Supervised and unsupervised segmentation using superpixels, model estimation, and Graph Cut

Jiří Borovec[a,*], **Jan Švihlík**[a,b], **Jan Kybic**[a], **David Habart**[c]
[a]Biomedical Imaging Algorithms Group, CMP, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, Prague, Czech Republic
[b]Department of Computing and Control Engineering, Faculty of Chemical Engineering, University of Chemistry and Technology, Technická 5, Prague, Czech Republic
[c]Institute for Clinical and Experimental Medicine, Vídeňská 1958/9, Prague, Czech Republic

Abstract. Image segmentation is widely used as an initial phase of many image analysis tasks. It is often advantageous to first group pixels into compact, edge-respecting superpixels, because these reduce the size of the segmentation problem and thus the segmentation time by an order of magnitudes. In addition, features calculated from superpixel regions are more robust than features calculated from fixed pixel neighborhoods. We present a fast and general multi-class image segmentation method consisting of the following steps: (i) computation of superpixels; (ii) extraction of superpixel-based descriptors; (iii) calculating image-based class probabilities in a supervised or unsupervised manner; and (iv) regularized superpixel classification using Graph Cut. We apply this segmentation pipeline to five real-world medical imaging applications and compare the results with three baseline methods — pixel-wise Graph Cut segmentation, supertexton-based segmentation, and classical superpixel-based segmentation. On all datasets, we outperform the baseline results. We also show that unsupervised segmentation is surprisingly efficient in many situations. Unsupervised segmentation provide similar results to the supervised method, but does not require manually annotated training data, which is often expensive to obtain.

Keywords: segmentation, superpixels, Graph Cut, Gaussian Mixture Model, Drosophila, Langerhans Islets .

*Jiří Borovec, jiri.borovec@fel.cvut.cz

1 Introduction

Image segmentation is a key image processing task with a vast range of applications, especially in biomedical imaging,[1–3] including organ segmentation,[4] object detection and extraction,[5] or preprocessing for feature extraction or object tracking.

Segmentation techniques[6–8] have evolved from considering pixels independently to using information from local neighborhoods via Markov Random Fields (MRF) and graph representations.[9–12] Taking pixel neighborhoods into consideration improves the segmentation performance, at the cost of increased computational complexity.

With the steadily growing number and spatial resolution of the images to be processed, we need methods which are simultaneously fast and accurate. The size of microscopy images may be up to 10^{10} pixels for a histology slice. Current segmentation techniques can be very slow, specially if their computational complexity is not linear with respect to the number of pixels, as is the case e.g. for many MRF optimization techniques. One way to address this issue is to work on superpixels[13] instead of pixels. Superpixels group neighboring pixels together on the basis of their spatial and color similarity, and can be regarded as edge-preserving downsampling. Assuming that class boundaries coincide with superpixel boundaries, this allows image segmentation to be performed on the superpixel representation,[14–16] which leads to huge computational savings. At the same time, it is possible to use more reliable superpixel-based descriptors than standard fixed-neighborhood descriptors.

1.1 Previous work

Existing superpixel-based segmentation methods[14,17–20] usually consist of 3 steps: (i) compute suitable superpixels, preserving required details; (ii) extract superpixel appearance features ; (iii) use a classifier to assign labels to superpixels and extend them to the whole image. The classification can be performed by a standard classifier in a supervised[14,17,19,20] or unsupervised manner.[15,16,21,22] The extracted features include color,[15,22] texture,[17,19] and shape.[14] It has been shown that superpixels can perform better than direct pixel-level segmentation, with lower demands on time and resources.

In this work, we extend this classical segmentation pipeline by spatial regularization using Graph Cut[10] to encourage spatial continuity. The most similar previous work was done by Kitrungrotsakul,[23] Kitrungrotsakul also uses superpixels and Graph Cuts, but is nevertheless focused on a single application — binary single object segmentation. Briefly some other work in this field: Puniyani[22] uses a much simpler superpixel extraction method and much simpler edge terms. Ye[24] uses mean shift clustering, and Hsu[25] uses region merging, while Wang[26] considers long-range similarity-based interactions instead of interactions based on neighborhoods.

Our key contributions with respect to[23] are as follows: First, we formulate the task as general multi-class segmentation. Second, we propose a new formula for the edge weights based on differences in model-based class probabilities. In experiments, this approach is shown to perform better than traditional methods. Third, our method incorporates both unsupervised and supervised modeling. Fourth, we include a comprehensive set of experiments that shows that our method can be applied to four different applications with little adjustment.

1.2 Organization of this paper

This paper is organized as follows. Section 2 discusses the proposed methods. Section 3 presents the applications, datasets and metrics used for the evaluation, as well as a description of the baseline methods. Section 4 contains experiments on all datasets, and results.

2 Method

Let us have an input image $I : \Omega \to \mathbb{R}^m$ defined on a pixel grid $\Omega \subseteq \mathbb{Z}^d$. We shall decompose the image pixels into superpixels, $\Omega = \bigcup_{s \in S} \Omega_s$, where Ω_s denotes pixels belonging to a superpixel s and S is a set of all superpixels. We will then try to find a superpixel segmentation function $Y : S \to L$, where L is a set of class labels. Superpixel segmentation Y can easily be interpolated to all pixels by assigning its label to all pixels within a superpixel, yielding a pixel-level segmentation function $Y_\Omega : \Omega \to L$. The scheme of the complete segmentation pipeline is presented in Figure 1.

2.1 Minimization problem

For each superpixel $s \in S$, we compute a vector of (color and texture) features $\boldsymbol{x}_s \in X$. We find the superpixel classes $y_s = Y(s)$ from the maximum a posteriori (MAP) estimate

$$Y^* = \arg\max_Y \; P(Y|X) = \arg\max_Y \; \frac{p(X|Y) \cdot P(Y)}{p(X)} \tag{1}$$

where $P(Y)$ is the *a priori* probability of a specific segmentation (of all pixels) regardless of the descriptors, and $p(X|Y)$ is the conditional density of the set X of all descriptors $\boldsymbol{x}_s$ given the labels

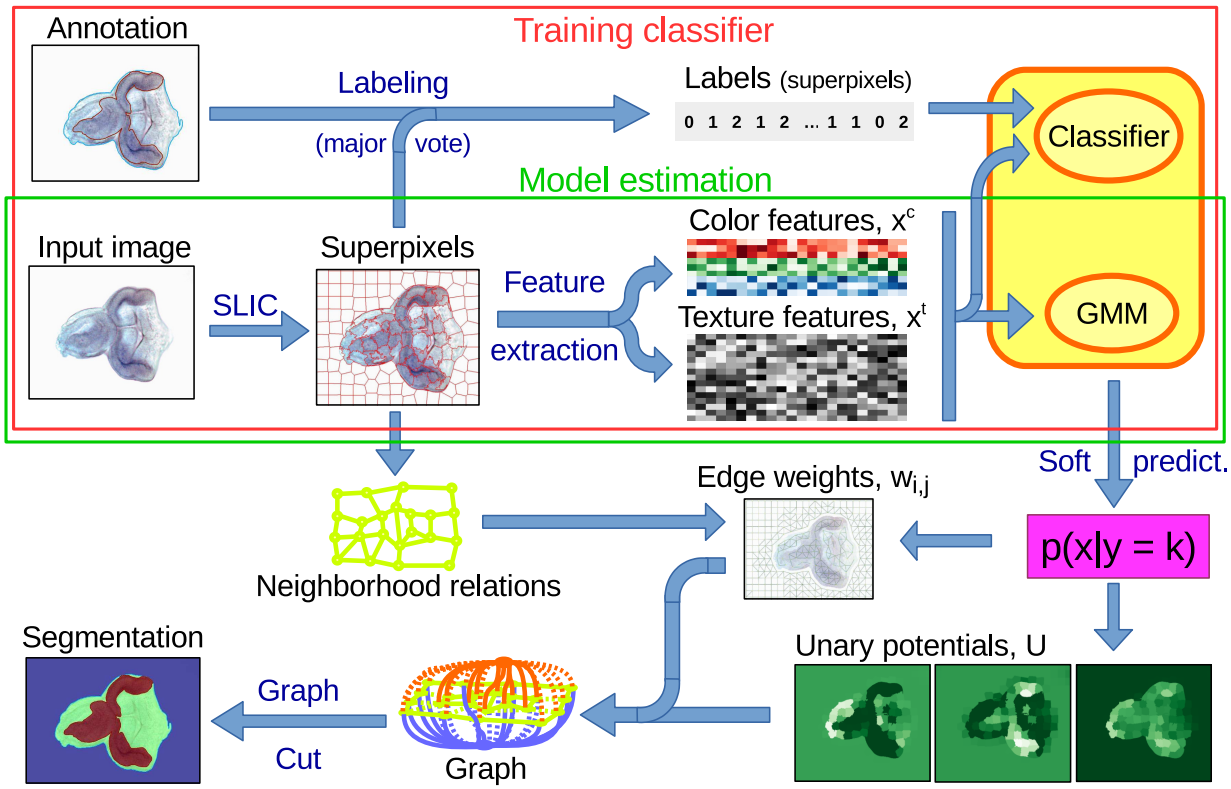


Figure 1 The scheme of the complete image segmentation pipeline with supervised and unsupervised learning.

Y, and $p(X)$ is the marginal probability density function of X, which is constant for given image I and independent on Y.

We express the spatial dependency of superpixel labels using a Markov random field and factorize the term $P(Y)$ as follows

$$P(Y) = \prod_{s \in S} h(y_s) \cdot \prod_{(i,j) \in \mathcal{N} \subseteq S^2} R(y_i, y_j) \tag{2}$$

The first term, $h : L \to \mathbb{R}$, is the prior probability of each class, independent of position. The second term $R(y_i, y_j)$ describes the relation between the classes of neighboring superpixels, favoring neighboring superpixel classes to be the same. The R can be learned from reference segmentation or can be designed by the user. Because superpixel features $\boldsymbol{x}_s$ are conditionally independent given Y, equation (1) can be written as follows:

$$Y^* = \arg\max_Y \prod_{i \in S} \big(p(\boldsymbol{x}_i | y_i) \cdot h(y_i)\big) \cdot \prod_{(i,j) \in \mathcal{N}} R(y_i, y_j) \tag{3}$$

Applying the logarithm to equation (3), we obtain the widely used Potts model which can be solved by Graph Cuts,[9] where we minimize the sum of unary and pairwise potentials

$$Y^* = \arg\min_Y \sum_s \underbrace{-\log\left(p(\boldsymbol{x}_s | y_s) \cdot h(y_s)\right)}_{U_s(y_s)} + \sum_{(i,j) \in \mathcal{N}} \underbrace{-\log R(y_i, y_j)}_{\beta w_{i,j} B(y_i, y_j)} \tag{4}$$

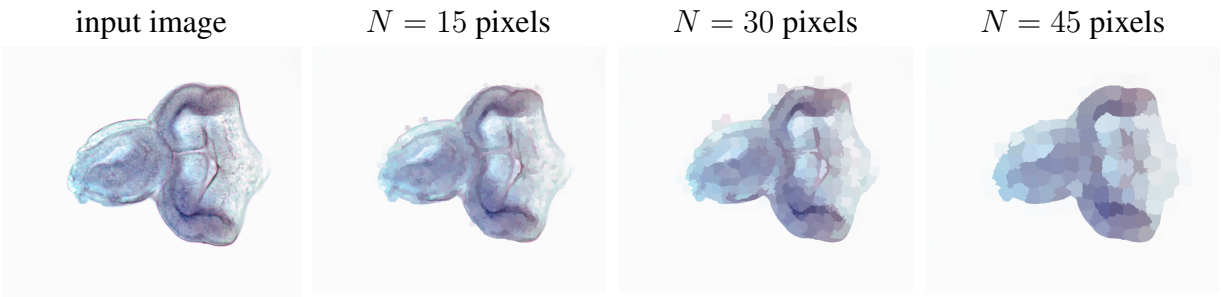


Figure 2 We show the input image (a) and its approximation using mean colors of the superpixels. The SLIC superpixel regularization is $r = 0.2$, and the sizes are 15, 30 and 45 pixels.

where we have factorized the regularization into a global coefficient β, position-dependent weight w_{ij} and a label-dependent part B. The unary term $U_s(y_s)$ represents the observations (image) and the a priori class probabilities, and the binary potential $B : L^2 \to \mathbb{R}$ leads to spatial regularization. If training data for estimating R is not available, we set $B(k, l) = [\![k \neq l]\!]$.

2.2 Superpixel clustering

We use the Simple Linear Iterative Clustering[27] (SLIC) algorithm to calculate the superpixels, so that the superpixels are compact both in space and in color. The SLIC algorithm is an adaptation of a widely-used k-means clustering algorithm. It uses a combined color and spatial distance

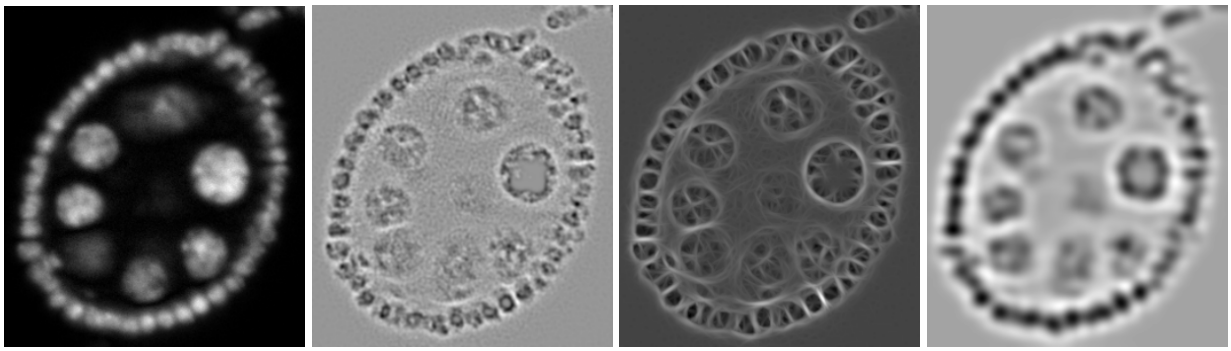
$$D = d_c + \frac{m}{N^2} \cdot d_s \tag{5}$$

The color distance d_c is a Euclidean distance calculated in the CIELAB color space, and d_s is a Euclidean distance measured in pixels. The superpixel centers are initially placed on a grid with spacing N, and the search is limited to a $2N \times 2N$ region, which greatly reduces the computational complexity. Instead of the user-provided parameter m, which can be seen as compensation for superpixel size N, we will use a relative regularization parameter[28] $r \in (0, 1)$, with $m^2 = N^3 r^2$.

2.3 Feature space

We create a feature vector $\boldsymbol{x}_s = \begin{bmatrix} \boldsymbol{x}_s^c & \boldsymbol{x}_s^t \end{bmatrix}$ for each superpixel s, containing color features $\boldsymbol{x}_s^c$ and texture features $\boldsymbol{x}_s^t$.[17] The features (or descriptors) are normalized element-wise as $\overline{\boldsymbol{x}} = \frac{\boldsymbol{x} - \boldsymbol{\mu}_x}{2(3\boldsymbol{\sigma}_x + 1)}$ where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are vectors of the means and standard deviations of each component over the whole dataset.

Color features For each color channel I^c, we define a feature vector $[\mu_s(I^c), \sigma_s(I^c), E_s(I^c), M_s(I^c)]$, where the components are the mean, the standard deviation, the energy, and the median of that color channel over the superpixel. To get the full color feature vector $\boldsymbol{x}_s^c$, the vectors for all three color channels are concatenated. The mean color feature is illustrated in Figure 2. A different color spaces or a combination thereof can be used (e.g. RGB, HSV, Lab).[18] In our experiments, we used RGB color space.



(a) (b) (c) (d)

Figure 3 Sample gray-scale image (Drosophila ovaries) *(a)* and responses to some Leung-Malik filter banks *(b-d)*.

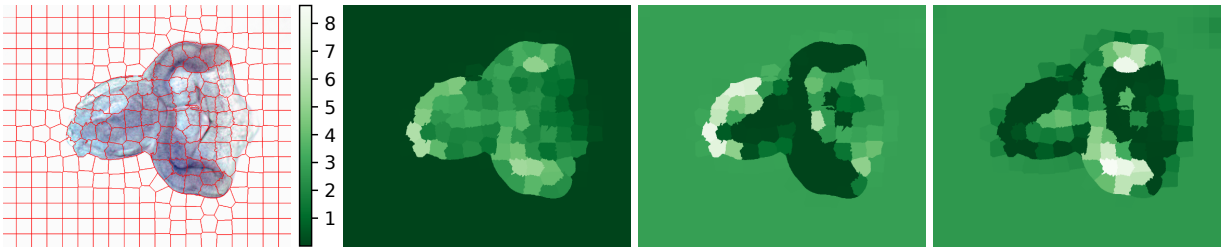
Texture features The Leung-Malik (LM)[29] filterbank is a multi-scale, multi-orientation filterbank with 48 filters in total. It consists of the first and second derivatives of a Gaussian at 6 orientations; 8 Laplacians of Gaussian filters; and 4 Gaussians. To characterize the texture, we compute the responses $F_j^c = I^c * \mathtt{LM}_j$, for each color channel independently[17] (see Figure 3 for examples). To achieve rotation invariance, we take for each pixel the maximum response over all orientations, determining the orientation and thus reducing the number of features per channel to 18. As above, for each filter and each color channel, we calculate the mean, the standard deviation, the energy, the median, and also the mean gradient amplitude $G_s(F_j^c) = {}^1/_{|\Omega_s|} \sum \|\nabla F_s^{c,j}\|$. To get the feature vector $\boldsymbol{x}_s^t$, the vectors $[\mu_s(F_j^c), \sigma_s(F_j^c), E_s(F_j^c), M(F_j^c), G_s(F_j^c)]$ are concatenated for all j and for all color channels c.

2.4 Multi-class modeling

For each class $k \in L$, we define a model $p(\boldsymbol{x}|y_s = k)$ for the feature vector probability density, with parameters θ_k.

Gaussian mixture model In the unsupervised case, we use the Gaussian Mixture Model (GMM).[22] We assume that for each class, the distribution on $\boldsymbol{x}_s$ is normal: $p(\boldsymbol{x}_s) = \sum_{k \in L} \rho_k N(\boldsymbol{x}_s; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $y_s = k$. The overall probability $p(\boldsymbol{x}_s)$ for unknown y_s is therefore a mixture, with mixing probabilities $h(k)$, used in (3) and (4). Parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and $h(k)$ are estimated using the Expectation-Maximization (EM) algorithm.[21] The basic scenario is an estimation of GMM for each image independently. This can be used in the situation when each segmented image is different from the others. Another case is estimating GMM over a set of images where we expect a similar appearance model. Both approaches are explored in Section 2.6.

Classifier-based model In the supervised case, we train a standard classifier such as Random Forest, logistic regression, k-Nearest Neighbours (KNN), Support Vector Machine (SVM) or Gradient Boost on a set of training examples. All these classifiers can produce the probability $p(\boldsymbol{x}_i|y_s)$, which we require in equation (1). The class labels are usually provided pixel-wise in the training data. We convert pixel labels to superpixel labels by taking the majority class. Superpixels, where less than 98 % belong to a single class are ignored — this way we lose less than 5% of the training instances while avoiding misleading training data.



(a) (b) (c) (d)

Figure 4 Visualization of superpixels *(a)* colored according to the computed unary potential U^k for classes $k = 0, 1, 2$ where the intensity of the green color reflects the potential. The class models were estimated from the image as GMM where images *(b-d)* are background, gene activation and the non-activated disc, respectively.

2.5 Graph Cut — Potentials

Following (4), we set the unary potentials U_s^k. The visualization is presented in Figure 4 for an example image where the GMM for 3 classes was learned using the EM algorithm.

From the training data, we can learn the probability $p(k, l)$ of two superpixels belonging to classes k and l being neighbors. Then we set the penalty as $B(k, l) = -\log(p_{k,l})$. In the case of non-availability of labeled training data, we turn to the use of unsupervised learning e.g. the Gaussian mixture model. We create the $B(k, l)$ matrix uniformly with zeros on the diagonal and with ones elsewhere.

2.6 Graph Cut — Edge weights

Let us turn our attention to the weights w_{ij} in (4), which we shall call edge weights, as they correspond to edges between nodes of the graph in the Graph Cut method. In our case, the nodes are superpixels. The idea is to penalize class boundaries between similar superpixels, and so the potentials can be seen as global pasteurization and the weighted edges can be seen as local regularization. Taking advantage of the rich superpixel descriptors, we show several formulas for w_{ij}, three from the literature and one new formula. The weights are normalized by the Euclidean distance $d_E(i, j)$ between the two object,[10] in our case superpixel centers $i, j \in S$, to compensate for nonuniformly distributed superpixels. The $\bar{d}_E^S$ is the mean Euclidean distance between all neighbouring superpixels $d_E(i, j)$.

Spatial weighting is given by $w_{i,j} = \frac{\bar{d}_E^S}{d_E(i,j)}$. It is the simplest case and it can be seen as a distance weighted Potts model.

Color weighting Normalized color distance[10,30] is also often used, assuming that superpixels with similar colors should be grouped together. We use the Euclidean distance $d_E(\cdot, \cdot)$ of the superpixel mean colors $\bar{I}_s$ in the RGB color space, normalized by its standard deviation σ_c over all superpixel colours.

$$w_{i,j} = \exp\left(-\frac{d_E(\bar{I}_i, \bar{I}_j)}{2\sigma_c^2}\right) \cdot \frac{\bar{d}_E^S}{d_E(i, j)} \tag{6}$$

We have observed that this approach works well for color images. For images without significant color differences, it has similar values as a spatial edge weight above.

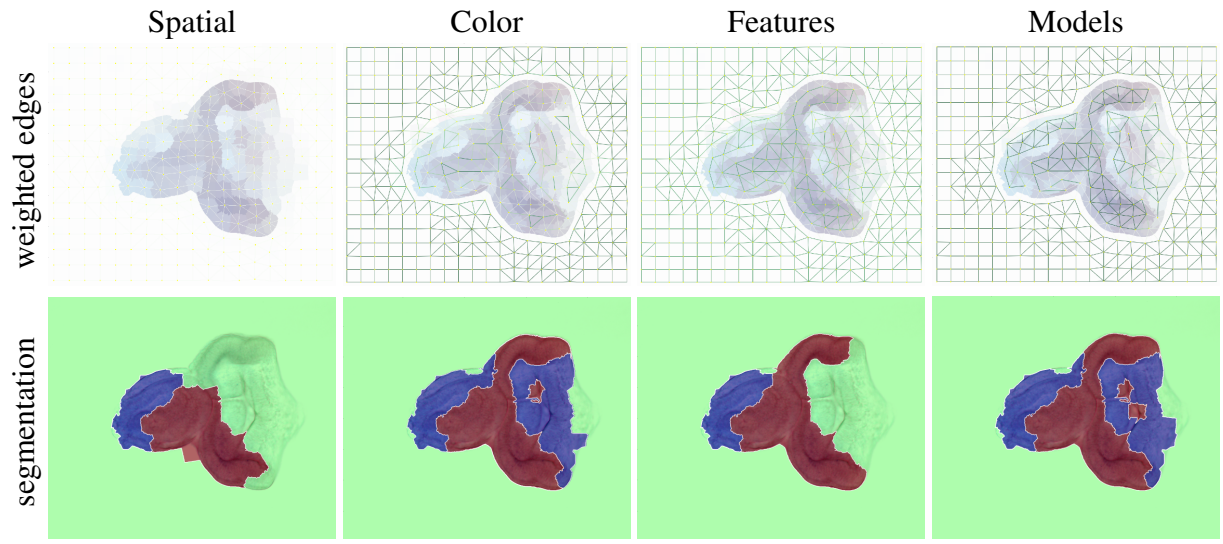


Figure 5 The edge weights $w_{i,j}$ represent the intensities of the green lines *(top row)* with the resulting segmentation *(bottom row)*. The same U and B potentials are used. The reference segmentation is presented in Section 3.1.2.

Feature weighting Color distance is a special case of a distance between feature vectors. Another option is to take the Manhattan distance $d_M(\cdot,\cdot)$ between the complete feature vectors[26] $\boldsymbol{x}_s$, normalized by their standard deviation σ_X over all superpixel features. We use Manhattan distance because the order of elements in feature vectors may be arbitrary.

$$w_{i,j} = \exp\left(-\frac{d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)}{2\sigma_X^2}\right) \cdot \frac{\bar{d_E^S}}{d_E(i,j)} \tag{7}$$

The disadvantage of feature weighting is that it may give too much weight to irrelevant features.

Model weighting. A new edge weighting that we propose is based on the comparison of a posteriori class probabilities derived from the learned model. The rationale is that the model 'knows' what is important for a particular application, by translating the high-dimensional feature vector into a low-dimensional vector of $|L|$ probabilities. We compute the l_∞ (Tchebychev) distance $d_T(\cdot,\cdot)$, even other metrics can be used, between the vectors of (non-normalized) class probabilities $p_s^k = p(x_s|y_s = k) \cdot h(y_s) \propto P(y_s = k|x_s)$. The probability differences d_T are normalized by their standard deviation σ_p over all distances.

$$\begin{aligned} d_T(p_i, p_j) &= \max_k \left(|p_i^k - p_j^k|\right) \\ w_{i,j} &= \exp\left(-\frac{d_T(p_i, p_j)}{2\sigma_p^2}\right) \cdot \frac{\bar{d_E^S}}{d_E(i,j)} \end{aligned} \tag{8}$$

Figure 5 shows examples of the use of different edge weights. We see that the newly proposed model weighting method works very well. See also Section 4.2.

3 Datasets and baseline methods

We shall compare the performance of our method with existing algorithms for five different applications, described below. First, we introduce the datasets and their usage in biomedical imaging. Then, we describe the baseline methods and the evaluation metrics.

3.1 Datasets and applications

3.1.1 Langerhans islets

Transplantation of Langerhans islets is a promising treatment for type I diabetes. The first applications is segmenting insulin-producing Langerhans islets[1] from microscopy images into three classes (islets, exocrine tissue, and background), which is required for evaluating the graft quality before transplantation. The aim of this segmentation is to make automatic measurements of the islet contents and of the purity of the samples, i.e. the ratio between the islet and the exocrine tissue area. Here we only compare the purity results. Figure 6 shows four examples of these images with expert segmentations.

Human islets were isolated from cadaver donors according to the protocol published in Ricordi et al.[31] There is a group of images consisting of 46 images from 4 independent donors, together with complete annotation, and the purity was assigned by a medical expert. The images were acquired using two different stereo microscopes (Olympus SZ60) equipped with the same type of camera, but using different settings. We applied color normalization and nonuniform background illumination compensation.[32]

The baseline method is called NPA,[33] and it is a specialized clustering method for segmenting Langerhans islets and exocrine tissue. It is based on over-clustering and cluster merging in HSV color space with a small number task specific parameters. Purity computed from a segmentation is compared with Expert Visual Assessment (EVA),[33] which is the mean of the visual estimates from five experts. We calculate the Pearson correlation coefficient and the mean absolute relative error $\mathrm{MARE} = \frac{1}{N}\sum_{i=1}^{N} |(\mathrm{Pu}_i^E - \mathrm{Pu}_i^A)|/\mathrm{Pu}_i^E$, where Pu^E denotes purity estimated by the EVA method, and Pu^A denotes purity computed from automatic segmentation.

3.1.2 Drosophila imaginal discs

Drosophila melanogaster, the fruit fly, is often used in biological research, due to its high genetic similarity to humans,[5,34,35] its rapid evolution, and its short life cycle. These features allow genetic changes to be observed easily across generations. Imaginal discs are parts of the larvae from which the adult body parts develop. The expression of about 6000 genes was imaged by RNA in situ hybridization.[36] Each gene was imaged a number of times for four imaginal disc types, with the aim of determining the role of the gene in the development process.[37] The disc anatomy is represented by the gray-scale part of the image, with the gene expression in blue (Figure 7). Segmentation of the disc is needed for the subsequent analysis, which consists of aligning all discs of the same type, detecting the activations and processing them statistically.[38] This dataset contains several thousand images of imaginal discs of 4 types – wing, leg, eye and haltere, respectively.

[1] All work with animal and human tissues was performed in accordance with approval 712a/13 (G 13-04-10) issued by the Ethics committee of the Institute for Clinical and Experimental Medicine and Thomayerova Hospital (Prague, Czech Republic).

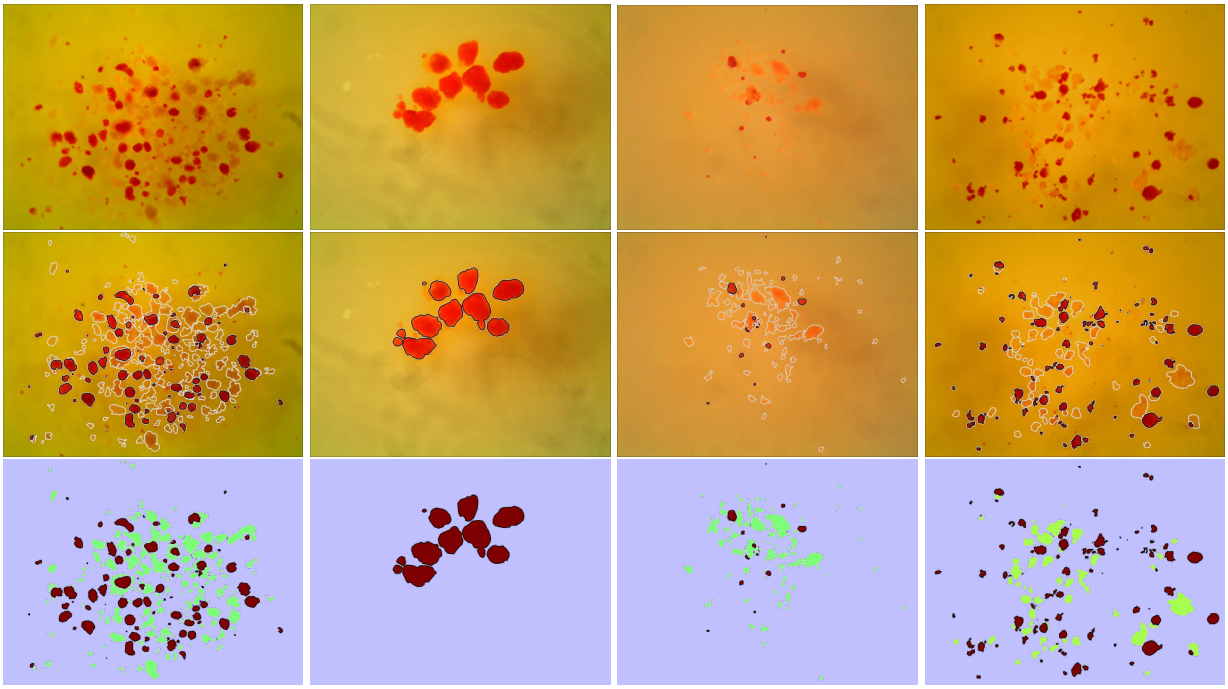


Figure 6 Examples of RGB microscopy images of Langerhans islets *(top row)* and their expert segmentation *(bottom row)* with islets in red and exocrine tissue in green. In the middle row, the class contours are shown as gray-scale lines, superimposed over the original images.

The evaluation is done on 15 images of imaginal disc type 3 — eye, for which we have reference segmentation.

3.1.3 Drosophila ovaries

Drosophila ovaries were imaged in 3D using fluorescence microscopy and fluorescently labeled RNA probes, in order to study gene expression patterns during Drosophila oogenesis. The images contain two channels, cellular anatomy and gene expression patterns. Only the cellular anatomy channel is used here. Almost 20 thousand 3D volumes were acquired, from which the experts extracted relevant 2D slices, each of them typically containing a few eggs, linked into a chain (see Figure 8). We have reference segmentations for 75 cropped images of individual eggs and 15 additional complete slices. The final aim of this segmentation is to identify and segment individual eggs, which will be further processed in a similar manner as the imaginal discs.[38]

The baseline method uses supertextons (texture descriptors computed on superpixels).[17] It is based on computing texture descriptors on superpixels, similar to this work, but it uses textons, a k-nearest neighbour classifier on the supertexton dictionary, and there is no spatial regularization.

3.1.4 Histology slices

Histology slices are routinely used for diagnosing cancer and other diseases. Consecutive tissue slices were stained with several different stains, e.g. hematoxylin and eosin (H&E), platelet endothelial cell adhesion molecule (PECAM-1, also known as CD31), antigen KI-67 (Ki67), and pro-Surfactant Protein C (proSPC), see Figure 9. High-resolution whole-slice images were acquired. Segmentation is an initial step in automatic analysis, and is used to suppress the background, to

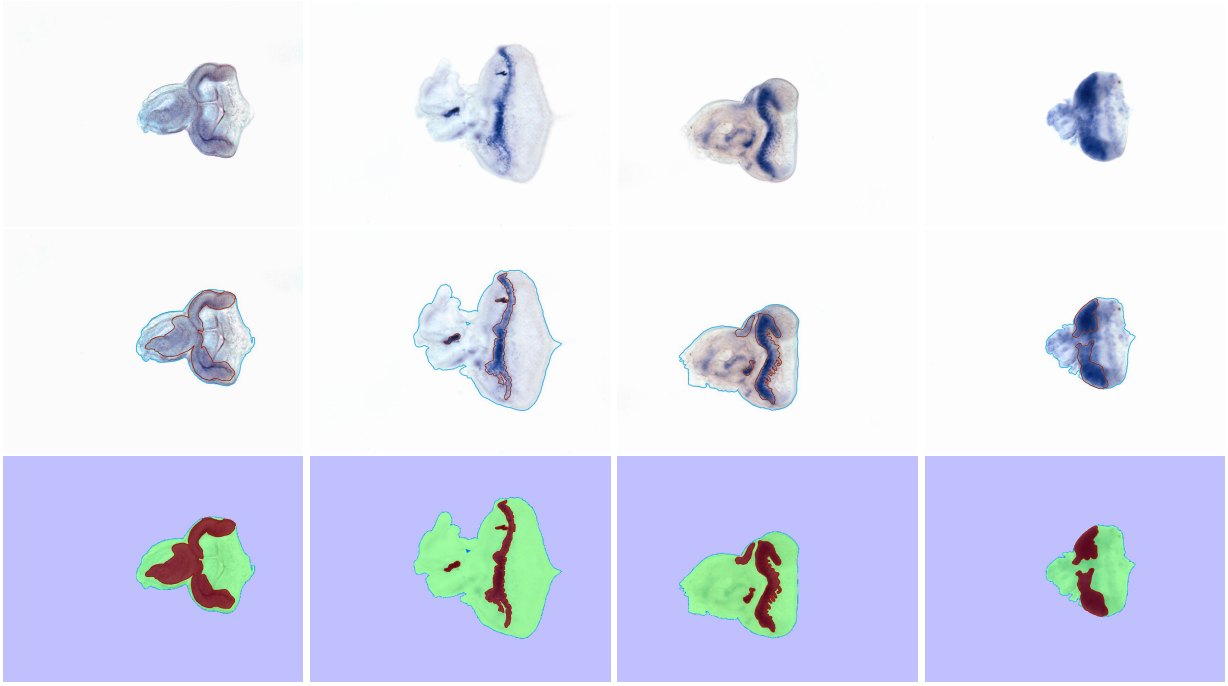


Figure 7 Example of microscopy images of Drosophila imaginal discs *(*top row) and manual segmentations *(bottom row)*, with the gene activation class colored in red and the rest of the imaginal disc colored in green. In the middle row, the class contours are shown as red and blue lines, superimposed over the original images.

support the alignment algorithm,[39,40] or to calculate the proportion of various tissue types. For this task, no reference segmentation is available, so we can evaluate the segmentation results only in terms of similarity among consecutive slices. The final task is to register all these slices together and to combine the information they provide, which is challenging due to large appearance differences.[40]

3.1.5 Computed Tomography images of the human chest

The fifth application is in segmenting lungs in Computed Tomography images of the human chest (see Figure 10). This is challenging due to the high noise and the presence of nodules inside the lungs. The standard methods use thresholds and some sophisticated approaches.[3] Some new approaches have been explored, e.g. the use of Graph Cut at pixel level,[41] which is not convenient for large images, and the use of prior shape information.[42] Here, we present just a few sample images without expert annotation.

3.2 Common baseline segmentation methods

Together with the baseline methods specific for Langerhans islets and Drosophila ovaries, we introduce two more general supervised segmentation methods which were used for experimental comparison.

Weka segmentation with Graph Cut We implemented two segmentation methods based on pixel-wise random forest classification[43] using the Weka toolbox.[4,44] For the drosophila imaginal discs and Langerhans islet microscopy images, we used HSV color channels as features. For

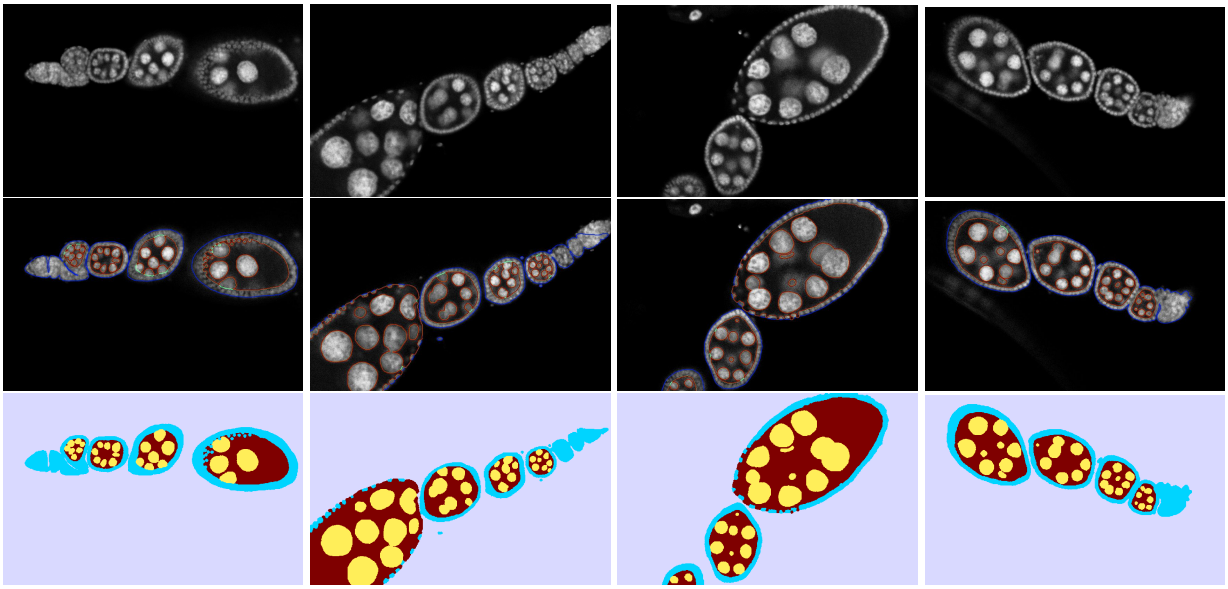


Figure 8 Examples of microscopy images of Drosophila ovaries *(top row)*, manual segmentations of individual eggs into 4 classes *(bottom row)* — follicle cells (cyan), nurse cells (yellow), cytoplasm (red) and background (blue). The same segmentations visualized as contours superimposed over the original images *(middle row)*

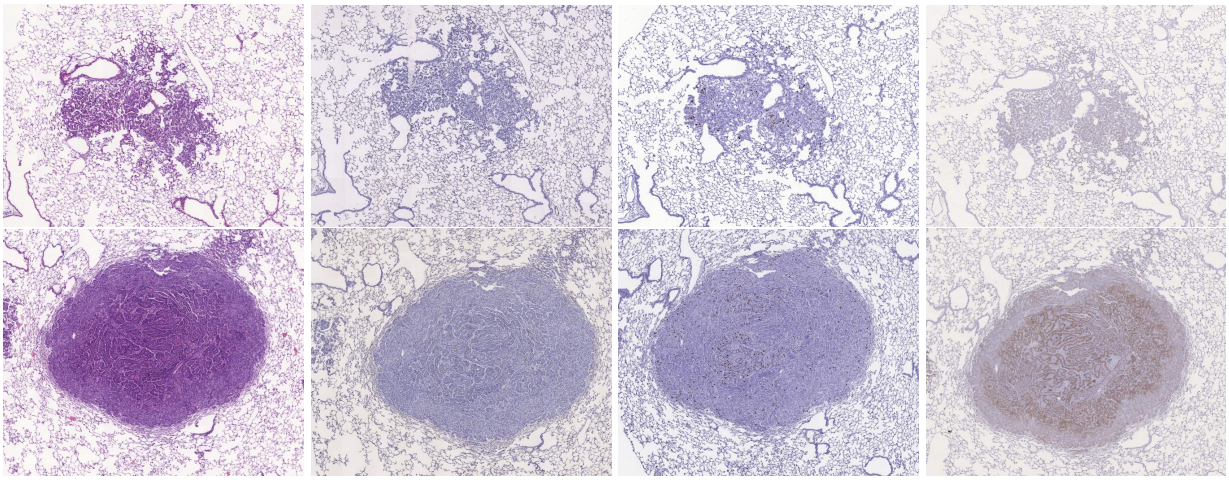


Figure 9 Sample images of lesion tissue in rows stained by four stains in columns— H&E, CD31, Ki67 and proSPC — going from left to right.

the gray-scale drosophila ovary, we used the following texture features calculated from a small neighborhood:[44] mean, median, and a Sobel filter response. The classifier was trained using reference pixel-level segmentation created by an expert. Regularization was applied to the resulting probability maps generated from the classifier by Graph Cut[45] to obtain the final segmentation. The method is denoted as 'Weka & GC(*smoothness cost*, *edge cost*)' in Section 4.

Superpixel segmentation. We have also tested the method proposed here without Graph Cut regularization, taking the classifier output directly as the final result. It is denoted as "our RF" because it uses the random forest classifier.

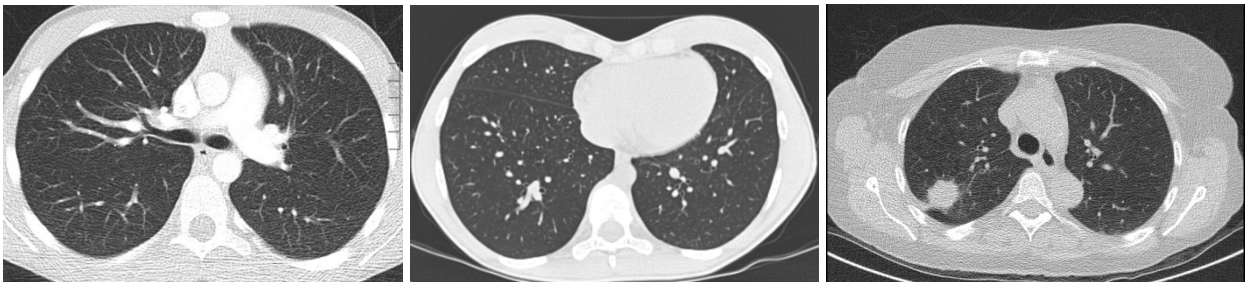


Figure 10 Computed Tomography images of the human chest.

Dataset	image size [px]	color space	SLIC size [px]	SLIC regul.	Features	Classif. / Model	GC regul.
Langer. islets	2000	HSV	20	0.2	color	RF (16 trees)	1.
imaginal disc	1600	RGB	25	0.25	color & texture	RF (20 trees)	2.
ovary	1000	gray	15	0.35	texture	RF (20 trees)	3.
histology	2400	RGB	35	0.2	color & texture	GMM	3.
human lungs	800	gray	20	0.3	texture	GMM	10.

Table 1 Default configuration of the segmentation parameters for each dataset together with the typical image size for each dataset.

3.3 Segmentation quality criteria

We use standard evaluation measures for multi-class classification[46] e.g. F_1 score, accuracy, precision, and recall. For images where we have ground truth segmentation (Langerhans Islets, Drosophila imaginal discs and ovaries), we use also the adjusted Rand score or index (ARS),[47] which is the corrected-for-chance version of the Rand index, which is a statistical measurement of the similarity between two groupings. It counts the number of times that two objects are classified into the same class or into different classes by the two methods. The adjusted Rand index can yield values in the range of $(-1, 1)$, the higher the better.

4 Experiments

Unless specified otherwise, all experiments use the parameters given in Table 1 and we report the F_1 measure using 10-fold cross-validation. We first show the effect of varying the most important parameters and design decisions on the segmentation quality. We then compare the proposed method with previously used methods. Quantitative evaluation is presented only for datasets where reference segmentations are available — the Langerhans islet and the two Drosophila datasets.

4.1 Superpixel parameters

Figure 11 shows the influence of the superpixel size N and regularization parameter r on the Drosophila imaginal disc images. Superpixel size N is a trade-off between computational complexity and the approximation error; values around $N = 25$ seems to work well for our data. Regularization r has a strong effect on superpixel regularity. Small values ($r \sim 0$) make the superpixels follow the contours in the image, with few shape constraints, while strong regularization ($r \sim 1$) makes superpixels mostly rectangular, regardless of the image. We observed that $r = 0.2$ provided good results for all our images.

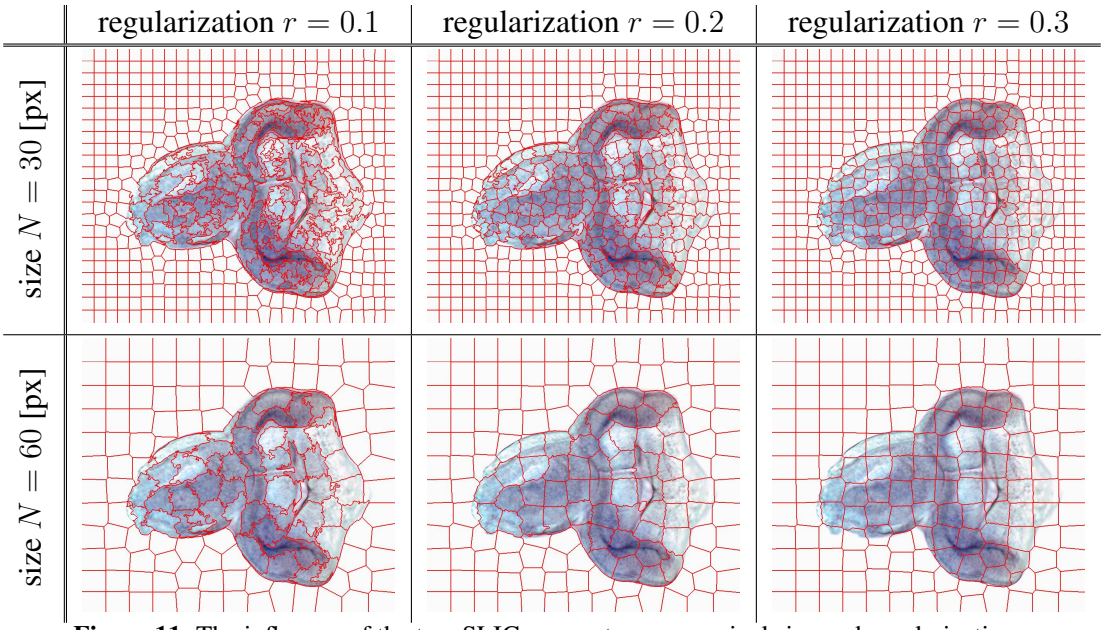


Figure 11 The influence of the two SLIC parameters: superpixel size and regularization.

Dataset	Random forest	Dec. Tree	Grad. boost	Log. regression	k-NN
Langer. islets	0.9898	0.9627	0.9933	0.9877	0.9692
imaginal disc	0.9901	0.9262	0.9923	0.9898	0.9873
Ovary	0.9902	0.9627	0.9913	0.9884	0.9863

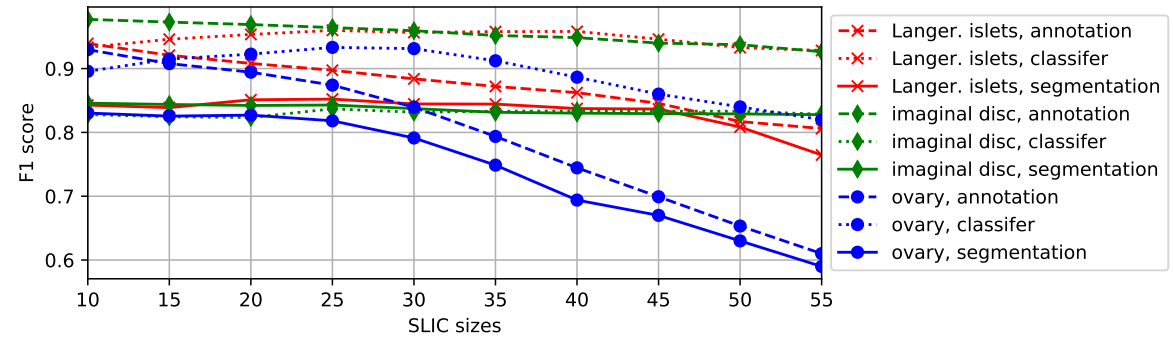
Table 2 Evaluation of the performance of the classifiers, as measured by the AUC for the best parameter configuration.

For each dataset, we computed the 'ideal' segmentation Y_A for given superpixels by assigning to superpixels the most frequent class of their pixels. We then evaluated quantitatively the effect of superpixel parameters on the segmentation quality (Figure 12) using 10-fold cross-validation. We compute the F_1 measure of the random forest classifier on the superpixel level (denoted 'classifier') and on the pixel-level Y_Ω (denoted 'segmentation'), with the superpixel-level F_1 measure using the ideal segmentation Y_A (denoted 'annotation'). We see that the optimum varies for the segmentation accuracy between applications, but the chosen values of $N = 25$ and $r = 0.2$ seems to be a good compromise in all cases.

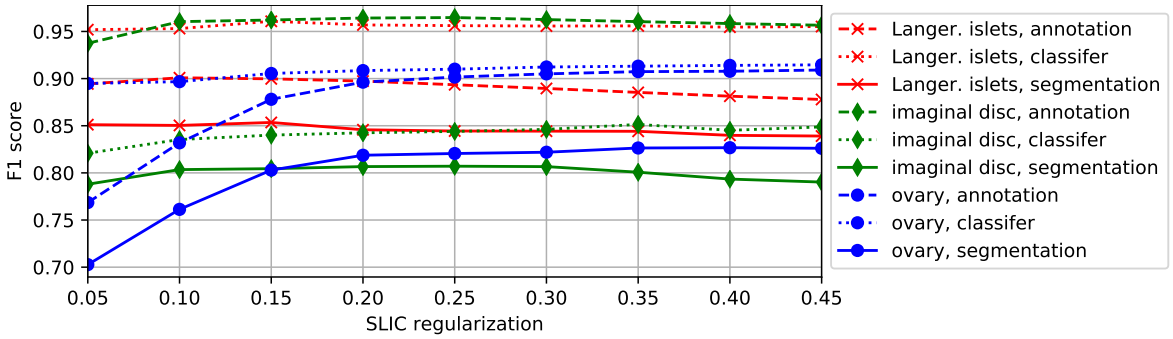
4.2 Classifiers and Graph Cut parameters

Classifiers. We experimented with 5 standard classifiers— random forest, decision trees, gradient boost, logistic regression, and k-nearest neighbors, as implemented in the Scikit-learn[48] Python library. For each of them, we chose the best parameters in terms of the F_1 score from 250 randomly generated parameter values, sampling uniformly from a user-defined range of values. The classifiers are compared in Table 2 using the AUC criterion. We chose random forest because it is one of the best performing classifiers, as it is fast and can handle large datasets.

Edge weights. We evaluate the effect of choosing different metrics in the model edge weighting (Section 2.6) in Table 3. The l_∞ metric was chosen as the best compromise. Table 4 compares



(a)



(b)

Figure 12 Evaluating the F_1 score for different superpixel sizes *(a)* and regularization *(b)* for the random forest on the superpixel level ('classifier') and on the pixel level Y_Ω ('segmentation'), compared with the ideal segmentation Y_A on the superpixel level ('annotation').

Dataset	nb. classes	l_1	l_2	l_∞
Langer. islets	2	0.772	0.781	0.774
imaginal disc	3	0.813	0.808	0.807
ovary	4	0.818	0.816	0.824

Table 3 The effect of the metrics used in model weights (Section 2.6) on the final segmentation accuracy measured by the F_1 score. Only two-class annotation were used for Langerhans islets, merging the classes for background and exocrine tissue.

the effect of different edge weight types on the segmentation results. We can see that the newly proposed edge weight based on model distance outperforms all others on all datasets.

Graph Cut regularization. We studied the influence of the Graph-Cut regularization constant β defined by equation (4). The optimal value of β depends on the dataset (see Figure 13) but the best value in all three cases was within the range $\beta = (1, 3)$.

4.3 Segmentation performance and evaluation

Our key experiment is a comparison of the performance of all methods described here and shown in Table 5. Our supervised segmentation is denoted as 'RF', because it uses the random forest classifier. The unsupervised segmentation is denoted as 'GMM', because the model is estimated

Datasets	Spatial weight	Color weight	Features weight	Model weight
Langer. islets	0.6826	0.7974	0.6894	0.7994
imaginal disc	0.6846	0.7985	0.6726	0.8133
ovary	0.7379	0.8150	0.7604	0.8236

Table 4 Comparison of the Graph Cut edge weight types defined in Section 2.6 in terms of the F_1 score.

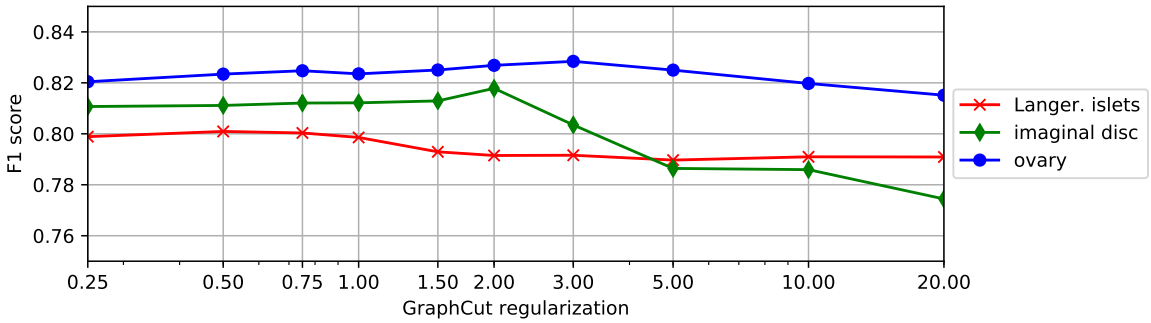


Figure 13 Evaluating the F_1 score for different Graph-Cut regularization parameters β.

via the Gaussian mixture model (GMM). A variant of GMM which learns from all images at once is denoted by '[gr.]'. Otherwise, the GMM model is learned for each image separately.

Denoted as 'ideal segm. Y_A', it uses the 'ideal' segmentation for given superpixels (Section 4.1). It is not a real method, as it needs to know the reference segmentation. It is meant to illustrate how close we are to the performance limit given by the superpixels. However, note that Y_A is optimal in the number of misclassified pixels, i.e. the sum of false positives and false negatives, which may not always translate to the best F_1 score, although the differences are usually small.

It can be seen that, on all datasets, our supervised segmentation with Graph Cut regularization works better than all other methods (except 'ideal segm. Y_A'). The use of Graph Cut regularization improves the segmentation results both for supervised segmentation and for unsupervised segmentation. More detailed discussion follows.

4.3.1 Langerhans islets

The detailed results on the Langerhans islet dataset are shown in Table 6. We show the pixel-wise segmentation quality with respect to islet reference segmentation as well as the correlation and MARE error of the purity estimate. We see that our supervised RF method with Graph Cut regularization yields the best results, which are also very close to 'ideal segm. Y_A'. It provides better results than the baseline NPA and 'Weka' methods. Our method also has almost perfect correlation with the purity estimate and the smallest error with respect to it. The improvement due to Graph Cut regularization is small because the segmented objects normally consist of just a few superpixels.

Unsupervised GMM performs better when the model is learned from all images (denoted by '[gr]') because the images have similar appearance and group learning of the GMM partially compensates the unbalance presence of tissue in individual images.

		Method	Lang. islets	imaginal disc	ovary
Pixel-wise	Supervised	Weka[44]	0.7374	0.6923	0.5800
		Weka & GC(0, 100)	0.7373	0.6887	0.5810
		Weka & GC(1, 50)	0.7376	0.6887	0.5965
		Weka & GC(10, 50)	0.6935	0.6887	0.1395
		Weka & GC(50, 100)	0.6862	0.6850	0.6007
		NPA[33]	0.8420	-	-
Superpixels		ideal segm. Y_A	0.8590	0.9696	0.9067
		Supertextons[17]	-	-	0.7488
		our RF	0.8565	0.8181	0.8201
		our RF & GC	0.8570	0.8229	0.8600
	Unsuper.	our GMM	0.5358	0.7542	0.5967
		our GMM & GC	0.5465	0.7644	0.6039
		our GMM [gr]	0.5682	0.7301	0.6009
		our GMM [gr] & GC	0.5816	0.7564	0.6083

Table 5 A comparison of all applicable methods for all datasets with ground truth segmentations in terms of the F_1 score. The baseline methods are 'Weka' (a Fiji plugin with Graph Cut regularization, see text), 'NPA'[33] for segmenting Langerhans islets and 'Supertextons'[17] for segmentation the Drosophila ovaries. We also introduce superpixel segmentation Y_A with an ideal classifier.

Method	accuracy	F_1 score	precision	recall	Pu Corr.	Pu MARE
Weka[44]	0.9748	0.7374	0.7946	0.7424	0.6572	1.381
Weka & GC(1, 50)	0.9772	0.7376	0.8043	0.7395	0.4825	1.777
Weka & GC(10, 50)	0.9767	0.6935	0.7406	0.6951	0.3805	2.163
Weka & GC(50, 100)	0.9761	0.6862	0.7175	0.6897	0.3609	2.265
NPA[33]	0.9774	0.8420	0.8522	0.8557	0.9820	0.296
ideal segm. Y_A	0.9819	0.8590	0.8894	0.8358	0.9992	0.058
our RF	0.9839	0.8565	0.8833	0.8391	0.9951	0.164
our RF & GC	0.9840	0.8570	0.8849	0.8390	0.9951	0.165
our GMM	0.6954	0.5358	0.5720	0.6186	0.6213	1.299
our GMM & GC	0.7253	0.5465	0.5753	0.6398	0.6990	1.082
our GMM [gr]	0.8360	0.5682	0.5926	0.6326	0.7500	0.955
our GMM [gr] & GC	0.8644	0.5816	0.5978	0.6447	0.8267	0.716

Table 6 Segmentation of Langerhans islets. Quantitative evaluation by standard metrics and the correlation and MARE error of the purity estimate Pu with respect to the expert validation.

4.3.2 Drosophila imaginal discs

Detailed quantitative results on the Drosophila imaginal disc segmentation are show in Table 7. As before, our superpixel segmentation performs better that the pixel-wise baseline 'Weka' method. Moreover, on this dataset, the baseline results are also matched or exceeded by the unsupervised GMM methods. In Figure 14, we show an example of segmentation by the supervised and unsupervised methods, together with the reference segmentation. It can be seen that the results are very similar for superpixel segmentations. The three desired classes — gene activation, disc and background — are clearly distinguished.

Method	ARS	accuracy	F_1 score	precision	recall
Weka[44]	0.9508	0.952	0.6923	0.7497	0.7101
Weka & GC(1, 50)	0.9563	0.9539	0.6887	0.7594	0.7078
Weka & GC(10, 50)	0.9563	0.9539	0.6887	0.7594	0.7078
Weka & GC(50, 100)	0.9584	0.9548	0.6850	0.7705	0.7073
ideal segm. Y_A	0.9843	0.9943	0.9696	0.9737	0.9656
our RF	0.9641	0.9728	0.8181	0.8424	0.8201
our RF & GC	0.9653	0.9739	0.8229	0.8506	0.8219
our GMM	0.9486	0.9557	0.7542	0.7631	0.8004
our GMM & GC	0.9504	0.9517	0.7644	0.7844	0.846
our GMM [gr]	0.9482	0.9533	0.7301	0.7377	0.7803
our GMM [gr] & GC	0.9521	0.9571	0.7564	0.7643	0.8032

Table 7 Segmentation of Drosophila imaginal discs. Quantitative evaluation of applicable methods by standard metrics.

Speaking about unsupervised segmentation, there is a considerable variance in appearance which leads to slightly worse results for estimating global appearance model from whole dataset (denoted by '[gr]') than learning the GMM for each image indigently.

4.3.3 Drosophila ovary

Segmentation of Drosophila ovaries turns out to be very challenging, possibly due to the high level of similarity between the two pairs of classes: nurse cells—follicular cells, and cytoplasm—background, which the baseline 'Weka' method fails to distinguish. By contrast, this task is well handled by texture features when superpixels are used.

The quantitative segmentation results are presented in Table 8. Our method have much in common with the baseline Supertextons.[17] We improved the F_1 score by 7% by using random forest and texture features instead of the k-NN classifier on the supertextons dictionary. We gained an additional 4% by applying Graph Cut spatial regularization. Some example results are shown in Figure 15, and we can observe much better spatial regularity of our Graph Cut regularized method with respect to the baseline 'Supertextons'. The unsupervised methods work less well for this application due to the high level of similarity between segmented classes.

4.3.4 Histopathology

We do not have a reference segmentation for the histopathology dataset, so we cannot use supervised segmentation and quantitative evaluation. Instead, we applied unsupervised segmentation, which seems to perform quite well. We learned the GMM from multiple equally stained images, see Figure 9. Due to extremely high image resolution, image registration typically uses a multiresolution approach and universally down-samples images/segmentation to individual levels. Instead we propose using Graph Cut regularization as smart reducing detail in different scale levels. Example results are shown in Figure 16, demonstrating also the effect of changing the regularization parameter.

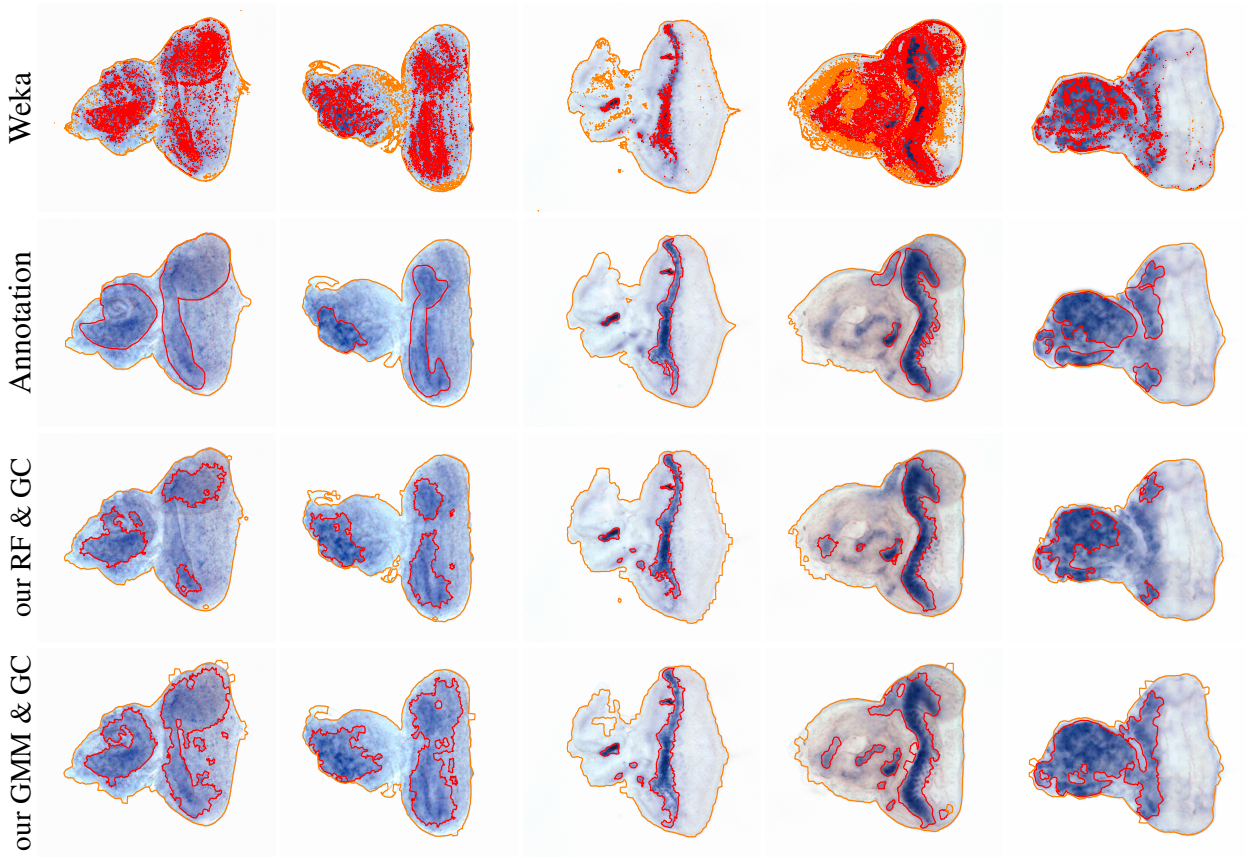


Figure 14 Visualization of the expert annotation and segmentation results with the 'Weka' baseline classifier and our supervised and unsupervised methods. Clearly, GMM was learned quite accurately.

4.3.5 Lung CT

In addition to the other applications, we also took a small number of Computed Tomography (CT) images of a human chest, and we applied our methods to the task of segmenting lungs from axial cuts. In Figure 17, we show the unsupervised segmentation results without and with Graph Cut which reduce number of artifacts in the lungs and still it preserves the lungs shape.

5 Conclusion

We have presented a feature-based segmentation method using an innovative combination of superpixels and GraphCut regularization to impose spatial compactness. This makes the segmentation both fast and robust. We have also introduced a new edge weight factor. Finally, we have tested our method extensively on five real applications, and have compared it with previously used methods. We found not only that superpixel methods work better than non-superpixel methods, but also that GraphCut regularization yields further improvement. Interestingly, an unsupervised variant of our method can generate completely acceptable results for some applications, without the need for manual segmentation.

Source code

The implementation is available on `http://github.com/Borda/pyImSegm`.

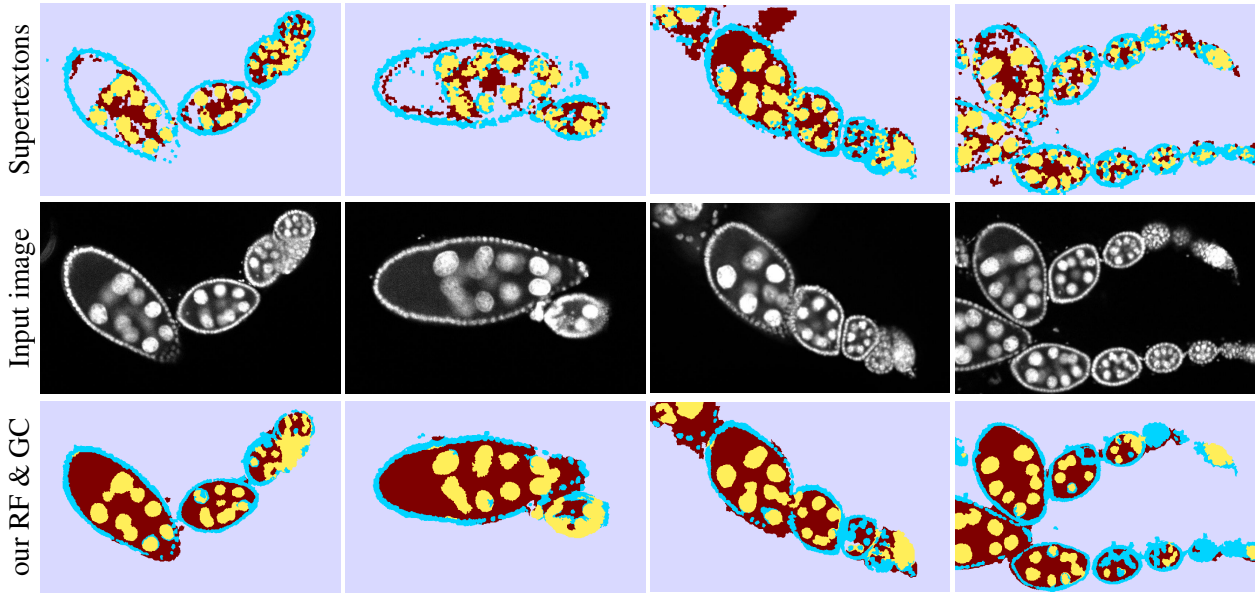


Figure 15 Example segmentations of Drosophila ovaries. Observe that our Graph Cut regularized segmentation *(bottom row)* is much more regular than the baseline method (Supertextons[17]). The input images are in the middle row.

Method	ARS	accuracy	F_1 score	precision	recall
Weka[44]	0.8008	0.8842	0.5800	0.6250	0.5833
Weka & GC(1, 50)	0.8167	0.8909	0.5965	0.6520	0.6023
Weka & GC(10, 50)	0.8235	0.7844	0.1395	0.1392	0.1399
Weka & GC(50, 100)	0.8214	0.8934	0.6007	0.6686	0.6085
ideal segm. Y_A	0.9528	0.9735	0.9067	0.9126	0.9021
Supertextons[17]	0.8633	0.9220	0.7488	0.7403	0.7798
our RF	0.8836	0.8509	0.8201	0.8298	0.8195
our RF & GC	0.8883	0.9090	0.8600	0.8627	0.8666
our GMM	0.7306	0.8578	0.5967	0.5854	0.6385
our GMM & GC	0.7481	0.8649	0.6039	0.5953	0.6472
our GMM [gr]	0.7385	0.8603	0.6009	0.5831	0.6519
our GMM [gr] & GC	0.7599	0.8666	0.6083	0.5805	0.6578

Table 8 Segmentation of Drosophila ovaries. Quantitative evaluation by standard metrics using baseline pixel-level segmentation, with and without Graph Cut, supertexton[17] segmentation, and the proposed 'RF' superpixel-based segmentation.

Disclosures

The authors have no relevant financial interests in this article and no potential conflicts of interest to disclose.

Acknowledgments

This work was supported by Czech Science Foundation projects GA14-21421S and GA14-10440S, and by the Grant Agency of CTU in Prague under grant no. SGS15/154/OHK3/2T/13. A part of

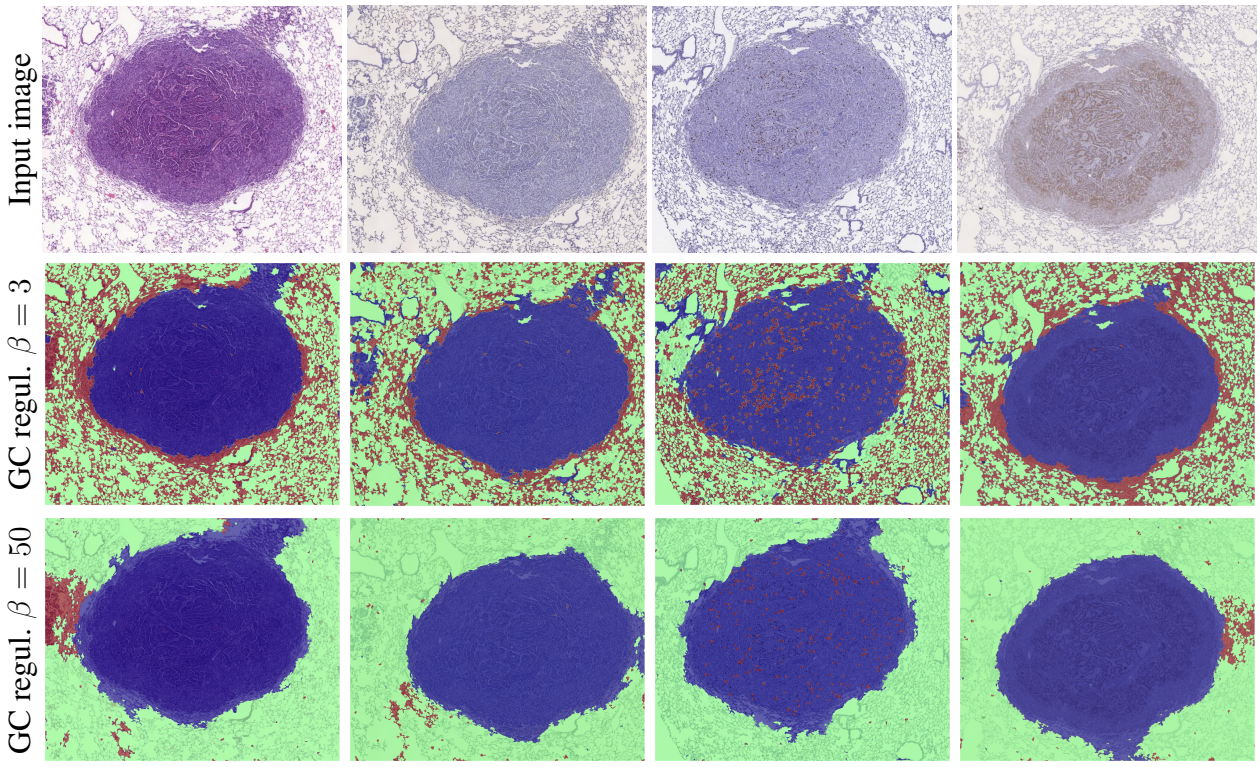


Figure 16 Unsupervised segmentation using GMM of a lesion stained by four stains — H&E, CD31, Ki67 and proSPC — shown in the columns. In the rows, we show the input image, and the segmented image with Graph Cut regularization $\beta = 3$ and $\beta = 10$.

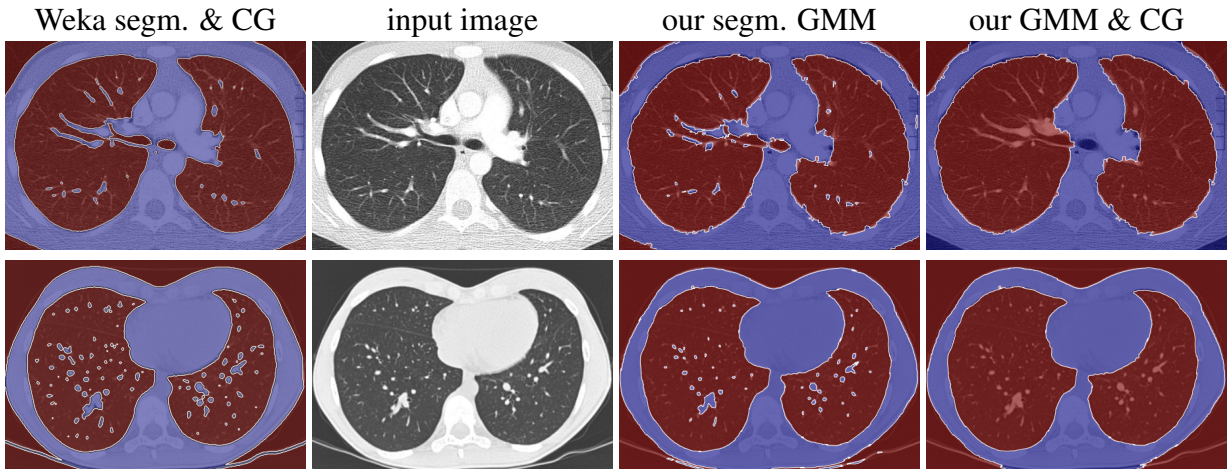


Figure 17 Unsupervised segmentation of a CT image of human lungs with and without Graph Cut. We present also trained 'Weka' segmentation with just a small subset of foreground and background pixels.

this work was carried out under the NII International Internship Program.

References

1 D. L. Pham, C. Xu, and J. L. Prince, "A Survey of Current Methods in Medical Image Segmentation," *In Annual Review of Biomedical Engineering* **2**, 315–338 (2000).

2 K.-P. Wong, "Medical Image Segmentation: Methods and Applications in Functional Imaging," in *Handbook of Biomedical Image Analysis*, J. S. Suri, D. L. Wilson, and S. Laxminarayan, Eds., 111–182, Springer US (2005).

3 A. Elnakib, G. Gimel'farb, J. Suri, *et al.*, "Medical Image Segmentation: A Brief Survey," in *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, A. S. El-Baz, R. Acharya U, A. F. Laine, *et al.*, Eds., 1–39, Springer New York (2011).

4 M. Hall, E. Frank, and G. Holmes, "The WEKA data mining software: an update," *SIGKDD Explorations* **11**(1) (2009).

5 P. Tomancak, A. Beaton, R. Weiszmann, *et al.*, "Systematic determination of patterns of gene expression during Drosophila embryogenesis.," *Genome biology* **3**(12), RESEARCH0088 (2002).

6 M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*, Cengage Learning, 3 ed. (2007).

7 W. Khan, "Image Segmentation Techniques: A Survey," *Journal of Image and Graphics* **1**(4), 166–170 (2013).

8 S. Agrawal, "Survey on Image Segmentation Techniques and Color Models," *(IJCSIT) International Journal of Computer Science and Information Technologies* **5**(3), 3025–3030 (2014).

9 Y. Boykov and O. Veksler, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE* **23**(11), 1222–1239 (2001).

10 Y. Boykov, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision* **70**, 109–131 (2006).

11 F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *2012 International Conference on Systems and Informatics, ICSAI 2012*, 1936–1941 (2012).

12 A. Tremeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Transactions on Image Processing* **9**(4), 735–744 (2000).

13 D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An Evaluation of the State-of-the-Art," *Computer Vision and Image Understanding* **abs/1612.0**(35), – (2016).

14 A. Lucchi, K. Smith, and R. Achanta, "Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks With Learned Shape Features," *Medical Imaging, IEEE* **31**(2), 474 – 486 (2012).

15 J. Borovec and J. Kybic, "Fully automatic segmentation of stained histological cuts," in *17th International Student Conference on Electrical Engineering*, L. Husník, Ed., 1–7, CTU in Prague, (Prague) (2013).

16 L. Lalaoui, T. Mohamadi, and A. Djaalab, "New Method for Image Segmentation," *Procedia - Social and Behavioral Sciences* **195**, 1971–1980 (2015).

17 R. Nava and J. Kybic, "Supertexton-based segmentation in early Drosophila oogenesis," in *Proceedings - International Conference on Image Processing, ICIP*, **2015-Decem**, 2656–2659 (2015).

18 M. E. A. Bechar, N. Settouti, V. Barra, *et al.*, "Semi-supervised superpixel classification for medical images segmentation: application to detection of glaucoma disease," *Multidimensional Systems and Signal Processing* **28**(97), 1–20 (2017).

19 D. Boschetto and E. Grisan, "Superpixel-based classification of gastric chromoendoscopy images," in *Medical Imaging, SPIE*, **10134**, 101340W (2017).
20 O. Csillik, "Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using," *Remote Sensing* **9**(243), 19 (2017).
21 G. Xuan and W. Zhang, "EM algorithms of Gaussian mixture model and hidden Markov model," *Image Processing, 2001.* **1**, 145–148 (2001).
22 K. Puniyani, C. Faloutsos, and E. P. Xing, "SPEX2: Automated concise extraction of spatial gene expression patterns from fly embryo ISH images," *Bioinformatics* **26**(12), 47–56 (2010).
23 T. Kitrungrotsakul, X.-H. Han, and Y.-W. Chen, "Liver segmentation using superpixel-based graph cuts and restricted regions of shape constraints," *International Conference on Image Processing (ICIP)* **3**, 3368–3371 (2015).
24 X. Ye, G. Beddoe, and G. Slabaugh, "Automatic graph cut segmentation of lesions in CT using mean shift superpixels," *International Journal of Biomedical Imaging* **2010** (2010).
25 C. Y. Hsu and J. J. Ding, "Efficient image segmentation algorithm using SLIC superpixels and boundary-focused region merging," in *ICICS 2013 - Conference Guide of the 9th International Conference on Information, Communications and Signal Processing*, (2013).
26 X. Wang, H. Li, C.-E. Bichot, *et al.*, "A graph-cut approach to image segmentation using an affinity graph based on l0-sparse representation of features," in *2013 IEEE International Conference on Image Processing*, 4019–4023 (2013).
27 R. Achanta and A. Shaji, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," *Pattern Analysis and Machine Intelligence, IEEE* **34**(11), 2274 – 2282 (2012).
28 J. Borovec and J. Kybic, "jSLIC : superpixels in ImageJ," in *Computer Vision Winter Workshop*, Z. Kunbelova and J. Heller, Eds., 14–18, Czech Society for Cybernetics and Informatics, (Praha) (2014).
29 T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International journal of computer vision* **43**(1), 29–44 (2001).
30 A. Li, X. Wang, K. Yan, *et al.*, "Multilevel affinity graph for unsupervised image segmentation," in *ICIP*, (2016).
31 C. Ricordi, D. W. Gray, B. J. Hering, *et al.*, "Islet isolation assessment in man and large animals," *Acta Diabetologica Latina* **27**(3), 185–195 (1990).
32 J. Svihlik, J. Kybic, and D. Habart, "Color normalization for robust evaluation of microscopy images," in *Applications of Digital Image Processing, SPIE*, (2015).
33 D. Habart, J. Svihlik, J. Schier, *et al.*, "Automated analysis of microscopic images of isolated pancreatic islets," *Cell Transplantation* **25**, 2145–2156 (2016).
34 A. A. S. Hammonds, C. C. a. Bristow, W. W. Fisher, *et al.*, "Spatial expression of transcription factors in Drosophila embryonic organ development.," *Genome biology* **14**(12), R140 (2013).
35 Medzhitov R., Preston-Hurlburt P., and Janeway C.A. Jr, "A human homologue of the Drosophila Toll protein signals activation of adaptive immunity," *Nature* **388**(6640), 394–397 (1997).
36 J. N. Wilcox, "Fundamental principles of in situ hybridization.," *The journal of histochemistry and cytochemistry : official journal of the Histochemistry Society* **41**, 1725–33 (1993).

37 C. L. Harmon, P. Ahammad, A. Hammonds, *et al.*, "Comparative Analysis of Spatial Patterns of Gene Expression in Drosophila melanogaster Imaginal Discs," in *Research in Computational Molecular Biology*, T. Speed and H. Huang, Eds., *Lecture Notes in Computer Science* **4453**, 533–547, Springer Berlin Heidelberg (2007).

38 J. Borovec and J. Kybic, "Binary pattern dictionary learning for gene expression representation in drosophila imaginal discs," in *Mathematical and Computational Methods in Biomedical Imaging and Image Analysis (MCBMIIA) workshop at ACCV*, 555–569 (2016).

39 J. Borovec, J. Kybic, M. Bušta, *et al.*, "Registration of multiple stained histological sections," in *International Symposium on Biomedical Imaging, IEEE*, 1034–1037, (San Francisco) (2013).

40 J. Kybic and J. Borovec, "Automatic simultaneous segmentation and fast registration of histological images," in *International Symposium on Biomedical Imaging, IEEE*, 774 – 777 (2014).

41 S. Dai, K. Lu, J. Dong, *et al.*, "Neurocomputing A novel approach of lung segmentation on chest CT images using graph cuts," *Neurocomputing journal* **168**, 799–807 (2015).

42 K. Nakagomi, A. Shimizu, and H. Kobatake, "Multi-shape graph-cuts and its application to lung segmentation from a chest CT volume," in *MICCAI, Proc. of Workshop on Lung Image Analysis*, **157**(1), 185–199 (2011).

43 L. Breiman, "Random forests," *Machine Learning* **45**(1), 5–32 (2001).

44 I. Arganda-Carreras, V. Kaynig, C. Rueden, *et al.*, "Trainable weka segmentation: a machine learning tool for microscopy pixel classification," *Bioinformatics* **33**(15), 2424–2426 (2017).

45 S. Bagon, "Matlab wrapper for graph cut," (2006). [Online; accessed 2013-08-15].

46 O. Koyejo, P. Ravikumar, N. Natarajan, *et al.*, "Consistent Multilabel Classification," in *Advances in Neural Information Processing Systems*, 3321—-3329 (2015).

47 L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification* **2**(1), 193–218 (1985).

48 F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

Biography

Jiří Borovec was awarded an MSc double-degree in intelligent systems by Université Paul Sabatier (UPS), France, and the Technical University of Liberec (TUL), Czechia in 2011. He is now pursuing a PhD degree at the Center of Machine Perception, which is a part of the Department of Cybernetics of the Faculty of Electrical Engineering of the Czech Technical University (CTU) in Prague.

Jan Švihlík was awarded his MSc (Ing.) degree in 2005 and his PhD degrees in 2008 by the Czech Technical University (CTU) in Prague. Since 2008, he has been an assistant professor at the University of Chemistry and Technology Prague. He held a post-doc position at the Center for Machine Perception, CTU in Prague, from 2013–2016. He is a member of SPIE.

No biographies of the other authors are available.