

INAUGURAL – DISSERTATION

ZUR
ERLANGUNG DER DOKTORWÜRDE
DER
NATURWISSENSCHAFTLICH–MATHEMATISCHEN–GESAMTFAKULTÄT
DER
RUPRECHT–KARLS–UNIVERSITÄT
HEIDELBERG

vorgelegt von
Dipl.-Inform. Christoph Sommer
aus Freiburg

Tag der mündlichen Prüfung: 05. Juli 2010

Interactive Learning for the Analysis of Biomedical and Industrial Imagery

Gutachter: Prof. Dr. Fred A. Hamprecht

Prof. Dr. Bernd Jähne

Abstract

In this thesis methods and applications of supervised learning for the segmentation and analysis of digital imagery coming from a variety of research domains are investigated. Segmentation and classification are important tasks in biomedical and industrial imaging and often provide the basis for recognition and quantification. Various specialized solutions exist for an enormous amount of distinct types of data and these are usually designed to meet and exploit given domain knowledge. In this work, an interactive supervised learning framework is proposed that is able to tackle multi-object segmentation and multi-class discrimination in a unified way. The method is general enough to cover a reasonable range of use cases in which local image descriptors are sufficient. The performance of the segmentation results is demonstrated on various data sets with distinct tasks to solve. This emphasizes the versatility of this approach to many biomedical and industrial data sets without requiring explicit image processing expertise and the need for custom programming. The approach builds upon a generic feature set that is able to characterize local cues such as color, texture and edges. To this end, an interactive tool that performs real-time processing on usual image sizes was developed, enabling domain experts to perform segmentation and classification tasks in an explorative fashion. No prior expertise in image processing is required since user interaction is facilitated via intuitive brush strokes. Once the algorithm/system has been trained, it can be applied to thousands of images with no further interaction with the user. The approach is limited to the segmentation of objects that can be discriminated based on local cues such as color or texture; but within this setting, the supervised framework yields surprisingly good results; on top of those, application-dependent post-analysis can be applied. The framework supports up to 4-dimensional multi-spectral data in an integrated fashion.

In order to show the applicability and transferability of the method, several real world data sets – from very diverse imaging fields – are examined. Among them is the segmentation of tumor tissue from fluorescent wide-field microscopy, quantification of cell migration in confocal microscopy images for surveys on adult neurogenesis, segmentation of blood vessels in the retina of the eye, tracing of copper wires spread on tags for brand-owner authentication in an industrial context, and the application to image quality control for high-throughput siRNA screens. Furthermore an industrial problem is considered: a novel sequence classification procedure on the basis of localized frequency estimates is proposed for the process control and visualization of the sheet-feeding process for offset printing machines.

Zusammenfassung

In der vorliegenden Dissertation werden Methoden des überwachten Lernens untersucht und auf die Analyse und die Segmentierung digitaler Bilddaten angewendet, die aus diversen Forschungsgebieten stammen. Die Segmentierung und die Klassifikation spielen eine wichtige Rolle in der biomedizinischen und industriellen Bildverarbeitung, häufig basiert darauf weitere Erkennung und Quantifikation. Viele problemspezifische Ansätze existieren für die unterschiedlichsten Fragestellungen und nutzen meist spezifisches Vorwissen aus den jeweiligen Bilddaten aus. In dieser Arbeit wird ein überwachtes Lernverfahren vorgestellt, das mehrere Objekte und deren Klassen gleichzeitig segmentieren und unterscheiden kann. Die Methode ist generell genug um einen wichtigen Bereich von Anwendungen abzudecken, für deren Lösung lokale Merkmale eine Rolle spielen. Segmentierungsergebnisse dieses Ansatzes werden auf verschiedenen Datensätzen mit unterschiedlichen Problemstellungen gezeigt. Die Resultate unterstreichen die Anwendbarkeit der Lernmethode für viele biomedizinische und industrielle Anwendungen, ohne dass explizite Kenntnisse der Bildverarbeitung und Programmierung vorausgesetzt werden müssen. Der Ansatz basiert auf generellen Merkmalsklassen, die es erlauben lokal Strukturen wie Farbe, Textur und Kanten zu beschreiben. Zu diesem Zweck wurde eine interaktive Software implementiert, welche, für gewöhnliche Bildgrößen, in Echtzeit arbeitet und es somit einem Domänenexperten erlaubt Segmentierungs- und Klassifikationsaufgaben interaktiv zu bearbeiten. Dafür sind keine Kenntnisse in der Bildverarbeitung nötig, da sich die Benutzerinteraktion auf intuitives Markieren mit einem Pinselwerkzeug beschränkt. Das interaktiv trainierte System kann dann ohne weitere Benutzerinteraktion auf viele neue Bilder angewendet werden. Der Ansatz ist auf Segmentierungsprobleme beschränkt, für deren Lösung lokale diskriminative Merkmale ausreichen. Innerhalb dieser Einschränkung zeigt der Algorithmus jedoch erstaunlich gute Resultate, die in einer applikationsspezifischen Prozedur weiter verbessert werden können. Das Verfahren unterstützt bis zu vierdimensionale, multispektrale Bilddaten in vereinheitlichter Weise.

Um die Anwendbar- und Übertragbarkeit der Methode weiter zu illustrieren wurden mehrere echte Anwendungsfälle, kommend aus verschiedenen bildgebenden Bereichen, untersucht. Darunter sind u. A. die Segmentierung von Tumorgewebe, aufgenommen mittels Weitfeldmikroskopie, die Quantifikation von Zellwanderungen in konfokalmikroskopischen Aufnahmen für die Untersuchung der adulten Neurogenese, die Segmentierung von Blutgefäßen in der Retina des Auges, das Verfolgen von Kupferdrähten in einer Anwendung zur Produktauthentifikation und die Qualitätskontrolle von Mikroskopiebildern im Kontext von Hochdurchsatz-Experimenten. Desweiteren wurde eine neue Klassifikationsmethode basierend auf globalen Frequenzschätzungen für die Prozesskontrolle des Papieranlegers an Druckmaschinen entwickelt.

Erklärung

Hiermit erkläre ich, Christoph Sommer, dass ich die vorgelegte Dissertation selbst verfasst und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe.

Heidelberg, den 30. März 2010

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my doctoral adviser, Prof. Dr. Fred A. Hamprecht, for his support and encouragement over the last three years. His group, in which I have had the pleasure to participate, provided me with a friendly and fruitful environment of beaming enthusiasm. I am also greatly indebted to Dr. Ullrich Köthe for his guidance and many fertile discussions. In particular, I would like to thank Dr. Tobias Müller from Heidelberger Druckmaschinen AG, who gave me the opportunity to work in a most interesting field of application: printing machines. I am equally grateful to Kathleen Börner, Nina Corsini, and Dr. Maik Lehmann. I had an amazing time and the experience of working with such brilliant life scientists was an inspiring one.

I would also like to thank my colleagues and fellow doctoral students at Heidelberg Collaboratory for Image Processing for the many friendly and enlightening discussions and the good times spent over the past years. These are Björn Andres, Nikos Gianniotis, Michael Hanselmann, Frederik Kaster, Jens Röder, and Andreas Walstra. It has been a pleasure to work with you.

Many thanks go to Dominik, the best friend one could wish for, and to Moni, for her love and patience. Last but not least, I would like to thank my parents for their warmth and support during my time as a student.

*'The time has come,' the Walrus said,
 'To talk of many things:
Of shoes – and ships – and sealingwax –
 Of cabbages – and kings –
And why the sea is boiling hot –
 And whether pigs have wings.'*

by Lewis Carrol

Contents

I. Introduction and Basics	1
1. Introduction	3
1.1. Related Work	7
1.2. A Guide to this Thesis	9
2. Basics	11
2.1. Feature Extraction	11
2.2. Dimension Reduction	13
2.2.1. Principal Component Analysis	14
2.3. Supervised Learning Methods	14
2.3.1. Bootstrap Aggregation	15
2.3.2. Classification and Regression Trees	16
2.3.3. Random Forests	17
2.3.4. Linear Support Vector Machines	19
2.3.5. Kernel Trick	20
2.4. Evaluation Methods	21
2.4.1. Cross-Validation	21
2.4.2. Receiver Operator Characteristics Curve	21
II. Interactive Framework and Applications	23
3. The Interactive Framework and its Properties	25
3.1. General Framework	25
3.1.1. Image Features	26
3.1.2. Categorization of Features	27
3.1.3. Pixel-Wise Classification	27
3.1.4. Feature Selection	28
3.1.5. Application-Dependent Quantification	28
3.1.6. Interactive Classification	28
3.2. The Graphical User Interface	29

3.3.	Evaluation of the Framework	29
3.3.1.	Retinal Imaging	30
3.3.2.	Fluorescence Wide Field Imaging	31
3.4.	Experimental Evaluation	32
3.4.1.	STARE Database	32
3.4.2.	Human HT29 Colon Cancer 1	33
3.5.	Discussion	34
3.5.1.	Related work	34
3.5.2.	Conclusion	35
4.	Adult Neurogenesis	37
4.1.	Introduction	37
4.1.1.	Adult Neurogenesis	37
4.1.2.	Material	38
4.1.3.	Image Acquisition	40
4.2.	Methods and Workflow	40
4.2.1.	Overview	40
4.2.2.	Related Work	42
4.2.3.	Feature Generation	43
4.2.4.	Random Forest	43
4.2.5.	Segmentation	44
4.2.6.	Interface	46
4.3.	Results	47
4.3.1.	Evaluation	47
4.4.	Conclusion	50
5.	Tumor-Stroma Interaction	53
5.1.	Introduction	53
5.1.1.	Tumor-Stroma Interaction	53
5.1.2.	Related Work	54
5.2.	Material and Methods	55
5.2.1.	Material	55
5.2.2.	Methods	56
5.3.	Results	60
5.4.	Conclusion	61
6.	Viroquant and Image Quality Control	65
6.1.	Introduction	65
6.2.	RNA Interference	66
6.3.	Related Work	67

6.4.	Preparation and Experimental Setup	68
6.4.1.	Image Acquisition	69
6.5.	Image Analysis	69
6.5.1.	Cell Nuclei Segmentation	70
6.5.2.	Cell Nuclei Marker Extraction	70
6.5.3.	Cell Segmentation	71
6.5.4.	Watershed Segmentation	71
6.5.5.	Quantification	72
6.6.	Image Quality Control Using One-Class Learning	75
6.6.1.	One-Class Support Vector Machine	75
6.7.	Implementation and Pipeline	76
6.7.1.	Graphical User Interface	77
6.7.2.	Parallel Pipeline	77
6.8.	Results and Conclusion	78
7.	Copper Wire Tracing	81
7.1.	Overview	81
7.2.	Material and Methods	83
7.2.1.	The Data	83
7.2.2.	Feature Generation	83
7.2.3.	Skeletonization	85
7.3.	Results	85
7.3.1.	Conclusion	85
8.	Process Control for the Sheet Feeder	87
8.1.	The Sheet Feeder - An Overview	87
8.1.1.	Experimental Setup	89
8.2.	Data Acquisition	89
8.2.1.	Data Sets	90
8.3.	Feature Extraction	91
8.3.1.	Short-Time Fourier Transform	92
8.3.2.	Dimension Reduction	94
8.3.3.	Feature Selection with Random Forest	94
8.4.	Classification	95
8.5.	Online Visualization of Process State	97
8.5.1.	Self-Organizing Maps	98
8.6.	Discussion	100
9.	Conclusion	101
10.	Outlook	105

List of Figures	109
Bibliography	113

Part I.
Introduction and Basics

1. Introduction

Analysis, processing and quantification of digital imagery is a challenging research area that demands advanced image processing techniques. The great variety in biological, medical and industrial imagery gives rise to problem-specific algorithm design. Image segmentation is a crucial step toward recognizing and understanding image content. Segmentation in the context of image processing has been defined as:

“The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.”

(Linda Shapiro)

With this in mind, image segmentation is a prerequisite step for many recognition and quantification problems; it aims to provide a basis for drawing conclusions on a more abstract and semantic ground than mere image pixels. Despite the fact that several thousand scientific papers are published every year in this research area, the segmentation problem is still one of the main unsolved problems in image processing. However, many approaches that solve segmentation tasks in specific data domains by utilizing prior knowledge (e. g. objects of interest are elliptically shaped) or prior data normalization (e. g. objects of interest are centered in each image) do exist. In addition, semi-automatic approaches, where a computer aids a user in accomplishing a certain segmentation task, have been proposed in the past. Most of these approaches are bound to work on only one image: the image the user acts on, and all user input must be repeated for each and every data item; further images must be processed independently as they cannot utilize previously entered user input. On the other hand, fully automated methods lack the ability to generalize into distinct segmentation tasks and are designed to capture a combination of predefined object characteristics, such as texture, color, or shape.

Over the last 15 years, machine learning has emerged as a key component in many image processing applications and a major ingredient in related research fields such as computer vision, pattern recognition and computational neuroinformatics. Machine learning methods are used to estimate parameters of a given model from the available data that would have been set manually otherwise. Learning from examples in a supervised fashion has proven to be a fruitful methodology to improve the reliability and accuracy of image segmentation in various data domains. However, reliable automated

segmentation techniques are still restricted to specific data sets, whereas generic approaches lack the desirable accuracy. Similar statements apply also with respect to object discrimination.

We focus on supervised image analysis techniques that can be adapted to new with little effort expended by the user. As a first step, segmentation is certainly one of the most crucial tasks. The quality of further analysis is highly dependent on the accuracy and relevance of the segmentation results. Fully automated techniques can usually handle limited variability in data appearance, while generic approaches attempt to facilitate any kind of image data, which is still an unsolved problem. In addition, the identification of different types of objects is of vital importance for many problems. Hence the generic formulation of segmentation and classification in a unified manner can help address many standard problems in the field of biomedical imaging.

On a practical level, the goal is to provide a software tool for interactive exploration and segmentation of image data covering a wide range of biomedical and industrial use cases that do not require higher-level semantic image understanding. Fast algorithms and efficient implementation, together with a convenient interface, are the basic ingredients of the supervised classification procedure. Offering effective classifiers, feature selection and the support of up to four-dimensional multi-spectral data comprise the major focus of this work. The user is able to label objects of interest via an intuitive graphical user interface. The restriction to two classes is removed (foreground and background as often imposed in other approaches) and allow for any number of classes, as required by the task at hand. The classifications and segmentations are based on a large number of features, which are computed for every pixel's neighborhood. These features include filter bank responses, differential properties and statistical measurements, thus encoding the appearance of each pixel's neighborhood on different scales. Given the features and the user's labels, a classifier is trained in a batch or incremental fashion. Here, the focus lies on the random forest classifier. The class probabilities computed by the classifier are then used to segment the image.

The proposed framework has been successfully applied to various biomedical and industrial standard applications. The graphical user interface was initially implemented in Matlab¹ using the underlying algorithms written in C++ [62]. Matlab – as a rapid prototyping language for scientific calculations – is rather limited in its handling of more advanced graphical effects (e. g. transparent overlays, 3D) and multiple threads, both of which are indispensable for concurrent processing. Hence, this prototype was ported and extended to Python/Qt, still interfacing the C++ algorithms. Python was chosen due to its excellent support of various libraries (e. g. boost, OpenMPI, HDF5, Qt, etc.) and its elaborate object model, enabling us to benefit from both worlds: rapid

¹Convenient Learning and Segmentation Interface for Matlab: <http://hci.iwr.uni-heidelberg.de/people/csommer/clasi/>

development cycles and fast executions times.

The applications described within this thesis range from cell classification with concurrent segmentation to the tracing of fine copper wire structures in an industrial context. Many interesting data sets and questions arising in life sciences have been successfully addressed and give witness the performance of the proposed approach. A brief overview of the applications processed with this framework follows:

- Cell classification is a frequent task in micro biological experiments. A standard approach is to segment cell nuclei first (which is usually an easier task, due to good separation of the nuclei) in order to have a good starting point for analyzing cell morphology. The features computed from whole cells (area, eccentricity, mean gray value, etc) serve as the input for the final classification step. For various phenotypes of cells the tasks can be accomplished in a more direct manner. Often the local appearance and structure of cells convey important aspects of their phenotype. Thus the problem can be tackled in a direct low-level manner by providing discriminant features of pixel's neighborhoods to an effective classification system. An appropriate standard segmentation method used on top of the classification stage then yields the required readouts, such as counts of cells with their corresponding type.
- To emphasize the transferability of the classification-based approach, a segmentation problem focusing on fine elongate structures is presented. The blood vessels in the retina of mammal eyes provide ophthalmologists with information about the health of the eye. The approach fares well when compared to other state-of-the-art methods on this problem, while maintaining good computational performance. Another example from ophthalmic medicine depicts the inherent support of multi-class problems. The discrimination of cellular layers from a cross section of the retina is performed in exactly the same way and is able to distinguish five different cell layers reliably without incorporation of prior domain knowledge.
- The studies on adult neurogenesis have become important precursors for a better understanding of the mammal brain. Hundreds of volumes of data that came from cross sections of mice brains were analyzed² in this work to monitor the state of newly originated stem cells. The suggested framework was applied to various fluorescent channels in order to quantify the state and maturity of the neurons' development process. Especially valuable in this application is the fact that the approach is easily adaptable to three-dimensional volume data.
- Tumor-stroma interaction and blood vessel densities are important factors for tu-

²In collaboration with Jörg Greis [40] and the Ana Martin-Villalba Laboratory (DKFZ) within the SB-Cancer Project as part of the Helmholtz Initiative for Systems Biology

mor growth. The method was applied to segment tumor and stroma tissues³, which is the basis of estimating blood vessel densities. In this case, the classification was not applicable to the regions (tissues) themselves (due to the lack of significant local evidence for either tissue type), but to the border separating the tissues. A customized segmentation method had to be developed to trace this *learned* border.

- High-throughput RNAi screens help reveal genetic factors for the pathways involved in virus infection. In this thesis, the image processing routines and the subjacent data-flow pipeline were developed to analyze millions of images in a few days⁴. The fully automated analysis requires effective quality control methods in order to ensure its reliability.
- To that end, the main approach is adapted to play the role of *image quality control*. Multi-class learning is replaced by one-class learning, the class of clean images. A classifier is trained on a sufficient number of images that do not exhibit artifacts. After training, the classifier can detect unexpected artifacts like dust, pollution or hairs in unseen images. Subsequent tasks performed on the processed data set, clearly benefit from the exclusion of such artifacts.
- As a further demonstration of the versatility of the approach, a industrial problem is presented that is concerned with obtaining a segmentation of thin copper wires spread on printing tags. A set of selected features were identified to feed the classifier, together with user annotations. The resulting probability map of the two-class problem then served the input to skeletonization algorithm to construct a continuous one-pixel line for each wire segment found in the printing tag.

Global features based on frequency analysis served as a predictor for the estimation of the error state for the process control of the sheet-feeder in printing machines. In this collaboration with a printing machine manufacturer, a novel technique has been developed that classifies image frames from video sequence data from inside the feeder into normal or error states. Sequences were also acquired under variation of important control parameters. These were used to train an online state visualizer, which could then be implemented into a press to aid the machine operator in adjusting feeder parameters.

³In collaboration with Stephan Kassemeyer [53] and the Margareta Müller Laboratory (DKFZ) within the SBCancer Project as part of the Helmholtz Initiative for Systems Biology

⁴Part of the RNAi screening facility in BioQuant. In collaboration with Kathleen Börner, Maik Lehmann, Julian Kunkel, Lars Kaderali and Hans-Georg Kräusslich. [15]

1.1. Related Work

Several interactive learning methods, with numerous applications and objectives, have already been proposed in literature. The most prominent approach of the last few years was the work of Boykov and Jolly [10], which was extended by Rother et al. to the famous *GrabCut* [89] framework. Their approach is based on basic and fast learning procedures that use the pixel's color distribution. Histograms or fitted Gaussian mixture models with rather few components are used as representations of the foreground and background model. Because these are iterated procedures, the algorithm begins with a few initial user labels (often called seeds) and then gradually refines the color space representation. In order to achieve this, new labels are generated from a *graph cut* [39, 11] segmentation. Graph cut is an efficient (polynomial time) algorithm and a globally optimal method for computing the solution for a submodular energy [61] functional (typically living on a lattice graph) and is one optimization strategy for MRF-like problems [61]. An inherent property of graph cut is the restriction that it is only guaranteed to find the globally optimal solution in binary segmentation problems. However, approximative methods exist in which the problem with multi-valued variables is reduced to a sequence of graph cut subproblems with binary variables [11]. Despite these restrictions, GrabCut is a highly suitable and efficient interactive technique for segmenting color images into a binary partition.

Other, similar approaches have been proposed: Simple Interactive Object Extraction (SIOX) [30] is based on clustering and morphological post-processing. The idea of formulating the graph cut problem on precomputed super-pixels was adapted, which speeds up processing time immensely (Lazy Snapping [69]), but it can also be negatively affected by under segmentation of the super pixels. Because they utilize simple pixel descriptors in the underlying color space, methods based on interactive graph cut may fail when encountering textured or highly cluttered images. Han et al.'s work proposed, [43] the integration of *multi-scale structure tensor* into the GrabCut framework. The authors showed that this step can overcome the restriction of a pure color representation, although it is still bound to single specific descriptors.

In contrast, the approach followed in this thesis can inherently cope with arbitrary input domains, including multi-spectral data. It is not sensitive to feature scaling, such as the Gaussian mixture model and enables multi-class segmentation in a unified way. A very recent publication follows a very similar idea. In their work, Santner et al. [93] proposed the use of random forest classification, together with *Histograms of Oriented Gradients (HOGs)* [23] computed as pixel descriptors for color images in several orientations and scales. Their method, which used a supervised approach, is in general not restricted to those types of features and possibly could include arbitrary feature sets. They also made use of a *Total Variational* regularization approach to computing the final segmentation. Total variational methods in general aim to optimize a convex energy functional, comprising a point-wise data term and regularizer, which penalizes

non-smooth solutions by means of total variation. Although promising results were presented, it is not obvious how this idea could be extended to multi-class segmentation problems.

A major component of the framework is the incorporation of effective classifiers. The random forest classifier (see Section 2.3.3) has been shown to be suitable for interactive learning. In several other domains related to image processing, random forest was also successfully applied. In the work of Bosch et al. [9] random forest was used for image categorization by learning a *bag of visual words* together with the *spatial pyramid* representation [65]. Random forest was extended to clustering in [79] in order to learn a visual codebook for object recognition. A tracking application presented in [68] relies on random forest classification for fast key point detection. In addition, random forests were combined with Hough voting in [33] in application to pedestrian recognition. In [112], the authors emphasize the much faster training and prediction times of random forests as compared to traditional classifiers. Recent examples of specific 3D image segmentation where random forest are used can for instance be found in [67, 3].

The authors in [5] describe a related method to the one proposed here. They also employ local features, classifiers and feature selection for adaptive binary segmentation. A pruned decision tree is applied to classify the data samples. For each predefined scale a tree was evaluated separately, thus restricting the approach to objects present at a single scale. Under typical conditions the training and prediction time of random forests are near real-time, whereas the results reported in [5] took several minutes. This was due to the employed forward selection [44] scheme and various parameter tunings during the classification process.

Interactive segmentation tools exist for a wide range of target groups and fields of application. In the area of digital photography, an important task is extracting a foreground object from the background. The standard tool set supported in almost all standard software comprises the magic wand and the intelligent scissors tool. The magic wand selects color values and groups unmarked pixels that are similar to the selected ones. Intelligent scissors [80] (also called magnetic scissors) is a boundary oriented path tool to aid the user in contouring the object of interest, letting the selection snap to image edges in a data-adaptive manner. The previously mentioned approaches, such as SIOX and GrabCut, also made their way into image editing related software. SIOX, for example, is integrated into GIMP (GNU Image Manipulation Program), while the latter will be part of MS Office 2010⁵.

Of particular interest to life sciences are tools that interactively segment microscopic or MRF images. Tools such as ITK-Snap [114], Fiji [110] (image processing package based on ImageJ) or Seg3D [83] offer 3D visualization in combination with manual and (semi-)automatic thresholding. These tools, which provide a convenient user interface,

⁵for more information see <http://blogs.technet.com/office2010/archive/2009/11/30/more-about-background-removal-in-office-2010.aspx>

are used to examine and condition single data items. Basic image processing techniques such as user initialized snakes [52] are integrated into ITK-Snap, for example. Commercial software, such as Amira [37], offers comfortable interfaces to basic semi-automatic segmentation techniques with integrated real time 3D rendering.

In contrast to the interactive learning tool developed as part of this thesis, these tools are not explicitly designed to process unseen data. The incorporation of more expressive feature sets is vital for various applications beyond color image foreground extraction. The developed methodology is able to cope with multi-classes and multi-objects in a unified manner and can be customized for application-specific demands as needed. The combination of discriminative local features, together with effective classifiers, is also applicable to a wide range of standard settings in life sciences and industrial image processing. The interactive tool demonstrates excellent run-time performance of the underlying algorithms. Thus the concepts described in this thesis may be helpful in lessening the gap between image analysis research and real-world applications in other research domains.

Since several applications are presented in this thesis, the related work is reviewed separately in the corresponding sections of the application description.

1.2. A Guide to this Thesis

This thesis starts by providing a brief overview of the theory behind several tools that were used in this work: Feature extraction, dimension reduction, supervised classifiers and evaluation methods (Chapter 2). In the next part, the framework for learning the segmentation from user labels is introduced along with details regarding the developed software tool. Chapter 4 provides details on applications for data arising in adult neurogenesis and explains both the biological background and the customized quantification procedure. The evaluation of achieved results is also discussed. The quantification of blood-vessels in tumor-stroma tissues is described in Chapter 5. Also the globally optimal shortest path algorithm applied is depicted in more detail there. The developed image processing pipeline in the context of high-throughput RNAi screening and its application to two large assays can be found in Chapter 6. Details on the technical implementation on the ViroQuant RNAi platform are also described. Image quality control, which is a relevant part of the pipeline, is introduced in that chapter and the results are also described. Chapter 7 reviews an industrial image segmentation task and depicts the solution achieved using the classification-based approach; results are compared to the standard method. The work on process control for sheet feeders in offset printing machines can be found in Chapter 8.

In Chapter 9 the thesis is summarized and conclusions are drawn about the presented work. A outlook is given in 10. In addition, each application chapter contains separate introduction, conclusion and outlook sections since applications and their various issues

differ widely in the embedding research context.

2. Basics

In this chapter, the basic methods and algorithms involved are described. Typically, a recognition system comprises several steps. The feature extraction maps the raw data samples into a (usually) high-dimensional features space in order to characterize relevant aspects of the data. Afterwards, an optional step is to compact the information – now contained in the feature representation – via dimension reduction. Classification yields the desired mapping: from features of data samples to their corresponding class membership.

2.1. Feature Extraction

To achieve a successful classification it is crucial to compute a sufficiently discriminative feature vector for each data point. Because this thesis, deals with multi-dimension data, data points are pixels or voxels $\mathbf{x} \in V$, where V is a two-dimensional or volumetric image. Mere processing of individual image elements (pixel or voxels) using point operations does not provide information about spatial relations present in local neighborhood. Hence, features are computed using neighborhood operations that are able to capture such local structures, intensity values and textures. A neighborhood around \mathbf{x} consists of a discrete set of image elements within a small sub window centered at \mathbf{x} . Typically, such features are nonlinear filters comprising morphological spectra, in particular opening- and closing operations, rank-order filters, non-linear transformations and eigenvalues of structure tensors and a Hessian matrix. In the following, a brief description of the features that were employed in this work is presented. For a more thorough presentation, the reader is referred to Jähne [47].

For each voxel \mathbf{x} , the feature vector $f(\mathbf{x})$ of length P comprises the filter response $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_P(\mathbf{x})]$. The filters f_i are:

- **Eigenvalues of the structure tensor:** A spatial neighborhood representation that can distinguish between constant values and an isotropic orientation distribution is the squared scalar product $(\nabla g^T \mathbf{n})^2$ between the neighborhood gradient vector $\nabla g(\mathbf{x})$ and the unit vector \mathbf{n} , which denotes the direction of local orientation. Thus, to determine the local orientation within a neighborhood that is described by a window function w the convolution

$$\int w(\mathbf{x} - \mathbf{x}') (\nabla g^T \mathbf{n})^2 d\mathbf{x}'$$

has to be maximized. This corresponds to $\mathbf{n}^T \mathbf{J} \mathbf{n}$ with the structure tensor \mathbf{J} that contains the components:

$$\mathbf{J}_{pq}(\mathbf{x}) = \int w(\mathbf{x} - \mathbf{x}') \left(\frac{\partial g(\mathbf{x}')}{\partial \mathbf{x}'_p} \frac{\partial g(\mathbf{x}')}{\partial \mathbf{x}'_q} \right) d\mathbf{x}'$$

The tensor is symmetric and characterizes orientation and coherence of local structures as described in Jähne [47]. Implementation consists of a pixel wise multiplication of the convolutions of the image with each discrete partial derivative operator and a subsequent smoothing using a Gaussian filter.

- **Eigenvalues of the Hessian matrix:** All combinations of second-order partial derivatives form the Hessian matrix:

$$H = \begin{pmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial xy} & \frac{\partial^2}{\partial xz} \\ \frac{\partial^2}{\partial xy} & \frac{\partial^2}{\partial y^2} & \frac{\partial^2}{\partial yz} \\ \frac{\partial^2}{\partial xz} & \frac{\partial^2}{\partial yz} & \frac{\partial^2}{\partial z^2} \end{pmatrix}$$

As in the case of the structure tensor, the eigenvalues of the Hessian matrix yield information about local properties. Second-order derivatives relate to the local curvature.

The eigenvalues of the aforementioned matrices are rotationally invariant. In order to be able to account for oriented structures, the raw entries of the symmetric matrices are maintained. For example the local orientation vector \mathbf{o} in a two-dimensional image can be determined by using only the entries of the structure tensor:

$$\mathbf{o} = \begin{bmatrix} \mathbf{J}_{22} - \mathbf{J}_{11} \\ 2\mathbf{J}_{12} \end{bmatrix}$$

Note that, only additions and constant multiplications are necessary in this case.

- **Higher order moments of local neighborhood:** The first few moments have an intuitive interpretation: The second order moment equals the variance, the third and fourth moments define the skewness and kurtosis. The n th-order moment about the mean is given by

$$\mu_n = \sum_{i=0}^n \binom{n}{i} (-m)^{n-i} \mu'_i,$$

where the mean m of the local neighborhood is implemented by convolving the image with an averaging mask, and the moment about the origin μ'_i is computed by convolving the image raised to the power of i with the averaging mask. Moments up to order four are included in the feature vector.

- **Local entropy and range:** The entropy quantifies the information contained in the neighborhood in the sense of an expected value and can be used to characterize the texture. It is defined as

$$-1 \sum_i p_i \log_2 p_i,$$

where p is a vector containing the normalized histogram counts of the local neighborhood. Moreover, the difference between the maximum- and the minimum value in the neighborhood are computed.

- **Morphological opening and closing:** The results of morphological opening- and closing operations [47] with differently sized structure elements can be computed as well, yielding a morphological spectrum [63].
- **Rank-order filter:** Another class of operations is defined by rank-order filters. These filters map a discrete set of voxels onto themselves by sorting and comparing the intensity values of neighboring voxels. Famous examples are the median-, maximum- and minimum filters, but any percentile of the population can be taken for separation (i.e. the lower and upper quantiles).

Supplementary to the above-mentioned features, some basic operations, such as the convolution with an averaging or Gaussian mask and the gradient magnitudes, are computed as additional features. Of course these features depend on the size of the neighborhood, which can also be varied. In most cases, a Gaussian mask that corresponds to a weighted window function is taken in the convolution. Naturally, an *optimal* size of the bandwidth used depends to a certain degree on the size and shape of the object of interest. If now prior knowledge about the objects' size is available, these features are computed on multiple scales. Hence, the choice of the appropriate scale is left up to the classifier.

2.2. Dimension Reduction

All classifiers inevitably suffer from the *curse of dimensionality* [44]. In short, the limited amount of data samples one cannot sufficiently populate high-dimensional spaces. The volume of the space grows exponentially with the dimension. However, the amount of data is doomed to limited over time. Therefore various methods have been proposed in literature to effectively reduce the dimensionality of the input space for machine learning methods. A reduction can be achieved by two basic approaches. Feature selection (see Section 2.3.3) aims to reduce dimensionality by selecting important variables, leaving the data unchanged. Methods related to data compression intend to transform the data in such a way so that redundancy is minimized. A prominent example of the latter approach is principal component analysis.

2.2.1. Principal Component Analysis

Principal component analysis (PCA) can be defined as the orthogonal projection of the data onto a lower dimensional space, called the principal subspace [7]. The projection is done by finding a rotated orthogonal coordinate system, such that the variance of the data is maximized. The principal subspace is spanned by the ordered principal components, which gradually account for as much variance of the data as possible. The first component corresponds to the maximal variance, while the last principal components can be discarded without much loss of information. This fact makes PCA a widely used dimension-reduction technique in explorative data analysis [103].

The computation of the principal components of a given $n \times m$ data matrix \mathbf{X} involve the following steps:

1. Subtract the sample mean of the data such that $\mathbf{E}(\mathbf{X}) = 0$
2. Find the eigenvectors and their corresponding eigenvalues of the covariance matrix of $\mathbf{C} = \mathbf{X}^T \mathbf{X}$
3. Rearrange the eigenvectors by sorting them according to their descending eigenvalues
4. Select a subset of the of the eigenvectors \mathbf{W}_q by choosing the first q sorted eigenvectors, corresponding to the q largest eigenvalues
5. Project the data points onto the new basis by $\mathbf{Y} = \mathbf{W}^T \mathbf{X}$

For a derivation of this algorithm, the reader is referred to [7]. One major drawback of PCA is that it is an unsupervised procedure, since it makes no use of the class label of the data points. This fact can negatively affect classification performance when the variance of the labels does not coincide with the variance of the data. Supervised methods for decorrelation, such as *partial least squares (PLS)* overcome this problem by incorporating the label information.

The data normalization technique called *Whitening* is very similar to PCA. It also finds an optimal coordinate transformation, but without projecting on a principal subspace.

2.3. Supervised Learning Methods

In machine learning, supervised learning refers to the process of *deducing* a function from labeled training data. Supervised learning methods are input as a pair, consisting of a data sample and its corresponding class label. The *learned* function (classifiers) is then able to infer the class of new unseen data points autonomously. Important characteristics of classifiers is their *bias* and *variance*.

Variance refers to the stability of the prediction made by the classifier. It requires large values if different training sets (drawn from the same underlying distribution) also result in different classification decisions. Variance is low if a particular realization of training data only has a minor effect on the classifiers. In short, the variance of a classifier characterizes its inconsistency and does not indicate whether its decisions are correct or incorrect.

The bias of a classifier describes the flexibility of its subjacent model. The lower the bias, the more expressive is the classifier in terms of complex decision making. Together with high variance, a low bias typically leads to *overfitting* to training data. Classifiers with a high bias reveal a more rigid behavior and are “biased” by the imposed restrictions from the model (e. g. decision function must be linear).

A good classifications system is a system that has found the best trade-off between bias and variance in terms of generalization ability: i.e. true accuracy achieved on unseen data. Bias and variance are not utterly disjunctive and the values are interlinked for each particular classifier. For many classifiers, tuning parameters that to some extent control the amount of bias with regard to variance do exist. In the following the most important classification methodologies used in this thesis are explained.

2.3.1. Bootstrap Aggregation

Bootstrapping is commonly used to assess the accuracy of a classifier or parameter setting (see Section 2.4). A bootstrap sample B of data points $\mathbf{x}_1, \dots, \mathbf{x}_n$ refers to the process of drawing n -times a sample with replacement. The probability that a data point \mathbf{x}_j is in the bootstrap sample B is given by:

$$P(x \in B) = 1 - \left(1 - \frac{1}{n}\right)^n \approx 0.632, \text{ for a reasonably large } n$$

Hence, a bootstrap sample contains about 63% of the data items (some are drawn multiple times). The main idea behind bootstrap aggregation or *bagging* is to train a classifier multiple times on each new bootstrap sample and to average the outputs obtained afterwards. Especially for high-variance and low-bias classifiers, bagging works well [44]. If the individual classifiers based on the bootstrap sample are generated independently, averaging results leaves the bias unchanged while drastically reducing the ensemble variance.

An ideal candidate for bagging are *unpruned* decision trees (see Section 2.3.2). Since they are able to capture complex decision boundaries, fully grown trees have a relatively low bias. On the other hand, trees can greatly profit from variance reduction due to their inherently instable nature.

2.3.2. Classification and Regression Trees

A widely used technique to partition the feature space into hyper rectangles is called *decision trees*. A famous representative among different variants are *Classification and Regression Trees (CARTs)* introduced by Breiman [13]. In Figure 2.1, the partition of a two-dimensional feature space is illustrated. This partition is generated by a binary tree, where each node corresponds to a decision, or a partition, in the feature space. For instance the root node of the tree in Figure 2.1 divides the whole feature space according to whether $x_1 \geq \Theta_1$ or $x_1 < \Theta_1$. This process is recursively applied to all daughter nodes. Each leaf node of the tree then corresponds to a particular hyper rectangle that is localized in the feature space. The set of $\Theta_1, \dots, \Theta_4$ and their corresponding feature axis (variable) are the parameters of the decision tree model. A new observation from the feature space can be mapped to a partition by simple traversing down a path in the tree following each node's decision. In each leaf node, which corresponds to a region in the feature space, a separate model is applied to predict the target value. In classification this refers to the assignment of a leaf partition to a specific class.

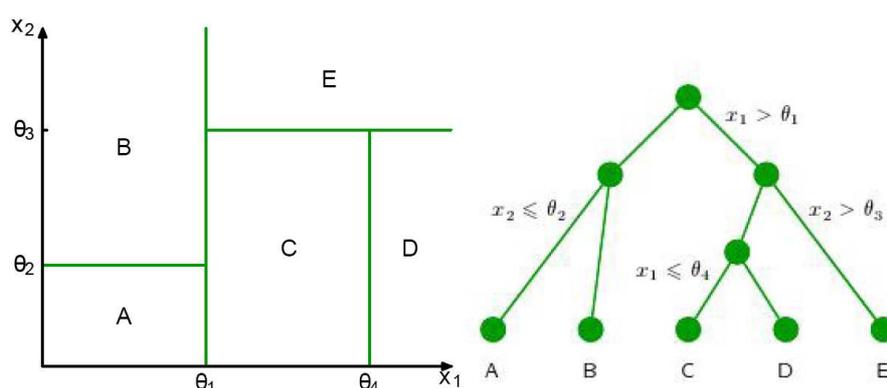


Fig. 2.1.: Illustration of the feature space partitioning with classification and regression trees (CARTs). Picture taken from [7]

Learning a decision tree consists of determining the trees' structure from training examples. The variables involved in splitting and their thresholds were assigned for each node. Finding the optimal tree structure that comprises all combinations of choices for variables and their corresponding thresholds is generally infeasible due to the large number of possible solutions. Thus, a greedy optimization scheme is applied for the tree construction. Starting at the root node, optimal split criteria are computed recursively. In each node all variables m are examined. The best split threshold Θ – corresponding to an impurity measure Q_m – is computed separately. The variable that achieves the best split according to the measure is selected. Different impurity measures were proposed [13] to assess a node's impurity Q_m , including:

- Gini index:
$$\sum_{k=1}^K p_{mk}(1 - p_{mk})$$
- Cross-entropy:
$$-\sum_{k=1}^K p_{mk} \log p_{mk},$$

where K is the number of classes and p_{mk} is the fraction of training samples in node m having class k . In this thesis the Gini index is used. In general the cross-entropy criterion achieves very similar results [87].

The generation of the decision tree is stopped when all leaf nodes are pure (or purity is smaller than a predefined constant), i.e. only samples from one class fall into to the corresponding hyper rectangle. However, a large tree where all leaf nodes are pure might overfit the data, hence pruning strategies have been proposed to prune the tree in a bottom-up manner. In the following, decision trees are used in the context of random forests, where each single tree is fully grown. Therefore, a description of pruning strategies [44] is of minor relevance and has been omitted.

2.3.3. Random Forests

Random forest is a procedure that grows an ensemble of decision trees and collects their votes for the most popular class, injecting several moments of randomness along the way. Random forests were introduced in Breiman [14]. Tree-based classifiers partition the feature space into a set of hyper rectangles. In particular, classification and regression trees, called CARTs [13], use a recursive binary splitting. These tree-based methods and the determination of the best split point for a feature variable are described in Hastie et al. [44, 7].

Training the classifier defines the decision boundary within this feature space. Each new sample can be classified according to its position relative to the decision boundary. Since the classifier used is an ensemble of single decision tree classifiers, the classification step yields a probability map $\mathbf{P}(\mathbf{x})$ of the same size as the data volume for each given class, which equals the arithmetic mean of the outcomes of the N_T single base-classifiers T_i where $i = 1, \dots, N_T$. The probability of sample \mathbf{x} for class c is then computed by:

$$\mathbf{P}_c(\mathbf{x}) = \frac{1}{N_T} \sum_{i=1}^{N_T} I(T_i(\mathbf{x}) = c)$$

where $I(\cdot)$ is the indicator function.

Random forests have been demonstrated to have excellent performance in [14], run efficiently and quickly on large data sets, and are able to handle high numbers of input variables. There are two main input parameters to be set: The number of trees N_T to

grow and the number of variables that is chosen randomly at each node split. Following the observations of Breiman [14], the implementation used here sets the second parameter to $\sqrt{N_T}$. The default value for the number of trees is set to $N_T = 100$. The Vigna¹ implementation of random forests is used. Note that, the higher number of trees does not necessarily imply overfitting. The number of trees used here is a trade-off between run time and classification accuracy. The random forest algorithm is summarized in Algorithm 1.

The random forest classifier has been shown superior performance as compared to standard boosting techniques such as AdaBoost [44] and achieves comparable results to support vector machines depending on the application [101, 88]. Since random forests are easy to train and no hyper parameter has to be tuned, they are a reasonable choice for interactive learning.

Algorithm 1 Random Forest for classification

Set N_T : the number of trees
 Let n be the total number of samples
 Let p be the total number of variables
for $i = 1$ to N_T **do**
 Draw a bootstrap sample B_i from the training data
 repeat
 Grow a decision tree T_i on the training data in B_i recursively by:
 Selecting $m_{\text{try}} < p$ variables randomly
 Computing the best split among the m_{try} variables according to the split criterion
 Splitting the node and propagate the examples
 until All terminal nodes are pure
end for

To predict a the probability of a new point \mathbf{x}_j having class c compute:

$$\frac{1}{N_T} \sum_{i=1}^{N_T} I(T_i(\mathbf{x}_j) = c)$$

where $I(\cdot)$ is the indicator function.

¹The Vigna Computer Vision Library

Variable Importance

Not all features of the generic feature set contain enough predictive information to justify their calculation. A huge amount of memory and computation time can be saved when we restrict the feature set to those features that are essential for the classification.

The out-of-bag samples can be used to estimate the variable importance, which measures the role of a feature dimension in discriminating the classes. The estimation can be performed during the training of the classifier and introduces very little additional computational cost. This is done by evaluating the prediction accuracy of the out-of-bag cases, permuting the values of a variable m , the importance of which should be measured, and reevaluating the prediction accuracy with the disturbed values. Under the assumption that a variable is not important if disturbing it does not increase the misclassification rate, the difference gives an importance score for the variable: Subtract the number of votes for the correct class in the permuted out-of-bag data from the number of votes for the correct class in the untouched out-of-bag data. The average of this number over all trees in the forest is the raw importance score for variable m . The described estimation is called permutation importance, because the disturbance of variables is carried out by randomly permuting all out-of-bag cases.

There is another way to quickly assess variable importance: Every time a split of a node is made on variable x_i , the Gini impurity criterion (see decision trees in Section 2.3.2) for the two descendant nodes is less than the parent node. Averaging the amount of Gini decrease for each individual variable over all trees in the forest gives a fast variable importance, because a high decrease of the impurity implies high predictive information. Very often results are consistent with the permutation importance measure [14].

2.3.4. Linear Support Vector Machines

Given the training data as a set of features and assigned class labels:

$$\mathbf{D} = \{(\mathbf{x}_i, c_i) \mid \mathbf{x}_i \in \mathbf{R}^p, c_i \in \{-1, 1\}\} \quad \text{for } i = 1, \dots, n \quad (2.3.1)$$

where the label c_i indicates the class error or normal to which the example belongs. Each \mathbf{x} is a p -dimensional feature vector extracted from the original frame. The aim is to find the maximum-margin hyperplane that separates the points that have $c_i = 1$ from those that have $c_i = -1$. A hyperplane can be written as the set of points \mathbf{x} satisfying:

$$\langle \mathbf{w}, \mathbf{x} \rangle - b = 0 \quad (2.3.2)$$

The vector \mathbf{w} is a normal vector of the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector \mathbf{w} . Now, \mathbf{w} and b must be chosen so as to maximize the margin, that is the distance between the parallel

hyperplanes. These hyperplanes must be as far apart as possible, while still being able to distinguish between data points. These hyperplanes can be described by the equations:

$$\begin{aligned}\langle \mathbf{w}, \mathbf{x} \rangle - b &= 1 \\ \langle \mathbf{w}, \mathbf{x} \rangle - b &= -1\end{aligned}\tag{2.3.3}$$

Note that if the training data is linearly separable, two hyperplanes can be selected in a way that there are no points between them and then try to maximize their distance. Using geometry, we find that the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$, thus we want to minimize $\|\mathbf{w}\|$. In order to prevent data points from falling into the margin, we add the following constraint:

$$c_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b) \geq 1, \quad \text{for all } 1, \dots, n\tag{2.3.4}$$

Substituting $\|\mathbf{w}\|$ with $\frac{1}{2}\|\mathbf{w}\|^2$ leads to a quadratic optimization problem which can be efficiently solved using Lagrange multipliers. The software package LIBSVM [19], which we used for the experiments, implements the *soft margin* extension of support vector machines. If the data is not linearly separable, the Soft Margin method chooses a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance between the nearest cleanly split examples. The method introduces slack variables, which assign a misclassification cost of the data items.

2.3.5. Kernel Trick

Linear support vector machines separate the data points with a linear decision function. With the so called *kernel trick*, this procedure can be extended to non-linear classification problems. A remarkable observation in the formulation of the linear support vector machine is that the data samples \mathbf{x}_i appear only in scalar products of the form $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Therefore, it is possible to substitute the scalar product of the original input space with $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, where Φ is a non-linear mapping of the input space to a higher dimensional space. The use of positive definite kernel functions:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$$

which implicitly define a scalar product in the mapped space enables efficient computation. A typical choice for such a kernel function is the RBF-Kernel:

$$k_\sigma(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i, \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

The parameter σ controls the size of the Gaussian kernel and therefore the smoothness of the decision boundary.

2.4. Evaluation Methods

The generalization performance of a learning method, i.e. the predictive power of a learned model on unseen data, is a matter of great concern. In this chapter the fundamental performance measures are discussed in brief.

2.4.1. Cross-Validation

Perhaps the most widely used method for estimation the prediction error is cross-validation [44]. This method uses some parts of the data to train the classifier and other parts for testing. In K -fold cross-validation, the data is split up into K (roughly equally sized) chunks. Each of the K parts is chosen once as test data while the classifier is taught on the remaining $K - 1$ parts (training data). This results in K prediction estimates, which are then averaged. Overall, using ten folds ($K = 10$) is well accepted choice in literature [44]. In stratified K -fold cross-validation, the folds are selected so that each fold contains roughly the same proportion of classes. This is in particular important for highly unbalanced data.

When the number of folds is equal to the number of data points ($K = n$), K -fold cross-validation is called *leave-one-out* cross-validation.

When dealing with sets of images or volumes, the so called *leave-one-image-out* cross-validation is applied, where performance is assessed on each image while training on the remaining ones.

2.4.2. Receiver Operator Characteristics Curve

To assess the classifier performance regarding which type of errors are made, the confusion matrix (see Table 2.1) is widely used. From this representation it is easy to calculate the accuracy of a classifier by:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

For unbalanced data the accuracy is not an appropriate measure, since the classifier could achieve a high accuracy simply by predicting the majority class for each and every data sample. Other measures can aid the evaluation of classifiers by looking at the

false positive and false negative rates:

$$\begin{aligned} \text{TP rate (recall)} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{FP rate} &= \frac{\text{FP}}{\text{FP} + \text{TP}} \\ \text{FN rate} &= \frac{\text{FN}}{\text{FN} + \text{TN}} \end{aligned}$$

If the classifier outputs a probability value, the trade-off between recall and the false positive rate can be visualized by the *receiver operator characteristics* (ROC), see [29] for an introduction. There, the false positive rate is plotted against the recall for varying output thresholds. The best possible prediction method yields a point in the upper left corner of the ROC space, accounting for a full recall with no false alarms. A classifier that assigns class labels randomly would result in a diagonal line. The area under curve A_c is a popular single-valued measure to compare classifiers.

Prediction/Truth	positive	negative
positive	TP	FP
negative	FN	TN

Table 2.1.: Confusion matrix

Part II.

**Interactive Framework and
Applications**

3. The Interactive Framework and its Properties

3.1. General Framework

The great variety of biological and medical imagery gives rise to problem-specific algorithm design. As a first step, segmentation (the partitioning of a digital image into multiple regions) is certainly one of the most crucial tasks. The quality of further analysis is highly dependent on the accuracy and relevance of the segmentation results. Fully automated techniques can usually handle limited variability in data appearance, while generic approaches attempt to facilitate any kind of image data, which is still an unsolved problem. Semi-automatic segmentation techniques do not commit to a particular type of image and try to incorporate the genericity property by allowing user interaction.

Additionally, the identification of the different types of objects is of vital importance for many problems. Hence, the generic formulation of segmentation and classification in a unified manner can help address many important standard problems in the field of biomedical imaging.

Similarly, a biomedical expert is enabled to label objects of interest. There are in general no restrictions to the number of object classes; this choice depends solely on the specific question posed to the image data. Given these labels provided by the expert, supervised learning is used to train a classifier. To that end, a set of generic image features is computed that are able to capture diverse, local image characteristics. This ensures the transferability of our approach to a variety of applications in biomedical image analysis. To demonstrate the performance of the proposed methodology, experimental results on two biomedical problems are presented. Yet, neither problem-specific preprocessing nor special-purpose features are needed to achieve reliable results.

The contributions of this work are the following:

- The use of generic image features which are not dedicated to a specific problem. The features are designed to represent diverse, local image characteristics at different scales (see Section 3.1.1);
- Coping with a few user labels, which are inputted via a paint brush interface. Only these partial user labels are used to train the system;

- Supervised learning on a pixel-level with a state-of-the-art ensemble classifier (random forest, see Section 2.3.3), which is able to address the multi-class problem without the need to optimize any hyper-parameters;
- Results and ROC analysis of our method on public biomedical data sets (illustrated in Section 3.3), that corroborates its performance and transferability with other segmentation and classification tasks (see Section 3.4);
- The implementation of an easy-to-use tool that bundles all the processing steps in a single piece of software (see Section 3.2 and Figure 3.1);
- The use of this framework to various applications in the field of life sciences and industrial image processing (see Chapters 4-7).

It is important to note that the proposed method is limited by heavy overlap between objects and cannot solve tasks that require a higher-level semantic understanding of the image. However, it is a useful tool that facilitates further post-processing of more difficult data.

The user can label regions of the input image as instances of meaningful objects (e.g. background, object-type one, ...). Secondly, for each labeled pixel, a set of generic features is computed. The pairs of user-provided label and computed features for each pixel are then used to train a classifier in a supervised fashion. In the prediction step, the classifier assigns a *soft* label to every pixel in the image between 0 and 1. This label stands for the probability of the pixel being of either object class. The resulting image of soft-labeled pixels is called a probability map. The final segmentation operation is performed on the *probability map* rather than on the original image. It is important to emphasize that there is no hard assignment of user labels.

The combination of discriminative local features and effective classifiers is also applicable to a wide range of standard settings in life sciences and industrial image processing. The interactive tool demonstrates the excellent run-time performance of the underlying algorithms. Thus the concepts described in this thesis were deployed to collaboration partners in bundled software. Hence, a variety of applications could be assessed as described in Chapters 4-7.

3.1.1. Image Features

In addition to the original pixel gray values of the image, the following list of generic image descriptors at two different scales were used:

- 1 Morphological closing
- 2 Morphological opening
- 3 Local entropy

4	Local range
5	Gradient magnitude
6	Gaussian filter
7-8	Sorted eigenvalues of Hessian
9-12	Local moments up to the 4th order
13-14	Sorted eigenvalues of the structure tensor

For the an explanation of these features the reader is referred to Section 2.1. In Figure 7.3 several feature responses are displayed. In our experiments the above features were computed using two neighborhood sizes of 5x5 and 11x11, resulting in 29 feature per pixel. This feature set compromises statistical measures, linear and non-linear filter responses. It is able to express diverse local image characteristics depending on the choice of the fixed scales. In contrast to the work [5] the subsequent classification is not restricted to one particular scale.

3.1.2. Categorization of Features

In the interactive learning interface, the user can choose between different previously defined feature sets. Features accounting for a specific type of structure are grouped together:

- **Color and intensity** consists of the raw intensity value of the smoothed image 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 local range.
- **Texture** comprises the structure tensor, Hessian matrix (and its eigenvalues), morphological operations, entropy, higher order moments and local quantiles.

All these groups can be selected on five scales, with a total of 15 different choices. The user is free to combine these selections in order to generate an appropriate feature set for the problem at hand.

3.1.3. Pixel-Wise Classification

As mentioned before, classification is performed using the random forest classifier introduced by Breiman [14] (see Section 2.3.3). Random forests consist of many decision trees. The individual trees are not pruned in the training phase and are built under random influence. During prediction each tree classifies the new example. The ratio of

these votes is interpreted as a posterior probability and provides the basis of the segmentation step. 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 set of use cases. For our experiments, a random forest of $N_T = 100$ decision trees is trained.

3.1.4. Feature Selection

Feature selection refers to the process of selecting a subset of relevant features from entire set. Indeed, feature selection reduces the dimensionality of the classification problem, thus reducing the negative effects of high-dimensional problems. 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. For the two data sets evaluated in this Chapter feature selection is not applied. An application, where feature selection is successfully used together with this framework, can be found in in Chapter 5.

3.1.5. Application-Dependent Quantification

Different segmentation problems require also different outputs. In cell segmentation, for example, one is interested in reliable and accurate cell counts, whereas in other domains pixel-precise tracing is required. The framework supports different types of application-dependent post-processing. The seeded watershed transform operating on probability maps, for instance, is used for cell segmentation. Locally dominant thresholding (i.e. smoothing of the probability maps with an appropriate kernel before hard class assignment) can also be applied in a multi-class sense. In Chapter 5 it is illustrated how to combine globally optimal shortest path algorithms with the learning framework.

3.1.6. Interactive Classification

During labeling the user can enable the *interactive mode*. This means that the classifier is updated each time the user inputs new labels; and current image is then predicted. The results are transparently overlayed onto the raw image. This concept guides the user to places of the image, where the classification is incorrect and enables the user to refine the output by given new labels there. Shortly afterwards, the user will see the effect of his/her refinement by the updated prediction overlay. The *margin* [14] of the output of the classifier can also be displayed interactively. The margin relates to the uncertainty of the prediction and yields high values near decision boundaries and in places of the feature space, which have not been explored by labeling so far. Providing labels in regions of high uncertainty accelerates the whole classification process by promoting steep learning curves.

The classifier update is achieved using a ring buffer that is implemented via a *double-ended priority queue*. The buffer is filled with random forest classifiers comprising 10 decision trees. The capacity of the ring buffer is 10 resulting in an ensemble of 100 trees once the buffer is full. As new classifiers arrive the *oldest* trees according to a priority measure drop out the queue and are replaced. Currently the *first in - first out* principle is used promoting the classifiers trained on the latest user labels. Each classifier is trained in a new *thread*, thus exploiting the parallel computing capability of modern multiprocessor computers.

3.2. The Graphical User Interface

The graphical user interface was initially implemented in Matlab using the underlying algorithms written in C++ [62]. This rapid prototyping language is rather limited to handle more advanced graphical effects (e. g. transparent overlays, 3D) and multiple threads, which is indispensable for concurrent processing. Hence, this prototype was ported and extended to Python/Qt; still interfacing the C++ algorithms. Python was chosen due to its excellent support of various libraries (e. g. boost, OpenMPI, HDF5, Qt, etc.) and its elaborate object model. The graphical user interface is shown in Figure 3.1 and a binary version compiled on the 9th of March 2010 can be downloaded at: <http://hci.iwr.uni-heidelberg.de/software.php>. The initial Matlab prototype is also available¹. The user interface written in Qt is modularized using the widget concept in order to provide reusable user interface units. Code was developed simultaneously on windows 32-bit and linux 64-bit to ensure portability right from the beginning. In order to interface to the Vigna C++ library appropriate bindings, documentation, unit tests and module layout were also developed² and included in Vigna.

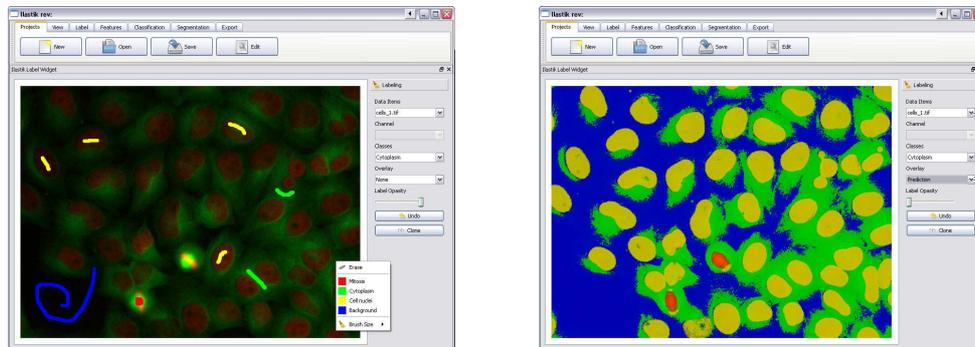
In the following, the graphical user interface guides the user and calls C++ (random forest) routines and displays the results. In the tool the user can refine his/her labels after seeing the segmentation result. In the experiments, however, the labels were not refined. The labeling and feature computation were carried out on a conventional notebook.

3.3. Evaluation of the Framework

To evaluate the basic framework two data sets with available ground truth were selected. In the former, the aim was to segment low-contrast blood vessels in retinal RGB images. In the latter, the task was to segment cell nuclei in fluorescence images and to classify their phenotype into mitotic or normal.

¹at <http://hci.iwr.uni-heidelberg.de/people/csommer/clasi/>

²together with Ullrich Köthe, Nathan Hüsken



(a) Labels are provided by the paint interface. Classes and brush size are selected in a context menu.

(b) Interactive mode: real-time predictions are overlaid

Fig. 3.1.: The Ilastik GUI: Biomedical use case of the implemented user interface. Left: A domain expert labels background, cell nuclei, cytoplasm and a phenotype of interest (red label color). Right: After selecting color features and the interactive classification, a real-time update of the current classifiers output is overlaid. The interactively trained classifier can then be applied to more images.

3.3.1. Retinal Imaging

A common procedure in the examination of the human eye is retinal imaging. An optical camera is used to see through the pupil of the eye to the rear inner surface of the eyeball. The recorded picture shows the optic nerve, fovea, surrounding vessels and the retinal layer. The medical doctor can then reference this image in the analysis of any observed findings. A still active research area is the segmentation and localization of blood vessels in retinal images.

Here, our method is tested on the publicly available database STARE (Hoover et al. 2000 [46]), which consists of 20 retinal images captured by a TopCon TRV-50 fundus camera at 35° FOV. The images were digitized to 700 x 605 pixels, eight bits per color channel. Ten of the images contain pathology and the other ten are normal. Two observers manually segmented all images. The first observer segmented 10.4% of pixels as vessel, against 14.9% vessels for the second observer. The segmentations of the two observers are fairly different in that the second observer segmented much more of the thinner vessels than the first one. Performance is computed by taking the segmentations of the first observer as ground truth. For the sake of comparability, the evaluation procedure of Soares et al. [99] is adopted which includes leave-one-image-out cross-validation and ROC-curve analysis (see Section 3.4).

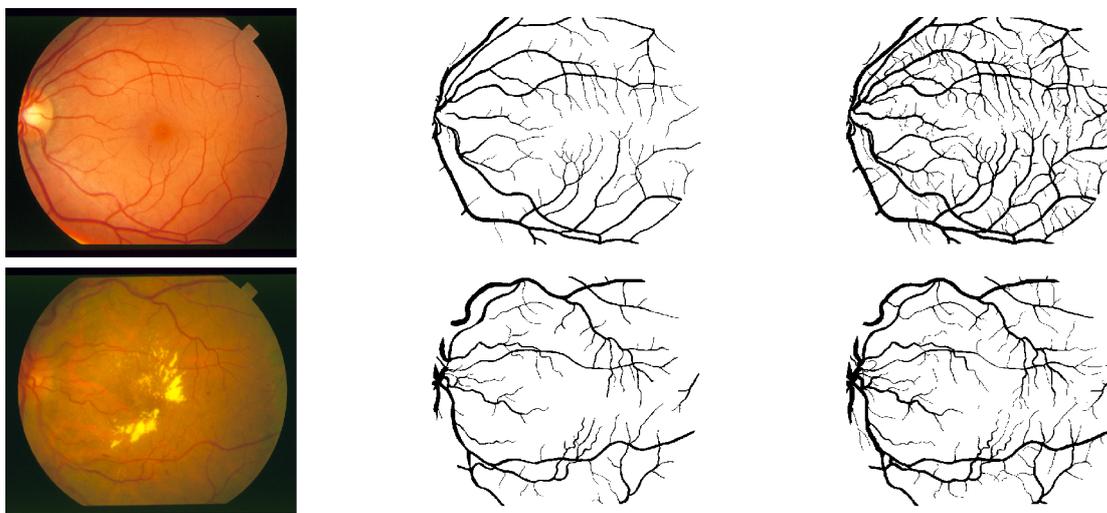


Fig. 3.2.: The STARE database. Rows: Two examples. From left to right: Original RGB image, ground truth given by first observer, ground truth given by second observer.

3.3.2. Fluorescence Wide Field Imaging

Identification and segmentation of cells and their phenotype is a standard task in the analysis of fluorescence imagery. 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.

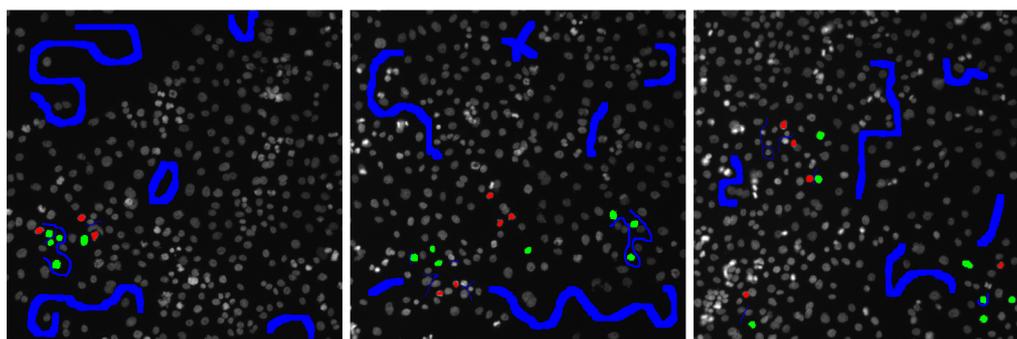


Fig. 3.3.: Cell nuclei of normal and mitotic cell nuclei and overlaid user labels. Blue: Background; Red: mitotic cell labels; Green: normal cell nuclei

The *Human HT29 Colon Cancer 1* image set available from the Broad Bioimage Benchmark Collection was used. The image set consists of 6 images. The samples were stained with Hoechst 33342, pH3 and phalloidin. Hoechst 33342 is a DNA stain that labels the nucleus. Only the images from the Hoechst 33342 channel were included, as

the other channels were not useful for counting cells. The cells in all six images were treated with an RNA interference reagent in the course of a high-throughput screen; see Moffat et al. [78] for detail. The images were acquired at the Whitehead-MIT Bioimaging Center on a Cellomics ArrayScan. The images provided here show a single-channel DNA. Image size is 512 x 512 pixels. Images are available in 8-bit TIFF format.

3.4. Experimental Evaluation

3.4.1. STARE Database

In contrast to other supervised methods, the ground truth was not used for training the classifier in the cross-validation experiments. Instead, only a few labels given by one user were used (see Figure 3.4). Performance, however, was computed against the ground truth segmentation. A user labeled the 20 images from the STARE database in approx 1.5h (including loading and saving of the images), resulting in an average of 0.84% of foreground and 11% background coverage.

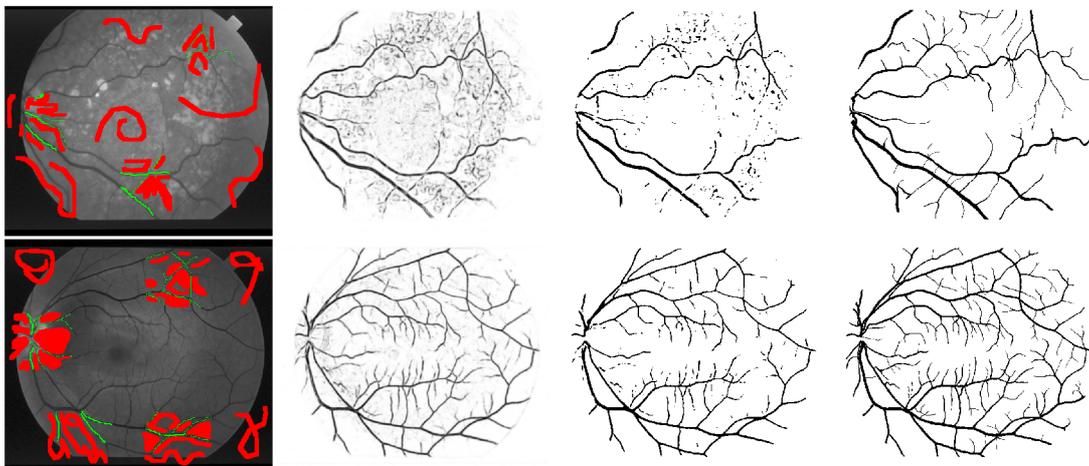


Fig. 3.4.: Segmentation results after leave-one-out cross-validation. First row: pathological case, second row: normal case. From left to right: Raw image with overlaid user labels (red: non-vessel/ green: vessel), resulting probability map after leave-one-out cross-validation (training with user labels of the remaining 19 images), segmentation at threshold level $t = 0.5$, ground truth given by first observer.

Every image was segmented using the user labels from the other 19 images from the training set. The segmentation was produced by thresholding the resulting probability map. Before thresholding, the probability map was smoothed with a Gaussian filter

($\sigma = 1$), to reduce spurious noise pixels. The results were compared to the matched filter results from Chaudhuri et al. [20], 2-D Gabor wavelets from Soares et al. [99], adaptive local thresholding scheme from Jiang et al. [49] and the ridge-based segmentation proposed by Staal et al. [100]. Performance was measured using ROC curves [29]. ROC curves are plots of true positive rates versus false positive rates for varying thresholds on posterior probabilities. The true positive rate is computed by dividing the number of true positives by the total number of positive (vessel) pixels in the ground truth segmentation, while the false positive rate is the ratio between the false positives and the total number of non-vessel pixels in the ground truth. Also the areas A_z under the ROC curves and accuracies of the methods (see Figure 3.5) are presented. The closer the ROC curve gets to the top left corner, the better the performance of the method, leading to $A_z = 1$, an area of which signifies perfect agreement with the ground truth.

Method	A_z	Accuracy
Soares et al. ^a	0.9671	0.9480
Chaudhuri et al.	0.8987	—
Jiang et al.	0.9298	0.9009
Staal et al.	0.9641	0.9516
our results	0.9448	0.9593 ^b

^a(GMM $k = 20$)

^bAt threshold level 0.455

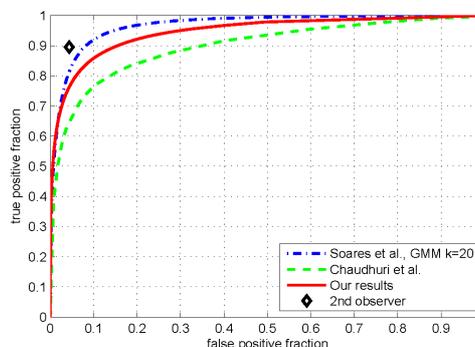


Fig. 3.5.: ROC analysis results after leave-one-out cross-validation

3.4.2. Human HT29 Colon Cancer 1

The task of this data set is to segment and count the cell nuclei. Some of the cells are mitotic and appear slightly brighter. In the original experiment a chemical mitosis dye was used to detect mitotic cells. This data set is used to demonstrate the multi-class ability of our approach. The background and the two different 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) coverage. Once again leave-one-out cross-validation is performed. The probability maps from the supervised classification were fed into a modified marker-based watershed transform [6]. 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. In Figure 3.6 segmentation results for the two different cell phenotypes are shown.

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

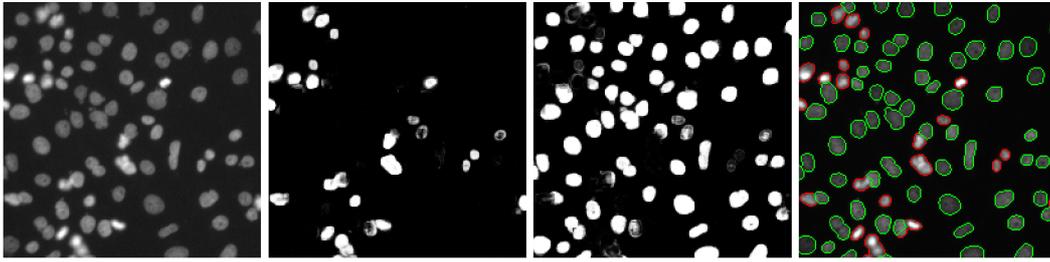


Fig. 3.6.: Images from the Human HT29 colon cancer data set

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.

3.5. Discussion

3.5.1. Related work

Most of the related approaches evolved in the field of digital photography with the aim to crop foreground objects out of background clutter. Boykov et al. and Rother et al. [10, 89] proposed the use of iterated Graph Cuts to solve the segmentation task on one specific image utilizing user mouse inputs. While there, the single side potentials of the graph model are based on histograms with respect to the Gaussian mixture models in color space, Lefevre [66] proposed the use of the k-nearest neighbors classifier for pixel classification. Furthermore, he extended the marker-based watershed transform to operate on a learned gradient map seeded by the given user brush strokes. Other methods try to model the object boundaries such as intelligent scissors, which [80] are widely used in biomedical applications, where the user just needs to roughly delineate the object boundary.

Implemented with an intuitive interface, the algorithm provides the user with a segmentation tool of satisfactory accuracy. The system can be trained swiftly with an labeling effort ranging from 20 seconds to several minutes per image depending on problems' complexity. Once the classifier is trained on a set of images, it is able to process more images autonomously. Using the supervised learning concept as an initial step for the segmentation renders this method highly adaptive for tackling a wider range of related problems in biomedical image analysis.

3.5.2. Conclusion

In this chapter it was illustrated how to tackle standard image processing tasks in the field of biomedics without resorting to programming expertise. Our approach combines a convenient user interface with a state-of-the-art classifier. The approach is robust since the user only needs to enter a few labels and because the classifier is trained on a set of generic features, which work across many types of images. The segmentation framework was extended in a multi-class sense to allow for different object types and problems. The method was tested on two public data sets, which demonstrated the good transferability of our approach. On the STARE database it was shown that the ad-hoc performance of our proposed framework was in the range of previously published methods [49, 100, 20, 92] which, however, explicitly take advantage of domain knowledge and were perhaps developed over months dedicated to that specific problem.

However, this proposed framework is limited to local brightness, color and texture and is not designed to capture global requirements. Indeed it can easily be utilized as a first stage tool to solve advanced problems on top of its output. A very interesting point in our experiments is that the few user labels are more informative than the abundant labels from the ground truth, which leads to superior classification performance. This raises the question of how the classifier can actively propose regions in which the system needs more labels from the user. One idea is to use the out-of-bag samples in the training phase of the random forest classifier to compute a *certainty map*. This enables the system to ask for labels that it deems informative.

4. Adult Neurogenesis

4.1. Introduction

Analysis of 3D confocal-microscopy images is an essential yet very challenging problem in many biological studies where quantitative or high-throughput analysis is needed. The difficulties stem from the variability and complexity of the images, as well as from artifacts due to staining and imaging. Here, the interactive framework introduced before is applied. It builds upon discriminant local feature descriptors together with non-linear classifiers. Again, this approach utilizes the interactive software that allows the domain expert to annotate a representative subset of the data in order to train a segmentation system that can then process hundreds of data items in an automated fashion. Based on these results, relative geometric measures and cell characteristics are extracted that quantify neurogenetic properties of the underlying experiments. Experimental results verify the performance and reliability of the proposed approach. Additionally, to cope with the large amount of data, a parallel pipeline is implemented on high-performance computers, which dramatically reduces computation time.

4.1.1. Adult Neurogenesis

Adult neurogenesis is a recent example of a century-old dogma being overturned: The central nervous system was traditionally thought of being fixed or at least very limited in its regenerative powers and repair mechanisms, limited mainly to synaptic reorganization. During the past decade, it has become generally accepted that new, functional neurons are added in distinct regions of the adult mammalian brain. According to Ming and Song [76], the focus of this newly formed research field has now shifted from the mere investigation of existence of neurogenesis to understanding the regulatory mechanisms and the evolutionary benefit. However, the essence of this phenomenon remains elusive in mammals, even today.

The adult neurogenesis originates from neural stem cells that are multipotent, unspecified precursor cells that have the capacity to proliferate and generate the main phenotypes of the nervous system. In the intact mammalian brain, active neurogenesis occurs throughout life in two particular regions of the central nervous system (CNS), the subventricular zone (SVZ) of the lateral ventricle and the subgranular zone (SGZ) of the dentate gyrus in the hippocampus.

The development in the subgranular zone undergoes five different stages: The stem cells located within the SGZ give rise to transient new cells, which differentiate into immature neurons during the second stage; followed by a short distance migration into the so-called granule cell layer in stage 3. Afterwards, these immature neurons start to extend their axonal projections to the pyramidal cell layer and their dendrites in the opposite direction towards the molecular layer. Stage 5 is synaptic integration: The new granule neurons receive input signals from the entorhinal cortex, which plays an important role in memory processes in the brain, and send signals to the pyramidal layer [76, 104, 54].

The mechanisms that are responsible for integrating the newborn neurons into the existing neuronal networks are unknown. Understanding the structural plasticity in the adult CNS is of special interest since this knowledge in particular may provide strategies for integrating transplanted neuronal cells for replacement therapies [76]. What is known, however, is the fact that adult neurogenesis is an extremely dynamic process [31] with many factors that do influence the development at different stages [31].

The biomedical experiments were designed to monitor the influence of several factors on the neurogenic process of different groups of mice. In particular, the impact of specific signaling molecules was investigated: The programmed cell death is usually controlled by signal triggering biomolecules, the so-called death-ligand systems such as CD95L; activation of CD95 via CD95L naturally triggers apoptosis. However, this is not the case in either developing CNS or newly generated adult neural stem cells. Instead the CD95/CD95L system is believed to have a controlling function that regulates the differentiation and branching process of immature neurons.

In order to understand the function of the CD95/CD95L system in the adult CNS, the focus now lies on comparative investigations between wild-type animals and mice with a functionally deficient CD95 receptor. Thus, the aim is to investigate the signaling events downstream of the CD95. One necessary task is to do a quantitative comparison of all the different stages of neurogenic development mentioned above for these two types of mice.

This comparative analysis, however, comprising mainly cell counts and cell position identification, is difficult and exhausting for human experts due to the large amount of data and bad visualization of three-dimensional imagery. Thus, processing of these neurobiological images and the development of adequate semi-automated methods constitutes a central contribution.

4.1.2. Material

The resulting digital multichannel data sets are four-dimensional arrays: A series of three-dimensional volumes where each volume contains the observed emission from one of the separated dyes. These different channels will be referred to as DAPI-, BrdU- and

DCX-Volume and are shown in Figure 4.1. Each of the three images in the bottom line of this figure is one slice of a corresponding 3D-Volume. The data is displayed as gray-scale images to represent the fact that each voxel in a volume only holds one intensity value although, depending on the volume and the corresponding dye, this intensity value naturally corresponds to a specific emission wavelength and thus to a color. The upper image is an overlay of the three channels in the RGB color space. It gives an impression of how the single, separate biological features in each channel are related.

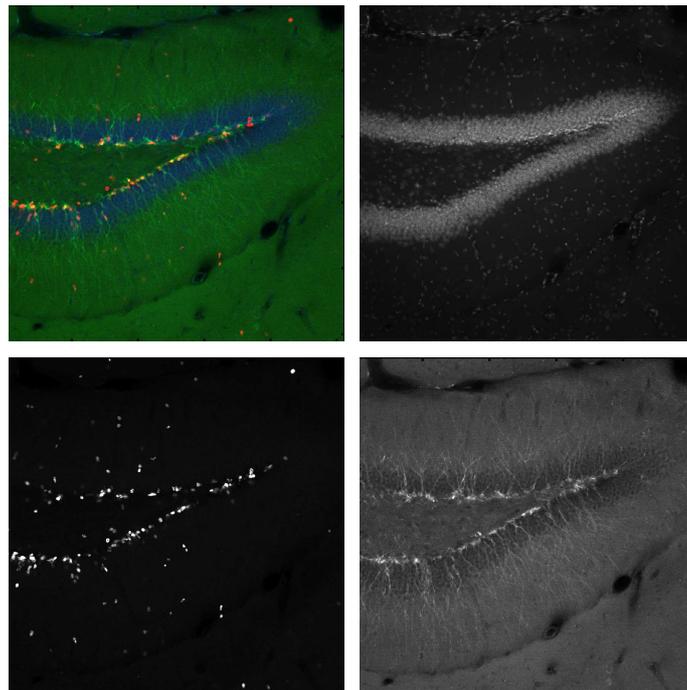


Fig. 4.1.: Clockwise, starting with RGB overlay of the 3 data channels, the DAPI channel with a granule cell layer, BrdU-positive cells and the DCX channel

The number of proliferating cells and their location within the granule cell layer of the dentate gyrus is of particular interest. As described in the earlier section about neurogenesis, the cell position allows conclusions to be drawn about the developmental stage. After proliferation and fate specification, the cells start to migrate through the layer before finally extending their dendrites. The main tasks to be accomplished include the following:

- Segmentation of the proliferating - and thus BrdU-stained - cells
- Segmentation of the DAPI-stained granule cell layer
- Extraction of the relative distance of the proliferating cells to the inner border of

the segmented granule layer (this is done on the basis of the results from the first two steps)

- Classification of proliferating cells that have differentiated into immature neurons, thus both BrdU and DCX positive - since immature neurons in transgenic mice can be made to express the histology marker DCX.

In order to tackle these tasks, a system is implemented that 1) allows novices in the field of image processing to train a segmentation system via an intuitive graphical user interface and 2) that allows the deployment of the trained system on high-performance computers in order to carry out experiments consisting of several groups and replicates.

4.1.3. Image Acquisition

The task of image acquisition gives rise to several problems such as cell clustering and decomposition, non-uniform staining, light attenuation throughout the volume and spatial anisotropy due to different resolutions in (x,y)- and z-direction in the confocal scanning. Usually, images are stored in 16 bit unsigned integer Tiff- or LIF¹ format.

4.2. Methods and Workflow

4.2.1. Overview

Segmentation of the granule cell layer in the DAPI channel and the proliferating cells in the BrdU channel is realized via supervised classification. This requires pixel-precise user annotation and discriminant image descriptors. Figure 4.2 illustrates the process and the necessary user interaction. The labels are provided by brush strokes via a graphical user interface. The labeler can choose between the background class and the foreground object (in one case the granule cell layer in the other BrdU positive cells). Note that, it is important to label over the whole range of the z-dimension in a 3D image. This means that illumination artifacts from the confocal microscopy setup, resulting in lower signal strength in the tail end of the data stack, can be taken into account. Pre-processing steps are also a possible option for correcting illumination artifacts, but this was deliberately abandoned in favor of a more generic and less heuristic classification approach.

For each voxel in the original image data, a descriptive feature vector is computed (see section 4.2.3) and used as input for the classification stage. Trained on the labeled

¹Leica Image Format, for reading lif-files the Bio-Formats Java-library from LOCI the laboratory at the University of Wisconsin-Madison is used (<http://www.loci.wisc.edu/software/bio-formats>)

samples/ voxels, the classifier outputs a probability for each voxel as either belonging to the background or to one of the specified object classes. The random forest classifier [14] is used in the classification stage. These resulting *probability maps* provide the basis for subsequent segmentation. The seeded watershed transform [6] is utilized to segment BrdU-positive cells in which both the seeds and the topological surface are extracted from the classifier output. The granule cell layer is segmented by simple smoothing and thresholding of the corresponding probability map in the DAPI channel.

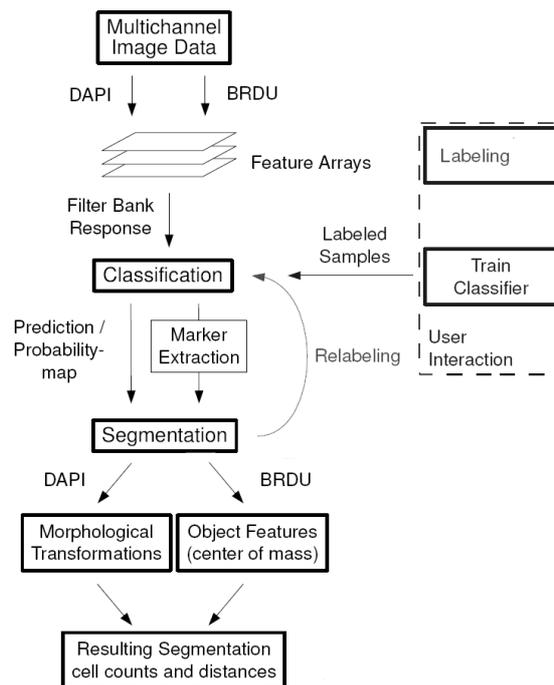


Fig. 4.2.: The proposed workflow: For each voxel of the original image data, a feature vector is computed. Trained with the labeled samples, the classifier outputs a probability as either belonging to the background or to a specified object class for each unseen voxel. The resulting probability map provides both the topological surface for the segmentation and the marker extraction [40]. Results can be corrected by re- or additive labeling

Having found an appropriate segmentation of the objects, two more steps are performed. First the distance of each BrdU-positive cell to the inner border of the granule cell layer is calculated. To determine these measurements, the inner region (i. e. inner molecular cell layer) enclosed by the granule cell layer is computed. Using the anisotropic 3D Euclidean distance transform, each BrdU-positive cell in this inner region can be associated with a distance to the inner border of the granule cell layer. We were also interested in whether a BrdU-positive cell would also be DCX-positive. To

achieve this, the extracted cell masks are utilized. The signal value in the DCX channel is computed for each BrdU pixel in the cell mask. Based on the histogram of those values, an optimal threshold is computed. Each cell that has a mean DCX signal greater than this threshold is designated as DCX positive.

Memory intensive computations like the feature extraction, classification and watershed segmentation are done on a Gaussian downsampled version of the original data. The computations are performed on a computer cluster consisting of up to 30 multi-core PCs² with up to 12 GB of main memory. Analysis results and readouts were exported in csv-file format. The implementation was based on Matlab with underlying C++ routines.

4.2.2. Related Work

Cell segmentation in particular is a key component in most analytic tasks concerned with microscopic images: Not only does a segmentation provide access to specific features related to cell morphology, it also yields information about the spatial organization and distribution of cells. Roughly speaking, one can categorize the most prominent techniques into edge-based and region-based methods based on the features they use.

Edge finding approaches based on active contours or level-set functions use a user initiated boundary that is evolved iteratively to minimize a cost function that includes both local image-related information and prior information about the object boundary. According to Gudla et al. [41] and Ahmed et al. [2], they are able to detect sharp changes in the topology of the objects but do not perform well in segmenting clustered cells. They also require a good initialization and usually results are highly sensitive to this step [45].

Region-based approaches are the most commonly used techniques. They combine region-growing and region-merging methods to cope with over-segmentation resulting from spatial variation in the gray level intensities of objects. A popular method is the watershed algorithm [6]. In Wahlby et al. [108] a seeded watershed is used for cell segmentation in which the foreground seeds are created using the extended h-maxima transform. Merging is done by comparing the mean values of the object borders to a threshold and cell clusters are separated using an inverse distance transformed segmentation result as input for a second watershed run. Lin et al. [70] performs several preprocessing steps, including morphological filtering such as erosion and dilation operations and then uses a gradient-weighted distance transform as input for the watershed segmentation. Following these steps a statistical model-based merging is performed. This merging procedure is refined and extended in two companion papers, Lin et al. [72, 71].

²High-performance computer cluster located at the German Cancer Research Center

Other approaches include Gudla et al. [41] who performs multi-scale edge enhancement followed by watershed segmentation, region merging based on object area, and a subsequent classification into single- and clustered nuclei and Wang et al. [111], who uses a particle moving algorithm on the gradient vector field of an image to generate the seeds for the watershed segmentation. In Fuchs et al. [32] a combination of edge detection and morphological opening and closing operations is used to segment potential nuclei. In order to detect possible false positives, geometrical features of the nuclei such as ellipticity and shape regularity are computed and used to train a support vector machine.

All of the aforementioned region-based approaches require seeds for the (watershed-) segmentation to be effective. Seeds are generated in an unsupervised fashion, i.e. by subjecting the image to appropriate filtering techniques or transformations. In fact, all of these operations make the seeds-, or marker-, extraction dependent on parameters that vary from image to image and thus prior domain knowledge is required.

4.2.3. Feature Generation

To achieve a successful classification it is crucial to compute a sufficiently discriminative feature vector for each voxel $\mathbf{x} \in V$ where V is the volumetric image. Therefore, the features described in Section 2.1 are computed for each volume in a three-dimension manner. In contrast to mere processing in 2D, this features capture information across several z -stacks, which facilitates further processing. The window function is adjusted according to the anisotropy of the x, y, z -resolution.

4.2.4. Random Forest

Random forest is a procedure that grows an ensemble of decision trees and collects their votes for the most popular class, injecting several moments of randomness along the way. Random forests were introduced in Breiman [14]. Tree-based classifiers partition the feature space into a set of hyper rectangles. In particular, classification and regression trees, called CARTs [13], use a recursive binary splitting. These tree based methods and the determination of the best split point for a feature variable are described in Hastie et al. [44, 7].

Each voxel is thus mapped into a high dimensional feature space by computing P feature values for each voxel. Training the classifier defines the decision boundary within this feature space. Each new sample can be classified according to its position relative to the decision boundary. Since the used classifier is an ensemble of single decision tree classifiers the classification step yields a probability map. Random forests have been demonstrated in [14] to have excellent performance, run efficiently and fast

on large data sets and can handle high numbers of input variables (also see Section 2.3.3).

4.2.5. Segmentation

Having obtained the segmentations of the molecular layer and the proliferating cells, the relation between the two channels has to be established to answer the underlying biological questions and to obtain a final result. The DCX-stained channel has to be included to identify the number of proliferating cells that have differentiated into immature neurons. The final part of the workflow is illustrated in Figure 4.3.

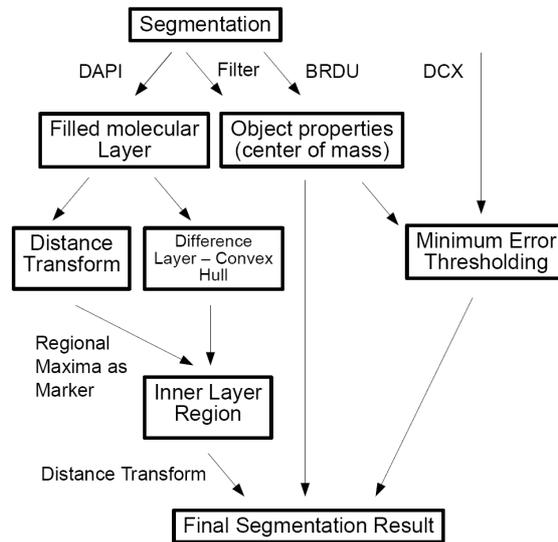


Fig. 4.3.: Post-processing workflow resulting in 3D distance measures of BrdU-positive cells to the inner border of the granule cell layer and DCX positive negative classification [40].

DAPI-stained volume: In order to measure the distance of the proliferating cells from the inner border of the molecular layer, the inner region has to be extracted from the whole layer segmentation. To avoid the introduction of further constraints or assumptions of object properties (i.e. horizontal orientation by simply going through the layer column by column and selecting the two inner points of the layer), the following strategy is applied: As can be seen in Figure 4.4, the layer segmentation is used to generate the difference between the segmentation and its convex hull. One result is shown in Figure 4.4 in the image on the upper right. Different colors correspond to differently labeled connected components after building the difference. Not only does this procedure extract the inner region, it also extracts regions on the outer border of the layer. To resolve the ambiguity, the Euclidean distance transform [47] of the layer segmentation

is computed. All regional maxima of the distance transformed image, that *are not* at the image volume border are selected as markers. These markers are shown in Figure 4.4 in the image on the lower left (for presentation purposes the markers are dilated). By only selecting regions that contain at least one of the markers, the inner region of interest is reliably and uniquely reconstructed. Note that, the Euclidean distance transform takes the anisotropy (i.e. the different resolutions in (x,y) and z-direction) into consideration. The Vigna Computer Vision Library [62] provides fast implementation of the anisotropic distance transform.

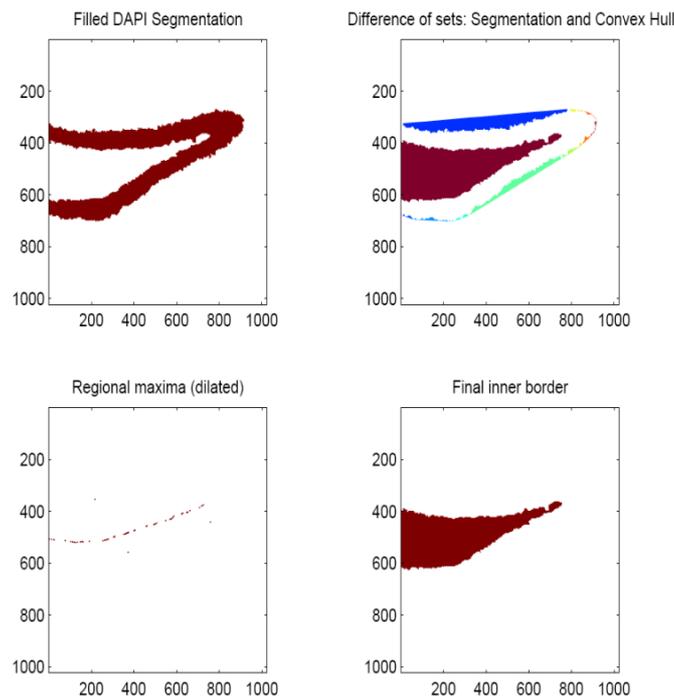


Fig. 4.4.: Extraction of the inner region inside the granule cell layer. Clockwise, starting with the original cell layer segmentation, connected components of the convex hull, maxima of distance transform that are not on the image border, reconstruction of the inner region

BrdU-stained volume: The seeded watershed transform is applied to segment BrdU-positive cells. The seeds are extracted by selecting the local maxima after smoothing the probability map. The watershed operates on the gradient magnitude of the probability map.

Aftwerwards, the DAPI layer is first used to *exclude* the BrdU-positive cells (i.e. discarding all proliferating cells), that are outside the molecular layer. Only cells in which at least one voxel touches the layer are considered. Afterwards, the geometric center of each segmented object is computed. Using the resulting coordinates to look

up the corresponding distance in the final distance transform volume obtained from the inner region of the granule cell layer, we are able to obtain the required distance of the proliferating cells relative to the inner border of the molecular layer.

DCX-stained volume: In order to identify the cells that have differentiated into immature neurons, the DCX stained channel has to be analyzed. Instead of subjecting the DCX volume to the classification procedure it is far more efficient (time and memory wise) to use the already obtained segmentation masks of the proliferating cells and look up the intensity values in the DCX channel at the corresponding locations. Assuming that the histogram is bimodal and consists of two normally distributed populations of intensity values $g(\mathbf{x})$ from object and background voxels respectively, with mean μ_i and standard deviation σ_i :

$$p(g|i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp -\frac{(g - \mu_i)^2}{2\sigma_i^2} \quad i = 1, 2$$

then for given $p(g|i)$ and a priori probability P_i there exists the Bayes minimum error threshold τ such that:

$$p(g|1) P_1 \geq p(g|2) P_2 \begin{cases} g \leq \tau \\ g > \tau \end{cases}$$

The implementation follows the idea of Kittler and Illingworth [56]. This method is more appropriate than the method from Otsu [85] for thresholding images with highly unequal population sizes. Due to the already obtained segmentation, the algorithm is relatively insensitive to the chosen threshold over a wide range. Having obtained τ , it is compared to the mean intensity value computed for each cell in the DCX volume.

4.2.6. Interface

A graphical user interface was developed to guide the biologist through the offline training phase. After the data set has been imported, the three axes showing the x/y-, x/z- and y/z-view can be used for navigation in the 3D-case. In addition, the slider can be used to browse through the slides along the z-axis. Standard functions such as zooming in and out, along with a tool to pan the image are integrated into the interface. The tool is able to load 2D image stacks and 3D data sets and can handle the most common file formats. Arrays stored as Matlab-files that have the extension *.mat* can also be opened. A toolbar provides fast access to the most important tools and functions. Typically, the user provides the algorithm with class labels using the brush tool. There, users can simply paint over each object class with its corresponding color using a brush of selectable size. Mistakes can be corrected with the eraser brush. Correcting or providing additional labels is possible at any point of the process. Feature computation is handled by the feature module.

		Human Labels			Algorithm Results		
		1	2	3	1	2	3
Human Labels	1	0	3.298	2.896	3.303	3.260	3.518
	2		0	2.743	4.186	4.354	3.528
Algorithm Results	2			0	3.278	3.425	3.961
	1				0	1.352	4.373
	2					0	4.735
	3						0

Table 4.1.: Typical Baddeley distances in pixel taken from a 2D slice

As soon as the features have been computed and at least two distinct class labels have been provided, the classification button is enabled. The classifier used is the default random forest with $N_T = 100$. The output is shown in a separate window, where all axes are geometrically linked to the original main view to easily compare the predictions and the original image. Unsatisfying prediction results can be improved by providing additional labels in critical regions. A new random forest is then trained from scratch using the augmented set of labels. The prototype was implemented in Matlab with underlying fast implementations of the algorithms used and features implemented in C++ and is available for download³.

4.3. Results

4.3.1. Evaluation

The algorithm was tested on five different data sets in which the labels were provided by human experts [40]. The evaluation was split into two parts: The comparison of the segmentation results for the molecular layer shape with labels obtained by visual inspection, and those from analysis of the cell segmentation accuracy.

The segmentation results for three different data sets are shown in Figure 4.5. Each column corresponds to one data set, showing one 2D-slice from the volume. The first row shows the original intensity value data, the second row displays the layer segmentation, and the third row shows the extracted inner region that was used for the distance computation in the last part of the algorithm. As the data set on the left indicates, the proposed method yields the expected results even if the layer does not touch the image border. For evaluation purposes, a numerical measure of the discrepancy between two images has to be utilized. In particular, an error metric proposed in Baddeley [4] was used as a numerical benchmark. Pixel/voxel misclassification errors perform

³at <http://hci.iwr.uni-heidelberg.de/people/csommer/clasi/>

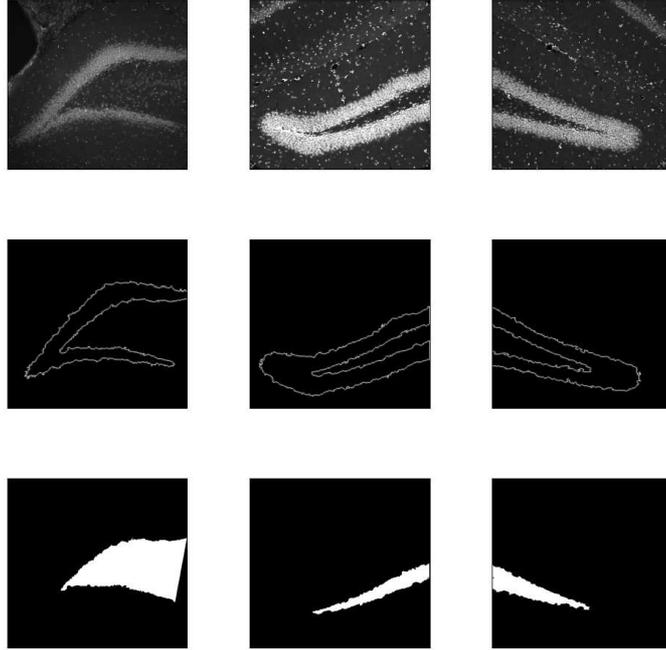


Fig. 4.5.: DAPI segmentation results: Each column shows the original data and segmentation results for a different data set. The first row displays one slice out of the original data volume, the second row shows the layer segmentation result, and the third row provides the extracted inner region of the layer.

poorly when analyzing an algorithm's reconstruction ability since pattern- and shape-information are not regarded. As stated in [4], the displacement of a boundary is given high error values since it usually affects a large number of voxels, even though it might not severely affect the underlying shape. The proposed measure is thus based on the generalized p -th order mean of distance values, which is defined as follows:

$$\Delta^p(\mathbf{L}, \mathbf{S}) = \left[\frac{1}{N} \sum_{\mathbf{x} \in V} |d(\mathbf{x}, \mathbf{L}) - d(\mathbf{x}, \mathbf{S})|^p \right]^{1/p}, 1 \leq p \leq \infty \quad (4.3.1)$$

where \mathbf{L} denotes the label map obtained by visual inspection, \mathbf{S} is the algorithm's segmentation result; and N is the total number of voxels in the data volume V . Finally, $d(\mathbf{x}, \mathbf{S})$ denotes the closest distance between pixel \mathbf{x} and any of the segmentation \mathbf{S} .

Actually, in a more general way, Baddeley [4] uses a cutoff transformation of the distance-transformed images : $w(d(\mathbf{x}, \cdot)) = \min(d(\mathbf{x}, \cdot), c)$ introducing a parameter c that controls the scale. Positions x that are further than c units away from the shapes in \mathbf{L} and \mathbf{S} do not contribute to the sum in equation 4.3.1. This means that the value of the metric does not change if the layer shape is embedded in a space of a different size. Since the interesting question deals with the calculation of the distance of the proliferat-

data set	# cells	TP	FP	FN	Accuracy	Precision	Recall
1	25	25	5(4)	0	83.3	86.2	100
2	18	16	2(2)	2	80.0	88.9	88.9
3	16	13	0(0)	3	81.3	100	81.3
4	23	22	3(2)	1	84.6	88.0	95.7
Average:					82.3 ± 2.1	90.0 ± 6.3	91.5 ± 8.2

Table 4.2.: Performance of the proposed algorithm in segmenting the proliferating cells from the BrdU-stained channel. The algorithm was trained on one of the available data sets; the results from the residual test sets are shown below. The numbers in brackets indicate how many of the false positives are due to over-segmentation.

ing cells within the molecular layer, a value of $c = 200$ that corresponds to an estimate of the upper bound of the layer thickness for all available data sets is chosen. The value of p , on the other hand, determines the relative importance of large localization errors. Here, the arithmetic mean ($p = 1$) is used as a statistical measure since it provides the average pixel difference to the *ground truth*-distance as an intuitive quantity.

Visual inspection (red), algorithm result (blue)

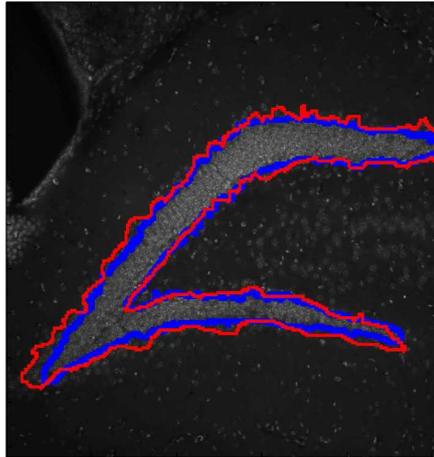


Fig. 4.6.: One slice out of the original data with the labels obtained by visual inspection (red) and the algorithm's segmentation result (blue) as overlay

Even though equation 4.3.1 is directly applicable to the 3D case, evaluation was done on 2D slices only, measuring only the molecular layer discrepancy in x- and y direction. This was done because labeling by visual inspection is performed separately on each slice and does not take information from different depths into account. Also, continuous labels for whole data set volumes are very time-consuming to gather and were

data set	#cells	#detected cells (TP)	#differentiated cells (DCX-pos.)	#differentiated cells (vis. inspection)
1	25	25	21	18
2	28	16	10	9
3	16	13	2	2
4	23	22	16	12

Table 4.3.: Detected cells that have differentiated into immature neurons in the DCX channel and the number of cells found by visual inspection

not available. Figure 4.6 depicts the proposed metric referred to as Baddeley distance for one slice from each data set volume, respectively. Three different human labels were compared to three different algorithm results. The algorithm results differ in the provided labels for classification and in the fact that several steps of randomness are injected during classification. As the Baddeley distance satisfies the axioms of a metric, it is symmetrical and equals zero if and only if $L = S$. The results are shown in Table 4.1 and Figure 4.6 indicating that the results from the algorithm were slightly worse but still comparable to the human labeler.

Cell segmentation Several criteria were used to evaluate the performance of the proliferating cell segmentation. Respectively, TP, TN, FP, FN denote the number of objects being labeled as true positive, true negative, false positive and false negative. The segmentation accuracy is defined as $(TP + TN)/(TP + TN + FP + FN)$, precision - or positive predictive value - is $TP/(TP + FP)$ and recall or sensitivity is defined as $TP/(TP + FN)$, i.e. it measures the proportion of actual positives that were correctly identified as such. From the five available data sets, one was chosen to train the model, while the other four were used as test sets. The second table, in Figure 4.2, shows the segmentation results on the residual data sets. The values denoted in brackets demonstrate how many of the false positives were due to over-segmentation. Table 4.3 shows the algorithm's DCX-positive counts. Note that these closely match the counts governed by visual inspection.

4.4. Conclusion

In order to deal with the overwhelming amount of information coming from digital biomedical image data, (semi-)automated algorithms capable of batch-processing and adapting to the natural variability in data are needed. The method proposed here is able to analyze confocal microscopy images from the dentate gyrus in the mammalian brain, enabling it to establish relations between the different staining channels of the multichannel three-dimensional data volumes. The techniques used within this strategy

include a generic part, where the image voxels are classified by a random forest classifier as either belonging to the object or to the background class. Subsequently, a segmentation is performed on the resulting probability map. Object properties from different channels are set in relation. Particularly, the locations of proliferating cells within the molecular layer of the dentate gyrus are identified. Equipped with an intuitive interface the algorithm provides the user with a segmentation tool of satisfactory accuracy. Once the classifier has been trained on a set of images, it can be processed automatically. The ease with which the system is trained renders the method highly adaptive, and endows it with the potential for application to a wider range of related problems in biomedical image analysis.

5. Tumor-Stroma Interaction

Another application of the proposed framework is the segmentation and quantification of tumor tissues. Images of tumor tissue (and its surrounding) were taken using fluorescence microscopy at different wave lengths to monitor important growth factors. In particular, one is interested in the distribution of blood vessels in the tumor tissue over time. To this end, the tumor and the adjacent tissues are segmented based on the interactive framework as introduced in Chapter 3.

5.1. Introduction

The two most common kinds of skin cancer have advanced drastically in the last decades. The increasing rate of skin cancer is attributed to intensified exposure to UV radiation due to changing leisure behavior in combination with the depletion of the earth's ozonosphere. A long latency period can be observed prior to the formation of malignant skin tumors. Gaining knowledge about the mechanisms involved during the transformation of normal cells into tumor cells (the carcinogenesis) is of great concern for preventive medicine and the development of therapies. In vitro models for human carcinogenesis of skin cancer can be developed to study the biochemical aspects of tumor formation [107] as well as genetic aspects and mutations that are involved in this process. However, in vitro models lack important interactions with the host. Also, it is not possible to distinguish between benign and malignant tumors in these models. Therefore an in-vivo step had been developed in which cell cultures of human cell lines are transplanted onto nude mice. Based on fluorescence microscopy, the interactive framework introduced in Chapter 3 is applied to analyze the carcinogenesis of the transplanted cells. In particular, the learning approach facilitates the segmentation of tumor and stroma tissues.

5.1.1. Tumor-Stroma Interaction

Tumor cells are steadily interacting with the surrounding tissue. Many tumors are *interfused* with normal cells. These non-tumorous tissue parts are called stroma. Analysis of tumor-stroma interactions is crucial to understanding the mechanisms that are responsible for tumor growth and proliferation. An interesting finding is that the influences are bidirectional. The stroma can influence the conditions necessary for tumor growth by

activating special cell types, leading to the formation of supporting cells, inflammatory cells, progenitor cells and vascular cells. Tumor cells, on the other hand, can induce a mechanism similar to wound healing, which leads to the activation of stroma cells. On a molecular level, these interactions are triggered by special proteins called growth factors. Growth factors are capable of stimulating cellular growth, proliferation and cellular differentiation.

By emitting growth factors, the stroma can start or accelerate vascularization, the formation of blood vessels. A therapy might involve modification or destruction of the stroma, or stopping it from growing. This would have an immediate impact on the supply of the tumor and provide a means of influencing its growth and survival. On the other hand, the influence of the stroma has to be taken into account when therapeutic agents are transported into a tumor. In cancer research, the tumor-stroma interactions can help monitor the state of tumors. For example, the amount of blood vessels penetrating the tumor can be linked to tumor malignancy. The main contribution of this work concentrates on establishing stable methods for quantification of such indicators based on image processing to characterize tumor stroma interactions, i.e. is the growth of blood vessels (vascularization / angiogenesis). The basic idea is to compare these indicators inside and outside the tumor and track the changes over time. In summary, the main challenges to solve are the following:

1. Detection of the tumor-stroma border
2. Segmentation of blood vessels
3. Calculation of blood vessel densities in the respective regions

In order to evaluate the algorithms, a data set consisting of 24 multi-channel images with user annotated tumor-stroma borders is given.

5.1.2. Related Work

This approach is motivated by *livewire* [28] image segmentation. Livewire is an image-feature driven method that finds the optimal path between user-selected image locations, thus reducing the need to manually define the complete boundary. This interaction is formulated as a dynamic programming problem: the displayed livewire contour is the shortest path found between the selected points, where the distance metric is based on image information. The image information used by implementations of livewire has included image gradients, Laplacian zero-crossings and intensity values [80]. Here, this approach is extended in two ways. First, the distance metric is based on intermediate and learned *boundary indicator map*, which is the output of a classification stage. Second, an anisotropic metric, i.e. structure tensor, is utilized, which leads to an extension of

the well known Dijkstra algorithm, similar to work done by Jbabdi et al. [48]. Note that the techniques in [80, 28] are user-controlled approaches, where a user is enabled to interactively select points in an image, while in our approach prior knowledge allows the start and end locations to be located on the left or right border of the image; thus there is no need for interactive point selection.

5.2. Material and Methods

5.2.1. Material

Cells from the human skin (keratinocyte cell line HaCaT) are grown in a cell culture. They are subjected to mutations that are essential for carcinogenesis. After being transplanted onto the fascia of nude mice, tissue sections are extracted in regular time intervals. Note that the creation of tissue sections is too destructive and cannot be used to capture the state of a transplant at different times. Different mice with different transplants have to be used to infer the temporal changes.

The different cell types of these tissue sections are then specifically stained with methods from immunohistochemistry. Fluorescent dyes are used to visually separate blood vessels, keratin and cell nuclei. Finally, a fluorescence microscope equipped with a digital camera is used to capture images. The images have a size of 1024×1280 pixels which corresponds to approximately $1mm^2$ resolution. Figure 5.1 shows an example image. The cells are not directly transplanted onto the mouse; instead they are put on a layer of collagen gel. The collagen gel separates the tumor cells from the host tissue for about one to two weeks. This clear separation enables analysis of tumor invasion and the activation of stromal cells. Further, benign and malignant cells can be distinguished at an early stage: benign cells form an epithelial layer similar to human skin and do not invade the collagen gel and the host tissue. Malignant cells induce the creation of granular tissue and growth of blood vessels. Invasive growth can be observed as early as two to three weeks later.

The nuclei of cells that make up the boundary are denser than cells that are not part of the boundary. Also, these cells are slightly bigger. These observations encourage the use of image filters to estimate a *boundary indicator map*, which corresponds to the probability that a given pixel belongs to the tumor boundary. A separation based on such a boundary indicator map might not be without ambiguity and there might be holes in the border due to a lack of local evidence. Also, based on the setup of the transplantation experiment, the border is guaranteed to be a single line; there cannot be any stroma regions inside a tumor. Thus the optimal *boundary path* was to be inferred from the boundary map (optimal in the sense that the path separating stroma and tumor has minimal traveling cost).

Pixels are interpreted to be nodes in a regular image graph with associated weights

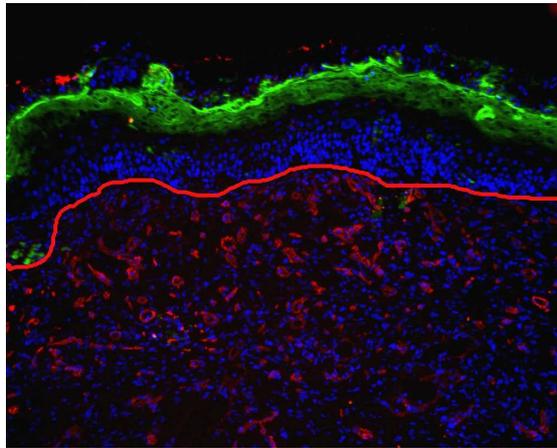


Fig. 5.1.: Example fluorescence image. Blue staining: cell nuclei, red staining: blood vessels, green staining: keratin, continuous red line: manually labeled tumor-stroma border, the tumor is located in the upper region

that are the reciprocal to the probability of belonging to the border. The optimization finds the shortest path through the boundary indicator map, where regions with high boundary evidence correspond to good terrain conditions. All images have the property that the tumor-stroma border originates somewhere on the left and ends on the right brim. Such problems can be solved efficiently with fast-marching algorithms. The result is a smooth polygonal representation of the boundary.

5.2.2. Methods

Two methods come into operation to solve the problem of finding the tumor border: robust classification based on supervised learning to get local evidence of the border location and an optimization technique that uses this evidence to find the border as a global optimal path via fast marching algorithms.

Boundary indicator map: classification strategy: A supervised learning strategy is employed to construct the boundary indicator map from the raw data. As compared to other choices from literature, random forest classifiers show excellent empirical performance [cite]. Random forests are ensemble classifiers consisting of decision trees. They were introduced by Breiman [14] (see Section 2.3.3). A main advantage of random forest is its intrinsic measure for variable importance, which allows for feature selection in the training phase. For the experiments, a random forest consisting of 100 decision trees was used. As illustrated in Figure 5.2, user labels (left) indicating tumor-stroma border and background pixels were used to learn the random forest classifier.

A generic feature set: Image intensities, edge strengths and the eigenvalues of the structure tensor can indicate border pixels and their orientation. A Gaussian filter can

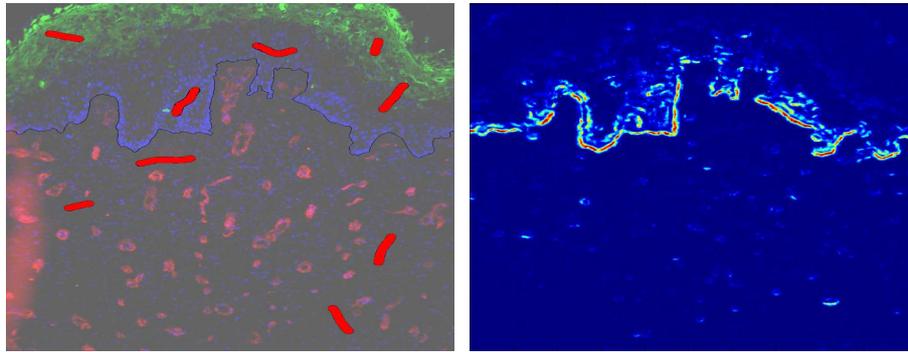


Fig. 5.2.: Generation of the boundary indicator map. The left image shows user labels from biologists for boundary and non-boundary pixels. A typical boundary indication map predicted by the trained classifier (right image) [53].

be used to estimate the density of cell nuclei. Derivative filters might be useful to distinguish between the areas above and below the boundary. To make the feature set more generic, an exhaustive feature set is compiled, computed on multiple scales of the image.

The features are calculated for the green (keratin) and blue (cell nuclei) channels. In total, this amounts to a full feature set containing 130 features consisting of

- Image intensities
- Gaussian filters
- Gradient magnitude
- Derivatives (up to second order)
- Eigenvalues of Hesse matrix
- Eigenvalues of structure tensor
- Central moments (mean, variance, skewness, kurtosis)
- Local quantiles (e.g. median filter)

Feature selection: Not all features of the generic feature set contain enough predictive information to justify their calculation. A huge amount of memory and computation time can be saved when we restrict the feature set to those features that are essential for the classification.

The variable importance scores described in Section 2.3.3 are applied. As shown in Figure 5.3, the Gini importance and permutation importance measure are consistent

with one another. The most important variables are the highly smoothed keratin intensities (V39), vertical derivatives of cell nuclei (V88, V107), gradient magnitude (V105) and eigenvalues of the structure tensor (V12, V116). This is what we would expect: the gradient magnitude attracts the line to the edges, the vertical derivative is important because the images are aligned and the boundary is horizontal most of the time. The derivative can identify the correct side of the edge. Keratin is always part of the tumor and therefore prevents the boundary from crossing it. A strong smoothing filter is helpful in this context, because in most cases there is a gap between boundary and keratin.

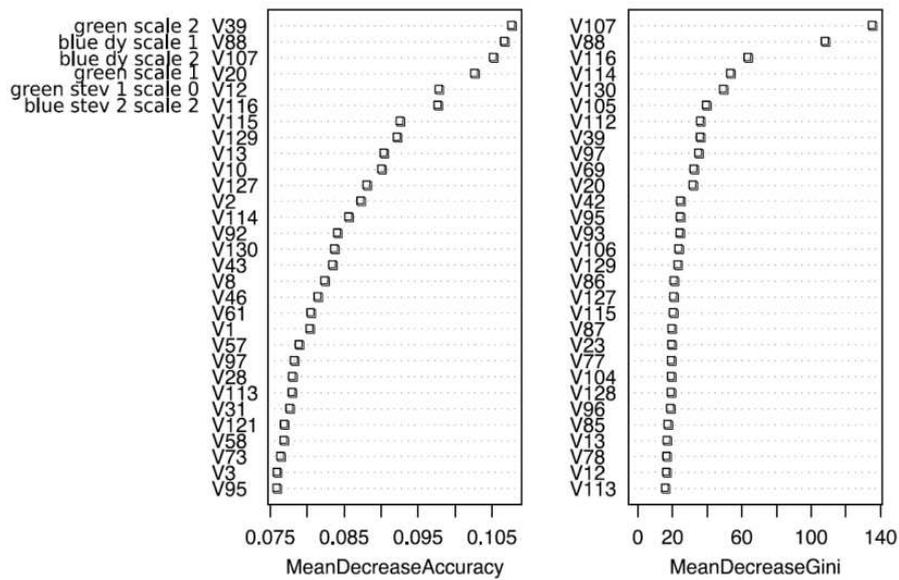


Fig. 5.3.: Variable importance for the used features

Fast marching optimization: Fast marching algorithms are numerical methods used to solve a class of nonlinear partial differential equations called Eikonal equations [48]. These describe the time of arrival at each point in the space as a function of the local speed. When the speed is constant, the geodesics are simply straight lines. The geodesics can curve, when the speed varies across the space, preferring high local speed locations. In addition, if the speed depends on the local direction of travel (e.g., along versus across the tumor-stroma border), then the formulation is said to be anisotropic. Existing methods to solve the isotropic problem are those by Dijkstra [25], Tsitsiklis [102] and Sethian [97]. The well-known Algorithm of Dijkstra discretizes the problem and takes advantage of the fact that transition costs are positive in each point. The positive costs imply that each minimal path is an extension of another minimal path to one of the adjacent grid points. Starting from the first known minimal route (the one to the starting point), the algorithm only considers neighbor points of already-found minimal

routes to extend these to new minimal routes. In [48], a fast anisotropic extension for Dijkstra's algorithm is proposed. It calculates the accumulated distances $\mathbf{u}(x)$ by solving the Eikonal equation with a method similar to Dijkstra's to find the geodesic pathway by following the gradient of $\mathbf{u}(x)$ with respect to some inverse metric tensor \mathbf{M}^{-1} . In our case the distance metric is set to be the structure tensor that provides information about the local orientation and strength.

A geodesic is a pathway minimizing an integral:

$$J(\gamma) = \int F(s, \gamma, \gamma') ds, \quad (5.2.1)$$

where $F(s, \gamma, \gamma') = \sqrt{\gamma'(s)^T \mathbf{M}(\gamma(s)) \gamma'(s)}$ is an infinitesimal distance along a path γ , relative to a metric tensor \mathbf{M} . Let $\mathbf{u}(x)$ be the arrival time function starting from a location x_0 . Then $\mathbf{u}(x)$ is equal to the minimum of $J(\gamma)$ along a pathway connecting x_0 to x . The arrival time function and the geodesics satisfy these two fundamental equations:

$$\nabla \mathbf{u}^T \mathbf{M}^{-1} \nabla \mathbf{u} = 1 \quad (5.2.2)$$

$$\gamma' \propto \mathbf{M}^{-1} \nabla \mathbf{u} \quad (5.2.3)$$

where $\nabla \mathbf{u}$ is the spatial gradient of \mathbf{u} [48]. Equation 5.2.2 is the anisotropic version of the Eikonal equation. In [48] a fast and accurate approximation of the solution is formulated, which is used here to compute the optimal tumor-stroma border based on the previously predicted boundary indicator map.

Blood vessel segmentation: The last step is the segmentation of the stained blood vessels. Until now, the segmentation was performed by a simple thresholding operation, with a manually adjusted threshold. The main problem is getting rid of shading artifacts that occur due to autofluorescence and inhomogeneous illumination. An approach has been developed to simultaneously perform shading correction and segmentation: an estimate of the background is assessed by means of a low pass filter. Then, pixels that deviate more than a multiple of a standard deviation are considered to belong to the foreground and the background is then estimated disregarding the foreground pixels. This is repeated until convergence or a given count of passes. This procedure gives satisfactory results, and is both simple and efficient.

Start and end points: Images are aligned and the tumor boundary starts somewhere on the left side of the image and ends on the right side. The average start and endpoints have been calculated for a set of training images with known boundaries. When processing an image, an additional pixel column is added to each side. The metric on these columns is set to the smallest possible value. Then artificial start and endpoints are set on these additional rows at the height of the averages from the training set. With this trick, the geodesic enters the unpadded image at the start point of the tumor boundary,

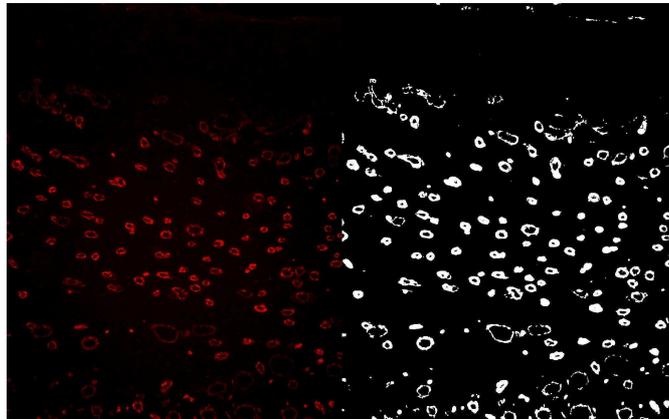


Fig. 5.4.: Segmentation of blood vessels by automated thresholding

which means there is no need for user interaction. Afterwards, the artificial columns can be removed.

User interaction: In cases where the automatic detection has to be refined, the already computed classification can be used to limit the amount of user interaction to a few mouse clicks. The user can draw barriers and attractors into the boundary indicator maps. The barriers simply consist of infinite weights; the attractors amplify the boundary in a region around the user-specified attractor. In this way, the user interaction can be very coarse and the global optimization finds the best path inside a region around the attractor. Figure f5.5 shows the result of an exemplary user interaction.

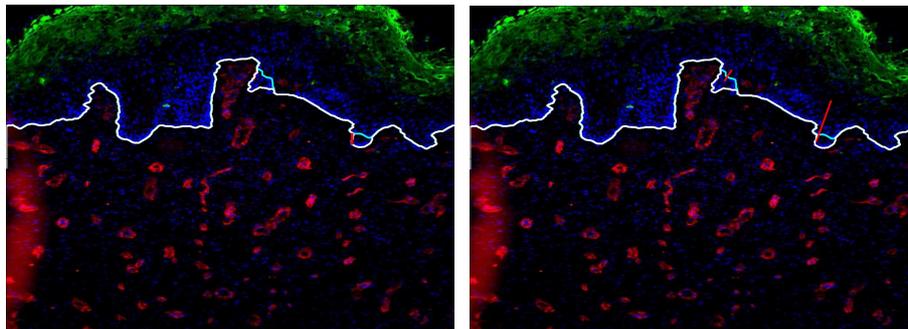


Fig. 5.5.: User refinement using attractors (left) and barriers (right)

5.3. Results

Based on visual inspection, the results are very precise for images from earlier stages that exhibit a good separation between tumor and stroma. The accuracy easily matches

the highest manual labeling quality and can surpass the quality of manual labels that underlie the conditions of a typical workday (see Figure 5.7). In images from later time points where the tumors heavily invade the host tissue (shown in Figure 5.6, the results are still satisfactory, albeit not reaching those from the manual reference.

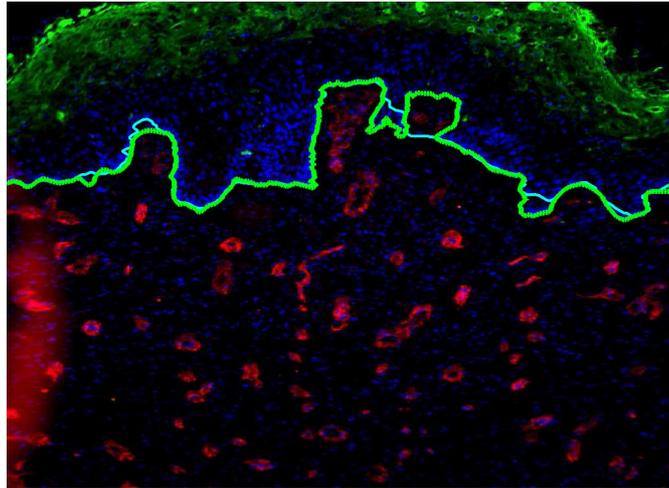


Fig. 5.6.: Tumor-stroma border result for an image from a later time point (heavily invaded tumor): green: reference border, cyan: extracted border. The extracted path (cyan) is taking several short cuts.

5.4. Conclusion

A framework for analysis of fluorescence microscope images is presented to automate a recurring task in cancer research: the quantification of tumor growth and its influencing factors. The labor-intensive manual workflow is replaced by a semi-automated system. Flexible statistical learning methods have been applied to make the solution generic and easily adaptable. Detection of tumor boundaries posed the main challenge up to now. This has been approached by choosing sets of generic features that contain enough discriminative information. Supervised statistical learning is then performed on these features, resulting in a classifier that is able to predict the boundary probability. Finally, a global optimization technique is applied to obtain the course of the boundary as the best path based on the boundary probabilities. Results present excellent performance for an image coming from an earlier point in time with regard to the underlying biological experiment. For more complex formations of the tumor-stroma border, usually occurring at later time points, the algorithm yields satisfactory results. The problem of short cuts is still a challenging research topic. In order to supply the user means to

correct wrong border sections, an interface that provides attractor or blocking guides to refine the border probability was developed.

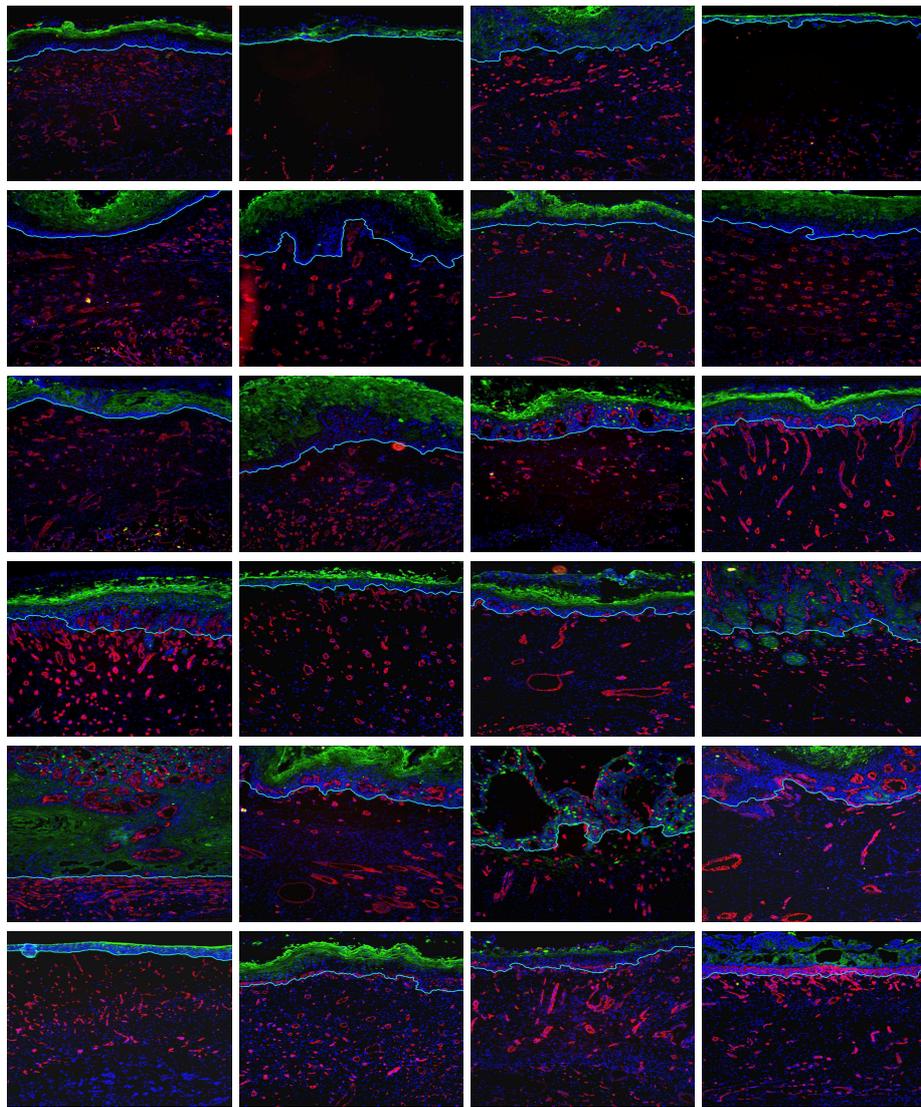


Fig. 5.7.: Resulting tumor-stroma borders (cyan) overlaid onto the 24 microscope images

6. Viroquant and Image Quality Control

6.1. Introduction

An important source of information about cells is provided by fluorescence staining in combination with optical microscopy. Digital analysis of images produced by a digital camera attached to a fluorescence microscope allows for fast automated high-throughput detection and quantification of spatial, spectral and temporal distribution of emitted fluorescence from single cells. The basis for all automatic image analysis required in high-content cell screening applications is cell segmentation.

RNA interference (RNAi) is a system within living cells that allows to control which genes are active and not (knocked down). RNAi plays an important role in understanding cell mechanisms, allowing the investigation of virus-host interactions. Performed in a high-throughput manner – meaning the investigation of thousands of genes – the procedure is referred as *RNAi screening*. Cells are automatically seeded (and transfected with siRNA) onto so-called well plates: an array of wells that can be accessed by an automated microscope. Subsequently images of each well are taken corresponding to different wavelengths of the fluorescent cells. Several distinct image positions in a well can be accessed, resulting in millions of images. Biological replicates of the transfection specimen help to improve the significance of the statistical read-out. The enormous amount of image data taken during the screening process requires a reliable and fully automated image processing architecture. More than 1.5 million images, each with a size of more than one mega pixel, have been processed, which is way beyond what human labor alone could accomplish.

In order to ensure reliable and reproducible analysis results, not only do the underlying methodology and algorithms have to meet high quality requirements, but also the image data itself. For large-scale screens, the absence of disturbing contaminants cannot be fully guaranteed by careful experimentation. Only high-quality microscopic images should be used to assess biomedical readouts. Image regions containing dust, hairs, contaminants or other unwanted objects would falsify the fully automated analysis and must therefore be excluded beforehand. To that end, the interactive framework (Chapter 3 is adapted to play the role of *image quality control*. Multi-class learning is replaced by one-class learning, the class of clean images. A classifier is trained on a sufficient

number of images that do not exhibit artifacts. After training, the classifier can detect unexpected artifacts like dust, pollution or hairs in unseen images.

In particular data handling and access are of vital importance in assuring efficient processing. Within the proposed screening platform, a whole workflow has been developed, containing interface definitions from the raw output of the microscope, as well as the design of databases to feed further statistical processing.

The organization of this chapter is as follows: In Section 6.2 the basics of RNA interference and related work concerning the high-throughput analysis of HIV assays are briefly reviewed. All essential information regarding the preparation and experimental setup for the screening assay¹ can be found in Section 6.4. Image analysis is described in Section 6.5 in more detail. Section 6.6 depict the proposed approach to image quality control and its related methods. How all those different steps are integrated into an approved workflow can be found in Section 6.7. Finally, results are shown in Section 6.8 and discussed regarding future directions of the screening platform.

6.2. RNA Interference

RNA interference (RNAi) has emerged as a powerful technique for studying loss-of-function phenotypes by specific down-regulation of gene expression, allowing the investigation of virus-host interactions by large-scale high-throughput RNAi screens. In collaboration with the University Hospital Heidelberg (department of virology) a robust and sensitive siRNA screening platform [15] is presented². The workflow was established to elucidate host gene functions exploited by viruses, monitoring both suppression and enhancement of viral replication simultaneously by fluorescence microscopy. The platform comprises a two-stage procedure in which potential host factors are first identified in a primary screen and then re-tested in a validation screen to confirm true positive hits. Subsequent bioinformatics enables identification of cellular genes participating in metabolic pathways and cellular networks utilized by viruses for efficient infection. The workflow has been used to investigate host factor usage by the human immunodeficiency virus-1 (HIV-1), but can also be adapted to other viruses. Importantly, the description of the platform will guide further screening approaches for virus-host interactions.

Despite considerable advances in virological research over the last few decades, viruses continue to represent a major health risk, responsible for millions of deaths worldwide each year. Like other viruses, HIV-1 has evolved the ability to successfully infect - and efficiently transmit between - human cells by recruiting various host proteins for each step of its life cycle [18, 74, 75]. Unraveling these critical cellular factors will not

¹Assay and experiments done by Kathleen Börner, Johannes Hermle and Dr. Maik Lehmann

²This work was funded by the BMBF (FORSSYS) project VIROQUANT.

only improve the fundamental understanding of HIV-host interactions, but could eventually lead to novel anti-HIV therapeutics. Since the rate of mutations of cellular genes is substantially lower than for viral genomes, the particular benefit of targeting host factors is that it may provide a higher barrier to the generation of anti-drug resistance. A most powerful and versatile approach to identifying such potential cellular interaction partners of HIV-1 are RNAi-based loss-of-function screens, as suggested by very recent reports [12, 57, 116].

Although each of the three high-throughput screens published thus far reported a large number of potential host cell factors, there is very little overlap between the different sets [16, 38, 86]. This might be explained by differences in the individual experimental conditions, such as the use of distinct cell lines, siRNA libraries or virus strains, all of which could have significantly affected the results. However, it may also be due to the use of different criteria for defining a *hit* or inconsistencies concerning the techniques applied to validate potential hits. This highlights the need for comparable experimental conditions in further studies and for the selection of consistent analytical methods for future screening approaches.

In [15], a sensitive, automated microscopy-based siRNA screening platform, designed to elucidate host factors utilized by a variety of viruses, was described. In the Viroquant platform, several sub-genomic siRNA libraries were tested in a primary screen and the identified potential hits were subsequently re-confirmed in a validation screen using different siRNAs. In both screening stages, a non-silencing siRNA and an siRNA targeting the HIV-1 specific cell surface receptor CD4 are used as negative and positive control, respectively. In addition, the knockdown efficacy and cytotoxicity of siRNAs are determined within the validation screen. Genes that show similar effects in both the primary and validation screens are considered to be validated. Bioinformatics and modeling approaches on the validated hits, in combination with published HIV-1 host factors, known metabolic pathways, and protein-protein interactions enable the identification of cellular networks and pathways involved in the replication of HIV-1. Further studies using this platform will characterize fundamental cellular functions of the identified hits and shed light on their role in viral pathogenesis.

6.3. Related Work

Several other pipelines for supporting high-throughput screening have been developed in the past. The most famous one is perhaps *CellProfiler* [17]. The software supports basic image processing, databases and visualization and is implemented in Matlab. The classification package *CellAnalyst* [50] is designed to closely interact with CellProfiler in order to promote cell classification and more advanced readout statistics. The Large-Scale Digital Cell Analysis System (LSDCAS) [24] was designed to provide a highly extensible open-source live cell imaging system. The LSDCAS software runs on Linux

and consists of a core library written in C++ and graphical analysis software (GTK). It also supports Web access to the generated experimental data. The chief design goals were to provide for the analysis of living cells [24].

P. Horvath and colleagues develop the tool *Advanced Cell classification (ACC)*, which is a follow up extension of the *HTC-Browser*. It enables browsing through data recorded in standard plate formats and integrates an illustrious set of classification techniques for cell classification. The software requires several Matlab toolboxes and was applied in a human genome-wide screen³.

The *High-Content Data Chain (HC/DC)* is a data management, processing and visualization software designed for biologists working in the field of high-throughput screening. HC/DC is a platform for workflow generation and data exploration. It is based on *KNIME (Konstanz Information Miner)*, an open-source project that tries to embed data integration, processing and analysis in a comprehensive exploration platform. Several interfaces, for example for CellProfiler and ACC, exist. The famous *Bioconductor* project [35] also contains several packages relating to high-throughput screening, which focus on statistical analysis as presented on the workshop [84].

6.4. Preparation and Experimental Setup

A key element of the experimental setup is the reverse transfection of commercially available siRNA libraries into HIV-permissive HeLa P4 cells. Following a 36 h incubation period to allow target knockdown, the cells are infected with HIV-1 virions encoding for GFP. This allows the straightforward detection and quantification of infected cells via a highly sensitive automated microscopy-based assay. 384-well plates were used for the high-throughput screening approach. For validation screens, 96-well plates were used, allowing collection of even more images per well compared to the 384 wells. All experiments were performed with HeLa P4 cells, as they are well suited for culturing in all of the tested well plates and also highly transfectable with siRNAs.

To keep replicates as reproducible as possible, the siRNA libraries were printed by batch using a previously described reverse transfection protocol [26, 27]. To this end, the respective siRNA are transferred automatically to a 384-well plate. After drying the well, the substrates can either be stored for up to 15 months without any loss of efficacy or can be directly used for knockdown studies. This is a major advantage of the reverse transfection method compared to liquid transfection, as it permits better reproducibility and comparability between different plates of the same batch.

For the primary screen [15], various libraries containing silencer siRNAs (Ambion, Applied Biosystems, Austin, TX, USA) were used in 384-well plates, with one individual siRNA per well and three distinct siRNAs per target gene. Potential host factors

³as mentioned on <http://acc.ethz.ch/acc.html>

identified in the primary screen as *hits* were subsequently re-tested within a validation screen with a minimum of five replicates in 96-well plates. At least two novel chemically modified siRNAs (silencer select siRNAs, Ambion) were used per gene to enhance specificity and minimize *off-target effects*. Two siRNAs against CD4 were chosen as positive controls. CD4 mediates cell entry of HIV-1 and thus represents a pivotal host factor for HIV-1 infection. Gene knockdown efficiency of approximately 90% within a period of 24-92 h after siRNA transfection. Thus, in subsequent experiments cells were infected 36 h after transfection and fixed after an additional 36 h.

6.4.1. Image Acquisition

For high-throughput image acquisition a fully automated epifluorescence ScanR screening microscope equipped with the ScanR acquisition software is used. Images are acquired with a 10x objective in nine positions per well for 384-well plates, up to 16 positions for 96-well plates. In each position images are acquired in the Hoechst and in the GFP channel using the corresponding excitation and emission filters. In the validation screen a third channel is acquired, Phalloidin, which stains actin filaments in the cell, and as such, is a useful marker for investigating cell morphology.

6.5. Image Analysis

Each position in a well consists of two channels (or three in the validation screen), which depict the cell nuclei and the cell body, respectively. This fact suggest the following two-step procedure: First segment all cell nuclei in the primary DNA channel (Hoechst), then use the obtained cell nuclei mask as initial solution for the whole cell segmentation in the secondary channel (GFP or Phalloidin) and gradually refine them to obtain a full cell mask. This well-established approach has been widely used and various variants (usually in a region based manner) exist in literature [17, 51, 109]. For the sake of efficiency in the context of high-throughput analysis, a similar approach is adopted within the cell segmentation procedure. Thus, the image analysis consists of the following three successive steps:

1. Cell nuclei segmentation in the Hoechst channel
2. Cell segmentation in the GFP/ Phalloidin channel based on cell nuclei identification
3. Quantification of cell characteristics

A detailed workflow can be found in Figure 6.1. The inputs of the image analysis routine are the Hoechst and GFP channels, representing cell nuclei and cell cytoplasm

(see Figure 6.2a). As output, we compute the number of nuclei, the average signal intensity over all cells (in the GFP channel), and the proportion of cells that have been infected. These calculations are performed per position in a well and serve as input to the statistical analysis.

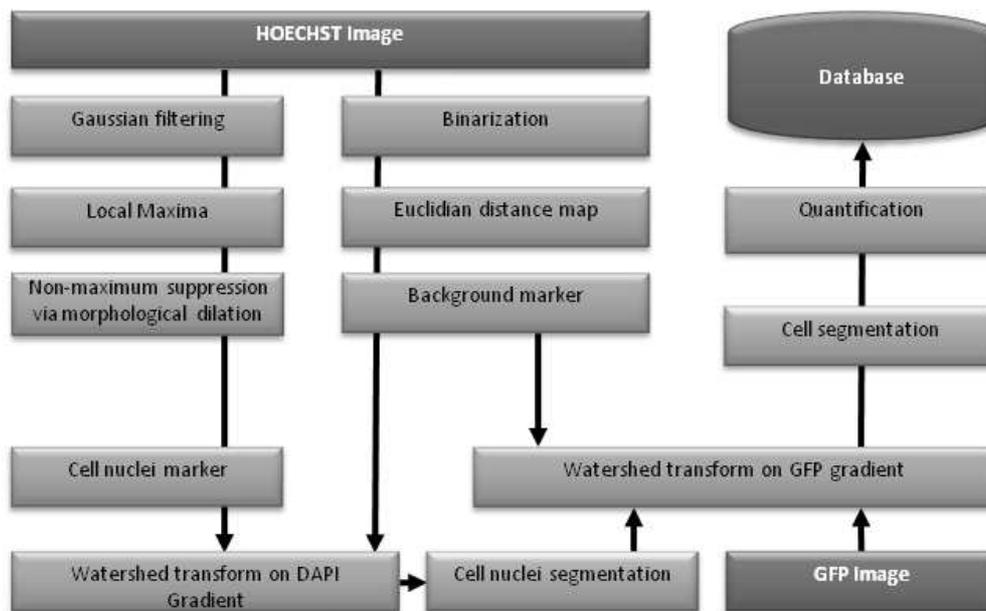


Fig. 6.1.: Workflow diagram for the primary screen.

6.5.1. Cell Nuclei Segmentation

To segment cell nuclei, we use the marker-controlled watershed transform method [6]. This method consists of two basic steps. First, the algorithm detects a unique initial foreground marker for each cell nucleus (typically the marker that corresponds to the center of the object). At the same time background markers that lie in the dark region between cell nuclei are extracted (see Figure 6.2b). Second, the watershed transform (see Section 6.5.4) expands markers spatially to enclose the cell nuclei in the image. Finally, basins originating from foreground markers correspond to masks of cell nuclei. A recent review on the topic of cell segmentation can be found in [22].

6.5.2. Cell Nuclei Marker Extraction

A preliminary binarization of the Hoechst image is found using a histogram adaptive threshold [85], leading to an initial nuclei versus background mask. In the case of

uneven illumination of the image, background correction can be applied before binarization. A method proposed in [73] is employed for this purpose. A Gaussian blurred version of the cell nuclei image is used to detect foreground markers (the variance for the Gaussian kernel is empirically adapted to cell nuclei size). All local maxima above the binarization threshold are determined and dilated to suppress spurious markers (see Figure 6.2b).

To extract background markers, the Euclidean distance map from the initial binary mask is computed. This map contains the distance from each non-nucleic pixel to its nearest nucleic pixel in the binarized image. All ridges from the distance map are extracted to get a Voronoi-net like background marker (as illustrated in Figure 6.2b). Both types of markers (background from distance map, foreground from Gaussian blur) serve as the final markers for the watershed transform operating on the gradient image.

6.5.3. Cell Segmentation

The extracted cell nuclei masks are passed to the cell segmentation process as foreground markers. In the primary screen where no actin channel (dyed by Phalloidin) is available, background markers from the cell nuclei segmentation are reused. Note that we cannot compute background markers directly from the GFP-channel, as only infected cells are visible in this channel. When an actin channel is available – like in the validation screen – this channel is used to directly compute background markers from its gray values. Segmentation is also done on this channel if available, while the major readout still is taken from the HIV-sustaining channel GFP. Actin is a very appropriate candidate for cell segmentation because it is distributed throughout the cell's cytoskeleton and on the cell wall indicating the cell boundary.

6.5.4. Watershed Segmentation

One algorithm (and its many variants [82]) that is considered to be a very accurate tool for the segmentation of fluorescence microscope cell images is *seeded watershed*. In the watershed algorithm, the intensities of an image are interpreted as elevations in a landscape. To build the watersheds, the gradient magnitude image in which water begins to rise from local minima was computed corresponding to areas of low gradient (such as areas inside the cell or in the background regions). Wherever two water basins are about to merge, a watershed barrier is built (see left sketch of Figure 6.3 for an illustration). Ideally, the watersheds coincide with the boundary of the cell, assuming those edges are well defined. Instead of letting water rise from every local minimum in the image, water rises only from previously extracted seeds (right picture in Figure 6.3). This design ensures that there are as many regions as the number of initial seeds. The

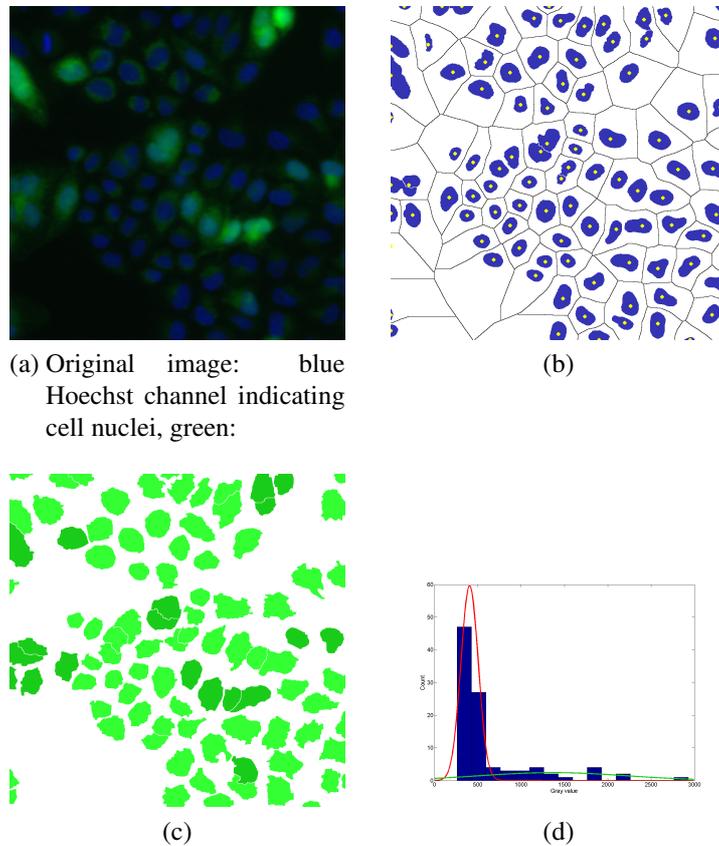


Fig. 6.2.: Illustration of voronoi-like watershed segmentation

algorithm is implemented as a seeded region as described in [1] and is used as realized in the *Vigra* computer vision library [62].

6.5.5. Quantification

The main readouts of the image analysis – used in the further statistical analysis so far – are the overall mean gray value over all cells in the image, the total cell count and the infection ratio. Uninfected cells appear dark, while infected cells show rather high gray values over their area. A Gaussian mixture model with two components is fitted to the extracted mean gray values (see Figure 6.2d). One Gaussian component explains the variation of uninfected cells, while the other accounts for the variation of infected cells. If either all cells or no cells are infected, one of the two components should vanish, however, in practice, an intermediate number of infected cells produces two overlapping components. To remedy this, a prior is imposed on the position of the two Gaussian means, which essentially acts as a repelling force between the mixture

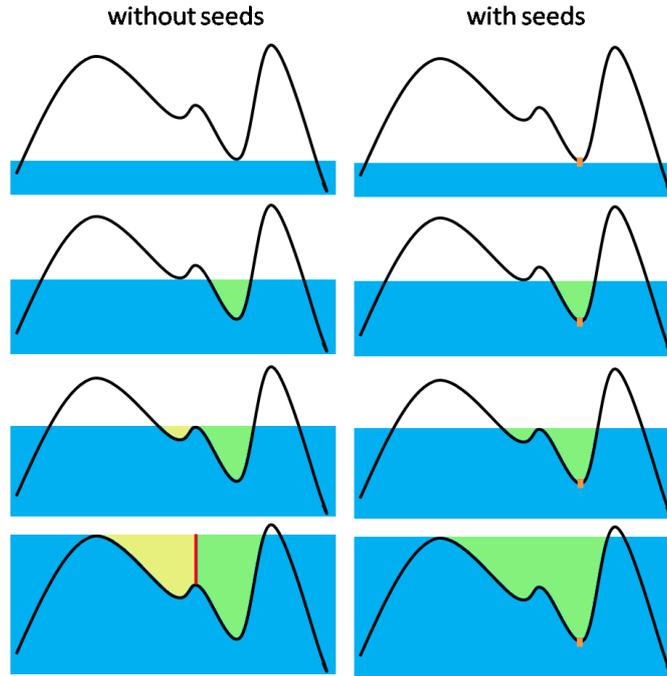


Fig. 6.3.: Illustration of the watershed algorithm. Left: Original watershed formulation: water rises from all local minima, Right: Seeded watershed transform: seeds (orange) are used to control the segmentation process]

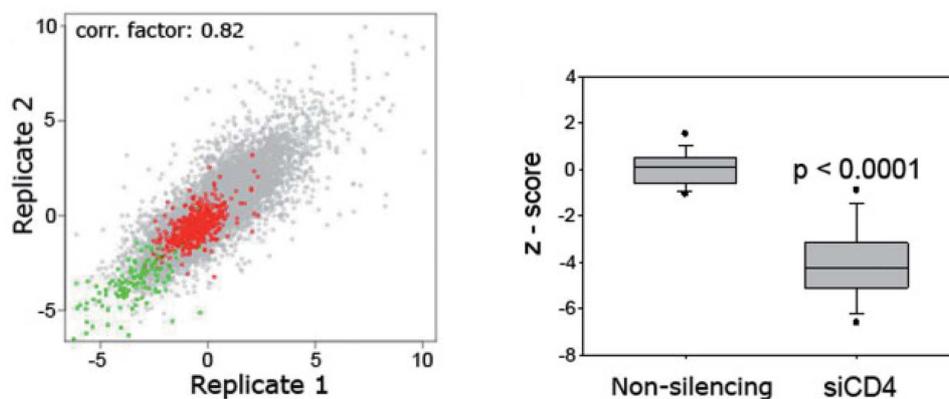
components. Having found the parameters of the mixture model, an optimal threshold:

$$\omega_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} = \omega_2 \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \quad (6.5.1)$$

$$\Rightarrow \sigma_1^2(x - \mu_2)^2 - \sigma_2^2(x - \mu_1)^2 - 2\sigma_1^2\sigma_2^2 \ln\left(\frac{\sigma_1\omega_2}{\sigma_2\omega_1}\right) = 0 \quad (6.5.2)$$

is computed by solving the quadratic equation in Eq. 6.5.2, where ω_i are the mixing coefficients, μ_i the Gaussian means and σ_i their standard deviation. Then each cell can be classified by thresholding as infected or not infected. Finally, the infection ratio is computed as the number of infected cells divided by the total cell count.

The readouts generated on different biological replicates correlated well as shown in Figure 6.4a. The aforementioned positive control (CD4) showed the expected down regulation with a z -score of about -4 compared to non-silencing control (see Figure 6.4b).



- (a) Correlation between z-scores of two replicates of an RNAi screen to investigate host cell functions in HIV-1 pathogenesis with individual siRNA-mediated gene knockdowns (gray), non-silencing control siRNAs (red) and CD4 positive controls (green). The Pearson correlation coefficient is 0.82.
- (b) Z-score distributions of the non-silencing control siRNA and the positive control CD4. The separation of the two distributions confirms CD4 as a significant down-regulator in the assay.

Fig. 6.4.: Comparisons between the non-silencing control siRNAs and CD4 positive controls

Integration into iChip

Besides the primary readouts to assess HIV infections under genetic variation, more additional measures have been taken. In addition to the primary readouts to assess HIV infections under genetic variation, additional measures have been taken. For example in the work of Young et al. [113] various measures describing the cell's appearance and shape were successfully employed to investigate cell morphology. Akin to that approach, a set of generic cell descriptors (see Table 6.1) is calculated on a single cell level from each channel. In total, this sums up several billions of cell measurements (assuming about 400 cells per image). Future comparative studies – concerning cell morphology changes under the reported cell treatment – may benefit from the huge data collection.

To ensure accessibility of these results, all readouts, including cell segmentations and quality assessments, are stored and integrated into the iChip framework [64]. iChip is presently installed within several European and German research networks. It holds experimental data from key members of the *National Genome Research Network*. The goal is to avoid isolated data silos by using comprehensible and reusable data structures, while being a comfortable and easy to use solution for management of experimental high-throughput experiments. It enables meta analysis to public scientist based

on well annotated data of different incorporated studies. Data types of different technologies (Affymetrix, CDNA, tissue microarray, array CGH, mass spectrometry, gel electrophoreses, RNAi) are supported in iChip.

intensity mean	Average intensity in gray values
intensity variance	Standard deviation of intensities in gray values
area	Area in pixels
perimeter	Perimeter in pixels
eccentricity	Eccentricity as a measure of elongatedness
position	Center position measured in pixel of its center

Table 6.1.: Additional single-cell measures for the channels Hoechst, GFP and Phalloidin

6.6. Image Quality Control Using One-Class Learning

Only high-quality microscopic images should be used for performing a reliable and reproducible siRNA screen. Image regions containing dust, hairs, cords or other unwanted objects could falsify the fully automated analysis and must therefore be excluded beforehand. To find these erroneous regions, we propose an unsupervised learning approach. A one-class support vector machine is trained on faultless images to detect impurities without the need for user interaction.

6.6.1. One-Class Support Vector Machine

The one-class support vector machine was introduced by Schölkopf et al. in [95]. The basic idea is as follows. Given input samples

$$\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X}$$

and a regularization parameter λ , the aim is to minimize a regularized empirical risk functional given by:

$$\hat{f} = \arg \min_{f \in \mathcal{H}} \lambda \|f\|^2 + \frac{1}{n} \sum_{i=1}^n (1 - f(\mathbf{x}_i))_+,$$

where the function $(\cdot)_+$ denotes

$$(a)_+ = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$$

and \mathcal{H} is a Hilbert space with an associated reproducing kernel [105]. The minimization of this regularized empirical risk yields an estimate that tends to be 1 at the locations of input samples (data term), while being as small as possible at the same time (regularizer), as measured by the associated scalar product of the used kernel. When a kernel is used that only depends on the difference between sample points $\mathbf{x}_i - \mathbf{x}_j$ (such as the Gaussian kernel, Schölkopf et al. [95] proved that this minimization can efficiently be calculated using the standard support vector machine methodology (see 2.3.4) with the origin as the only negative example in training, thus finding a hyperplane with maximum margin of separation to the origin.

The features used to characterize local quality appearance comprise the aforementioned feature descriptors in a patch-wise manner. Features are calculated on the Dapi and GFP channel (most contamination are visible in these channels) and agglomerated in image patches of size 64×64 . Several statistics are computed for each patch, including: mean response, standard deviation, higher-order central moments, entropy and quantiles. In the inference step, a sliding window approach is utilized to spatially predict the regions of high deviation from the patches discovered in the training phase. This results in a quality map of the same size as the input images. Thresholding this map provides the region mask for bad quality areas. Spurious responses due to noise (small isolated regions) are suppressed by introducing a minimal lower limit for the size of found regions. Once regions are extracted, all cells that overlap these regions are excluded from further analysis⁴. In Figure 6.5 an example that shows a correctly identified contaminated region as such is given. Cells depicted in red are excluded from the readout generation.

For the experiments and the processing of the two screens, the implementation of the one-class support vector machine in the LIBSVM library [19] is used.

6.7. Implementation and Pipeline

A fully automated pipeline was set up to process the large number of images produced by the screens. Image processing was implemented in Matlab and C++ and runs in a data parallel fashion distributed over a compute cluster. For each well position, output that automatically maps data into a relational database is written to a virtual file system for convenient post-processing.

A graphical user interface has been developed and deployed to the biomedical partners (see Figure 6.6a for a screenshot). Although it is doomed to run on a single desktop machine, it has turned out to be a useful tool for visually exploring the image data in order to tune certain parameters (for instance smoothing parameters σ).

⁴For more details regarding the implementation and a for comparison with other one-class learning techniques see the forthcoming diploma thesis of Christian Scheelen

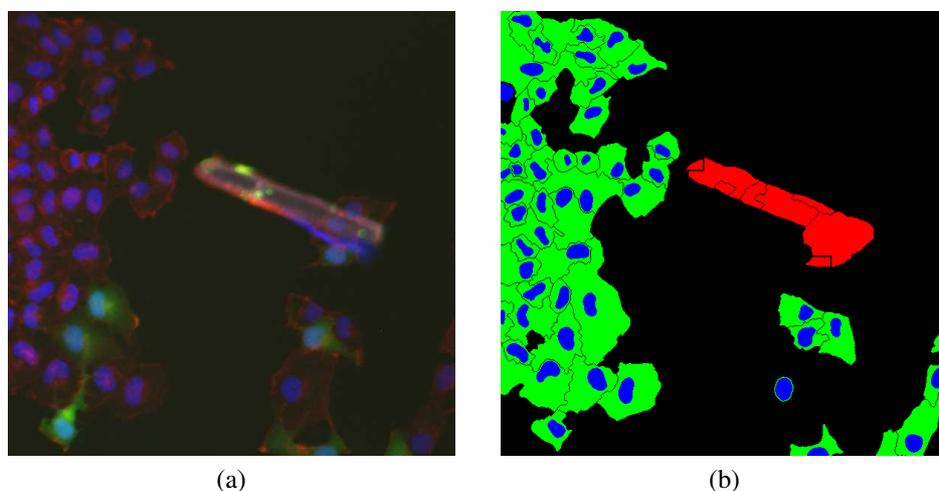


Fig. 6.5.: Example of a contaminant correctly classified as such (red) and excluded from the result.

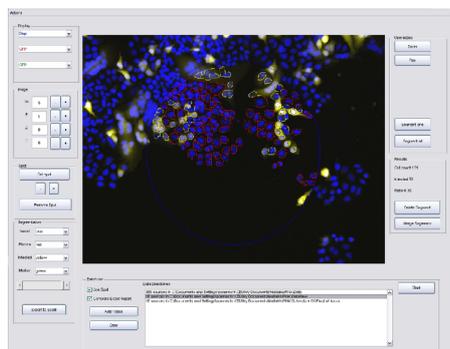
6.7.1. Graphical User Interface

The user interface can handle the image data of whole plates and facilitates browsing the microscope data. Channels are visualized in their corresponding color. Both single images and whole sets of selected plates can be processed. When processing single positions within a plate, the segmentation boundaries and the positive/ negative decisions (HIV infection) are overlaid onto the original image. The adjusted parameters (within the graphical user interface) are stored in a configuration file which is then directly usable by the parallel pipeline.

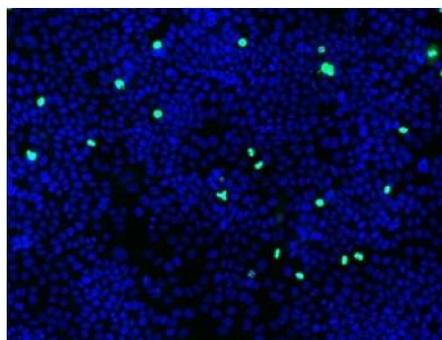
Additionally, abnormal mitotic and apoptotic behavior was tested using the graphical user interface as shown in Figure 6.6b. Here, several plates were customized for this purpose. A mitosis marker (phospho-Histone H3) indicates chromosomes separation while the TUNEL-method [34] (TdT-mediated dUTP-biotin nick end labeling) is used to dye apoptosis. Instead of counting HIV-positive cells, the tool is used to classify the cell's phenotype into mitotic and apoptotic. The parameters can easily be adjusted to cope with the new appearance in the readout channel. Analysis of these results revealed that mitosis and/or apoptosis rates stay in ordinary ranges.

6.7.2. Parallel Pipeline

In order to process the enormous amount of data, a parallelized pipeline was set up as an integral part of the screening platform (see Figure 6.7). At first, the data is stored on a NAS (Network Attached Storage) file server in a hierarchical fashion. Starting from



(a) Screenshot of the graphical user interface



(b) Example image showing cell nuclei (blue); some are in the mitotic phase (green) marked by Phospho-Histone H3 (PH3)

Fig. 6.6.: The user interface (left) enables users to browse the image files according to the plate coordinates. An example of using the interface is shown (right) to check for mitotic abnormality.

the root folder *raw-data* the image can be found by following the path: *microscope* → *plate-layout* → *replicate* → *data*. All image files are stored in 16-bit TIFF format and their file name encodes the well number, position number and channel name. A virtual file systems maps all image files to a similar tree structure, where both the well and the position are made explicit by extending the path by: $\dots \rightarrow \textit{well} \rightarrow \textit{position}$. In these leaf folders, the images are saved by their channel name (for example: *gfp.tif*). Now, the image processing routines are designed to get a tuple (or triple for the validation screen) of ordered image file names and to write the analysis results back to the input folder – one file per previously defined readout. These generated files (in the virtual file system) are dynamically mapped to a backend database. A batch module triggers the computation by dividing the raw data entities into blocks of 100 serial jobs. Each block is executed in a separate process on the computer cluster comprising twelve 8-core machines.

6.8. Results and Conclusion

The use of parallel batch processing yields a major increase in speed, as a typical screen with 1.7 million images requiring 4.5 TB of storage is processed in approximately 36 h (see Figure 6.2 for the key data of the main screens). Automated image quality control is being added to exclude foreign bodies and artifacts in images before quantification, resulting in improved measurements. Another goal is to integrate the interactive learning framework for cell classification into this pipeline, allowing simple adaptation to

	Primary Screen	Validation Screen
Replicates	2-3	5-9
Format	384 wells	96 wells
#Positions	9	16
#Plates	182	115
#Channels	2	3
#Images	1,257,984	529,920
Time Required to Process	< 36h	< 24h
Σ	1,787,904 images \approx 4.5 TB	

Table 6.2.: Key data of the main RNAi and its validation screen

novel cell types, sensor settings, chemical dyes and assay-specific phenotypes, without reprogramming the image processing routines.

A modular and flexible microscopy-based RNAi screening platform is described for investigation of host factors involved in virus-host interaction. This uses a two-stage procedure (primary and subsequent validation screen) comprising four main steps: experimental assay, image analysis, statistical analysis and bioinformatics, each of which has been presented in detail. The platform was demonstrated in stably CD4 expressing HeLa P4 cells using a modified infectious HIV-1 strain carrying a GFP reporter (HIV-1-AGFP). The procedure was shown to be suitable for robust and sensitive detection of host cell factors involved in HIV-1 replication using different cell culture formats. Design and testing of standardized experimental setups for production of sufficient amounts of virus were found to be critical to obtaining reliable and comparable data sets. Dedicated image processing procedures were developed to process the very

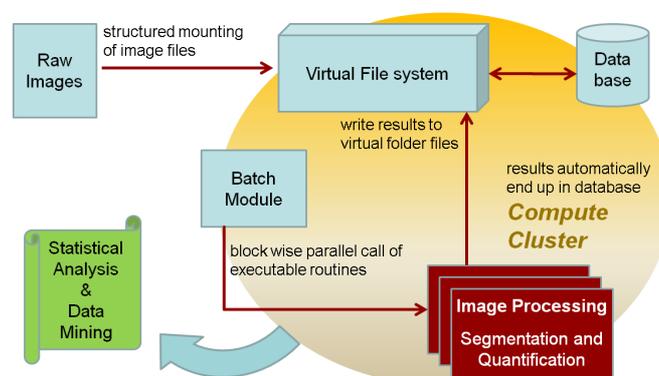


Fig. 6.7.: Sketch of the parallel pipeline

large amounts of high-throughput image data (tens of terabytes over the lifetime of a full genome screen) in reasonable time. Problems associated with the need for different data normalization approaches in primary and validation screens were addressed in the statistical analysis step, which also dealt with multiple potential sources of error in the readout. The bioinformatics step integrate experimental results with data mined from public databases, allowing screening hits to be functionally classified and embedded into known pathways and protein-protein interaction networks. Future extensions will include automated classification of images based on cell morphologies (using the interactive framework), so that knockdown phenotypes can be classified in more detail and better correlated with biological processes. Comparative analysis using other viruses on the same platform will elucidate commonly used cellular host factors exploited by different viruses, which may serve as novel drug targets for broad spectrum antivirals. Finally, we would like to encourage efforts at standardization of RNAi screening procedures and offer experience with this platform as a robust basis on which to build new systems.

7. Copper Wire Tracing

In the last Chapters, the interactive framework was successfully applied to problems arising in life science. Here, an application to an industrial segmentation problem is described, where accurate pixel-wise tracing of fine copper structures is necessary. Again, the interactive framework is utilized to provide the basis for further processing. The whole interactive learning approach was integrated into a framework for product authentication currently developed by our collaboration partner.

7.1. Overview

Authentication frameworks¹ for consumer commodities often rely on forgery-proof labels in combination with barcodes. Here, the aim is to provide an image processing methodology to the backend of the label generation process. Labels are generated by spreading copper wires on an approximately 1cm^2 sized square label. The randomness of the spatial distribution of the spread copper wires is of essential concern. Furthermore, the whole process is designed to be very difficult to imitate. Once it has placed the copper wire onto the label square, the probe is varnished with a transparent coating for fixation. Now, the labels must be automatically analyzed to extract a vectorial description of all copper wire segments inside the field of interest (that is the label square).

An industrial, monochrome camera is used to capture images of the labels under favorable light conditions. Preprocessing steps such as cropping the field of interest, affine transformation to align the image, and rescaling to a fixed resolution are designed to provide constant conditions for the extraction process. However, the problem of light reflexes due to the coating and raised wires structures cannot be entirely solved by the image acquisition setup. Therefore a robust method that 1) detects the copper wires reliably and 2) is able to deal with the inverted edge appearance due to light reflexes must be developed (see Figure 7.1a for an example).

The extraction is based on the segmentation of the copper wires and undergoes three different main steps:

1. Feature generation: A set of discriminative features is manually selected; among them Gaussian Filters, eigenvalues of the Hessian matrix and a structure tensor.

¹The real underlying application is still confidential, which is why it is not explained here in more detail. As a part of this thesis, however, the contribution from the image analysis side needs to be explained.

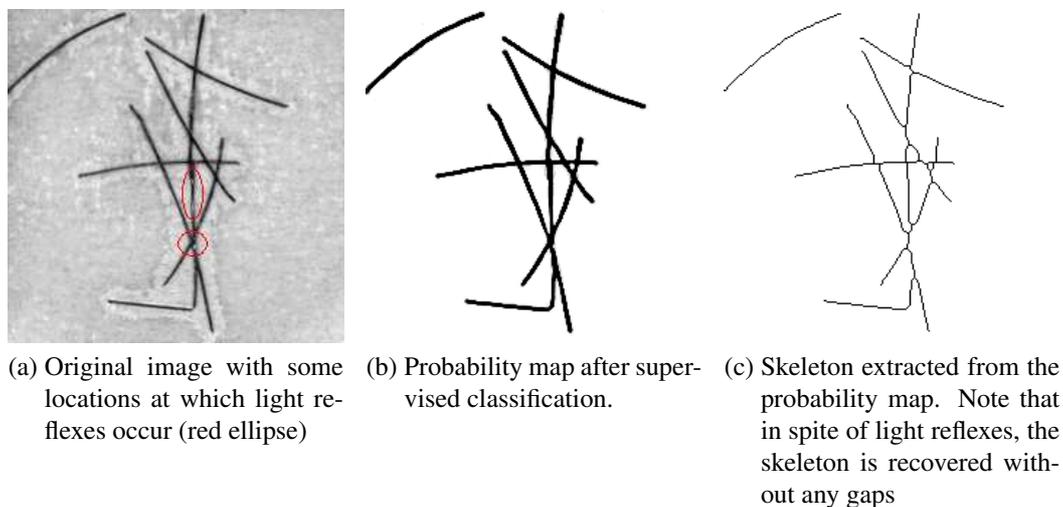


Fig. 7.1.: The proposed methodology based on supervised learning

The features describe important aspects that appear in the images. For example, a copper wire corresponds to spatially oriented curvature in the digital image domain. The large eigenvalue of the Hessian matrix is an excellent measure for this, as was demonstrated in the domain of blood vessel segmentation [91].

2. Supervised classification: The random forest classifier [14] is used to learn from the examples given. Copper wires and backgrounds were marked by brush strokes of varying sizes via a user interface. In Figure 7.2, images with overlaid user labels are shown. Note, that the images were partially labeled, meaning that not all the pixels were marked as a representative sample for copper wire respond/or background. Having *learned* a random forest from the extracted features and label information, the classifier predicts new, unseen pixels. Each tree of the forest votes for the class as either copper wire or background in this case. Once the votes are collected and averaged, a probability value is yielded for each of the classes at every pixel. The majority vote (probability > 0.5 for a two-class problem) is used to obtain a binary class assignment to background versus copper wire.
3. Skeletonization: Roughly speaking, skeletonization refers to subsequent *thinning* of a binary image until only a pixel wide spine is left. This step is necessary to form the basis for a vectorial description of the copper wires. Sampling from the skeleton then yields a vectorial description of copper wire sections.

7.2. Material and Methods

7.2.1. The Data

Preprocessing steps such as cropping the field of interest, affine transformation to align the image, and rescaling to a fixed resolution are performed prior to labeling and copper wire extraction. This is done and implemented based on the library BIAS² [81] by an external company.

Three different labelers partially marked copper wires and background. A user interface supported the labeling process by providing zoom and pan tools. The effect of labeling on the classification can be previewed by performing an intermediate learning and prediction step on the current image.

A data set comprising 39 copper wire images was labeled. Two-thirds of it (26 images) were used as training images. The remaining set (13 images) served as test images for estimating the generalization ability of the proposed supervised classification methodology. In this way, the results of the classifier, learned on two-thirds of the labeled sets, can be compared to the remaining user labels given in the test set.

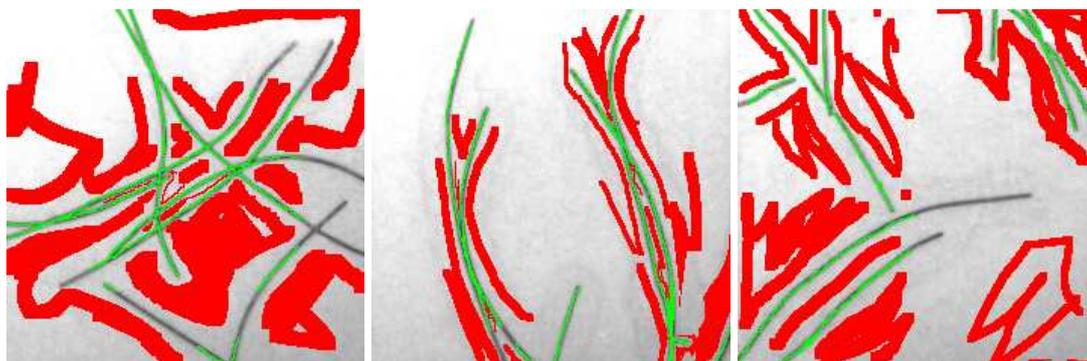


Fig. 7.2.: Three examples from the labeling process. Copper wires (green) and background (red)

7.2.2. Feature Generation

The features in Table 7.1 were selected due to their ability to describe local intensity value, edge strength, local curvature and corner response. The original value was also used. The feature response is shown in Figure 7.3 for five image sections. Each column corresponds to a feature in ascending order from left to right. The features were computed at different scales, that is under different Gaussian filter bandwidths σ . Gaussian

²The Basic Image AlgorithmS C++ Library (BIAS)

smoothing needs to be done because gradient generation typically increases the noise in the image.

- 1: (OV) Original value
- 2: (GF1) Gaussian filter (scale 1)
- 3: (GF2) Gaussian filter (scale 2)
- 4: (E1H1) Large eigenvalue of the Hessian matrix (scale 1)
- 5: (E1H2) Large eigenvalue of the Hessian matrix (scale 2)
- 6: (E1S1) Large eigenvalue of the structure tensor (scale 1)
- 7: (E2S1) Small eigenvalue of the structure tensor (scale 1)

Table 7.1.: Manually selected features

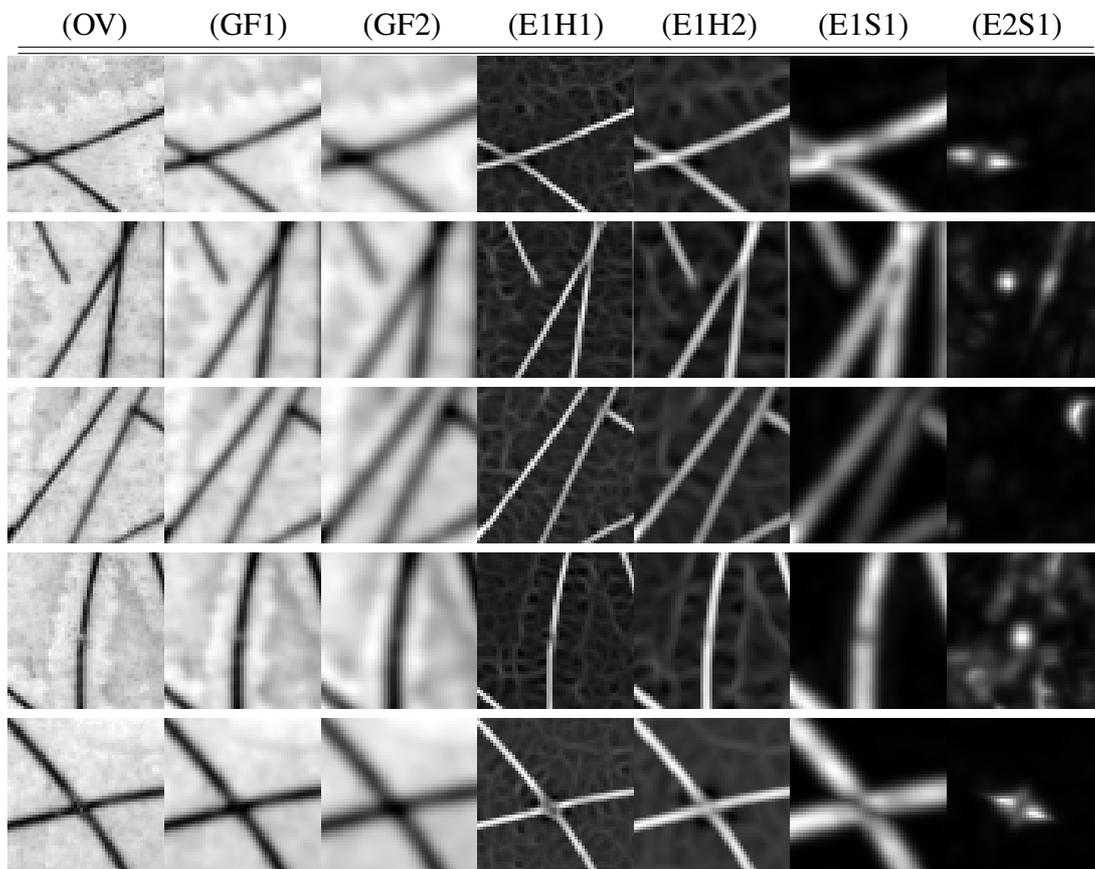


Fig. 7.3.: Feature response for the selected feature set. A center section of the original-sized image is enlarged for better visibility.

Feature 5 (=E1H2) presents, for example, a strong increase in contrast. The small eigenvalue of the structure tensor (feature 6 = E2S1) can be interpreted as a corner detector and indicates locations where two copper wires meet, cross or get very close to

each other. Gaussian Filters reduce background artifacts and reveal the main run of the wire. A non-linear combination of these features can achieve a high level of accuracy as shown in the results section.

7.2.3. Skeletonization

Skeletonization is the *thinning* of a binary image to a one pixel-wide spine. Lookup tables are usually used to repeatedly remove pixels from the brim of objects. The skeletonization is based on an a thinning algorithm from [115] implemented in BIAS [81]. A resulting skeleton is shown in Figure 7.1c.

7.3. Results

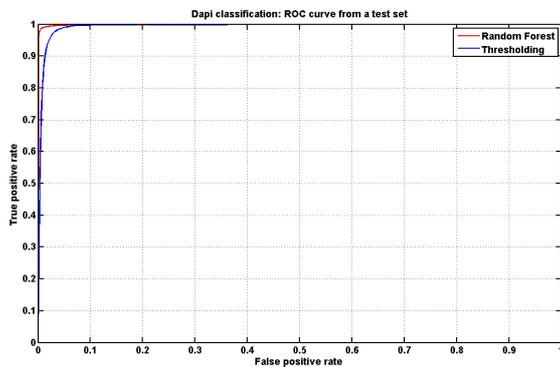
To evaluate the strength of the proposed methodology, a ROC curve analysis was performed. In signal detection theory, a receiver operating characteristic (ROC) is a graphical plot of the sensitivity vs. (1 - specificity) for a binary classification system over a varied discrimination threshold (see Section 2.4).

The resulting ROC curve can be seen in Figure 7.4a. There, the threshold on the probability map for the hard assignment to copper wire or background is varied and the false positive rate versus the true positive rate is plotted at every threshold. An alternative approach based on optimal threshold selection [85] is also shown. The proposed method achieves an area under curve of $A_z = 0.9963$, while the more simplistic thresholding approach (on gray values) produces many false alarms, resulting in a flatter curve shape.

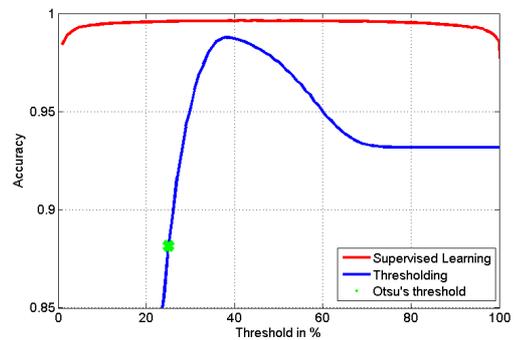
Figure 7.4b depicts the threshold robustness of the two methods. Pixel accuracy is plotted versus the selected threshold. One can conclude that, for the classification approach, a broad and almost symmetric range of possible threshold values exist, while for the alternative thresholding method (Otsu's method), a non-optimal threshold would have been selected.

7.3.1. Conclusion

The results of the proposed methodology attest: discriminant features, together with a non-linear classifier, achieved a more accurate and robust performance. In particular, light reflexes can be handled without problem-specific measures and are intrinsically learned by the classifier. The random forests show very good generalization performance, which is demonstrated by the ROC curve analysis. However, the feature generation and classification procedures take more time to compute. The bottleneck is identified to be the prediction time of the random forest. The feature computation is done in less than 0.2 sec. on a 1.5 GHz notebook, while prediction time is about 1.5



(a) ROC analysis of the proposed method (red) on a hold-out test set. Training was done on 26 images and prediction on 13 images. For comparison, results from thresholding are plotted in blue.



(b) Threshold selection: The robustness of the probability map on its threshold selection (red). For comparison, the results from thresholding are also shown.

Fig. 7.4.: Evaluation of the proposed method in comparison to thresholding.

sec (random forest with 100 trees) on the same machine. An obvious solution to this problem is to use an appropriate multi-core CPU, where the prediction can be easily distributed due to the fact that each tree in the forest is independent from the others. An alternative is to stop the prediction process earlier when a first fraction of trees agree to some extent.

8. Process Control for the Sheet Feeder

8.1. The Sheet Feeder - An Overview

Process control is a statistics and engineering discipline that deals with mechanisms and algorithms for controlling the output of a specific process. In this case, the process refers to one subprocess in offset-printing. Offset printing is a commonly used printing technique in which the inked image is transferred (or *offset*) from a plate to a rubber blanket, then to the printing surface [55]. In this thesis, we are particularly interested in how sheets of paper originating from a paper stack arrive at the core units of a printing machine. The very beginning of the print process is the sheet feeding. *Sheet-fed* refers to individual sheets of paper or paperboard being fed into a press. A lithographic press uses principles of lithography to apply ink to a printing plate. Sheet-feeding is commonly used for printing magazines, brochures, letter headings and general commercial (job) printing.

Studying this process and the development of adequate algorithms to quantify the state and quality of the current process state is the focus in this chapter. At first, we describe the sheet-feeding process. As illustrated in Figure 8.1 the process undergoes five major steps. In the side sketch of the sheet feeder (Figure 8.1a) the different steps are illustrated. They consist of:

- 1) Constant loosening of the upper sheet stack with various air blasts from the side and back of the stack (see Figure 8.1b)
- 2) Lift off the top sheet using suction cups
- 3) Triggered air support in order to provide a air cushion
- 4) Handover to the drag suction cups which
- 5) Movement in flow direction towards the suction belt

Various errors can occur during the feeding process. Amongst them are the feeding of two or more sheets of paper at once (double sheet), slanted papers (slant sheet) and errors in arrival time (early or late sheet). Usually these errors can be detected at later

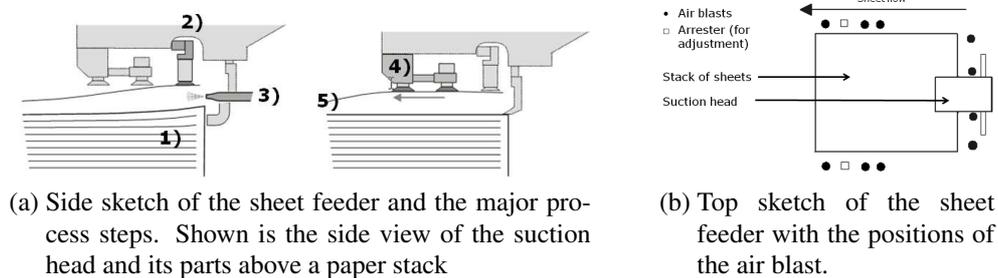


Fig. 8.1.: The sheet-feeding processes

stages in the process (when the sheets enter the press), which usually leads to an instant stop of the whole work process. This involves comprehensive checkups and clearance by the machine operator. After all sheets have been removed or aligned the process can be resumed.

Not only do errors of this kind generate spoilage (maculature), they also delay work progress. The average time a printing machine is out-of-order due to technical disturbance is an essential measure of printing efficiency and an important cost factor. Thus, early detection of sheet errors and adequate automatic adjustment are in significant demand.

To this end, a camera that points toward the upper front part of the paper stack was set up. Constructional restrictions do not allow for frontal imaging, so the camera system is attached near the front corner of the paper stack. A laser diode strip is worked into a mechanical hatch to provide adequate illumination. The exposure event is triggered by the machine clock, in the way that one image is acquired once per machine cycle. An example of such an image can be found in Figure 8.5; note the (unavoidable) projective distortion due to the lateral camera setup.

With this technique, image sequences are recorded in order to approach the problem of sheet errors by image processing and machine learning techniques. Process parameters such as machine speed and feeder parameters are adjusted and varied to enforce errors and non-error sequences. Sequences and frames are labeled weakly¹ with a binary process state: normal or error. Image processing is used to extract meaningful characteristics of the sheet separation as a possible predictor for the error state. Feature selection is applied to purify the information content. Then a classification system is trained to learn a decision boundary between two classes from the given examples. Once the new system has been trained, new sequences can be predicted on the basis of the learned classifier. In particular, a learned characterization can be used to project the (usually) high-dimensional description onto a human-readable representation in the

¹Weak labeling refers to labels that are given roughly. For example: a whole image has a specific label without indicating an exact location

form of a *heat-map*, where the color of the current state (represented as a dot on the map) indicates its process state. Hence, the feeder parameters can be adjusted by a human operator to guide the current state point into uncritical (green) regions of the state map.

8.1.1. Experimental Setup

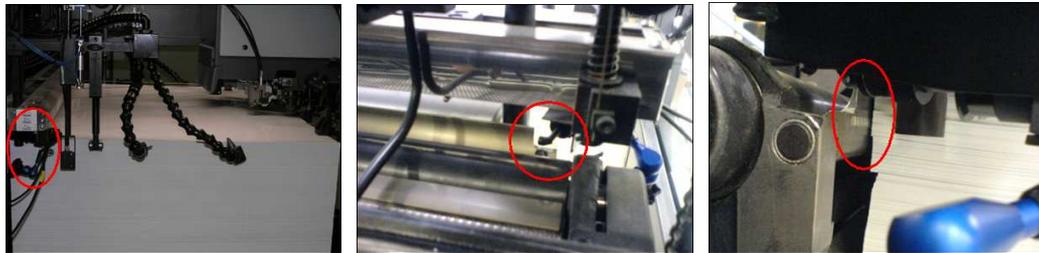
The experimental assembly is shown in Figure 8.2, and summarized in the following:

- Illumination and exposure is triggered automatically once per machine cycle. The camera is therefore connected to the machine signal generator.
- It is possible to automatically record other machine/ process properties via the COM interface, for instance the current speed, sheet format and the kind of paper.
- Due to the fact that an increased depth of focus is required; a lens was developed by an external company.
- A industrial camera with global shutter and trigger is used (see Section 8.2 for details).
- The scene was illuminated by a diode array integrated into a mechanical hatch emitting red light.
- Exposure and flash time were set in the range of 400-800 μs to avoid motion blur.

All components were optimized by direct experimentation to ensure ideal acquisition conditions.

8.2. Data Acquisition

The sequences were recorded using the industrial camera uEyu 2250-M/C [36] from the vendor IDS. The camera is equipped with the Sony CCD image sensor ICX274AL, which features square pixel arranged in a 1600×1200 array. Each pixel has a size of $4.4\mu m$ width and height. Progressive scan allows all pixel signals to be output independently. The chip has a global shutter, which makes it possible to realize full-frame images without a mechanical shutter. At a wavelength of $600nm$ (red) the chip has a relative efficiency of 0.75. However, some sequences were recorded using an IR-A diode array emitting light with wavelength $750nm$, where the relative response drops to ≈ 0.5 . Especially with the red light, the decreased chip response was compensated by the rather high efficiency factor of the diodes.



(a) Side view of the sheet-feeder with loaded paper stack: red ellipse depicts the camera setup
 (b) Frontal view of the camera setup with the first prototype lens in use (red ellipse)
 (c) Front corner of the feeder: in the red ellipse, diodes built into the mechanical (movable) hatch become apparent

Fig. 8.2.: The camera setup: assembly of the camera, diode array and optical lens at the sheet feeder

The aforementioned restriction of the camera location at the corner of the sheet stack created several drawbacks. The projective distortion results in a high resolution of parts near the camera, however, parts far apart appear smaller. This effect can be harmless as long as the projected object is in focus over most of its length. To this end, an engineering office² was delegated to manufacture a lens system capable of exploiting the *Scheimpflug principle* [94]. This principle states that in optical imaging, the lens plane, image plane and plane of focus intersect in one line. For the normal camera setup, all of these planes are parallel and intersect in infinity. When the camera is tilted, one can manipulate the plane of focus. A planar object that is not parallel to the image plane can be completely in focus. The Scheimpflug principle is illustrated in Figure 8.4. A test image (stripe pattern) of the camera setup illuminated with red light can be found in Figure 8.3 depicting the good depth of focus over a range of 500mm.

8.2.1. Data Sets

Two different data sets were recorded. The first data set consisted of 1075 frames from several sequences. 183 Frames were from an error sequence (double sheet), while the remaining 892 frames originate from sequences where no error occurred. To provoke these kinds of errors, feeder parameters were deliberately distorted by an experienced machine operator.

The second data set was designed to capture the dynamics of the feeding process. This time, parameters of the feeder were not left static (as in the first data set), but were varied during the process. 58 Sequences are recorded in which up to three parameters

²IB/E OPTICS Ing.-Büro Klaus Eckerl IB/EOPTICSIng.-BüroKlausEckerl

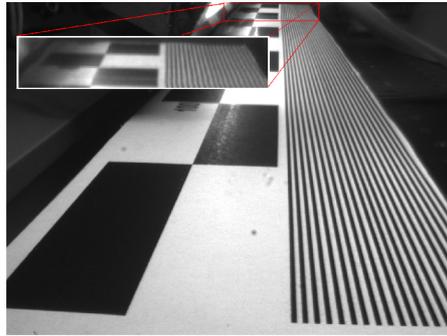


Fig. 8.3.: Test pattern image with the camera and the newly developed optic lens. Length of both the black and white box is 50 mm. Each line (on the right part) has a width of 1mm.

were changed at the same time. The parameters to be varied were selected based on their apparent influence on the process and on the basis of experiential reports [77]:

1. Speed of the machine in unit: sheets per minute
2. Air nozzle strength, corner position
3. Air nozzle strength, side position
4. Air cushion strength
5. Position of the side air nozzles (front, middle, back)

Each parameter is discretized into three values. The middle value refers to the default setting (proposed in the machine's manual), while the others severely deviate into both extremes (for example: almost no air versus full air). Recording of the second data set, which was supported by a machine operator, took about two days.

8.3. Feature Extraction

The computation of adequate and discriminant features is crucial for successful classification. The features considered for this kind of data capture important aspects of the imaged paper stack, such as the height of the stack and the loosening and separation of the paper. To this end, a feature extraction procedure that measures frequencies and their place in the image is proposed. The frequency corresponds to the loosening and separation of the paper stack (subsequent transition from dark to bright), while the spatial position accounts for the stack's structure. Short-Time Fourier Transform (STFT see section 8.3.1) is performed to gray level profiles of the image and its *spectrogram*

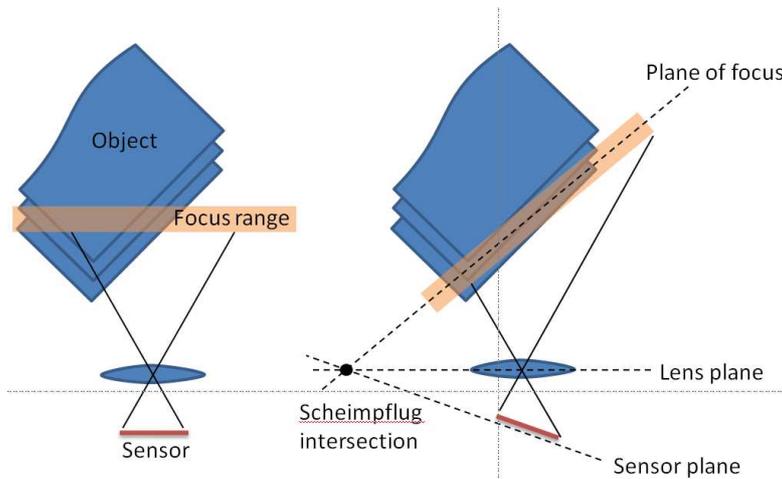


Fig. 8.4.: Scheimpflug principle. On the left the usual camera setup is depicted. The Scheimpflug principle can be utilized to align the plane of focus with the object to achieve a focused projection.

is computed. The profiles correspond to gray value columns taken from the image at different locations. In Figure 8.5, the feature extraction procedure is illustrated. This allows for the efficient computation of image features in 1D.

Similar features have been used in blood vessel segmentation and fingerprint enhancement, for example [21, 42].

8.3.1. Short-Time Fourier Transform

The short-time Fourier transform (STFT) is a frequency-related transform used to determine the frequency content of local sections of a signal as it changes. In the discrete case, the signal to be transformed could be broken up into sections (which usually overlap each other, to reduce artifacts at the boundary). Each section is Fourier transformed, and the complex result is added to a matrix, which records magnitude and phase for each locus and frequency. This can be expressed as:

$$\mathbf{STFT}\{x\} = X(m, \omega) = \sum_{n=-\infty}^{\infty} x(n)w(n - m)e^{-i\omega n} \quad (8.3.1)$$

The magnitude squared of the STFT yields the *spectrogram* of the function:

$$\mathbf{spectrogram}\{x\} = |X(m, \omega)|^2 \quad (8.3.2)$$

Spectral leakage causes energy from distinct spectral features to *leak* into adjacent frequency channels, giving rise to spurious components in the signal's frequency spec-

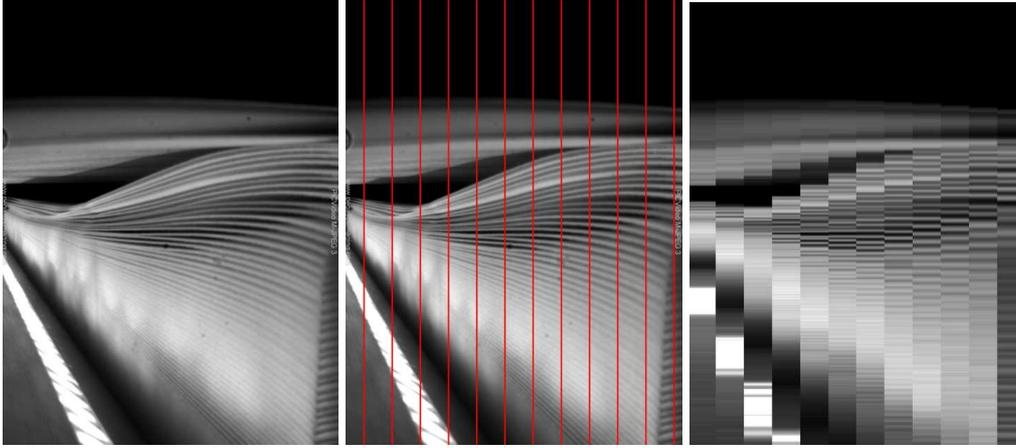


Fig. 8.5.: Feature extraction. From left to right: Original image, selected columns in red, and the resulting gray-value profiles

trum [98]. While spectral leakage cannot be eliminated entirely, its effects can be reduced. This is done by applying a tapered window function in which the sampled signal is multiplied by a function that tapers toward zero at either end. This reduces the effect of discontinuities in which the mismatched sections of the signal join, and hence also the amount of leakage. A number of tapered window functions have been devised. A popular choice is the *Hann window* function (see equation 8.3.3), which is essentially one cycle of a shifted and scaled cosine function, so that it has a maximum value of 1 in the middle and fades to zero at either ends:

$$\omega(n) = 0.5 \left(1 - \cos \left(\frac{2\pi n}{N-1} \right) \right) \quad (8.3.3)$$

Before the signal is transformed, its gradient is calculated and smoothed with a Gaussian filter in order to reduce noise. The discrete Fourier transform is computed with the same size as the window size, leading to 64 frequency bins. The frequency is cut off, maintaining the bins of only the lower 32 frequencies. The highest frequency within those 32 corresponds to a wavelength of four pixels, which was found to be a reasonable choice in the compromise between preserving information and feature vector size. All the parameters of the STFT are summarized in the following table.

Signal Length	Window Size	Overlap	Gaussian	Frequencies (Cutoff)
$N_x = 1600$	$W_l = 128$	$W_o = 64$	$\sigma = 1$	$N_{FFT} = 64(32)$

Resulting in a feature vector of length 768 per column profile

The selected STFT parameters yield the following dimensioned spectrogram, where

each row corresponds to a frequency in the signal and column entries to positions:

$$\# \text{position bins} \times \# \text{frequency bins} = \left\lfloor \frac{N_x - W_o}{W_l - W_o} \right\rfloor \times N_{FFT} = 24 \times 32 \quad (8.3.4)$$

The position bins, in ascending order, correspond to locations from top to bottom of the corresponding column profile. In Figure 8.6, three mean spectrograms of three different column profiles are visualized. Note that the frequencies increase in the images from left to right. This is explained by the perspective distortion found in the images³.

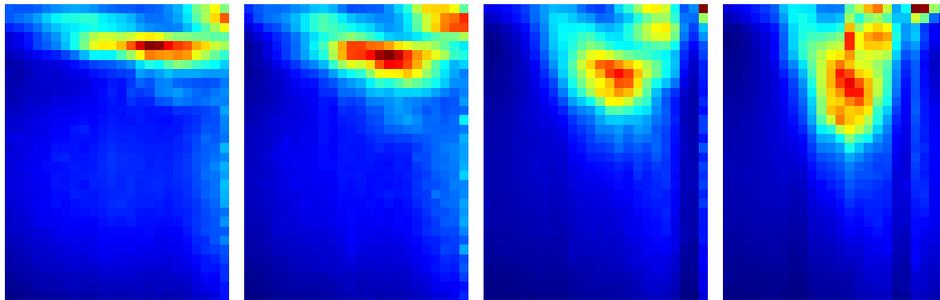


Fig. 8.6.: Mean spectrograms of different column profiles. From left to right: column profile 2, 6, 10 and 12 are shown. The y-axis corresponds to frequencies (from top to bottom), while the x-axis indicated the location in the profile.

8.3.2. Dimension Reduction

In the order to reduce the high-dimensional features, of which there are 9,216 for the first data set (12 column profiles) and 13,824 for the second data set (18 column profiles), Principal Component Analysis (PCA) is applied (see Section 2.2.1). Similar to the *eigenface* method [103], proposed for face recognition, PCA is used to decorrelate and reduce the feature dimensions. Eigenvectors (in this case called *eigenfeatures*) are computed to project the image onto a lower dimensional space. The first resulting eigenfeatures are shown in Figure 8.7. Any feature (spectrogram) can be considered to be a linear combination of these eigenfeatures. The eigenfeatures are sorted according to their contribution toward explaining the variance in data (eigenvalue).

8.3.3. Feature Selection with Random Forest

The feature selection described in [14] was applied to assess the importance of each feature. The *Mean Gini Decrease* method was employed to do so. The variable im-

³Efforts made to *rectify* the images as a preprocessing step caused various types of interpolation artifacts, which could severely bias the feature extraction.

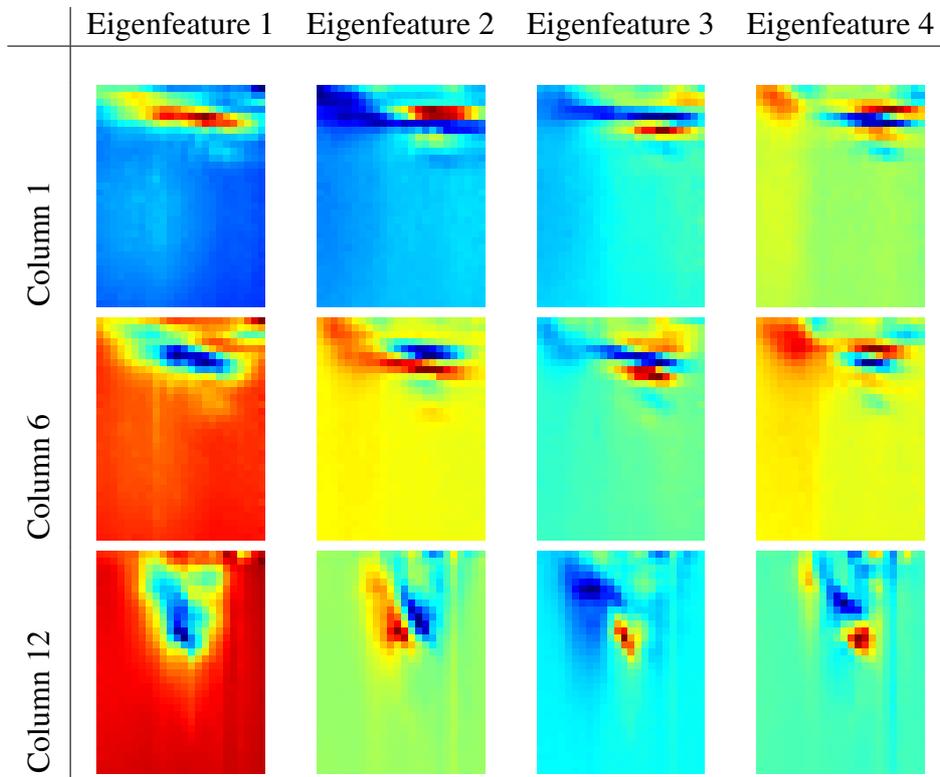


Fig. 8.7.: First four eigenfeatures for column profile one, six and twelve.

portance score is visualized in Figure 8.8a. A Gaussian filter ($\sigma = 1$) was applied to spatially smooth the importance map. 32 different random forests with 200 trees each were used to assess the variable importance, and the 32 resulting importance maps were averaged. In Figure 8.8b, the two most important features selected by the random forest criteria were plotted against one another. The results provide evidence that the problem of distinguishing between the two classes (red and blue) is almost linearly separable. This finding facilitates the use of linear classifiers as explained in the next section.

8.4. Classification

In order to classify each example, two different linear methods were examined. At first, Linear Discriminant Analysis (LDA), sometimes also called Fisher's Discriminant Analysis, was applied. LDA approaches the problem by assuming that the conditional probability density functions $P(\mathbf{x}|y = \text{error})$ and $P(\mathbf{x}|y = \text{normal})$ are both normally distributed for features \mathbf{x} . In contrast to Quadratic Discriminant Analysis (QDA), it is assumed that both classes y share the same covariance matrix. The Gaussian assumption causes linear decision boundaries in the feature space and renders LDA a widely used

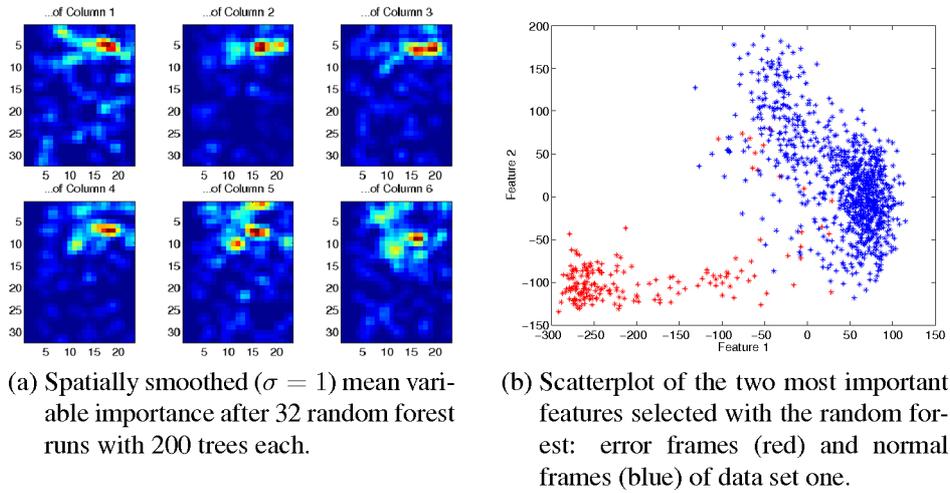


Fig. 8.8.: Random forest feature selection

Feature Selection	Accuracy	False Positive Rate
None	0.997	0.014
PCA Threshold	0.950	0.153
Random Forest	0.989	0.038

Table 8.1.: 10×10 -fold cross-validation accuracy and false positive rate of the linear support vector machine with and without prior feature selection (25 most important features from random forest/ PCA thresholding)

techniques for linear separable problems.

Linear support vector machines [96] (as described in 2.3.4) are also used. The results obtained by linear SVM are quite similar, but SVM still outperforms the LDA approach. This result indicates that the Gaussian assumption underlying LDA is not fulfilled and LDA results are therefore omitted.

To illustrate the method used to classify frames as being either from an error or from normal sequences, the basic concept of linear support vector machines is reviewed in Section 2.3.4 on page 19.

Table 8.1 shows the classification results achieved using linear support vector machines on data set one. Using no feature selection at all yields the best performance. Reducing the feature set by using only the 25 top scored features by random forest or PCA decreases recognition rates. However, the classification performance after the reduction of features to about three out of thousand of its previous size is remarkable; making feature selection an interesting choice concerning real-time application.

8.5. Online Visualization of Process State

Ideally, a trained classification system should close the control circuit, meaning that the feeder parameters are adjusted on the basis of the current estimate of the process state in relation to a stable state, which was identified during the training phase of the classifier. Technically this requires an interface that is tunable to the machine's feeder. However, the parameters can only be actuated by hand and no computerized interface exists so far. To this end, an online visualization was developed that assists a human operator in readjusting the feeder. On the basis of dimension reduction, the high-dimensional feature space can be mapped onto a two-dimensional grid, indicating the process' state by an underlying heat map. This heat map is generated on the basis of self-organizing maps (see section 8.5.1).

Data set two was designed to capture the dynamics of the feeder process. As mentioned above, different controllers at the feeder can be adjusted manually. The variation of up to three different parameters is examined in 58 sequence experiments, where the parameter of interest was varied continuously from a start state to an end state (for example from almost no air to full pressure at the corner air nozzle). The variation was designed to end when approximately 100 frames (=100 sheets) were recorded.

In Figure 8.9a an example is plotted. The three dimensional subspace of decorrelated features (computed using PCA) is shown, where each axis corresponds to one of the first three principal component directions. The blue curve illustrates the variation of a main, single factor: strength of the air nozzle at the front corner location. Both a normal sequence - in which all parameters are set to default values (green) - and a double sheet sequence are also plotted as a reference (red). In order to see the temporal progress of the feature points in that space, the frame number of the blue curve is added. Starting next to the green sequence, the blue curve progresses to another region in the feature space (starting at frame 13 and ending up at frame 25).

Self-organizing maps (see Section 8.5.1) were applied to map the feature space spanned by the spectrogram features onto a two-dimensional grid. The basic procedure is as follows:

1. Computation of the principal components of the training data. The PCA is done on each column profile separately (to avoid a high memory load).
2. Training of a self-organizing map, where the first m principal components from each column profile serve as the input space. For $m = 5$ this makes up 90-dimensional space.
3. Use the label information of the sequence data to associate each labeled sample with a best matching neuron and count how often a neuron was selected by the given class. This results in a count map per class.

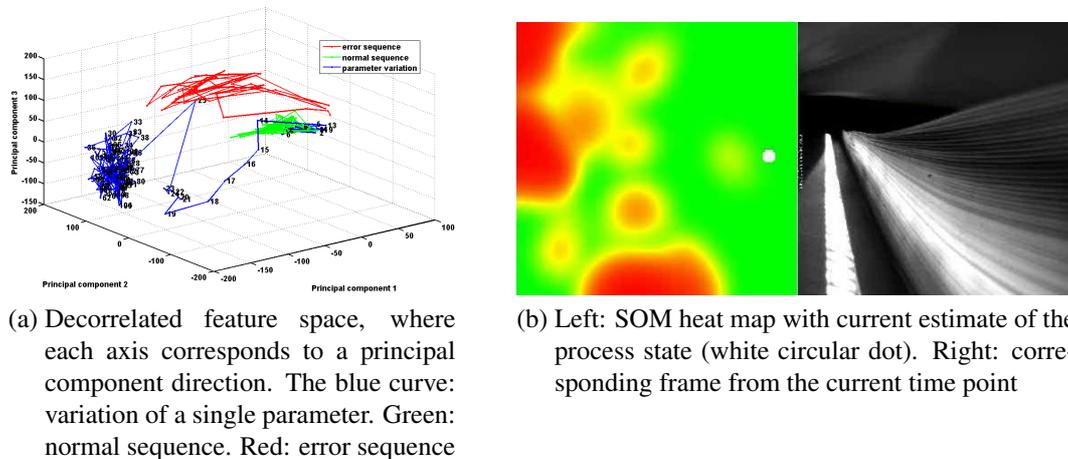


Fig. 8.9.: Variation of feeder parameters visualized in the feature space and on the self-organizing map

4. The counts of the error class are put into the red-channel of an RGB image where the normal counts are put into the green channel. Smoothing and appropriate rescaling of the color values yield a color coded image (see Figure 8.9b for an example).

Once the color coded image is computed, it is used to indicate the current state of the process by overlaying the present best matching neuron as a white dot. An offline demonstration tool is implemented in Matlab, where different sequences can be loaded and visualized. The tool shows the *learned* SOM heat map and the current winner neuron next to the current image frame. A more detailed description of self-organizing maps can be found in section 8.5.1.

8.5.1. Self-Organizing Maps

A self-organizing map (SOM) is a special type of artificial neural network related to vector quantization. Unsupervised learning of the network results in a low-dimensional representation of the input space called the *map*. Self-organizing maps use a neighborhood function to preserve the topological properties of the input space. This discretized representation of the input space is typically a two-dimensional grid of nodes/neurons which is useful for visualizing high-dimensional data. The model was first described as an artificial neural network by Kohonen [58, 59], and is sometimes referred to as a *Kohonen map*.

Self-organizing maps operate – like most classifiers – in two modes: training and prediction (in this context also called *mapping*). Training builds the map in a competitive

process using input examples. Predicting automatically classifies a new input vector. A self-organizing map consists of components called neurons. A weight vector of the same dimension as the input data vectors is associated with each neuron. The neurons are also located in a low dimensional map space (typically two dimensional). The usual arrangement of neurons is a regular spacing rectangular grid. The self-organizing map describes a mapping from the higher dimensional input space to the lower dimensional map space. The procedure for placing a vector from an input space onto the map is to find the neuron with the closest weight vector called the best matching unit (BMU).

As mentioned above, the self-organizing map is built in an unsupervised fashion. Given a set \mathbf{S} of training stimuli characterized by their d -dimensional feature vectors:

$$\mathbf{S} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^d\} \quad \text{for } i = 1, \dots, n \quad (8.5.1)$$

and a map topology \mathbf{N} containing each neuron's weight vector and its position in the net

$$\mathbf{N} = \{n_i = (\mathbf{w}_i, p_i) | \mathbf{w}_i \in \mathbb{R}^d, p_i \in \mathbb{N} \times \mathbb{N}\} \quad \text{for } i = 1, \dots, n \quad (8.5.2)$$

the adaptation (or learning) phase consist of several steps. From the set of training stimuli \mathbf{S} , a sample \mathbf{x}_j^t is drawn randomly. Since the learning of the SOM is an iterative procedure, t refers to the current time point. Now, the best matching neuron n_i^t is found by comparing the neuron's weights with the current stimulus \mathbf{x}_j^t . The neuron with the smallest distance to \mathbf{x}_j^t is selected by a given metric $d(\cdot, \cdot)$. A typically choice for the metric is the Euclidean distance:

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (8.5.3)$$

The location p_j of the winning neuron n_i^t is called the excitation center of the stimulus \mathbf{x}_j^t . All neurons close to that excitation are determined by:

$$\mathbf{E}^t = \{n_i = (\mathbf{w}_i, p_i) | d(p_j, p_i) < \delta^t\}, \quad (8.5.4)$$

where δ^t is a time-dependent threshold on the radius surrounding n_i^t . A learning step is applied to all neurons in the set \mathbf{E}^t . The weights of each neuron change towards the current input stimuli by:

$$\mathbf{w}_i^{t+1} \leftarrow \mathbf{w}_i^t + \epsilon^t \cdot h^t \cdot (\mathbf{x}_j^t - \mathbf{w}_i^t), \quad (8.5.5)$$

where ϵ^t is the learning rate and h^t spatial neighborhood function (typically Gaussian). In order to achieve stable result, the learning rate has to decrease over time. Also the neighborhood size involved in updating the weights of the neurons has to shrink. The whole process is iterated until the maximal iteration t_{\max} is reached. For the experiments, the SOM toolbox [106] is used. The learning rate is set to:

$$\epsilon^t = \frac{\epsilon_0}{1 + \frac{100t}{t_{\max}}}, \text{ with } t_{\max} = 4 \cdot n, \quad (8.5.6)$$

where n is the number of training examples. The neighborhood function and threshold h_t, δ_t (which defines the circular mask) is chosen to be a Gaussian with default parameters provided by the toolbox. The training comprises two phases: rough training with large initial neighborhood radius and initial learning rate $\epsilon_0 = 0.5$, and fine tuning with a small radius and initial learning rate of $\epsilon_0 = 0.05$. The map size is set to 31×31 neurons and is initialized by the principle components method [60].

8.6. Discussion

The dynamics of the sheet feeder process were studied here. Efficient feature extraction on the basis of Fourier transform were proposed and presented good results together with a linear classification methodology. Random forests were used to score the feature importance to select discriminant features. Two data sets were recorded to study the static and varying feeder parameters. In the first, error frames can be reliably distinguished from normal ones. The latter is used to characterize the parameter variation of the feeder by means of feature response. A dimension reduction method (PCA) is used to reduce the dimensionality of the feature vectors and to visualize parameter variation in the decorrelated feature space. Finally the SOM framework was applied to map the still high-dimensional essence of the dynamics down to a two-dimensional grid of neurons. This is a prerequisite for semi-automated process control, integrated into the printing machine, where a human operator reacts to the online visualized process state by adjusting feeder parameters.

However, the question remains as to whether the learned models (either linear support vector machine or self-organizing map) generalize to different (printing) substrates. Because it is a multi-purpose tool, a printing machine is able to handle a variety of paper, cardboard and transparent films. Cardboard results in a lower-frequency response due to its thickness and less dynamics as a result of its rigidity. The current camera and illumination setup deals with problems in underexposure and reflections when printing on transparent films. As a proof-of-concept, the experiments examined in this thesis were accomplished on standard paper with a height a 135 g/m^2 .

An important aspect for future considerations is autonomous generation of training data. Signals (such as feeder parameters, speed and error state) and images could be recorded under typical workaday conditions and a fixed camera assembly would guarantee reproducibility of the acquisition process. This demands a more computerized machine interface and a computer dedicated to this purpose.

9. Conclusion

In Chapter 3, a framework and an interactive implementation tool is introduced and applications to biomedical imagery are evaluated. The learning procedure aids domain experts in interactively classifying and segmenting images in an explorative and convenient fashion. Results for autonomous cell counting show only little deviation from manual counts and can easily be applied to bigger data sets, where human labor is virtually impossible. In addition, it is shown that the DNA channel is sufficient to segment and classify mitotic cells, thus avoiding the need for an additional mitosis markers.

The framework was successfully applied to retina blood-vessel segmentation, yielding an area under the curve of $A_z = 0.94$, which lies in the range of previously published results for that data source. Hence, the learning approach with a general feature set is applicable to fine structures as exist in the retina images. The incorporation of multi-scale features has shown to be a beneficial ingredient in coping with structures of varying size (blood vessels). That the framework inherently supports multiple classes has been shown in the application to retina cross-sections.

Highly optimized implementations of the underlying algorithms made it possible to deploy the whole approach in a unifying tool kit that comes along with a graphical user interface; it is designed to be easily usable, even for novices in the field of image processing. Convenient interfaces and object-oriented programming ensure maintainability and short development cycles for further improvements. The tool supports up to four-dimensional multi-spectral data; a file specification on the basis of the HDF5-format was introduced to jointly save raw data, labels, processing results and meta information. The approach combines a convenient user interface with a state-of-the-art classifier. The generalization ability of the classifier and its intrinsic support for variable selection has been exploited for several segmentation tasks. The approach is robust since the user only needs to input a few labels, and the classifier is trained on a set of generic features that work across many types of images. The learning procedure was applied without explicitly taking advantage of domain knowledge; however, for some data domains, specific application dependent post-processing was applied, e.g. seeded watershed transform or shortest path reasoning.

In an application to adult neurogenesis, object properties from various channels are set in relation. Particularly, the locations of proliferating cells within the molecular layer of the dentate gyrus are identified. Executed with the intuitive learning interface, the algorithm provides the user with segmentation that has satisfactory accuracy. Due to the supervised learning concept used in the initial step for the segmentation, this method

is highly adaptive to tackling imaging artifacts such as inhomogeneous illumination in image stacks without employing preprocessing.

Using this framework, the classifier is able to predict the boundary probability of tumor-stroma borders. A global optimization technique is applied to obtain the course of the boundary as the best path based on these boundary cues. Results demonstrate excellent performance for images coming from early time points in the underlying biological experiment. For more complex formations of the stroma-tumor border, usually occurring at later time points, the algorithm yields satisfactory results, as is witnessed by visual inspection.

Apart from the interactive learning approach, a modular and flexible microscopy-based RNAi screening platform is described for the investigation of host factors involved in virus-host interaction. The procedure was shown to be suitable for robust and sensitive detection of host cell factors involved in HIV-1 replication using different cell culture formats. Dedicated image processing procedures were developed to process the very large amounts of high-throughput image data in reasonable time. Image quality control mechanisms ensure the reliable and relevance of the autonomously generated readouts by excluding erroneous image regions originating from contamination and imaging artifacts. One-class learning techniques on top of similar feature descriptors (as used in the interactive framework) have been successfully applied for this purpose and extended to a patch-based recognition system.

For the segmentation of copper wires spread in tag-labels in an industrial collaboration, results of the proposed methodology attest: discriminant features, together with a non-linear classifier, achieved a more accurate and robust performance compared to previously used thresholding approaches. In particular, light reflexes can be handled without problem-specific measures and are intrinsically learned by the classifier. The random forests present very good generalization performance, which is suggested by ROC curve analysis.

The dynamics of a sheet feeder process were also studied within this thesis. Global features extracted on the basis of short-term Fourier transform were proposed and showed good results when taken together with a linear classification methodology. Random forests are used to score the feature importance in order to select the most discriminant feature subset. Two data sets were recorded by the author to study static and varying feeder parameters. In the first, error frames can reliably be distinguished from normal ones with a test accuracy of 0.997. The latter is used to characterize the parameter variation of the feeder by means of dimension reduction. Principal component analysis is used to reduce the dimensionality of the feature vectors and to visualize parameter variation in the decorrelated feature space. Finally the self-organizing map framework was applied to map the still high-dimensional essence of the dynamics onto a two-dimensional grid. This is a prerequisite for semi-automated process control integrated into the printing machine, where human operators are enabled to react to the online

visualized process state by manually adjusting feeder parameters.

10. Outlook

The proposed interactive framework is limited to local brightness, color and texture, and is not designed to capture global context. Indeed it can easily be utilized as a first stage tool to solve advanced problems on top of its output. A very interesting point in the experiments is that the few user labels are more informative than the abundant labels from the ground truth, which leads to superior classification performance. This raises the question of how the classifier can actively propose regions where the system needs more labels from the user. One idea is to use the out-of-bag samples in the training phase of the random forest classifier to compute a *certainty map*. This enables the system to ask for labels that it deems informative. For active learning to be useful in practice, it is vital that the learner responds swiftly. This may require incremental learning strategies that are able to gradually adapt their model to new incoming labels. Two incremental learning algorithms were implemented¹ and are currently being examined. The first is based on *LaSVM* [8]; the latter extends the random forest for online learning [90].

The interactive learning framework has proven to be a useful tool in several applications. The main reason for this was the deployment of a tool that bundles the learning and segmentation methodology into a graphical user interface. A wide scope of improvements and extensions are planned to be integrated in the future. A well-engineered visualization of three-dimensional volume data will further facilitate the labeling process. One-class learning showed very good results in image quality control and is a promising extension to preprocessing general image data. Of particular interest for the collaboration partners from industry and biology is the organization of meta data belonging to the data items in an persistent and integrated manner. This will be added using the HDF5 file format specification. Huge data sets, which cannot even be loaded into main memory of standard work stations, need a concept for distributed processing. Remote objects and the *message passing interface (MPI)* are currently examined to extend the framework to cluster computing. Python supports both techniques and promising pilots test were carried out.

The successful analysis of data from the hippocampus of mouse brains opens up several new directions in which to proceed. The framework can be applied to other experiments that are currently being carried out at the laboratory of Ana-Martin Vilalba, concerning similar questions of the adult neurogenesis in the subventricular zone.

The calculation of the optimal path based on a trained classifier model enabled the

¹by Nathan Hüsken

reliable quantification of blood vessels in tumor tissue. The problem of short cuts is still a challenging research topic. The interface refinement tool enabled the user to provide attractor or blocking guides to refine the border probability. However, these corrections steps could also be used to further improve the learner.

Standard image processing on a large scale, combined with machine learning to integrate image quality control, are key components in the Viroquant RNi platform. Future extensions will include automated classification of images based on cell morphologies, so that knockdown phenotypes can be classified in more detail and better correlated with biological processes. First steps have been taken by integrating the screening results into the iChip project. Comparative analysis using other viruses on the same platform will elucidate commonly used cellular host factors exploited by different viruses, which may serve as novel drug targets for broad spectrum antivirals. Finally, efforts at standardization of RNAi screening procedures are made and offer experience with this platform as a robust basis on which to build new systems. Moreover, correlative microscopy will benefit immensely from learning techniques to bridge the gap between conventional wide field microscopy to high resolution imaging techniques.

Segmentation of copper wire structures for product authentication has demonstrated superior performance to other examined methods. However, the bottleneck in runtime performance is identified to be the prediction time of the random forest. The feature computation is done in less than 0.2 sec on a conventional notebook, while prediction time is about 1.5 sec (random forest with 100 trees) on the same machine. An obvious solution to this problem is to use an appropriate multi-core CPU, where the prediction can be easily distributed due to fact that each tree in the forest is independent from the others. An alternative is to stop the prediction process earlier when a first fraction of trees agree to some extent.

In this thesis several video sequence data sets were acquired to investigate the process state of the sheet-feeding process. A novel classification based approach based on global frequency estimates was proposed for this purpose. An unsupervised online visualization of the process state was also suggested. However, the question remains as to whether the learned models (either linear support vector machine or self-organizing map) can be generalized to different (printing) substrates. Because it is a multi-purpose tool, a printing machine is able to handle a variety of paper, cardboard, and transparent films. Cardboard results in lower frequencies due to its thickness and less dynamics as a result of its rigidity. The current camera and illumination setup will be confronted with problems in underexposure and reflections when printing on transparent films. As a proof-of-concept, the experiments examined in this thesis were accomplished on standard paper. An important aspect for future consideration is autonomous generation of training data. Machine signals and images could be recorded under typical workaday conditions and a fixed camera assembly would guarantee reproducibility of the acquisition process. This requires a more computerized machine interface to the printing

press.

List of Figures

2.1.	Illustration of the feature space partitioning with classification and regression trees (CARTs). Picture taken from [7]	16
3.1.	The Ilastik GUI: Biomedical use case of the implemented user interface. Left: A domain expert labels background, cell nuclei, cytoplasm and a phenotype of interest (red label color). Right: After selecting color features and the interactive classification, a real-time update of the current classifiers output is overlaid. The interactively trained classifier can then be applied to more images.	30
3.2.	The STARE database. Rows: Two examples. From left to right: Original RGB image, ground truth given by first observer, ground truth given by second observer.	31
3.3.	Cell nuclei of normal and mitotic cell nuclei and overlaid user labels. Blue: Background; Red: mitotic cell labels; Green: normal cell nuclei	31
3.4.	Segmentation results after leave-one-out cross-validation. First row: pathological case, second row: normal case. From left to right: Raw image with overlaid user labels (red: non-vessel/ green: vessel), resulting probability map after leave-one-out cross-validation (training with user labels of the remaining 19 images), segmentation at threshold level $t = 0.5$, ground truth given by first observer.	32
3.5.	ROC analysis results after leave-one-out cross-validation	33
3.6.	Images from the Human HT29 colon cancer data set	34
4.1.	Clockwise, starting with RGB overlay of the 3 data channels, the DAPI channel with a granule cell layer, BrdU-positive cells and the DCX channel	39
4.2.	The proposed workflow: For each voxel of the original image data, a feature vector is computed. Trained with the labeled samples, the classifier outputs a probability as either belonging to the background or to a specified object class for each unseen voxel. The resulting probability map provides both the topological surface for the segmentation and the marker extraction [40]. Results can be corrected by re- or additive labeling	41

4.3.	Post-processing workflow resulting in 3D distance measures of BrdU-positive cells to the inner border of the granule cell layer and DCX positive negative classification [40].	44
4.4.	Extraction of the inner region inside the granule cell layer. Clockwise, starting with the original cell layer segmentation, connected components of the convex hull, maxima of distance transform that are not on the image border, reconstruction of the inner region	45
4.5.	DAPI segmentation results: Each column shows the original data and segmentation results for a different data set. The first row displays one slice out of the original data volume, the second row shows the layer segmentation result, and the third row provides the extracted inner region of the layer.	48
4.6.	One slice out of the original data with the labels obtained by visual inspection (red) and the algorithm's segmentation result (blue) as overlay	49
5.1.	Example fluorescence image. Blue staining: cell nuclei, red staining: blood vessels, green staining: keratin, continuous red line: manually labeled tumor-stroma border, the tumor is located in the upper region . .	56
5.2.	Generation of the boundary indicator map. The left image shows user labels from biologists for boundary and non-boundary pixels. A typical boundary indication map predicted by the trained classifier (right image) [53].	57
5.3.	Variable importance for the used features	58
5.4.	Segmentation of blood vessels by automated thresholding	60
5.5.	User refinement using attractors (left) and barriers (right)	60
5.6.	Tumor-stroma border result for an image from a later time point (heavily invaded tumor): green: reference border, cyan: extracted border. The extracted path (cyan) is taking several short cuts.	61
5.7.	Resulting tumor-stroma borders (cyan) overlaid onto the 24 microscope images	63
6.1.	Workflow diagram for the primary screen.	70
6.2.	Illustration of voronoi-like watershed segmentation	72
6.3.	Illustration of the watershed algorithm. Left: Original watershed formulation: water rises from all local minima, Right: Seeded watershed transform: seeds (orange) are used to control the segmentation process]	73
6.4.	Comparisons between the non-silencing control siRNAs and CD4 positive controls	74
6.5.	Example of a contaminant correctly classified as such (red) and excluded from the result.	77

6.6.	The user interface (left) enables users to browse the image files according to the plate coordinates. An example of using the interface is shown (right) to check for mitotic abnormality.	78
6.7.	Sketch of the parallel pipeline	79
7.1.	The proposed methodology based on supervised learning	82
7.2.	Three examples from the labeling process. Copper wires (green) and background (red)	83
7.3.	Feature response for the selected feature set. A center section of the original-sized image is enlarged for better visibility.	84
7.4.	Evaluation of the proposed method in comparison to thresholding. . . .	86
8.1.	The sheet-feeding processes	88
8.2.	The camera setup: assembly of the camera, diode array and optical lens at the sheet feeder	90
8.3.	Test pattern image with the camera and the newly developed optic lens. Length of both the black and white box is 50 mm. Each line (on the right part) has a width of 1mm.	91
8.4.	Scheimpflug principle. On the left the usual camera setup is depicted. The Scheimpflug principle can be utilized to align the plane of focus with the object to achieve a focused projection.	92
8.5.	Feature extraction. From left to right: Original image, selected columns in red, and the resulting gray-value profiles	93
8.6.	Mean spectrograms of different column profiles. From left to right: column profile 2, 6, 10 and 12 are shown. The y-axis corresponds to frequencies (from top to bottom), while the x-axis indicated the location in the profile.	94
8.7.	First four eigenfeatures for column profile one, six and twelve.	95
8.8.	Random forest feature selection	96
8.9.	Variation of feeder parameters visualized in the feature space and on the self-organizing map	98

Bibliography

- [1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:641–647, 1994.
- [2] W.M. Ahmed, S.J. Leavesley, B. Rajwa, M.N. Ayyaz, A. Ghafoor, and J.P. Robinson. State of the art in information extraction and quantitative analysis for multi-modality biomolecular imaging. *Proceedings of the IEEE*, 96:512–531, 2008.
- [3] Björn Andres, Ullrich Köthe, Moritz Helmstaedter, Winfried Denk, and Fred A. Hamprecht. Segmentation of sbfsem volume data of neural tissue by hierarchical classification. In *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pages 142–152, Berlin, Heidelberg, 2008. Springer-Verlag.
- [4] A. J. Baddeley. An error metric for binary images. In *International Workshop on Robust Computer Vision*, 1992.
- [5] M. Beller, H. Gemmeke, T.O. Müller, and R. Stotzka. Semi-supervised segmentation of images. In *Iranian Conference on Machine Vision and Image Processing*, 2005.
- [6] S. Beucher. The watershed transformation applied to image segmentation. In *Scanning Microscopy International*, pages 299–314, 1992.
- [7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer,, 2006.
- [8] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619, September 2005.
- [9] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.
- [10] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *ICCV'01*, 1:105–112, 2001.
- [11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.

- [12] A. L. Brass, D. M. Dykxhoorn, Y. Benita, N. Yan, A. Engelman, R. J. Xavier, J. Lieberman, and S. J. Elledge. Identification of host proteins required for HIV infection through a functional genomic screen. *Science*, 319:921–926, Feb 2008.
- [13] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. Republished by CRC Press, 1984.
- [14] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
- [15] Kathleen Börner, Johannes Hermle, Christoph Sommer, Nigel P. Brown, Bettina Knapp, Bärbel Glass, Julian Kunkel, Gloria Torralba, Jürgen Reymann, Nina Beil, Jürgen Beneke, Rainer Pepperkok, Reinhard Schneider, Thomas Ludwig, Michael Hausmann, Fred Hamprecht, Holger Erfle, Lars Kaderali, Hans-Georg Kräusslich, and Maik J. Lehmann. From experimental setup to bioinformatics: An rnai screening platform to identify host factors involved in hiv-1 replication. *Biotechnology Journal*, 5(1):39–49, 2010.
- [16] F. D. Bushman, N. Malani, J. Fernandes, I. D’Orso, G. Cagney, T. L. Diamond, H. Zhou, D. J. Hazuda, A. S. Espeseth, R. König, S. Bandyopadhyay, T. Ideker, S. P. Goff, N. J. Krogan, A. D. Frankel, J. A. Young, and S. K. Chanda. Host cell factors in HIV replication: meta-analysis of genome-wide studies. *PLoS Pathog.*, 5:e1000437, May 2009.
- [17] 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.*, 7:R100, 2006.
- [18] C. A. Carter and L. S. Ehrlich. Cell biology of HIV-1 infection of macrophages. *Annu. Rev. Microbiol.*, 62:425–443, 2008.
- [19] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [20] 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*, 8(3):263–269, 1989.
- [21] Sharat Chikkerur, Alexander N. Cartwright, and Venu Govindaraju. Fingerprint enhancement using stft analysis. *Pattern Recognition*, 40:198–211, 2007.
- [22] L.P. Coelho, A. Shariff, and R. F. Murphy. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 518–521, 2009.

- [23] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 886–893, 2005.
- [24] P. J. Davis, E. A. Kosmacek, Y. Sun, F. Ianzini, and M. A. Mackey. The large-scale digital cell analysis system: an open system for nonperturbing live cell imaging. *J Microsc*, 228:296–308, Dec 2007.
- [25] Edsger. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [26] H. Erfle, B. Neumann, U. Liebel, P. Rogers, M. Held, T. Walter, J. Ellenberg, and R. Pepperkok. Reverse transfection on cell arrays for high content screening microscopy. *Nat Protoc*, 2:392–399, 2007.
- [27] H. Erfle, B. Neumann, P. Rogers, J. Bulkescher, J. Ellenberg, and R. Pepperkok. Work flow for multiplexing siRNA assays by solid-phase reverse transfection in multiwell plates. *J Biomol Screen*, 13:575–580, Aug 2008.
- [28] A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. de A. Lotufo. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60:233 – 260, 1998.
- [29] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.
- [30] G. Friedland, K. Jantz, and R. Rojas. SIOX: Simple interactive object extraction in still images. In *Seventh IEEE International Symposium on Multimedia*, 2005.
- [31] E. Fuchs and E. Gould. In vivo neurogenesis in the adult brain: regulation and functional implications. *European Journal of Neuroscience*, 12:2211, 2000.
- [32] Thomas J. Fuchs, Tilman Lange, Peter J. Wild, Holger Moch, and Joachim M. Buhmann. Weakly supervised cell nuclei detection and segmentation on tissue microarrays of renal clear cell carcinoma. In *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pages 173–182, 2008.
- [33] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *Computer Vision and Pattern Recognition*, pages 1022–1029, 2009.
- [34] Y. Gavrieli, Y. Sherman, and S. A. Ben-Sasson. Identification of programmed cell death in situ via specific labeling of nuclear DNA fragmentation. *Journal of Cell Biology*, 119:493–501, Nov 1992.

- [35] Robert C Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Y. H. Yang, and Jianhua Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.
- [36] IDS Imaging Development Systems GmbH. *Manual: uEye 2250-M/C*.
- [37] Visage Imaging GmbH. *Amira 5: User's Guide*.
- [38] S. P. Goff. Knockdown screens to knockout HIV-1. *Cell*, 135:417–420, Oct 2008.
- [39] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51:271–279, 1989.
- [40] Jörg Greis. Semi-automatic analysis of highinformation-content neurobiological image data. Master's thesis, Faculty of Physics and Astronomy, University of Heidelberg, 2009.
- [41] P. R. Gudla, K. Nandy, J. Collins, K. J. Meaburn, T. Misteli, and S. J. Lockett. A high-throughput system for segmenting nuclei using multiscale techniques. *Cytometry Part*, 73:451–466, 2008.
- [42] H. Hajer, J. abd Kamel and E. Noureddine. Blood vessels segmentation in retina image using mathematical morphology and the stft analysis. *Information and Communication Technologies*, 1:1130–1134, 2006.
- [43] S. Han, W. Tao, D. Wang, X. C. Tai, and X. Wu. Image segmentation based on GrabCut framework integrating multiscale nonlinear structure tensor. *IEEE Trans Image Process*, 18:2289–2302, Oct 2009.
- [44] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Science+Business Media, 2001.
- [45] Lei He, Zhigang Peng, Bryan Everding, Xun Wang, Chia Y. Han, Kenneth L. Weiss, and William G. Wee. A comparative study of deformable contour methods on medical image segmentation. *Image and Vision Computing*, 26:141–163, 2007.

- [46] 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*, 19(3):203–210, 2000.
- [47] B. Jaehne. *Digital Image Processing*. Springer-Verlag, 2006.
- [48] S. Jbabdi, P. Bellec, R. Toro, J. Daunizeau, M. Pélégriani-Issac, and H. Benali. Accurate anisotropic fast marching for diffusion-based geodesic tractography. *Journal of Biomedical Imaging*, 2008:1–12, 1.
- [49] 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*, 25(1):131–137, 2003.
- [50] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter. CellProfiler Analyst: data exploration and analysis software for complex image-based screens. *BMC Bioinformatics*, 9:482, 2008.
- [51] Thouis R. Jones, Anne Carpenter, and Polina Goll. Voronoi-based segmentation of cells on image manifolds. In *ICCV Workshop on Computer Vision for Biomedical Image Applications*, pages 535–543, 2005.
- [52] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.
- [53] Stephan Kassemeyer. Quantification of tumour angiogenesis using pattern recognition. Master’s thesis, Faculty of Physics and Astronomy, University of Heidelberg, 2009.
- [54] G. Kempermann and D. Ehniger. Neurogenesis in the adult hippocampus. *Cell and Tissue Research*, 331:243, 2008.
- [55] Helmut Kipphan. *Handbook of Print Media: Technologies and Production Methods*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [56] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19:41, 1985.
- [57] R. König, Y. Zhou, D. Elleder, T. L. Diamond, G. M. Bonamy, J. T. Irelan, C. Y. Chiang, B. P. Tu, P. D. De Jesus, C. E. Lilley, S. Seidel, A. M. Opaluch, J. S. Caldwell, M. D. Weitzman, K. L. Kuhlen, S. Bandyopadhyay, T. Ideker, A. P. Orth, L. J. Miraglia, F. D. Bushman, J. A. Young, and S. K. Chanda. Global analysis of host-pathogen interactions that regulate early-stage HIV-1 replication. *Cell*, 135:49–60, Oct 2008.

- [58] T Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
- [59] T. Kohonen and T. Honkela. Kohonen network. *Scholarpedia*, 2(1):1568, 2007.
- [60] Teuvo Kohonen. *Self-Organizing Maps (2nd ed)(Springer Series in Information Sciences, 30)*. Springer, 2nd edition, 1995.
- [61] Vladimir Kolmogorov and Carsten Rother. Minimizing nonsubmodular functions with graph cuts—a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.
- [62] Ullrich Köthe. *VIGRA: The VIGRA Computer Vision Library*. Software available at <http://hci.iwr.uni-heidelberg.de/vigra/>.
- [63] Juliusz Kulikowski and M. Przytulska. Biomedical image segmentation based on morphological spectra. In *4th European Conference of the International Federation for Medical and Biological Engineering*, volume 22, pages 406–409, 2009.
- [64] C. Lawerenz, J. Eils, and R. Eils. iCHIP: Plattform für NGFN Datenintegration. *GenomXPress*, 2:7–10, 2007.
- [65] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 2169–2178, 2006.
- [66] S. Lefevre. Knowledge from markers in watershed segmentation. *Int. Conf. on Computer Analysis of Images and Patterns*, 4673:579–586, 2007.
- [67] Victor Lempitsky, Michael Verhoeck, J. Alison Noble, and Andrew Blake. Random forest classification for automatic delineation of myocardium in real-time 3d echocardiography. In *FIMH '09: Proceedings of the 5th International Conference on Functional Imaging and Modeling of the Heart*, pages 447–456, Berlin, Heidelberg, 2009. Springer-Verlag.
- [68] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE Transactions Pattern Analysis Machine Intelligence*, 28(9):1465–1479, 2006.
- [69] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *SIGGRAPH*, 2004.

- [70] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam. A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry Part A*, 56:23–36, 2003.
- [71] G. Lin, M. K. Chawla, K. Olson, C. A. Barnes, J. F. Guzowski, and C. Bjornsson. A multi-model approach to simultaneous segmentation and classification of heterogeneous populations of cell nuclei in 3d confocal microscope images. *Cytometry Part A*, 71:724–736, 2007.
- [72] G. Lin, M. K. Chawla, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam. Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. *Cytometry Part A*, 63:20–33, 2005.
- [73] J. Lindblad and E. Bengtsson. A comparison of methods for estimation of intensity nonuniformities in 2d and 3d microscope images of fluorescence stained cells. In *Proceedings of the 12th Scandinavian Conference on Image Analysis (SCIA)*, pages 264–271, 2001.
- [74] M. H. Malim and M. Emerman. HIV-1 accessory proteins—ensuring viral survival in a hostile environment. *Cell Host Microbe*, 3:388–398, Jun 2008.
- [75] N. Martin and Q. Sattentau. Cell-to-cell HIV-1 spread and its implications for immune evasion. *Current Opinion in HIV and AIDS*, 4:143–149, Mar 2009.
- [76] G. Ming and H. Song. Adult neurogenesis in the mammalian central nervous system. *Annual Review of Neuroscience*, 28:223, 2005.
- [77] Tobias Müller. personal communication, 2008.
- [78] J. Moffat, D. Grueneberg, X. Yang, S. Kim, A. Kloepfer, G. Hinkle, B. Piqani, T. Eisenhaure, B. Luo, and J. Grenier. A lentiviral rai library for human and mouse genes applied to an arrayed viral high-content screen. *Cell*, 124(6):1283 – 1298, 2006.
- [79] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. *Advances in Neural Information Processing Systems*, 19:985–992, 2006.
- [80] E. Mortensen and W Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349 – 384, 1998.
- [81] University of Kiel Germany Multimedia information processing group. The basic image algorithms c++ library (bias).

- [82] Laurent Najman and Michel Couprie. Watershed algorithms and contrast preservation. In *Lecture Notes in Computer Science*, pages 62–71. Springer, 2003.
- [83] University of Utah Scientific: Computing and Imaging (SCI) Institute. Seg3D: Volumetric Image Segmentation and Visualization. Scientific Computing and Imaging Institute (SCI).
- [84] Workshop on. High throughput sequence analysis tools and approaches with bioconductor. <http://www.bioconductor.org/workshops>, November 2009.
- [85] Nobuyuki Otsu. A threshold selection method from grey level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979.
- [86] M. Prudêncio and M. J. Lehmann. Illuminating the host - how RNAi screens shed light on host-pathogen interactions. *Biotechnol J*, 4:826–837, Jun 2009.
- [87] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.
- [88] Gordon Rios and Hongyuan Zha. Exploring support vector machines and random forests for spam detection. In *CEAS*, 2004.
- [89] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.
- [90] Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. On-line random forests. In *3rd IEEE ICCV Workshop on On-line Learning for Computer Vision*, 2009.
- [91] Nancy M. Salem, Sameh A. Salem, and Asoke K. Nandi. Segmentation of retinal blood vessels based on analysis of the hessian matrix and clustering algorithm. In *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007.
- [92] S. Salem, N. Salem, and A. Nandi. Segmentation of retinal blood vessels using a novel clustering algorithm (racal) with a partial supervision strategy. *Medical and Biological Engineering and Computing*, 45(3):261–273, 2007.
- [93] Jakob Santner, Markus Unger, Thomas Pock, Christian Leistner, Amir Saffari, and Horst Bischof. Interactive texture segmentation using random forests and total variation. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009. to appear.

- [94] Theodor Scheimpflug. Improved method and apparatus for the systematic alteration or distortion of plane pictures and images by means of lenses and mirrors for photography and for other purposes, 1904.
- [95] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, 2001.
- [96] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2001.
- [97] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. of the National Academy of Sciences of the USA*, 93:1591–1595, 1996.
- [98] Steven W. Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Newnes, 2002.
- [99] 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*, 25(9):1214–1222, 2006.
- [100] 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*, 23(4):501–509, 2004.
- [101] A. Statnikov, L. Wang, and C. F. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9:319, 2008.
- [102] John N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40:1528–1538, 1995.
- [103] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings Computer Vision and Pattern Recognition (CVPR)*, 1991.
- [104] H. van Praag, AF. Schinder, BR. Christie, N. Toni, TD. Palmer, and FH. Gage. Functional neurogenesis in the adult hippocampus. *Nature*, 415:1030, 2002.
- [105] Régis Vert and Jean-Philippe Vert. Consistency and convergence rates of one-class svms and related algorithms. *J. Mach. Learn. Res.*, 7:817–854, 2006.
- [106] Juha Vesanto, Johan Himberg, Esa Alhoniemi, and Juha Parhankangas. Som toolbox for matlab. Technical report, Helsinki University of Technology, 2000.

- [107] Silvia Vosseler. *Tumorangiogenese, Vaskularisierung und Invasion: Tumor-Stroma Interaktionen während der malignen Tumor-Progression*. PhD thesis, Universität Hohenheim, 2001.
- [108] C. Wahlby, I. M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson. Combining intensity, edge and shape information for 2d and 3d segmentation of cell nuclei in tissue sections. *Journal of Microscopy*, 215:67 – 76, 2004.
- [109] C. Wahlby, J. Lindblad, M. Vondrus, E. Bengtsson, and L. Bjorkesten. Algorithms for cytoplasm segmentation of fluorescence labelled cells. *Anal Cell Pathol*, 24:101–111, 2002.
- [110] T. Walter, D. W. Shattuck, R. Baldock, M. E. Bastin, A. E. Carpenter, S. Duce, J. Ellenberg, A. Fraser, N. Hamilton, S. Pieper, M. A. Ragan, J. E. Schneider, P. Tomancak, and J. K. Hériché. Visualization of image data from cells to organisms. *Nat. Methods*, 7:26–41, Mar 2010.
- [111] M. Wang, X. Zhou, F. Li, J. Huckins, R. W. King, and S. T. C. Wong. Novel cell segmentation and online svm for cell cycle phase identification in automated microscopy. *Bioinformatics*, 24:94–101, 2008.
- [112] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 37–44, Washington, DC, USA, 2006. IEEE Computer Society.
- [113] D. W. Young, A. Bender, J. Hoyt, E. McWhinnie, G. W. Chirn, C. Y. Tao, J. A. Tallarico, M. Labow, J. L. Jenkins, T. J. Mitchison, and Y. Feng. Integrating high-content screening and ligand-target prediction to identify mechanism of action. *Nat. Chem. Biol.*, 4:59–68, Jan 2008.
- [114] P. A. Yushkevich, J. Piven, H. Cody, S. Ho, J. C. Gee, and G. Gerig. User-guided level set segmentation of anatomical structures with ITK-SNAP. *Insight Journal*, 1, 2005. Special Issue on ISC/NA-MIC/MICCAI Workshop on Open-Source Software.
- [115] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27:236–239, 1984.
- [116] H. Zhou, M. Xu, Q. Huang, A. T. Gates, X. D. Zhang, J. C. Castle, E. Stec, M. Ferrer, B. Strulovici, D. J. Hazuda, and A. S. Espeseth. Genome-scale RNAi screen for host factors required for HIV replication. *Cell Host Microbe*, 4:495–504, Nov 2008.