Tree-wise Discriminative Subtree Selection for Texture Image Labeling

Tsubasa Hirakawa, Toru Tamaki, Takio Kurita, Bisser Raytchev, Kazufumi Kaneda, Chaohui Wang, Laurent Najman

Tree-wise Discriminative Subtree Selection for Texture Image Labeling

Tsubasa Hirakawa, Toru Tamaki, Takio Kurita, Bisser Raytchev, Kazufumi Kaneda, Chaohui Wang, and Laurent Najman

***Abstract*—In this study, we propose a method for texture image labeling that works with a small number of training images. Our method is based on a tree of shapes and histogram features computed on the tree structure. Labeling results could be obtained by simply classifying the histogram features of all nodes in a tree of shapes. However, it is difficult to obtain satisfactory results because features of smaller nodes are not sufficiently discriminative. Consequently, our method selects optimal discriminative subtrees for image labeling. We model an objective function that includes the parameters of a classifier and a set of thresholds for each training image to be used to select optimal subtrees. Then, labeling is performed by mapping the classification results of selected subtrees into corresponding blobs in the image. Experimental results with synthetic and real datasets that we created for evaluation show that the proposed method performs qualitatively and quantitatively much better than the existing methods.**

***Index Terms*—Image labeling, Texture analysis, Tree of shapes, Histogram features, Mathematical morphology**

I. Introduction

TEXTURE is one of the most important cues for image understanding. A number of texture descriptors, such as wavelet transforms [1], [2], local binary patterns (LBPs) [3], Gabor [4], [5], and textons [6], have been proposed to model texture in images, and image labeling and segmentation methods using such texture descriptors have been proposed. One of the limitations of these texture descriptors is the difficulty of representing a wide variety of changes in texture appearance. More specifically, texture appearance changes in geometry, scale, and contrast. Therefore, learning-based image segmentation methods must have a large number of training images with ground truth segmentation labels to be able to adapt to the texture variations. However, these methods are not applicable to cases with a small number of training images.

In this study, we propose a method for texture image segmentation that works with a small number of training images. In some cases, such as segmentation of natural images, we might be able to collect and use a large number of training images, whereas in other cases, such as in medical image analysis, collecting data and creating ground truth labels are very expensive and difficult.

There has been a promising attempt to deal with geometrical and contrast changes in texture. Xia et al. [7] have proposed a texture descriptor, shape-based invariant texture analysis (SITA), for a texture image classification task based on the tree of shapes [8], [9]. In the field of mathematical morphology, a hierarchical representation, the morphological tree, is popular, and a number of hierarchical trees, such as min/max trees [10], [11], binary partition trees [12], minimum spanning forests [13], the tree of shapes [8], [9], and color tree of shapes [14], have been proposed. Morphological trees have been applied to, for example, biomedical imaging [15], [16], [17]. Xia et al. [7] focused on the natural scale-space structure and invariance of contrast change in the tree of shapes and proposed the SITA descriptor based on the tree of shapes (details are described in Section III-B). To the best of our knowledge, this was the first attempt to create texture descriptors from the tree of shapes. A SITA feature is a histogram of texture features aggregated from all of the nodes in a tree, with the root node of the tree representing the SITA feature of the image. It can be noted that a node corresponds to a part (or a region or blob) of the image, whereas the root represents the entire image. A parent node corresponds to a blob that contains blobs of children nodes. This constitutes a hierarchical structure of the image, which is called the tree of shapes. Xia et al. [7] show through their experimental results that this hierarchical structure renders SITA features invariant to local geometric, scale, and radiometric changes, with good performance in image classification and retrieval problems. Other texture descriptors based on the tree of shapes have also been proposed. Liu et al. [18] introduced a bag-of-words model of the branches in a tree of shapes and represented the co-occurrence patterns of shapes. He et al. [19] adopted the basic idea of LBPs to propose a texture descriptor. However, these methods handle only texture patch classification and retrieval tasks, and no work has been performed on handling multiple textures in a single image for texture segmentation.

Inspired by the invariance property of SITA, in this study, we propose a novel segmentation method for texture images [1]. An overview of the proposed method is shown in Figure 1. The idea of our method is to adopt SITA, but to use it for segmentation of an image rather than for classification of images. In the original work on SITA [7], a SITA feature

T. Hirakawa, T. Tamaki, T. Kurita, B. Raytchev, K. Kaneda are with the Department of Information Engineering, Faculty of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima, 739-8527, Japan.
E-mail: hirakawat@hiroshima-u.ac.jp (T. Hirakawa), tamaki@hiroshima-u.ac.jp (T. Tamaki)
C. Wang and L. Najmain are the Université Paris-Est, LIGM Laboratory (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, 77420 Marne-la-Vallée, France.
The last two authors are joint senior authors.

[1]A conference version of this paper was presented [20]. This paper extends that version with the extension of the objective function, effective optimization, and more quantitative evaluations with different datasets.

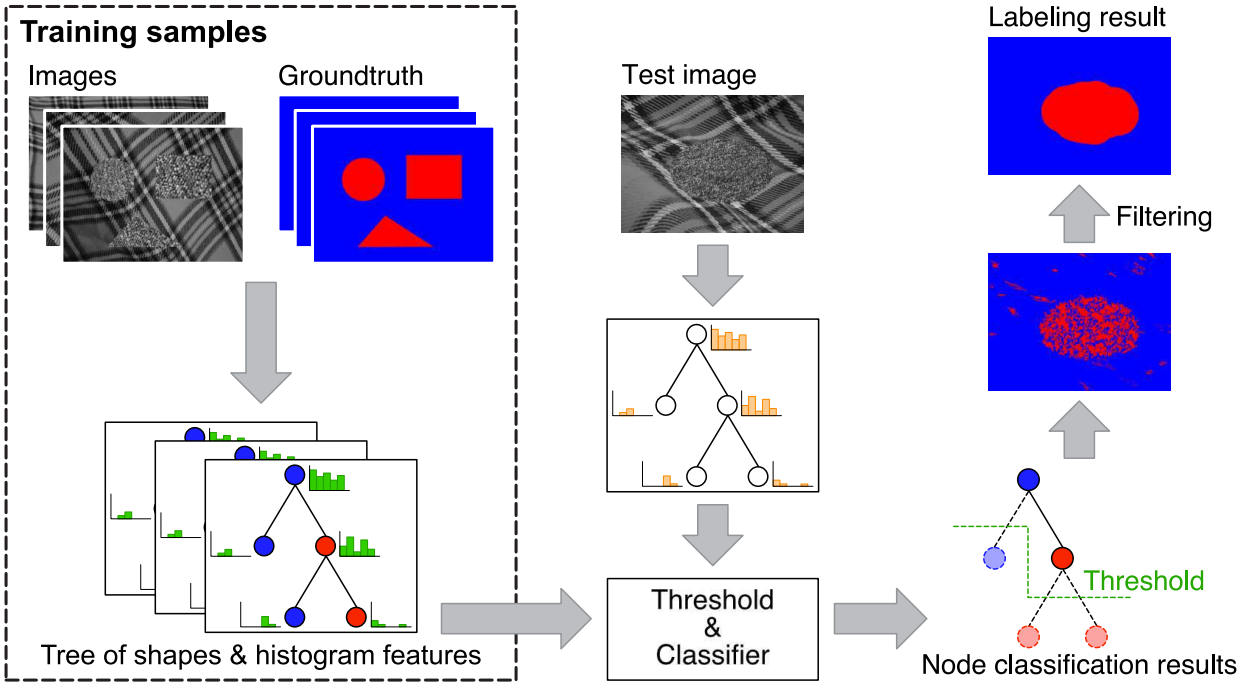


Fig. 1. Overview of the proposed method.

is computed for a classification task at the root of the tree of an image. Here, for a segmentation task, we compute the SITA features at all of the nodes in the tree and classify every node to predict labels of pixels corresponding to the nodes. In other words, we compute SITA features at root nodes of all of the subtrees of the original tree. This simple concept is rather straightforward but has a problem of instability for histogram feature computation. If we compute a SITA feature at the root of a small subtree, for example, near the leaf nodes of the original tree, then the resulting histogram (i.e., the SITA feature) is less stable and discriminative for classification because the small subtree has a small number of nodes available for SITA histogram computation. For this reason, we propose to find subtrees that are sufficiently stable and discriminative for classification by jointly estimating the sizes of subtrees and training a classifier in the training stage. At the labeling stage, given a test image, we estimate (again simultaneously) the sizes and labels of the subtrees of the tree of the test image.

The contribution of our work is two-fold. First, we propose a novel image segmentation framework based on the tree of shapes that makes our method robust to changes in texture appearance. Second, our method works on a small dataset of training images. We use the SITA features of many nodes from an image instead of a single SITA feature at the root node per image. Therefore, our method learns sufficient features to be discriminative for training, whereas the number of training images can still be small (details are described in Section V).

The remainder of the study is organized as follows. Section II reviews related work on texture analysis and image labeling. In Section III, we briefly introduce the basic notions and definitions of the tree of shapes and SITA. Then, in Section IV, we describe the details of the proposed texture image segmentation method. Experimental results with a synthetic texture image dataset and a real image dataset and discussion of those results are provided in Section V. We conclude the study in Section VI.

II. Related work

Texture image labeling (or segmentation) is a well-studied task in the field of computer vision, medical imaging [21], [22], [23], and synthetic aperture radar (SAR) image processing [24], [25], and a number of methods for performing that task have been proposed. One popular approach uses Markov random fields (MRFs). For modeling spatial consistency, an MRF comprises unary data terms of individual pixels (patches, sometimes super pixels) and pairwise terms between neighbors. The accuracy of MRFs highly depends on the unary term, for which various texture descriptors are used. A number of unsupervised texture image segmentation methods based on MRF have been proposed [26], [27], [28], but we focus here on supervised texture image segmentation methods using MRFs. One of the supervised MRF approaches was proposed by Hirakawa et al. [29]. They proposed a patch-based method that uses a posterior probability obtained from a support vector machine (SVM) with bag-of-visual words (BoVW) histograms using more than 1,000 labeled patches for training.

Another popular approach uses conditional random fields (CRFs); MRF is a generative model, whereas CRF is a discriminative model. CRF has a structural-learning property and can train the spatial structures of labels. Shotton et al. [30], [31] proposed TextonBoost for object segmentation. They introduced a novel texture descriptor, the texture-layout filter, into the CRF framework. For evaluation, they used the MSRC-21 dataset and 271 images [31] for training. Bertelli et al. [32] adopted a kernel-structured SVM for object segmentation.

They introduced a pairwise term to a structured SVM that is the same as the CRF framework. Their method was evaluated via 3-fold cross validation with three datasets [33], [34]. For each trial, 400, 218, and 56 images for the three datasets, respectively, are used for training. A fully connected CRF has been proposed [35] that uses a mean field approximation with a linear combination of Gaussian kernels for a pairwise edge potential for efficient inference. In their experiment, they used approximately 270 images on the MSRC-21 [31] dataset and 770 images on the PASCAL Visual Object Classes (PASCAL VOC) dataset [36] for training.

Recently, convolutional neural networks (CNNs) have been proposed for computer vision tasks that include image labeling. Farabet et al. [37] proposed a labeling method for scene parsing. Their approach assigns estimated labels to pixels and then refines the results using superpixels, CRFs, and optimal-purity covers on a segmentation tree. Long et al. [38] used a fully convolutional network trained in an end-to-end manner. These two methods use the SIFT flow dataset [39] for evaluation, with 2,488 images used for training and 280 images used for testing. Other methods have also been proposed [40], [41], [42] using hundreds of images for training. Consequently, these CNN-based approaches have shown good performance, as long as large numbers of training images are available. It would be difficult to achieve good performance for smaller datasets.

In contrast to the methods above, our proposed method works effectively with a small number of training images. In this study, we show a comparison of the proposed method with these related approaches using a small dataset of texture image segmentations.

III. Tree of shapes and SITA

Herein, we briefly describe the definition of tree of shapes and the SITA histogram feature.

A. Tree of shapes

A tree of shapes [8], [9] is an efficient image representation in a self-dual form. Given an image $u : \mathbb{R}^2 \to \mathbb{R}$, the upper and lower level sets of u are defined for $\lambda \in \mathbb{R}$ as follows:

$$\chi_\lambda(u) = \{x \in \mathbb{R}^2 | u(x) \geq \lambda\} \tag{1}$$
$$\chi^\lambda(u) = \{x \in \mathbb{R}^2 | u(x) < \lambda\}. \tag{2}$$

From these level sets, we can obtain tree structures $\mathcal{T}_\geq(u)$ and $\mathcal{T}_<(u)$ that comprise connected components of upper- and lower-level sets as follows:

$$\mathcal{T}_\geq(u) = \{\Gamma \mid \Gamma \in \mathcal{CC}(\chi_\lambda(u)), \forall\lambda\} \tag{3}$$
$$\mathcal{T}_<(u) = \{\Gamma \mid \Gamma \in \mathcal{CC}(\chi^\lambda(u)), \forall\lambda\}, \tag{4}$$

where $\mathcal{CC}$ is an operator giving a set of connected components.

Furthermore, we define a set of upper shapes $\mathcal{S}_\geq(u)$ and lower shapes $\mathcal{S}_<(u)$. These sets are obtained by the cavity filling (saturation) of components of $\mathcal{T}_\geq(u)$ and $\mathcal{T}_<(u)$. A *tree of shapes* of u is defined as the set of all shapes defined as $\mathcal{G}(u) = \mathcal{S}_\geq(u) \cup \mathcal{S}_<(u)$.

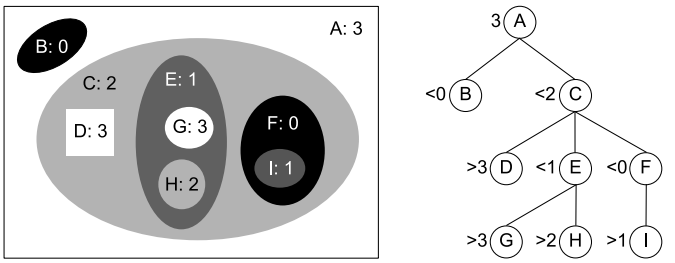


Fig. 2. Example of a synthetic image (left) and corresponding tree of shapes (right). Alphabet letters denote the correspondence between blobs and tree nodes, and numbers denote gray levels. Inequality signs, i.e., $<$ and $>$, denote dark and bright nodes, respectively.

As a consequence of the nesting property of level sets, the tree of shapes forms a hierarchical structure. Figure 2 shows an example of a tree of shapes. Let $T = \{V, E\}$ be a tree of shapes, where $V = \{v_j\}$ is a set of nodes and $E = \{(v_j, v_k)\}$ is a set of edges. Let s_j be a blob in u corresponding to v_j, and let a_j be the area (the number of pixels) of s_j. The area of an image u is denoted as A. We define parent and children nodes of v_j as

$$Pa(v_j) = \{v_k | (v_j, v_k) \in E, a_j < a_k\} \tag{5}$$
$$Ch(v_j) = \{v_k | (v_j, v_k) \in E, a_j > a_k\}, \tag{6}$$

respectively, and we similarly define $Pa(s_j)$ and $Ch(s_j)$. Note that we use blob s_j and node v_j interchangeably in the following discussion.

B. SITA

Xia et al. [7] proposed SITA, a texture descriptor based on the tree of shapes. It comprises four features of blobs corresponding to nodes. For more details on SITA, we refer the interested reader to [7].

The first two features of SITA are elongation

$$\epsilon(s_j) = \frac{\lambda_{2j}}{\lambda_{1j}} \tag{7}$$

and compactness

$$\kappa(s_j) = \frac{1}{4\pi\sqrt{\lambda_{1j}\lambda_{2j}}}, \tag{8}$$

where $\lambda_{1j}, \lambda_{2j}$, $(\lambda_{1j} \geq \lambda_{2j})$ are major and minor axes of blob s_j approximated to ellipse. The third feature is the scale ratio $\alpha(s_j)$ defined by

$$\alpha(s_j) = \frac{\mu(s_j)}{\sum_{s_k \in \bigcup_M Pa^M(s_j)} \mu(s_k)/M}, \tag{9}$$

where $Pa^M(s_j)$ is the Mth ancestor blob and $\mu(s_j)$ is area of s_j. This is the ratio of blob sizes between s_j and the ancestor blobs. In accordance with [7], we set $M = 3$ in our experiments. The fourth feature comprises normalized gray values, $\{\gamma(x)\}$, computed for each pixel $x \in s_j$ as follows:

$$\gamma(x) = \frac{u(x) - m_{j(x)}}{\sigma_{j(x)}}, \tag{10}$$

where $m_{j(x)}$ and $\sigma^2_{j(x)}$ are the mean and variance of $u(x)$ over $s_{j(x)}$, respectively.

Then, the computed four texture features are used for histogram feature computation with respect to dark and bright nodes. Here, let $\nu(s_j)$ be the gray level of blob s_j defined as follows:

$$\nu(s_j) = \frac{1}{\mu(s_j) - \mu(Ch(s_j))} \sum_{x \in s_j / Ch(s_j)} u(x). \tag{11}$$

A blob is defined as dark if $\nu(s_j) \leq \nu(Pa(s_j))$ and as bright otherwise. The root node is simply defined as dark. For all dark nodes, we compute three histograms of the first three texture features, i.e., $\epsilon(s_j)$, $\kappa(s_j)$, and $\alpha(s_j)$, and we do the same for all bright nodes. For all nodes, we compute a histogram of $\{\gamma(x)\}$. Consequently, we obtain seven histograms[2]. These seven histograms are concatenated into a single one, which is called a SITA feature.

C. Recursive representation of SITA

In the original study, the authors did not describe how to compute a SITA feature. Here, we propose a recursive procedure because of the relation to our proposed method. The SITA computation can be performed by aggregating histograms from leaf nodes to the root as follows. Let $\boldsymbol{g}_j$ be a concatenated histogram computed from node v_j only. The aggregated histogram $\boldsymbol{h}(v_j)$ from nodes below v_j in the tree is computed recursively as

$$\boldsymbol{h}_j = \boldsymbol{g}_j + w_{agg} \sum_{v_k \in Ch(v_j)} \boldsymbol{h}_k, \tag{12}$$

where w_{agg} is the weight for aggregating histograms of children nodes. Finally, the histogram $\boldsymbol{h}_{\mathrm{root}}$ at the root node v_{root} is normalized to have a unit L_1 norm with respect to each of the seven histograms.

IV. Proposed Method

Here, we define notions of trees of shapes for a set of images. Let $\{u_i\}_{i=1}^N$ be a set of images and A_i be the area of u_i. A tree of shapes of u_i is defined as $T_i = \{V_i, E_i\}$, and $n_i = |V_i|$ is the number of nodes in T_i. Each node $v_{ij} \in V_i$ has the corresponding blob s_{ij} with the area a_{ij}.

We assume that each blob is given a ground truth label $y_{ij} \in L$ for training images, where L is a set of labels (in our case $L = \{-1, 1\}$.)

In contrast to the original SITA, we compute aggregated histograms at *every* node. Let $\boldsymbol{g}_{ij}$ be a histogram computed from node v_{ij} only. Then, the aggregated histogram $\boldsymbol{h}_{ij}$ nodes below node v_{ij} in the tree are computed recursively as

$$\boldsymbol{h}_{ij} = \boldsymbol{g}_{ij} + w_{agg} \sum_{v_{ik} \in Ch(v_{ij})} \boldsymbol{h}_{ik}, \tag{13}$$

and then normalized to have a unit L_1 norm.

[2]In this study, we set the number of bins as 25 and the histogram range as $(0, 1)$ for $\epsilon(s_j)$, $\kappa(s_j)$, and $\alpha(s_j)$. For $\{\gamma(x)\}$, we set the number of bins as 50 and the histogram range as $(-25, 10)$. Note that we set the histogram range for $\{\gamma(x)\}$ experimentally.

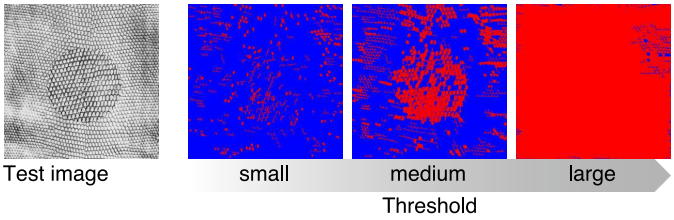


Fig. 3. Examples of labeling results using subtrees with different node sizes. Subtrees used for labeling are determined using an estimated threshold.

We mainly compute SITA features at root nodes of *all* subtrees in the tree, whereas the original SITA features are computed at the root node only. One of the simple ideas for image labeling is to classify these node-wise SITA features to obtain labels of blobs. As mentioned previously, small subtrees are not useful for classification. Figure 3 shows examples of unstable labeling results. Here, we set three different thresholds to areas of subtree root nodes and classify SITA features of the subtrees that are larger than the thresholds for labeling. As shown in the figure, the results would not be satisfactory with excessively small (or large) subtrees with an excessively small (or large) area threshold. Based on this observation, we assume that there exists an optimal threshold for the area (or size) of subtrees. Furthermore, we have no reason to expect that a single area threshold is desirable for different training images whose texture contents might be different.

Therefore, we formulate the task as a joint optimization problem, estimating thresholds for each training image and training a classifier. Let θ_i be a threshold for u_i, and let $\boldsymbol{w}$ and b be parameters of a classifier (here, using SVMs, these are the weight vector and the bias, respectively). We define the objective function for u_i as follows:

$$\begin{aligned} E_i(\theta_i, \boldsymbol{w}, b) =& \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &+ \frac{1}{n_i} \sum_j^{n_i} W_{ij} \ell\left(y_{ij}(\boldsymbol{w}^T \boldsymbol{h}_{ij} + b)\right) \\ &+ \frac{\lambda}{2}\theta_i^2. \end{aligned} \tag{14}$$

The first term is the SVM regularizer, and in the second term, $\ell(\cdot)$ is the hinge loss function of the SVM. The third term is the regularizer for θ_i, and λ is the scale parameter.

In the objective function, we introduce the sample weight W_{ij} for $\boldsymbol{h}_{ij}$. In the proposed method, we use θ_i to threshold smaller subtrees. In other words, we use the histograms of subtrees larger than θ_i and ignore the others. This is the basic concept, and it can be implemented by setting zero or one as values of W_{ij}. However, this is difficult to solve as an optimization with gradient-based solvers. Therefore, we adopt a sigmoid function for representing the thresholding and define W_{ij} as follows:

$$W_{ij} = W(a_{ij}, \theta_i) = \frac{1}{1 + \mathrm{e}^{-\beta(a_{ij} - \theta_i)}}, \tag{15}$$

where β is the gain parameter of the sigmoid.

In the training phase, we have a set of N training images $\{u_i\}_{i=1}^N$, and the objective function to be minimized for training is

$$\begin{aligned} E(\boldsymbol{\theta}, \boldsymbol{w}, b) &= \frac{1}{N}\sum_i^N E_i(\theta_i, \boldsymbol{w}, b) \\ &= \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &+ \frac{1}{N}\sum_i^N \frac{1}{n_i}\sum_j^{n_i} W(a_{ij}, \theta_i)\ell\left(y_{ij}(\boldsymbol{w}^T\boldsymbol{h}_{ij} + b)\right) \\ &+ \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2, \end{aligned} \tag{16}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)^T$.

A. Optimization

Given a training set of images, we estimate parameters $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{w}}, \hat{b}$ by

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{w}}, \hat{b} = \underset{\boldsymbol{\theta}, \boldsymbol{w}, b}{\operatorname{argmin}} E\left(\boldsymbol{\theta}, \boldsymbol{w}, b\right). \tag{17}$$

Since this is non-linear and non-convex, we use a block-coordinate decent approach, that is, given initial value $\boldsymbol{\theta}_0$, we iteratively estimate, first, the classifier parameters $\boldsymbol{w}$ and b and then the thresholds $\boldsymbol{\theta}$.

1) Classifier training: To estimate $\boldsymbol{w}$ and b, given $\boldsymbol{\theta}_{k-1}$, we solve

$$\boldsymbol{w}_k, b_k = \underset{\boldsymbol{w}, b}{\operatorname{argmin}} E(\boldsymbol{\theta}_{k-1}, \boldsymbol{w}, b). \tag{18}$$

This is an SVM formulation with sample weights, which is convex. We solve this problem using the primal solver of LIBLINEAR [43] because dual solvers are difficult to apply to a large number of training samples. (In our case, the SVM is trained on approximately hundred thousand node features.)

2) Threshold estimation: To estimate $\boldsymbol{\theta}$, given $\boldsymbol{w}_k$ and b_k, we solve

$$\boldsymbol{\theta}_k = \underset{\theta}{\operatorname{argmin}} E(\boldsymbol{\theta}, \boldsymbol{w}_k, b_k). \tag{19}$$

This is non-convex because $\boldsymbol{\theta}$ depends on histograms $\boldsymbol{h}_{ij}$. We solve this using Newton's method because we confirmed experimentally that the cost function is smooth and has a single minimum in many cases (details are described in Section V-A), and the Hessian is diagonal, as shown below.

The gradient is given by

$$\nabla E = \left(\frac{\partial E}{\partial \theta_1}, \ldots, \frac{\partial E}{\partial \theta_N}\right), \tag{20}$$

where

$$\frac{\partial E}{\partial \theta_i} = \frac{1}{Nn_i}\sum_j^{n_i} -\beta W_{ij}\left(1 - W_{ij}\right)\ell\left(y_{ij}(\boldsymbol{w}^T\boldsymbol{h}_{ij} + b)\right) + \lambda\theta_i. \tag{21}$$

The Hessian is

$$\nabla^2 E = \begin{pmatrix} \frac{\partial^2 E}{\partial \theta_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial^2 E}{\partial \theta_N^2} \end{pmatrix}, \tag{22}$$

where

$$\frac{\partial^2 E}{\partial \theta_i^2} = \frac{1}{Nn_i}\sum_j^{n_i} \beta^2 W_{ij}\left(1 - W_{ij}\right)\left(1 - 2W_{ij}\right)\ell\left(y_{ij}(\boldsymbol{w}^T\boldsymbol{h}_{ij} + b) + \lambda\right) \tag{23}$$

The Hessian is diagonal because there are no cross terms in the second order derivatives. Therefore, we can parallelize the implementation to reduce the computation time.

3) Stopping condition: We stop the alternation when $\boldsymbol{\theta}_k$ converges with the termination criterion of

$$\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}\| = \epsilon. \tag{24}$$

B. Labeling procedure

Typically, in the labeling phase of a test image u, we first construct the tree of shapes of u, compute $\boldsymbol{h}_j$ for every node v_j, and then conceptually classify those $\boldsymbol{h}_j$ whose area a_j is larger than a threshold. However, here we choose to do this differently because we have estimated a set of thresholds θ_i for training images u_i. Instead of the above approach, we propose minimizing the objective function Eq. (14) again for the test image as we did in the training phase.

First, we fix the classifier parameters (hence, the loss in the objective function) and minimize the following objective function to estimate $\hat{\theta}$ as follows:

$$E(\theta, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_j^n W(a_j, \theta)\ell\left(y_j(\boldsymbol{w}^T\boldsymbol{h}_j + b)\right) + \frac{\lambda}{2}\theta^2, \tag{25}$$

where $\boldsymbol{y} = \{y_j\}_{j=1}^n$ is a set of labels. This is the same as with the threshold estimation in the training phase but for only a single test image (i.e., $N = 1$).

Algorithm 1 provides details of the labeling procedure. To obtain a segmentation result, we perform the classification procedure starting from the root node and proceeding down to the leaf nodes. At each node n_j, if $a_j \geq \hat{\theta}$, we classify $\boldsymbol{h}_j$ and then assign the resulting y_j to all pixels in s_j, even including those that have been assigned labels by parent nodes (i.e., overwriting labels). This downward traversal of the tree stops once it reaches smaller nodes.

To refine the segmentation result, we apply simple morphological filtering [44] as a post process.

V. Experimental results

We tested the proposed method and compared it with existing methods using synthetic and real image datasets.

Algorithm 1 Labeling procedure

Input: threshold $\hat{\theta}$
Input: SVM parameters $\boldsymbol{w}$ and b
Input: tree of shapes T of test image u
Output: labeling result u_l
initialize $u_l = \mathbf{0}$
for node $j = 1 \dots n$ **do**
  if $a_j \geq \hat{\theta}$ **then**
    $y_j \leftarrow \text{sign}(\boldsymbol{w}^T \boldsymbol{h}_j + b)$ \\ classify a histogram.
  else $\{a_j < \hat{\theta}\}$
    $y_j \leftarrow y_{Pa(v_j)}$ \\ assign label of parent node.
  end if
end for
$u_l(x) = y_{j(x)}, j(x) = \text{argmin}_j\{a_j | x \in s_j\}$ \\ map label to the corresponding pixel.
return u_l

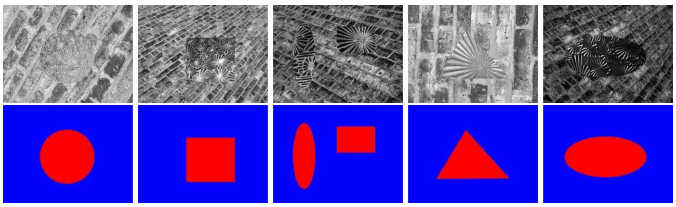


Fig. 4. Example of a texture image sub-dataset. The upper and bottom rows show created texture images and corresponding ground truths, respectively. Blue and red indicate the class of each pixel.

For the experiments, we created a synthetic dataset from the UIUC database [45], which comprises 25 various texture classes, each of which contains 40 images of size 640 × 480 pixels. This dataset comprises seven sub-datasets. Each sub-dataset includes five images containing one to three regions made of two classes. Figure 4 shows an example of a created sub-dataset containing five images and corresponding ground truth labels. Trees of shapes of these images have sufficient nodes, 18,855 nodes per image, on average. For each sub-dataset, we randomly select three images for training and two for testing. In the experiments, we set sigmoid gain $\beta = 0.1$, initial values of thresholds to 1,000, and weight for aggregating histogram $w_{agg} = 1.0$. For quantitative evaluation, we use the Dice coefficient [46].

We used three methods for comparison: Felsenszwalb's unsupervised image segmentation [47], a patch-based MRF segmentation with SVM [29], and a fully connected CRF [35] with TextonBoost [30], [31].

A. Energy convergence

First, we show the convergence property of our proposed method. As mentioned previously, our method minimizes the cost function using Newton's method for $\boldsymbol{\theta}$ and SVM training for $\boldsymbol{w}, b$. Here, we focus on the convergence of Newton's method for the nonlinear optimization of $\boldsymbol{\theta}$ because SVM training is convex and guaranteed to converge. Figure 5 shows the cost function values over different initial values and scale parameters λ for a training image. Note that we did not add the regularizer of SVM, i.e., $\|\boldsymbol{w}\|^2$, to the cost function because it

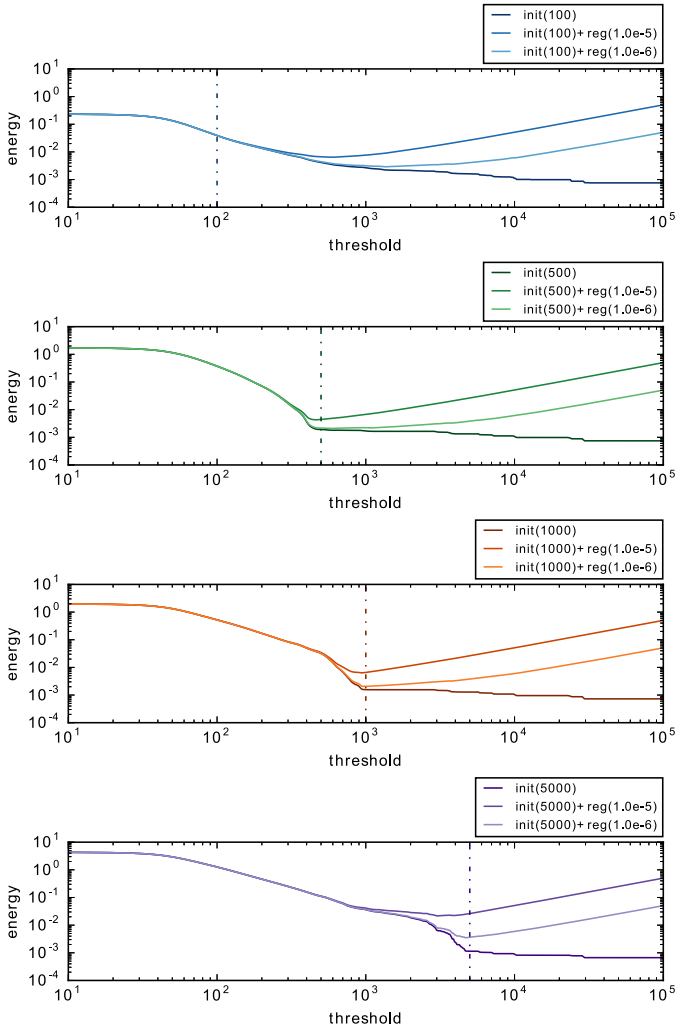


Fig. 5. Cost function over different initial values and thresholds for a training image. The horizontal and vertical axes show the threshold and corresponding cost function value, respectively. Different colors are used for different initial values of θ_i. From top to bottom, the initial thresholds are 100, 500, 1,000, and 5,000.

takes excessively large values that interfere with quantitatively observing the plot. In the following figures for cost function values, we also do not add the regularizer of SVM. As can be observed, the cost function is not convex, but adding the regularizer for $\boldsymbol{\theta}$ renders the energy values rather convex. Figure 6 shows the cost function values against the threshold over different iterations, as well as for different initial values $\boldsymbol{\theta}_0$. Note that this figure shows the cost function values of a test image in the labeling phase because in the training phase we estimate θ_i separately for each training image, and it is difficult to visualize all of them in a single plot. We observe that the minimum of the cost function decreases, and the threshold θ_i converges with different initial values. However, it should be avoided to use a small initial value, such as $\boldsymbol{\theta}_0 = 10$, because the cost function between $\theta = [10^0, 10^1]$ looks almost flat and non-convex and the iteration might not converge. It would be better to use a large initial value, typically larger than 1000. For other image datasets, initial values should also be larger than the minimum. We may use the image size (i.e.,

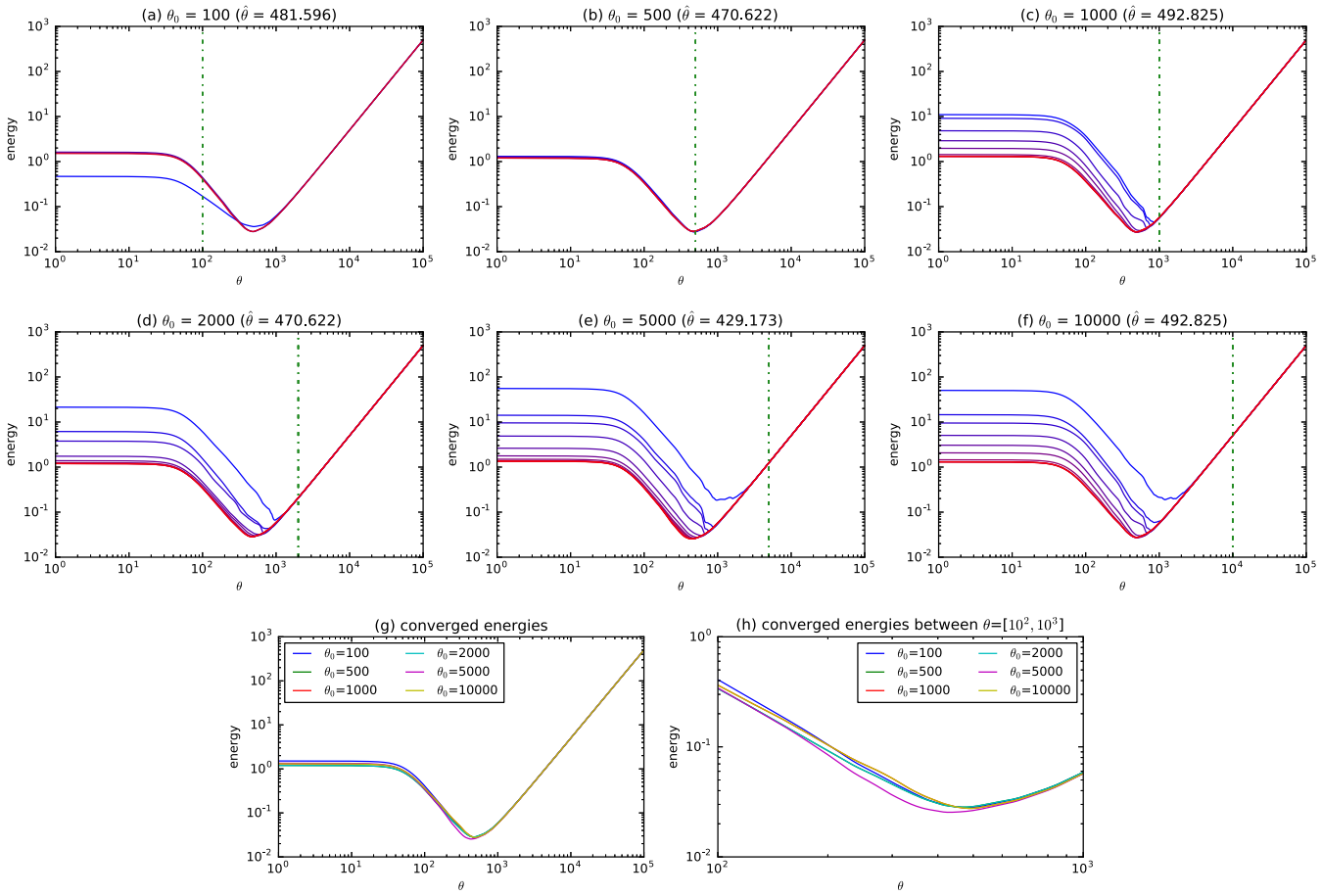


Fig. 6. Cost function values against the threshold over different iterations, starting with different initial values $\boldsymbol{\theta}_0 = 100$, 500, 1000, 2000, 5000, and 10000. The horizontal and vertical axes show the threshold and corresponding cost function value, respectively. (a – f) Convergence with different initial values indicated as vertical green lines. Different colors are used for different iterations; the first iteration is in blue, and the final one is in red. (g) Final iterations taken from (a) to (f), and (h) its magnified version in the range of 10^2 and 10^3.

the maximum of $\boldsymbol{\theta}$) as the initial value because as shown in Figure 6 the cost function is convex for the large values of $\boldsymbol{\theta}$ due to the regularizer $\|\boldsymbol{\theta}\|^2$ in Eq. (16), pulling the estimated threshold toward the optimal value. The bottom two plots show the cost function values of the final iterations of different initial values. The estimated thresholds $\hat{\theta}_i$ with different initial values are close to each other, however, the number of iterations and computation cost increase when a large initial value is used. Therefore, using too large values are not recommended. Figure 7 shows the cost function values and estimated thresholds of figure 6(c) to show the convergence of the entire optimization procedure over iterations. We observe that the energy and threshold converge appropriately.

Next, we show the cost function values with different scale parameter values λ in Figure 8. The top-left plot $\lambda = 10^{-1}$ indicates that the value is too large so that the cost function is over-regularized and only the trivial estimates was obtained. As λ getting smaller, minimum becomes prominent and the estimated threshold shifts toward larger values. Labeling results with different λ values will be shown in Figure 11 in the following section.

B. Labeling results on a synthetic dataset

Figures 9 show the results for a synthetic sub-dataset. In Felzenszwalb's segmentation, there are too many boundaries that do not fit the ground truth label. The results of SVM-MRF and CRF are not qualitatively and quantitatively better than the results of the proposed method. The performances of the MRF- and CRF-based approaches are highly dependent on the unary term. In other words, failures by SVM and TextonBoost have too much impact on performance. For this kind of small dataset, MRF- and CRF-based methods are not the best choice for achieving good performance. In contrast, the proposed method gives reasonable labeling results and better performance in terms of the Dice coefficient. Note that since 42,169 nodes (or samples) are used for training, the primal solver for SVM training is necessary. Figure 10 shows results for another sub-dataset shown in Figure 4 that contains large geometrical, scale, and contrast changes. In this result, three low-contrast images are used for training, and the remaining ones are used for testing. MRF and CRF fail, whereas the proposed method labels test images reasonably. Figure 11 shows labeling results of the proposed method with different scale parameter values λ. A large value of

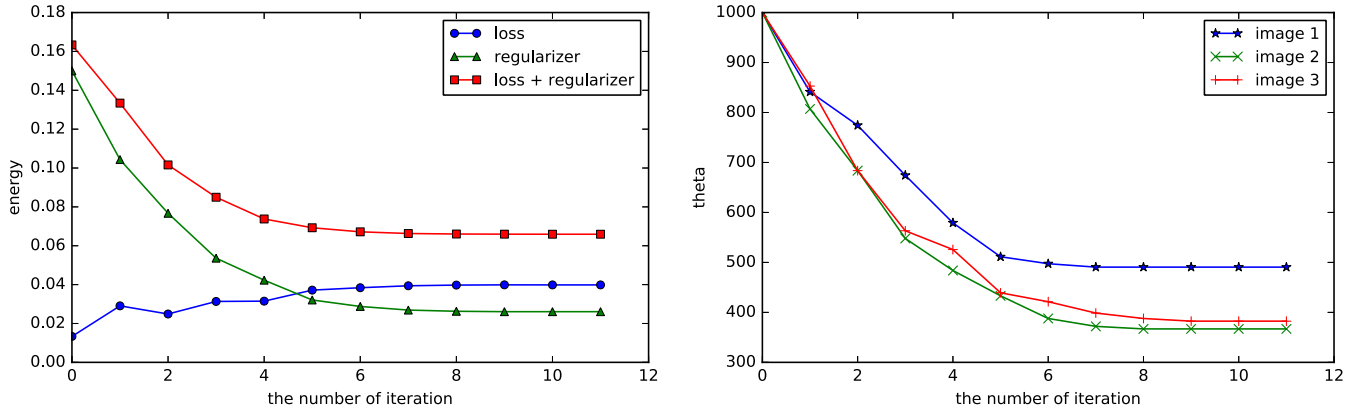


Fig. 7. Energies (left) and thresholds (right) at each iteration. The horizontal and vertical axes show the number of iterations and the energy (left) and estimated threshold (right), respectively.

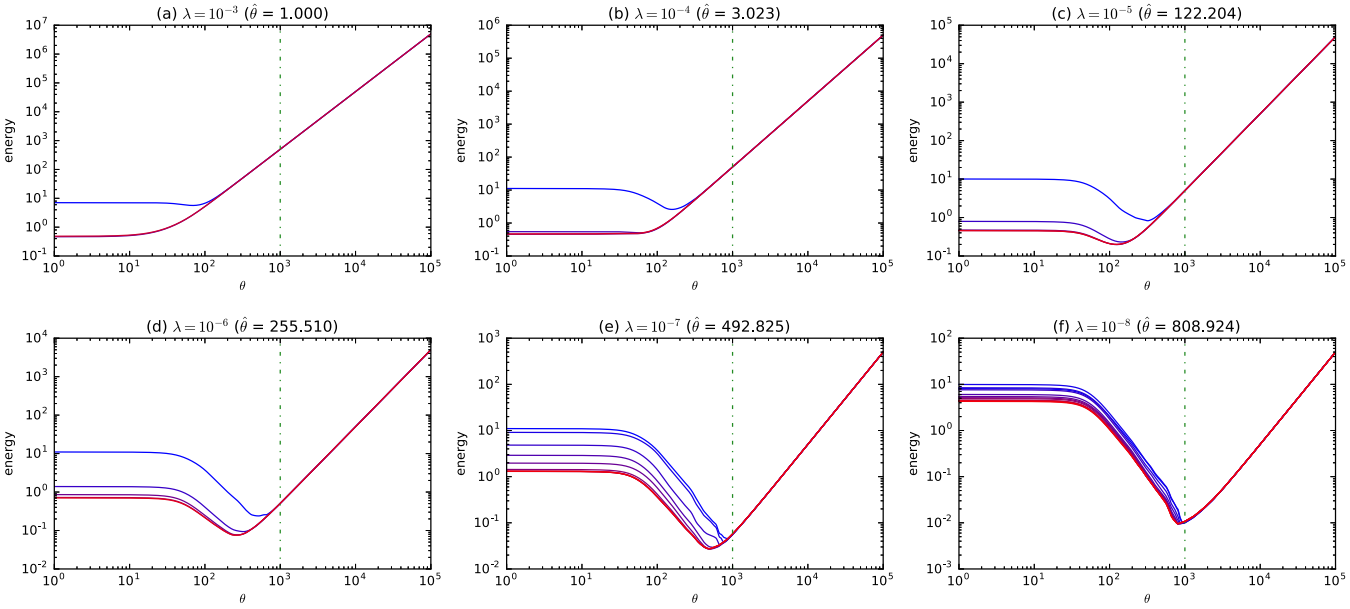


Fig. 8. Cost function values against the threshold over different iterations with different scale parameter values $\lambda = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$, and 10^{-8}. The horizontal and vertical axes show the threshold and corresponding cost function value, respectively. (a – f) Convergence with different scale parameter values. The initial value $\boldsymbol{\theta}_0 = 1000$ is indicated as vertical green lines. Different colors are used for different iterations; the first iteration is in blue, and the final one is in red.

$\lambda = 10^{-3}$ provides a small threshold value, and the boundary between foreground and background disappear. Smaller values $\lambda = 10^{-7}$ and 10^{-8}, provide larger threshold values, and foreground objects becomes smaller. A better labeling result can be obtained when an appropriate value of λ, in this case 10^{-5}. The value should be tuned as the accuracy of labeling results is sensitive to it. Results for other sub-datasets are shown in Figure 12 to 14. In these results, MRF and CRF provide some successful results but the proposed method is stable and better in most of the cases.

For quantitative evaluation, we compute Dice coefficients over different numbers of training images. In this experiment, we used a sub-dataset containing ten images by adding five images to the sub-dataset of Figure 4. Figure 15 shows the box plots of Dice coefficients for different numbers of training images. The overall Dice coefficients of MRF and CRF are lower than those of the proposed method. Even when nine images are used for training, the median of the Dice coefficients of MRF and CRF are approximately 0.6. Some of the Dice coefficients of CRF are extremely low. In contrast, the proposed method works effectively and provides adequately high Dice coefficients, even when only one training image was used.

Here, we show some failure cases of the proposed method, such as the examples shown in Figure 16. In these results, CRF outperforms the other methods. Two textures in the sub-dataset are of pebbles that are similar to each other, and the SITA feature used in the proposed method is invariant to this difference between geometrical, scale, and contrast changes. Therefore, the subtrees cannot be classified correctly, and we

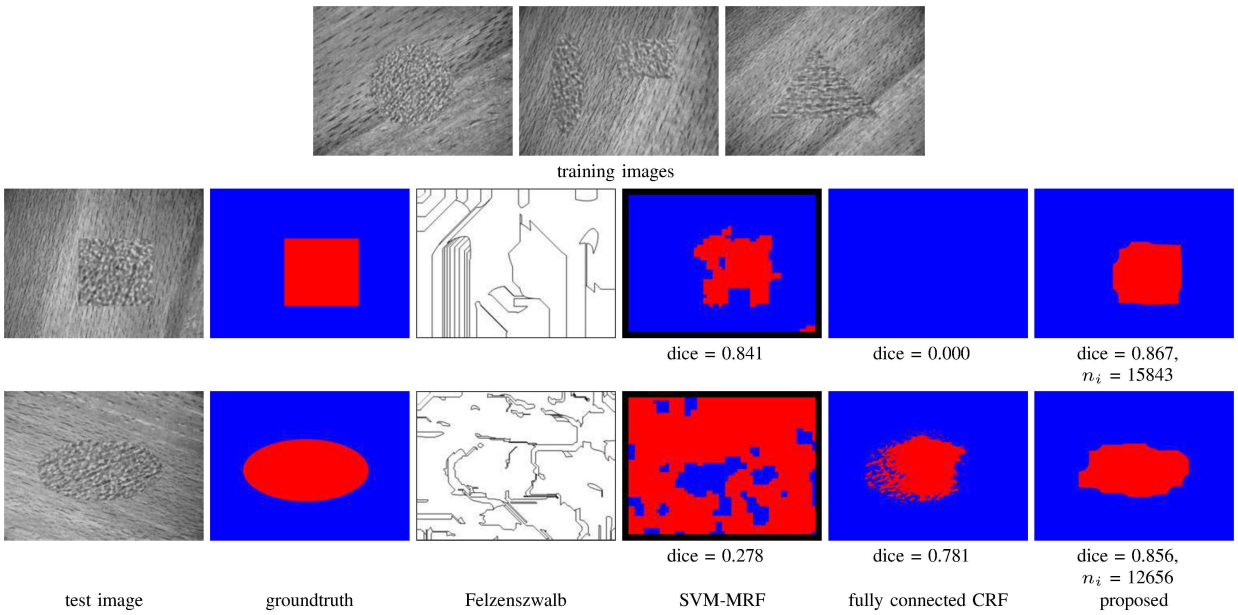


Fig. 9. Labeling results. The Dice coefficient and number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-6}, and the number of training samples is 42,169.

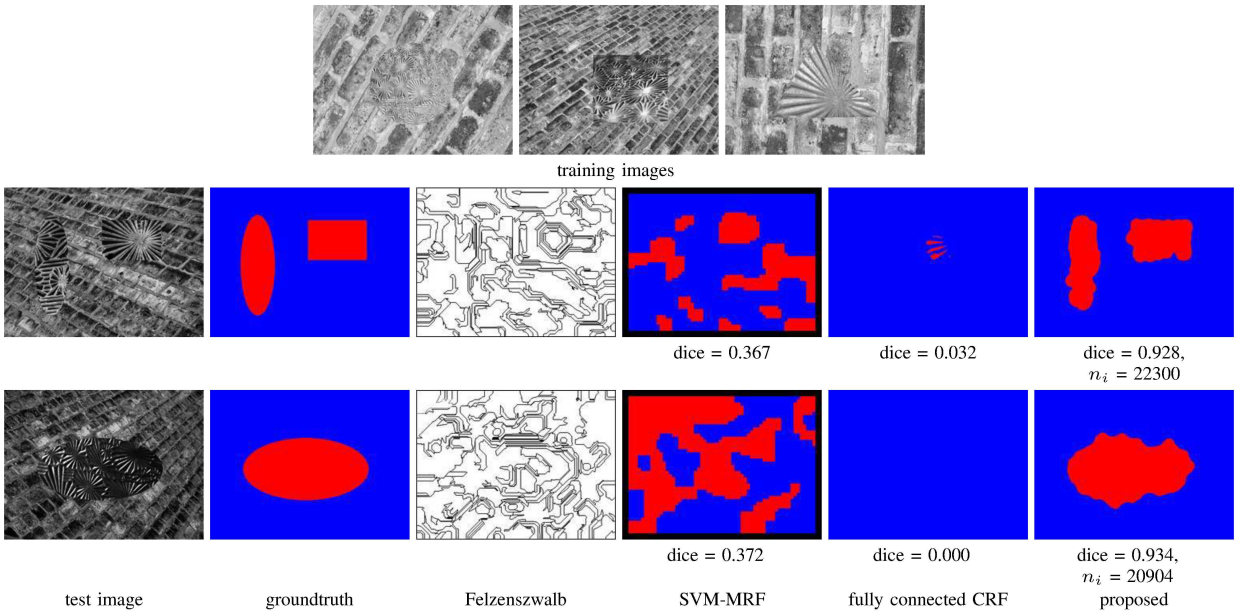


Fig. 10. Labeling results. The Dice coefficient and number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-5}, and the number of training samples is 59,701.

obtained poor labeling results.

Figure 17 shows further results wherein all of the methods do not work well. The possible reason for this failure is that the number of nodes n_i is relatively small compared to that used for the successful results, such as in those in Figure 9. In this result, we observe that the number of training samples or nodes in the tree of shapes must be greater than at least 10,000 per image in order to obtain satisfactory labeling results.

Regarding the computational time, our Python implementation of the proposed method takes approximately 200 s for training and approximately 30 s for labeling an image in the above experimental condition. Although we handle more than 10,000 training samples (or nodes), our method can be trained within a practical time.

C. Labeling results on the MSRC-21 dataset

Hereafter, we show labeling results on the MSRC-21 dataset [31]. This dataset consists of 591 images with ground truth labels of 21 object classes. It has 20 subsets according to main objects of the image shown in the center, such as cow, sheep, tree, car, and building. To evaluate the proposed method with a few training images, we selected two subsets having rich

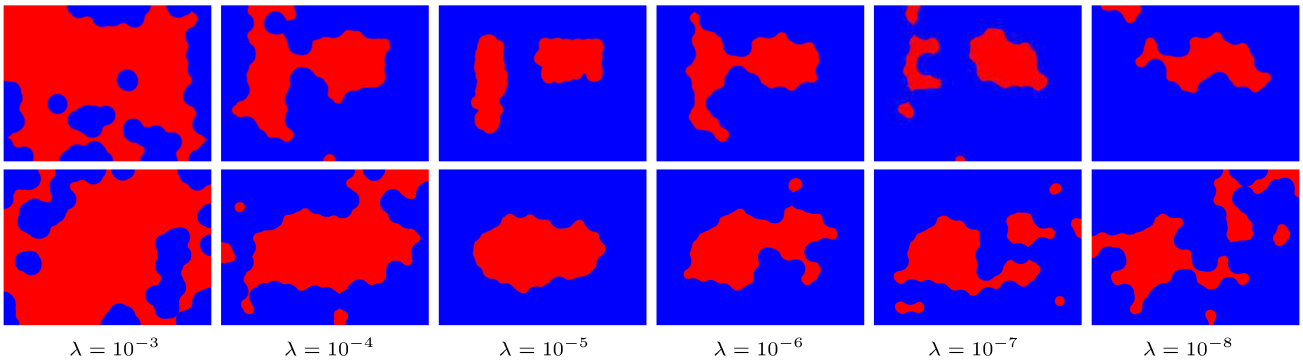


Fig. 11. Labeling results with different scale parameter values λ, with the same images with Figure 10.

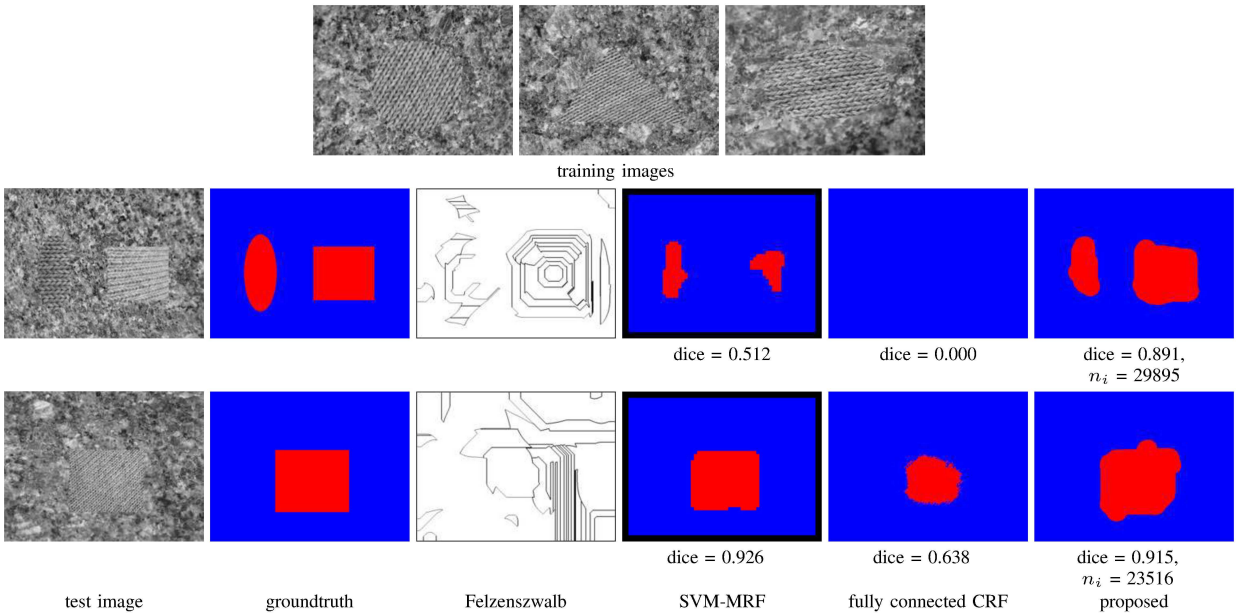


Fig. 12. Labeling results. The Dice coefficient and the number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-5}, and the number of training samples is 82,218.

texture contents; subset 2 (tree, grass, and sky) and 9 (sheep and grass). In each subset, the label of the main object of the subset (trees of subset 2, and sheep of subset 9) is used as foreground, and the others are as background.

In natural images, color is an important cue for segmentation. To demonstrate the proposed method for color natural images in this experiment, we use the following tree of shapes and histogram features. First, we adopted the tree of shapes for color images [14]. The color version of the tree of shapes is constructed by merging a set of trees of shapes of each color component (in this paper, we used RGB) based on shapes (connected components) and their inclusion relationships. For more details, please refer to [14]. Moreover, we introduce two new histogram features in addition to the SITA. One is a histogram of HSV color values. We construct histograms of each HSV component and concatenate to the original SITA histogram feature. The number of bins of each color component histogram is set to 50, then the total number of bins of the HSV histogram is 150. The other is a histogram of textons. We used 17 kernels used in the TextonBoost [31], and also 32 Gabor kernels [3]. The 49-dimensional responses of training images are clustered by the K-means algorithm. Then, each response is assigned to the nearest cluster center, or texton, and a histogram of these textons is created. The number of bins of the texton histogram is set to 200. In the experiments, we set sigmoid gain $\beta = 0.1$, initial values of thresholds to 1,000, and weight for aggregating histogram $w_{agg} = 0.8$. As a comparison, we used the fully connected CRF [35].

We show effect of the weight w_{agg} for aggregating children histograms. Figure 18 shows box plots for Dice coefficients over different values of w_{agg}. When $w_{agg} = 1.0$, dice coefficients are relatively lower than that of smaller w_{agg} values. With large values of w_{agg}, histograms are affected

[3]Used Gabor kernels consist of real and imaginary part of scales 3 and 5, frequencies 0.1 and 0.2, and rotations 0, $\pi/4$, $\pi/2$ and $3\pi/4$, which is decided experimentally. These Gabor kernels are convolved with the L component of the Lab color space.

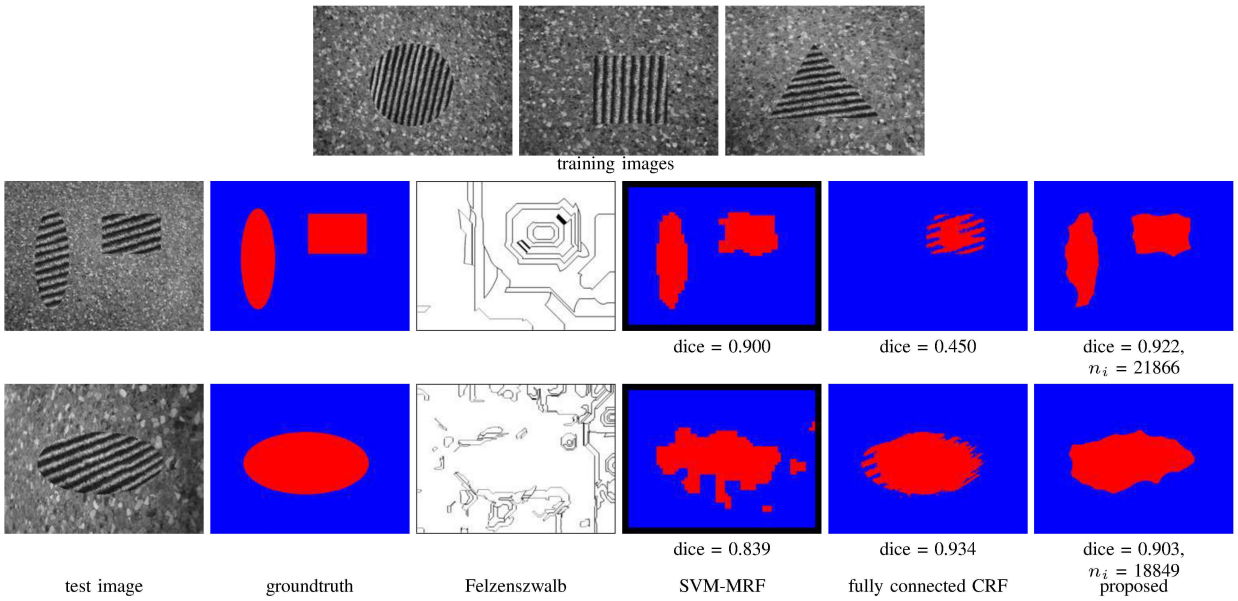


Fig. 13. Labeling results. The Dice coefficient and the number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-5}, and the number of training samples is 76,066.

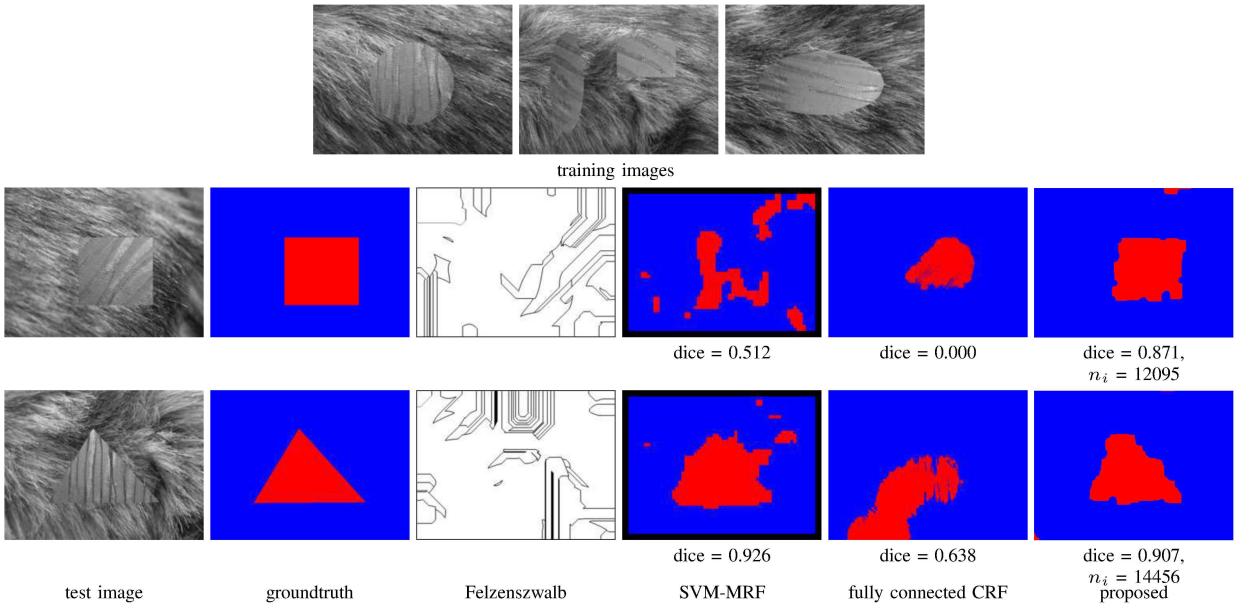


Fig. 14. Labeling results. The Dice coefficient and the number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-5}, and the number of training samples is 46,620.

by features of small children nodes, while smaller values of w_{agg} result in less discriminative histogram features. In this experiment, we empirically set $w_{agg} = 0.8$.

Figures 19 and 20 show segmentation results on subset 2 and 9. These results are obtained with five training images ($N = 5$). CRF fails to classify pixels correctly due to the small number of training images. Meanwhile, better results are obtained by the proposed method.

Figure 21 shows box plots for Dice coefficients over different number of training images from each of two subsets. N training images are randomly selected from a subset, and then 10 test images are randomly selected from the rest of the subset. For subset 2 (top row), the proposed method performs better than CRF when fewer than 7 images are used. With 9 and 10 training images, CRF works better as expected because typically CRF needs many training images. For subset 9 (bottom row), the proposed method consistently outperform CRF even with 10 training images used.

VI. Conclusion

In this study, we proposed a labeling method for texture image segmentation that works with a few training images. Our method is based on a tree of shapes and histogram features derived from the tree structure and selects optimal

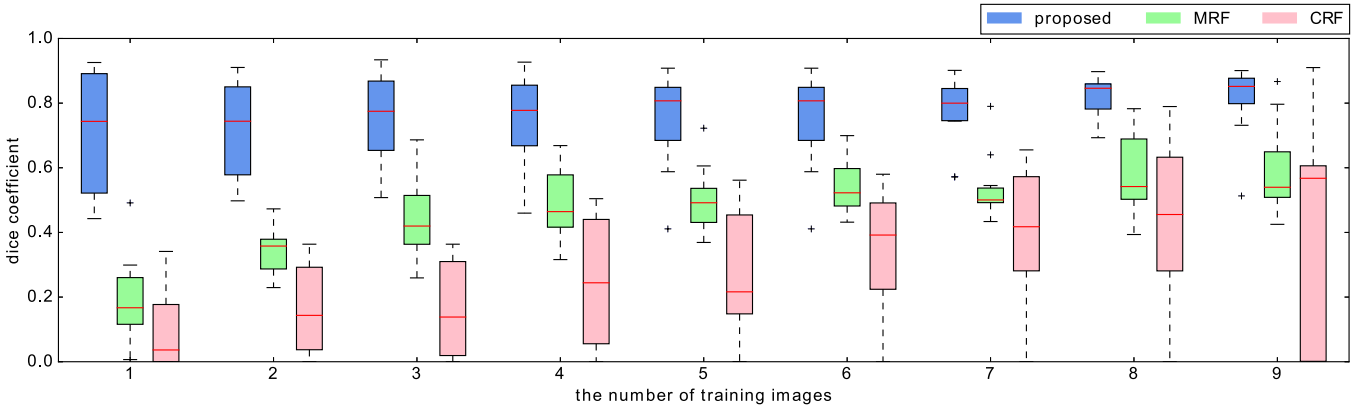


Fig. 15. Box plots for Dice coefficients over different numbers of training images. The horizontal and vertical axes show the number of training samples and the Dice coefficients, respectively.

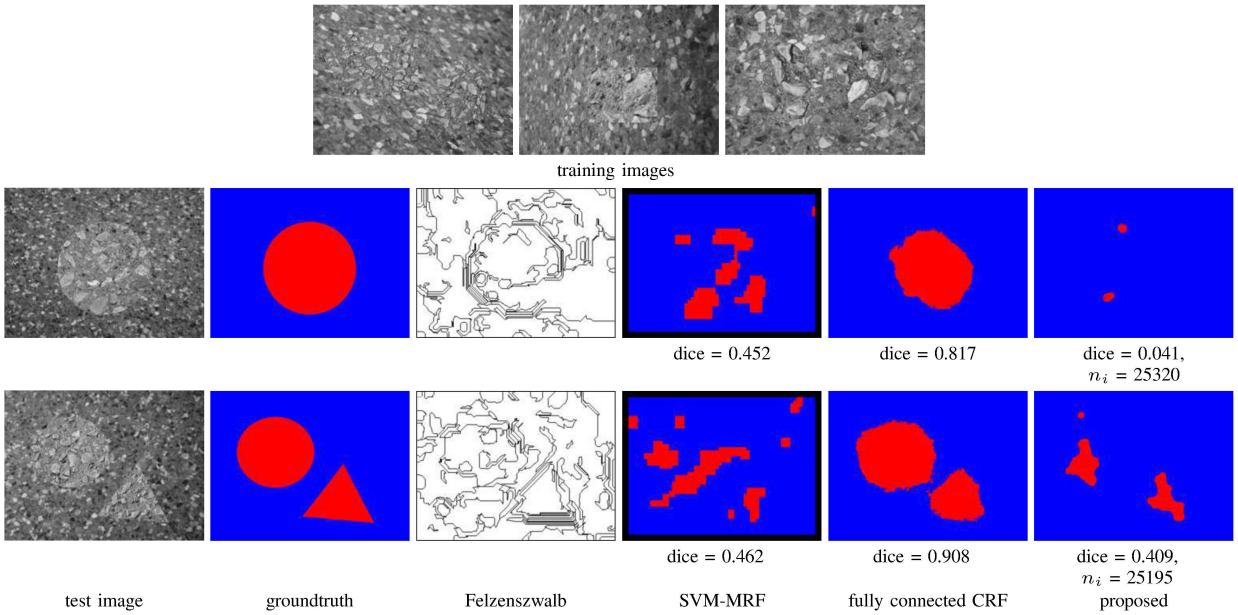


Fig. 16. Some failure labeling results. The Dice coefficient and number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-6}, and the number of training samples is 63,403.

discriminative subtrees for tree node classification. This is formulated as a joint optimization problem for estimating the threshold and classifier parameters and is solved using iterative block-coordinate decent. Then, images are labeled using the estimated parameters and the tree of shapes by classifying each node from the root node to leaf nodes and then mapping classification results into the corresponding pixels. We evaluated the proposed method on the two datasets: a synthetic texture image datasets based on the UIUC database and the MSRC-21 dataset. Experimental results show that the proposed method outperforms other methods and provides more reliable results. Our future work includes improving the sample weights, extending our method to a multi-class problem, and seeking a more effective form of the labeling procedure using the hierarchical structure.

ACKNOWLEDGMENT

The authors thank Tetsushi Koide, Shigeto Yoshida, Hiroshi Mieno, and Shinji Tanaka for their comments and advice. This work was supported in part by JSPS KAKENHI grant numbers JP14J00223 and JP16H06540.

REFERENCES

[1] S. G. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$," *Transactions of the American mathematical society*, vol. 315, no. 1, pp. 69–87, 1989.

[2] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, Jul 1989.

[3] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul 2002.

[4] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55–73, Jan 1990.

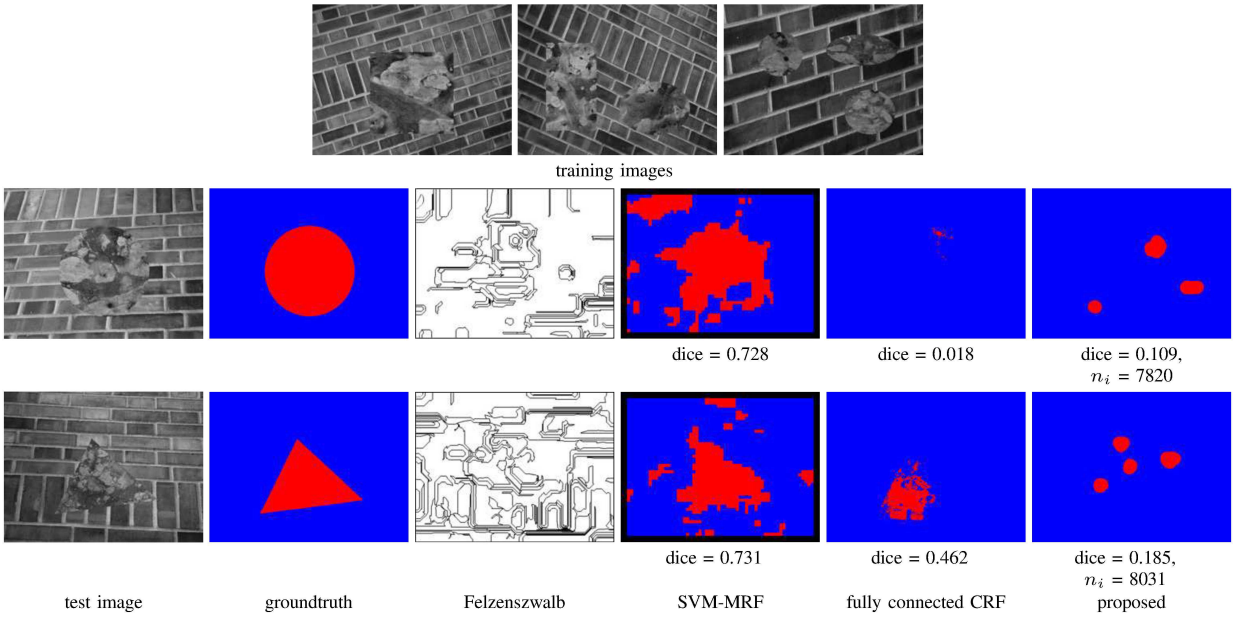
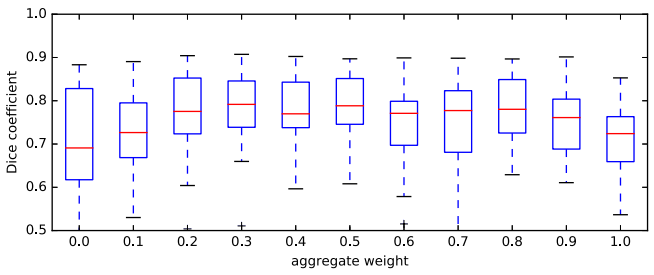
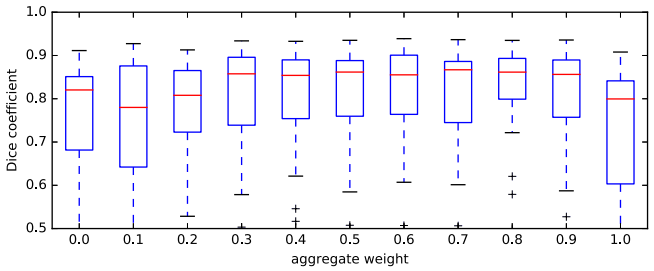


Fig. 17. Some failure labeling results. The Dice coefficients and number of nodes n_i are shown below the images. Red and blue represent each texture class, and the black of the SVM-MRF results represents a region that is unlabeled as a result of the boundary effect. λ is set to 10^{-5}, and the number of training samples is 27,045.



(a)



(b)

Fig. 18. Box plots for Dice coefficients over different w_{agg} on (a) subset 2 and (b) subset 9 of the MSRC-21 dataset. The horizontal and vertical axes show w_{agg} and the Dice coefficients, respectively. The number of training images N is fixed to 5.

[5] M. R. Turner, "Texture discrimination by gabor functions," *Biological Cybernetics*, vol. 55, no. 2, pp. 71–82, 1986.
[6] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.
[7] G.-S. Xia, J. Delon, and Y. Gousseau, "Shape-based invariant texture indexing," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 382–403, 2010.
[8] P. Monasse and F. Guichard, "Scale-space from a level lines tree," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 224 – 236, 2000.
[9] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, *A Quasi-linear Algorithm to Compute the Tree of Shapes of nD Images*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 98–110.
[10] R. Jones, "Connected filtering and segmentation using component trees," *Computer Vision and Image Understanding*, vol. 75, no. 3, pp. 215 – 228, 1999.
[11] L. Najman and M. Couprie, "Building the component tree in quasi-linear time," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3531–3539, Nov 2006.
[12] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, Apr 2000.
[13] J. Cousty and L. Najman, *Incremental Algorithm for Hierarchical Minimum Spanning Forests and Saliency of Watershed Cuts*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 272–283.
[14] E. Carlinet and T. Géraud, "Mtos: A tree of shapes for multivariate images," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5330–5342, Dec 2015.
[15] Y. Xu, T. Géraud, and L. Najman, *Two Applications of Shape-Based Morphology: Blood Vessels Segmentation and a Generalization of Constrained Connectivity*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 390–401.
[16] A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokládal, and N. Passat, "Filtering and segmentation of 3d angiographic data: Advances based on mathematical morphology," *Medical Image Analysis*, vol. 17, no. 2, pp. 147 – 164, 2013.
[17] B. Perret and C. Collet, "Connected image processing with multivariate attributes: An unsupervised markovian classification approach," *Computer Vision and Image Understanding*, vol. 133, pp. 1 – 14, 2015.
[18] G. Liu, G. S. Xia, W. Yang, and L. Zhang, "Texture analysis with shape co-occurrence patterns," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 1627–1632.
[19] C. He, T. Zhuo, X. Su, F. Tu, and D. Chen, "Local topographic shape patterns for texture description," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 871–875, July 2015.
[20] T. Hirakawa, T. Tamaki, B. Raytchev, K. Kaneda, C. Wang, L. Najman, T. Koide, Y. Kominami, S. Yoshida, and S. Tanaka, "Discriminative subtree selection for nbi endoscopic image labeling," in *The ACCV2016 workshop on mathematical and computational methods in biomedical imaging and image analysis (MCBMIIA2016)*, Nov 2016.

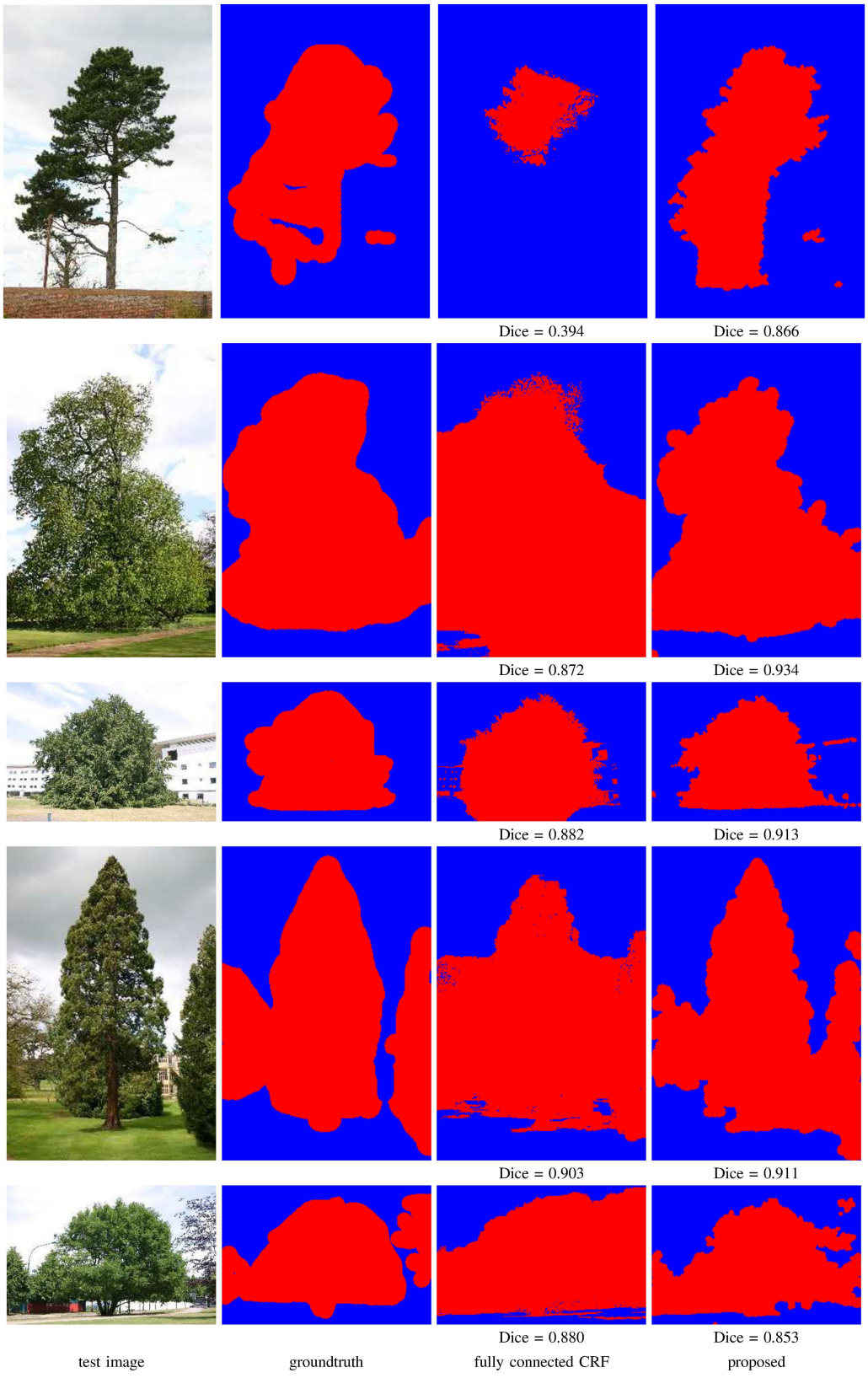


Fig. 19. Labeling results on a subset 2 of the MSRC-21 dataset. Dice coefficient is shown below the images.

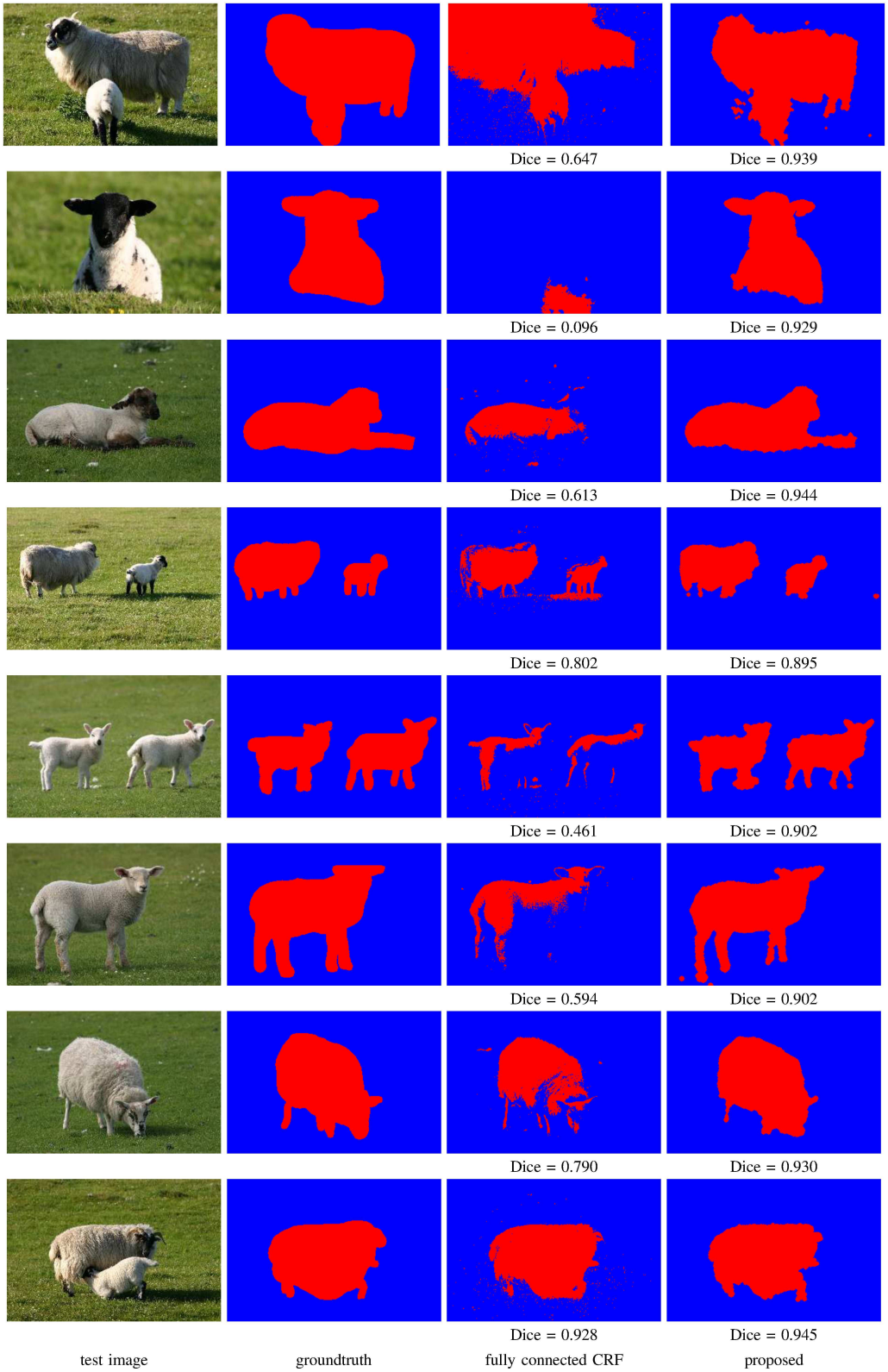
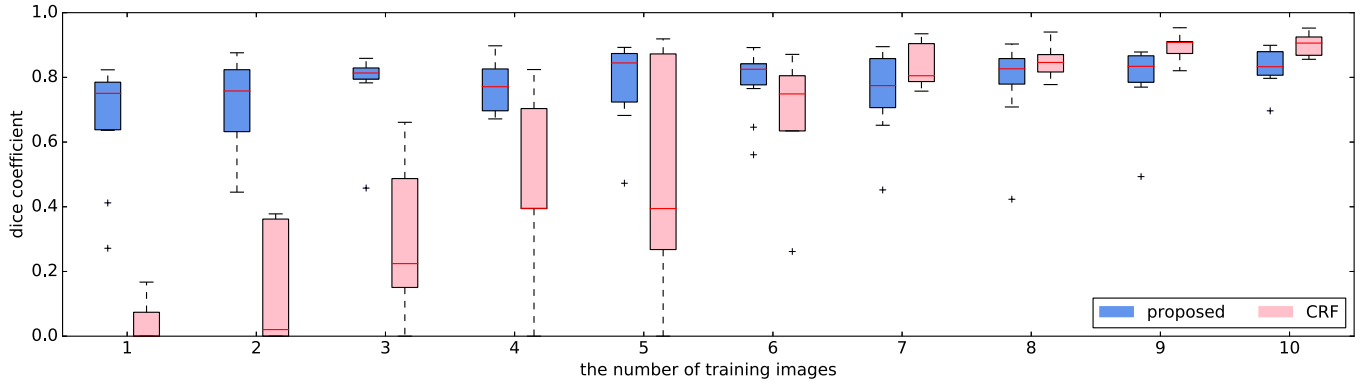
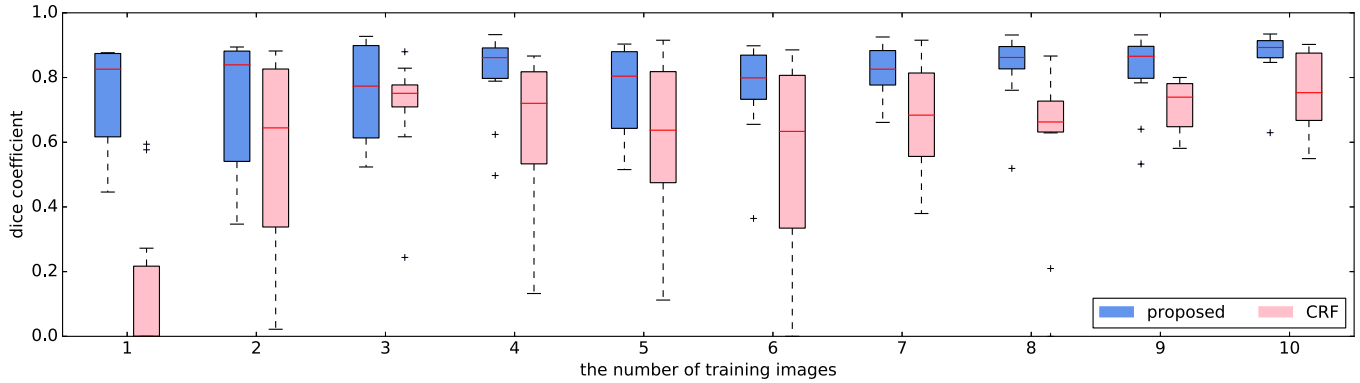


Fig. 20. Labeling results on a subset 9 of the MSRC-21 dataset. Dice coefficient is shown below the images.



(a)



(b)

Fig. 21. Box plots for Dice coefficients over different numbers of training images on (a) subset 2 and (b) subset 9 of the MSRC-21 dataset. The horizontal and vertical axes show the number of training images and the Dice coefficients, respectively.

[21] K. Held, E. R. Kops, B. J. Krause, W. M. Wells, R. Kikinis, and H. W. Muller-Gartner, "Markov random field segmentation of brain mr images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 878–886, Dec 1997.

[22] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm," *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45–57, Jan 2001.

[23] W. Tang, Y. Wang, and W. He, "An image segmentation algorithm based on improved multiscale random field model in wavelet domain," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 2, pp. 221–228, 2016.

[24] A. Voisin, V. A. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Classification of very high resolution sar images of urban areas using copulas and texture in a hierarchical markov random field model," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 1, pp. 96–100, Jan 2013.

[25] C. D'Elia, S. Ruscino, M. Abbate, B. Aiazzi, S. Baronti, and L. Alparone, "Sar image classification through information-theoretic textural features, mrf segmentation, and object-oriented learning vector quantization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1116–1126, April 2014.

[26] Z. Kato and T.-C. Pong, "A markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103 – 1114, 2006.

[27] J. Tighe and S. Lazebnik, *SuperParsing: Scalable Nonparametric Image Parsing with Superpixels*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 352–365.

[28] J. Tighe and S. Lazebnik, "Superparsing," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 329–349, 2013.

[29] T. Hirakawa, T. Tamaki, B. Raytchev, K. Kaneda, T. Koide, Y. Kominami, S. Yoshida, and S. Tanaka, "Svm-mrf segmentation of colorectal nbi endoscopic images," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2014, pp. 4739–4742.

[30] J. Shotton, J. Winn, C. Rother, and A. Criminisi, *TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–15.

[31] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[32] L. Bertelli, T. Yu, D. Vu, and B. Gokturk, "Kernelized structural svm learning for supervised object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 2153–2160.

[33] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, June 2004, pp. 46–46.

[34] M.-E. Nilsback and A. Zisserman, "Delving deeper into the whorl of flower segmentation," *Image and Vision Computing*, vol. 28, no. 6, pp. 1049 – 1062, 2010.

[35] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 109–117.

[36] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[37] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[39] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," *Artificial Intelligence*, pp. 1972–1979, 2009.
[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.
[41] F. Liu, G. Lin, and C. Shen, "CRF learning with CNN features for image segmentation," *Pattern Recognition*, vol. 48, no. 10, pp. 2983 – 2992, 2015, discriminative Feature Learning from Big Data for Visual Recognition.
[42] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1520–1528.
[43] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
[44] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983.
[45] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, Aug 2005.
[46] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
[47] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

Tsubasa Hirakawa received his B.E. and M.S. degrees in information engineering from Hiroshima University, Japan, in 2012 and 2013, respectively. He was a visiting researcher at ESIEE Paris, France, in 2014 and 2015, and is currently pursuing the Ph.D. degree at the Department of Information Engineering, Hiroshima University, Japan. His research interests include machine learning and medical image analysis.

Toru Tamaki received his B.E., M.S., and Ph.D. degrees in information engineering from Nagoya University, Japan, in 1996, 1998 and 2001, respectively. After being an assistant professor at Niigata University, Japan, from 2001 to 2005, he is currently an associate professor at the Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan. He was an associate researcher at ESIEE Paris, France, in 2015. His research interests include computer vision, image recognition, machine learning, and medical image analysis.

Takio Kurita received the B.Eng. degree from Nagoya Institute of Technology and the Dr. Eng. degree from the University of Tsukuba, in 1981 and in 1993, respectively. He joined the Electrotechnical Laboratory, AIST, MITI in 1981. From 1990 to 1991 he was a visiting research scientist at Institute for Information Technology, National Research Council Canada. From 2001 to 2009, he was a deputy director of Neuroscience Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). Also he was a Professor at Graduate School of Systems and Information Engineering, University of Tsukuba from 2002 to 2009. He is currently a Professor at Hiroshima University. His current research interests include statistical pattern recognition and its applications to image recognition. He is a member of the IEEE, the IPSJ, the IEICE of Japan, Japanese Neural Network Society, The Japanese Society of Artificial Intelligence.

Bisser Raytchev Bisser Raytchev received his Ph.D. in Informatics from Tsukuba University, Japan in 2000. After being a research associate at NTT Communication Science Labs and AIST, he is presently an assistant professor in the Department of Information Engineering, Hiroshima University, Japan. His current research interests include computer vision, pattern recognition, high-dimensional data visualization, and image processing.

Kazufumi Kaneda is a professor in the Department of Information Engineering, Hiroshima University, Japan. He joined Hiroshima University in 1986. He was a visiting researcher in the Engineering Computer Graphics laboratory at Brigham Young University in 1991. Kaneda received the B.E., M.E., and D.E. in 1982, 1984 and 1991, respectively, from Hiroshima University. His research interests include computer graphics, scientific visualization, and image processing.

Chaohui Wang received his Ph.D. in applied mathematics and computer vision from Ecole Centrale Paris, Châtenay-Malabry, France, in 2011. After that, he was postdoctoral researcher at the Vision Lab of University of California, Los Angeles, CA, USA (from January 2012 to March 2013), and at the Perceiving Systems Department, Max Planck Institute for Intelligent Systems, Tbingen, Germany (from March 2013 to August 2014). Since September 2014, he is Maître de Conférences at Université Paris-Est, Marne-la-Vallée, France. His research interests include computer vision, machine learning, image processing, and medical image analysis.

Laurent Najman received the Habilitation from the University of Marne-la-Vallée, in 2006, the PhD in applied mathematics from Paris-Dauphine University in 1994, and the Engineering degree from the École des Mines de Paris, in 1991. After 10 years of research work on image processing and computer graphics problems in several companies, he joined ESIEE Paris in 2002, where he is currently a professor. His research interests include discrete mathematical morphology and discrete optimization.