ilastik: Interactive Learning and Segmentation Toolkit

Christoph Sommer, Christoph Straehle, Ullrich Köthe, Fred A. Hamprecht

Heidelberg Collaboratory for Image Processing (HCI), University of Heidelberg

January 24, 2011

Abstract

Segmentation is the process of partitioning digital images into meaningful regions. The analysis of biological high content images often requires segmentation as a first step. We propose ilastik as an easy-to-use tool which allows the user without expertise in image processing to perform segmentation and classification in a unified way. ilastik learns from labels provided by the user through a convenient mouse interface. Based on these labels, ilastik infers a problem specific segmentation. A random forest classifier is used in the learning step, in which each pixel's neighborhood is characterized by a set of generic (nonlinear) features. ilastik supports up to three spatial plus one spectral dimension and makes use of all dimensions in the feature calculation. ilastik provides realtime feedback that enables the user to interactively refine the segmentation result and hence further fine-tune the classifier. An uncertainty measure guides the user to ambiguous regions in the images. Real time performance is achieved by multi-threading which fully exploits the capabilities of modern multi-core machines. Once a classifier has been trained on a set of representative images, it can be exported and used to automatically process a very large number of images (e.g. using the CellProfiler pipeline). ilastik is an open source project and released under the BSD license at `www.ilastik.org`.

1 Introduction

ilastik is a software that combines interactive machine learning, active learning, and the ability to cope with complex textures within a convenient and unified user interface. The basic recipe behind ilastik consists of three parts: **(1)** Nonlinear image features provide a generic basis to represent diverse local image characteristics. **(2)** A state-of-the-art classifier is used to learn from user input, which is given by a paint interface. The user can define an arbitrary number of classes (e.g. background, type one, type two, etc) **(3)** In the training phase, the user can fine-tune the classifier by interactively providing new labels. To

guide the user ilastik displays several overlays. The prediction by the classifier indicates where the classifier may still be wrong. Another overlay, called *uncertainty map*, highlights regions in which the prediction is the most ambiguous. It allows to actively guide the user to difficult image regions, where the classifier deems new labels most informative. Once ilastik has been interactively trained on a set of representative images, it can be used to automatically process a very large number of images in a batch processing mode or using the CellProfiler [1] pipeline.

To demonstrate ilastik, we present experimental results in section 3 that corroborate ilastik's performance and transferability to other segmentation and classification tasks. Yet, neither problem-specific preprocessing nor special-purpose features are needed to achieve reliable results. ilastik also shows an excellent run-time performance of the underlying algorithms (in 2D and 3D). As an open source tool, ilastik is designed to be extensible without requiring a detailed understanding of the aspects of the internal software architecture. The basic ingredients of ilastik (features, classifier) can easily be extended and exchanged (e.g. using different classifiers). Whole new modules, which may offer custom functionality within ilastik can be integrated via a plug-in mechanism.

1.1 Related Work

The majority of interactive segmentation approaches are based on user seeds [2, 3, 4, 5, 6, 7]. The seeds, provided by brush strokes, act as hard constraints from which the segmentation works outwards to fill a desired region. Methods based on graph cut [5, 8, 4] extend this idea by explicitly modeling a boundary component. A segmentation is achieved by modeling the region and the boundary term as a weighted combination and optimizing the minimal cut between foreground and background seeds. Usually, the region weights are inferred from the respective seed pixels, while edge filters act as boundary weights. ilastik can be used to learn the regional properties of objects or even to explicitly learn the boundary weights between different objects. This flexibility may be a useful input to seeded segmentation techniques.

2 Methods

2.1 Feature Computation

The features are computed in the full 2D/3D pixel neighborhoods, depending on the available data. While the provided set of features includes popular color, edge and texture descriptors, the plug-in functionality allows advanced users to add their own problem-specific features. The standard features of ilastik accounting for a specific type of image structure are grouped together: **Color** and intensity consist of the raw intensity values of the image smoothed with a Gaussian. **Edge** is defined by including edge indicator functions such as the eigenvalues of the structure tensor, eigenvalues of the Hessian matrix, gradient

magnitude, difference of Gaussians and Laplacian of Gaussian. **Texture** comprises the eigenvalues of the structure tensor, eigenvalues of the Hessian matrix. **Orientation** comprises the raw structure tensor and Hessian matrix entries. All these groups can be selected on 7 scales, with a total of 35 different choices. The user is free to combine these selections in order to generate an appropriate feature set for the problem at hand.

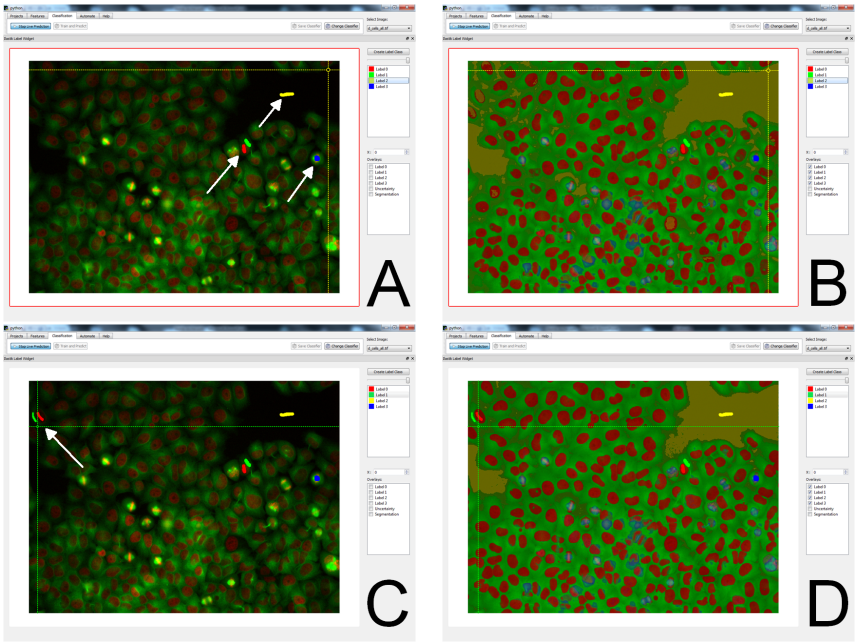


Figure 1: The basic ilastik work flow: (A) Initial labels, (B) live prediction mode showing interactive prediction, (C) a few more labels to correct for wrong classification due to illumination inhomogeneity, (D) updated classification result

In order to provide feedback to the user as to which features were important for a specific classification task, ilastik outputs a variable importance (mean Gini decrease [9]). This refers to the process of selecting a subset of relevant features from the entire set. In particular, for applications (e.g. dealing with multi-spectral data) in which each variable has an associate interpretation the detection of informative features can facilitate the imaging procedure [10].

2.2 Classification and Active Learning

The ability of the random forest to capture highly non-linear decision boundaries in feature space is a major prerequisite for the application to general sets of use cases. The classification is performed using the random forest classifier introduced in [9]. Random forests consist of many decision trees. The individual

trees are not pruned in the training phase and are built under random influence: (1) Each tree is built based on a bootstrap sample (bag) of the training data . The *out-of-bag* samples are used to estimate of the real test error. (2) In each node only a random subset of candidate features are evaluated to find the best split according to the Gini impurity.

During prediction, each pixel is classified by collecting the votes of each individual tree. The ratio of the tree votes is interpreted as a posterior probability and provides the basis of the segmentation step. ilastik defaults to train $N_T = 100$ decision trees. Besides the prediction ilastik computes a *uncertainty map*, which guides the user to ambiguous regions in the image. For this purpose, ilastik implements the *margin* [9] of the classification.

2.3 Graphical User Interface

To illustrate the basic work flow of ilastik[1], we consider the problem of cell segmentation. The cell images consist of two channels. One channel shows the cell nuclei (red) while the other shows the cytoplasm (green). Some of the cell nuclei are of the mitosis phenotype (condensed nuclei, appearing yellow). To discriminate these classes and the background one would use ilastik in the following way: **(0)** Create new project and add the RGB images to it **(1)** Create four label classes (background, nuclei, mitotic nuclei, cytoplasm), **(2)** give some label strokes to indicate the four different objects, **(3)** compute color features, **(4)** switch to live prediction mode, **(5)** correct with more labels until a satisfactory result is achieved. This process is illustrated in Figure 1.

3 Results

To evaluate the performance of ilastik, we show results on three biological data sets.

3D neuron data: The first data set (courtesy of Graham Knott, EPF Lausanne) is a 3D-volume containing neurons, which include mitochondria and vesicles. One aim in preprocessing for automated neuron segmentation is to distinguish between the four classes neuron interior (red), neuron boundary (green), mitochondria (yellow), and vesicles (blue).The high-resolution volume in Figure 2 shows a 3D sub-cube (150^3 voxels) of the mammalian brain recorded with the FIBS-technique (Focused Ion Beam). ilastik computes the image features in 3D and can thus benefit from the isotropic resolution of the 3D volume. Figure 2 shows that after training with a few examples ilastik is able to predict other instances of the same kind (e.g. mitochondria in the upper left slice view).

Fundus images: The publicly available STARE database [11], consists of 20 retinal images. The aim is to segment low-contrast blood vessels in retinal fundus images. Two observers manually segmented all images. Performance is estimated by taking the segmentation of the first observer as ground truth. For the sake of comparability, the evaluation procedure of Soares et al. [12]

[1] Also consider the tutorial videos on `www.ilastik.org`

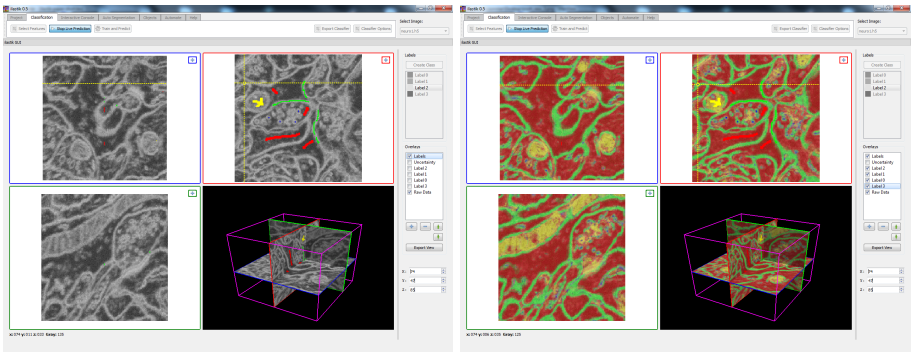
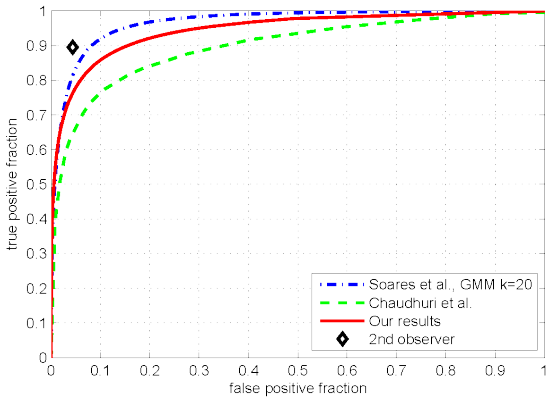


Figure 2: ilastik for 3D volumes: (top) initial labels in different slices, (bottom) prediction after first classification round into four classes: neuron interior (red), neuron boundary (green), mitochondria (yellow) and vesicles (blue)

is adopted which includes leave-one-image-out cross-validation and ROC-curve analysis. We labeled the 20 images from the STARE database in approximately 1h, resulting in an average of 0.84% of foreground and 11% background coverage. Every image was segmented using the user labels from the other 19 images of the training set. Performance, however, was computed against the ground truth segmentation. The segmentation was produced by thresholding the smoothed probability map. The smoothing with a Gaussian filter ($\sigma = 1$) was used to reduce spurious vessel detections. The results were compared to the matched filter results from Chaudhuri et al. [13], 2-D Gabor wavelets from Soares et al. [12], adaptive local thresholding scheme from Jiang et al. [14], and the ridge-based segmentation proposed by Staal et al. [15]. Performance was measured using ROC curves (see Figure 3). Results indicate that ilastik achieves competitive performance without a problem-specific development effort.

Cell counting: Identification and segmentation of cells and their phenotype is a standard task in biological image processing. Especially in high-throughput experiments, it facilitates the study of many normal, neoplastic and replication processes. A main discipline in that area is the counting of cells with a particular phenotype of interest. We use the *Human HT29 Colon Cancer 1* image set [16] published in the Broad Bioimage Benchmark Collection. Some of the cells are mitotic and appear slightly brighter. The background and the two different types of cell nuclei types were marked by a user in about 5 minutes, resulting in 5.12% background and 0.37% cell nuclei (normal and mitotic) label coverage. Once again leave-one-image-out cross-validation was performed. The probability maps from the supervised classification were fed into a modified marker-based watershed transform (not yet part of ilastik). Seeds for the three different classes are generated by smoothing ($\sigma = 1$) and thresholding ($t = 0.5$) each probability map. The actual watershed transform is computed on the gradient of the background probability. The ground truth for this data set is the total cell count of two observers for each image. The average absolute deviation

from the mean count of the two observers is 10.47% while the two human observers vary by 11% for this image set. The average cell count of the algorithm is slightly higher than the two human counts due to over-segmentation. While the human observers merely counted all cells in this data set, ilastik predicts each cell to be mitotic or not.



Method	A_z	Accuracy
Soares et al. [12]	0.967	0.948
Chaudhuri et al. [13]	0.899	–
Jiang et al. [14]	0.930	0.901
Staal et al. [15]	0.964	0.952
our results	0.945	0.959

Figure 3: ROC analysis for fundus images after leave-one-image-out cross-validation

Outlook: ilastik is released in version 0.5 and available at `www.ilastik.org`. Future versions are already in the making and will include: (1) seeded segmentation algorithms such as watershed (2) unsupervised dimension reduction techniques (e.g. PCA, PLSA) to help condense information from image features and multi-spectral images. (3) Hierarchical processing by allowing subsequent analyses on top of intermediate results (e.g. interactive learning inside a previously learned mask).

4 Conclusion

In this paper we proposed ilastik to tackle standard image processing tasks (2D and 3D) in the field of biomedical image processing without resorting to programming expertise. ilastik combines a user friendly interface with a set of state-of-the-art algorithms. The approach is robust and work across many types of images and volumes. The segmentation framework is extended in a multi-class sense to allow for different object types and problems.

ilastik is limited to local cues such as brightness, color and texture and is not designed to capture global configurations. For many standard problems,

however, ilastik yields results of surprisingly good quality and hence allows to solve problems that previously would have required hand-tailored algorithms.

We release ilastik under the open source BSD license to endorse the idea of free software. It can be used and extended without any restrictions. We expect ilastik to be a useful tool in many applications in the field of biomedical image processing and invite developers to contribute in a collaborative setting.

References

[1] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes," *Genome Biol.*, vol. 7, pp. R100, 2006.

[2] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647, 1994.

[3] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1768–1783, 2006.

[4] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[5] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," *ICCV'01*, vol. 1, pp. 105–112, 2001.

[6] G. Friedland, K. Jantz, and R. Rojas, "SIOX: Simple interactive object extraction in still images," in *Seventh IEEE International Symposium on Multimedia*, 2005.

[7] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[8] B.L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3161–3168.

[9] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[10] M. Jehle, C. Sommer, and B. Jähne, "Learning of optimal illumination for material classification," in *Pattern Recognition*. 2010, pp. 563–572, Springer.

[11] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response," *IEEE Trans. on Medical Imaging*, vol. 19, no. 3, pp. 203–210, 2000.

[12] B Soares, G. Leandro, M. Cesar, F. Jelinek, and J. Cree, "Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification," *IEEE Trans. on Medical Imaging*, vol. 25, no. 9, pp. 1214–1222, 2006.

[13] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum, "Detection of blood vessels in retinal images using two-dimensional matched filters," *IEEE Trans. on Medical Imaging*, vol. 8, no. 3, pp. 263–269, 1989.

[14] X Jiang and D. Mojon, "Adaptive local thresholding by verification-based multi threshold probing with application to vessel detection in retinal images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 131–137, 2003.

[15] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[16] J. Moffat and et al., "A lentiviral RNAi library for human and mouse genes applied to an arrayed viral high-content screen," *Cell*, vol. 124, no. 6, pp. 1283–1298, 2006.