Learning to Detect Cells Using Non-overlapping Extremal Regions

Carlos Arteta[1], Victor Lempitsky[2], J. Alison Noble[1], and Andrew Zisserman[1]

[1] Department of Engineering Science, University of Oxford, U.K.
[2] Yandex, Moscow, Russia.

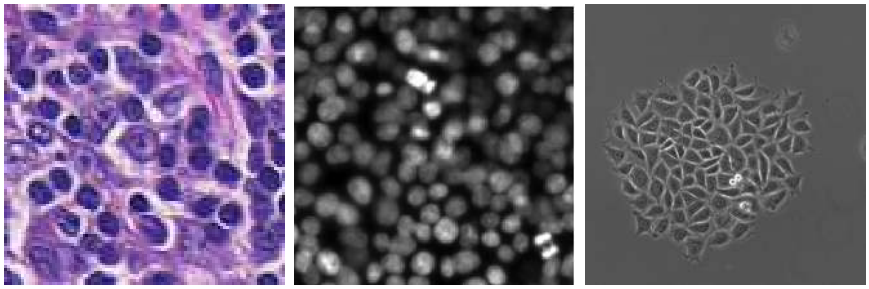
Abstract. Cell detection in microscopy images is an important step in the automation of cell based-experiments. We propose a machine learning-based cell detection method applicable to different modalities. The method consists of three steps: first, a set of candidate cell-like regions is identified. Then, each candidate region is evaluated using a statistical model of the cell appearance. Finally, dynamic programming picks a set of non-overlapping regions that match the model. The cell model requires few images with simple dot annotation for training and can be learned within a structured SVM framework. In the reported experiments, state-of-the-art cell detection accuracy is achieved for H&E-stained histology, fluorescence, and phase-contrast images.

1 Introduction

Automatic cell detection is a subject of interest in a wide range of cell-based studies, as it is the basis of many automatic methods for cell counting, segmentation and tracking. The broad diversity of cell lines and microscopy imaging techniques require that cell detection algorithms adapt well to different scenarios. The difficulty of the problem also increases when the cell density of the sample is high, as in this case the cell size can vary and cell clumping is usual. Moreover, in some applications different cell types or other similar structures can be present in the same image, and in this case the algorithm is required to detect only the cells of interest, posing a barrier hard to overcome with classical image processing techniques.

In this paper we propose a learning-based method that is general enough to perform well across different microscopy modalities. Rather than invoking computationally-intensive segmentation frameworks [1,9], or classifying all image patches in a sliding-window manner [15], it uses a highly-efficient MSER region detector [8] to find a broad number of candidate regions to be scored with a learning-based measure. The non-overlaping subset of those regions with high similarity to the class of interest can then be selected via dynamic programming, while the learning can be done within the structured output framework [12].

The new method is evaluated on three data sets (Figure 1), which are annotated with dots; a dot is placed inside each cell. Given only this minimalistic annotation, the method is able to learn a model that achieves state-of-the-art detection accuracy, in our evaluation, despite all the variation between the data sets.



(a) Histopathology (b) Fluorescence HEK (c) Phase-contrast HeLa

Fig. 1. Example images from the data sets used for cell detection. (a) Histopathology image of breast cancer tissue, which is stained to highlight lymphocyte nuclei (100×100 pix.; cell size 6–8 pix.) (b) Fluorescence microscopy image of human embryonic kidney cells (190×190 pix.; cell size 10–20 pix.) (c) Phase-contrast image of cervical cancer cells of the HeLa cell line (400×400 pix.; cell size 10–40 pix.)

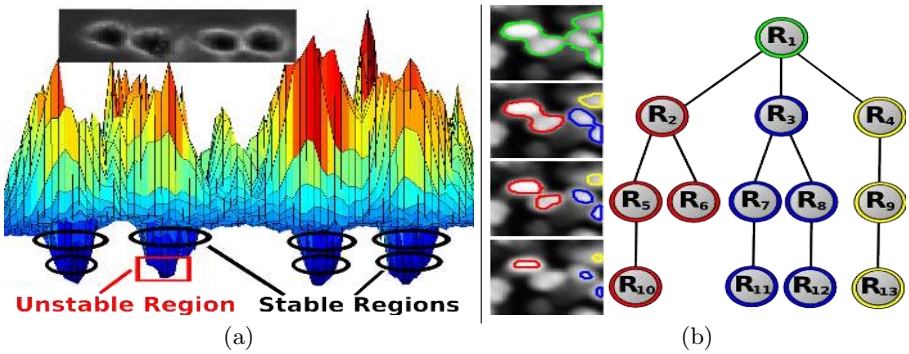
2 Learning Non-overlapping Extremal Regions

The model operates by first producing a set of candidate *extremal regions*, and then picking a subset of those regions based on a learned classifier score and subject to a *non-overlap constraint*. We discuss the components of the method, namely the detection of candidate regions, the inference, and the structured learning, next.

Extremal regions of the grey-value image $\mathcal{I}$ are defined as connected components of a thresholded image $\mathcal{I}_{>t} = \{\mathcal{I} > t\}$ for some t. In other words, a region is extremal if the image intensity everywhere inside of it is higher than the image intensity at its boundary. Our approach thus builds on the fact that in many microscopy modalities, cells show up as bright or dark blobs in one of the intensity channels, and therefore can be closely approximated by extremal regions of this intensity channel. An important property of extremal regions is their *nestedness*, i.e. the fact that for the same image $\mathcal{I}$ two extremal regions R and S can be either nested or non-overlapping ($R \subset S$ or $R \supset S$ or $R \cap S = \emptyset$. See Figure 2).

The number of extremal regions can be combinatorial, so in practice we consider only regions that are *maximally stable* in the sense of [8], i.e. the speed of their area variation w.r.t. changing threshold t is a local minimum and is below a separate *stability threshold*. We thus use a popular and efficient maximally stable extremal region detector (MSER) [8] to find a representative subset of all extremal regions. To boost the recall for cell detection, we set the stability threshold to a very high value, so that the MSER-detector produces a manageable but very large (thousands) number of candidate regions. Our inference procedure then determines which of those candidates correspond to cells.

Inference under the non-overlap constraint. Let $R_1, R_2, \ldots R_N$ be the candidate set of N extremal regions detected in an image. Let us assume that each region R_i is assigned a value V_i, which is produced by a classifier and



(a) (b)

Fig. 2. (a) Example of the intensity profile of an image region containing cells. The MSER algorithm detects extremal regions that are stable in area growth while varying an intensity threshold. Typically, many extremal regions are nested within and between cells (especially when there is cell clumping) forming a tree structure. For example, (b) the boundaries of several MSERs that appear in the close-up of a cell image are shown, which can be represented by the tree structure. The parent-child relationships in the tree correspond to the nestedness of the regions. The tree structure is utilized by the inference algorithm.

indicates the appropriateness score of this region to the class of cells we want to detect. Our method then picks a subset of extremal regions so that the sum of scores of the picked regions is maximized, while the picked regions do not overlap (the non-overlap constraint).

To formalize this task, we define a set of binary indicator variables $\mathbf{y} = \{y_1, y_2, \dots y_N\}$ so that $y_i = 1$ implies the region R_i being picked. Let $\mathcal{Y}$ be a set of those region subsets that do not have region overlap, that is, $\mathcal{Y} = \{\mathbf{y} \,|\, \forall\, i, j : (i \neq j) \wedge (y_i = 1) \wedge (y_j = 1) \Rightarrow R_i \cap R_j = \emptyset\}$. Then, the optimization task faced by the model is:

$$F(\mathbf{y}) = \max_{\mathbf{y} \in \mathcal{Y}} \sum_{i=1}^{N} y_i \, V_i \ . \tag{1}$$

For an arbitrary set of regions, maximizing (1) over $\mathbf{y} \in \mathcal{Y}$ is NP-hard (equivalent to *submodular maximization*). Fortunately, the nestedness property of extremal regions permits fast and exact maximization of (1). The idea is to organize the extremal regions into trees according to the nestedness property, so that each tree corresponds to a set of overlapping extremal regions (Figure 2b). The exact solution of (1) can then be obtained via dynamic programming on those trees [11] after an appropriate variable substitution (see implementation details).

Learning formulation. As discussed above, our method relies on machine learning to score each region for the detection task. A suitable scoring can be learned in a principled fashion from the dot-annotated training data as follows. Assume a set of M training images $\mathcal{I}^1, \mathcal{I}^2, \dots \mathcal{I}^M$, where each training image $\mathcal{I}^j$ has a set of N^j MSER regions $R_1^j, R_2^j, \dots R_{N^j}^j$. For each of these regions R_i^j a feature vector $\mathbf{f}_i^j$ is computed (the feature vector choice is described in the

implementation details). Finally, assume that the images are annotated, so that n_i^j denotes the number of user-placed dots (annotations) inside the region R_i^j.

To obtain the score for each region, we use a linear classifier so that the value V_i^j for the region R_i^j is computed as a scalar product $(\mathbf{w} \cdot \mathbf{f}_i^j)$ with the *weight vector* $\mathbf{w}$. The goal of learning is then to find a weight vector so that the inference procedure tends to pick regions with $n_i^j = 1$, and also to ensure that for each dot a region is picked that contains it. In this way, the produced set of regions tends to be in a one-to-one correspondence with the user-placed dots.

Learning via binary classification. The simplest way to learn $\mathbf{w}$, and one that already produces competitive results in our comparisons, is to learn a binary classifier. For this, all regions in the training images are considered, and those with $n_i^j = 1$ are assigned to the positive class while all others are assigned to the negative class. Training any linear classifier, e.g. via a support vector machine algorithm, then produces a desired $\mathbf{w}$.

Structured learning. Learning via binary classification does not take into account the non-overlap constraint. A more principled approach is to use a structured SVM [12] that directly optimizes the performance of the inference procedure on the training set. Consider the configuration $\mathbf{y}^j \in \mathcal{Y}^j$ defining the set of non-overlapping regions for the image $\mathcal{I}^j$. It is natural to define an error measure (the *loss*) associated with $\mathbf{y}^j$ as the deviation from the one-to-one correspondence between the user-placed dots and the picked regions:

$$L(\mathbf{y}^j) = \sum_{i=1}^{N^j} y_i^j \, |n_i^j - 1| + U^j(\mathbf{y}^j) \tag{2}$$

$U^j(\mathbf{y}^j)$ denotes the number of user-placed dots that are not covered by any region R_i^j with $y_i^j = 1$ (i.e. have no correspondence).

To perform the learning, the "ground truth" configuration $\bar{\mathbf{y}}^j = \{\bar{y}_1^j, \bar{y}_2^j, \ldots \bar{y}_{N^j}^j\} \in \mathcal{Y}$ is defined for each training image by assigning a unique extremal region to each dot (see implementation details). The structured SVM method [12] then finds the optimal weight vector $\mathbf{w}$ by minimizing the following convex objective:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{M} \sum_{j=1}^{M} \max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(\sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j) \, y_i^j - \sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j) \, \bar{y}_i^j + L(\mathbf{y}^j) \right) \tag{3}$$

where the first term is a regularization on $\mathbf{w}$, C is a scalar regularization parameter, and the maximum inside the sum represents a convex (in $\mathbf{w}$) upper bound on the loss (2), that the inference (1) incurs on the jth training image [12].

The objective (3) can be optimized with a standard cutting-plane algorithm [12] provided that it is possible to perform the *loss-augmented inference*, which corresponds to finding maxima inside the second term of (3) for a fixed $\mathbf{w}$. Thus, one needs to solve:

$$\max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(\sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j) \, y_i^j - \sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j) \, \bar{y}_i^j \;\; + \;\; \sum_{i=1}^{N^j} y_i^j \, |n_i^j - 1| + U^j(\mathbf{y}^j) \right) \tag{4}$$

We then note that under the non-overlap constraint, the number of un-matched dots $U^j(\mathbf{y}^j)$ can be rewritten as $D^j - \sum_{i=1}^{N^j} y_i^j n_i^j$, where D^j is the total number of dots in the jth training image. After substituting $U(\mathbf{y}^j)$ and omitting the terms independent of $\mathbf{y}^j$, an equivalent optimization problem is obtained:

$$\max_{\mathbf{y}^j \in \mathcal{Y}^j} \sum_{i=1}^{N^j} \Big((\mathbf{w} \cdot \mathbf{f}_i^j) + |n_i^j - 1| - n_i^j \Big) \; y_i^j \tag{5}$$

which has exactly the same form as (1) with $V_i = (\mathbf{w} \cdot \mathbf{f}_i^j) + |n_i^j - 1| - n_i^j = (\mathbf{w} \cdot \mathbf{f}_i^j) - [n_i^j \geq 0]$. Thus, we can perform loss-augmented inference exactly via dynamic programming on trees, and get an optimal $\mathbf{w}$ through the cutting-plane procedure [12].

Implementation details. We use the MSER implementation from [14]. The feature vector for each region in a grayscale image is 92-dimensional, and consists of several concatenated histograms: (a) a 10-dimensional histogram of intensities within the region (separate histograms are computed for color images), (b) two 6-dimensional histograms of differences in intensities between the region border and a dilation of it for two different dilation radii (these histograms capture the spatial context of the region), (c) a shape descriptor represented by a 60-dimensional histogram of the distribution of the boundary of the region on a size-normalized polar coordinate system, and, finally, (d) the area A of the region represented by a 10-dimensional binary vector with the entry $\lceil \log A \rceil$ set to 1.

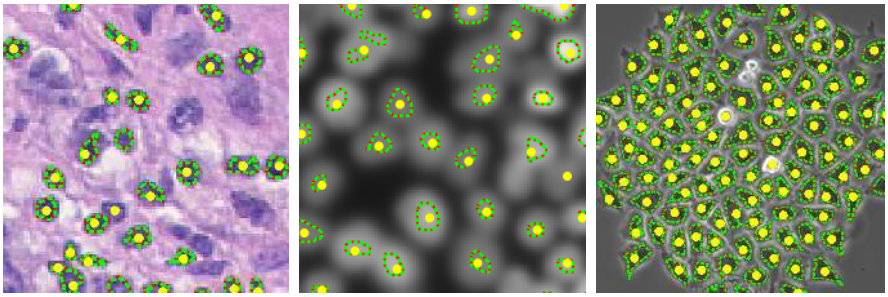
To generate the ground truth configuration for the structured learning, we first score all regions using the weight vector w_{bin} learned through a binary SVM. Then, for each dot, we include into the ground truth configuration the region that contains only this dot and has the highest score.

The dynamic programming within the inference (1) can be implemented via the following variable substitution: each $\mathbf{y}$ is mapped to a new set of binary variables $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$, so that $z_i = 1$ iff the y-variable for either the ith node or any of its ancestors in the MSER-tree is 1. The tree-structured graphical model on z-variables is defined for each region tree. For the root node i, the cost for $z_i = 1$ is set to V_i. For every edge in the tree connecting nodes i (parent) and j (child), the cost for $z_i = 0$ and $z_j = 1$ is set to V_j, while the cost for $z_i = 1$ and $z_j = 0$ is set to $-\infty$. The latter restricts inference to only those z-configurations that correspond to $\mathbf{y} \in \mathcal{Y}$. All other costs within pairwise and unary terms are set to 0. A standard max-product algorithm is run in each tree and the optimal z-variables are mapped back to y-variables.

The learning is done via the SVM^{struct} code [5,13]. In general, detecting cells on a 400-by-400 pixel HeLa image takes 30 seconds on an i7 CPU (dominated by our unoptimized MATLAB code for feature computation).

3 Experiments

Evaluating the model. Although the algorithm produces a set of regions, our aim is to optimize the detection accuracy (and not the segmentation) w.r.t. the



(a) Histopathology (b) Fluorescence HEK (c) Phase-contrast HeLa

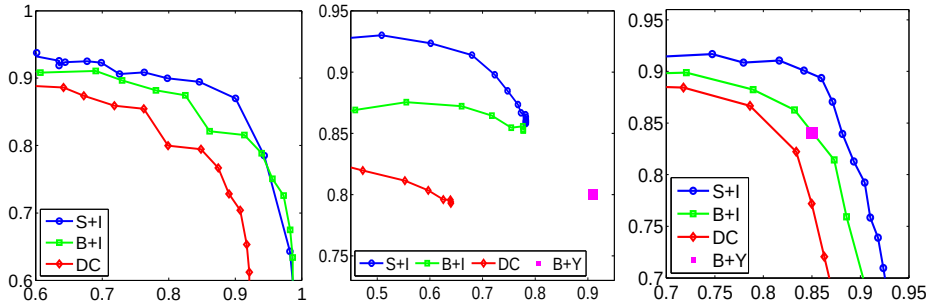
Fig. 3. Example results on each of the data sets. The boundaries of the detected MSER regions are shown in dashed green/red over the test images with yellow dots indicating the ground truth annotations. Note, the features are computed over a support region that is larger than the MSER region.

ground truth provided in the form of dots. Therefore, we evaluate the output of our method based on the position of the region centroids. A centroid is considered a true positive (TP) if it is within a radius ρ of the ground truth dot. In our experiments, ρ is set to the radius of the smallest cell in the data set. Thus, only centroids that lie inside cells are considered correct. Centroids further than ρ from ground truth dots are considered false positives (FP). Finally, missed ground truth dots are counted as false negatives (FN). The results are reported in terms of Precision=$\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and Recall=$\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$.

Three data sets for cell detection have been used to validate the method (Figure 1). Firstly, the ICPR 2010 Histopathology Images contest [4], which consists of 20 images of stained breast cancer tissue. It is required to detect lymphocyte nuclei, while discriminating them from breast cancer nuclei having very similar appearance. The second data set comes from [1] and contains 12 fluorescence microscopy images of human embryonic kidney (HEK) cells, where the detection task is challenging due to the significant intensity variation between cells across the image, fading boundaries, and frequent cell clumping. The third data set contains 22 phase-contrast images of cervical cancer cell colonies of the HeLa cell line, which presents a high variability in cell shapes and sizes.

Three variations of our method are evaluated: (I) *direct classification (DC)*, which evaluates all MSERs with a $\mathbf{w}$ vector learned via a binary classifier and chooses the region with the highest score in every set of overlapping regions with positive scores, (II) *binary SVM + inference (B+I)*, which does the full inference (1) based on the weight vector learned through binary classification, and (III) *structured SVM + inference (S+I)*, which uses inference with the weight vector learned by the structured SVM (3). The histopathology and the HeLa datasets were split into halves for training and testing, whereas the HEK data was evaluated in a leave-one-out fashion in order to test on the entire set and be able to fully compare results with [1].

Figure 4 shows the precision-recall curves for the three variations of our method. The curves were obtained by varying a constant τ added to the score



(a) Histopathology (b) Fluorescence HEK (c) Phase-contrast HeLa

Fig. 4. Precision (vertical) vs Recall (horizontal) curves for the three datasets for the three variations of our approach and [1] (denoted as B+Y, where available). Significant improvements brought by the non-overlap constraint (B+I) and the structured SVM (S+I) can be observed.

Table 1. Results for the data set of the ICPR 2010 Pattern Recognition in Histopathological Images contest [4]. Seven measures are reported: precision, recall and F1-score (when available), where higher numbers are better, and the four measures used in the evaluation of the ICPR contest, where lower numbers are better. The contest criteria consisted of the mean and standard deviation of two measurements: the Euclidean distance between detected dots and ground truth dots (**d**), and the absolute difference between the number of cells found and the ground truth number of cells (**n**).

Method	Prec.	Rec.	F_1-score	$\mu_d \pm \sigma_d$	$\mu_n \pm \sigma_n$
Our method	**86.99**	**90.03**	**88.48**	**1.68** $\pm$ 2.55	**2.90** $\pm$ 2.13
LIPSyM [6]	70.21	70.08	69.84	3.14 $\pm$ 0.93	4.30 $\pm$ 3.09
Bernadis et al. [1]	-	-	-	3.13 $\pm$ 3.08	12.7 $\pm$ 8.70
Kuse et al. [7]	65.23	69.99	67.29	3.04 $\pm$ 3.40	14.01 $\pm$ 4.40
Cheng et al. [2]	-	-	-	8.10 $\pm$ 6.98	6.98 $\pm$ 12.5
Graf et al. [3]	-	-	-	7.60 $\pm$ 6.30	24.5 $\pm$ 16.2
Panagiotakis et al. [10]	-	-	-	2.87 $\pm$ 3.80	14.23 $\pm$ 6.30

of each region. It can be seen that enforcing the non-overlap constraint increases the accuracy of the method considerably, especially when **w** is learned within the structured SVM framework.

Comparison with state of the art. Table 1 compares our experimental results (S+I method) on the histopathology data set to the methods presented in the ICPR 2010 contest, and to [6] and [1], published since then. The overall comparison is favourable to our method, with a considerable improvement on precision and recall over all other methods.

Figure 4 includes the results of the method [1] on the HEK and HeLA data sets, kindly provided by its authors. Overall, on the HeLa data set our method was uniformly better (Figure 4(c)) (despite [1] requiring masking out the homogeneous areas of the images to remove the phantom detections), and achieves higher precision but lower recall on the HEK data set.

4 Summary

We have presented a method for cell detection in microscopy images that is able to achieve state-of-the-art performance across different scenarios. It is tolerant to changes in image intensities, cell densities and cell sizes, whilst being specific to the structures of interest. The in-built non-overlap constraint, which is taken into account during learning, allows the method to perform well even in the presence of cell clumping.

Acknowledgements. We are grateful to Dr. N. Rajpoot, Dr. E. Bernadis, Dr. B. Vojnovic and Dr. G. Flaccavento for providing cell data sets. Financial support was provided by the RCUK Centre for Doctoral Training in Healthcare Innovation (EP/G036861/1) and ERC grant VisRec no. 228180.

References

1. Bernardis, E., Yu, S.X.: Pop out many small structures from a very large microscopic image. Med. Image Anal. 15(5), 690 – 707 (2011)
2. Cheng, J., Veronika, M., Rajapakse, J.: Identifying cells in histopathological images. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010, LNCS, vol. 6388, pp. 244–252. Springer Berlin / Heidelberg (2010)
3. Graf, F., Grzegorzek, M., Paulus, D.: Counting lymphocytes in histopathology images using connected components. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010, LNCS, vol. 6388, pp. 263–269. Springer Berlin / Heidelberg (2010)
4. Gurcan, M., Madabhushi, A., Rajpoot, N.: Pattern recognition in histopathological images: An ICPR 2010 contest. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010, LNCS, vol. 6388, pp. 226–234. Springer Berlin / Heidelberg (2010)
5. Joachims, T., Finley, T., Yu, C.N.: Cutting-plane training of structural SVMs. Mach. Learn. 77, 27–59 (2009)
6. Kuse, M., Khan, M., Rajpoot, N., Kalasannavar, V., Wang, Y.F.: Local isotropic phase symmetry measure for detection of beta cells and lymphocytes. J. Pathol. Inform. 2(2), 2 (2011)
7. Kuse, M., Sharma, T., Gupta, S.: A classification scheme for lymphocyte segmentation in H&E stained histology images. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010, LNCS, vol. 6388, pp. 235–243. Springer Berlin / Heidelberg (2010)
8. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Comput. 22(10), 761 – 767 (2004)
9. Nath, S., Palaniappan, K., Bunyak, F.: Cell segmentation using coupled level sets and graph-vertex coloring. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006, LNCS, vol. 4190, pp. 101–108. Springer Berlin / Heidelberg (2006)
10. Panagiotakis, C., Ramasso, E., Tziritas, G.: Lymphocyte segmentation using the transferable belief model. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010, LNCS, vol. 6388, pp. 253–262. Springer Berlin / Heidelberg (2010)
11. Pearl, J.: Probabilistic reasoning in intelligent systems. Morgan Kaufmann (1988)
12. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML 2004. pp. 104–. ACM (2004)
13. Vedaldi, A.: A MATLAB wrapper of SVM^{struct}. `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.html` (2011)
14. Vedaldi, A., Fulkerson, B.: VLFeat. `http://www.vlfeat.org/` (2010)
15. Yin, Z., Bise, R., Chen, M., Kanade, T.: Cell segmentation in microscopy imagery using a bag of local Bayesian classifiers. In: ISBI 2010. pp. 125 –128 (2010)