Constrained feature selection for semisupervised color-texture image segmentation using spectral clustering

Abderezak Salmi, Kamal Hammouche, Ludovic Macaire

Constrained feature selection for semi-supervised color-texture image segmentation using spectral clustering

Abderezak Salmi [a], Kamal Hammouche[a], Ludovic Macaire[b,*]
[a]Université Mouloud Mammeri, Laboratoire Vision Artificielle et Automatique des Systèmes (LVAAS), Tizi-Ouzou, Algeria
[b]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

Abstract. Color-texture image segmentation remains a challenging problem due to extensive color-texture variability. Thus, the limited prior knowledge that is expressed by pairwise constraints can be exploited to guide the segmentation process. In this paper, we propose a new semi-supervised method by combining constrained feature selection and spectral clustering to perform color-texture image segmentation. The pairwise constraints are used by the constrained feature selection to choose the most relevant features among an available set of color and texture features. For this purpose, a new constraint score is developed to evaluate a subset of features at one time. A specific constrained spectral clustering algorithm involving the pairwise constraints is then applied to regroup the pixels into clusters. Experimental results on four benchmark datasets show that the proposed constraint score outperforms the main state-of-the-art constraint scores and that our semi-supervised segmentation method is competitive compared with supervised, semi-supervised and unsupervised state-of-the-art segmentation methods.

Keywords: Color texture segmentation, pairwise constraints, constrained feature selection, constrained spectral clustering.

*Ludovic Macaire, ludovic.macaire@univ-lille.fr

1 Introduction

Color-texture image segmentation is one of the fundamental low-level problems in computer vision.[1] Specifically, it aims to segment a color-texture image into disjointed homogeneous regions in terms of colors and textures. Although a wide variety techniques have been developed, color-texture image segmentation remains an open and challenging problem in computer vision due to the high variability of color textures. Color-texture segmentation by pixel classification is divided into two main steps. First, the color and texture features that characterize pixels must be selected in order to build a discriminating feature space. Second, a classification procedure analyzes pixel distribution in the feature space in order to assign each pixel to a class. Depending on the availability of the labeled pixels, color-texture image segmentation can be performed in a supervised, unsupervised, or semi-supervised

learning context. Deep learning approaches such as convolutional neural networks (CNNs) have shown good performance for supervised color-texture segmentation.[2] However, these algorithms require a large labeled pixel subset whose construction is time-consuming. Deep CNNs have also been recently developed for unsupervised color-texture image segmentation.[2] But, the training step of these CNNs is supervised and requires a dictionary of labeled textures. In contrast, the unsupervised segmentation methods based on clustering models such as slope difference distribution clustering, k-means, mean-shift, and spectral clustering can classify pixels without any prior knowledge about the classes.[3–5] Besides the afore mentioned methods, several techniques have been applied for unsupervised color-texture segmentation.[6–8] However, the parameters of unsupervised algorithms must be carefully tuned in order to yield satisfactory accuracy.

To overcome problems caused by supervised and unsupervised algorithms, weakly supervised or semi-supervised segmentation algorithms use limited amounts of supervised information. The prior knowledge is then expressed by a few labeled pixels or by a few links between pixels. A must-link constraint specifies that two pixels must belong to the same class, whereas a cannot-link constraint specifies that two pixels must belong to different classes. These pairwise constraints can be obtained from the available labeled pixels. Conversely, the labels of pixels cannot be deduced from the pairwise constraints. The labeled pixels or the pairwise constraints can be interactively given by a user. Semi-supervised algorithms aim to classify pixels by propagating the labels[9] or by incorporating the pairwise constraints into popular clustering algorithms such as k-means, estimation-maximization, hierarchical clustering[10] and spectral clustering.[11]

Spectral clustering, which is based on graph theory, computes the Laplacian matrix from

a similarity matrix between pixels and analyzes the distribution of pixels that are projected into a low-dimensional subspace composed of the eigenvectors of the Laplacian matrix. A trivial way to integrate pairwise constraints in spectral clustering is to enforce the similarities between pixels to 1 and 0 for must-link and cannot-link constraints, respectively.[12,13] As the number of given pairwise constraints is low, they can be propagated across the similarity matrix.[14–16] Propagation rules use Gaussian processes[17,18] or the similarities between constrained and unconstrained pixels.[19] Han et al.[20] proposed the propagation of pairwise constraints only on a randomly selected subset of pixels. The selectively propagated constraints were then used to adjust the weights of the similarity matrix. Constrained spectral clustering can be also formulated as a constrained optimization problem, where only linear equality constraints (analogous to must-link constraints) are exploited for image segmentation.[21] Another formulation of constrained spectral clustering under general convex constraints has been developed for gray level image segmentation.[22]

These constrained spectral clustering methods are based on a pixel similarity that is computed using Euclidean distance in the original feature space. Nevertheless, using all the features may bias the pairwise similarity computation and thus degrade the clustering performance. Indeed, they would provide irrelevant, redundant, and even contradictory information. To compute relevant similarity, the Mahalanobis distance is modified so that it minimizes the distance between the must-link pixels and maximizes the distance between the cannot-link pixels.[23,24] Furthermore, the constraints can be propagated to the nearest neighborhood pixels that are found in the discriminating feature sub-spaces, rather than those that are identified in the original feature space.[15] However, the random forest procedure used for this purpose, requires a time consuming training step.

In this paper, we propose an original approach for color-texture segmentation based on constrained spectral clustering. Specifically, it consists of propagating the pairwise constraints in order to select the most relevant features. Rather than computing a score for each color-texture feature, we propose a new constraint score that evaluates the relevance of a subset of color-texture features. Thus, the similarity matrix that is computed in this selected subspace is truer to the data structure and respects as much as possible the given must-link and cannot-link constraints.

We develop an efficient color-texture image segmentation that combines constrained feature selection and spectral clustering (CFS-SC). Its main contributions are as follows:

- Pairwise must-link and cannot-link constraints are involved in feature selection to improve the color-texture image segmentation performance using spectral clustering.

- A new similarity-based constraint score is proposed to select the optimal number of relevant features.

The rest of this paper is organized as follows. Section 2 introduces the related work, including a brief overview of the constrained spectral clustering and primary state-of-the-art semi-supervised constraint scores. Our proposed color-texture image segmentation is detailed in Sec. 3. Section 4 provides experimental segmentation results achieved with four benchmark datasets. Finally, our conclusions are given in Sec. 5.

2 Related work

In this section, we first introduce spectral clustering and detail how to include the constraints for spectral clustering and feature selection.

2.1 Spectral clustering

Let X be the set of n sample pixels represented by the matrix $\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n] \in \mathbb{R}^{n \times d}$, where $\mathrm{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ir}, \ldots, x_{id}]^T \in \mathbb{R}^d$ represents the i-th pixel of X, and x_{ir},$(r = 1, \ldots, d)$, the r-th feature value of the i-th pixel. Let $F_d = \{f_1, f_2, \ldots, f_r, \ldots, f_d\}$ be the set of d features that is represented by $\mathrm{X}^T = [\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{f}_r, \ldots, \mathrm{f}_d]$, where $\mathrm{f}_r = [x_{1r}, x_{2r}, \ldots, x_{nr}] \in \mathbb{R}^n$ is the r-th feature vector.

Spectral clustering represents dataset X by an un-directed weighted similarity graph $G = (V, E, \mathrm{W})$ in which each sample pixel corresponds to a node. V is a non-empty set that contains all nodes, and E is the set of edges between any two nodes in V. Each edge in E is weighted by a similarity value w_{ij} $(i, j = 1, 2, \ldots, n)$ between two nodes. The similarity matrix W that gathers similarities between all pairs of nodes is positive semi-definite and symmetric. Generally, the similarity w_{ij} between two pixels x_i and x_j is computed by the following Gaussian kernel function[11]

$$w_{ij} = \exp\left(-\frac{\delta^2(\mathrm{x}_i, \mathrm{x}_j)}{2\sigma^2}\right) \quad i, j = 1, 2, \ldots, n \tag{1}$$

where σ is a scaling parameter and $\delta(\mathrm{x}_i, \mathrm{x}_j)$ is the Euclidean distance between the two pixels x_i and x_j.

Spectral clustering is an unsupervised classification method that computes the eigen spectrum of the Laplacian matrix, which is deduced from the similarity matrix W, to separate the dataset into clusters. The most commonly used un-normalized Laplacian matrix is $\mathrm{L} = \mathrm{D} - \mathrm{W}$ while the most used normalized Laplacian matrices are $\mathrm{L}_{Sym} = \mathrm{D}^{-1/2}\mathrm{L}\mathrm{D}^{-1/2}$ and $\mathrm{L}_{Asym} = \mathrm{D}^{-1}\mathrm{L}$.[5,11] $\mathrm{D} \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix whose elements are $d_{ii} =$

$\sum_{j=1}^{n} w_{ij}$. To identify k clusters, the data are mapped onto a low-dimensional space that is based on the k-dominant eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix. Finally, an unsupervised clustering algorithm such as k-means identifies the k clusters of so mapped data.[5,11]

2.2 Constraint spectral clustering

In the semi-supervised learning context, the prior knowledge is expressed by a small number of labeled pixels or pairwise constraints. In this paper, we consider that few labeled pixels (prototypes) characterize the k classes ω^l, $l = 1, \ldots, k$ of the input image I. These prototypes can be interactively selected by the user or randomly selected from the ground truth. Let X^l ($X^l \subset X$) be the subset of p prototypes associated with the class ω^l. From the set of prototypes denoted as X^P $\left(X^P = \bigcup_{l=1,\ldots,k} X^l\right)$, we can build the set M of $(k \cdot p \cdot (p-1))$ must-link pairs that are composed of two prototypes belonging to the same class:

$$M = \left\{(x_i, x_j) \in X^2 \mid \exists\ l = 1, \ldots, k \text{ so that } x_i \in X^l \text{ and } x_j \in X^l\right\}. \tag{2}$$

We can also build the set C of $(k \cdot (k-1) \cdot p^2)$ cannot-link pairs that are composed of two prototypes belonging to different classes:

$$C = \left\{(x_i, x_j) \in X^2 \mid \exists\ (l, m);\ l \neq m;\ \text{so that } x_i \in X^l \text{ and } x_j \in X^m\right\}. \tag{3}$$

Of all possible pixel pairs that can be extracted from X, those belonging to M or C are called constrained pairs while the remaining pairs are called unconstrained pairs. Furthermore, the pixels that are not prototypes are called unlabeled sample pixels, and are gathered

in subset $X^U = X/X^P$ (with $|X^U| = n - (k \cdot p)$).

A straightforward way to integrate must-link and cannot-link pairs in spectral clustering is computing the similarity matrix $\mathrm{W}^{M,C}$.[10,12] For a given must-link pair $(x_i, x_j) \in M$, $w_{ij}^{M,C}$ is set to 1; for a given cannot-link pair $(x_i, x_j) \in C$, $w_{ij}^{M,C}$ is set to 0. Thus, the modified matrix $\mathrm{W}^{M,C}$ is defined as follows:

$$w_{ij}^{M,C} = \begin{cases} 1 & \text{if} \quad (x_i, x_j) \in M \\ 0 & \text{if} \quad (x_i, x_j) \in C \\ \exp\left(-\frac{\delta^2(\mathrm{x}_i, \mathrm{x}_j)}{2\sigma^2}\right) & \text{otherwise.} \end{cases} \qquad i, j = 1, 2, \ldots, n \tag{4}$$

The next steps of constrained spectral clustering are not specific and are summarized in Algorithm 1.

Algorithm 1 Constrained spectral clustering algorithm.

Input: dataset X represented by $\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n] \in \mathbb{R}^{n \times d}$, number k of desired clusters, set M of must-link constraints, and set C of cannot-link constraints.

1. Compute the similarity matrix $\mathrm{W}^{M,C}$ defined by Eq. (4).
2. Compute the Laplacian matrix $\mathrm{L}_{Sym}^{M,C}$ associated to $\mathrm{W}^{M,C}$.
3. Compute the spectrum of $\mathrm{L}_{Sym}^{M,C}$, and extract k dominant eigenvectors $\mathrm{U}^{M,C} = [\mathrm{u}_1, \ldots, \mathrm{u}_k] \in \mathbb{R}^{n \times k}$.
4. Construct the matrix $\mathrm{T}^{M,C} \in \mathbb{R}^{n \times k}$ from $\mathrm{U}^{M,C}$ by normalizing each row of $\mathrm{U}^{M,C}$ to have unit length
$$t_{ij} = \frac{u_{ij}}{\sqrt{\sum_{j=1}^{k} u_{ij}^2}}$$
5. Consider each row of $\mathrm{T}^{M,C}$ as a point, and cluster all points into k clusters by k-means algorithm.

Output: k clusters of dataset X.

The performance achieved by constrained spectral clustering strongly depends on the pairwise similarities between the unconstrained pixels, which are based on the Euclidean

distance δ in the original d-dimensional feature space (see Eq. (4)). Nevertheless, using all of these features may bias the pairwise similarities. To deal with this problem, we propose to select the relevant subset of features among those available. In the semi-supervised context, many feature selection methods have been proposed to make full use of prior knowledge.[25] In the next section, we focus on the constraint scores that are used by semi-supervised feature selection.

2.3 State-of-the-art about semi-supervised constraint scores

Feature selection based on semi-supervised constraint scores involves the analysis of both pairwise constraints and unlabeled sample pixels. Specifically, it considers both the discriminating power of the pairwise constraints and the local properties of the unlabeled sample pixels. In the context of spectral graph theory, the dataset X can be represented by the nearest neighbor graph G^{KNN} whose node v_i is connected to v_j when x_j is one of the K-nearest neighbors ($KNNs$) of x_i, the distance being computed in the original d-dimensional feature space. The similarity matrix $\mathrm{W}^{\mathrm{KNN}} \in \mathbb{R}^{n \times n}$ is defined as follows:

$$w_{ij}^{\mathrm{KNN}} = \begin{cases} w_{ij} & \text{if } x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The two graphs G^M and G^C can also be built from the sets of must-link constraints M and cannot-link constraints C. The corresponding similarity matrices $\mathrm{W}^M \in \mathbb{R}^{n \times n}$ and

$\mathrm{W}^C \in \mathbb{R}^{n\times n}$ are defined as follows:

$$w_{ij}^{M} = \begin{cases} 1 & \text{if } (x_i, x_j) \in M \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$w_{ij}^{C} = \begin{cases} 1 & \text{if } (x_i, x_j) \in C \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Zhao et al.[26] introduced the semi-supervised constraint score C_r^1, which combines the similarity matrix W^C constructed from the cannot-link constraints (Eq. (7)) and the similarity matrix $\mathrm{W}^{\mathrm{KNN1}} \in \mathbb{R}^{n\times n}$, which is built from the set of must-link constraints and unlabeled sample pixels as follows:

$$w_{ij}^{\mathrm{KNN1}} = \begin{cases} \gamma & \text{if } (x_i, x_j) \in M \\ 1 & \text{if } (x_i \in X^U \text{ or } x_j \in X^U) \text{ and } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where γ is a constant parameter. In our experiments, γ is set to 100 and K is set to 5 as in Ref. 26.

The constraint score C_r^1 is defined as

$$C_r^1 = \frac{\mathrm{f}_r^T \mathrm{L}^{\mathrm{KNN1}} \mathrm{f}_r}{\mathrm{f}_r^T \mathrm{L}^{C} \mathrm{f}_r} \tag{9}$$

where $\mathrm{L}^{\mathrm{KNN1}} = \mathrm{D}^{\mathrm{KNN1}} - \mathrm{W}^{\mathrm{KNN1}}$ is the un-normalized Laplacian matrix of $\mathrm{W}^{\mathrm{KNN1}}$, and $\mathrm{D}^{\mathrm{KNN1}}$ is the degree matrix computed from $\mathrm{W}^{\mathrm{KNN1}}$.

Kalakech et al.[27] proposed the semi-supervised constraint score C_r^2 that is less sensitive to the constraint sets

$$C_r^2 = \frac{\tilde{\mathrm{f}}_r^T \mathrm{L} \tilde{\mathrm{f}}_r}{\tilde{\mathrm{f}}_r^T \mathrm{D} \tilde{\mathrm{f}}_r} \cdot \frac{\mathrm{f}_r^T \mathrm{L}^M \mathrm{f}_r}{\mathrm{f}_r^T \mathrm{L}^C \mathrm{f}_r} \tag{10}$$

where $\bar{\mathrm{f}}_r = \frac{\sum_{i=1}^n x_{ir} d_{ii}}{\sum_{i=1}^n d_{ii}} = \frac{\mathrm{f}_r^T \mathrm{D1}}{1^T \mathrm{D1}}$, $\tilde{\mathrm{f}}_r = \mathrm{f}_r - \bar{\mathrm{f}}_r$, and $1 = [1, \ldots, 1]^T$. L and D are deduced from the similarity matrix W (Eq. (1)). $\mathrm{L}^M = \mathrm{D}^M - \mathrm{W}^M$ and $\mathrm{D}^M = \sum_{j=1}^n w_{ij}^M$.

Benabdeslem and Hindawi proposed another constraint score C_r^3, which combines the similarity matrices W^C and $\mathrm{W}^{\mathrm{KNN2}}$.[28] The similarity matrix $\mathrm{W}^{\mathrm{KNN2}}$ is expressed as

$$w_{ij}^{\mathrm{KNN2}} = \begin{cases} w_{ij} & \text{if } \big((x_i, x_j) \in M\big) \text{ or } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The score C_r^3 is defined as follows:[28]

$$C_r^3 = \frac{\mathrm{f}_r^T \mathrm{L}^{\mathrm{KNN2}} \mathrm{f}_r}{\mathrm{f}_r^T \mathrm{L}^C \mathrm{D}^{\mathrm{KNN2}} \mathrm{f}_r} \tag{12}$$

A second semi-supervised constrained Laplacian score, referred to C_r^4, has been proposed by Benabdeslem and Hindawi in Ref. 29 as follows:

$$C_r^4 = \frac{\mathrm{f}_r^T \mathrm{L}^{\mathrm{KNN3}} \mathrm{f}_r}{\mathrm{f}_r^T \mathrm{L}^C \mathrm{D}^{\mathrm{KNN}} \mathrm{f}_r} \tag{13}$$

where the diagonal matrix $\mathrm{D}^{\mathrm{KNN}}$ is deduced from the similarity matrix $\mathrm{W}^{\mathrm{KNN}}$ (see Eq. (5)). The Laplacian matrix $\mathrm{L}^{\mathrm{KNN3}}$ is computed from the similarity matrix $\mathrm{W}^{\mathrm{KNN3}}$, which is defined

as follows:

$$w_{ij}^{\mathrm{KNN3}} = \begin{cases} w_{ij}^2 + w_{ij} & \text{if } \big((x_i, x_j) \in C\big) \text{ and } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ w_{ij}^2 & \text{if } \big((x_i, x_j) \in M\big) \text{ and } \big(x_i \notin \mathrm{KNN}(x_j) \text{ and } x_j \notin \mathrm{KNN}(x_i)\big) \\ w_{ij} & \text{if } \big(x_i \in X^U \text{ or } x_j \in X^U\big) \text{ and } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Recently, Yang et al.[30,31] introduced the new semi-supervised constraint score C_r^5, which takes advantage of the local geometrical structure of unlabeled data samples as well as the constraint information deduced from prototypes as follows:

$$C_r^5 = \frac{2(\mathrm{f}_r)^T \mathrm{L}^{\mathrm{KNN4}} \mathrm{f}_r}{(\tilde{\mathrm{f}}_r)^T \mathrm{D}^{\mathrm{KNN4}} \tilde{\mathrm{f}}_r + 2(\tilde{\mathrm{f}}_r^P)^T \mathrm{L}^P \tilde{\mathrm{f}}_r^P - (\tilde{\mathrm{f}}_r^P)^T \mathrm{D}^P \tilde{\mathrm{f}}_r^P} \tag{15}$$

The similarity matrix $\mathrm{W}^{\mathrm{KNN4}}$ is expressed as

$$w_{ij}^{\mathrm{KNN4}} = \begin{cases} 1 & \text{if } \big((x_i, x_j) \in M\big) \text{ and } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ \lambda & \text{if } \big((x_i, x_j) \in M\big) \text{ and } \big(x_i \notin \mathrm{KNN}(x_j) \text{ and } x_j \notin \mathrm{KNN}(x_i)\big) \\ (1-\gamma) w_{ij} & \text{if } \big((x_i, x_j) \in C\big) \text{ and } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ w_{ij} & \text{if } \big(x_i \in X^U \text{ or } x_j \in X^U\big) \text{ and } \big(x_i \in \mathrm{KNN}(x_j) \text{ or } x_j \in \mathrm{KNN}(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where γ and λ are the parameters set to the empirical values of 0.9 and 0.5, respectively.[30,31] The cells of W^P are set to $1/|X^l|$ when two data samples are prototypes that belong to the

same class and to 0, otherwise

$$w_{ij}^{P} = \begin{cases} 1/|X^l| & \text{if } x_i \in X^l \text{ and } x_j \in X^l \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Note that $\tilde{\mathrm{f}}_r^P = \mathrm{f}_r^P - \bar{\mathrm{f}}_r^P$. $\mathrm{L}^P = \mathrm{D}^P - \mathrm{W}^P$ is the un-normalized Laplacian matrix of W^P, and D^P is the degree matrix computed from W^P.

Because feature scores are based on the similarity matrices that are computed in the original d-dimensional feature space, they may be corrupted by irrelevant features. Furthermore, as features are ranked according to their individual scores, the selected features can be correlated and may lead to low pixel classification performance.

3 Proposed color-texture image segmentation

This section details our CFS-SC method for semi-supervised color-texture image segmentation. The prior knowledge provided by a user is expressed by a small number of labeled pixels that represent the prototypes of each color-texture class. The binary similarities between the prototypes are integrated into a matrix that quantifies the pairwise similarities of the pixels, and pixel spectral clustering that analyzes similarity matrix is used to perform semi-supervised segmentation.

Because some color and texture features may be irrelevant or redundant, the computation of pairwise similarities, which is based on the Euclidean distance in the full feature space, is biased. Thus, the segmentation performance from spectral pixel clustering can be degraded. To circumvent this problem, a new constrained feature selection is included in the color-

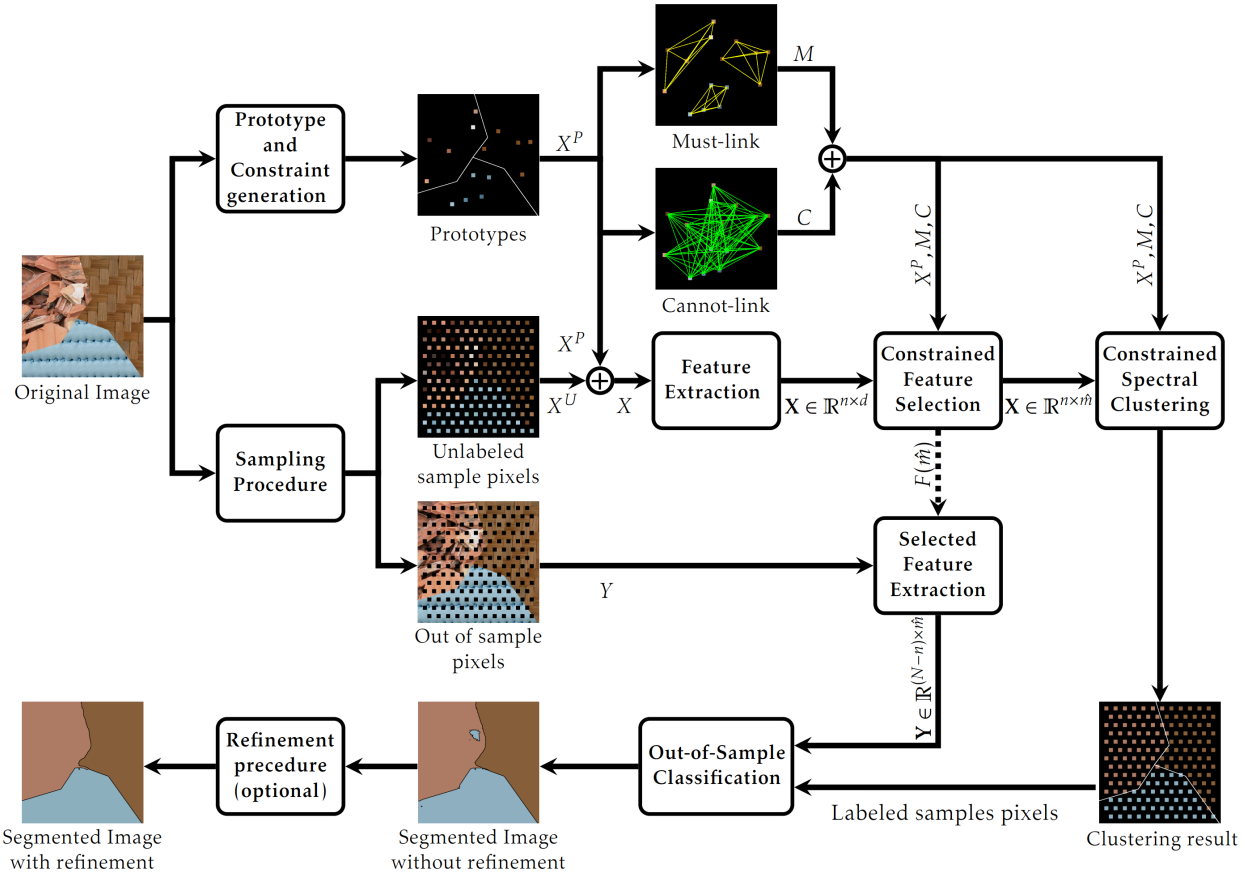


Fig 1 Flowchart of the proposed color-texture image segmentation (CFS-SC).

texture image segmentation that is outlined in Fig. 1. The proposed constrained feature selection and spectral clustering is first applied to a limited number of sampled pixels.

3.1 Pixel sampling procedure

Spectral clustering is based on Laplacian matrix eigen-decomposition, whose computational complexity is $O(n^3)$, where n is the number of pixels that are sampled from the color image I, whose size is N pixels. From I, we build the subset $X = \{x_1, x_2, \ldots, x_n\}$ of n sample pixels. This set includes the subset X^P of the $(k \cdot p)$ prototypes and the subset X^U of the $(n - (k \cdot p))$ unlabeled sample pixels that are picked up thanks to a regular hexagonal grid. Once the subset X has been classified by constrained spectral clustering, a specific

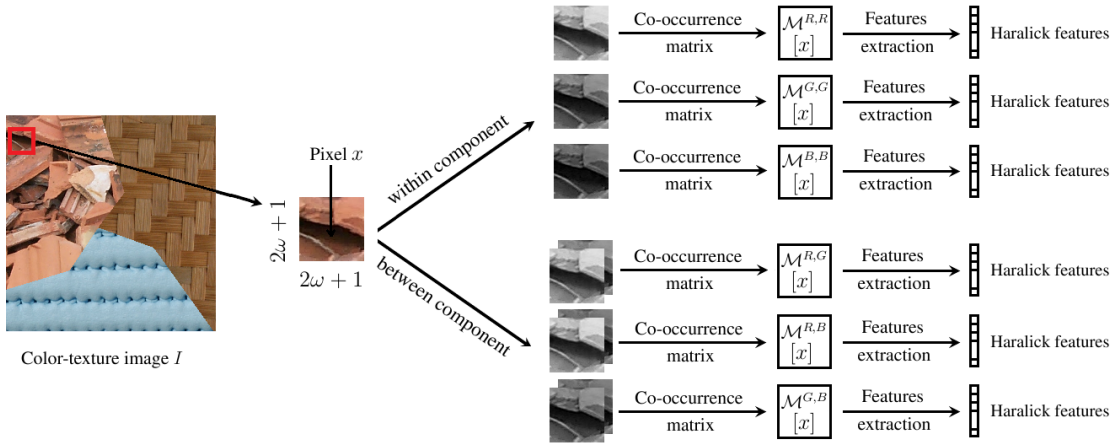


Fig 2 Illustration of local Haralick texture feature computing.

classification step assigns the remaining $(N-n)$ out-of-sample pixels that are represented by the subset $Y = \{y_1, y_2, \ldots, y_{(N-n)}\}$.

3.2 Color and texture features

To segment the color-texture image, the pixels are characterized by both Haralick features (spatial features)[32] and Gabor features (spectral features).[33,34] Haralick texture features describe correlation between neighboring pixels and are computed from co-occurrence matrices. In a color image I, each pixel x is represented by three color components R, G, and B. Six co-occurrence matrices (three between-component matrices $\mathcal{M}^{R,R}[x]$, $\mathcal{M}^{G,G}[x]$, and $\mathcal{M}^{B,B}[x]$, and three within-component matrices $\mathcal{M}^{R,G}[x]$, $\mathcal{M}^{R,B}[x]$, and $\mathcal{M}^{G,B}[x]$) are locally computed for each pixel x by considering all co-occurrences in a window with a size of $(2\omega+1) \times (2\omega+1)$ centered on pixel x (see Fig. 2) (ω is set to 11 for our experiments). Overall, 84 Haralick features are extracted from six co-occurrence matrices.[32]

Gabor features are powerful texture descriptors generated from the convolution of Gabor filters. The impulse response of a Gabor filter with specific parameters (scale γ and

orientation θ) depends on spatial coordinates (u, v) and is expressed as

$$g(u, v; \gamma, \theta) = \exp^{-\frac{1}{2}\left(\frac{u'^2}{\sigma_u{}^2}+\frac{v'^2}{\sigma_v{}^2}\right)} \exp^{j2\pi\gamma u'} \tag{18}$$

with $u' = u\cos\theta + v\sin\theta$ and $v' = -u\sin\theta + v\sin\theta$. γ is the central frequency of the filter and σ_u and σ_v are the space constants of the Gaussian envelope along the horizontal and vertical axes, respectively. A bank of Gabor filters, tuned with several scales (frequencies) and orientations, are designed to characterize different textures at different scales and orientations in the input image.[33] Seven scales ($\gamma^\star = 2\sqrt{2}$, $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$, $64\sqrt{2}$, and $128\sqrt{2}$), and four orientations ($\theta^\star = 0°$, °45°, 90°, and 135°) are considered as in Ref. 33 for each one of three color components R, G, and B (with $^\star = R, G$, or B). As magnitude response of each filter corresponds to a feature, 84 Gabor features are computed at each pixel.

Besides texture features, color features are represented by color components. Two device-depend color spaces, namely (R, G, B) and (H, S, V), and two device-independent color spaces, namely (X, Y, Z) and (L^*, a^*, b^*) are considered in this paper. The conversion from standard (R, G, B) to (H, S, V), (X, Y, Z) and (L^*, a^*, b^*) color spaces is performed using CIE standard illuminant D65 as reference white.[35]

Finally each pixel is characterized by a set of $d = 180$ features (84 Haralick texture features, 84 Gabor texture features, and 12 color features) $F_m = \{f_1, \ldots, f_d\}$. Each of the $d = 180$ pixel features is normalized between 0 and 1 to ensure that the scales of all features are equal.

3.3 Proposed constrained feature selection

Among extracted features, some of them are irrelevant for the faithful computation of pairwise similarities. Therefore, analyzing all of these features could degrade the clustering performance. To avoid this problem, we propose applying a constrained feature selection using constraint feature scores to select the most relevant features among the d available features.[25] As mentioned in Section 2, the state-of-the-art feature scores based on the similarity matrices evaluate the features one by one and ignore any correlation between features. Thus, clustering algorithms that operate in a subspace of individually relevant features do not necessarily provide favorable results.[29] In this section, we propose a new constraint score that estimates the relevance of a subset of features, making it possible to identify an optimal number of relevant features.

3.3.1 Proposed constraint score

The proposed constraint score, denoted $\varepsilon^{SS}(F_m)$, uses the similarity matrices to evaluate the relevance of a subset of m features $F_m = \{f_1, \ldots, f_m\}$ $(m = 1, 2, \ldots, d)$ and exploits both the pairwise constraint sets and the distribution of unlabeled sample pixels in the F_m feature space.

The relevance of F_m is evaluated by means of the distance between a target similarity matrix $\hat{\mathrm{W}}^{SS}$, which is defined from the given constraints, and the similarity matrix $\mathrm{W}(F_m)$ computed with the subset F_m of features. The score $\varepsilon^{SS}(F_m)$, which should be as low as possible, is expressed as follows:

$$\varepsilon^{SS}(F_m) = \|\mathrm{W}(F_m) - \hat{\mathrm{W}}^{SS}\|_2 \tag{19}$$

where $\|\cdot\|_2$ is the Euclidean norm. Thus, $\varepsilon^{SS}(F_m)$ can be rewritten as

$$\varepsilon^{SS}(F_m) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_{ij}(F_m) - \hat{w}_{ij}^{SS}\right)^2 \tag{20}$$

$\mathrm{W}(F_m) \in \mathbb{R}^{n\times n}$ is the similarity matrix computed on the sample pixel set X that is characterized by the subset of features F_m

$$w_{ij}(F_m) = \exp\left(-\frac{\delta^2\left(\mathrm{x}_i^{(m)}, \mathrm{x}_j^{(m)}\right)}{2\sigma^2}\right) \qquad i,j = 1,2,\ldots,n \tag{21}$$

where $\delta(\mathrm{x}_i^{(m)}, \mathrm{x}_j^{(m)})$ is the Euclidean distance between two sample pixels x_i and x_j characterized by the subset of m features F_m.

$\hat{\mathrm{W}}^{SS} \in \mathbb{R}^{n\times n}$ is the target matrix whose cells correspond to must-link pairs and are set to 1, while cells corresponding to the cannot-link pairs are set to 0. $\hat{\mathrm{W}}^{SS}$ is then a binary matrix defined as follows:

$$\hat{w}_{ij}^{SS} = \begin{cases} 1 & \text{if } (x_i, x_j) \in M^{SS} \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

The cell $\hat{w}_{ij}^{SS}$ is set to 1 when (x_i, x_j) belongs to M^{SS} (0 otherwise), which is a new set of must-link pairs that results from the propagation of prototype subsets X^l, $l = 1,\ldots,k$ to unlabeled sample pixels

$$M^{SS} = \left\{(x_i, x_j) \in X^2 \mid \exists\, l = 1,\ldots,k \text{ so that } NP(x_i) \in X^l \text{ and } NP(x_j) \in X^l\right\} \tag{23}$$

The nearest prototype $NP(x_i)$ is the prototype whose distance in the original d-dimensional feature space from a sample pixel $x_i \in X$ is the smallest

$$NP(x_i) = \underset{z \in \bigcup_{l=1,\ldots,k} X^l}{\arg\min} \Big(\delta^2(\mathrm{x}_i, \mathrm{z})\Big). \tag{24}$$

Thus, the pair (x_i, x_j) belongs to M^{SS} when the nearest prototypes of x_i and x_j both belong to the subset X^l of prototypes that is associated to the class ω^l. Because $NP(x_i)$ is x_i when x_i belongs to prototype subset X^l, set M is included in M^{SS} ($M \subset M^{SS}$). Thus, M^{SS} increases the contribution of the pairwise constraints M for semi-supervised feature selection. Finally, $\varepsilon^{SS}(F_m)$ makes it possible to assess the ability of the feature subset to preserve the pairwise constraint sets M and C, and the extended pairwise constraints provided by M^{SS}. These extended pairwise constraints should represent the geometric structure of the classes well.

3.3.2 Feature selection procedure

Because the score ε^{SS} evaluates the relevance of a subset of features simultaneously, we use a simple sequential forward feature selection technique.[36] To evaluate the relative relevance of d features, we first consider each feature one by one ($m = 1$). The feature f_r,which minimizes $\varepsilon^{SS}(F_1)$ with $F_1 = \{f_r\}$, is selected and combined with each of the remaining $d-1$ features to form $(d-1)$ feature subsets, denoted as F_2. The corresponding $d-1$ scores $\varepsilon^{SS}(F_2)$ are then computed, and the pair of features that minimizes $\varepsilon^{SS}(F_2)$ is retained. When m features among the d features have been selected, the $(m+1)$-th feature that minimizes $\varepsilon^{SS}(F_{m+1})$ when combined with the m previously chosen features, is selected. This sub-

optimal procedure is iterated until d features have been ordered. Finally, the subset $F_{\hat{m}}$ that corresponds to the minimum of $\varepsilon^{SS}(F_m)$ is selected (see Algorithm 2).

Algorithm 2 Feature selection procedure.

Input: Set of d feature $F_d = \{f_1, \ldots, f_r, \ldots, f_d\}$.

1. Create empty set of features $F_0 = \{\varnothing\}$.
2. For $m = 1$ to d
   a. Select the most relevant feature f_r^+
   $$f_r^+ = \underset{f_r \in F_d \setminus F_{m-1}}{\arg\min} \left(\varepsilon^{SS}(F_{m-1} \cup \{f_r\}) \right).$$
   b. Update $F_m = F_{m-1} \cup \{f_r^+\}$.
3. Select the number $\hat{m}$ of features such that
   $$\hat{m} = \underset{m=1,2,\ldots,d}{\arg\min} \left(\varepsilon^{SS}(F_m) \right).$$

Output: Subset of $\hat{m}$ relevant features $F_{\hat{m}}$.

3.4 Constrained spectral clustering

Once the subset $F_{\hat{m}}$ of relevant features has been selected, the set of n sample pixels X is classified by the constrained spectral clustering algorithm (see Algorithm 1). For this purpose, we propose to replace the k-means algorithm with the nearest neighbor algorithm, which uses the available k prototype subsets $X^l, l = 1, \ldots, k$. Moreover, the matrix $\mathrm{W}^{M,C}$ of Eq. (4) is computed in the $\hat{m}$-dimensional feature space and becomes

$$w_{ij}^{M,C}(F_{\hat{m}}) = \begin{cases} 1 & \text{if } \left(x_i, x_j\right) \in M \\ 0 & \text{if } \left(x_i, x_j\right) \in C \\ \exp\left(-\frac{\delta^2(\mathrm{x}_i^{(\hat{m})}, \mathrm{x}_j^{(\hat{m})})}{2\sigma^2} \right) & \text{otherwise.} \end{cases} \qquad i, j = 1, 2, \ldots, n \tag{25}$$

3.5 Out-of-sample classification

The last step of the proposed image segmentation is to assign the subset Y of $(N-n)$ out-of-sample pixels to clusters that have been previously determined by the constrained spectral

clustering. Generally, this procedure is performed by assigning each out-of-sample pixel y_i to the class of its nearest sample pixels $x_i \in X$ in the $\hat{m}$-dimensional feature space.[23] In this paper, we propose to classify the out-of-sample pixels in this low-dimensional space, where the k clusters of sample pixels have been determined.

For this purpose, we propose to embed out-of-sample pixels y_i onto the k-dimensional subspace that is formed by the k-dominant eigenvectors u_l, which correspond to the k smallest eigenvalues λ_l of the Laplacian matrix $\mathrm{L}_{Sym}^{M,C}$ computed on the set X of sample pixels.[37] The projection of out-of-sample pixel y_i on the l^{th} k-dominant eigenvector is defined as follows:

$$z_l(y_i) = \frac{1}{(1-\lambda_l)} \sum_{j=1}^{n} u_{lj} \tilde{w}^{(\hat{m})}(y_i, x_j) \qquad l = 1, \ldots, k \tag{26}$$

where $\tilde{w}^{(\hat{m})}(y_i, x_j)$ is the equivalent normalized similarity between the out-of-sample y_i and the sample pixel x_j.

$$\tilde{w}^{(\hat{m})}(y_i, x_j) = \frac{1}{n} \frac{w^{(\hat{m})}(y_i, x_j)}{\sqrt{E\big[w^{(\hat{m})}(y_i, x)\big] E\big[w^{(\hat{m})}(x_j, x')\big]}} \tag{27}$$

$E[\cdot]$ represents the average operator such that $E\big[w^{(\hat{m})}(y_i, x)\big] = \frac{1}{n} \sum_j^n w^{(\hat{m})}(y_i, x_j)$. The similarity $w^{(\hat{m})}(y_i, x_j)$ between two pixels y_i and x_j is defined as follows:

$$w^{(\hat{m})}(y_i, x_j) = \exp\left(- \frac{\delta^2\big(\mathrm{y}_i^{(\hat{m})}, \mathrm{x}_j^{(\hat{m})}\big)}{2\sigma^2} \right) \tag{28}$$

The embedded out-of-sample pixel y_i onto the k-dimensional subspace is normalized to have

the unit length

$$\tilde{z}_l(y_i) = \frac{z_l(y_i)}{\sqrt{\sum_{j=1}^{k} z_j(y_i)^2}} \qquad l = 1, \ldots, k. \tag{29}$$

Following this, each out-of-sample pixel y_i is assigned to the cluster of the nearest sample pixel of $\tilde{z}(y_i)$ in the k-dimensional subspace.

As in Ref. 2, a refinement procedure can be applied to further improve the segmentation accuracy. This procedure consists of keeping all regions larger than 0.5% of N while all other regions are relabeled with the class index corresponding to the largest adjacent region. Finally, all steps of the proposed CFS-SC method for color-texture image segmentation are summarized in Algorithm 3.

4 Experimental results

In this section, we evaluate our CFS-SC method for color-texture image segmentation. We first evaluate the performance of the proposed semi-supervised feature score ε^{SS}. For this purpose, we consider two color-texture images from the Prague texture segmentation benchmark.[38] Next, we present segmentation results obtained by the proposed algorithm and we compare them with those obtained by several state-of-the-art algorithms on four benchmark datasets of color and grayscale textures.

4.1 Evaluation of the semi-supervised constrained feature selection

To evaluate the performance of the proposed constraint score ε^{SS}, we compare it with the state-of-the-art constraint scores C^1, C^2, C^3, C^4, and C^5 (see section 2.3). The scaling parameter σ used to compute similarity matrices is set to 1 for all experiments. Feature

Algorithm 3 Color-texture image segmentation by CFS-SC method.

Input: - Image parameters: Image I, number k of pixel classes;
- Feature extraction parameters: Haralick parameter (ω), Gabor parameters (θ, γ);
- Constrained feature selection and spectral clustering parameters : number n of sample pixels, number p of prototypes by class.

3.1 Sampling and pairwise constraint set generation

1. Extract n sample pixels X from I and regroup remaining out-of-sample pixels in the subset Y.
2. Select p pixel prototypes by class and generate the sets of pairwise constraints M and C.
3. Compute d color texture features for each sample pixel of X.

3.2 Constrained feature selection and spectral clustering of sample pixel set X

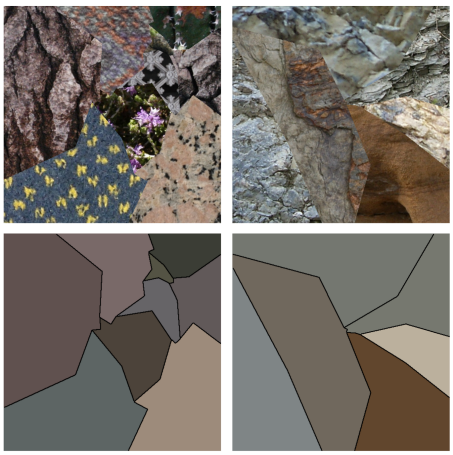
4. Select $\hat{m}$ most relevant features $F_{\hat{m}}$ using our constrained feature selection (see Algorithm 2).
5. Construct the similarity matrix $\mathrm{W}^{M,C}(F_{\hat{m}}) \in \mathbb{R}^{n \times n}$ from X using Eq. (25).
6. Compute the normalized Laplacian matrix $\mathrm{L}_{Sym}^{M,C}$ associated to $\mathrm{W}^{M,C}(F_{\hat{m}})$.
7. Compute the spectrum of $\mathrm{L}_{sym}^{M,C}$, and find k dominant eigenvectors $\mathrm{U}^{M,C} = [u_1, \ldots, u_k] \in \mathbb{R}^{n \times k}$ corresponding to k smallest eigenvalues $\lambda^{M,C} = [\lambda_1, \ldots, \lambda_k] \in \mathbb{R}^k$.
8. Construct the matrix $\mathrm{T}^{M,C} \in \mathbb{R}^{n \times k}$ from $\mathrm{U}^{M,C}$ by normalizing each row of $\mathrm{U}^{M,C}$ to have unit length
$$t_{ij} = \frac{u_{ij}}{\sqrt{\sum_{j=1}^{k} u_{ij}^2}}$$
9. Handle each row of $\mathrm{T}^{M,C}$ as a point, and regroup them into k clusters using the nearest neighbor algorithm with the p prototypes of each class.

3.3 Segmentation by classification of the out-of-sample pixel set Y

10. For each out-of-sample pixel y_i:
    - Compute $\hat{m}$ most relevant features.
    - Compute the normalized kernel $\tilde{\mathrm{W}}$ defined by Eq. (28).
    - Compute the embedded point $\mathrm{z}(y_i)$ defined by Eq. (26).
    - Compute the normalized embedded point $\tilde{\mathrm{z}}(y_i)$ using Eq. (29).
    - Assign y_i to the cluster of the nearest sample point, among $\mathrm{T}^{M,C}$, of $\tilde{\mathrm{z}}(y_i)$.
11. Refinement procedure (optional).

Output: Segmented image.

selection procedures are performed on the sample pixel set X, and repeated over 100 runs. At each feature selection run, the $(k \cdot p)$ prototype pixels with p ranging from 2 to 5 are randomly



(a) (b)

Fig 3 Two test color-texture images from the Prague dataset (first row) and their corresponding ground truth images (bottom row): (a) image 7_1_1 and (b) image 17_1_1.

selected from the ground truth of an image. Following this, the sets of pairwise constraints are derived using Eqs. (2), (3), and (23). These same prototype pixels are used for the feature selection and the constrained spectral clustering to evaluate the relevance of the selected features. The sampling procedure described in Section 3.1 extracts the $n - (k \cdot p)$ unlabeled sample pixels from each image. Several numbers of sample pixels ($n = 200, 300$, and 400) are tested to study the influence on the quality of feature selection. Furthermore, two images with 6 and 9 textures from the Prague dataset are used to illustrate our experiments regarding parameter adjustment (see Fig. 3).

4.1.1 Constraint scores with respect to number of features

The performance obtained by the proposed constraint score ε^{SS} is compared with that reached by the constraint scores C^c ($c = 1, \ldots, 5$). Figure 4 illustrates the variation of the constraint scores ε^{SS}, C^1, C^2, C^3, C^4, and C^5 with respect to the number m of features when the number p of prototypes is set to 5 and the number n of sample pixels is set to 300. The curves of ε^{SS} are quasi convex, whereas those of scores C^c ($c = 1, \ldots, 5$) monotonically

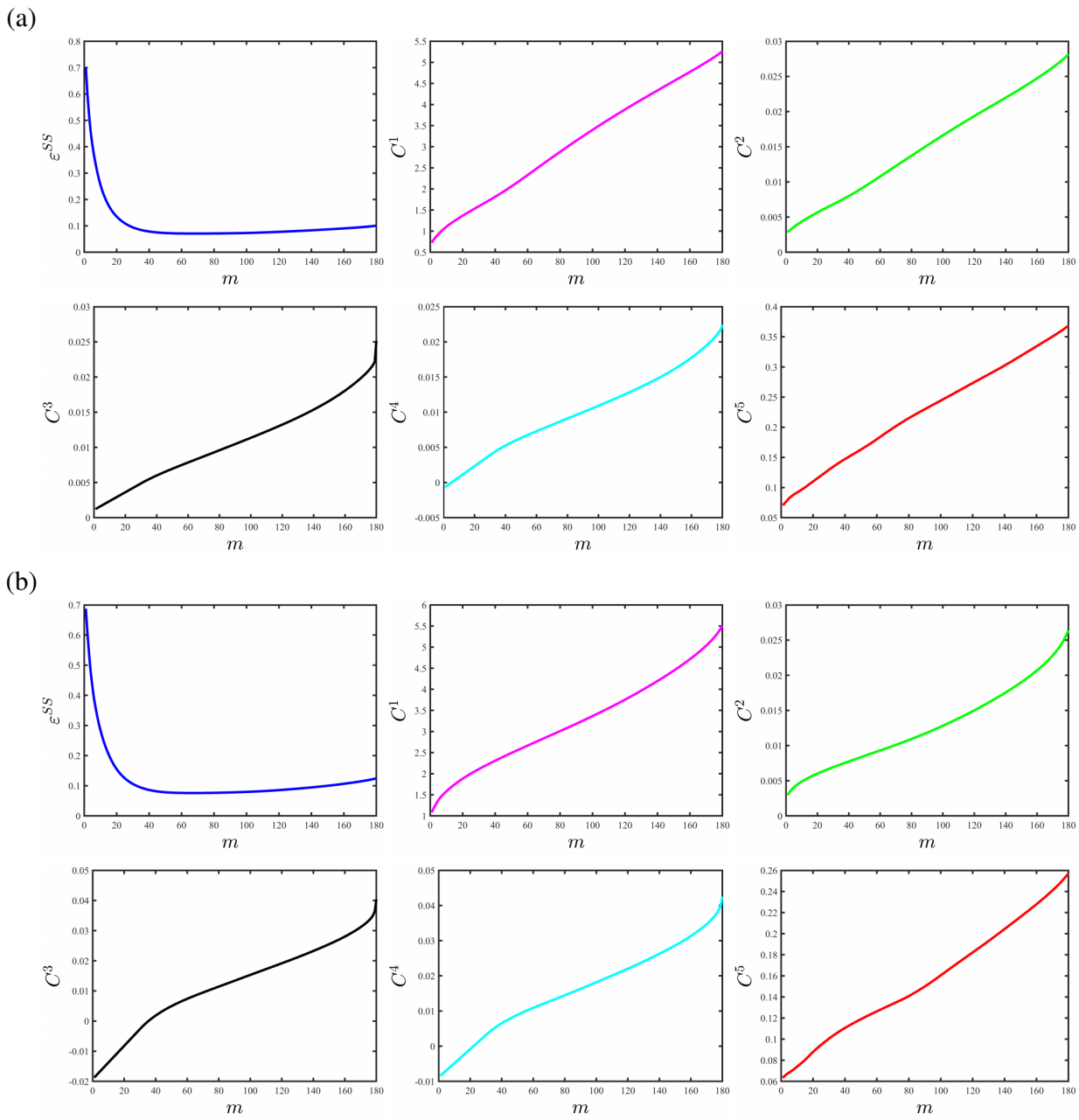


Fig 4 Constraint scores versus number m of selected features for the two test images: (a) image 7_1_1 and (b) image 17_1_1.

increase with respect to the number m of features. Similar results are observed for other numbers of prototypes and sample pixels (these results are omitted from this paper to avoid excessive length). We can conclude that only ε^{SS} presents a minimum value that can be considered as the optimal number of features.

4.1.2 Accuracy of unlabeled pixels

The performance of the constraint score ε^{SS} is evaluated according to the classification accuracy of the set X^U of unlabeled sample pixels, which is obtained by the constrained spectral clustering described in Sec. 3.4. First, we study the influence of the number p of prototypes by class and the number n of sample pixels on the accuracy achieved with ε^{SS}. Next, we compare the accuracy obtained using our constraint score with those obtained using state-of-the-art constraint scores.

Accuracy versus number p of prototypes. The number p of prototypes by class is related to the number of pairwise constraints and may modify the classification result provided by the constrained spectral clustering. To stay in the semi-supervised learning context, we have chosen small values for p ranging from 2 to 5. Figure 5 shows the variation of the average accuracy of X^U classification obtained with ε^{SS} over 100 runs according to the number p of prototypes and the number n of sample pixels. As expected, the curves of this figure show that, for a given number n of sample pixels, the accuracy achieved with our constraint score increases with respect to the number p of prototypes by class.

Accuracy versus number n of sample pixels. The number n of sample pixels is related to the spatial sampling of the image. On the one hand, n must be large enough to reach a satisfying spatial definition of the detected regions. On other hand, n must be small enough to reduce the computational complexity of the constrained spectral clustering of the unlabeled sample pixels. To satisfy a trade-off between the spatial definition and complexity, the n range is $[200 - 400]$, which corresponds on average to 0.10% of the number N of pixels. To show

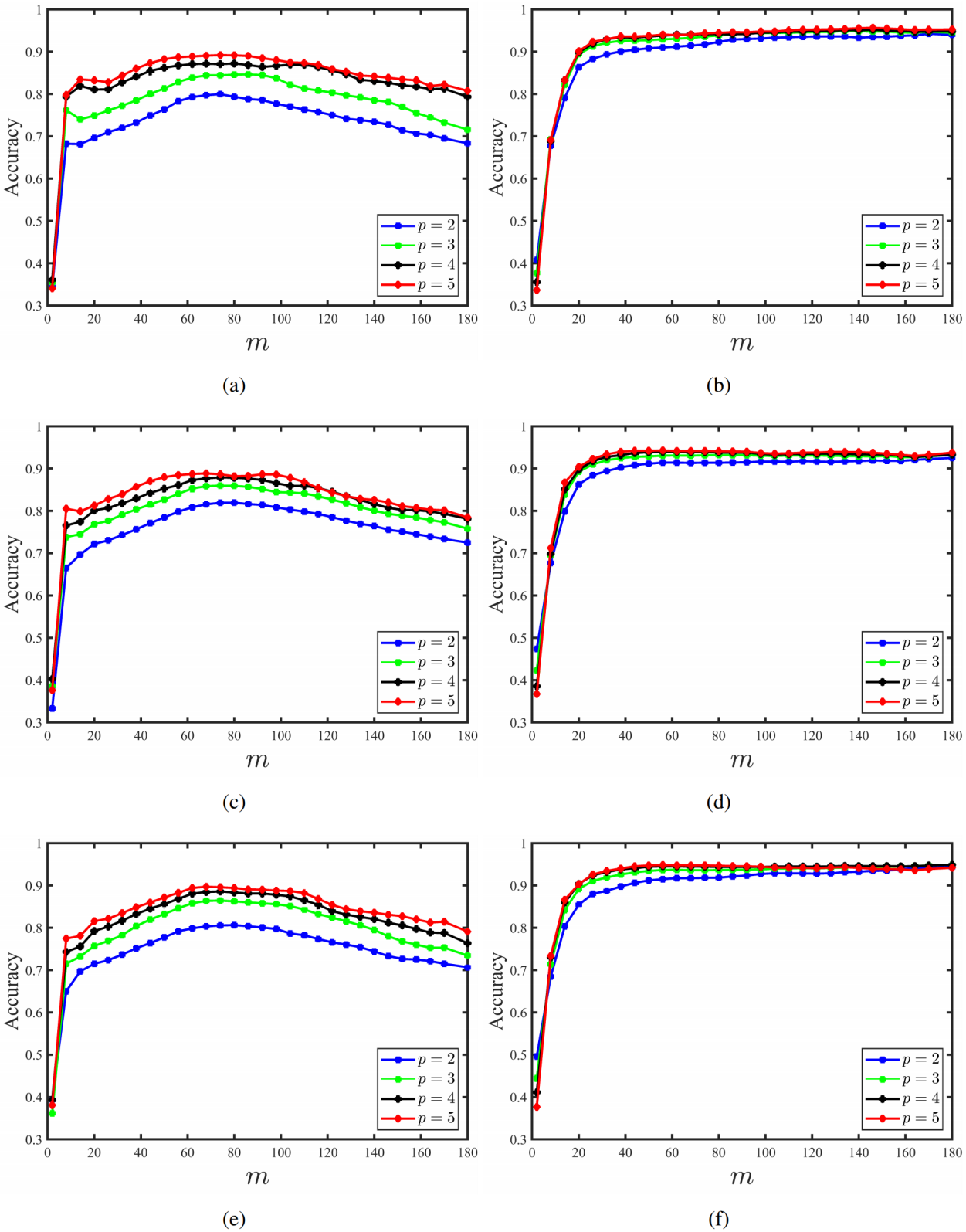


Fig 5 Accuracy versus number m of features obtained using the proposed semi-supervised constraint scores on test images for different numbers of n and p. Left column: image 7_1_1 and right column: image 17_1_1. (a)-(b) $n = 200$, (c)-(d) $n = 300$, and (e)-(f) $n = 400$.

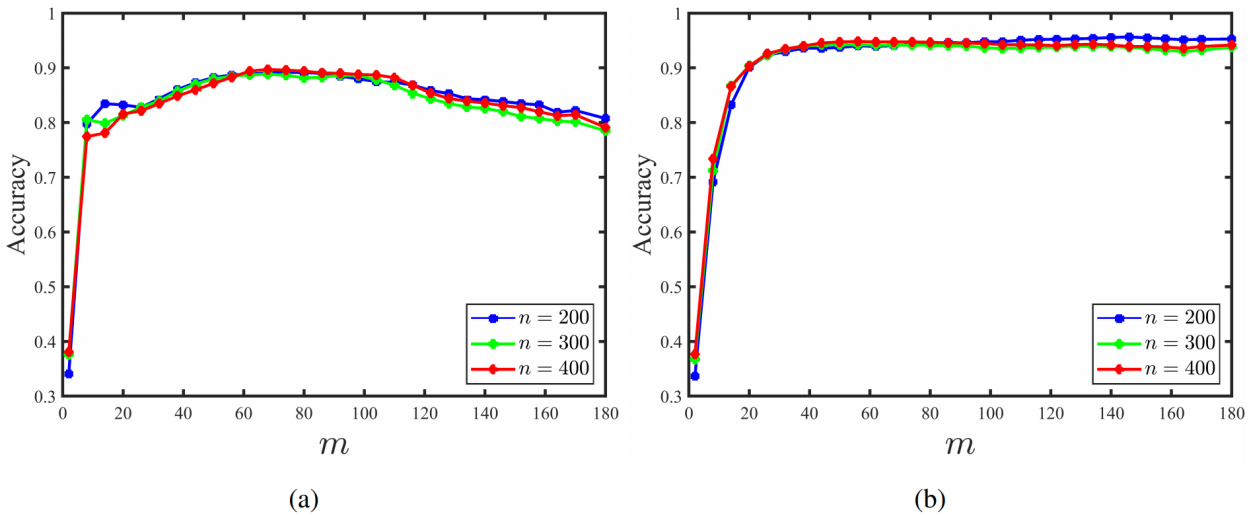


Fig 6 Accuracy versus number m of features obtained using the proposed semi-supervised constraint score on the two test images for different numbers n of sample pixels: (a) image 7_1_1 and (b) image 17_1_1.

the influence of this parameter, Fig. 6 displays the average accuracy of X^U classification obtained with ε^{SS} over 100 runs when the number p of prototypes is set to 5. The curves of accuracy obtained with $n = 200, 300$, and 400 overlap, which means that the classification accuracy of unlabeled pixels does not significantly vary when the n range is $[200 - 400]$.

4.1.3 Optimal number of selected features

As indicated in Sec. 4.1.1, the curve of constraint score ε^{SS} versus the number of features presents a minimum that can be considered as the optimal number of features. To validate this assertion, we simultaneously display in Fig. 7 the curves of ε^{SS} and the accuracy versus the number of features achieved when n is set to 300 and p is set to 5. We can see that the accuracy $T_c(\hat{m})$ obtained with the subset of features $F_{\hat{m}}$ that corresponds to the minimum of ε^{SS} coincides with or is close to the maximum value of the accuracy $T_c(m^*)$.

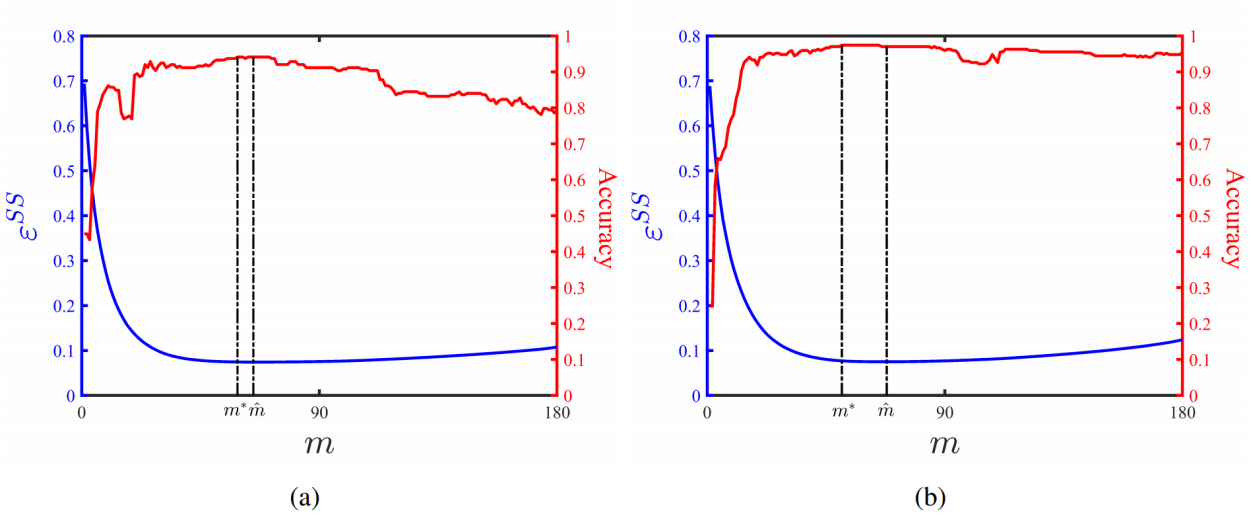


Fig 7 Accuracy-score ε^{SS} versus number m of selected features for the test images: (a) image 7_1_1 and (b) image 17_1_1.

4.1.4 Comparison with the state-of-the-art constraint scores

We compare the accuracy achieved by our constraint score with those obtained by the state-of-the-art constraint scores, according to the number of features and the number of prototypes. Figure 8 displays the average accuracy of the X^U classification for ε^{SS} and C^c $(c = 1, \ldots, 5)$ obtained over 100 runs when the number p of prototypes is set to 5 and the number n of sample pixels is set to 300 (i.e. the middle of the range $[200 - 400]$). This figure indicates that our constraint score ε^{SS} provides higher accuracy than those achieved with the state-of-the-art constraint scores C^c $(c = 1, \ldots, 5)$.

In order to compare the performance of our constraint score with the state-of-the-art scores according to the number of pairwise constraints, the number of selected features for each score is set to half of all features (i.e. $m = 90$). Figure 9 shows the average accuracy obtained over 100 runs by the constraint scores versus number p of prototypes when n is set to 300. The accuracy obtained with ε^{SS} is higher than those obtained with C^c $(c = 1, \ldots, 5)$ for any value of p.

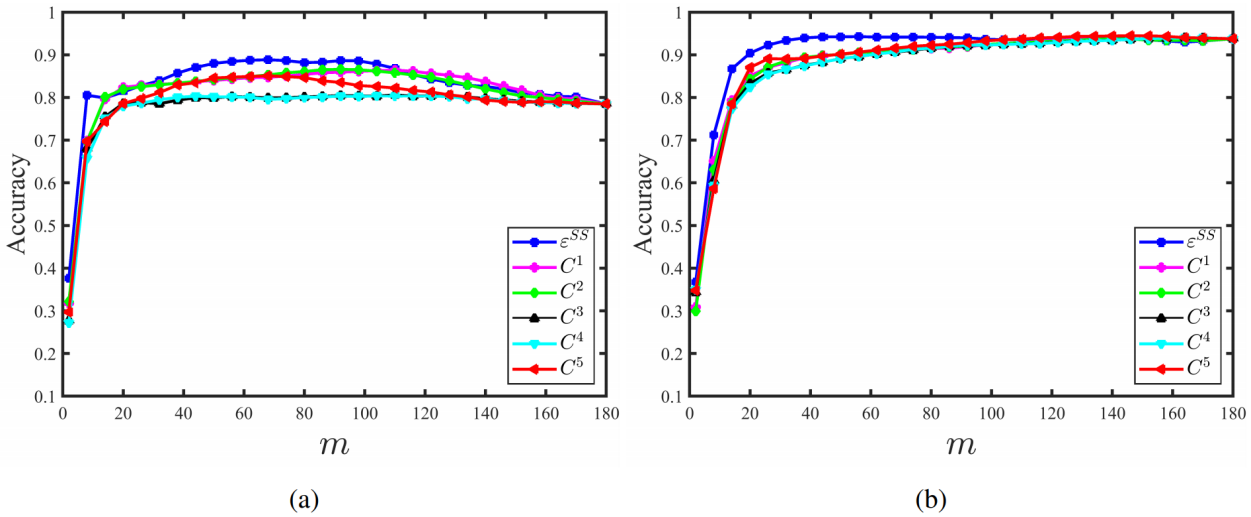


Fig 8 Accuracy versus number m of features for the two test images: (a) image 7_1_1 and (b) image 17_1_1.

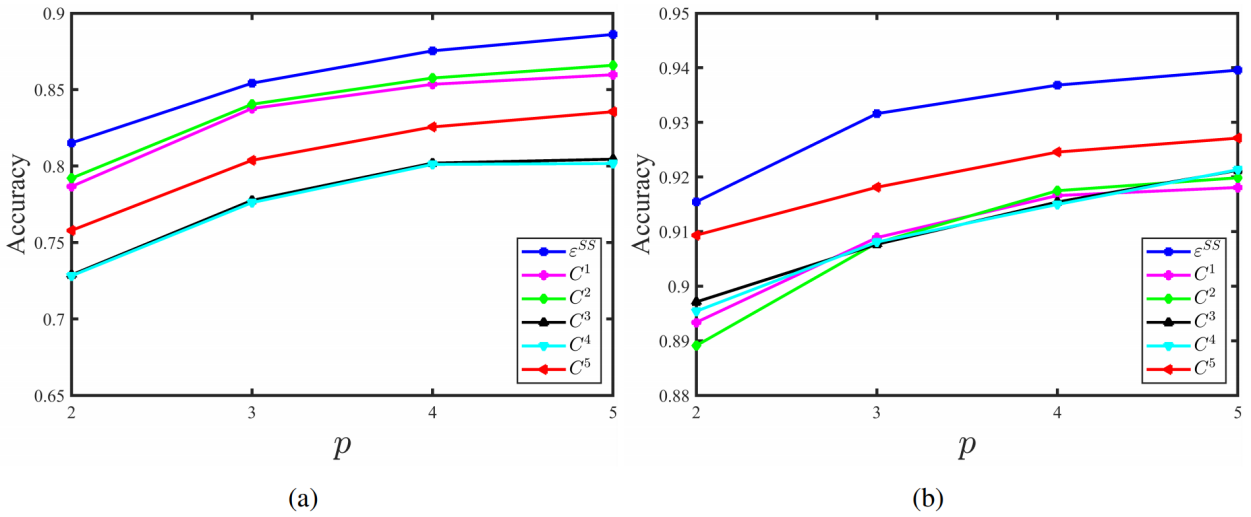


Fig 9 Accuracy versus number p of prototypes when $m = 90$ features have been selected and $n = 300$ on the two test images: (a) image 7_1_1 and (b) image 17_1_1.

4.2 Segmentation results on Prague dataset

Here, we present the results obtained by our proposed semi-supervised segmentation method with (CFS-SC) and without refinement (CFS-SC-nr; see Algorithm 3) on the Prague dataset and compare them with those of several state-of-the-art segmentation methods. The Prague dataset, known to be difficult to segment, contains 80 texture mosaics synthetically generated from 114 color-texture images in 10 categories (see ground truth in Ref. 39). The texture class number in each of the images, which are sized at 512×512 pixels, ranges from 3 to 12.

The assessment of segmentation performance is based on the conventional measures provided by the Prague texture segmentation website,[39] which include:

region-based criteria: correct segmentation (CS), over-segmentation (OS), under-segmentation (US), missed error (ME), noise error (NE);

pixel-wise based criteria: omission error (O), commission error (C), class accuracy (CA), recall (CO), precision (CC), type I error (I.), type II error (II.), mean class accuracy estimate (EA), mapping score (MS), root mean square proportion estimation error (RM), comparison index (CI);

consistency-error criteria: global consistency error (GCE) and local consistency error (LCE);

clustering comparison criteria: Mirkin metric (dM), Van Dongen metric (dD), variation of information (dVI).

The best score is displayed as bold in each table.

The methods involved in comparison include:

Six unsupervised algorithms: 1) the texNcut algorithm that is based on spectral clustering algorithm[5] and which uses texture features computed on super-pixels, 2) the variational

multi-phase segmentation framework (PCA-MS)[40] that uses a filter bank for feature extraction, 3) the model-based learning of local image features (MLLIF)[7] that learns features from images without ground truth segmentation, 4) the factorization-based texture segmentation (FSEG) algorithm[41] that characterizes pixels by local LoG/Gabor spectrum features, 5) the unsupervised fully convolutional network for texture (FCNTunsup) algorithm, and 6) the dictionary learning based sparse representation (DLSRC)[42] that extracts the features from wavelet transform and co-occurrence matrices.

Four semi-supervised algorithms: 1) the weakly-supervised sparse coding geometric priori (WSSCGP)[9] that requires a small amount of prototypes to perform segmentation. WSSCGP adopts the same local LoG/Gabor spectrum features than FSEG, 2) FSEG^{+}, which is a semi-supervised version of FSEG in which the same prototypes used in WSSCGP are integrated in FSEG,[9] 3) our proposed semi-supervised method using only the constrained spectral clustering (CSC), without feature selection procedure, i.e., with 180 features, and 4) our proposed semi-supervised method CFS-SC.

Eight supervised algorithms: 1) the MRF algorithm based on a Markov random field pixel classification model,[43] 2) the COF algorithm that uses the co-occurrence features and the nearest neighbor classifier,[39] 3) Con-Col algorithm,[39] 4) the supervised fully convolutional network for texture (FCNTsup) algorithm,[2] 5) the empirical wavelet transform based fully convolutional network for texture (EWT-FCNT)[2] that combines the empirical wavelet transform with FCNTsup, 6) U-Net,[44] 7) Deep Visual Model (DA),[45] and 8) the pyramid scene parsing network (PSP-Net).[46]

The results of supervised algorithms (Table 2) were obtained using the normal Prague dataset, which contains 20 test images.[39] The results of semi-supervised and unsupervised

algorithms (Table 1) were obtained over the large Prague dataset (80 test images), which includes the 20 test images of the normal Prague dataset.[39]

The segmentation results of FCNT, texNcut, FSEG, MRF, COF, and Con-Col were reported on the Prague benchmark website.[39] Those of DLSRC, WSSCGP, and FSEG$^+$ were described in Ref. 9 and in Ref. 7 for MLLIF. The results of EWT-FCNT, U-Net, DA, and PSP-Net were taken from Ref. 2, and those of PCA-MS from Ref. 40. The majority of these methods involved a refinement step to improve their performances, except for texNcut, MRF, COF, and Con-Col. It is important to note that our methods CSC and CFS-SC were run with $n = 300$ sample pixels and $p = 5$ prototypes for each texture class, while FSEG$^+$ and WSSCG used 81 prototypes per class. These prototypes are included in square regions of (9×9) whose centers are marked by the user inside each texture region.

Next, we examine tables 1 and 2 according to three general comparisons.

4.2.1 Comparison with unsupervised algorithms

In Table 1 we can see that all semi-supervised algorithms, except FSEG$^+$, outperformed the unsupervised algorithms. Moreover, our proposed methods with (CFS-SC) and without refinement (CFS-SC-nr) exhibited the best results for all the criteria except for GCE and LCE.

4.2.2 Comparison with semi-supervised algorithms

To assess the relevance of the proposed semi-supervised feature selection on the segmentation performance, we compare the segmentation accuracy benchmark results obtained by our method with feature selection (CFS-SC) and without feature selection (CSC). The results in

Table 1 Results of unsupervised and semi-supervised methods on the large Prague dataset (80 test images). The arrows ↑ | ↓ denote the required criterion direction. Here, 'nr' means no segmentation refinement.

Method	Unsupervised						Semi-supervised					
	FCNT unsup	DLS RC	tex Ncut	PCA MS	MLL IF	FSEG	FSEG$^+$	WSS CGP	CSC nr	CSC	CFS SC-nr	CFS SC
↑ CS	79.34	77.46	72.54	72.27	77.73	69.18	75.97	84.18	79.73	82.97	93.82	94.74
↓ OS	13.67	28.40	10.92	18.33	15.92	14.69	3.38	10.95	0.14	0.14	0.00	0.00
↓ US	6.25	0.00	9.61	9.41	6.31	13.64	5.53	0.00	0.00	0.00	0.00	0.00
↓ ME	3.80	7.13	10.25	4.19	3.93	5.13	11.82	6.90	14.64	11.92	1.98	1.41
↓ NE	3.80	7.39	9.83	3.92	3.92	4.63	11.49	7.04	15.48	12.34	2.59	1.85
↓ O	6.47	8.58	7.33	7.25	7.68	9.25	9.12	6.84	8.59	7.32	4.35	3.70
↓ C	22.88	29.48	8.17	6.44	24.24	12.55	9.34	7.09	7.33	6.02	3.87	3.39
↑ CA	84.17	83.41	80.58	81.13	82.80	78.22	80.26	87.05	84.33	86.01	91.89	92.62
↑ CO	87.97	87.36	86.89	85.96	86.89	84.44	88.09	91.58	90.50	91.68	95.51	95.98
↑ CC	94.15	95.16	88.28	91.24	93.65	87.38	88.19	94.85	91.92	92.92	95.94	96.31
↓ I.	12.03	12.64	13.11	14.04	13.11	15.56	11.91	8.42	9.50	8.32	4.49	4.02
↓ II.	1.42	1.19	2.36	1.59	1.50	2.53	2.47	1.28	1.79	1.64	0.75	0.71
↑ EA	88.97	89.70	86.39	87.08	88.03	84.24	87.40	92.53	90.69	91.76	95.61	96.04
↑ MS	85.23	84.74	80.33	81.84	83.93	78.81	82.46	88.63	85.75	87.51	93.27	93.97
↓ RM	3.12	2.42	3.69	5.11	3.27	4.74	2.99	1.89	1.99	1.91	0.81	0.77
↑ CI	89.91	90.44	86.97	87.81	89.03	85.03	87.76	92.86	90.94	92.02	95.67	96.09
↓ GCE	6.46	9.56	11.92	8.35	7.40	9.35	15.08	10.21	14.62	12.86	8.00	7.26
↓ LCE	4.75	7.17	6.85	5.61	5.62	6.08	11.71	7.71	11.52	9.79	6.28	5.59
↓ dD	7.79	9.08	9.18	9.06	8.57	10.01	10.34	7.21	9.28	8.12	4.48	4.02
↓ dM	4.88	5.40	6.03	5.89	5.30	7.01	6.59	4.40	5.57	4.95	2.66	2.41
↓ dVI	14.75	15.18	14.19	14.54	14.88	14.33	14.32	14.52	14.43	14.32	14.18	14.13

Table 1 clearly show the benefit provided by the proposed feature selection on the segmentation. When we compare the semi-supervised methods, we observe that the performance of the proposed semi-supervised method (CFS-SC), even without refinement, is superior to that of FSEG$^+$ and WSSCGP.

Table 2 Results of our method CFS-SC and supervised methods on the normal Prague dataset (20 test images). The arrows ↑ | ↓ denote the required criterion direction. Here, 'nr' means no segmentation refinement.

Method	Supervised									Semi-supervised	
	MRF	COF	Con Col	FCNT sup-nr	FCNT sup	EWT FCNT	U-Net	DA	PSP Net	CFS SC-nr	CFS SC
↑ CS	46.11	52.48	84.57	87.52	96.01	98.45	96.71	94.18	96.45	94.03	94.97
↓ OS	0.81	0.00	0.00	0.00	1.56	0.00	1.71	0.00	0.17	0.00	0.00
↓ US	4.18	1.94	1.70	0.00	1.20	0.00	0.00	1.18	0.41	0.00	0.00
↓ ME	44.82	41.55	9.50	6.70	0.78	0.37	0.68	3.42	1.23	1.63	1.09
↓ NE	45.29	40.97	10.22	6.90	0.89	0.46	0.48	3.24	1.12	2.32	1.58
↓ O	14.52	20.74	7.00	7.46	2.72	0.93	0.72	3.13	2.75	4.42	3.83
↓ C	16.77	22.10	5.34	6.16	2.29	1.04	0.70	1.32	2.39	4.45	3.80
↑ CA	65.42	67.01	86.21	87.08	93.95	97.67	95.86	94.53	93.89	91.85	92.61
↑ CO	76.19	77.86	92.02	92.61	96.73	98.78	96.91	96.23	96.06	95.49	95.97
↑ CC	80.30	78.34	92.68	93.26	97.02	98.81	97.38	97.01	96.41	95.91	96.29
↓ I.	23.81	22.14	7.98	7.39	3.27	1.22	3.09	3.77	3.94	4.51	4.03
↓ II.	4.82	4.40	1.70	1.49	0.68	0.25	0.41	0.58	0.69	0.79	0.76
↑ EA	75.40	76.21	91.72	92.68	96.68	98.77	97.01	96.24	96.08	95.59	96.04
↑ MS	64.29	66.79	88.03	88.92	95.10	98.17	95.37	94.35	94.08	93.24	93.96
↓ RM	6.43	4.47	2.08	1.38	0.86	0.24	0.61	1.07	0.70	0.79	0.70
↑ CI	76.69	77.05	92.02	92.81	96.77	98.78	97.08	96.41	96.15	95.65	96.08
↓ GCE	25.79	23.94	11.76	12.54	5.55	2.33	2.13	3.50	4.67	8.07	7.30
↓ LCE	20.68	19.69	8.61	9.94	3.75	1.68	1.46	2.47	3.52	6.40	5.75
↓ dD	20.35	17.86	7.50	-	3.06	1.21	1.45	2.41	2.59	4.50	4.02
↓ dM	13.25	10.62	4.69	-	1.96	0.74	0.77	1.35	1.56	2.76	2.48
↓ dVI	14.51	14.22	13.99	-	13.80	13.68	13.68	13.71	13.77	14.01	13.96

4.2.3 Comparison with supervised algorithms

Although the prior knowledge used by the supervised methods is more important than that used by our semi-supervised method, we can observe in Table 2 that our proposed method, both with and without refinement, largely outperformed the classical supervised methods

(MFR, COF, and Con-Col). The methods based on the deep learning (FCNTsup, DA, U-Net, PSP-Net, and EWT-FCNT) outperform CFS-SC for most criteria. However, our method CFS-SC is competitive with deep learning methods except EWT-FCNT that provides exceptional results due to expensive learning step. For example, the CFS-SC does not provide under-segmentation nor over-segmentation (US and OS measures are equal to 0) compared to deep learning methods. It is also noticeable that our method CFS-SC without refinement outperformed the FCNTsup without refinement.

For a visual comparison, Fig. 10 shows some segmentation results obtained by our CFS-SC method, WSSCGP, and by four top-performing methods whose segmented images were available in Ref. 39. For clarity, the black lines highlight the boundaries of segmented texture regions. Overall, CFS-SC, FCNTsup, and EWT-FCNT approaches provide satisfactory visual segmentation results that are close to the ground truth.

4.3 Segmentation results on Histology dataset

The Histology dataset[47,48] contains 36 color images of size 128 × 128 pixels that represent two types of tissue. The ground truth images were determined by professional pathologists through visual inspection. We use the same setup as for the Prague texture experiment except for the Haralick parameter ω that we set to 5. To assess the performance of our methods CFS-SC and CSC without and with refinement, we compare their results with several methods including, FSEG^{+}, WSSCGP, DLSRC, and FSEG as well as an another unsupervised method, called occlusion of random textures segmentation method (ORTSEG).[48] The quantitative evaluation is based on the same measures used for the Prague dataset. Table 3

Fig 10 Exemplary segmentation results on the Prague color benchmark. From top to bottom: Original images, ground truth, FCNTsup, EWT-FCNT, FCNTunsup, texNcut, WSSCGP, CFS-SC-nr, and CFS-SC.

shows that the proposed algorithm, even without refinement, largely outperforms the other methods. Fig. 11 displays the segmentation results achieved on some images of Histology dataset. To highlight the accuracy of segmentation results, the boundaries of detected regions are embedded in the original images. We clearly observe that the boundaries between the two tissues are better detected by our method and close to the ground truth.

Table 3 Results on the Histology dataset. The arrows ↑ | ↓ denote the required criterion direction and the best score is marked in boldface. Here, 'nr' means no segmentation refinement.

Method	Unsupervised			Semi-supervised				
	DLS RC	ORT SEG	FSEG	FSEG^{+}	WSS CGP	CSC	CFS SC-nr	CFS SC
↑ CS	82.98	72.53	38.98	82.06	86.90	82.54	96.93	97.07
↓ OS	3.34	1.71	35.95	1.78	3.29	0.00	0.00	0.00
↓ US	2.82	2.78	0.00	10.25	1.94	0.00	0.00	0.00
↓ ME	9.69	20.09	19.96	7.67	5.95	11.33	0.00	0.00
↓ NE	11.49	20.31	21.05	8.25	7.02	11.99	0.00	0.00
↓ O	7.64	10.17	24.77	13.71	6.70	8.69	3.26	3.10
↓ C	6.96	10.57	24.23	7.42	6.21	9.62	3.47	3.31
↑ CA	86.80	81.81	67.17	84.90	87.69	84.71	94.12	94.37
↑ CO	91.44	88.84	72.38	90.70	92.07	91.20	96.93	97.07
↑ CC	94.96	91.27	92.27	92.02	95.28	91.99	97.00	97.14
↓ I.	8.56	11.16	27.62	9.30	7.93	8.80	3.07	2.93
↓ II.	5.08	8.98	6.99	11.40	5.07	8.57	3.45	3.27
↑ EA	92.60	88.71	78.78	90.66	93.15	91.21	96.94	97.08
↑ MS	87.17	83.43	67.63	86.05	88.11	86.8	95.40	95.61
↓ RM	6.55	9.93	17.29	7.42	5.82	5.22	1.06	1.13
↑ CI	92.89	89.37	80.49	91.00	93.40	91.4	96.95	97.09
↓ GCE	10.42	12.36	14.10	9.64	9.73	13.65	5.77	5.52
↓ LCE	6.77	7.23	11.24	6.12	6.48	9.68	4.82	4.51
↓ dD	7.54	9.73	18.55	7.65	6.90	8.66	3.07	2.93
↓ dM	12.88	17.19	26.18	14.00	11.99	14.96	5.90	5.65
↓ dVI	5.88	5.64	7.23	5.61	5.85	5.68	5.43	5.42

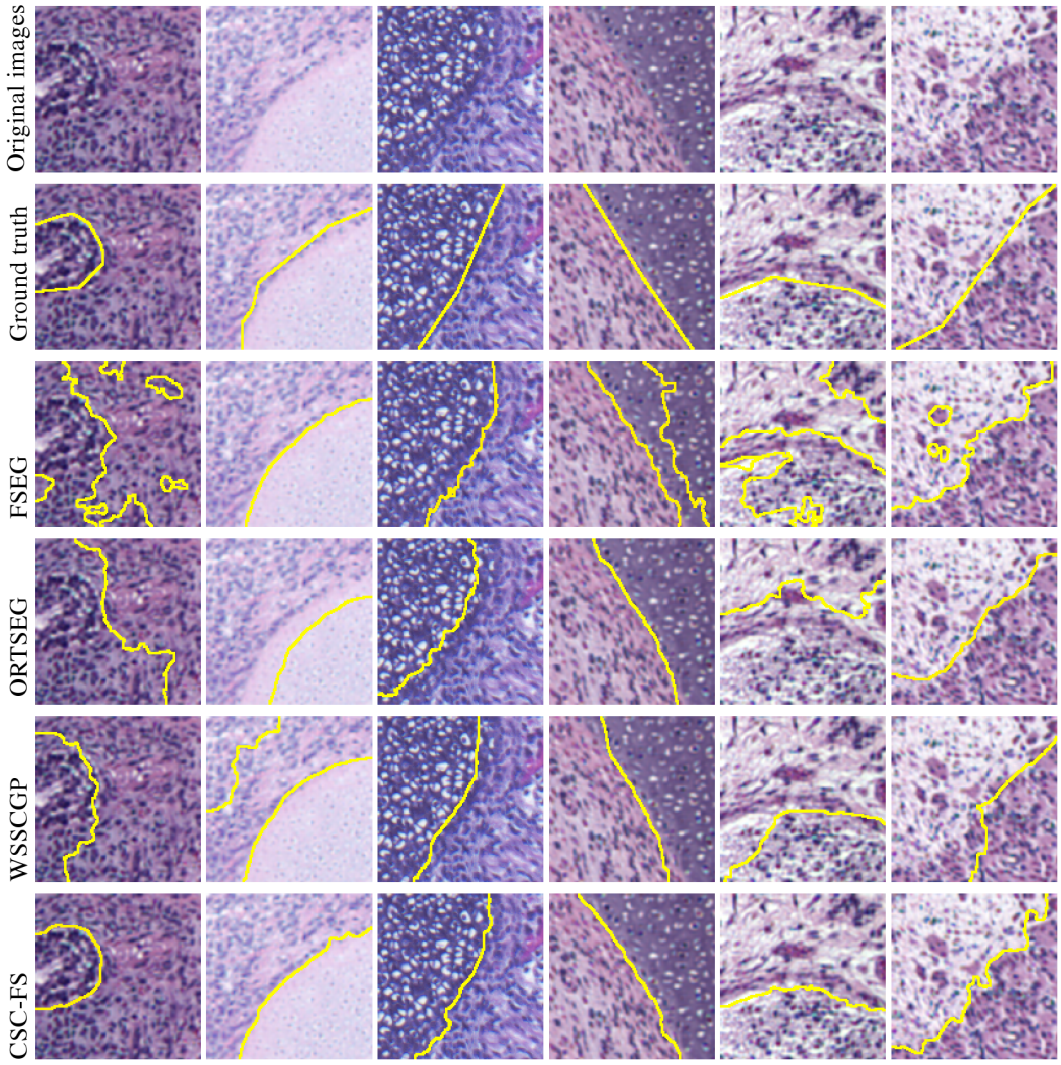


Fig 11 Exemplary segmentation results on the Histology dataset. From top to bottom: Original images, ground truth, FSEG, ORTSEG, WSSCGP, and CSC-FS.

4.4 Segmentation results on Outex dataset

Our proposed method CFS-SC is designed to segment color-texture images, but it is also suitable for grayscale texture segmentation. To assess its performance for grayscale textures, we apply it on the popular texture dataset Outex (US_00000 test suite)[49] and compare its results with those of state-of-the art algorithms: ORTSEG, PCA-MS, FSEG, EWT-FCNT, U-Net, DA, PSP-Net, and WSSCGP. This dataset contains 100 composite texture images of size 512×512 pixels, each one being a composite of five training textures. They are generated by mixing twelve different texture images with different rotations according to the regions depicted by the ground truth.[49] For convenience, we analyze only the texture features that are computed from the gray level images. The size of neighborhood ω used for computing the Haralick features is set to 5. To quantitatively assess segmentation performance, we use the same measures as in Ref. 2 : recall (CO), normalized variation of information (NVOI), swapped directional hamming distance (SDHD), Van Dongen distance (VD), swapped segmentation covering (SSC), bipartite graph matching (BGM) and bidirectional consistency error (BCE).[50] All these measures provide a number between 0% and 100%, the latter corresponding to perfect segmentation. Fig. 12 shows some segmentation results obtained on the Outex dataset. A visual inspection illustrates the efficacy of the proposed method (CFS-SC) compared with the other segmentation methods. Table 4 gives quantitative measurements for the Outex dataset based on the previously mentioned mesures. We clearly observe that the proposed algorithm (with and without refinement) largely outperforms the two unsupervised algorithms (PCA-MS, FSEG), the semi-supervised algorithm (WSSCGP) and especially the three supervised methods (U-Net, DA, PSP-Net). However, CFS-SC provides lower results

than EWT-FCNT whose expensive learning reaches impressive performance.

Table 4 Results on the Outex dataset. Each row corresponds to a segmentation quality measure and the best score is marked in boldface. Here, 'nr' means no segmentation refinement.

Method	Unsupervised			Supervised				Semi-supervised		
	ORT SEG	FSEG	PCA MS	EWT FCNT	U-Net	DA	PSP Net	WSS CGP	CFS SC-nr	CFS SC
NVOI	77.99	73.65	83.86	96.52	72.83	70.70	86.00	85.41	87.88	88.70
SSC	70.88	62.23	84.32	98.27	64.25	60.65	84.22	89.11	92.42	93.01
SDHD	80.64	65.08	89.67	99.13	71.40	67.77	88.00	94.71	96.02	96.34
BGM	76.42	64.97	89.24	99.13	70.85	67.27	87.83	93.59	96.02	96.34
VD	83.87	80.60	91.39	99.13	80.47	78.75	91.64	94.15	96.02	96.34
BCE	69.50	61.08	82.45	98.02	61.89	58.58	82.63	86.64	91.19	91.77
CO	76.42	64.97	89.24	98.36	70.28	67.28	86.72	93.59	96.02	96.34

4.5 Segmentation results on Berkeley dataset

To study the efficiency of our proposed method, we applied CFS-SC on some natural images selected from the Berkeley database.[51] To evaluate segmentation performances, three quantitative measures, namely, F-measure,[51] probabilistic rand index (PRI),[50] and recall (CO), are computed. The values of F-measure and PRI fall in [0,1], and the larger, the better. These images, displayed on Fig. 13, are often considered because they include some uncertain and complex color-texture patterns. We compare our method CFS-SC with robust self-sparse fuzzy clustering algorithm (RSSFCA) that is a very recent image segmentation method.[52] Fig. 13 shows that the obtained segmentation results from a qualitative and quantitative point of view are better than those of RSSFCA and close to ground truth.

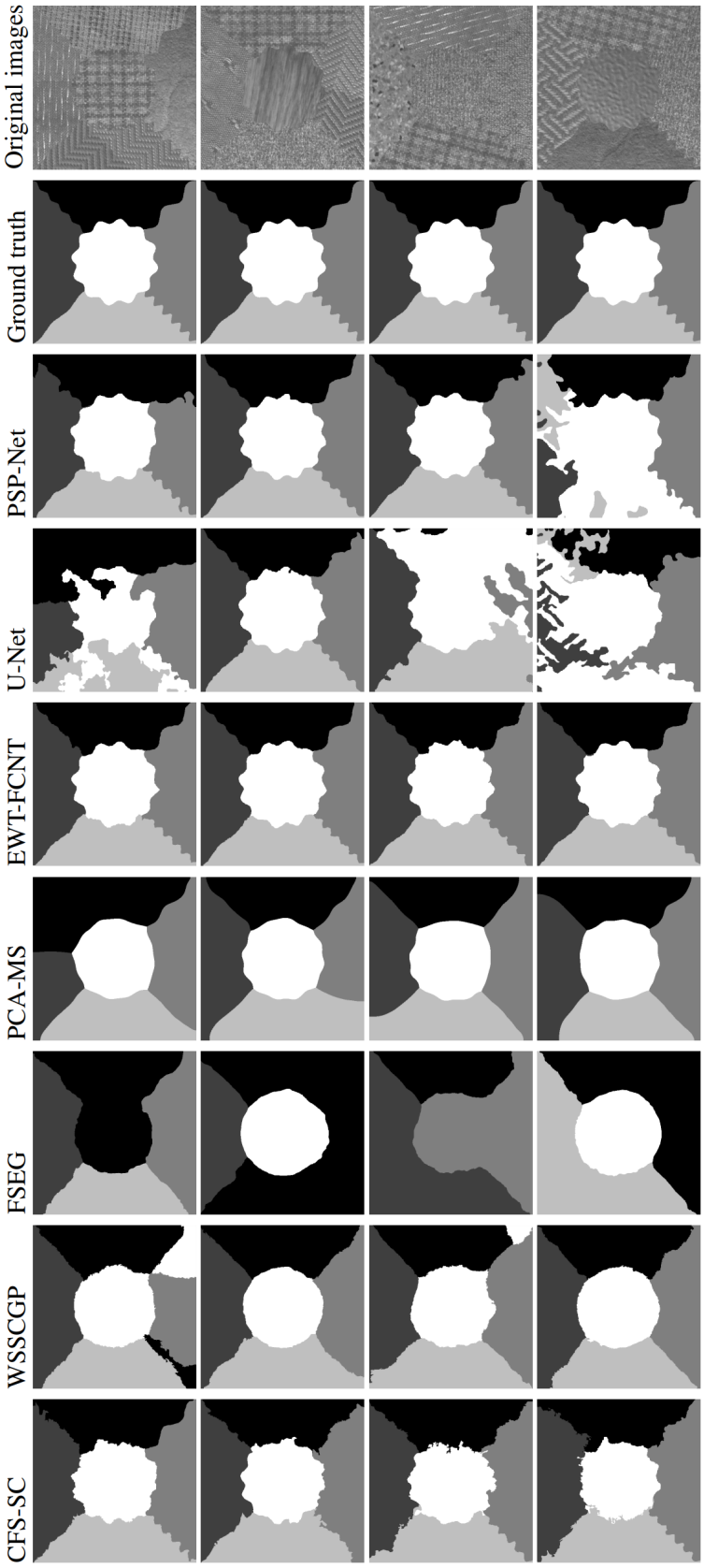


Fig 12 Exemplary segmentation results on the Outex dataset. From top to bottom: Original images, Ground truth, PSP-Net, U-Net, EWT-FCNT, PCA-MS, FSEG, WSSCGP, and CFS-SC.

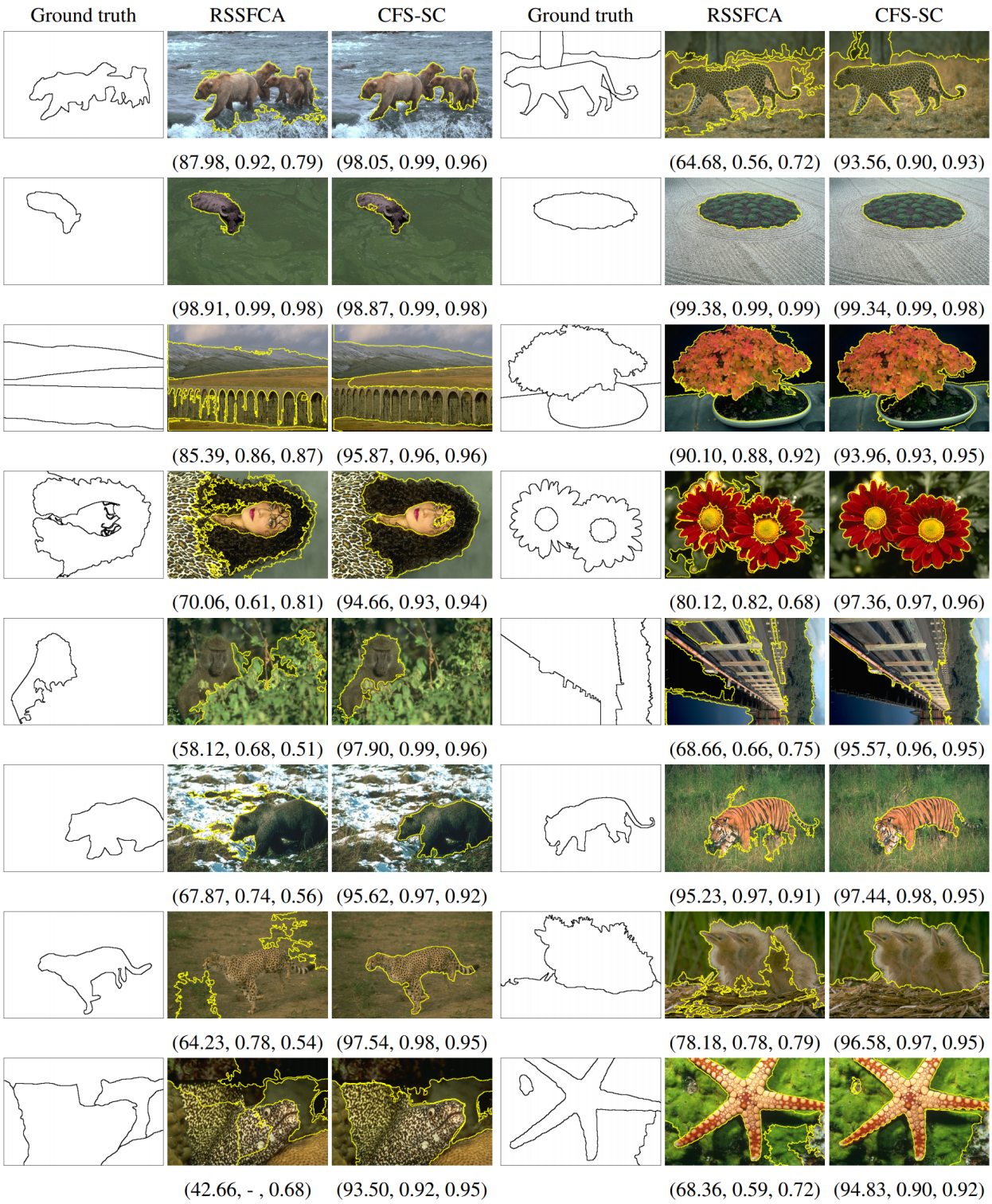


Fig 13 Segmentation results on some images of the Berkeley database. The values of CO, F-measure, and PRI for each result are presented in parentheses.

4.6 Processing time

To provide an overview on the computational requirements of the proposed CFS-SC algorithm, we estimated its processing time on an Intel Core i7 3.60GHz computer with 8GB RAM. Table 5 displaysthe average computing time of FSEG, WSSCGP, and CFS-SC for the tree datasets, i.e. according to size and type (color or grayscale) of image. The processing time of these methods can be divided into two parts: the processing times consumed by feature extraction step and by the other steps (feature selection and pixel classification steps). From this table, we can see that the processing time required by the CFS-SC algorithm is the highest. For example, it was about 518 seconds for a (512×512) image. This relatively long processing time led us revisit each step of our algorithm. The majority of this time was consumed by the feature extraction step (486 seconds) and more precisely by the Haralick feature extraction (403 seconds). The remaining time (34 seconds) was mainly shared by the three basic procedures of our CFS-SC algorithm; namely, the constrained feature selection (20.7 seconds), the constrained spectral clustering (0.033 seconds), and the classification of out-of-sample pixels (11.01 seconds). Apart from the feature extraction step, the algorithm runs relatively quickly. It should be emphasized that it is always possible to reduce the computational time for feature extraction step and that the computational time cannot be considered as a crucial drawback,given the appreciable segmentation results achieved by the CFS-SC algorithm on all the tested datasets.

5 Conclusion

In this paper, we proposed an efficient semi-supervised approach for color-texture image segmentation. The proposed method combines the constrained feature selection and the

Table 5 The average computing time (second) per image of FSEG, WSSCGP, and CFS-SC on the tree datasets. (FE: Feature extraction, OS: Other steps.)

Methods	Prague (512 × 512) color image			Histology (128 × 128) color image			Outex (512 × 512) grayscale image		
	FE	OS	Total	FE	OS	Total	FE	OS	Total
FSEG	0.41	0.67	1.08	0.02	0.04	0.06	0.14	0.41	0.55
WSSCGP	0.90	31.92	32.82	0.06	0.61	1.21	0.72	26.63	27.35
CFS-SC	486	31.74	517.74	26.89	18.21	45.10	67.87	10.69	78.56

constrained spectral clustering. A new constraint score is developed in order to select a subset of features at one time. As the proposed score evaluates the similarity between the sample pixels in the examined feature subspace, the selected features are better analyzed by constrained spectral clustering. Experiments on four benchmark datasets showed that the proposed constraint score outperforms the main state-of-the-art constraint scores in the semi-supervised learning context. Moreover, the result demonstrated that the proposed constrained feature selection and the constrained spectral clustering algorithm is very effective for color-texture image segmentation.

Finally, classical color-texture features and a simple sampling technique were used to conduct the experiments presented in this paper. It may to be advantageous to integrate other powerful texture descriptors, such as LBP, wavelets, or pre-trained CNNs, with a more elaborate sampling technique to further improve the segmentation accuracy. Moreover, we have considered that the prior knowledge is represented by class prototypes from which the pairwise constraints are deduced. All available constraints contribute to efficient feature selection and spectral clustering. In future work, we intend to generalize our method to prior knowledge that is only formalized by constraints.

References

1 S. Jain and V. Laxmi, "Color image segmentation techniques: A survey," in Proceedings of the International Conference on Microelectronics, Computing & Communication Systems, 189–197, Springer (2018). [doi:10.1007/978-981-10-5565-2_17].

2 Y. Huang, F. Zhou, and J. Gilles, "Empirical curvelet based fully convolutional network for supervised texture image segmentation," Neurocomputing 349, 31–43 (2019). [doi:10.1016/j.neucom.2019.04.021].

3 Z. Wang, "Image segmentation by combining the global and local properties," Expert Systems with Applications 87, 30 – 40 (2017). [doi:10.1016/j.eswa.2017.06.008].

4 W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37(5), 1382–1389 (2007). [doi:10.1109/TSMCB.2007.902249].

5 J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000). [doi:10.1109/34.868688].

6 D. E. Ilea and P. F. Whelan, "Image segmentation based on the integration of colour-texture descriptors a review," Pattern Recognition 44(10-11), 2479–2501 (2011). [doi:10.1016/j.patcog.2011.03.005].

7 M. Kiechle, M. Storath, A. Weinmann, et al., "Model-based learning of local image features for unsupervised texture segmentation," IEEE Transactions on Image Processing 27(4), 1994–2007 (2018). [doi:10.1109/TIP.2018.2792904].

8 M. Haindl and S. Mikeš, "A competition in unsupervised color image segmentation," Pattern Recognition 57, 136–151 (2016). [doi:10.1016/j.patcog.2016.03.003].

9 Y. Quan, H. Teng, T. Liu, et al., "Weakly-supervised sparse coding with geometric prior for interactive texture segmentation," IEEE Signal Processing Letters 27, 116–120 (2020). [doi:10.1109/LSP.2019.2959225].

10 S. Basu, I. Davidson, and K. Wagstaff, Constrained clustering: Advances in algorithms, theory, and applications, Chapman & Hall/CRC (2008).

11 U. Von Luxburg, "A tutorial on spectral clustering," Statistics and Computing 17(4), 395–416 (2007). [doi:10.1007/s11222-007-9033-z].

12 S. D. Kamvar, D. Klein, and C. D. Manning, "Spectral learning," in International Joint Conference of Artificial Intelligence, Stanford InfoLab (2003).

13 I. Ahn and C. Kim, "Face and hair region labeling using semi-supervised spectral clustering-based multiple segmentations," IEEE Transactions on Multimedia 18(7), 1414–1421 (2016). [doi:10.1109/TMM.2016.2551698].

14 W. Chen and G. Feng, "Spectral clustering: a semi-supervised approach," Neurocomputing 77(1), 229–242 (2012). [doi:10.1016/j.neucom.2011.09.002].

15 X. Zhu, C. C. Loy, and S. Gong, "Constrained clustering with imperfect oracles," IEEE Transactions on Neural Networks and Learning Systems 27(6), 1345–1357 (2015). [doi:10.1109/TNNLS.2014.2387425].

16 K. Xia, X. Gu, and Y. Zhang, "Oriented grouping-constrained spectral clustering for medical imaging segmentation," Multimedia Systems 26(1), 27–36 (2020). [doi:10.1007/s00530-019-00626-8].

17 J. Sourati, D. Erdogmus, J. G. Dy, et al., "Accelerated learning-based interactive image segmentation using pairwise constraints," IEEE Transactions on Image Processing 23(7), 3057–3070 (2014). [doi:10.1109/TIP.2014.2325783].

18 Z. Lu and M. A. Carreira-Perpinan, "Constrained spectral clustering through affinity propagation," in IEEE Conference on Computer Vision and Pattern Recognition, 1–8 (2008). [doi:10.1109/CVPR.2008.4587451].

19 P. He, X. Xu, and L. Chen, "Constrained clustering with local constraint propagation," in European Conference on Computer Vision, 223–232, Springer (2012). [doi:10.1007/978-3-642-33885-4_23].

20 P. Han, G. Liu, S. Huang, et al., "Segmentation with selectively propagated constraints," in Neural Information Processing, 585–592, Springer (2016). [doi:10.1007/978-3-319-46672-9_65].

21 S. X. Yu and J. Shi, "Segmentation given partial grouping constraints," IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 173–183 (2004). [doi:10.1109/TPAMI.2004.1262179].

22 B. Ghanem and N. Ahuja, "Dinkelbach ncut: An efficient framework for solving normalized cuts problems with priors and convex constraints," International Journal of Computer Vision 89(1), 40–55 (2010). [doi:10.1007/s11263-010-0321-2].

23 H. Zou, W. Zhou, L. Zhang, et al., "A new constrained spectral clustering for sar image segmentation," in Second Asian-Pacific Conference on Synthetic Aperture Radar, 680–683, IEEE (2009). [doi:10.1109/APSAR.2009.5374114].

24 Y. Jia and C. Zhang, "Learning distance metric for semi-supervised image segmenta-

tion," in Proceedings of the 15th IEEE International Conference on Image Processing, 3204–3207, IEEE (2008). [doi:10.1109/ICIP.2008.4712477].

25 R. Sheikhpour, M. A. Sarram, S. Gharaghani, et al., "A survey on semi-supervised feature selection methods," Pattern Recognition 64, 141–158 (2017). [doi:10.1016/j.patcog.2016.11.003].

26 J. Zhao, K. Lu, and X. He, "Locality sensitive semi-supervised feature selection," Neurocomputing 71(10), 1842–1849 (2008). [doi:10.1016/j.neucom.2007.06.014].

27 M. Kalakech, P. Biela, L. Macaire, et al., "Constraint scores for semi-supervised feature selection: A comparative study," Pattern Recognition Letters 32(5), 656–665 (2011). [doi:10.1016/j.patrec.2010.12.014].

28 K. Benabdeslem and M. Hindawi, "Constrained laplacian score for semi-supervised feature selection," in Proceedings of the Machine Learning and Knowledge Discovery in Databases, 204–218, Springer (2011). [doi:10.1007/978-3-642-23780-5_23].

29 K. Benabdeslem and M. Hindawi, "Efficient semi-supervised feature selection: constraint, relevance, and redundancy," IEEE Transactions on Knowledge and Data Engineering 26(5), 1131–1143 (2014). [doi:10.1109/TKDE.2013.86].

30 X.-K. Yang, L. He, D. Qu, et al., "Semi-supervised feature selection for audio classification based on constraint compensated laplacian score," EURASIP Journal on Audio, Speech, and Music Processing 2016(1), 9 (2016). [doi:10.1186/s13636-016-0086-9].

31 X.-K. Yang, L. He, D. Qu, et al., "Semi-supervised minimum redundancy maximum relevance feature selection for audio classification," Multimedia Tools and Applications 77(1), 713–739 (2018). [doi:10.1007/s11042-016-4287-0].

32 R. M. Haralick, K. Shanmugam, et al., "Textural features for image classification," IEEE Transactions on Systems, Man, and Cybernetics 3(6), 610–621 (1973). [doi:10.1109/TSMC.1973.4309314].

33 A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," Pattern Recognition 24(12), 1167–1186 (1991). [doi:10.1016/0031-3203(91)90143-S].

34 L. Liu, J. Chen, P. Fieguth, et al., "From bow to cnn: Two decades of texture representation for texture classification," International Journal of Computer Vision 127(1), 74–109 (2019). [doi:10.1007/s11263-018-1125-z].

35 L. Busin, N. Vandenbroucke, and L. Macaire, "Color spaces and image segmentation," Advances in Imaging and Electron Physics 151, 65 – 168, Elsevier (2009). [doi:10.1016/S1076-5670(07)00402-8].

36 W. Siedlecki and J. Sklansky, "On automatic feature selection," International Journal of Pattern Recognition and Artificial Intelligence 2(02), 197–220 (1988). [doi:10.1142/S0218001488000145].

37 Y. Bengio, J.-f. Paiement, P. Vincent, et al., "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," in Advances in Neural Information Processing Systems, 177–184, MIT Press (2004).

38 M. Haindl and S. Mikes, "Texture segmentation benchmark," in Proceedings of the 19th International Conference on Pattern Recognition, 1–4 (2008). [doi:10.1109/ICPR.2008.4761118].

39 Prague texture segmentation data generator and benchmark available from: http://mosaic.utia.cas.cz ((accessed May 23, 2020)).

40 N. Mevenkamp and B. Berkels, "Variational multi-phase segmentation using high-dimensional local features," in IEEE Winter Conference on Applications of Computer Vision (WACV), 1–9, IEEE (2016). [doi:10.1109/WACV.2016.7477729].

41 J. Yuan, D. Wang, and A. M. Cheriyadat, "Factorization-based texture segmentation," IEEE Transactions on Image Processing 24(11), 3488–3497 (2015). [doi:10.1109/TIP.2015.2446948].

42 S. Yang, Y. Lv, Y. Ren, et al., "Unsupervised images segmentation via incremental dictionary learning based sparse representation," Information Sciences 269, 48–59 (2014). [doi:10.1016/j.ins.2014.01.023].

43 Z. Kato, T.-C. Pong, and J. C.-M. Lee, "Color image segmentation and parameter estimation in a markovian framework," Pattern Recognition Letters 22(3-4), 309–321 (2001). [doi:10.1016/S0167-8655(00)00106-9].

44 O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical Image Computing and Computer-Assisted Intervention, 234–241, Springer (2015). [doi:10.1007/978-3-319-24574-4_28].

45 W. Wang and J. Shen, "Deep visual attention prediction," IEEE Transactions on Image Processing 27(5), 2368–2378 (2017). [doi:10.1109/TIP.2017.2787612].

46 H. Zhao, J. Shi, X. Qi, et al., "Pyramid scene parsing network," in IEEE Conference on Computer Vision and Pattern Recognition, 6230–6239 (2017). [doi:10.1109/CVPR.2017.660].

47 J. A. Ozolek and C. A. Castro, "Teratomas derived from embryonic stem cells as models

for embryonic development, disease, and tumorigenesis," in Embryonic Stem Cells-Basic Biology to Bioengineering, IntechOpen (2011). [doi:10.5772/23866].

48 M. T. McCann, D. G. Mixon, M. C. Fickus, et al., "Images as occlusions of textures: A framework for segmentation," IEEE Transactions on Image Processing 23(5), 2033–2046 (2014). [doi:10.1109/TIP.2014.2307475].

49 T. Ojala, T. Maenpaa, M. Pietikainen, et al., "Outex-new framework for empirical evaluation of texture analysis algorithms," in Object recognition supported by user interaction for service robots, 1, 701–706, IEEE (2002). [doi:10.1109/ICPR.2002.1044854].

50 J. Pont-Tuset and F. Marques, "Measures and meta-measures for the supervised evaluation of image segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2131–2138 (2013). [doi:10.1109/CVPR.2013.277].

51 D. Martin, C. Fowlkes, D. Tal, et al., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, 2, 416–423, IEEE (2001). [doi:10.1109/ICCV.2001.937655].

52 X. Jia, T. Lei, X. Du, et al., "Robust self-sparse fuzzy clustering for image segmentation," IEEE Access 8, 146182–146195 (2020). [doi:10.1109/ACCESS.2020.3015270].

Abderezak Salmi has received the Master degree in Electrical Engineering in 2011 from the Mouloud Mammeri University of Tizi-Ouzou, Algeria. He is currently pursuing the Ph.D. degree in Automatic Control in the Electrical Engineering and Computer Science Faculty at the Mouloud Mammeri University of Tizi-Ouzou, Algeria. His research interests include image processing and machine learning.

Kamal Hammouche received his Engineering degree in Electronics in 1989, his M.S. (Engineer) degree in Industrial Automatic Control in 1996 and his Ph.D. in Automatic Control from the Mouloud Mammeri University of Tizi-Ouzou, Algeria, in 2007, respectively. He is currently a Full Professor in the Automatic Control Department of the Electrical Engineering and Computer Science Faculty at the Mouloud Mammeri University of Tizi-Ouzou, Algeria. His research interests are in the area of image processing and machine learning.

Ludovic Macaire received his M.S. (Engineer) degree in Computer Science from the UTC Engineering School of Compiègne, France, in 1988 and his Ph.D. in Computer Science and Control from the University of Lille 1 in 1992. He is presently a Full Professor in the CRIStAL Laboratory of the University of Lille and in the ANR-11-EQPX-23 IrDIVE platform. His research interests include color representation, color image analysis applied to segmentation and retrieval.

List of Figures

List of Tables