



**HAL**  
open science

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

Abderezak Salmi, Kamal Hammouche, Ludovic Macaire

► **To cite this version:**

Abderezak Salmi, Kamal Hammouche, Ludovic Macaire. Constrained feature selection for semisupervised color-texture image segmentation using spectral clustering. *Journal of Electronic Imaging*, 2021, 3 (1), pp.013014. 10.1117/1.JEI.30.1.013014 . hal-03148319

**HAL Id: hal-03148319**

**<https://hal.science/hal-03148319>**

Submitted on 19 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Abderezak Salmi<sup>a</sup>, Kamal Hammouche<sup>a</sup>, Ludovic Macaire<sup>b,\*</sup>

<sup>a</sup>Université Mouloud Mammeri, Laboratoire Vision Artificielle et Automatique des Systèmes (LVAAS), Tizi-Ouzou, Algeria

<sup>b</sup>Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISAL, F-59000 Lille, France

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

12 Keywords: Color texture segmentation, pairwise constraints, constrained feature selection, constrained spec-  
13 tral clustering.

14 \*Ludovic Macaire, [ludovic.macaire@univ-lille.fr](mailto:ludovic.macaire@univ-lille.fr)

## 15 1 Introduction

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

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

12 To overcome problems caused by supervised and unsupervised algorithms, weakly super-  
13 vised or semi-supervised segmentation algorithms use limited amounts of supervised infor-  
14 mation. The prior knowledge is then expressed by a few labeled pixels or by a few links  
15 between pixels. A must-link constraint specifies that two pixels must belong to the same  
16 class, whereas a cannot-link constraint specifies that two pixels must belong to different  
17 classes. These pairwise constraints can be obtained from the available labeled pixels. Con-  
18 versely, the labels of pixels cannot be deduced from the pairwise constraints. The labeled  
19 pixels or the pairwise constraints can be interactively given by a user. Semi-supervised al-  
20 gorithms aim to classify pixels by propagating the labels<sup>9</sup> or by incorporating the pairwise  
21 constraints into popular clustering algorithms such as k-means, estimation-maximization,  
22 hierarchical clustering<sup>10</sup> and spectral clustering.<sup>11</sup>

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

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

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

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

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

- 10 - Pairwise must-link and cannot-link constraints are involved in feature selection to  
11 improve the color-texture image segmentation performance using spectral clustering.
- 12 - A new similarity-based constraint score is proposed to select the optimal number of  
13 relevant features.

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

## 19 2 Related work

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

## 1 2.1 Spectral clustering

2 Let  $X$  be the set of  $n$  sample pixels represented by the matrix  $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times d}$ ,  
3 where  $x_i = [x_{i1}, x_{i2}, \dots, x_{ir}, \dots, x_{id}]^T \in \mathbb{R}^d$  represents the  $i$ -th pixel of  $X$ , and  $x_{ir}$ , ( $r =$   
4  $1, \dots, d$ ), the  $r$ -th feature value of the  $i$ -th pixel. Let  $F_d = \{f_1, f_2, \dots, f_r, \dots, f_d\}$  be the set of  
5  $d$  features that is represented by  $X^T = [f_1, f_2, \dots, f_r, \dots, f_d]$ , where  $f_r = [x_{1r}, x_{2r}, \dots, x_{nr}] \in \mathbb{R}^n$   
6 is the  $r$ -th feature vector.

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

$$w_{ij} = \exp \left( - \frac{\delta^2(x_i, x_j)}{2\sigma^2} \right) \quad i, j = 1, 2, \dots, n \quad (1)$$

14 where  $\sigma$  is a scaling parameter and  $\delta(x_i, x_j)$  is the Euclidean distance between the two pixels  
15  $x_i$  and  $x_j$ .

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

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

## 5 2.2 Constraint spectral clustering

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

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

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

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

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

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

2 A straightforward way to integrate must-link and cannot-link pairs in spectral clustering  
 3 is computing the similarity matrix  $W^{M,C}$ .<sup>10,12</sup> For a given must-link pair  $(x_i, x_j) \in M$ ,  $w_{ij}^{M,C}$   
 4 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  
 5 matrix  $W^{M,C}$  is defined as follows:

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

6  
 7 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  $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times d}$ , number  $k$  of desired clusters, set  $M$  of must-link constraints, and set  $C$  of cannot-link constraints.

1. Compute the similarity matrix  $W^{M,C}$  defined by Eq. (4).
2. Compute the Laplacian matrix  $L_{Sym}^{M,C}$  associated to  $W^{M,C}$ .
3. Compute the spectrum of  $L_{Sym}^{M,C}$ , and extract  $k$  dominant eigenvectors  $U^{M,C} = [u_1, \dots, u_k] \in \mathbb{R}^{n \times k}$ .
4. Construct the matrix  $T^{M,C} \in \mathbb{R}^{n \times k}$  from  $U^{M,C}$  by normalizing each row of  $U^{M,C}$  to have unit length

$$t_{ij} = \frac{u_{ij}}{\sqrt{\sum_{j=1}^k u_{ij}^2}}$$

5. Consider each row of  $T^{M,C}$  as a point, and cluster all points into  $k$  clusters by k-means algorithm.

Output:  $k$  clusters of dataset  $X$ .

---

8  
 9 The performance achieved by constrained spectral clustering strongly depends on the  
 10 pairwise similarities between the unconstrained pixels, which are based on the Euclidean

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

### 7 2.3 State-of-the-art about semi-supervised constraint scores

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

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

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

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

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

2

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

3 Zhao et al.<sup>26</sup> introduced the semi-supervised constraint score  $C_r^1$ , which combines the  
4 similarity matrix  $W^C$  constructed from the cannot-link constraints (Eq. (7)) and the similar-  
5 ity matrix  $W^{\text{KNN}1} \in \mathbb{R}^{n \times n}$ , which is built from the set of must-link constraints and unlabeled  
6 sample pixels as follows:

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

7 where  $\gamma$  is a constant parameter. In our experiments,  $\gamma$  is set to 100 and  $K$  is set to 5 as in  
8 Ref. 26.

9 The constraint score  $C_r^1$  is defined as

$$C_r^1 = \frac{\mathbf{f}_r^T \mathbf{L}^{\text{KNN}1} \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^C \mathbf{f}_r} \quad (9)$$

10 where  $\mathbf{L}^{\text{KNN}1} = \mathbf{D}^{\text{KNN}1} - \mathbf{W}^{\text{KNN}1}$  is the un-normalized Laplacian matrix of  $W^{\text{KNN}1}$ , and  $\mathbf{D}^{\text{KNN}1}$   
11 is the degree matrix computed from  $W^{\text{KNN}1}$ .

1 Kalakech et al.<sup>27</sup> proposed the semi-supervised constraint score  $C_r^2$  that is less sensitive  
 2 to the constraint sets

$$C_r^2 = \frac{\tilde{\mathbf{f}}_r^T \mathbf{L} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T \mathbf{D} \tilde{\mathbf{f}}_r} \cdot \frac{\mathbf{f}_r^T \mathbf{L}^M \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^C \mathbf{f}_r} \quad (10)$$

3 where  $\bar{\mathbf{f}}_r = \frac{\sum_{i=1}^n x_i r_{di}}{\sum_{i=1}^n d_{ii}} = \frac{\mathbf{f}_r^T \mathbf{D} \mathbf{1}}{\mathbf{1}^T \mathbf{D} \mathbf{1}}$ ,  $\tilde{\mathbf{f}}_r = \mathbf{f}_r - \bar{\mathbf{f}}_r$ , and  $\mathbf{1} = [1, \dots, 1]^T$ .  $\mathbf{L}$  and  $\mathbf{D}$  are deduced from the  
 4 similarity matrix  $\mathbf{W}$  (Eq. (1)).  $\mathbf{L}^M = \mathbf{D}^M - \mathbf{W}^M$  and  $\mathbf{D}^M = \sum_{j=1}^n w_{ij}^M$ .

5 Benabdeslem and Hindawi proposed another constraint score  $C_r^3$ , which combines the  
 6 similarity matrices  $\mathbf{W}^C$  and  $\mathbf{W}^{\text{KNN2}}$ .<sup>28</sup> The similarity matrix  $\mathbf{W}^{\text{KNN2}}$  is expressed as

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

7 The score  $C_r^3$  is defined as follows:<sup>28</sup>

$$C_r^3 = \frac{\mathbf{f}_r^T \mathbf{L}^{\text{KNN2}} \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^C \mathbf{D}^{\text{KNN2}} \mathbf{f}_r} \quad (12)$$

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

$$C_r^4 = \frac{\mathbf{f}_r^T \mathbf{L}^{\text{KNN3}} \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^C \mathbf{D}^{\text{KNN}} \mathbf{f}_r} \quad (13)$$

10 where the diagonal matrix  $\mathbf{D}^{\text{KNN}}$  is deduced from the similarity matrix  $\mathbf{W}^{\text{KNN}}$  (see Eq. (5)).

11 The Laplacian matrix  $\mathbf{L}^{\text{KNN3}}$  is computed from the similarity matrix  $\mathbf{W}^{\text{KNN3}}$ , which is defined

1 as follows:

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

2 Recently, Yang et al.<sup>30,31</sup> introduced the new semi-supervised constraint score  $C_r^5$ , which  
 3 takes advantage of the local geometrical structure of unlabeled data samples as well as the  
 4 constraint information deduced from prototypes as follows:

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

5 The similarity matrix  $\mathbf{W}^{\text{KNN4}}$  is expressed as

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

6 where  $\gamma$  and  $\lambda$  are the parameters set to the empirical values of 0.9 and 0.5, respectively.<sup>30,31</sup>

7 The cells of  $\mathbf{W}^P$  are set to  $1/|X^l|$  when two data samples are prototypes that belong to the

1 same class and to 0, otherwise

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

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

4

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

### 9 3 Proposed color-texture image segmentation

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

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

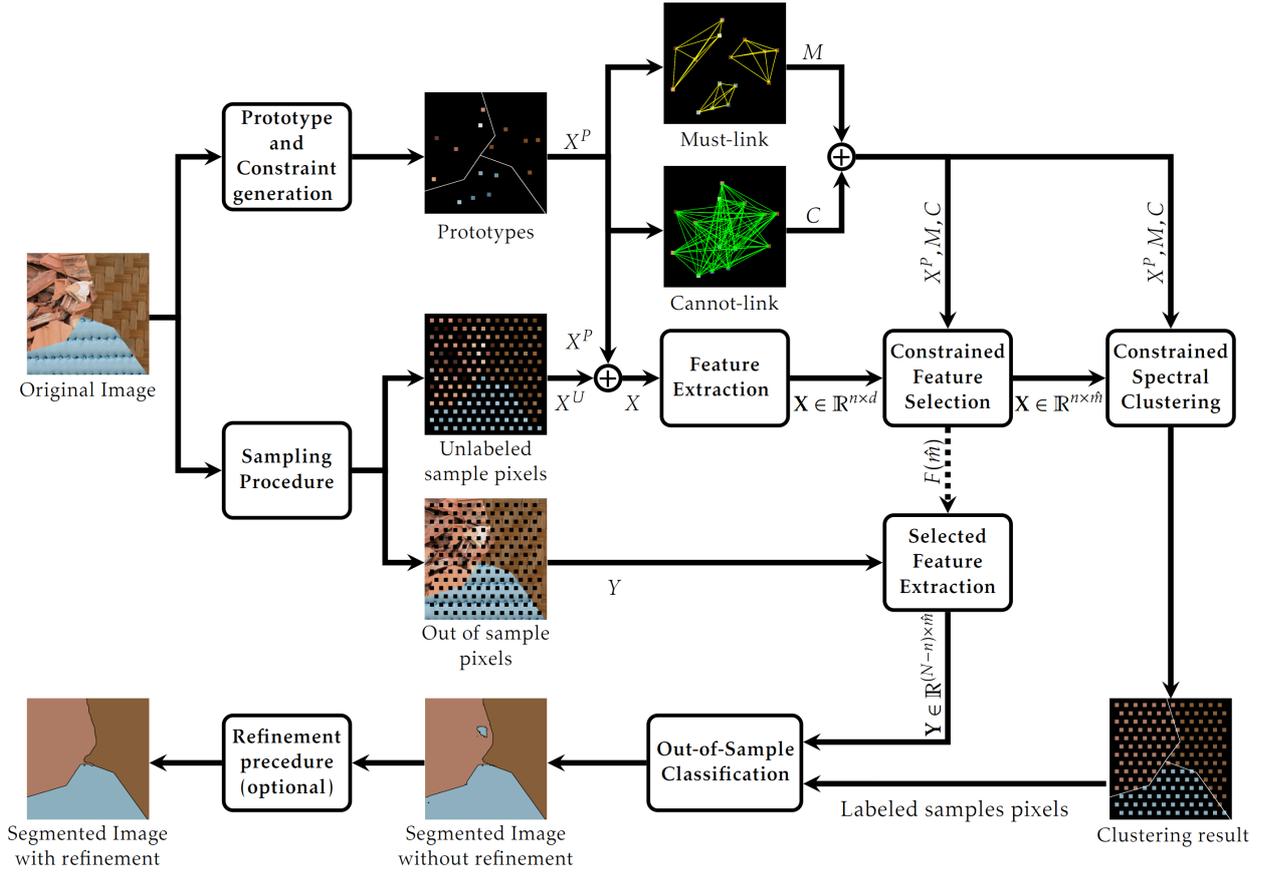


Fig 1 Flowchart of the proposed color-texture image segmentation (CFS-SC).

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

### 3 3.1 Pixel sampling procedure

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

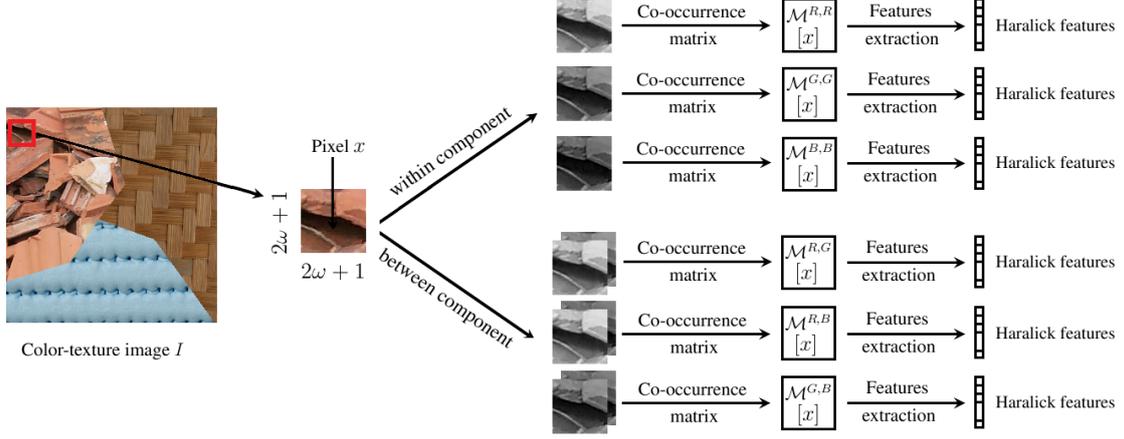


Fig 2 Illustration of local Haralick texture feature computing.

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

### 3 3.2 Color and texture features

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

13 Gabor features are powerful texture descriptors generated from the convolution of Ga-  
 14 bor filters. The impulse response of a Gabor filter with specific parameters (scale  $\gamma$  and

1 orientation  $\theta$ ) depends on spatial coordinates  $(u, v)$  and is expressed as

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

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

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

15 Finally each pixel is characterized by a set of  $d = 180$  features (84 Haralick texture  
16 features, 84 Gabor texture features, and 12 color features)  $F_m = \{f_1, \dots, f_d\}$ . Each of the  
17  $d = 180$  pixel features is normalized between 0 and 1 to ensure that the scales of all features  
18 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.<sup>25</sup> 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.<sup>29</sup> 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  $F_m = \{f_1, \dots, f_m\}$  ( $m = 1, 2, \dots, d$ ) 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  $\hat{W}^{SS}$ , which is defined from the given constraints, and the similarity matrix  $W(F_m)$  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:

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

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

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

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

$$w_{ij}(F_m) = \exp \left( - \frac{\delta^2(x_i^{(m)}, x_j^{(m)})}{2\sigma^2} \right) \quad i, j = 1, 2, \dots, n \quad (21)$$

4 where  $\delta(x_i^{(m)}, x_j^{(m)})$  is the Euclidean distance between two sample pixels  $x_i$  and  $x_j$  character-  
 5 ized by the subset of  $m$  features  $F_m$ .

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

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

9 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  
 10 must-link pairs that results from the propagation of prototype subsets  $X^l$ ,  $l = 1, \dots, k$  to  
 11 unlabeled sample pixels

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

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

$$NP(x_i) = \arg \min_{z \in \bigcup_{l=1, \dots, k} X^l} \left( \delta^2(x_i, z) \right). \quad (24)$$

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

### 11 3.3.2 Feature selection procedure

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

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

---

Algorithm 2 Feature selection procedure.

---

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

1. Create empty set of features  $F_0 = \{\emptyset\}$ .
2. For  $m = 1$  to  $d$ 
  - a. Select the most relevant feature  $f_r^+$ 

$$f_r^+ = \arg \min_{f_r \in F_d \setminus F_{m-1}} \left( \varepsilon^{SS}(F_{m-1} \cup \{f_r\}) \right).$$
  - b. Update  $F_m = F_{m-1} \cup \{f_r^+\}$ .
3. Select the number  $\hat{m}$  of features such that
 
$$\hat{m} = \arg \min_{m=1,2,\dots,d} \left( \varepsilon^{SS}(F_m) \right).$$

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

---

### 3 3.4 Constrained spectral clustering

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

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

### 9 3.5 Out-of-sample classification

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

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

5 For this purpose, we propose to embed out-of-sample pixels  $y_i$  onto the  $k$ -dimensional  
 6 subspace that is formed by the  $k$ -dominant eigenvectors  $u_l$ , which correspond to the  $k$  smallest  
 7 eigenvalues  $\lambda_l$  of the Laplacian matrix  $L_{Sym}^{M,C}$  computed on the set  $X$  of sample pixels.<sup>37</sup> The  
 8 projection of out-of-sample pixel  $y_i$  on the  $l^{th}$   $k$ -dominant eigenvector is defined as follows:

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

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

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

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

$$w^{(\hat{m})}(y_i, x_j) = \exp \left( - \frac{\delta^2(y_i^{(\hat{m})}, x_j^{(\hat{m})})}{2\sigma^2} \right) \quad (28)$$

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

1 the unit length

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

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

4  
5 As in Ref. 2, a refinement procedure can be applied to further improve the segmentation  
6 accuracy. This procedure consists of keeping all regions larger than 0.5% of  $N$  while all other  
7 regions are relabeled with the class index corresponding to the largest adjacent region.  
8 Finally, all steps of the proposed CFS-SC method for color-texture image segmentation are  
9 summarized in Algorithm 3.

## 10 4 Experimental results

11 In this section, we evaluate our CFS-SC method for color-texture image segmentation. We  
12 first evaluate the performance of the proposed semi-supervised feature score  $\varepsilon^{SS}$ . **For this**  
13 **purpose, we consider two color-texture images from the Prague texture segmentation bench-**  
14 **mark.**<sup>38</sup> Next, we present segmentation results obtained by the proposed algorithm and we  
15 compare them with those obtained by several state-of-the-art algorithms **on four benchmark**  
16 **datasets of color and grayscale textures.**

### 17 4.1 Evaluation of the semi-supervised constrained feature selection

18 To evaluate the performance of the proposed constraint score  $\varepsilon^{SS}$ , we compare it with the  
19 state-of-the-art constraint scores  $C^1$ ,  $C^2$ ,  $C^3$ ,  $C^4$ , and  $C^5$  (see section 2.3). The scaling  
20 parameter  $\sigma$  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 ( $\omega$ ), Gabor parameters ( $\theta, \gamma$ );  
- 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  $W^{M,C}(F_{\hat{m}}) \in \mathbb{R}^{n \times n}$  from  $X$  using Eq. (25).
6. Compute the normalized Laplacian matrix  $L_{Sym}^{M,C}$  associated to  $W^{M,C}(F_{\hat{m}})$ .
7. Compute the spectrum of  $L_{sym}^{M,C}$ , and find  $k$  dominant eigenvectors  $U^{M,C} = [u_1, \dots, u_k] \in \mathbb{R}^{n \times k}$  corresponding to  $k$  smallest eigenvalues  $\lambda^{M,C} = [\lambda_1, \dots, \lambda_k] \in \mathbb{R}^k$ .
8. Construct the matrix  $T^{M,C} \in \mathbb{R}^{n \times k}$  from  $U^{M,C}$  by normalizing each row of  $U^{M,C}$  to have unit length

$$t_{ij} = \frac{u_{ij}}{\sqrt{\sum_{j=1}^k u_{ij}^2}}$$

9. Handle each row of  $T^{M,C}$  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  $\tilde{W}$  defined by Eq. (28).
  - Compute the embedded point  $z(y_i)$  defined by Eq. (26).
  - Compute the normalized embedded point  $\tilde{z}(y_i)$  using Eq. (29).
  - Assign  $y_i$  to the cluster of the nearest sample point, among  $T^{M,C}$ , of  $\tilde{z}(y_i)$ .
11. Refinement procedure (optional).

Output: Segmented image.

---

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

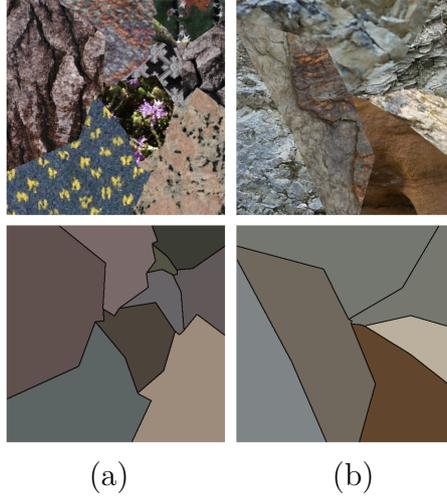


Fig 3 Two test color-texture images from the Prague dataset (first row) and their corresponding ground truth images (bottom row): (a) image 7\_1\_1 and (b) image 17\_1\_1.

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

#### 9 4.1.1 Constraint scores with respect to number of features

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

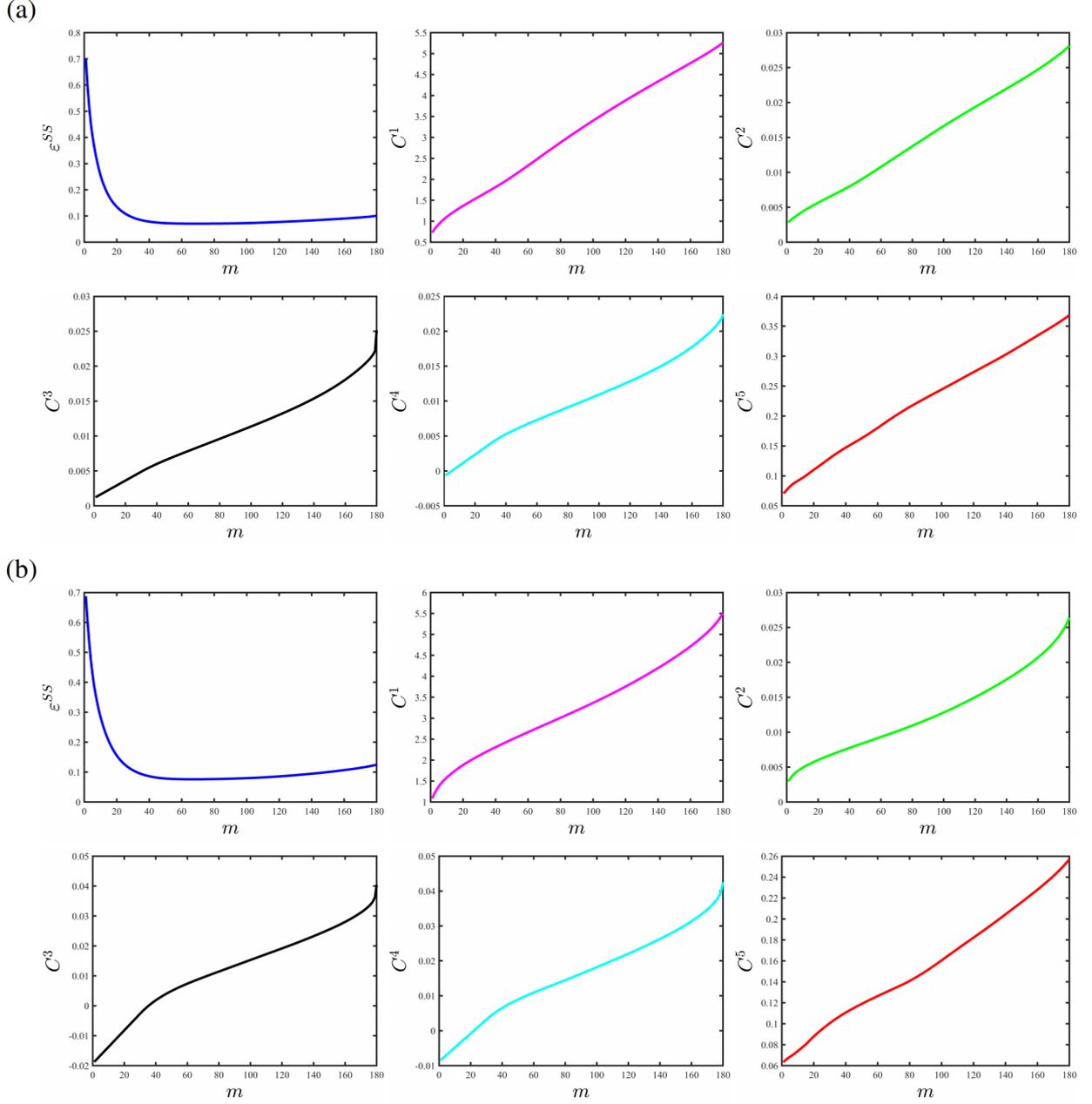


Fig 4 Constraint scores versus number  $m$  of selected features for the two test images: (a) image 7\_1\_1 and (b) image 17\_1\_1.

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

#### 1 4.1.2 Accuracy of unlabeled pixels

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

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

16 Accuracy versus number  $n$  of sample pixels. The number  $n$  of sample pixels is related to the  
17 spatial sampling of the image. On the one hand,  $n$  must be large enough to reach a satisfying  
18 spatial definition of the detected regions. On other hand,  $n$  must be small enough to reduce  
19 the computational complexity of the constrained spectral clustering of the unlabeled sample  
20 pixels. To satisfy a trade-off between the spatial definition and complexity, the  $n$  range is  
21  $[200 - 400]$ , which corresponds on average to 0.10% of the number  $N$  of pixels. To show

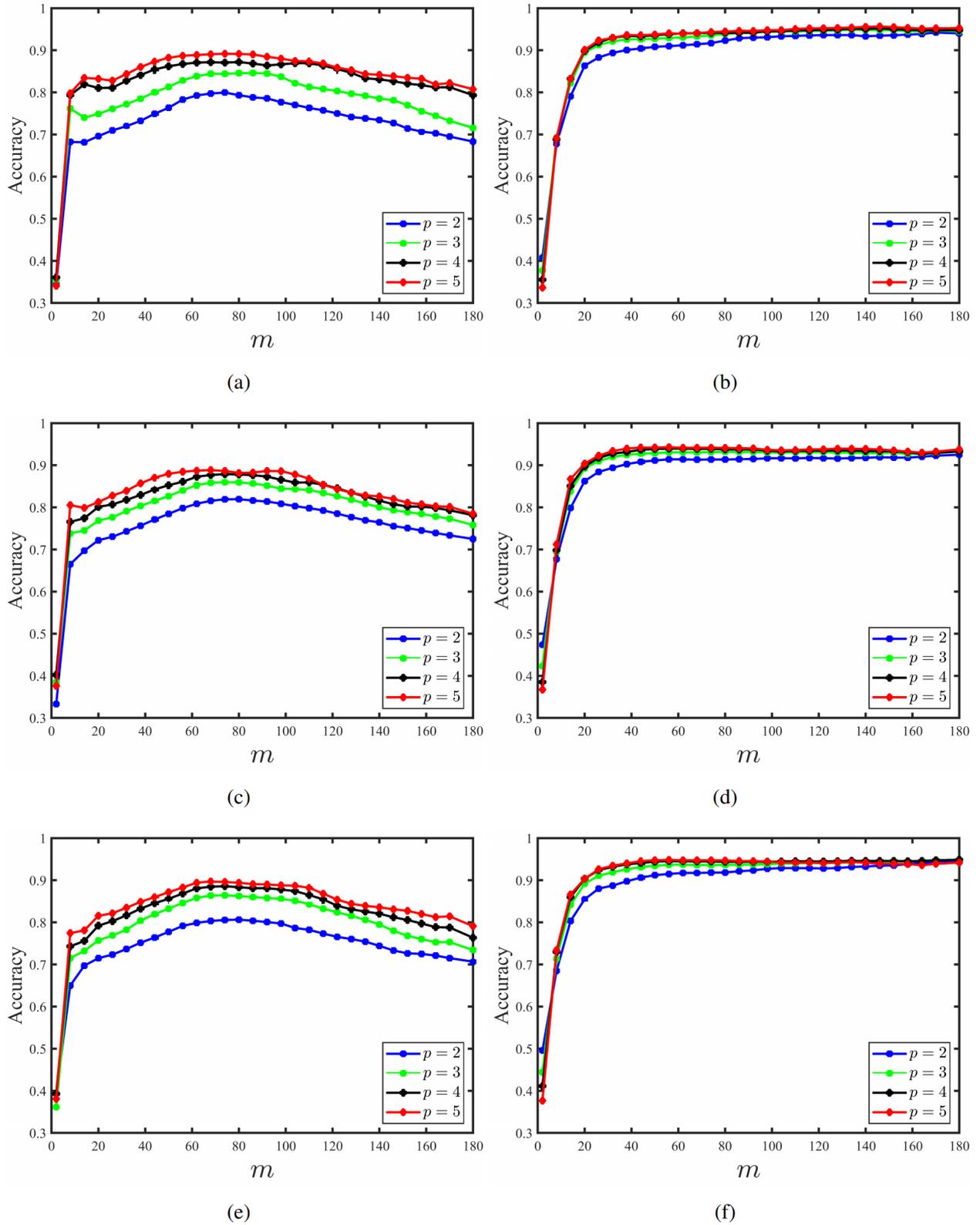


Fig 5 Accuracy versus number  $m$  of features obtained using the proposed semi-supervised constraint scores on test images for different numbers of  $n$  and  $p$ . Left column: image 7\_1\_1 and right column: image 17\_1\_1. (a)-(b)  $n = 200$ , (c)-(d)  $n = 300$ , and (e)-(f)  $n = 400$ .

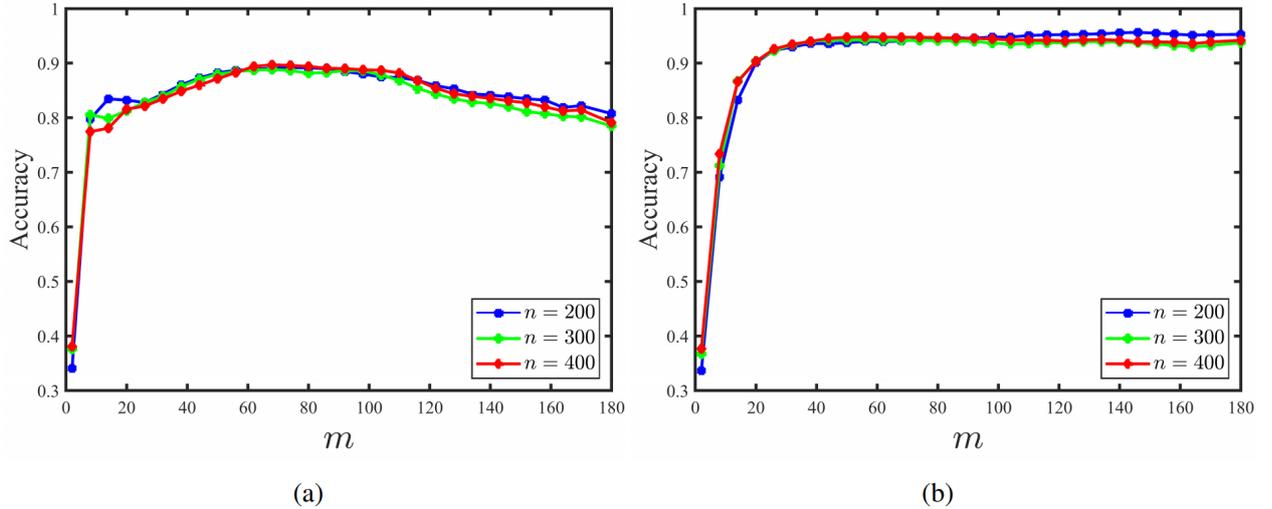


Fig 6 Accuracy versus number  $m$  of features obtained using the proposed semi-supervised constraint score on the two test images for different numbers  $n$  of sample pixels: (a) image 7\_1\_1 and (b) image 17\_1\_1.

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

#### 5 4.1.3 Optimal number of selected features

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

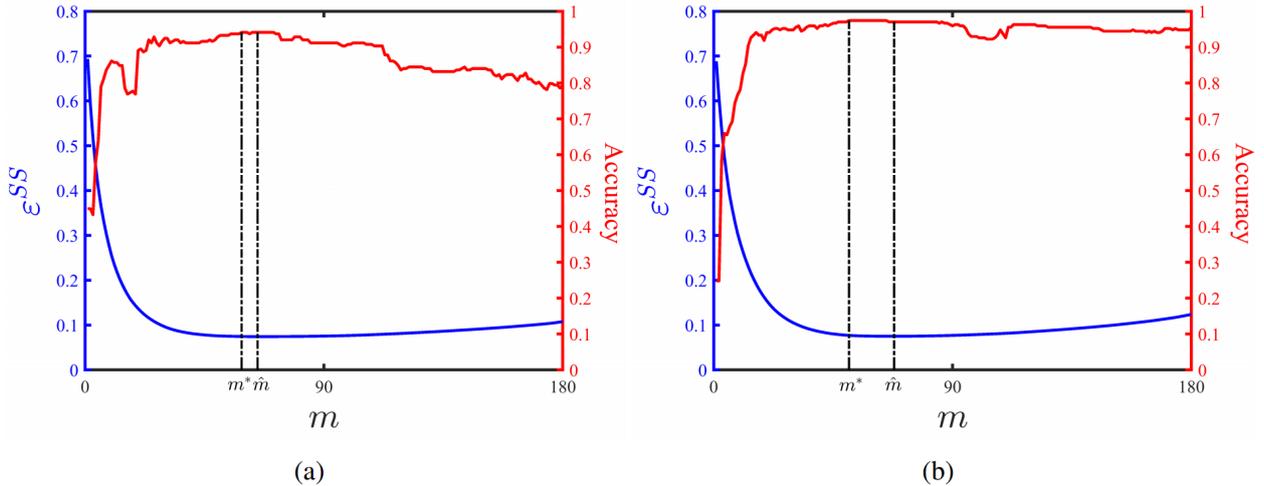


Fig 7 Accuracy-score  $\varepsilon^{SS}$  versus number  $m$  of selected features for the test images: (a) image 7\_1\_1 and (b) image 17\_1\_1.

#### 1 4.1.4 Comparison with the state-of-the-art constraint scores

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

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

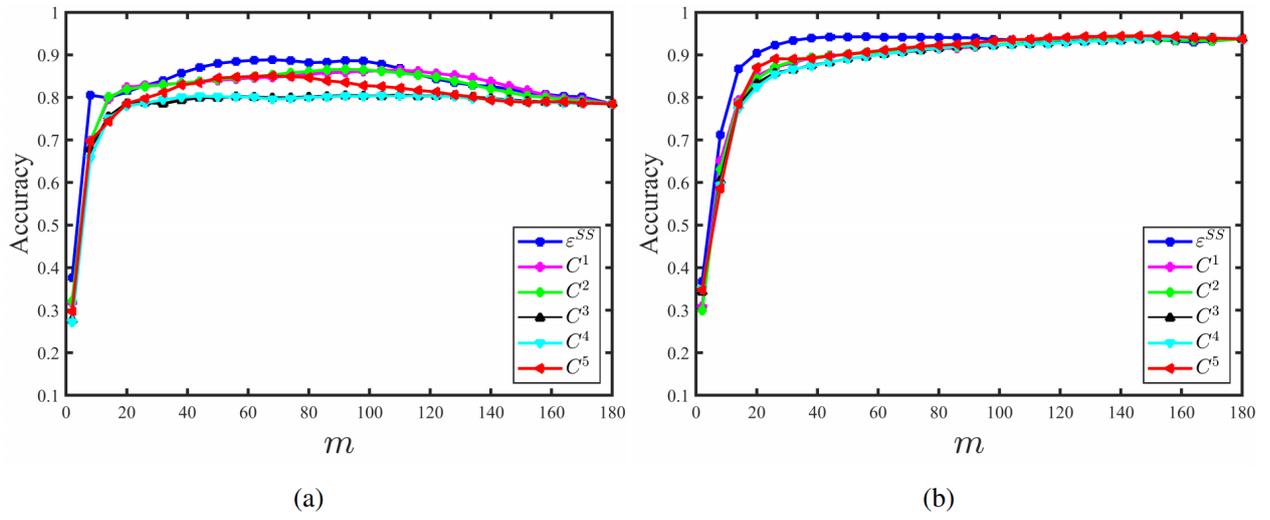


Fig 8 Accuracy versus number  $m$  of features for the two test images: (a) image 7\_1\_1 and (b) image 17\_1\_1.

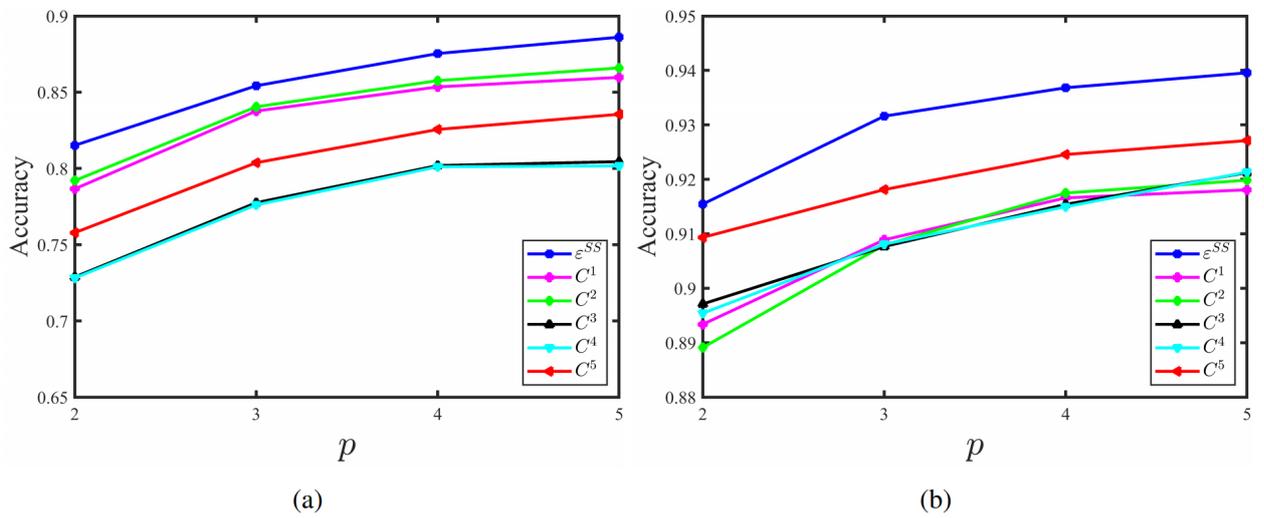


Fig 9 Accuracy versus number  $p$  of prototypes when  $m = 90$  features have been selected and  $n = 300$  on the two test images: (a) image 7\_1\_1 and (b) image 17\_1\_1.

## 1 4.2 Segmentation results on Prague dataset

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

8 The assessment of segmentation performance is based on the conventional measures pro-  
9 vided by the Prague texture segmentation website,<sup>39</sup> which include:  
10 region-based criteria: correct segmentation (CS), over-segmentation (OS), under-segmentation  
11 (US), missed error (ME), noise error (NE);  
12 pixel-wise based criteria: omission error (O), commission error (C), class accuracy (CA), re-  
13 call (CO), precision (CC), type I error (I.), type II error (II.), mean class accuracy estimate  
14 (EA), mapping score (MS), root mean square proportion estimation error (RM), comparison  
15 index (CI);  
16 consistency-error criteria: global consistency error (GCE) and local consistency error (LCE);  
17 clustering comparison criteria: Mirkin metric (dM), Van Dongen metric (dD), variation of  
18 information (dVI).

19 The best score is displayed as bold in each table.

20 The methods involved in comparison include:

21 Six unsupervised algorithms: 1) the texNcut algorithm that is based on spectral cluster-  
22 ing algorithm<sup>5</sup> and which uses texture features computed on super-pixels, 2) the variational

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

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

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

22 The results of supervised algorithms (Table 2) were obtained using the normal Prague  
23 dataset, which contains 20 test images.<sup>39</sup> The results of semi-supervised and unsupervised

1 algorithms (Table 1) were obtained over the large Prague dataset (80 test images), which  
2 includes the 20 test images of the normal Prague dataset.<sup>39</sup>

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

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

#### 13 4.2.1 Comparison with unsupervised algorithms

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

#### 18 4.2.2 Comparison with semi-supervised algorithms

19 To assess the relevance of the proposed semi-supervised feature selection on the segmenta-  
20 tion performance, we compare the segmentation accuracy benchmark results obtained by our  
21 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  $\uparrow$  |  $\downarrow$  denote the required criterion direction. Here, 'nr' means no segmentation refinement.

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

1 Table 1 clearly show the benefit provided by the proposed feature selection on the segmen-  
2 tation. When we compare the semi-supervised methods, we observe that the performance  
3 of the proposed semi-supervised method (CFS-SC), even without refinement, is superior to  
4 that of FSEG<sup>+</sup> and WSSCGP.

Table 2 Results of our method CFS-SC and supervised methods on the normal Prague dataset (20 test images). The arrows  $\uparrow$  |  $\downarrow$  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
$\uparrow$ CS	46.11	52.48	84.57	87.52	96.01	98.45	96.71	94.18	96.45	94.03	94.97
$\downarrow$ OS	0.81	0.00	0.00	0.00	1.56	0.00	1.71	0.00	0.17	0.00	0.00
$\downarrow$ US	4.18	1.94	1.70	0.00	1.20	0.00	0.00	1.18	0.41	0.00	0.00
$\downarrow$ ME	44.82	41.55	9.50	6.70	0.78	0.37	0.68	3.42	1.23	1.63	1.09
$\downarrow$ NE	45.29	40.97	10.22	6.90	0.89	0.46	0.48	3.24	1.12	2.32	1.58
$\downarrow$ O	14.52	20.74	7.00	7.46	2.72	0.93	0.72	3.13	2.75	4.42	3.83
$\downarrow$ C	16.77	22.10	5.34	6.16	2.29	1.04	0.70	1.32	2.39	4.45	3.80
$\uparrow$ CA	65.42	67.01	86.21	87.08	93.95	97.67	95.86	94.53	93.89	91.85	92.61
$\uparrow$ CO	76.19	77.86	92.02	92.61	96.73	98.78	96.91	96.23	96.06	95.49	95.97
$\uparrow$ CC	80.30	78.34	92.68	93.26	97.02	98.81	97.38	97.01	96.41	95.91	96.29
$\downarrow$ I.	23.81	22.14	7.98	7.39	3.27	1.22	3.09	3.77	3.94	4.51	4.03
$\downarrow$ II.	4.82	4.40	1.70	1.49	0.68	0.25	0.41	0.58	0.69	0.79	0.76
$\uparrow$ EA	75.40	76.21	91.72	92.68	96.68	98.77	97.01	96.24	96.08	95.59	96.04
$\uparrow$ MS	64.29	66.79	88.03	88.92	95.10	98.17	95.37	94.35	94.08	93.24	93.96
$\downarrow$ RM	6.43	4.47	2.08	1.38	0.86	0.24	0.61	1.07	0.70	0.79	0.70
$\uparrow$ CI	76.69	77.05	92.02	92.81	96.77	98.78	97.08	96.41	96.15	95.65	96.08
$\downarrow$ GCE	25.79	23.94	11.76	12.54	5.55	2.33	2.13	3.50	4.67	8.07	7.30
$\downarrow$ LCE	20.68	19.69	8.61	9.94	3.75	1.68	1.46	2.47	3.52	6.40	5.75
$\downarrow$ dD	20.35	17.86	7.50	-	3.06	1.21	1.45	2.41	2.59	4.50	4.02
$\downarrow$ dM	13.25	10.62	4.69	-	1.96	0.74	0.77	1.35	1.56	2.76	2.48
$\downarrow$ dVI	14.51	14.22	13.99	-	13.80	13.68	13.68	13.71	13.77	14.01	13.96

### 1 4.2.3 Comparison with supervised algorithms

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

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

8  
9 For a visual comparison, Fig. 10 shows some segmentation results obtained by our CFS-  
10 SC method, WSSCGP, and by four top-performing methods whose segmented images were  
11 available in Ref. 39. For clarity, the black lines highlight the boundaries of segmented tex-  
12 ture regions. Overall, CFS-SC, FCNTsup, and EWT-FCNT approaches provide satisfactory  
13 visual segmentation results that are close to the ground truth.

#### 14 4.3 Segmentation results on Histology dataset

15 The Histology dataset<sup>47,48</sup> contains 36 color images of size  $128 \times 128$  pixels that represent  
16 two types of tissue. The ground truth images were determined by professional pathologists  
17 through visual inspection. We use the same setup as for the Prague texture experiment ex-  
18 cept for the Haralick parameter  $\omega$  that we set to 5. To assess the performance of our methods  
19 CFS-SC and CSC without and with refinement, we compare their results with several meth-  
20 ods including, FSEG<sup>+</sup>, WSSCGP, DLSRC, and FSEG as well as an another unsupervised  
21 method, called occlusion of random textures segmentation method (ORTSEG).<sup>48</sup> The quan-  
22 titative evaluation is based on the same measures used for the Prague dataset. Table 3

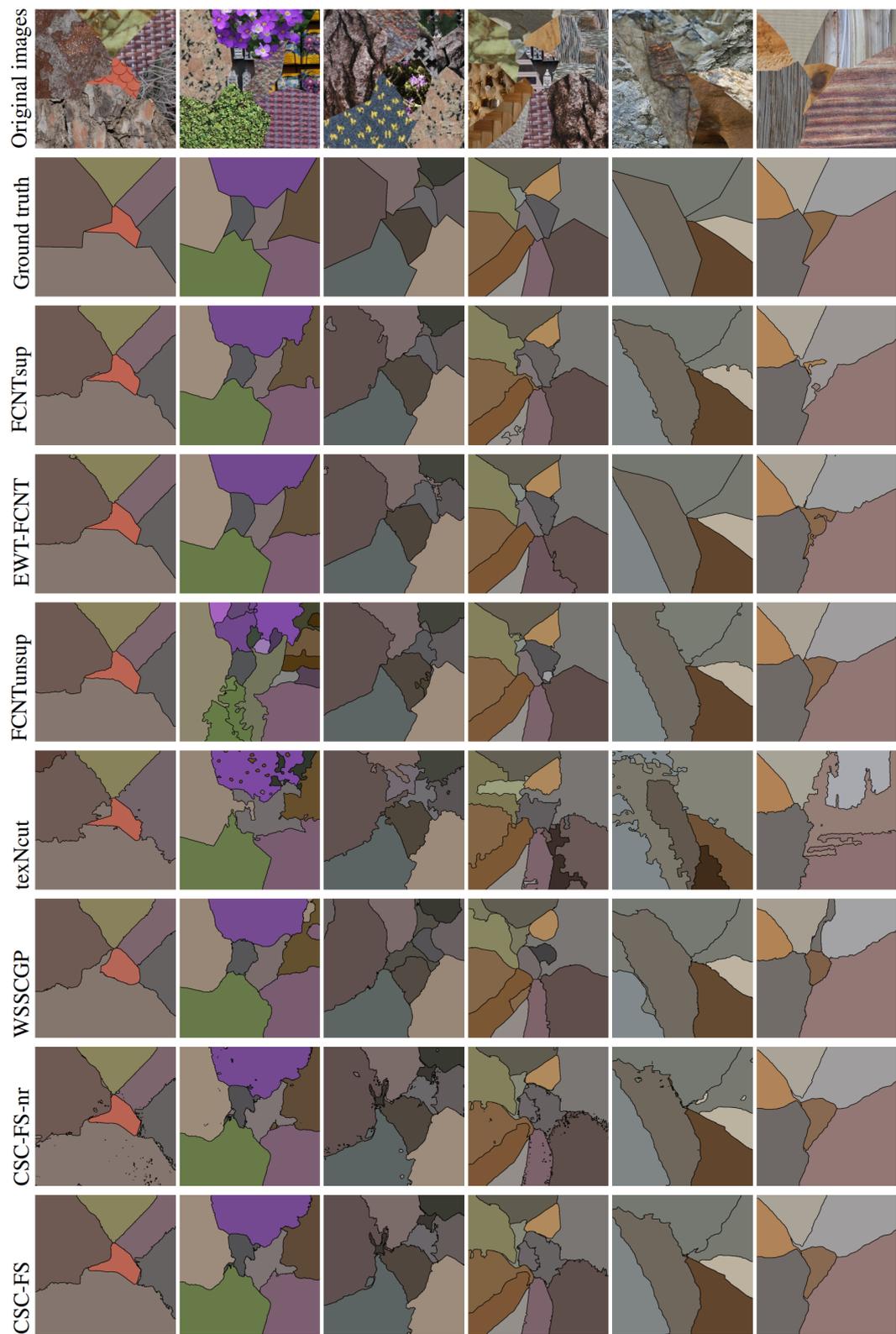


Fig 10 Exemplary segmentation results on the Prague color benchmark. From top to bottom: Original images, ground truth, FCNTsup, **EWT-FCNT**, FCNTunsup, texNeut, **WSSCGP**, CSC-FS-nr, and CSC-FS.

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

Table 3 Results on the Histology dataset. The arrows  $\uparrow$  |  $\downarrow$  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 <sup>+</sup>	WSS CGP	CSC	CFS SC-nr	CFS SC
$\uparrow$ CS	82.98	72.53	38.98	82.06	86.90	82.54	96.93	97.07
$\downarrow$ OS	3.34	1.71	35.95	1.78	3.29	0.00	0.00	0.00
$\downarrow$ US	2.82	2.78	0.00	10.25	1.94	0.00	0.00	0.00
$\downarrow$ ME	9.69	20.09	19.96	7.67	5.95	11.33	0.00	0.00
$\downarrow$ NE	11.49	20.31	21.05	8.25	7.02	11.99	0.00	0.00
$\downarrow$ O	7.64	10.17	24.77	13.71	6.70	8.69	3.26	3.10
$\downarrow$ C	6.96	10.57	24.23	7.42	6.21	9.62	3.47	3.31
$\uparrow$ CA	86.80	81.81	67.17	84.90	87.69	84.71	94.12	94.37
$\uparrow$ CO	91.44	88.84	72.38	90.70	92.07	91.20	96.93	97.07
$\uparrow$ CC	94.96	91.27	92.27	92.02	95.28	91.99	97.00	97.14
$\downarrow$ I.	8.56	11.16	27.62	9.30	7.93	8.80	3.07	2.93
$\downarrow$ II.	5.08	8.98	6.99	11.40	5.07	8.57	3.45	3.27
$\uparrow$ EA	92.60	88.71	78.78	90.66	93.15	91.21	96.94	97.08
$\uparrow$ MS	87.17	83.43	67.63	86.05	88.11	86.8	95.40	95.61
$\downarrow$ RM	6.55	9.93	17.29	7.42	5.82	5.22	1.06	1.13
$\uparrow$ CI	92.89	89.37	80.49	91.00	93.40	91.4	96.95	97.09
$\downarrow$ GCE	10.42	12.36	14.10	9.64	9.73	13.65	5.77	5.52
$\downarrow$ LCE	6.77	7.23	11.24	6.12	6.48	9.68	4.82	4.51
$\downarrow$ dD	7.54	9.73	18.55	7.65	6.90	8.66	3.07	2.93
$\downarrow$ dM	12.88	17.19	26.18	14.00	11.99	14.96	5.90	5.65
$\downarrow$ dVI	5.88	5.64	7.23	5.61	5.85	5.68	5.43	5.42

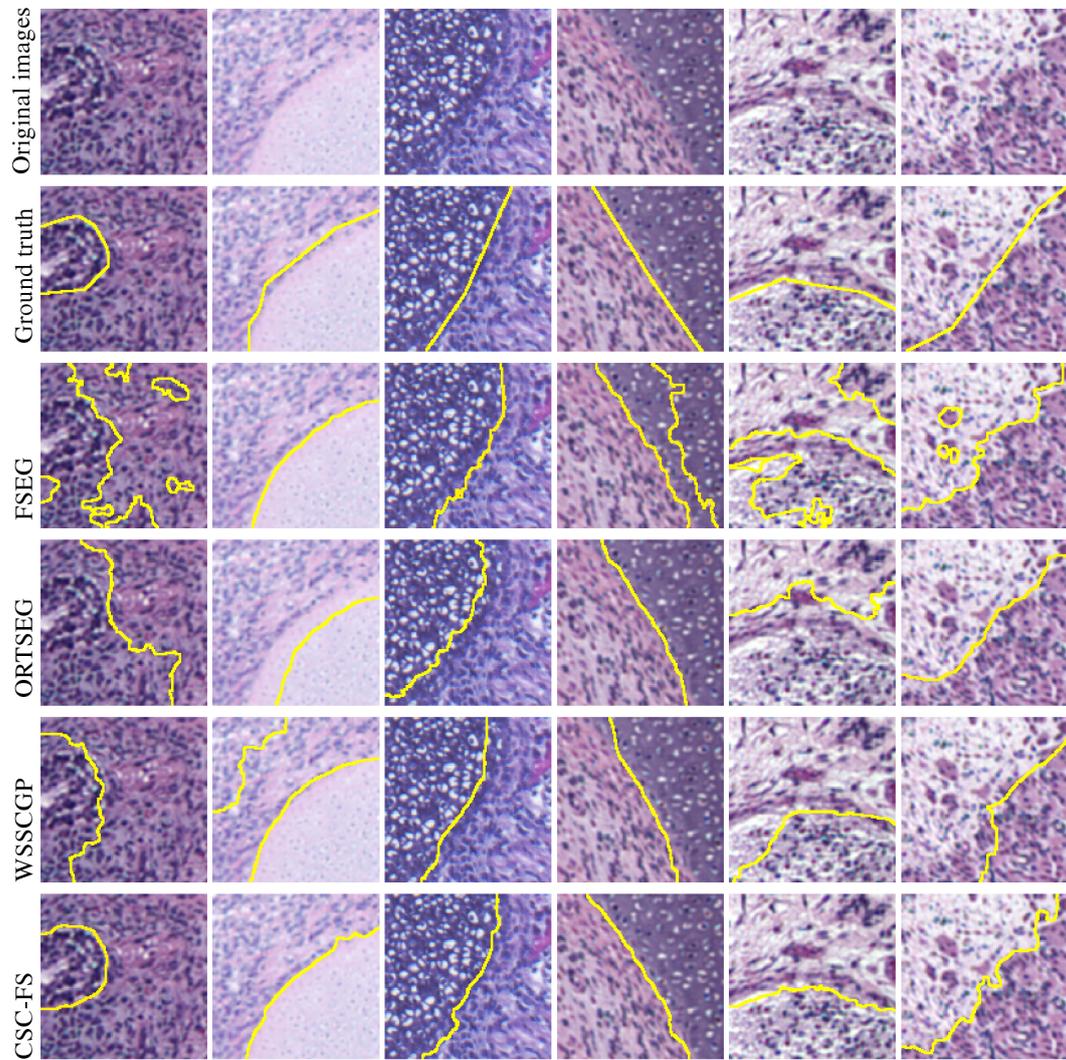


Fig 11 Exemplary segmentation results on the Histology dataset. From top to bottom: Original images, ground truth, FSEG, ORTSEG, WSSCGP, and CSC-FS.

#### 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)<sup>49</sup> 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  $512 \times 512$  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.<sup>49</sup> For convenience, we analyze only the texture features that are computed from the gray level images. The size of neighborhood  $\omega$  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).<sup>50</sup> 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 measures. 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

1 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

## 2 4.5 Segmentation results on Berkeley dataset

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

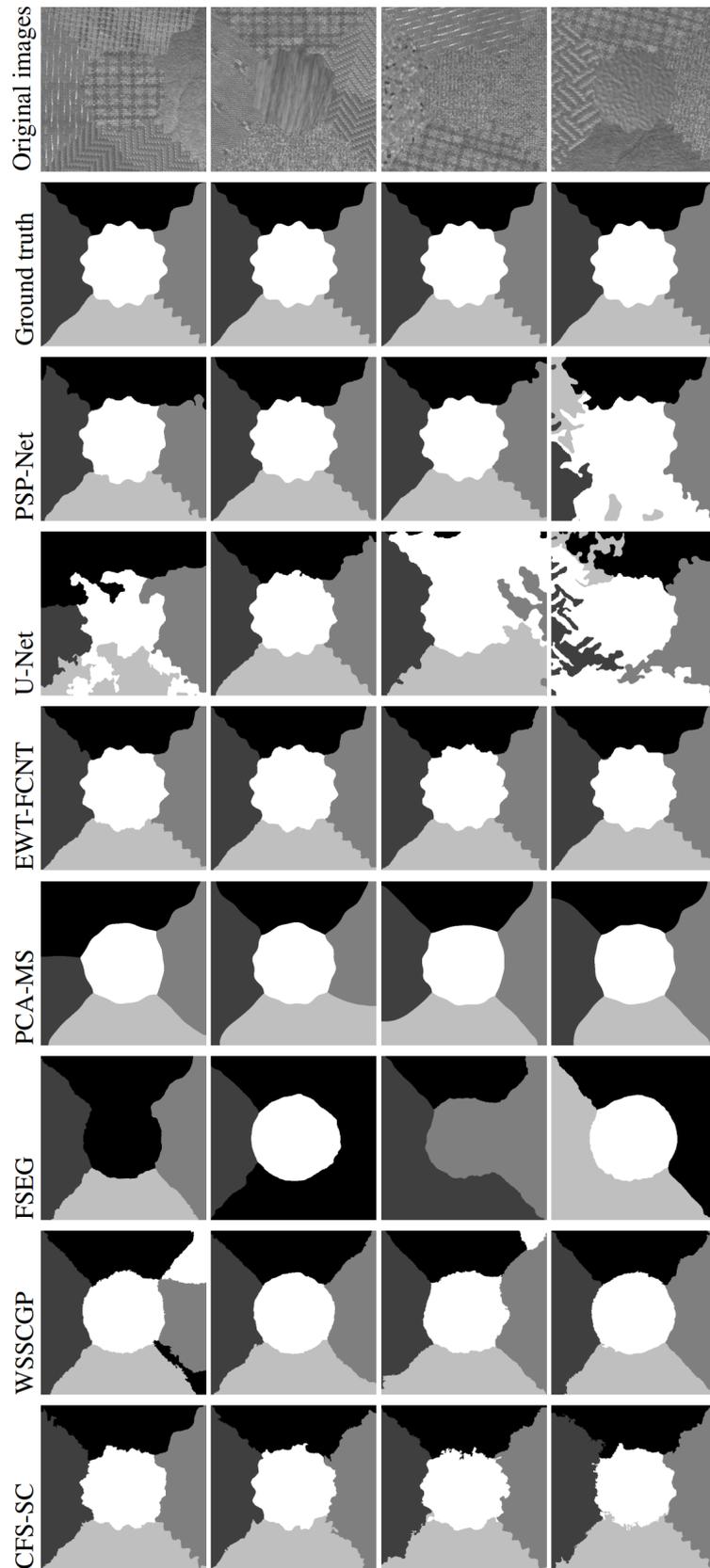


Fig 12 Exemplary segmentation results on the Outex dataset. From top to bottom: Original images, Ground truth, PSP-Net, U-Net, EWT-FCNT, PCA-MS, FSEG, WSSCGP, and CFS-SC.

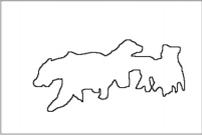
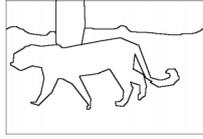
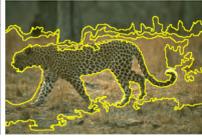
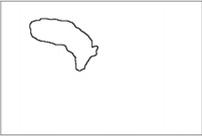
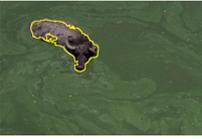
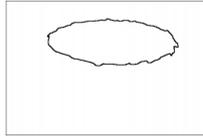
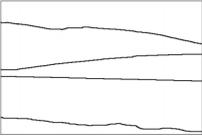
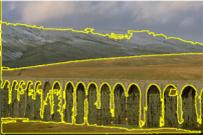
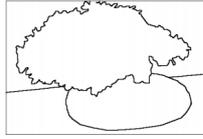
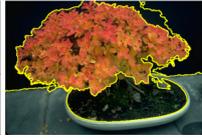
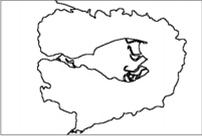
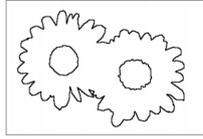
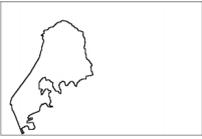
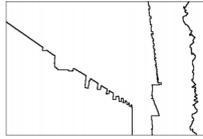
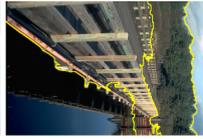
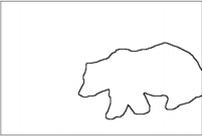
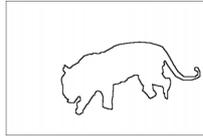
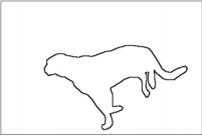
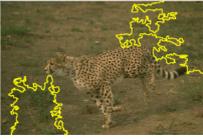
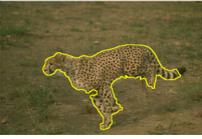
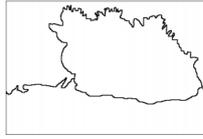
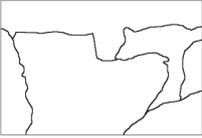
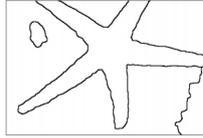
Ground truth	RSSFCA	CFS-SC	Ground truth	RSSFCA	CFS-SC
					
	(87.98, 0.92, 0.79)	(98.05, 0.99, 0.96)		(64.68, 0.56, 0.72)	(93.56, 0.90, 0.93)
					
	(98.91, 0.99, 0.98)	(98.87, 0.99, 0.98)		(99.38, 0.99, 0.99)	(99.34, 0.99, 0.98)
					
	(85.39, 0.86, 0.87)	(95.87, 0.96, 0.96)		(90.10, 0.88, 0.92)	(93.96, 0.93, 0.95)
					
	(70.06, 0.61, 0.81)	(94.66, 0.93, 0.94)		(80.12, 0.82, 0.68)	(97.36, 0.97, 0.96)
					
	(58.12, 0.68, 0.51)	(97.90, 0.99, 0.96)		(68.66, 0.66, 0.75)	(95.57, 0.96, 0.95)
					
	(67.87, 0.74, 0.56)	(95.62, 0.97, 0.92)		(95.23, 0.97, 0.91)	(97.44, 0.98, 0.95)
					
	(64.23, 0.78, 0.54)	(97.54, 0.98, 0.95)		(78.18, 0.78, 0.79)	(96.58, 0.97, 0.95)
					
	(42.66, -, 0.68)	(93.50, 0.92, 0.95)		(68.36, 0.59, 0.72)	(94.83, 0.90, 0.92)

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.

## 1 4.6 Processing time

2 To provide an overview on the computational requirements of the proposed CFS-SC algo-  
3 rithm, we estimated its processing time on an Intel Core i7 3.60GHz computer with 8GB  
4 RAM. Table 5 displays the average computing time of FSEG, WSSCGP, and CFS-SC for the  
5 tree datasets, i.e. according to size and type (color or grayscale) of image. The processing  
6 time of these methods can be divided into two parts: the processing times consumed by fea-  
7 ture extraction step and by the other steps (feature selection and pixel classification steps).  
8 From this table, we can see that the processing time required by the CFS-SC algorithm is  
9 the highest. For example, it was about 518 seconds for a  $(512 \times 512)$  image. This relatively  
10 long processing time led us revisit each step of our algorithm. The majority of this time was  
11 consumed by the feature extraction step (486 seconds) and more precisely by the Haralick  
12 feature extraction (403 seconds). The remaining time (34 seconds) was mainly shared by the  
13 three basic procedures of our CFS-SC algorithm; namely, the constrained feature selection  
14 (20.7 seconds), the constrained spectral clustering (0.033 seconds), and the classification of  
15 out-of-sample pixels (11.01 seconds). Apart from the feature extraction step, the algorithm  
16 runs relatively quickly. It should be emphasized that it is always possible to reduce the  
17 computational time for feature extraction step and that the computational time cannot be  
18 considered as a crucial drawback, given the appreciable segmentation results achieved by the  
19 CFS-SC algorithm on all the tested datasets.

## 20 5 Conclusion

21 In this paper, we proposed an efficient semi-supervised approach for color-texture image  
22 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 ( $512 \times 512$ ) color image			Histology ( $128 \times 128$ ) color image			Outex ( $512 \times 512$ ) 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

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

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

1 References

- 2 1 S. Jain and V. Laxmi, “Color image segmentation techniques: A survey,” in Proceed-  
3 ings of the International Conference on Microelectronics, Computing & Communication  
4 Systems, 189–197, Springer (2018). [doi:10.1007/978-981-10-5565-2\_17].
- 5 2 Y. Huang, F. Zhou, and J. Gilles, “Empirical curvelet based fully convolutional net-  
6 work for supervised texture image segmentation,” *Neurocomputing* 349, 31–43 (2019).  
7 [doi:10.1016/j.neucom.2019.04.021].
- 8 3 Z. Wang, “Image segmentation by combining the global and local properties,” *Expert*  
9 *Systems with Applications* 87, 30 – 40 (2017). [doi:10.1016/j.eswa.2017.06.008].
- 10 4 W. Tao, H. Jin, and Y. Zhang, “Color image segmentation based on mean shift and  
11 normalized cuts,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cy-*  
12 *bernetics)* 37(5), 1382–1389 (2007). [doi:10.1109/TSMCB.2007.902249].
- 13 5 J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans-*  
14 *actions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000).  
15 [doi:10.1109/34.868688].
- 16 6 D. E. Ilea and P. F. Whelan, “Image segmentation based on the integration of  
17 colour-texture descriptors a review,” *Pattern Recognition* 44(10-11), 2479–2501 (2011).  
18 [doi:10.1016/j.patcog.2011.03.005].
- 19 7 M. Kiechle, M. Storath, A. Weinmann, et al., “Model-based learning of local image fea-  
20 tures for unsupervised texture segmentation,” *IEEE Transactions on Image Processing*  
21 27(4), 1994–2007 (2018). [doi:10.1109/TIP.2018.2792904].

- 1 8 M. Haindl and S. Mikeš, “A competition in unsupervised color image segmentation,”  
2 Pattern Recognition 57, 136–151 (2016). [doi:10.1016/j.patcog.2016.03.003].
- 3 9 Y. Quan, H. Teng, T. Liu, et al., “Weakly-supervised sparse coding with geometric  
4 prior for interactive texture segmentation,” IEEE Signal Processing Letters 27, 116–120  
5 (2020). [doi:10.1109/LSP.2019.2959225].
- 6 10 S. Basu, I. Davidson, and K. Wagstaff, Constrained clustering: Advances in algorithms,  
7 theory, and applications, Chapman & Hall/CRC (2008).
- 8 11 U. Von Luxburg, “A tutorial on spectral clustering,” Statistics and Computing 17(4),  
9 395–416 (2007). [doi:10.1007/s11222-007-9033-z].
- 10 12 S. D. Kamvar, D. Klein, and C. D. Manning, “Spectral learning,” in International Joint  
11 Conference of Artificial Intelligence, Stanford InfoLab (2003).
- 12 13 I. Ahn and C. Kim, “Face and hair region labeling using semi-supervised spectral  
13 clustering-based multiple segmentations,” IEEE Transactions on Multimedia 18(7),  
14 1414–1421 (2016). [doi:10.1109/TMM.2016.2551698].
- 15 14 W. Chen and G. Feng, “Spectral clustering: a semi-supervised approach,” Neurocom-  
16 puting 77(1), 229–242 (2012). [doi:10.1016/j.neucom.2011.09.002].
- 17 15 X. Zhu, C. C. Loy, and S. Gong, “Constrained clustering with imperfect oracles,”  
18 IEEE Transactions on Neural Networks and Learning Systems 27(6), 1345–1357 (2015).  
19 [doi:10.1109/TNNLS.2014.2387425].
- 20 16 K. Xia, X. Gu, and Y. Zhang, “Oriented grouping-constrained spectral cluster-  
21 ing for medical imaging segmentation,” Multimedia Systems 26(1), 27–36 (2020).  
22 [doi:10.1007/s00530-019-00626-8].

- 1 17 J. Sourati, D. Erdogmus, J. G. Dy, et al., “Accelerated learning-based interactive im-  
2 age segmentation using pairwise constraints,” *IEEE Transactions on Image Processing*  
3 23(7), 3057–3070 (2014). [doi:10.1109/TIP.2014.2325783].
- 4 18 Z. Lu and M. A. Carreira-Perpinan, “Constrained spectral clustering through affinity  
5 propagation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8  
6 (2008). [doi:10.1109/CVPR.2008.4587451].
- 7 19 P. He, X. Xu, and L. Chen, “Constrained clustering with local constraint prop-  
8 agation,” in *European Conference on Computer Vision*, 223–232, Springer (2012).  
9 [doi:10.1007/978-3-642-33885-4\_23].
- 10 20 P. Han, G. Liu, S. Huang, et al., “Segmentation with selectively propagated constraints,”  
11 in *Neural Information Processing*, 585–592, Springer (2016). [doi:10.1007/978-3-319-  
12 46672-9\_65].
- 13 21 S. X. Yu and J. Shi, “Segmentation given partial grouping constraints,” *IEEE*  
14 *Transactions on Pattern Analysis and Machine Intelligence* 26(2), 173–183 (2004).  
15 [doi:10.1109/TPAMI.2004.1262179].
- 16 22 B. Ghanem and N. Ahuja, “Dinkelbach ncut: An efficient framework for solving nor-  
17 malized cuts problems with priors and convex constraints,” *International Journal of*  
18 *Computer Vision* 89(1), 40–55 (2010). [doi:10.1007/s11263-010-0321-2].
- 19 23 H. Zou, W. Zhou, L. Zhang, et al., “A new constrained spectral clustering for sar image  
20 segmentation,” in *Second Asian-Pacific Conference on Synthetic Aperture Radar*, 680–  
21 683, IEEE (2009). [doi:10.1109/APSAR.2009.5374114].
- 22 24 Y. Jia and C. Zhang, “Learning distance metric for semi-supervised image segmenta-

- 1 tion,” in Proceedings of the 15th IEEE International Conference on Image Processing,  
2 3204–3207, IEEE (2008). [doi:10.1109/ICIP.2008.4712477].
- 3 25 R. Sheikhpour, M. A. Sarram, S. Gharaghani, et al., “A survey on semi-  
4 supervised feature selection methods,” *Pattern Recognition* 64, 141–158 (2017).  
5 [doi:10.1016/j.patcog.2016.11.003].
- 6 26 J. Zhao, K. Lu, and X. He, “Locality sensitive semi-supervised feature selection,” *Neu-  
7 rocomputing* 71(10), 1842–1849 (2008). [doi:10.1016/j.neucom.2007.06.014].
- 8 27 M. Kalakech, P. Biela, L. Macaire, et al., “Constraint scores for semi-supervised feature  
9 selection: A comparative study,” *Pattern Recognition Letters* 32(5), 656–665 (2011).  
10 [doi:10.1016/j.patrec.2010.12.014].
- 11 28 K. Benabdeslem and M. Hindawi, “Constrained laplacian score for semi-supervised fea-  
12 ture selection,” in Proceedings of the Machine Learning and Knowledge Discovery in  
13 Databases, 204–218, Springer (2011). [doi:10.1007/978-3-642-23780-5\_23].
- 14 29 K. Benabdeslem and M. Hindawi, “Efficient semi-supervised feature selection: con-  
15 straint, relevance, and redundancy,” *IEEE Transactions on Knowledge and Data Engi-  
16 neering* 26(5), 1131–1143 (2014). [doi:10.1109/TKDE.2013.86].
- 17 30 X.-K. Yang, L. He, D. Qu, et al., “Semi-supervised feature selection for audio classifi-  
18 cation based on constraint compensated laplacian score,” *EURASIP Journal on Audio,  
19 Speech, and Music Processing* 2016(1), 9 (2016). [doi:10.1186/s13636-016-0086-9].
- 20 31 X.-K. Yang, L. He, D. Qu, et al., “Semi-supervised minimum redundancy maximum  
21 relevance feature selection for audio classification,” *Multimedia Tools and Applications*  
22 77(1), 713–739 (2018). [doi:10.1007/s11042-016-4287-0].

- 1 32 R. M. Haralick, K. Shanmugam, et al., “Textural features for image classifica-  
2 tion,” *IEEE Transactions on Systems, Man, and Cybernetics* 3(6), 610–621 (1973).  
3 [doi:10.1109/TSMC.1973.4309314].
- 4 33 A. K. Jain and F. Farrokhnia, “Unsupervised texture segmentation using gabor filters,”  
5 *Pattern Recognition* 24(12), 1167–1186 (1991). [doi:10.1016/0031-3203(91)90143-S].
- 6 34 L. Liu, J. Chen, P. Fieguth, et al., “From bow to cnn: Two decades of texture repre-  
7 sentation for texture classification,” *International Journal of Computer Vision* 127(1),  
8 74–109 (2019). [doi:10.1007/s11263-018-1125-z].
- 9 35 L. Busin, N. Vandenbroucke, and L. Macaire, “Color spaces and image segmenta-  
10 tion,” *Advances in Imaging and Electron Physics* 151, 65 – 168, Elsevier (2009).  
11 [doi:10.1016/S1076-5670(07)00402-8].
- 12 36 W. Siedlecki and J. Sklansky, “On automatic feature selection,” *International*  
13 *Journal of Pattern Recognition and Artificial Intelligence* 2(02), 197–220 (1988).  
14 [doi:10.1142/S0218001488000145].
- 15 37 Y. Bengio, J.-f. Paiement, P. Vincent, et al., “Out-of-sample extensions for lle, isomap,  
16 mds, eigenmaps, and spectral clustering,” in *Advances in Neural Information Processing*  
17 *Systems*, 177–184, MIT Press (2004).
- 18 38 M. Haindl and S. Mikes, “Texture segmentation benchmark,” in *Proceed-*  
19 *ings of the 19th International Conference on Pattern Recognition*, 1–4 (2008).  
20 [doi:10.1109/ICPR.2008.4761118].
- 21 39 Prague texture segmentation data generator and benchmark available from: [http://](http://mosaic.utia.cas.cz)  
22 [mosaic.utia.cas.cz](http://mosaic.utia.cas.cz) ((accessed May 23, 2020)).

- 1 40 N. Mevenkamp and B. Berkels, “Variational multi-phase segmentation using high-  
2 dimensional local features,” in IEEE Winter Conference on Applications of Computer  
3 Vision (WACV), 1–9, IEEE (2016). [doi:10.1109/WACV.2016.7477729].
- 4 41 J. Yuan, D. Wang, and A. M. Cheriyyadat, “Factorization-based texture seg-  
5 mentation,” IEEE Transactions on Image Processing 24(11), 3488–3497 (2015).  
6 [doi:10.1109/TIP.2015.2446948].
- 7 42 S. Yang, Y. Lv, Y. Ren, et al., “Unsupervised images segmentation via incremental dic-  
8 tionary learning based sparse representation,” Information Sciences 269, 48–59 (2014).  
9 [doi:10.1016/j.ins.2014.01.023].
- 10 43 Z. Kato, T.-C. Pong, and J. C.-M. Lee, “Color image segmentation and parameter  
11 estimation in a markovian framework,” Pattern Recognition Letters 22(3-4), 309–321  
12 (2001). [doi:10.1016/S0167-8655(00)00106-9].
- 13 44 O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomed-  
14 ical image segmentation,” in International Conference on Medical Image Computing  
15 and Computer-Assisted Intervention, 234–241, Springer (2015). [doi:10.1007/978-3-  
16 319-24574-4\_28].
- 17 45 W. Wang and J. Shen, “Deep visual attention prediction,” IEEE Transactions on Image  
18 Processing 27(5), 2368–2378 (2017). [doi:10.1109/TIP.2017.2787612].
- 19 46 H. Zhao, J. Shi, X. Qi, et al., “Pyramid scene parsing network,” in IEEE  
20 Conference on Computer Vision and Pattern Recognition, 6230–6239 (2017).  
21 [doi:10.1109/CVPR.2017.660].
- 22 47 J. A. Ozolek and C. A. Castro, “Teratomas derived from embryonic stem cells as models

- 1 for embryonic development, disease, and tumorigenesis,” in *Embryonic Stem Cells-Basic*  
2 *Biology to Bioengineering*, IntechOpen (2011). [doi:10.5772/23866].
- 3 48 M. T. McCann, D. G. Mixon, M. C. Fickus, et al., “Images as occlusions of textures: A  
4 framework for segmentation,” *IEEE Transactions on Image Processing* 23(5), 2033–2046  
5 (2014). [doi:10.1109/TIP.2014.2307475].
- 6 49 T. Ojala, T. Maenpaa, M. Pietikainen, et al., “Outex-new framework for empirical  
7 evaluation of texture analysis algorithms,” in *Object recognition supported by user in-*  
8 *teraction for service robots*, 1, 701–706, IEEE (2002). [doi:10.1109/ICPR.2002.1044854].
- 9 50 J. Pont-Tuset and F. Marques, “Measures and meta-measures for the supervised eval-  
10 uation of image segmentation,” in *Proceedings of the IEEE Conference on Computer*  
11 *Vision and Pattern Recognition*, 2131–2138 (2013). [doi:10.1109/CVPR.2013.277].
- 12 51 D. Martin, C. Fowlkes, D. Tal, et al., “A database of human segmented natural images  
13 and its application to evaluating segmentation algorithms and measuring ecological  
14 statistics,” in *Proceedings Eighth IEEE International Conference on Computer Vision*.  
15 *ICCV 2001*, 2, 416–423, IEEE (2001). [doi:10.1109/ICCV.2001.937655].
- 16 52 X. Jia, T. Lei, X. Du, et al., “Robust self-sparse fuzzy clustering for image segmenta-  
17 tion,” *IEEE Access* 8, 146182–146195 (2020). [doi:10.1109/ACCESS.2020.3015270].

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

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

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

## 13 List of Figures

- 14 1 Flowchart of the proposed color-texture image segmentation (CFS-SC).
- 15 2 Illustration of local Haralick texture feature computing.
- 16 3 Two test color-texture images from the Prague dataset (first row) and their  
17 corresponding ground truth images (bottom row): (a) image 7\_1\_1 and (b)  
18 image 17\_1\_1.
- 19 4 Constraint scores versus number  $m$  of selected features for the two test images:  
20 (a) image 7\_1\_1 and (b) image 17\_1\_1.

- 1     5     Accuracy versus number  $m$  of features obtained using the proposed semi-  
2            supervised constraint scores on test images for different numbers of  $n$  and  
3             $p$ . Left column: image 7\_1\_1 and right column: image 17\_1\_1. (a)-(b)  
4             $n = 200$ , (c)-(d)  $n = 300$ , and (e)-(f)  $n = 400$ .
- 5     6     Accuracy versus number  $m$  of features obtained using the proposed semi-  
6            supervised constraint score on the two test images for different numbers  $n$  of  
7            sample pixels: (a) image 7\_1\_1 and (b) image 17\_1\_1.
- 8     7     Accuracy-score  $\varepsilon^{SS}$  versus number  $m$  of selected features for the test images:  
9            (a) image 7\_1\_1 and (b) image 17\_1\_1.
- 10    8     Accuracy versus number  $m$  of features for the two test images: (a) image  
11           7\_1\_1 and (b) image 17\_1\_1.
- 12    9     Accuracy versus number  $p$  of prototypes when  $m = 90$  features have been  
13           selected and  $n = 300$  on the two test images: (a) image 7\_1\_1 and (b) image  
14           17\_1\_1.
- 15    10    Exemplary segmentation results on the Prague color benchmark. From top to  
16           bottom: Original images, ground truth, FCNTsup, EWT-FCNT, FCNTun-  
17           sup, texNcut, WSSCGP, CFS-SC-nr, and CFS-SC.
- 18    11    Exemplary segmentation results on the Histology dataset. From top to bot-  
19           tom: Original images, ground truth, FSEG, ORTSEG, WSSCGP, and CSC-  
20           FS.
- 21    12    Exemplary segmentation results on the Outex dataset. From top to bottom:  
22           Original images, Ground truth, PSP-Net, U-Net, EWT-FCNT, PCA-MS,  
23           FSEG, WSSCGP, and CFS-SC.

1     **13** Segmentation results on some images of the Berkeley database. The values  
2           of CO, F-measure, and PRI for each result are presented in parentheses.

### 3 List of Tables

4     **1** Results of unsupervised and semi-supervised methods on the large Prague  
5           dataset (80 test images). The arrows  $\uparrow$  |  $\downarrow$  denote the required criterion  
6           direction. Here, 'nr' means no segmentation refinement.

7     **2** Results of our method CFS-SC and supervised methods on the normal Prague  
8           dataset (20 test images). The arrows  $\uparrow$  |  $\downarrow$  denote the required criterion  
9           direction. Here, 'nr' means no segmentation refinement.

10    **3** Results on the Histology dataset. The arrows  $\uparrow$  |  $\downarrow$  denote the required  
11           criterion direction and the best score is marked in boldface. Here, 'nr' means  
12           no segmentation refinement.

13    **4** Results on the Outex dataset. Each row corresponds to a segmentation qual-  
14           ity measure and the best score is marked in boldface. Here, 'nr' means no  
15           segmentation refinement.

16    **5** The average computing time (second) per image of FSEG, WSSCGP, and  
17           CFS-SC on the tree datasets. (FE: Feature extraction, OS: Other steps.)