12], object tracking [14, 17], and scene analysis [44].

The rest of this paper is organized as follows. A literature review on this work is presented in Section 2. In Section 3, we describe the proposed approach to learn superpixel similarity for aggregations. The iterative image segmentation approach is detailed in Section 4. Section 5 provides experimental results and discussions. Conclusions are given in Section 6.

2 Related works

Superpixels over-segmentations are usually used as an initilization for image segmentation for two main reasons [49]. Firstly, they generally adhere to boundaries and produce semantically meaningful small regions. Secondly, they drastically reduce computation time by reducing the number of processed entities. [23] proposed a segmentation approach for natural images that uses Simple Linear Iterative Clustering (SLIC) [1] as superpixel generation technique. They first regroup superpixels into clusters by spectral clustering and then perform a merging step using explicitly established cluster similarities. Another work from [46] presented a kernel fuzzy similarity measure which is used to cluster superpixels. A 10-dimensional texture-based feature vector is extracted to characterize each superpixel. After clustering, a k-means final step is applied to group clusters into final regions. [50] proposed a graph-based coarse and fine merging strategy based on features extracted on superpixels including three Gestalt laws-inspired rules to model the superpixel context. In the same direction, [29] proposed a segmentation algorithm for action and event detection in videos. Similarity between regions is computed based on appearance, motion and geodesic spatio-temporal features. A hierarchical clustering with average linkage is then applied on the graph to produce final segmentation results. However this approach does not take into

consideration the fact that when regions become bigger the similarity does not conserve the same meaning. This means that a similarity value α will not reflect the same visual similarity when the sizes of the compared regions are disproportionate. [10] remedy to this flaw by adding a two-term similarity measure: a border-based and a content-based similarity. The first component captures the similarity between two regions on their shared border only while the second gives their similarity over the whole region.

Recently, segmentation-based tracking algorithms have been investigated actively [14, 17, 39]. Wang et al. [45] uses superpixels for discriminative appearance modeling by meanshift clustering, and they incorporate particle filtering to find the optimal target state. The authors in [39] proposed a tracking-by-segmentation algorithm that combines the information from pixels with bounding boxes. They focused on the only on the tracking and used an external technique for segmentation. In [47] authors proposed an algorithm using Absorbing Markov Chains (AMC) on superpixel segmentation, where target state is estimated by a combination of bottom-up and top-down approaches. From a graph of superpixels between consecutive frames, where background superpixels in the previous frame correspond to absorbing vertices and all other superpixels transient ones, the algorithm achieves target segmentation using the absorption time of each superpixel. Graph edges are weighted by the similarity of scores in the end superpixels, learned by support vector regression. Most of these tracking algorithms rely solely on pixel-level information that is not sufficient to semantically and correctly model the targeted objects.

In all these works, we can note that superpixels to be merged are assumed to satisfy two important criteria: spatial adjacency and perceptual similarity. This suggests that an efficient approach should be able to pick the most visually similar spatial neighbor of a superpixel. Superpixel similarity is largely measured using a mathematical function which introduces the well-known problem of semantic gap. The latter refers to the difference between the description of an object and its actual characteristics. Furthermore, those measures can hardly merge visually non-homogeneous superpixels even when they belong to the same semantic object.

In this paper, we present a superpixel-based segmentation approach that uses superpixels as inputs. Instead of relying on an explicit similarity measure to compare superpixels, we propose to train a Machine Learning (ML) classifier to infer the likelihood of two superpixels to be merged. This will particularly allow to seamlessly merge visually dissimilar superpixels that belong to the same semantic object. In particular, we choose the random forests technique to learn superpixel mergeability. This choice is based on several properties of the RF classifier including computational efficiency, robustness against outliers and probabilistic output. The learned model is then used to iteratively aggregate regions without explicitly computing similarity between superpixels, up to final semantic segmentation.

3 Learning superpixel aggregation

This paper addresses the problem of image segmentation by iterative region aggregations in a single scene image dataset. We assume that all images in the dataset, referred to as $\mathcal{D}$, depict the same scene composed of K object classes, $C_1, C_2, \ldots, C_K$. Moreover, $\mathcal{D}$ is split into two subsets: a training set $\mathcal{D}_{train}$ and a test set $\mathcal{D}_{test}$. $\mathcal{D}_{train}$ gathers randomly selected images from $\mathcal{D}$ with label annotations available whereas $\mathcal{D}_{test} = \mathcal{D} \setminus \mathcal{D}_{train}$ provides the images to be segmented. An example of such dataset is a video sequence where a random subset of the frames are provided with groundtruth label annotations.

We propose an approach based on machine learning that first builds a superpixel merging model from $\mathcal{D}_{train}$ and then achieves segmentation of images from $\mathcal{D}_{test}$ through iterative superpixel aggregations. The proposed approach is made of two main phases. Firstly, a random forests (RF) classifier training phase (Section 3.4) is performed over an initial superpixel decomposition of images from $\mathcal{D}_{train}$ in order to learn how to merge superpixels. In other words, the RF classifier is trained to predict whether two given neighboring superpixels belong to the same semantic object or not. Secondly, a segmentation phase (Section 4.2) is carried out, for any given image from $\mathcal{D}_{test}$, through iterative superpixel aggregations based on RF predictions. In particular, using the previously trained RF classifier, similar neighboring superpixels are iteratively merged by means of their predicted merging probability. The iterative step reaches a final segmentation of the image content into homogenous partitions. Figure 1 presents an overview of the proposed approach. This work presents two main contributions: (1) a machine learning-based mechanism for superpixel merging and (2) a graph-based aggregation technique to achieve image segmentation.

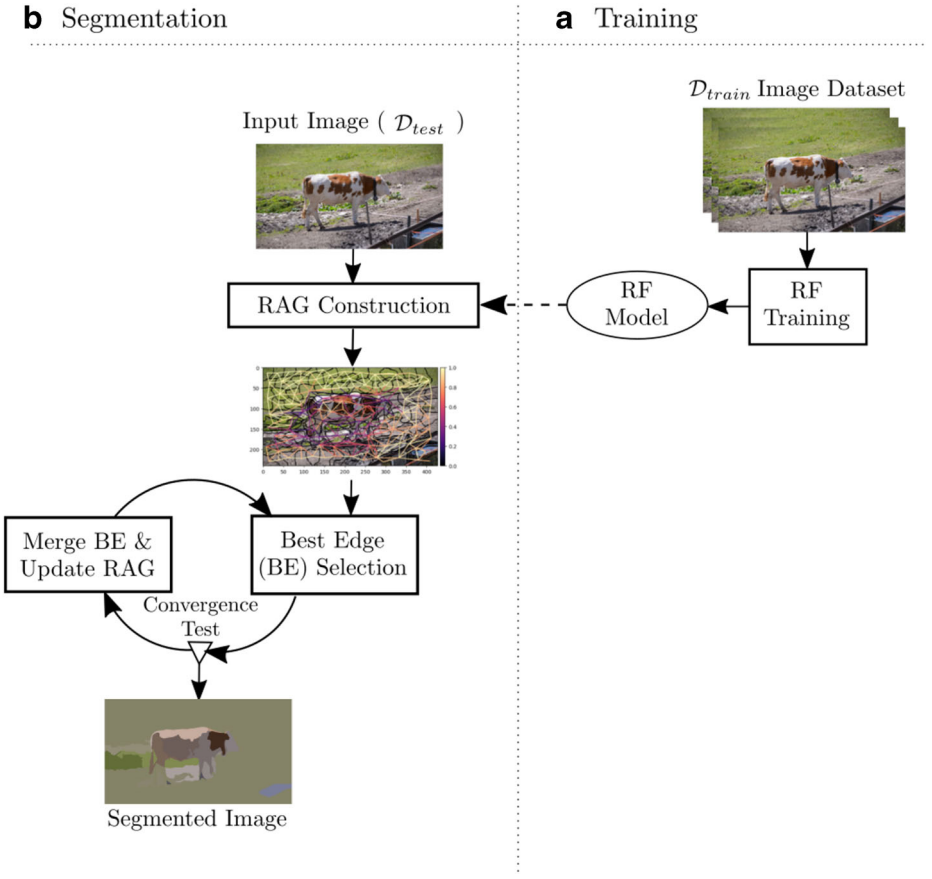

Fig. 1 Segmentation approach illustration. A prior training phase to build the RF model. A segmentation phase for input image segmentation. Image is first over-segmented and a region adjacency graph (RAG) is built and weighted with superpixel similarity using the RF model. From the weighted RAG, Best Edges (BE) are iteratively selected and merged until segmentation convergence

By using RF classifier to learn superpixel aggregations, the proposed image segmentation avoids the semantic gap related to similarity measure computation in region-growing-like approaches. In addition, the use of superpixels instead of pixels allows a better characterization and a drastic reduction in terms of computation time.

3.1 Initial superpixel decomposition

In this step, an over-segmentation technique is used to transform an image $\mathcal{I}$, by regrouping similar neighboring pixels, into a set of small perceptually meaningful pixel groups called superpixels [34] denoted here by $\mathcal{S}p_{\mathcal{I}}$. Several approaches devoted to this task are developed in the literature. In [40], superpixel over-segmentation algorithms are organized into seven categories: watershed-based, density-based, graph-based, contour-based, path-based, clustering-based and energy-based optimization. These approaches are used in image segmentation mostly as preprocessing steps to reduce computations time. Indeed, many pixel-based segmentation approaches can be applied on over-segmented images with negligible performance lost and huge time gain. In this work, we used the Simple Linear Iterative Clustering (SLIC) [1] algorithm to perform superpixel decomposition. SLIC is a clustering-based algorithm that uses a similarity measure based on color and spatial cues to generate superpixels. It offers a simple implementation and provides compact and nearly uniform superpixels. The bigger the number of generated superpixels, the finer the over-segmentation, as shown in the Fig. 2.

3.2 Superpixel context description

In region characterization, context information provides robustness against noise since it provides a broade and consistent view of the region. Superpixel-based segmentation approaches offer several configurations to characterize the local context of a superpixel. Among recent works, in [43] proposes to integrate the local context into thematic maps representing the superpixels. In this work, each superpixel of the image is described using

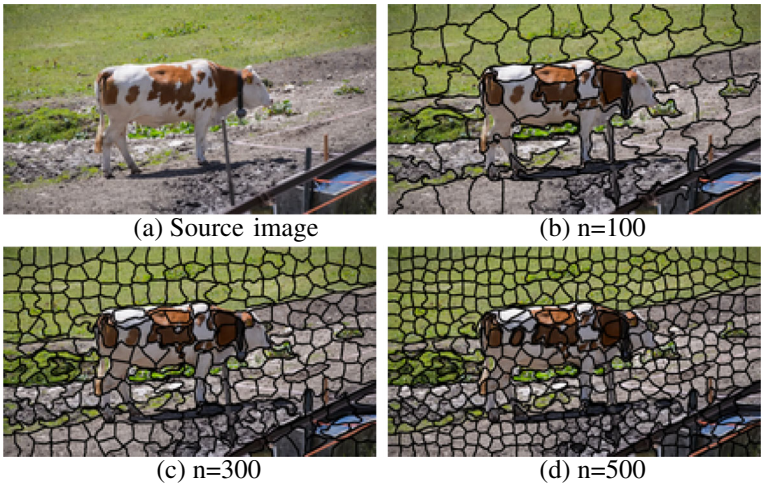

Fig. 2 Image over-segmentation using SLIC [1] algorithm. An image (2a) and its corresponding decomposition into 100 (2b), 300 (2c) and 500 (2d) superpixels

a histogram of its bag of visual words. The subsequent contextual information is encoded by concatenation of the superpixel description with a combination of the histograms of its neighbors. One of the main drawbacks of this method is the lack of explicit coding of relational aspects among features extracted from adjacent superpixels. [36] proposed to express the local context of a superpixel by a descriptor vector, which they call the *Star* descriptor. First, they group in a vector the characteristics of neighboring superpixels. Then, they calculate a texture vector between the superpixel and each of its neighbors through an overlapping rectangle whose diagonal is formed by the centroids of the two superpixels. The context descriptor corresponds to the concatenation of the superpixel descriptor vector, the neighbors vectors and the texture vector. These previous works make a representation of the local context by a flat structure unlike in [3] where the local context is coded by a graph. Furthermore, the authors here propose to use the superpixels that are in a radius r of the center of the current superpixel to avoid the bias that can be caused by very small neighbors. In addition they preserve the spatial order of the neighbors in the used graph structure. In the same direction, [41] extended context-rich appearance features from pixels to superpixels to describe the extended spatial context from the superpixel graph.

The proposed superpixel local context descriptor is graphically illustrated by the Fig. 3. We propose a representation of the local context similar to that proposed by [3] that we apply to the superpixel resulting from the merge of the candidate superpixels. However, for a pair $p = (a, b)$, instead of considering the superpixels separately, we compute the context on p to produce a unified description. This avoids the redundancy of common neighbors but also the imbalance in the case of superpixels with a significant difference in size. Finally, computing context on the pair of superpixels allows to keep the same idea of a pair-based learning, presented in the next sections. In addition, in [3], the context of a superpixel a is defined as the set of superpixels $\{a_i\}_{i \in \mathbb{N}}$ within the vicinity of radius r. This approach can

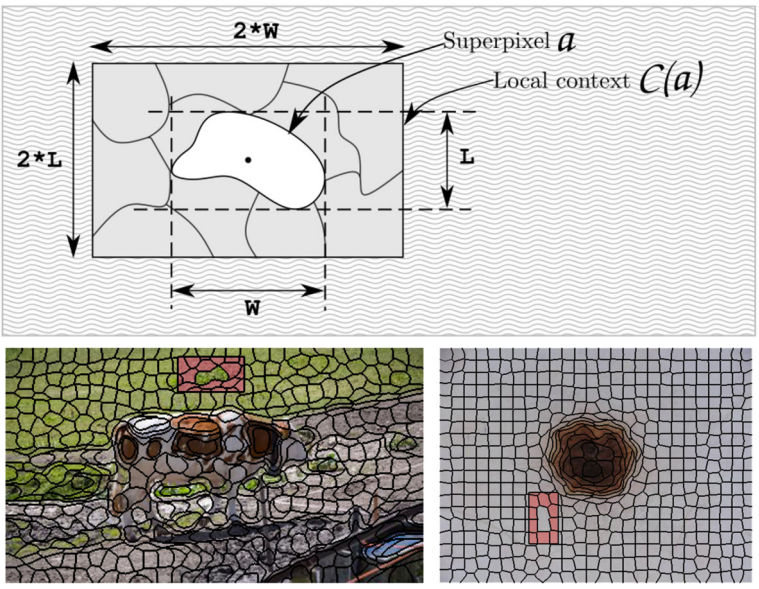

Fig. 3 The proposed context descriptor illustration. The top figure presents a superpixel a with its local context $\mathcal{C}(a)$. The bottom row show two examples of superpixel and corresponding local context (rectangle in red)

cause an imbalance when a superpixel $a_j \in \{a_i\}$ such that $|a_j| \gg |a|$ is reached by r. So, we define the context as a bounding box of the vicinity of the superpixel. First, we determine the minimum bounding box $\mathcal{R}_a$ of a, whose width and length are respectively denoted by W and L. Subsequently, the context $\mathcal{C}(a)$ to consider is contained in the rectangle with width 2×W, length 2×L and centered around the centroid of a. This allows us to avoid the empirical determination of these values and ensures that we obtain a self-adapting local context that changes according to the size of a. Indeed, in [3] whatever the size of the superpixel, r remains fixed. This constraint could distort the impact of the context because when it is too big compared to the superpixel, it will dominate the characterization of the superpixel while its effect could be negligible when it is too small w.r.t. the superpixel. Thus, the local context is formalized as follows:

$$\mathcal{C}(a) = \Big\{ p_i \mid p_i \in \mathcal{I} \text{ and } p_i \notin a \text{ and } p_{i_x} \in [x_0 - 2 \times \mathtt{W}, x_0 + 2 \times \mathtt{W}] \text{ and } p_{i_y} \in [y_0 - 2 \times \mathtt{L}, y_0 + 2 \times \mathtt{L}] \Big\} \tag{1}$$

With (x_0, y_0) the coordinates of the centroid of superpixel a. The local context $\mathcal{C}$ is represented as a single region in order to present a homogeneous view of the neighborhood of the superpixel. The context of the superpixel is described by a 100-bins color histogram.

In addition to its context description, each superpixel is characterized by a concatenation of 3×107 color features which includes, but not limited to, a 100-bins color histogram and some statistical values over the four first central color moments including the mean (E), the standard deviation (σ), the skewness ($Skew$) and the kurtosis ($Kurt$), computed as follows:

$$E(a) = \frac{1}{N} \sum_{i}^{N} (a_i) \tag{2}$$

where a_i is the i^{th} pixel of the superpixel a and N is the total number of pixels in a. The three other values are computed based on E value as follows $\sigma(a) = \sqrt{m_2(a)}$, $Skew(a) = \frac{m_3(a)}{[\sigma(a)]^3}$ and $Kurt(a) = \frac{m_4(a)}{[\sigma(a)]^4}$ with:

$$m_j(a) = \frac{1}{N} \sum_{i}^{N} [a_i - E(a)]^j \tag{3}$$

3.3 Random forests classifier

Random forests are an ensemble learning technique usually employed for classification and regression tasks. RF were introduced in the machine learning community by [2, 8]. In the computer vision community their popularity arose mainly from [24, 30]. They consist of a collection of T uncorrelated decision trees trained with a subset of M samples randomly extracted from the training dataset $\mathcal{X} = \{x_i,\ i = 1..\mathtt{N}\}$ where x_i describes the i^{th} sample data. During the learning phase, randomness can be injected to achieve independence between trees constructed from the same training set [31]. From a subset $\mathcal{X}_t$ the corresponding decision tree $\mathcal{T}_t$ is built by recursive bipartition of $\mathcal{X}_t$ until it meets certain condition. Starting by associating $\mathcal{X}_t$ to the root node of $\mathcal{T}_t$, the algorithm generates for each internal node n_t, a binary test Φ designed to *optimally* bipartition the subset provided as inputs. The two resulting subsets feed two child nodes considered as direct children of n_t. This process

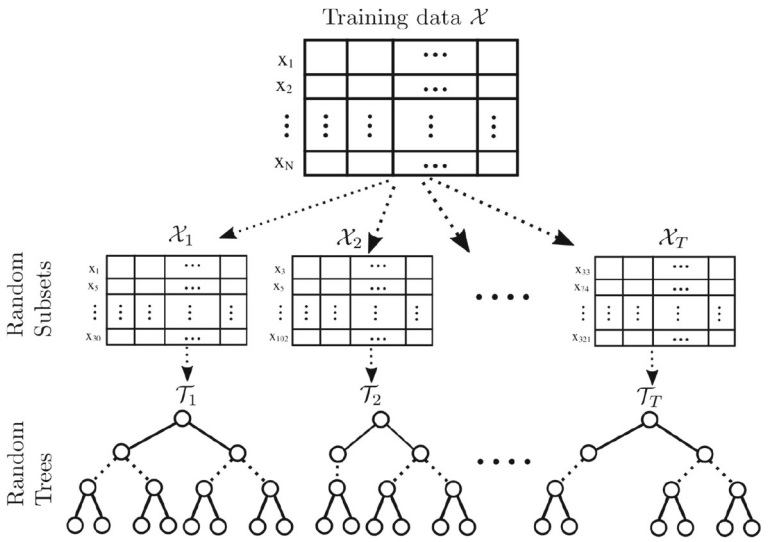

Fig. 4 Random Forest training illustration. A random forest classifier is a set of binary decision trees. Each tree is built from a random subset of the training data. Each internal node has a logical test which is used to split its input data into two subsets

is repeated until there is no more possible bipartition. Leaf nodes contain labels probabilities. The bipartition test Φ can be obtained by selecting the split with the impurity lower than a given threshold using an impurity measure, commonly the Gini criterion.

During classification, as shown in Fig. 4, an input sample data x_0 is passed down throughout all the trees. The internal nodes are used to push x_0 to one of their child nodes, depending on the result of the associated binary test Φ, until it reaches a leaf node. Each tree will produce a local class for the input sample and its final class is the most voted class over all the trees. A simplified visual illustration of the RF classifier prediction process is presented in Fig. 5.

Random forests have been demonstrated to produce performance comparable to SVM in multi-class problems [6] while maintaining high computational efficiency. The RF popularity is mostly due to their appealing properties which include: (i) their computational efficiency in both training and prediction, (ii) their probabilistic output, (iii) the seamless handling of a large variety of visual features (e.g. color, texture, shape, depth etc.), and (iv) the robustness against outliers. Furthermore, comparatively to other ML techniques for classification, RF offer a very good trade-off between the size of the training set and the quality of the classification.

Usually, in a machine learning process, features describing the data do not contribute equally in the prediction of the target response. In many situations, most features are actually irrelevant. Due to the principle of their construction, decision trees intrinsically perform the feature selection by selecting the appropriate data splitting points. Several methods are proposed for feature selection in RF training including Mean Decrease in Impurity (MDI) defined in [26], Mean Decrease Accuracy (MDA) [8, 9] and Recursive Feature Elimination (RFE) [18, 19].

In this work, feature selection is achieved through the MDI approach. Selected features depend on the training data. However, from the features described in Section 3.2, only `1/3` of the color histogram values are regularly selected.

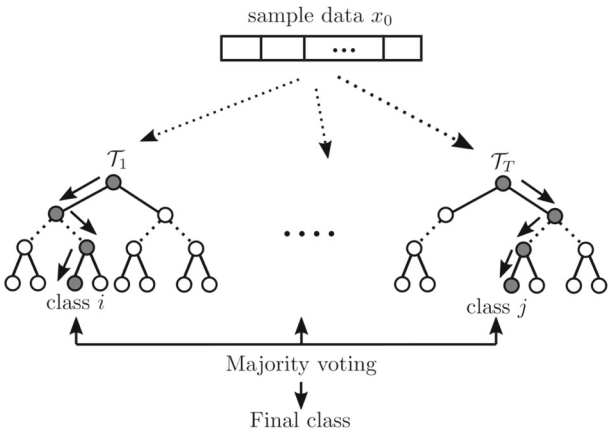

Fig. 5 Random Forest prediction illustration. A given sample data is passed through each tree until it reaches leaf nodes. Leaf nodes prediction probabilities are aggregated to produce final prediction by majority voting strategy

A learning algorithm $\mathcal{A}$ is a functional that maps a data set $\mathcal{X}$ to a function f [4]. However, most of the learning algorithms have their own parameters, known as hyper-parameters, that need to be set during their definition. Hyper-parameters are provided by the user during the construction of the estimator or indirectly fitted to the data. They provide some prior information on the learning data distribution. It is recommended to search the hyper-parameter space in order to optimize hyper-parameters of the estimator [32]. Common hyper-parameter space search strategies include Grid Search, Manual Search and Randomized Search [4]. The first two strategies proceed by instantiating the considered estimator. Then, a learning phase is iterated on all possible combinations of parameter values to finally retain the best combination. Randomized Search implements a randomized search over parameters, where each setting is sampled from a distribution over possible parameter values. This strategy is more robust and flexible than an exhaustive search [32].

3.4 Learning superpixel similarity by RF classifier

This work proposes to learn the similarity between two given regions. Given the efficiency of RF classifiers in classification problems, we suggest to model the problem of merging similar superpixels in the region gowing segmentation approaches as a classification task. Considering our initial image dataset $\mathcal{D}$, the learning phase is carried out using the training dataset $\mathcal{D}_{train}$. Each image $\mathcal{I} \in \mathcal{D}_{train}$ is decomposed into a set of superpixels $\mathcal{S}p_{\mathcal{I}}$. Let $\mathcal{P}_{\mathcal{I}}$ be the set of all the pairs of superpixels belonging to $\mathcal{S}p_{\mathcal{I}}$ and defined as

$$\mathcal{P}_{\mathcal{I}} = \{\, (a, b),\ \forall\, a, b \in \mathcal{S}p_{\mathcal{I}} \text{ and } \mathcal{N}(a, b) = 1\} \tag{4}$$

where the function $\mathcal{N}$ expresses the spatial adjacency between two superpixels, a and b by:

$$\begin{aligned} \mathcal{N} \ &: \ \mathcal{S}p \times \mathcal{S}p \rightarrow \{0, 1\} \\ \mathcal{N}(a, b) &= \begin{cases} 1, & \texttt{if } a \texttt{ is adjacent to } b \\ 0, & \texttt{otherwise} \end{cases} \end{aligned} \tag{5}$$

The elements of $\mathcal{P}_{\mathcal{I}}$ can be divided into two groups: mergeable pairs $\mathcal{P}^m_{\mathcal{I}}$ and non-mergeable pairs $\mathcal{P}_{\mathcal{I}} \setminus \mathcal{P}^m_{\mathcal{I}}$. Therefore, a pair (a, b) of adjacent superpixels a and b is labeled

with `mergeable` when they both belong to the same class C_i. Conversely, if a and b do not belong to the same class, then the merging label of (a, b) is `unmergeable`.

$$\mathcal{P}_{\mathcal{I}}^{m} = \{\, (a,b),\ \forall\, (a,b) \in \mathcal{P}_{\mathcal{I}} \mid \mathcal{N}(a,b) = 1,\ a,b \in C_i \,\} \tag{6}$$

From here, we can train a RF classifier to learn the mergeability for superpixel pairs. Once trained, the RF model is able to provide an aggregation probability for any input pair of superpixels.

3.4.1 RF classifier training

A pseudocode of this process is described in the algorithm 1. From the prepared sets, RF training data $\mathcal{X}$ consists of pairs of superpixel feature vectors associated with their corresponding merging label. Considering two superpixels a and b with intrinsic feature vectors f_a and f_b respectively, the feature vector of the superpixel pair (a, b) is computed as the concatenation of f_a, f_b and $\mathcal{C}(a, b)$. Thus, we have:

$$\mathcal{X} = \{\, (f_a,\, f_b, \mathcal{C}(a,b), \ell), \quad \forall\, (a,b) \in \mathcal{P}_{\mathcal{I}},\ \mathcal{I} \in \mathcal{D} \,\} \tag{7}$$

where the groundtruth class associated with the superpixel pair (a, b) is its merging label ℓ, given by:

$$\ell = \begin{cases} \texttt{mergeable}, & \texttt{if } (a,b) \in \mathcal{P}_{\mathcal{I}}^{m} \\ \texttt{unmergeable}, & \texttt{otherwise} \end{cases} \tag{8}$$

Algorithm 1 Training.

Input: $\mathcal{D}$: training image set, $\mathcal{D}_{gt}$: training groundtruth set
Output: $\hat{\mathcal{Y}}$: RF similarity prediction model

$\mathcal{X} = \emptyset$
foreach $\mathcal{I}, \mathcal{I}_{gt} \in \{\mathcal{D}, \mathcal{D}_{gt}\}$ **do**
  for $(i = \alpha_{min};\quad i < \alpha_{max};\quad i = i + \delta)$ **do** `// multi-level SLIC decomposition`
    $\mathcal{S}_p := \texttt{SLIC}(\mathcal{I}, i)$
    foreach $(a, b) \in \mathcal{S}_p \times \mathcal{S}_p$ **do**
      $f_a := \texttt{featureVector}(a)$
      $f_b := \texttt{featureVector}(b)$
      $\mathcal{C}(a, b) := \texttt{localContext}(a, b)$
      $\mathcal{L} := \mathcal{I}_{gt}(a) \cup \mathcal{I}_{gt}(b)$ `// GT labels covered by a and b`
      if $|\mathcal{L}| = 1$ **then** `// if a and b under same GT label`
        $\mathcal{X} := \mathcal{X} \cup (f_a,\, f_b, \mathcal{C}(a,b), \texttt{mergeable})$
      else
        $\mathcal{X} := \mathcal{X} \cup (f_a,\, f_b, \mathcal{C}(a,b), \texttt{unmergeable})$
$\hat{\mathcal{Y}} := \texttt{trainRF}(\mathcal{X})$

3.4.2 RF similarity prediction

The training phase produces a superpixel pairs classification model $\hat{\mathcal{Y}}$. For any pair of superpixels (a, b),

$$\hat{\mathcal{Y}}(a, b) = y, \;\; y \in \{\texttt{mergeable}, \texttt{unmergeable}\} \tag{9}$$

gives the predicted label of the pair (a, b). Actually, the RF classifier model provides, for the given pair, a probability value ω for each label. Therefore, we use the probability output form of the (9) as follows

$$\begin{cases} \tilde{\mathcal{Y}}(a, b) = \{\omega_m, \omega_u\}, \\ 0 \leqslant \omega_m, \omega_u \leqslant 1 \;\; \texttt{and} \;\; \omega_m + \omega_u = 1 \end{cases} \tag{10}$$

The assigned label is the one with the highest probability.

3.5 Multi-level superpixel decomposition

A simple superpixel decomposition uses a specific image over-segmentation algorithm to generate superpixels from input image. Figure 2 shows some examples using the SLIC [1] algorithm. This approach produces a regular set of superpixels used for mergeability learning. Thus the classifier is only trained on this particular type of superpixels, which can affect its predictions when it comes to superpixels with differents properties. To overcome this flaws we propose to generate a multi-level superpixel decomposition of each training image to train the RF classifier.

Indeed, the training data completely determines the scope and efficiency of the built model. Thus, instead of using one single decomposition, of the training images, into a predefined number of superpixels, we generate several decompositions of each image varying the number of superpixels from α_{min} to α_{max} with a predefined step δ with $\alpha_{min} < \alpha_{max}$ and $\delta > 0$. This results in a multi-level superpixel decomposition of the image going from coarser superpixels, for small values of α, to finer superpixels when α increases as shown in Fig. 6. Thus, the learning process is provided with different resolution views of the same image. Thereby, the (1) size of training data and the (2) variety of used cases are increased which in turn improves the robustness of the learned classification model.

4 Segmentation by RF-learning-based aggregations

Our image segmentation approach deals with RF learning-based region aggregations. Once the input image $\mathcal{I}$ is over-segmented into superpixels, a region adjacency graph $\mathcal{G}_{\mathcal{I}}$ is built from $\mathcal{S}p_{\mathcal{I}}$. Then, aggregation probabilities are computed between each pair of adjacent nodes in the graph using the trained RF classifier and finally the segmentation is performed by successively merging adjacent nodes according to computed merging probabilities. The image segmentation phase is performed over the $\mathcal{D}_{test}$ image dataset and the aggregation

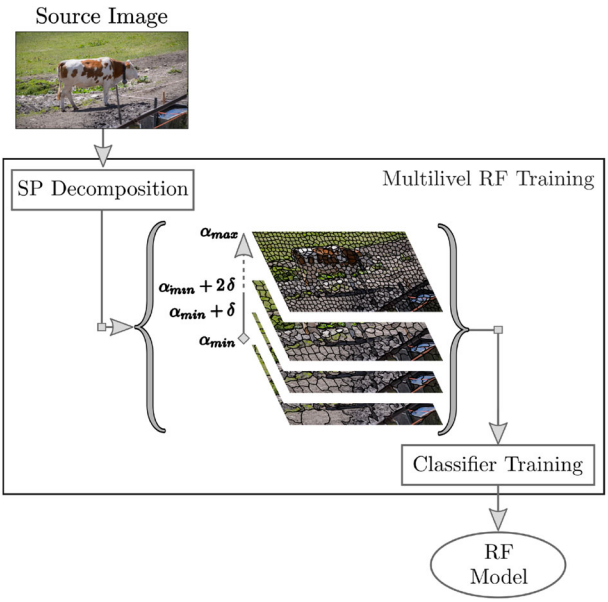

Fig. 6 Multi-level superpixel decomposition for classifier training. Given a source image $\mathcal{I}$, several decomposition into superpixels are performed. This multi-level decomposition provides learning samples from different levels of granularity, allowing the learned model to reflect more cases than through a single level decomposition

probability of a pair (a, b) is given by its probability ω_m to be labeled as `mergeable`. Algorithm 2 summarizes this process.

Algorithm 2 Segmentation.

Input: $\mathcal{I}$: image to segment, $\hat{\mathcal{Y}}$: RF similarity prediction model
Output: $\mathcal{G}$: final regions graph

$\mathcal{S}_p := \texttt{multilevelSLIC}(\mathcal{I})$
$\mathcal{G} := \texttt{buildRAG}(\mathcal{S}_p)$
$\mathcal{G} := \texttt{labelRAG}(\mathcal{G}, \hat{\mathcal{Y}})$
while $(convergence(\mathcal{G}) = \textit{False})$ **do**
  $B_E := \texttt{selectBE}(\mathcal{G})$
  $\mathcal{G} := \texttt{mergeBE}(\mathcal{G}, B_E)$
  $\mathcal{G} := \texttt{updateRAG}(\mathcal{G}, \hat{\mathcal{Y}})$

4.1 Image graph representation

The proposed segmentation approach uses an undirect weighted graph structure to represent the image. Initially, from the set of previously generated superpixels $\mathcal{S}p_{\mathcal{I}}$ a region adjacency graph (RAG) is created. The RAG $\mathcal{G}_{\mathcal{I}} = (V, E)$ of an image $\mathcal{I}$ is defined by the set of nodes V and the set of edges E. V represents the set of superpixels of $\mathcal{I}$ and E is the set of all

the pairs of superpixels linked by a neighborhood relationship. The graph is formalized as follows.

$$\mathcal{G}_{\mathcal{I}} = \{\ (v, e) \mid\ v \in V \ \texttt{and}\ e \in E\ \} \tag{11}$$

with $V = \mathcal{S}p_{\mathcal{I}}$ and $E = \{\, e \mid\ \forall\, e,\ \exists\, (a, b) \in \mathcal{S}p^2_{\mathcal{I}}$ $\mid\ e\ =\ (a, b)\ \texttt{and}\ \mathcal{N}(a, b)\ =\ 1\,\}$. In addition, the RAG edges are weighted with the merging probabilities predicted by the RF classifier. To this end, given two nodes a and b linked by an edge e in $\mathcal{G}_{\mathcal{I}}$, the merging weight ω_e represents the probability to merge the two nodes a and b as one single uniform region. Therefore, we used the trained random forest classifier $\tilde{\mathcal{Y}}$ detailed in Section 3.4 to predict the merging probability of any pair of two neighbor nodes in the graph.

4.2 RAG-based segmentation by iterative aggregations

Representing the image with a graph structure allows to solve the segmentation problem using graph cut. Therefore, the image segmentation can be interpreted as a graph partition problem. The literature of graph theory contains several works that have been carried out to address this problem [35, 38, 52].

We propose a simple and effective graph-based segmentation consisting of two steps that are recursively executed until convergence. First, given a weighted image RAG $\mathcal{G}_{\mathcal{I}}$, the best edge ε from $\mathcal{G}_{\mathcal{I}}$ is selected as the edge with the highest weight by

$$\varepsilon = \text{argmax}_{e \in E}\,[f(e)] \tag{12}$$

The function f assigns to each edge e its predicted weight ω_e. Thus, the edge ε gives the two neighboring nodes most likely to be merged in all the nodes from $\mathcal{G}_{\mathcal{I}}$. Then, assuming that the pair (a, b) forms ε, the two nodes a and b are merged into a new node c which replaces them in the RAG $\mathcal{G}_{\mathcal{I}}$. As a result, c will be linked to all neighbor nodes of a and b. Each new edge e', between c and a neighbor node n, is weighted by ω' estimated using the RF model $\tilde{\mathcal{Y}}$ as follows:

$$\omega' = \omega_m \ \mid \tilde{\mathcal{Y}}(c, n) = \{\omega_m, \omega_u\} \tag{13}$$

This produces a new RAG $\mathcal{G}'_{\mathcal{I}} = (V', E')$ representing $\mathcal{I}$, where

- $V' = \{V \setminus \{a, b\}\} \cup \{c\}$
- $E' = \{E \setminus \{\, E_a,\ E_b\,\}\} \cup \{\, E_c\,\}$ with

$E_a = \{\, e \mid e = (a, n);\ \forall n \in V\ \texttt{and}\ e \in E\}$ the set of all the edges involving the node a, $E_b = \{\, e \mid e = (b, n);\ \forall n \in V\ \texttt{and}\ e \in E\}$ the set of all the edges involving the node b and $E_c = \{\, e \mid e = (c, n);\ \forall n \in V\ \texttt{and}\ (a, n) \in E\ \texttt{or}\ (b, n) \in V\}$ the set of all the edges involving the newly added node c.

The selection of best pair of nodes to merge and the update of the RAG are iteratively repeated on the new RAG until all edge weights in the RAG are lower or equal to a threshold value ω_0.

4.3 Refining edge selection

A good segmentation approach produces results in which the internal inertia of the regions is minimal and that between these regions is maximal. Iterative approaches usually integrate these concepts by an objective function $\mathcal{J}$ which is formed of several terms associated with the targeted objectives. At each iteration, the function chooses the best regions to merge. From the graph $\mathcal{G}_{\mathcal{I}}$ of the image, we propose a function $\mathcal{J}_E$ of energy to minimize in order

to better select the edge of $\mathcal{G}_{\mathcal{I}}$ at each iteration. Therefore, the function that selects the best edge to merge is updated as follows:

$$\varepsilon = \text{argmax}_{e \in E} \left[\mathcal{J}_E(e) \times f(e) \right] \tag{14}$$

$\mathcal{J}_E$ consists of two parts: a size factor $\jmath_s$ and a context factor $\jmath_c$, added together.

$$\mathcal{J}_E(a, b) = \jmath_s(a, b) + \jmath_c(a, b) \tag{15}$$

The size factor $\jmath_s$ allows a weighting of the similarity between the pair of superpixels (a, b) that are candidates for fusion based on their size.

$$\jmath_s(a, b) = \frac{1}{2} \times \left[S_{\Delta}^{x_0, y_0} \left(\frac{min(|a|, |b|)}{max(|a|, |b|)} \right) + \frac{|a \cup b|}{|\mathcal{I}|} \right] \tag{16}$$

where $|a|$ calculates the size of the superpixel a and $S_{\Delta}^{x_0, y_0}(x) = \frac{1}{\pi} \arctan\left[\frac{x - x_0}{\Delta}\right] + y_0$ is the normalization function $arctan$ centered on x_0, of slope Δ and shifted by y_0. Note that $y_0 = 0.5$ is the value that produces normalization of values between 0 and 1. The Fig. 7 illustrates some special cases of normalization using the function $S_{\Delta}^{x_0, y_0}$. We used the values of $(x_0 = 0.5, y_0 = 0.5)$ for normalization of $\jmath_s$ and $(x_0 = 0.5, y_0 = -0.9)$ for $\jmath_c$. $|a \cup b|$ is the region covering the a and b superpixels. The first component of $\jmath_s$ ensures grouping of superpixels with similar sizes while the second component favors the merging of small sized superpixels.

Given an edge $e = (a, b)$, the context factor $\jmath_c(a, b)$ expresses the similarity of a and b in terms of their respective similarity with their neighboring superpixels. First, we define the edge context of a node a, as the vector $u = \{u_i\}_{i \in [0, \ldots, 2]}$ formed by the first three moments of the set of weights of the edges incident to a. Then, $\jmath_c$ is defined by:

$$\jmath_c(a, b) = \begin{cases} \alpha = min_{i \leqslant 2} \left\{ S_{\Delta}^{x_0, -y_0} \left(\frac{|u_i - v_i|}{u_i + v_i} \right) \right\}, \; if \; \alpha \geqslant 0 \\ 0 \; otherwise \end{cases} \tag{17}$$

u and v represent the respective context vectors of a and b. $y_0 \in \mathbb{R}$ is, in this case, a parameter used to regulate the similarity of the context distributions of the nodes. As an example,

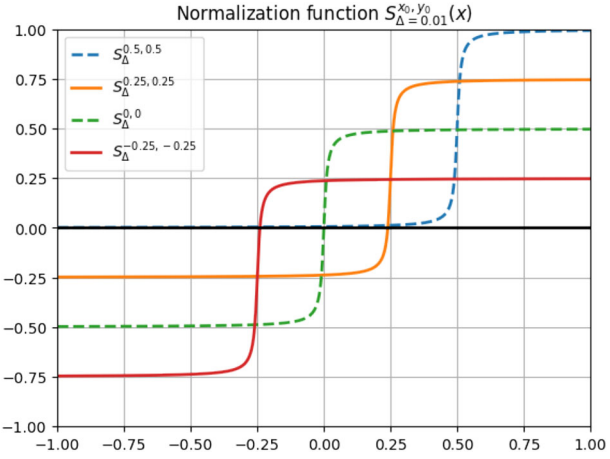

Fig. 7 Normalization function $S_{\Delta=0.01}^{x_0, y_0}(x) = \frac{1}{\pi} \arctan\left[\frac{x - x_0}{\Delta}\right] + y_0$. We used the values of $(x_0 = 0.5, y_0 = 0.5)$ for normalization of $\jmath_s$ and $(x_0 = 0.5, y_0 = -0.9)$ for $\jmath_c$

$y_0 = 0.85$ allows to impose a similarity of at least 85% between the nodes, in terms of the weight of the respective incident edges.

5 Experimentations and results

In all the experiments, the number of trees in the forest is set to 50. $\alpha_{min}, \alpha_{max}$ and δ are respectively set to 200, 1 000 and 200. Unless specified otherwise, the number of initial superpixels at segmentation phase is equal to 500. In this work the value of ω_0 is set to 0.55 in order to only allow fusion between neighbor nodes with probability higher than 55%.

5.1 Assessment datasets

The proposed segmentation approach is validated over four image datasets: DAVIS 2017 (DS17) [33], International Skin Imaging Collaboration (ISIC18), the Video Saliency (VS09) dataset [16] and the SegTrack v2 (ST2) [25].

The DAVIS dataset is publicly available and designed for video object segmentation. It gathers a collection of videos available as frame images. Thus, the images from the same video provides a single-scene image dataset. The dataset provides 60 video sequences for training purposes. For each video sequence, the classifier is provided with n = 2 randomly selected frames along with their annotations for training. The trained classifier is then used to segment the rest of the video sequence frames. Evaluation is carried out by comparison against the provided frame groundtruth (GT).

ISIC18 archives contain more than 13 000 images of skin lesions labeled as benign or malignant. We used the images prepared for the ISIC18 competition on lesion segmentation[1]. This is about 2 600 images that are each provided with the associated GT mask of lesions. Because of the different image collection conditions, we have created a subfolder containing 100 images that have similar conditions. Subsequently, we take n = 2 images randomly with their annotations for classifier learning. The segmentation phase is performed on the rest of the images.

The Video Saliency (VS09) dataset [16] is a collection of 10 video sequences composed from 936 frames where each video includes one salient object and its corresponding segmented sequences as groundtruth. The dataset includes challenging natural scenes with different types of objects with changing appearances, occlusion, motion blur, and interaction between objects.

The SegTrack v2 (ST2) [25] is a ground-truth dataset for the evaluation of segmentation accuracy in video tracking. It provides 14 video sequences with full pixel-level annotations on multiple objects at each frame within each video.

The performance of the approach is evaluated in two steps. First, in Section 5.2 we evaluate the quality of the similarity measure predicted by the RF model. Afterwards, the final segmentation results are assessed in the Section 5.3.

5.2 Superpixel similarity evaluation

ML-based superpixel similarity is the key component of the presented work. In the following section, we discuss the quality of this component and compare its results with others superpixel similarity mesures proposed in the literature.

[1] https://challenge2018.isic-archive.com/visitedon25/11/2018

To evaluate the quality of the model predictions, we calculate the following four metrics on the result weighted graphs.

- Mean Squarred Error (**MSE**) to measure the average squared difference between the estimated values and the actual ones.
- The **Sensitivity** and the **Specificity** to measure the ability to give a positive result when a hypothesis is verified and the ability to give a negative result when the hypothesis is not verified, respectively.
- Finally, the **F-measure** to evaluate the accuracy of the prediction model.

To do this, for each ω pairing probability of superpixel pair, we label by `1` when $\omega >$ `0.5`, otherwise the prediction is marked as incorrect by assigning it the tag `0`. Table 1 recapitulates the results of our experiments. The Figs. 8 and 9 expose a visual summary of such results, respectively for the DS17 and the ISIC18 datasets.

The decomposition into superpixels is a fundamental step of the work presented. The Fig. 10 presents a graph of the execution time for different values of superpixel number. This shows the computational efficiency gain by using the superpixel decomposition before segmentation. Furthermore, we have proposed two approaches to use superpixel decomposition to provide learning data: a simple decomposition (RF-SDC) and a multi-level decomposition (RF-MDC) (see Section §3.5). As shown by the results in the Fig. 8, the model obtained from a multi-level decomposition produces better predictions than the one using a single level decomposition in both datasets and over three of the four comparison criteria.

We introduced the neighborhood of the superpixel in its description in the learning phases of the RF model. The curves in Fig. 8 show a significant quality improvement made by the context descriptors added to multidecomposition (RF-MDC+CTX) into superpixels. Next, we compare the quality of the superpixel merging probability generated by the learned classifier w.r.t. three well-known similarity measures to compute the weights of the image's RAG: Hamming (HAM) [21], City-block (CTB) and the Chebyshev (CHEB) measure. From the RAG of the image, each distance is used to generate a weighted RAG where weights correspond to similarity measure over the feature vectors of the corresponding nodes. Table 1 summarizes results of the comparative evaluation of the proposed method and the three selected similarity measures. The visual results provided in Fig. 8 show that the Hamming measure produces the worst results in both datasets and over all evaluation critria. The City-block and the Chebyshev measures produce quite similar results. The proposed approach outperforms these similarity measures over all the evaluation criteria except for Specificty. Indeed, our approach allows to double up the quality in the three criteria for instance in terms of Sensitivity our approach's best rest result of `0.962` is twice better than the best of the compared measure with `0.504`. The same goes with the MSE and the F-measure, respectively for `0.078` to `0.531` and `0.958` to `0.627`. This shows that the trained RF model best handles similarity computation between superpixel pairs which is mainly due to the model learning approach since we consider superpixel pairs as elementary entities instead of superpixels to compute similarity. In addition, in the learning process, visually different superpixels can be consider as similar as long as they belong to the same semantic object. Classical similarity measures do not have any mechanism that allows them to handle this behavior. The main drawbacks of the proposed similarity measure approach is its faillure to detect superpixels from the background and the foreground as belonging to different semantic objects, as shown by specificity results. We can relate this flaw to the fact that the classifier is trained to predict the opposite.

Table 1 Comparative superpixel similarity measure results on DS17 and ISIC18 Datasets. Superpixel similarity computation evaluation results for the Hamming, City-block, Chebyshev and the proposed ML-based similarity measure with single (RF-SDC), multi level superpixel decomposition (RF-MDC) and context descriptor (RF-MDC+CTX)

		Sensitivity ↑			Specificity ↑			MSE ↓			F-mesure ↑		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
DS17	**Hamming**	0.811	0.887	0.859	0.008	0.016	0.011	0.183	0.386	0.242	0.756	0.899	0.859
	City-block	0.004	0.115	0.037	**0.735**	0.985	0.900	0.712	0.932	0.867	0.008	0.190	0.065
	Chebyshev	0.004	0.123	0.039	0.679	**0.990**	**0.905**	0.723	0.931	0.866	0.009	0.197	0.069
	RF-SDC	0.743	0.896	0.863	0.273	0.865	0.523	0.146	0.215	0.169	0.803	0.876	0.857
	RF-MDC	0.886	0.945	0.927	0.419	0.802	0.662	0.087	0.147	0.102	0.872	0.929	0.918
	RF-MDC+CTX	**0.952**	**0.991**	**0.977**	0.560	0.898	0.783	**0.031**	**0.065**	**0.043**	**0.948**	**0.982**	**0.974**
ISIC18	**Hamming**	0.344	0.824	0.504	0.000	0.095	0.034	0.298	0.666	0.531	0.492	0.821	0.627
	City-block	*0.023*	*0.126*	*0.049*	**0.895**	**0.999**	0.976	0.765	0.939	0.894	0.045	0.215	0.089
	Chebyshev	0.028	0.142	0.059	0.891	**0.999**	**0.977**	0.749	0.932	0.885	0.054	0.239	0.106
	RF-SDC	0.783	0.874	0.852	0.139	0.484	0.217	0.155	0.271	0.186	0.795	0.871	0.853
	RF-MDC	0.858	0.925	0.907	0.122	0.468	0.241	0.105	0.200	0.132	0.859	0.921	0.905
	RF-MDC+CTX	**0.913**	**0.982**	**0.962**	0.094	0.575	0.259	**0.054**	**0.138**	**0.078**	**0.917**	**0.972**	**0.958**

The bold values indicate the best results in the tables

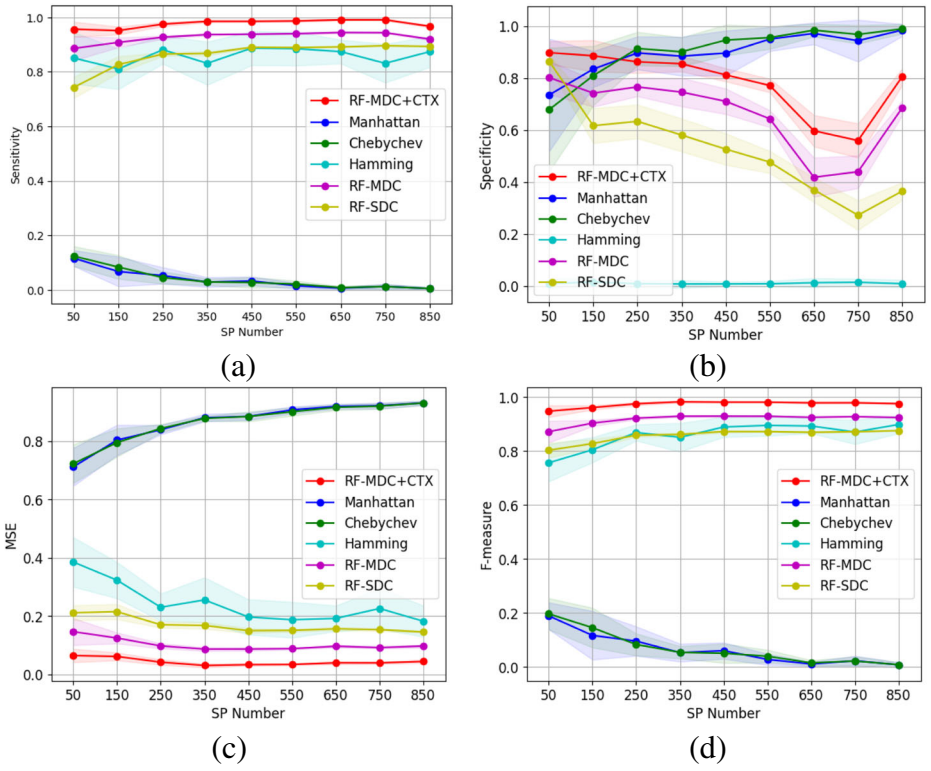

Fig. 8 Comparison of the predictive quality of the model generated between the decomposition in simple superpixels and the multi-level decomposition according to the number of superpixels on DS17 dataset. 9a: Sensitivity curve according to the number of superpixels. 9b: Specificty curve according to the number of superpixels. 9c: MSE curve according to the number of superpixels. 9d: F-mesure curve according to the number of superpixels

The Fig. 11 exposes some comparative visual results of the similarity measures. Edge color in the RAG denotes similarity value ($\in [0, 1]$) between node superpixels. The brighter the edge, the higher the value. As it can be seen in the figure, the proposed RF-MDC model produces plausible results and distinctly separates the foreground from the background as expected. The Hamming measure reveals some inconsistencies in the edge weights, especially in the vicinity of the foreground/background boundary. This behaviour leads to the bad segmentation results. Moreover, the proposed model produces good similarity values for superpixels inside the foreground object, even for those with distinct visual characteristics. This shows the main strength of the proposed similarity learning apporach compared to computed measures.

5.3 Final segmentation evaluation

In this section we discuss the quality of the final segmentation results of the proposed approach. The quality of the segmentation is evaluated by the following segmentation criteria which are calculated on the final results of the experiments.

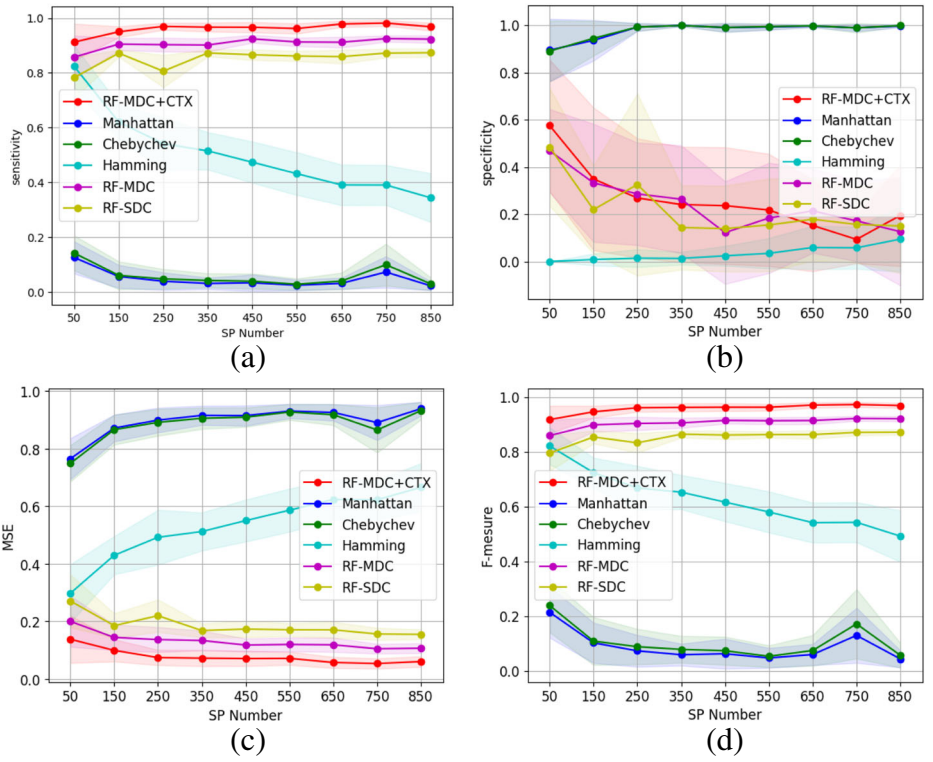

Fig. 9 Comparison of the predictive quality of the model generated between the decomposition in simple superpixels and the multi-level decomposition according to the number of superpixels on ISIC18 dataset. 10a: Sensitivity curve according to the number of superpixels. 10b: Specificty curve according to the number of superpixels. 10c: MSE curve according to the number of superpixels. 10d: F-mesure curve according to the number of superpixels

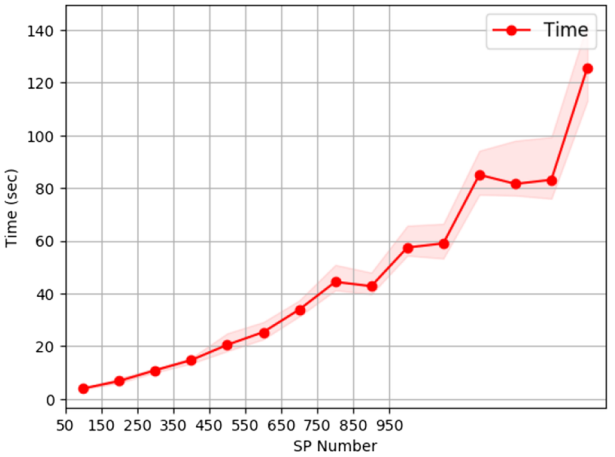

Fig. 10 Segmentation time according to superpixel number

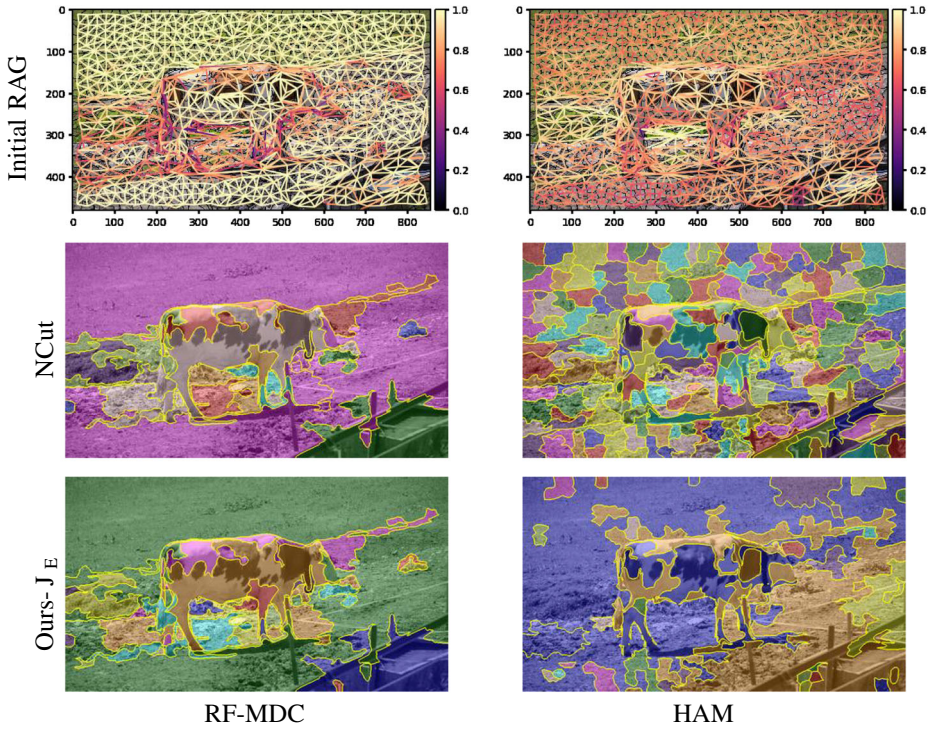

Fig. 11 Similarity measure illustration. Each column gives results from initial RAG built for one similarity measure. The results of our RF-MDC model are presented in the first column. The Hamming measure is presented in the second column. The first row gives the weighted RAG of the image. The second and the third row show respectively the segmented image using NCut and our approach. Final regions are overlaid on the source image

- Probabilistic Rand Index (**PRI**) [42]: measures the proportion of pixels that have the same labels compared to groundtruth segmentation.
- Variation of Information (**VoI**) [28]: measures the amount of randomly clustered pixels in the segmentation with no clues in the groundtruth.
- Boundary Displacement Error (**BDE**) [15]: measures the average displacement error of one boundary pixel and the closest boundary pixels in the groundtruth segmentation.
- The Global Consistency Error (**GCE**) [27] measures the extent to which one segmentation can be viewed as a refinement of the other. Segmentations which are related in this manner are considered to be consistent, since they could represent the same image which is segmented at different scales.

The higher PRI value, the better the segmentation unlike VoI, BDE and GCE whose high values denote low segmentation quality. The Fig. 12 presents the curves of PRI, VoI, BDE and GCE of the the proposed approach and some state-of-the-art graph-based segmentation approaches.

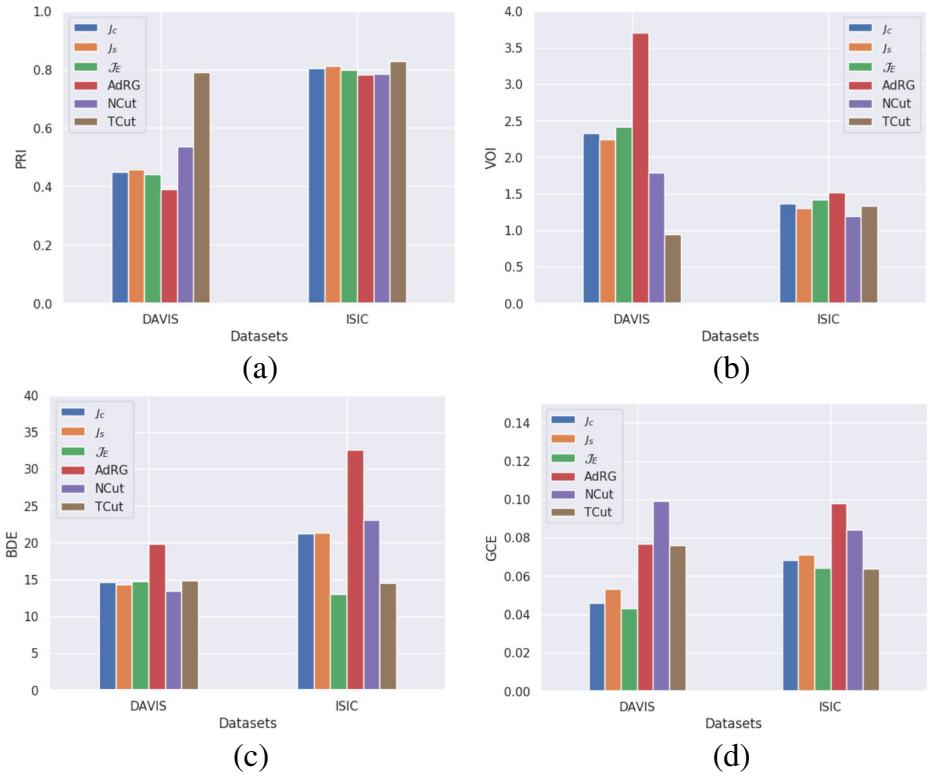

Fig. 12 Performance comparison of the best edge selection criterion for segmentation. 12a: PRI curve according to the number of superpixels. 12b: VoI curve according to the number of superpixels. 12c BDE curve according to the number of superpixels. 12d GCE curve according to the number of superpixels

From the valued graph of the image, the segmentation is performed by recursive groupings of pairs of neighboring superpixels in descending order of the weight of their connecting edge. We proposed a two-termed edge weighting coefficient in order to guide groupings to achieve coherent segmentation.

Results in the Fig. 12 show that the J_S model performs better than the J_C in the overall assessment. The compound model Ours-$\mathcal{J}_E$ in the otherhand produces best results in terms of BDE and GCE than both components J_S and J_C. We further assess the final results of the proposed segmentation approach by comparison with three graph-based techniques for image segmentation applied on the generated RAG: the Normalized Cut (NCut) [37], the Threshold Cut (TCut) and the Adaptive Region-Growing (AdRG) [10]. The NCut approach measures the cost of bipartitioning the graph as a fraction of the total edge connections to all the nodes in the graph. Segmentation is achieved by recursively computing an optimal bipartition of the graph until stability of the result. The Threshold Cut algorithm simply recursively merges nodes until there is no more edges with weight greater than a given threshold value. The Adaptive Region-Growing proceeds by recursive aggregations of neighbor regions with an adaptive merge criterion. Figures 13 and 14 expose some

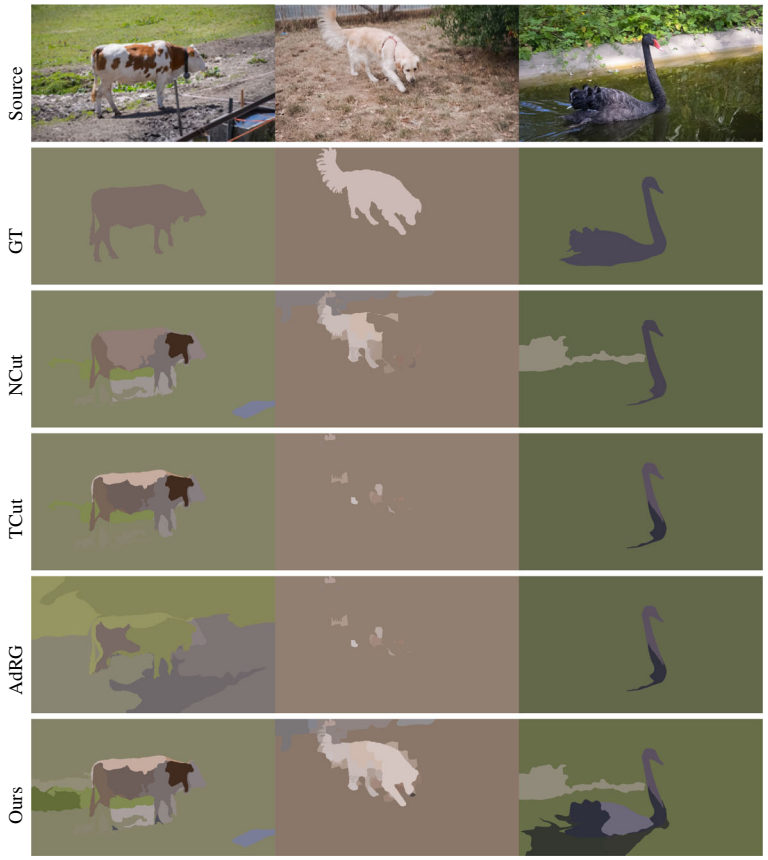

Fig. 13 Classical graph cut segmentation comparison results on DAVIS 2017. Source images are given in the first row followed by the ground-truth in the second row. The third, fourth and fifth rows give respectively the segmentation results for the NCut, TCut and the proposed method

visual final segmentation results for the NCut, TCut, AdRG and the proposed segmentation approach. For most images, our approach tends to over-segment the source image but keeps the background separated from the foreground. NCut and TCut generate under-segmented image results in comparison to the provided ground-truth. The segmentation assessment results are provided in Table 2. Although the Threshold Cut method achieves good results against most of the used criteria, the proposed technique produces the best BDE and GCE results accross the two datasets. Furthermore, our approach gives competitve results in PRI and VoI in the ISIC 2018 dataset. It is outperformed by the NCut and the TCut in DAVIS 2017 dataset on all criteria except for the GCE. This performance denotes a suboptimal global region homogeneity and can be linked to the lack of global view over the graph during the aggregations. Indeed, only the directly involved nodes are considered for nodes fusion and weights re-computation. The GCE results indeed support the main point of the proposed approach object consistency.

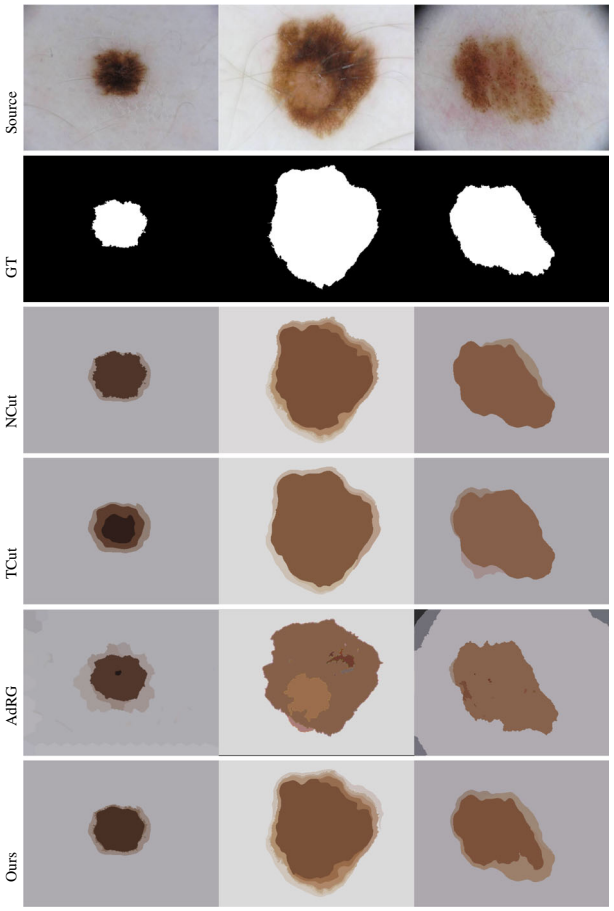

Fig. 14 Classical graph cut segmentation comparison results on ISIC 2018. Source images are given in the first row followed by the ground-truth in the second row. The third, fourth and fifth rows give respectively the segmentation results for the NCut, TCut and the proposed method

Table 2 Comparative segmentation results. Evaluation of segmentation results using Normalized Cut (NCut), Threshold Cut (TCut), Adpative Region-Growing (AdRG) and the proposed superpixel aggregations for segmentation (Ours-J_S, Ours-J_C, Ours-$\mathcal{J}_E$)

	DS17				ISIC18			
	PRI ↑	VoI ↓	BDE ↓	GCE ↓	PRI ↑	VoI ↓	BDE ↓	GCE ↓
NCut	0.535	1.793	**13.442**	0.099	0.785	**1.198**	23.088	0.084
TCut	**0.790**	**0.945**	14.805	0.076	0.830	1.339	14.498	**0.064**
AdRG	0.389	3.708	19.831	0.077	0.782	1.515	32.611	0.098
Ours-J_S	0.457	2.248	14.300	0.053	**0.812**	1.306	21.385	0.071
Ours-J_C	0.450	2.330	14.636	0.046	0.803	1.367	21.221	0.068
Ours-$\mathcal{J}_E$	0.442	2.415	14.751	**0.043**	0.799	1.420	**13.009**	**0.064**

The bold values indicate the best results in the tables

Table 3 Comparative segmentation results. Average overlap ratio (IoU in %) of segmentation masks for tracking-by-segmentation of the proposed method (Ours-$\mathcal{J}_E$) and four literature methods on three datasets

		AMCT	OGBDT	HT	SPT	PT	Ours-$\mathcal{J}_E$
Datasets	**VS09**	**83.8**	79.8	51.2	61.0	73.9	47.08
	DS17	**56.9**	44.9	33.1	27.1	26.1	53.08
	ST2	60.7	47.6	43.0	26.3	21.2	**80.52**

The bold values indicate the best results in the tables

The Table 3 shows the average Intersection over Union (IoU) scores of the proposed approach compared to PixelTrack (PT)[14], the HoughTrack (HT) [17], Superpixel Tracker (SPT) [45], the Online Gradient Boosting Decision Tree Tracker (OGBDT) [39] , and the Absorbing Markov Chains Tracking (AMCT) [47]. The IoU measures the precision of the method and is given by the intersection divided by the union of the segmented masks and the groundtruth masks, averaged over all frames. Our approach presnets the best IoU over the ST2 dataset and the second best over the DS17 dataset. However, our approach performs poorly over VS09. This result is due to the very challenging background/foreground contrast on the dataset. The proposed approach rely completely on the predicted value to track objects. A correction mechanism can be added to address this issue, especially on contour superpixel pairs.

In order to further access the proposed approach, we have computed the precision, recall and F1 metrics over the VS09 and the ST2 datasets. As shown in Table 4, the obtained results are compared against the four state-of-the-art following methods: Spatio-temporal saliency detection using objectness measure [7], Unsupervised video object segmentation using conditional random fields (uCRF) [5], Unsupervised object segmentation in video by efficient selection of highly probable positive features (SHPPF) [20] and Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions (STDE) [51]. The proposed approach outperforms the compared methods on the ST2 datasets in term of F1 and the second best precision result. On the VS09, our approach presents decent result as compared to the other methods. These results show that the presented methodology is generic because it is able to produce satisfactory results for all experiments datasets.

Table 4 Comparative segmentation results. Precision, recall and F1 of segmentation masks for tracking-by-segmentation of the proposed method (Ours-$\mathcal{J}_E$) and four literature methods on three datasets

	uCRF			STO			STDE			SHPPF			Ours-$\mathcal{J}_E$		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
VS09	0.60	0.63	0.58	**0.81**	**0.78**	**0.76**	0.66	0.65	0.51	0.51	0.56	0.53	0.49	0.70	0.53
ST2	**0.87**	0.66	0.73	0.63	0.61	0.58	0.68	0.59	0.50	0.52	**0.75**	0.61	0.84	0.59	**0.91**

The bold values indicate the best results in the tables

6 Conclusions

We proposed a superpixel merging approach for region-aggregation-based image segmentation approach. This approach employes machine learning to learn superpixel aggregation similarity instead of using explicit superpixel similarity measures. In particular, we used the Random Forests classifier to predict merging probability between any pair of neighboring superpixels. This allowed to cope with semantic gap in similarity estimation and to handle objects with heterogeneous parts. Superpixel characterization includes an effective context descriptor computed from surrounding superpixels for more consistent similarity learning. As image is represented by a Region Adjacency Graph structure weighted with merging probabilities, segmentation is achieved by a simple graph-based algorithm. The latter is based on the selection of the best merging superpixel pair at each iteration using an objective similarity function. Results obtained on four datasets indicated substantial performance improvement w.r.t. state-of-the-art approaches, both visually and quantitatively.

Unfornately, the learning approach fails when the image to be segmented presents a big variation of the observed scene. Also, the lack of global cues may reduce the overall consistency of the approach. However, balancing the number of foreground/background superpixels in training phase may allow for better similarity prediction. Furthermore, a hierarchical representation of the multi-level decomposition into superpixels can improve classifier performance and allow easier integration of the local context by hierarchical features.

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(11):2274–2282
2. Amit Y, Geman D (1997) Shape quantization and recognition with randomized trees. Neural Computation 9(7):1545–1588
3. Audebert N, Boulch A, Randrianarivo H, Le Saux B, Ferecatu M, Lefèvre S., Marlet R (2017) Deep learning for urban remote sensing. In: Urban Remote Sensing Event (JURSE), 2017 Joint, IEEE, pp 1–4
4. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(Feb):281–305
5. Bhatti AH, Rahman AU, Butt AA (2019) Unsupervised video object segmentation using conditional random fields. Image and Video Processing 13(1):9–16
6. Bosch A, Zisserman A, Munoz X (2007) Image classification using random forests and ferns. In: 2007. ICCV 2007. IEEE 11th International Conference on Computer Vision, IEEE, pp 1–8
7. Brahim K, Kalboussi R, Abdellaoui M, Douik A (2019) Spatio-temporal saliency detection using objectness measure. Signal, Image and Video Processing, pp 1–8
8. Breiman L (2001) Random forests. Mach learn 45(1):5–32
9. Breiman L (2002) Manual on setting up, using, and understanding random forests v3. 1. Statistics Department University of California Berkeley, CA, USA
10. Chaibou MS, Conze P-H, Kalti K, Solaiman B, Mahjoub MA (2017) Adaptive strategy for superpixel-based region-growing image segmentation. J Electron Imaging 26(6):061605
11. Chen C, Li S, Qin H, Pan Z, Yang G (2018) Bilevel feature learning for video saliency detection. IEEE Trans Multimedia 20(12):3324–3336
12. Chen C, Li S, Wang Y, Qin H, Hao A (2017) Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion. IEEE Trans Image Process 26(7):3156–3170
13. Conze P-H, Noblet V, Rousseau F, Heitz F, De Blasi V, Memeo R, Pessaux P (2017) Scale-adaptive supervoxel-based random forests for liver tumor segmentation in dynamic contrast-enhanced ct scans. International Journal of Computer Assisted Radiology and Surgery 12(2):223–233
14. Duffner S, Garcia C (2013) Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2480–2487

15. Freixenet J, Muñoz X, Raba D, Martí J, Cufí X (2002) Yet another survey on image segmentation: Region and boundary information integration. In: European Conference on Computer Vision, Springer, pp 408–422
16. Fukuchi K, Miyazato K, Kimura A, Takagi S, Yamato J (2009) Saliency-based video segmentation with graph cuts and sequentially updated priors. In: 2009 IEEE International Conference on Multimedia and Expo, IEEE, pp 638–641
17. Godec M, Roth PM, Bischof H (2013) Hough-based tracking of non-rigid objects. Comput Vis Image Underst 117(10):1245–1256
18. Granitto PM, Furlanello C, Biasioli F, Gasperi F (2006) Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. Chemometr Intell Lab Syst 83(2):83–90
19. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach learn 46(1-3):389–422
20. Haller E, Leordeanu M (2017) Unsupervised object segmentation in video by efficient selection of highly probable positive features. In: Proceedings of the IEEE International Conference on Computer Vision, pp 5085–5093
21. Hamming R (1950) The bell system technical journal. Bell Syst Tech J 26(2):147–160
22. Haralick RM, Shapiro LG (1985) Image segmentation techniques. Computer Vision, Graphics, and Image Processing 29(1):100–132
23. Hsu C-Y, Ding J-J (2013) Efficient image segmentation algorithm using slic superpixels and boundary-focused region merging. In: Communications and Signal Processing (ICICS) 2013 9th international conference on Information, IEEE, pp 1–5
24. Lepetit V, Fua P (2006) Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(9):1465–1479
25. Li F, Kim T, Humayun A, Tsai D, Rehg JM (2013) Video segmentation by tracking many figure-ground segments. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2192–2199
26. Louppe G, Wehenkel L, Sutera A, Geurts P (2013) Understanding variable importances in forests of randomized trees. In: Advances in Neural Information Processing Systems, pp 431–439
27. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proc. 8th Int'l Conf Computer Vision 2:416–423
28. Meilă M (2007) Comparing clusterings — an information based distance. J Multivar Anal 98(5):873–895
29. Oneata D, Revaud J, Verbeek J, Schmid C (2014) Spatio-temporal object detection proposals. In: European Conference on Computer Vision, Springer, pp 737–752
30. Ozuysal M, Fua P, Lepetit V (2007) Fast keypoint recognition in ten lines of code. In: 2007. CVPR'07. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 1–8
31. Pauly O (2012) Random forests for medical applications. PhD thesis, Technische Universität München
32. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
33. Pont-Tuset J, Perazzi F, Caelles S, Arbeláez P, Sorkine-Hornung A, Van Gool L. (2017) The 2017 davis challenge on video object segmentation. arXiv:1704.00675
34. Ren X, Malik J (2003) Learning a classification model for segmentation. In: ICCV, vol 1, pp 10–17
35. Sangsefidi N, Foruzan AH, Dolati A (2017) Balancing the data term of graph-cuts algorithm to improve segmentation of hepatic vascular structures. Computers in Biology and Medicine
36. Santana TM, Machado AM, Araújo AdA, dos Santos JA (2016) Star: a contextual description of superpixels for remote sensing image classification. In: Iberoamerican Congress on Pattern Recognition, Springer, pp 300–308
37. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Machine Intelligence 22(8):888–905
38. Silva RE (2017) An alternative approach to counting minimum (s; t)-cuts in planar graphs
39. Son J, Jung I, Park K, Han B (2015) Tracking-by-segmentation with online gradient boosting decision tree. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3056–3064
40. Stutz D, Hermans A, Leibe B (2016) Superpixels: an evaluation of the state-of-the-art. CoRR, arXiv:1612.01601
41. Tilquin F, Conze P-H, Pessaux P, Lamard M, Quellec G, Noblet V, Heitz F (2018) Robust supervoxel matching combining mid-level spectral and context-rich features. In: International Workshop on Patch-based Techniques in Medical Imaging, Springer, pp 39–47
42. Unnikrishnan R, Pantofaru C, Hebert M (2007) Toward objective evaluation of image segmentation algorithms. IEEE transactions on pattern analysis and machine intelligence, 29(6)

43. Vargas JE, Falcão AX, Dos Santos J, Esquerdo JCDM, Coutinho AC, Antunes J (2015) Contextual superpixel description for remote sensing image classification. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE 2015, pp 1132–1135
44. Vasconcelos MJM, Tavares JMR (2015) Human motion segmentation using active shape models. In: Computational and Experimental Biomedical Sciences: Methods and Applications, Springer, pp 237–246
45. Wang S, Lu H, Yang F, Yang M-H (2011) Superpixel tracking. In: 2011 International Conference on Computer Vision, IEEE, pp 1323–1330
46. Yang Y, Wang Y, Xue X (2016) A novel spectral clustering method with superpixels for image segmentation. Optik-International Journal for Light and Electron Optics 127(1):161–167
47. Yeo D, Son J, Han B, Hee Han J (2017) Superpixel-based tracking-by-segmentation using markov chains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1812–1821
48. Yin P, Criminisi A, Winn J, Essa I (2007) Tree-based classifiers for bilayer video segmentation. In: 2007. CVPR'07. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 1–8
49. Yin S, Qian Y, Gong M (2017) Unsupervised hierarchical image segmentation through fuzzy entropy maximization. Pattern Recognition
50. Yu H, Zhang X, Wang S, Hou B (2013) Context-based hierarchical unequal merging for sar image segmentation. IEEE Transactions on Geoscience and Remote Sensing 51(2):995–1009
51. Zhang D, Javed O, Shah M (2013) Video object segmentation through spatially accurate and temporally dense extraction of primary object regions (open access). Technical report University of Central Florida Orlando United States
52. Zhang Y, He K (2017) Multi-scale gaussian segmentation via graph cuts. DEStech Transactions on Computer Science and Engineering (csae)

Mahaman Sani Chaibou is a Ph.D. student in the Laboratory of Advanced Technology and Intelligent Systems (LATIS) at the University of Sousse. His research interests focus on image content analysis especially segmentation and interpretation. He is currently in charge of Scientific Production Verification and Web-Site Administration at the LATIS. He is also a part time software engineer and DBA in a local company, specialized in healthcare software.