UNIVERSITY OF CALIFORNIA, SAN DIEGO

Automated Cancer Detection and Drug Discovery : Two Biomedical Vision Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Computer Engineering)

by

Mayank Kabra

Committee in charge:

Professor Yoav Freund, Chair
Professor Truong Nguyen, Co-Chair
Professor Stephen Baird
Professor Pamela Cosman
Professor Gert Lanckriet

2011

The dissertation of Mayank Kabra is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2011

DEDICATION

To my parents.

EPIGRAPH

Chaos is inherent in compounded things. Strive on with diligence.

—Buddha

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to begin by thanking Yoav for I am in complete awe of his attitude and dedication towards research. For Yoav, research is not just to satisfy academic curiosity but an essential tool to improve lives. For this, he does not hesitate to take on challenging problems that he senses lurking at the horizon but are out of perception of other researchers. His enthusiasm, optimism and capacity to learn new things to solve these challenging problems have inspired me throughout the time spent working with him. He without hesitation learned to program FPGAs when it was clear that without FPGAs progress would be stalled. I hope that after being advised by Yoav for five years, I will be as open as him to new ideas and that I will not hesitate to dirty my hands when required. To solve pressing problems, he also introduced me to the fascinating field of modern biology, for which I'm very grateful. But more than any of these, I am most grateful to him for the many golden nuggets of advice on how to do good research that he has shared with me over the years.

While working on the pathology project, I had the wonderful opportunity to work with Steve Baird and Lucila Ohno-Machado. Steve is one of the most wonderful person I ever worked with. His affable personality and openness to discuss anything under the sun made me look forward to our meetings. I'm grateful for his patience in teaching me about pathology and his encouragement to work on challenging projects. I'm also grateful to Lucila who in her brief collaboration on the pathology project, critically shaped the project with her warmth, understanding, and sage advice.

The worms project came out of my internship at Broad Institute. For this, I would like to thank Anne Carpenter for the internship opportunity. The internship gave me a chance to experience first hand how research is conducted at a cutting-edge biology research lab. I would like to thank my collaborators Ray, Vebjorn, Mark, Adam, Kate and others at Broad who helped me during the internship. I would also like to thank Annie Lee Conery who was as a critical link between me and the biologists at MGH.

I'm also extremely grateful to my friends on the fourth floor: Nakul, Daniel,

Matus and Brian. The discussions with them, both technical and non-technical, helped me in thinking clearly about research and life as well. I'm especially grateful to Nakul for his time and effort into reading my awful drafts and presentations, which at times even I could not read. Their company at lunch took the edge of the monotony of eating in UCSD. The weekly dinners, where we tried new eating joints to discover great food, made the time spent here enjoyable. I'm also thankful to William, Sunsern, Evan, Matt and other members of Freund lab who over the years helped me out with many technical and non-technical problems.

As my room mate since the time we came to UCSD, I have much to thank Gaurav for. My discussions with him, where I tried to understand anything from innocuous of details to gravest of issues, were critical to my development during the time spent at UCSD. His enthusiasm for hindi movies, the trips we undertook and recommendations on many stuff made the time spend with him a pleasant experience. I am also much thankful to Ankit, Nikhil, Vijay, Himanshu, Vikram, Saumya, Rathinakumar and Anshuman for their vibrant and enthusiastic company over the years that made the time spent in San Diego memorable. Also, thanks to Manmohan and Manish who have been ever helpful since IIT days. I'm also thankful to Rohit for giving constant company for squash and gym.

Finally, I am most thankful to my parents who over the years have sacrificed the most and who also have been my first and most important teachers.

Chapter 4, in full, is a reprint of the material as it has been submitted to Arxiv in March 2010, Mayank Kabra; Annie L. Conery; Eyleen J. O'Rourke; Xin Xie; Vebjorn Ljosa; Thouis R. Jones; Frederick M. Ausubel; Gary Ruvkun; Anne E. Carpenter; Yoav Freund. The dissertation author was the primary investigator and author of this paper.

VITA AND PUBLICATIONS

2003	B. Tech. in Electrical Engineering , Indian Institute of Technology, Powai, Mumbai.
2003-2004	C.A.D Engineer, Intel Technologies, Bangalore.
2004-2005	Software Engineer, Infineon Technologies, Bangalore.
2008	M. S, Electrical Engineering University of California, San Diego.
2011	Ph. D. in Electrical Engineering, University of California, San Diego.

M. Kabra, S. Baird, S. Mahooti, J. Kim, L. Ohno-Machado and Y. Freund, *Prostate Cancer Detector for Pathology Images*, in submission.

M. Kabra, A. Conery, E. O'Rourke, X. Xie, V. Ljosa, T. Jones, F. Ausubel, G. Ruvkun, A. Carpenter and Y. Freund, *Towards automated high-throughput screening of* C. elegans *on agar*, Poster in International Symposium on Molecular Biology 2010.

Y. Freund, S. Dasgupta, M. Kabra and N. Verma, *Learning the structure of manifolds using random projections*, in NIPS 2007.

M. Kabra, S. Saha and B. Lin, *Fast Buffer Memory with Deterministic Packet Departures*, Hot Interconnects, Aug 2006.

ABSTRACT OF THE DISSERTATION

Automated Cancer Detection and Drug Discovery : Two Biomedical Vision Systems

by

Mayank Kabra

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California, San Diego, 2011

Professor Yoav Freund, Chair
Professor Truong Nguyen, Co-Chair

Statistical methods from machine learning have been key to the progress of computer vision in recent years. Use of machine learning has led to development of many successful vision applications such as face and pedestrian detectors. Along the same principles of using robust statistical methods, in this thesis we build systems for two biomedical imaging domains.

The first system detects cancer in prostate pathology images. Recent technological advancements have made it possible to commercially build whole-slide scanning microscopes that generate digital images of whole-slides at magnifications required for an effective clinical diagnosis. The availability of digital images

of pathology slides allows development of Computer Aided Diagnostic (CAD) tools that can improve pathologist's accuracy and efficiency in diagnosis. An automated screener can assist pathologists in diagnosing by suggesting suspicious locations. The screening tool can also reduce the bandwidth required for diagnosing remotely by transferring only the suspicious parts. To provide a base on which such CAD tools can be developed, we build a cancer detector for prostate needle core biopsies, which is one of the most frequently diagnosed tissue.

The second system analyzes High-Throughput Screening (HTS) images of *C. elegans* worms to identify their phenotype. HTS is a class of biological experiments where a large number of similar experiments are conducted to identify a small number of drugs or genes relevant to a biological process. Recently, researchers have started conducting HTS experiments using *C. elegans* in which the experimental output are images. To assist biologists in analyzing the large number of images generated by the HTS experiments on *C. elegans*, we develop a system that identifies a worm's phenotype. A preferred way of conducting such experiments is to image the worms in agar. The shadows cast by track marks left by the worms in agar appear similar to the worms which complicates segmenting images of worms. To reliably segment worms in such conditions, we develop a novel segmentation technique that uses multiple visual cues such as texture, contrast and shape to segment the worms. After segmenting the worms, our system also analyzes the fluorescent patterns inside the worms to identify their phenotype.

Chapter 1

Introduction

With digital imaging, computers are now as essential to microscopy as optical lenses. By automating capture of the image and control of the microscope's stage using computers, microscopes now collect much more detailed observations. Images can be collected at regular time intervals using time lapse microscopes, or can be collected over large sample areas using scanning microscopes. Computers have also made it easier to manipulate, store, transfer and view microscopy images. Increasingly, computers are being used by doctors to diagnose diseases and save lives, and by biologists to understand life.

However, the automation of microscopy is now generating data in such large amounts that doctors and biologists are increasingly struggling with data deluge. To help the doctors and biologists in drawing meaningful conclusions from the large amount of image data, it is necessary to develop methods that can assist the user in interpreting the images. Developing such automated methods is becoming an important challenge for modern microscopy.

Fortunately, progress in computer vision and machine learning in recent years has given new tools and insights to analyze the enormous amounts of image data. Based on these tools and insights, we build two vision systems for microscopic images. The systems analyze images from pathology and drug-discovery screening experiments. In the first system, we develop a cancer detector for prostate pathology images. In the second system, we develop a phenotype detector that analyzes High-Throughput Screening (HTS) images to categorize Caenorhabditis elegans

(*C. elegans*) worms according to their phenotype.

1.1 Pathology

The most effective way to treat cancer is to detect it early, and to do so, elderly people are regularly screened for cancer as a diagnostic measure. The regular screening generates a large number of cases for examination by pathologists because pathological examination is the definitive way to diagnose cancer. For example, in the U.S. each year around a million biopsies are conducted for prostate alone. Due to the large number of biopsies, each day pathologists spend between 6 to 12 hours examining slides. The repetitiveness of the task and the long hours make pathologists susceptible to fatigue that can result in misdiagnoses. In fact it has been stated that "one of the best ways to reduce errors in anatomic pathology is for the pathologist to avoid fatigue and physician burnout" [HLE04].

With digitization of pathology, due to the recent availability of scanning microscopes that can image whole-slides, some of the pathologist's burden can be reduced by building computer-assisted diagnosis (CAD) tools. This is inspired by the use of CAD tools at many clinics to detect breast cancer in mammograms [TRX+09].

How can CAD tools assist pathologists? If we consider prostate biopsies, we find that as much as 80% of biopsies are benign [GBC+09]. This suggests that a CAD tool that screens biopsy tissues can save pathologists time spent looking at benign tissues. A tool that draws pathologist's attention to diseased parts in the large images would make it easier for pathologists to spot and diagnose the disease. Such tools can also be used to improve the pathologists workflow. For example, to detect prostate cancer in a patient, around 8 to 20 biopsy samples are extracted. With a CAD tool that sorts the slides according to likelihood of cancer, it will increase the chances of pathologists encountering diseased slides earlier. This would reduce the average number of slides that pathologists examine per patient.

Such CAD tools can also help with storage and transfer of the digital pathol-

ogy images. Images generated by whole-slide scanning microscopes for pathology are enormous because the tissue samples are typically scanned at a 40x magnification. For example, a typical prostate biopsy slide with 3 horizontal sections of the needle core has a size of $150,000 \times 20,000$ pixels when digitized. Even after applying image compression, the file size of each slide's image can be more than 300MB. Consequently, to diagnose remotely, more than 30GB of data per pathologist per day would have to be stored and transferred. Storing and transferring such large amounts of data increases hospital's running costs and can be prohibitively expensive for pathology clinics. These costs can be reduced if an automated screener can identify parts of the images that have diagnostic information and send only those parts. If required, more details can be transfered later if the pathologist is unable to diagnose based on the screening.

CAD screening tools can also help in more consistent grading of cancer. Pathologists grade cancer to predict the chances of metastasis and fatality. Grading cancer is challenging, and errors such as under- and over-grading for prostate cancer have been reported [AMJ+01, DKK+98, RvLM+00, EAJE05]. To make grading consistent, pathologists suggest increasing consultations on difficult cases with experienced pathologists [HLE04, Sir00, DL01]. Such consultations will become more common once telepathology is widely adopted. CAD screening tools can accelerate the adoption of telepathology by cutting down the bandwidth required. Additionally, by screening out uninteresting cases, CAD screening tools will give experienced pathologists more time to consult on difficult cases.

CAD screening tools have proven to be effective in assisting pathologists with cytopathology slides, where free cells or tissue fragments are studied for diagnosis, *e.g.*, Pap smears [BDD+05, DdI+07]. However, few tools exist to help pathologists with histopathology, where pathologists examine biopsies or surgical specimen.

As a step towards building CAD tools for histopathology images, we build a cancer detector for prostate needle core biopsies. Prostate biopsies are some of the most common histopathological tissues examined by pathologists. Hence a tool that reduces the time required to diagnose the prostate would have a large impact

in reducing pathologist's fatigue and improving diagnostic accuracy. Furthermore, prostate biopsy tissues are similar to breast and lung biopsy tissues, so it is likely that techniques developed to detect prostate cancer would also be useful when developing breast and lung cancer detectors. Finally, along with assisting pathologists in diagnosis, our cancer detector is also a step towards automated cancer detection even though an automated cancer detector that can be used clinically may be years away.

1.2 High-Throughput Screening Experiments

Similar to pathologists, biologists conducting High-Throughput Screening (HTS) experiments spend a large amount of time manually sifting through numerous images to look for interesting cases. The aim of HTS experiments is typically to identify a small number of genes or chemicals from large libraries that cause a particular phenotype. In HTS experiments, robots and computers do a significant portion of the experimental work that allows probing of large numbers of chemicals or genes. The number of experiments conducted in HTS starts at about tens of thousands and can go up to millions. HTS has been critical for discovering drugs in the pharmaceutical industry and with the recent availability of affordable equipment, HTS is becoming more common in academia as well.

Recently, researchers have started conducting HTS experiments on model organisms instead of cell-culture or biochemical screens. In biochemical screens, chemicals are tested on isolated proteins, whereas in cell-culture screens chemicals are tested on thousands of identical cells. But the chemicals found using biochemical and cell-culture screens have been found to be ineffective when tested on whole organisms like mice [GS10]. The identified chemicals often suffer from absorption, solubility, metabolic stability and toxicological problems which make them unacceptable as drugs. However, in whole organism screens, chemicals are effective only if they don't have these problems. Additionally, screens to identify chemicals or genes that impact multicellular phenomena such as aging and immunity can only be done on whole organisms. Furthermore, whole organism screens can identify

chemicals that might be ineffective at the individual cell- or protein-level but may be effective for intact multi-cellular animals. Several such chemicals have already been identified in screens related to infection by pathogens [MBA+06, BFA+07].

For HTS experiments conducted on model organisms, experimental observations in the form of microscopic images are still manually analyzed. Manual inspection of the images is labor intensive, slow, and subjective. The importance of automating this bottleneck has been recently stated as:

> Although many large-scale studies have been performed manually over the last decades, either through effective design of the phenotyping assay or thanks simply to the sheer doggedness of the scientists involved, there is little doubt that the development of automated or semiautomated phenotyping approaches could enable studies that have previously not been possible. [dS10]

The time saved by automated analysis of HTS images can be substantial. For cell-culture screens, a study at the Broad Institute found that the time required to conduct cell-culture genome-wide screens of 20,000 samples is 8 to 48 people-months and for 300,000 chemical screens is 120 to 720 people-months. With automated image acquisition and analysis, the time was cut down to 1.25 people-months for a genome-wide screen and 4 people-months for the chemical screens [Car11].

For conducting whole organism HTS screens, the worm *C. elegans* is one of the most promising organisms. *C. elegans* is a free-living, 1mm long, transparent nematode (roundworm) found in compost. It is used to study complex multicellular biological processes by over 600 laboratories worldwide [wor11]. *C. elegans* is attractive for HTS not only for the conservation of genetic pathways relevant to human disease but also because worms are small (many worms can fit in a single well), transparent (thus easy to image), and easy to grow and manipulate. Silencing individual genes is fairly straight forward for *C. elegans* using RNA interference (RNAi). To silence a particular gene, worms are simply fed with bacteria that express double stranded RNA (dsRNA) targeting that gene. This simplicity of conducting RNAi makes *C. elegans* the only animal in which full-genome screens are currently feasible. Earlier studies on *C. elegans* have led to major discoveries, including the identification of key genes in the insulin pathway

[KCG+93, OPG+97], and the targets of common drugs, such as Prozac [RSTH01]. *C. elegans* also allowed the discovery of fundamental ancient biological processes such as programmed cell death [EH86](Nobel Prize 2002) and gene regulation by small RNAs [FXM+98, WHR93, LFA93](Nobel Prize 2006).

The availability of large-scale libraries of RNAi reagents and chemicals has driven the demand for automated, high-throughput methods to handle *C. elegans*. Consequently, all the sample preparation and image acquisition steps have been automated for *C. elegans* imaging screens. The only bottleneck that remains is the automated image analysis of the screens.

One of the ways to conduct HTS experiments on worms is to breed them in water. When the worms are imaged in liquid media, segmenting worms from background is simple. However there are inherent problems with growing worms in liquid media. Worms grown in liquid media grow slowly and are stressed [HBL+02]. In addition, RNAi is less effective in liquid media [LCT+06]. In past experiments, the worms were grown in agar and then transferred to liquid media for imaging and analysis [MCLF+09]. However, transferring worms to a liquid medium is labor-intensive. In addition, the stress of transfer can affect the worms and invalidate the results.

One such screen where worms have to be imaged on agar is Nile Red study. It has been recently proposed that the intensity of the fluorescent Nile Red dye correlates with the *C. elegans* rate of aging [OSCR09]. Worms with low Nile Red signal live longer than wild-type (or normal), and those with increased Nile Red signal live shorter than wild-type. With Nile Red as a marker, HTS screening experiment could be conducted to find the genes that impact worm's life-span by silencing each gene using RNAi. Experiments with Nile Red can only be conducted with worms bred and imaged on agar as the stress of being in water can overwhelm the subtle Nile Red signals.

Agar significantly complicates segmentation of worms from the background (Figure 1.1). To make use of the equipment developed for cell-culture, HTS screens for *C. elegans* are carried out in the same 96-well microtiter plates that are used for cell-culture experiments. Each well in the microtiter plate is a few millimeters

wide. In such small wells, the meniscus of semisolid agar is deep. This results in inconsistent lighting inside the well, and in the images only the central part is well lit and in focus. Additionally, the track marks left by worms in agar appear similar to the worms.

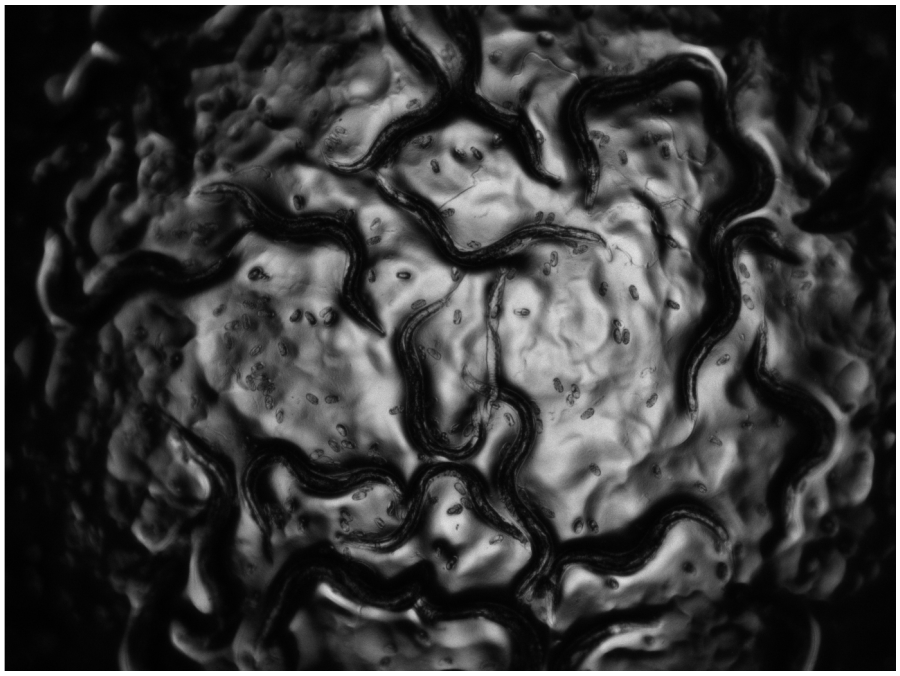


Figure 1.1: Examples of an HTS image of worms in agar. Limitations of the automated imaging setup cause the edges of the image to be out of focus and dim. The tracks left in the bacterial food source, and touching and overlapping worms complicate segmenting the worms.

To assist biologists in phenotyping worms in such complex images, we build a vision system that segments worms from the complex agar background. Based on the segmentation of worms and the Nile Red fluorescent patterns inside them, our system could identify the Nile Red phenotypes of worms in control experiments.

1.3 Computer Vision and Machine Learning

To develop the two vision systems we use insights from computer vision. The recent advances in computer vision are based on statistical methods from ma-

chine learning. Earlier computer vision methods relied on manually tuned models that resulted in methods that worked well in a few controlled environments. In contrast, current machine learning based vision systems work well in more diverse environments. The reason for their success is that machine learning algorithms learn predictors based on large training sets with higher diversity. The predictors thus learned, use a larger number of features to make their decision, which makes them much more robust. In addition, machine learning algorithms such as Support Vector Machines (SVMs) [CV95] and Boosting [FS97] do feature selection. This provides a considerable engineering advantage to developers, as they can now add any features they think are likely to help without worrying that the resultant classifier may over-fit.

Based on its success in detecting faces, we decided to build our systems using Boosting. Boosting's strengths are its speed and its ability to predict confidences as scores. Additionally, Boosting used with decision stumps as weak rules makes it easier to combine different visual cues such as texture and color. Features with different scales and types can be used without any manipulation.

Even though Boosting and other machine learning algorithms have made it considerably easier to develop robust vision systems, two important design questions that still have to be answered while developing systems for new imaging domains are: 1) How should we break down the system in smaller parts so that machine learning can predict accurately? 2) What features or descriptors should we use to predict?

The first question has to be answered because machine learning algorithms predict well only if the task is well formulated. For example, we cannot expect machine learning to predict the dominant phenotype of worms in an image by giving a raw image as an input. For any complex task, it is important to break it down into smaller pieces that can be handled by machine learning. For example, to segment worms from complex agar background, we first detect worm segments. The worm segments are large objects that can be accurately predicted based on the information available in their neighborhood. In contrast, we could have worked with pixels and tried to predict if each pixel belongs to worm or not based on

the information in the pixel's neighborhood, but such a classifier cannot predict reliably because a small neighborhood around the pixel does not have sufficient information. In both the projects, we try to formulate the tasks for machine learning so that the predictions are reliable. We do this by detecting intermediate objects. We selected the objects by judging whether they can be reliably detected and whether they would improve the performance after detection. One way we judged the usefulness of these objects was by checking whether the objects were biologically or physically significant.

The second question is important because predictors learned by machine learning perform poorly if the features are uninformative. In the absence of informative features, predictors learned using machine learning algorithms predict with low confidence and do not generalize well. To learn accurate predictors, machine learning methods need features that capture the distinguishing aspect of the task. For example, a face detector learned with features that capture contrast between facial structures performs better than with face detectors learned with features that use raw pixel values [VJ01]. Designing good features is tricky because we still do not understand how humans perceive and what are the distinguishing visual cues for any object. For example, what makes a chair a chair? Over the years many good features have been developed for color, texture and shape in computer vision [MBSL99, DT05]. Yet, for many imaging domains these features are often not sufficient and it is necessary to develop additional features that can capture traits particular to that domain. Designing features for biomedical domain is particularly challenging because images are messier, noisier and more complex. Problems such as incorrect focus, low resolution, inconsistent lighting or damaged samples are common. The large amount of noise sometimes makes it difficult even for an expert to discern the signal. In our case, we identified the visual cues that help in designing informative features by studying the underlying biology and also by using insights provided by the pathologists and the biologists.

1.4 Thesis Organization

We begin with a brief introduction to Boosting in Chapter 2 and give details of the implementation used. In Chapter 3, we describe the cancer detector we developed for prostate pathology images. In this chapter we give details of the prostate's structure and later describe how we developed the features based on the structure to improve the cancer detector's performance. In Chapter 4, we describe our machine learning based segmentation technique for worms imaged on agar. We report the accuracy of our method at detecting Nile Red phenotypes when the segmentation technique is combined with techniques to analyze a worm's fluorescent patterns. We finally conclude the thesis in Chapter 5.

Chapter 2

Boosting

2.1 Background

Boosting was posed as an open problem by Kearns and Valiant in [KV89]. The problem asked, in the PAC (Probably Approximately Correct) setting, whether it is possible to combine weak learners that predict better than random guessing into a strong learner that predicts with arbitrarily high accuracy?

In PAC learning [Val84], a concept class C over instance space X is a collection of rules or hypothesis $c : X \to \{0, 1\}$. D is any fixed distribution over $X \times Y$, where $Y = \{0, 1\}$. The error of an hypothesis is

$$\mathrm{err}(h) = \Pr_{(x,y)\sim D}[h(x) \neq y]$$

The training examples are returned by an oracle procedure that takes unit time per example. The concept class C is called PAC learnable if there exists an algorithm A such that for any $\epsilon > 0$ and $\delta > 0$, using as input $m = \mathrm{poly}(1/\epsilon, 1/\delta)$ labeled examples generates a hypothesis h such that with probability at least $1 - \delta$ over the choice of the training set, the error is smaller than ϵ.

$$\Pr[\mathrm{err}(h) \leq \epsilon] \geq 1 - \delta$$

Suppose we are given a weak learning algorithm L for C that outputs a hypothesis with error at most $\beta < 1/2$ *i.e.,* L is only guaranteed to give some

hypothesis whose error is slightly better than random guessing. The problem of boosting is to know if it is possible to transform L into an algorithm A that outputs a hypothesis with any arbitrarily small error $\epsilon > 0$. In other words, can an algorithm that learns with low accuracy be boosted into an algorithm that learns with as high an accuracy as desired.

Schapire [Sch90] showed a weak learner can indeed be boosted into a strong learner. He showed that three weak hypotheses can be combined into a hypothesis with higher accuracy. By modifying the distribution D that was given to L, the weak learner L was forced to find hypotheses that complemented each other. The final hypothesis which took the majority vote over the three weak hypotheses had lower error than the three weak hypotheses. By hierarchically combining groups of three weak rules, it was shown that a learner with arbitrary accuracy could be learned. Later, Freund [Fre95] developed "Boost by Majority" in which weak rules were combined simultaneously into a strong learner.

Both the above algorithms required that all the weak rules have certain pre-fixed advantage over random guessing. Freund and Schapire [FS97] later developed Adaboost that didn't require weak rules to satisfy any such condition. Also, Adaboost was practical and easier to implement.

Due to its simplicity and superior performance, Adaboost provided a stable base to develop computer vision applications when computer vision started using statistical approaches. The best known example of Adaboost's use in computer vision is the first practical face detector by Viola and Jones[VJ01] that detected faces in real time. Boosting's strengths are its speed and its ability to predict confidences as scores. Additionally, boosting used with decision stumps as weak rules makes it easier to combine different visual cues such as texture and color. Features with different scales and types can be used without any manipulation.

2.2 Adaboost Algorithm

Adaboost and other boosting learning algorithms learn an accurate classifier by combining many rules of thumb. The input to the boosting algorithm is a

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$

For $t = 1, \ldots, T$:

- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$

- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

  where Z_t is a normalization factor that makes D_{t+1} sum to 1 ($Z_t = \sum D_t(i) \exp(-\alpha_t y_i h_t(x_i))$)

Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Figure 2.1: Adaboost algorithm.

training set $(x_1, y_1), \ldots, (x_n, y_n)$ where each x_i belongs to an *instance space* X, and each *label* y_i is in a label set Y. For our system $Y = \{-1, +1\}$ as our detector classifies each patch as cancerous or not-cancerous. The boosting algorithm receives the rule of thumbs as weak rules repeatedly in a series of rounds $t = 1, \ldots, T$. Boosting combines these received weak rules into an accurate predictor by keeping a distribution or a set of weights over the training set. The weight of this distribution on training example i on round t is denoted by $D_t(i)$. Initially, all the instances are given equal weight. But with each round the weights of incorrectly classified examples are increased so that the weak learner is forced to concentrate on the hard examples.

Boosting is able to combine the weak rules into an accurate predictor if at each step the weak rule is better than random prediction. In other words, the weak rule $h_t : X \rightarrow \{-1, +1\}$ received by the boosting algorithm should have an error ϵ_t with respect to D_t less than 0.5 .

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i) \neq y_i} D_t(i) < 0.5$$

Once the weak rule h_t has been received, Adaboost selects a parameter α_t that assigns weight to the rule. The distribution D_t is next updated using the rule shown in Figure 2.1. As a result of this update, weights of instances misclassified by h_t are increased and weights of correctly classified instances are decreased. Thus, the weight tends to concentrate on "hard" examples.

The *final classifier* H is a weighted majority vote of the T weak rules where α_t is the weight assigned to h_t. The final rule puts more trust into weak rules with lower error as the weight of a weak rule α_t is more if the error ϵ_t is small.

2.3 Generalization Error

The aim of a learning algorithm is to learn a classifier that has low generalization error *i.e.*, error on instances that were not part of the training set. The main strength of Adaboost and other boosting algorithms is that they learn classifiers that have low generalization error.

The low generalization error of Adaboost is explained by using *margins* of the training examples. The margin of an instance (x, y) is defined as

$$\text{margin}(x, y) = \frac{y \sum_t \alpha_t h_t(x)}{\sum_t \alpha_t}$$

It is a number between -1 and $+1$ which is positive if and only if H correctly classifies the example. Moreover, the magnitude of the margin can be interpreted as a measure of confidence in the prediction.

It has been shown by [SFBL98] that larger margins on the training set lead to better upper bounds on the generalization error. Specifically, the generalization error is at most

$$\hat{\Pr}[\text{margin}(x, y) \leq \theta] + \tilde{O}\left(\sqrt{\frac{d}{n\theta^2}}\right)$$

for any $\theta > 0$ with high probability ($\hat{\Pr}[\text{margin}(x, y) \leq \theta]$ is the fraction of training examples that have margin less than θ, d is the VC dimension of the weak rules that measures the richness and complexity of the weak rules, and n is the number of training examples).

Adaboost learns good classifiers as it continues to improve the minimum margin of the training set even after the error on the training set is zero. As a result, the generalization error of the learned classifier decreases with each round and Adaboost does not over-fit even though it is run for a large number of iterations.

2.4 Decision Stumps

For our system, instances are given as a vector of feature values that are numerical descriptions of the properties of the patch. The numerical features are used to construct *decision stumps* that are used as the weak rules. Decision stumps predict by comparing a particular feature value against a threshold *i.e.*, if the feature's value is smaller than the threshold then predict a particular class or else predict the other class. In our implementation of boosting, we find the best decision stump in each iteration by searching over all features and thresholds and supplying it as the weak rule.

2.5 Feature Selection

Another advantage of using boosting is its ability to select features. This allows us to add any set of features if it is likely that some of the features in the set are discriminatory. Boosting is able to learn high performing classifiers even when the number of features is very large. The ability of boosting to select features allows us to add a number of feature sets that capture different types of information.

This is in contrast to previous machine learning techniques where addition of such large numbers of features would result in classifiers that over-fit. To avoid over-fitting, the number of features had to be reduced before using the machine learning algorithm. Boosting and support vector machines [CV95] are learning algorithms that remove the need for feature selection.

2.6 Logitboost

While Adaboost resists over-fitting when classes are separable, it is sensitive to outliers and mislabeled examples in the training set. This sensitivity arises from the exponential weighting that puts large weights on outliers. In such cases, Adaboost over-fits and has poor generalization error.

This influence of outliers is lessened in Logitboost [FHT98]. Logitboost limits the maximum weight that any example can get to 1. Figure 2.2 shows the Logitboost algorithm; the only difference from Adaboost is the weighting scheme. The advantages of this weighting scheme have been confirmed empirically where Logitboost outperformed Adaboost. Based on these benefits, we train our classifiers using Logitboost.

2.7 Accuracy of weak rules

Accuracy of boosting depends on the margin distribution. The final margin distribution depends on the accuracy of the weak rules. In [SFBL98] it is shown

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$

For $t = 1, \ldots, T$:

- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$$

- Choose $\alpha_t = 1 - 2\epsilon_t$

- Update:

$$\begin{aligned} \text{margin}_i &= y_i \times \sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x_i) \\ D_{t+1}(i) &= \frac{1}{1 + \exp(\text{margin}_i)} \times \frac{1}{Z_t} \end{aligned}$$

where Z_t is a normalization factor that makes D_{t+1} sum to 1.

Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Figure 2.2: Logitboost algorithm.

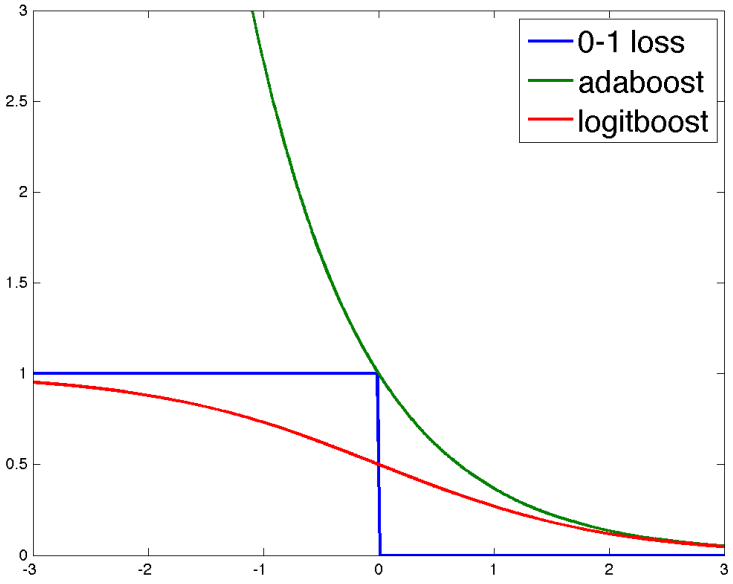


Figure 2.3: Weights as a function of the margins: Logitboost better approximates the ideal 0-1 loss as compared to Adaboost. Logitboost gives less weight to examples with large negative margins and hence is less sensitive to outliers and labeling errors.

that if the weak rule's accuracy is greater than γ then most training examples have margins greater than γ.

But on the other hand it can also be shown that if all the weak rules have accuracy less than γ then few training examples can have margin greater than γ [SsS08]. This signifies that unless we have good weak rules, the final classifier is unlikely to generalize well. Thus, to make sure that the final classifier generalizes well, we need to make sure that the weak rules are sufficiently accurate.

From the above, it can be reasoned why raw pixels as features do not lead to accurate classifiers. This is because decision stumps on raw pixel values do not have high accuracy. For example, consider the task of classifying faces from non-faces. At any pixel location, non-faces are equally likely to have high intensity as well as low intensity. For faces too at any pixel location, intensity is also likely to be higher or lower based on person's skin color. As a result, each pixel has only limited prediction capacity whether the patch around it is a face or not. And consequently, the final classifier's accuracy is also low.

This analysis can guide a developer in designing features for better predictors. The features should be such that a weak rule can get as many examples right as possible. For example, consider a toy case where our examples are patches of size 100×100 pixels. We have two classes: for patches in the first class all pixels are empty (has value 0) while for patches in the second class there is one random pixel that has a non-zero value. One way to build features would be simply use the pixel values as features. With such features, we will have to use 10,000 different weak rules (1 per pixel) to separate the two classes. But had we used the sum of all the pixel values we would have been able to separate the two classes using a single weak rule. One way to use fewer features to capture a particular trait is to normalize the data. Before normalization, boosting will use a large number of weak rules to learn a classifier but after normalization boosting captures the same trait with fewer rules. And as a result we get classifiers that generalize well.

Given: $(x_1, y_1, w_1), \ldots, (x_m, y_m, w_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$ and w_i are the example weights

Initialize $D_1(i) = 1/m$

For $t = 1, \ldots, T$:

- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} w_i \times [h_t(x_i) \neq y_i]$$

- Choose $\alpha_t = 1 - 2\epsilon_t$

- Update:

$$\begin{aligned} \text{margin}_i &= y_i \times \sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x_i) \\ D_{t+1}(i) &= \frac{1}{1 + \exp(\text{margin}_i)} \times \frac{1}{Z_t} \end{aligned}$$

  where Z_t is a normalization factor that makes D_{t+1} sum to 1.

Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Figure 2.4: Logitboost algorithm with weights on examples.

2.8 Implementation

In our case the training set has many more examples of one type than the other type. For example, in prostate cancer detection, our training set has more non-cancerous or negative examples than positive examples. This bias towards negative examples leads to poor classifiers. To reduce the effect of this bias, we assign weights to examples. The weights are assigned such that the total weight of the positive examples equals the total weight of the negative examples. The boosting algorithm uses these weights when it calculates the accuracy of the weak rules ϵ_t. Because of the this weighting, the boosting algorithm penalizes the mistake on a negative example lesser than the mistake on a positive example. We found that such reweighting results in a more accurate classifier. This reweighting is done only on the training set so that boosting finds a better combination of weak rules.

We use `JBoost`[JBo10], a Java software package that implements Adaboost, Logitboost and other boosting algorithms along with additional features. We run the algorithm for 200 iterations using Logitboost.

Chapter 3

Prostate Cancer Detection

Digitization of pathology due to the recent availability of commercial whole-slide scanning microscopes has provided an opportunity to help pathologists in diagnosing better and faster by building Computer-Aided Diagnostic (CAD) tools. As prostate is one of the most commonly diagnosed tissue by pathologists, we developed a cancer detector that screens prostate biopsy slides. One of the ways the screening tool can make it easier for pathologists to spot cancer is by suggesting suspicious regions. Also, by transferring only suspicious parts over the Internet, the screening tool can reduce the bandwidth required to diagnose from remote locations. This will accelerate the adoption of telepathology by hospitals.

To accurately detect cancer in prostate tissue, we found that color and texture features were not sufficient. To improve the performance, we developed a new set of features based on prostate anatomy. In the prostate, the main functional unit is the gland. The structure of the gland is one of the important visual cues that pathologists study to detect cancer. This is because glands undergo mild to severe deformation depending upon the grade of the cancer. To capture the deformation underwent by glands in a particular location, we detect the three most common structural deformations of glands. With the addition of features based on these structures, the performance measured in terms of Area Under the ROC curve improved from 0.85 to 0.95

In this chapter we give the details of our cancer detector. In Section 3.1 we give details of prostate anatomy. In Section 3.2 we review related work. We

give a brief overview of our system in Section 3.3. We give details of the feature engineering in Sections 3.4 and 3.5 and give the details of the data in Section 3.6. In Section 3.7 we give the results we obtained and finally conclude in Section 3.9.

3.1 Prostate Anatomy

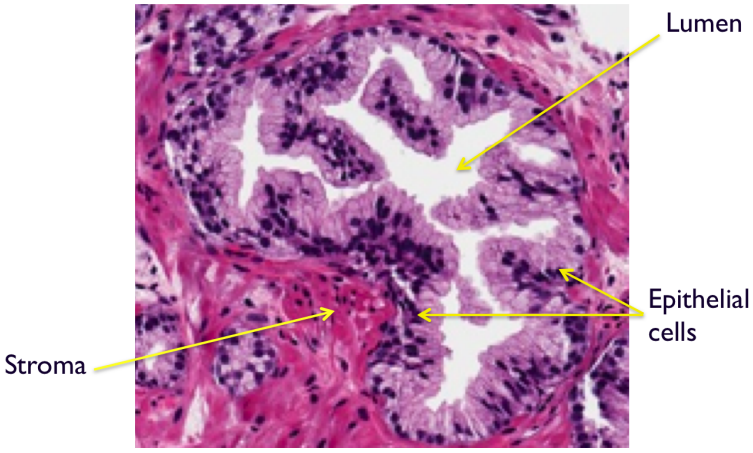


Figure 3.1: An example of prostate gland. The glands are the main functional unit of the prostate. The epithelial cells that line the lumen secrete seminal fluid. The network of ducts and glands that collects the fluid is supported by stroma. In prostate cancer, the epithelial cells mutate and alter the gland structure and invade the stroma. These cells can also metastasize. Details of the glands are best visible at 40x magnification.

The visual appearance of prostate tissue is determined by its anatomy and hence a good understanding of the anatomy is crucial for building the cancer detector. In this section we briefly describe the anatomy ([AR74]).

The prostate's function is to produce seminal fluid. The fluid is secreted by specialized epithelial cells. The epithelial cells are arranged in glands and ducts (Figure 3.1). The lumen form a network that collects the seminal fluid and delivers it out of the prostate. The network of glands and ducts is surrounded by connective tissue (stroma). Between the connective tissue and the epithelial cells, there is a layer of basement membrane that keeps the epithelial cells attached to the stroma and confined within the glands. The prostate mainly consists of stroma and glands.

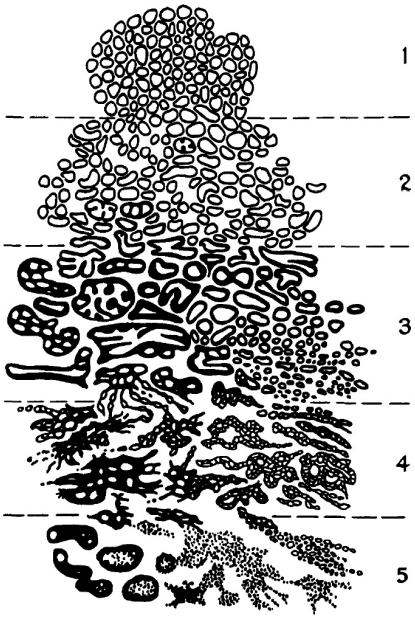


Figure 3.2: Deformation of the gland structure according to the cancer grade. The glands become more and more unstructured with increasing gleason grade [Gle66] denoted on the right.

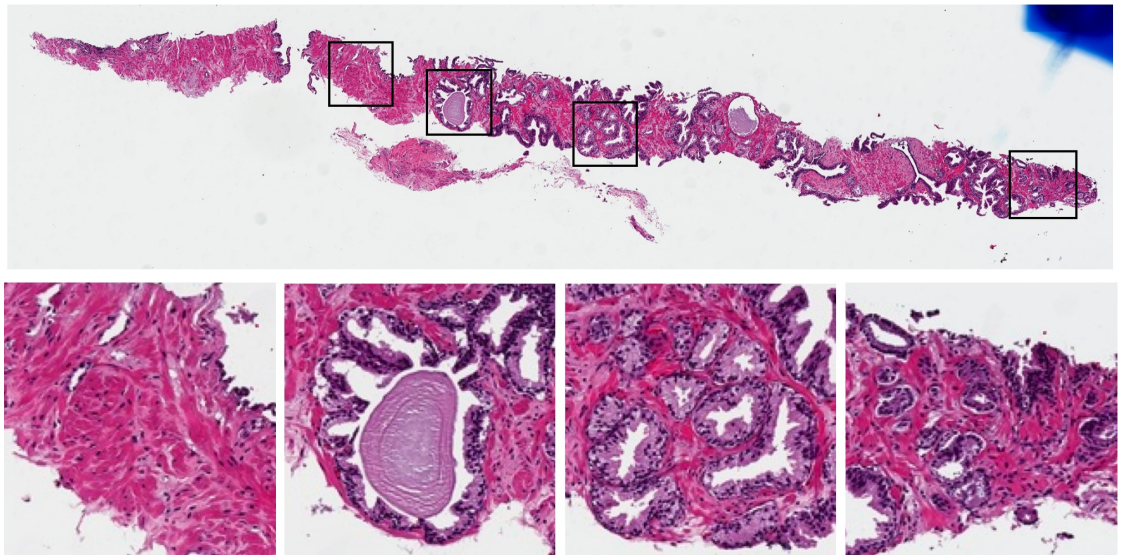


Figure 3.3: An example of needle core biopsy. Most of the tissue is composed of stroma which is featureless and not difficult to separate. Most of the diagnostic information is in the structure of glands that can be distorted and broken in needle core biopsies. A tissue sample has a large number of structures such as concretions in the second patch that make diagnosing cancer a difficult task.

For disease diagnosis thin slices of tissue samples are stained with hematoxylin and eosin (H&E stain). The thin slices mounted on glass slides are first examined at low magnification to study the gross appearance and then at a higher magnification to study the finer details.

At low magnification, the glands are distinctive with a peculiar structure. While the lumen is irregular, the epithelial cells are constrained to be in a single layer. Also the nuclei of the epithelial cells are programmed to stay away from lumen. Since nuclei are stained dark blue by hematoxylin, lumens are always surrounded by first a pale outline of the epithelial cell's cytoplasm which in turn is surrounded by a dark line of epithelial cell nuclei. Stromal cells on other hand have smaller nucleus and larger cell size. Hence it appears smooth without any activity and is colored pink by eosin.

The underlying cause of prostate cancer is mutation of the genome of the epithelial cells. Prostate cancer is diagnosed by studying the changes brought about by these mutations. The stroma does not usually undergo important changes and is not usually studied for diagnosis. It sometimes does undergo increased cell division when stimulated by invading epithelial cells in which case it is studied for diagnosis.

Two important changes exhibited in H&E examination are the change in nuclear to cytoplasmic ratio, and the change in the cytoplasmic color of the epithelial cells and the structure of glands.

The change in cytoplasm composition in the mutated cells is due to their fast multiplying nature. To multiply faster, cancer cells need a larger concentration of messenger RNA. The larger concentration of messenger RNA increases their affinity for the blue Hematoxylin stain. As a result, the cytoplasm of the mutated cells is darker and more blue than the cytoplasm of normal cells. This color difference between the cancer cells and normal cells is further enhanced due to the presence of larger nuclei in the cancerous cells.

The other important change is the alteration of the gland structure. The alterations depend on the cancer grade as shown in Figure 3.2. In low grade cancer, the epithelial cells multiply at a faster rate while still being bound to the basement

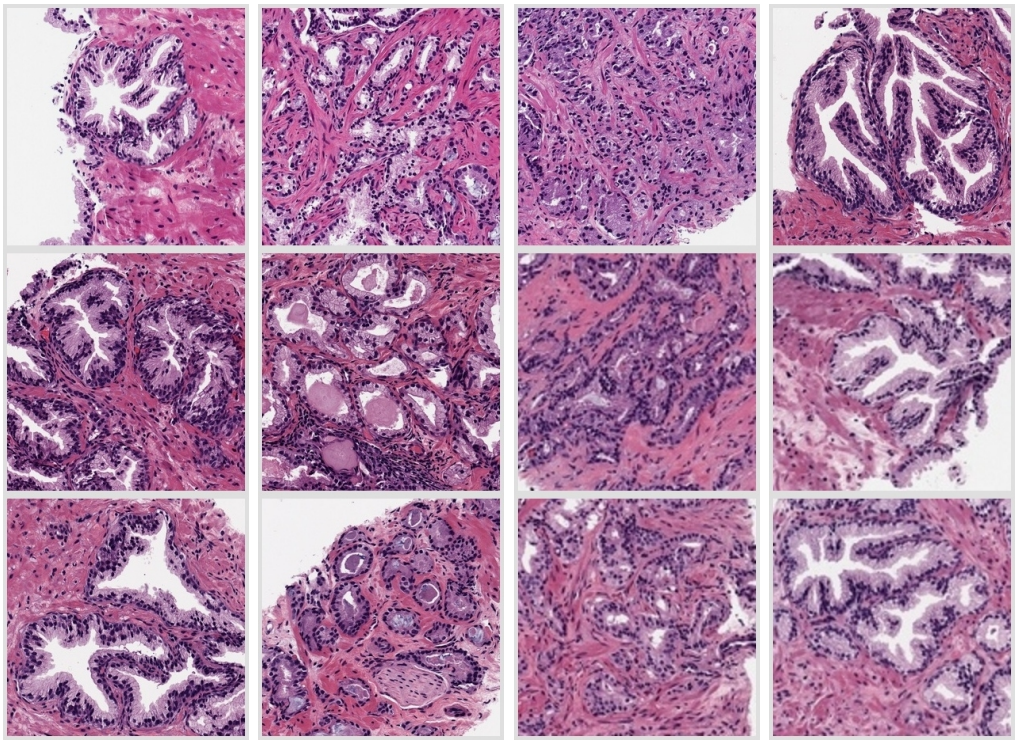
membrane. As a result, low grade cancer glands are typically smaller than normal glands and appear in closely packed groups pushing into the stroma. In high grade cancer, the fast multiplying epithelial cells do not remain bound to the basement membrane and can invade the stroma singly or in clusters. As a result, high grade cancer cells appear as an unstructured mass of epithelial cells. The glands often have bubble shaped lumens or no lumen at all.

But not all gland structure alterations are cancerous. A common non-cancer alteration of the glands is hyperplasia. Hyperplasia is a benign condition that is common in the prostates of aging men, the same group of men that is screened for cancer. Therefore, discriminating hyperplasia from cancer is critical in diagnostics. In hyperplasia, the epithelial cells multiply at a higher than normal but do not invade the stroma. This results in glands with a convoluted lumen as the extra epithelial cells lead to infolding as shown in Figure 3.4(d).

3.2 Related Work

The work most related to ours is the recent work by [DFTM10]. They also developed a cancer detector in prostate needle core images. The features used were first order pixel value statistics and texture features based on co-occurence and Gabor filters. For each feature, probability densities were estimated for the cancer and non-cancerous pixels. Based on these density estimates the optimal threshold values were calculated. These weak rules were then combined using boosting to learn a pixel classifier. To speed up the processing, the cancerous regions were found by classifying the image at multiple resolutions, starting with the coarsest resolution and discarding obvious negatives and then moving to finer resolutions. The area under the ROC curve for their detector was 0.84.

Recently, [XSJ$^+$10] developed a method to detect glands and study the morphological structure of the glands for needle core prostate biopsies. They combined normalized cuts with active contours to segment the glands. The segmented structures are then classified as cancerous or non-cancerous based on the shape features developed by [SM10]. Similarly, [MTF$^+$10] also developed a gland seg-



(a) Normal glands (b) Low-grade Cancer Glands (c) High-grade Cancer (d) Hyperplastic glands

Figure 3.4: Examples of common formations found in prostate tissue biopsies.

mentation and classification technique for whole prostate sections. They used a region growing technique to segment the glands. Glands with similar sizes are clustered using Markov random fields to mark the extent of similar regions.

Prior to these studies most of the work [TTP+07, DAB+04, FSZJKZ07] concentrated on separating small homogenous patches of tissue mainly because whole slide scanners were unavailable. Lack of whole slide scanners also meant that the data sets used were small. [DAB+04] developed a system to distinguish homogenous patches of stroma, benign epithelium and prostate cancer with an accuracy of 79.3%. [TTP+07] developed a system to distinguish small spots on a tissue microarray. Their system could separate the cancerous and non-cancerous images with an accuracy of 96.7% and low-grade cancer and high-grade cancer with an accuracy of 81.0%. They detect tissue parts such as nuclei, cytoplasm, red blood cells etc. using hand-coded parameters. Properties of these tissue components are used as features. In our work, we also detect structure but we detect different types of structures and also we use learning to detect the structures. [FSZJKZ07] achieved an accuracy of 96.5% in grading homogenous image patches.

3.3 Overview of the System

The basic operation of our cancer detector is to assign scores to small square patches (also called windows). The scoring procedure's goal is to assign high scores to patches with cancer and low scores to patches without cancer.

Cancer is a complex disease and does not have a unique form. It is necessary to combine multiple visual cues to detect cancer accurately. We use a two-level hierarchy to perform the detection. The low level analysis uses color, texture and structural features to detect discriminative local features of size up to a single gland. The output from the low level analysis is combined by a *patch level classifier* to produce a score associated with the image patch.

The low level features were developed to capture different visual cues suggested as relevant by the pathologists in our team. We use color and texture features which have been shown to discriminate cancerous regions from non-cancerous

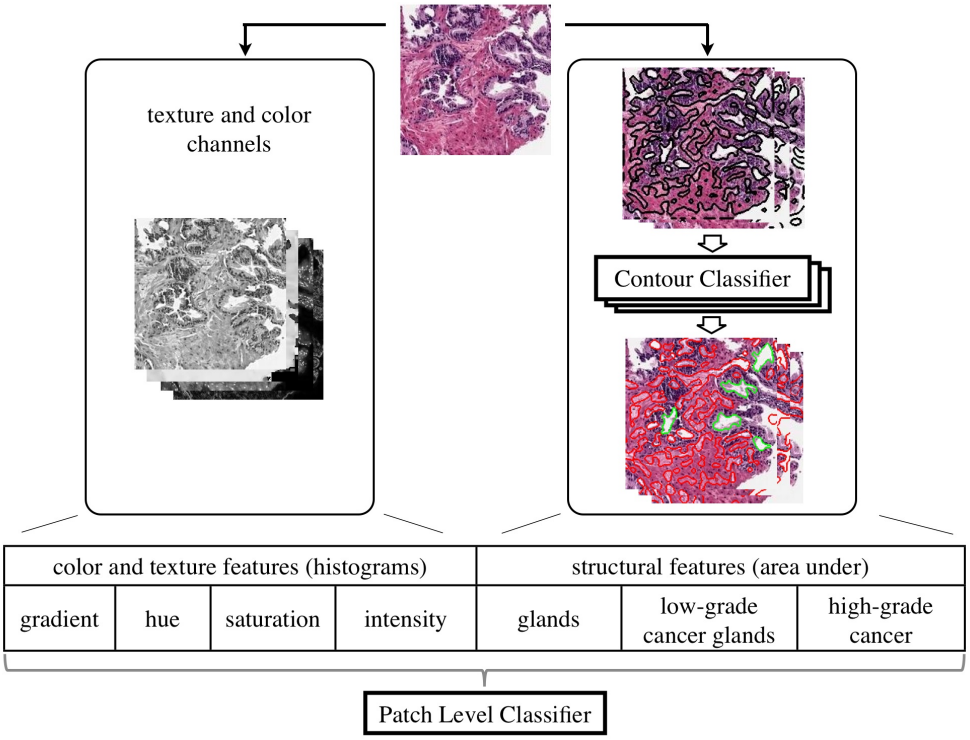


Figure 3.5: The two level hierarchy that defined the structure of our scoring algorithm.

regions [DFTM10]. We developed features based on the structure of glands which discriminate between single cancerous and non-cancerous glands. The color and texture features are described in detail in Section 3.4, and the structural features are described in Section 3.5.

The hierarchical design reflects the observation that cancer manifests itself in different scales. The color and texture features correspond to the scale of a cell, the structural features correspond to the scale of a gland, and the patch level classifier corresponds to a group of 5-20 glands.

The patch level detector and the structural detectors are created using machine learning. This reduces the amount of hand-tuning that is required. We use boosting as our machine learning algorithm as it is highly robust to non-informative features. This allows us to use all features that might be relevant for the task without worrying about over-fitting.

The slide magnification we chose for the detector is based on the way pathologists perform their diagnoses. Pathologists initially analyze the tissue sample at 4x magnification. At this magnification they can see sufficient tissue details to identify cancerous parts and later examine these parts at 20x or 40x magnification to confirm their suspicion. Following this practice, we built our system to find suspicious regions by examining the tissue at 4x magnification. At this screening magnification, we scan the image with a 120x120 pixel window to identify the parts in an image that have cancer. The window size of 120x120 pixels was chosen as a trade-off; smaller patches have fewer glands and other structures crucial for detection. On the other hand, a larger window is likely to contain both cancerous and benign regions, making it difficult to assign a label to the window. With this window of 120x120 pixels, we scan the whole image with a step size of 30 pixels to score the whole image.

3.4 Color and Texture Features

3.4.1 Texture features

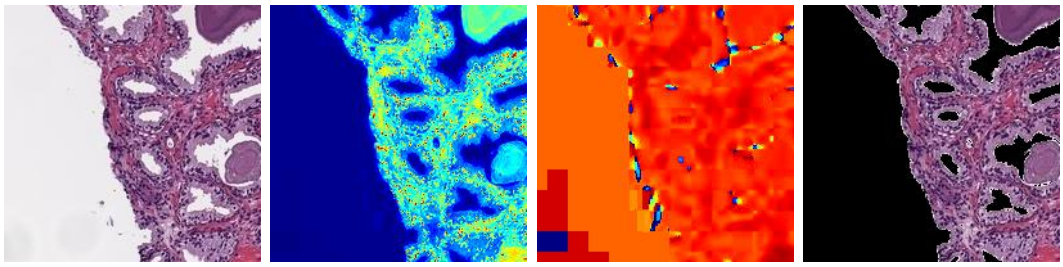
The purpose of the first set of features is to capture the irregularity within a patch. Such features can separate stroma which is smooth from the rest of the tissue that is not smooth. We capture the presence of irregularities by using the gradient distribution within the patch. We first find the gradient within the patch with a gradient filter ([GW06]). If I is the pixel intensity,

$$\begin{aligned} G_x(x,y) &= I(x+1,y) + I(x-1,y) - 2 * I(x,y) \\ G_y(x,y) &= I(x,y+1) + I(x,y-1) - 2 * I(x,y) \\ \text{gradient}(x,y) &= \|\nabla I(x,y)\| = [G_x^2 + G_y^2]^{1/2} \end{aligned}$$

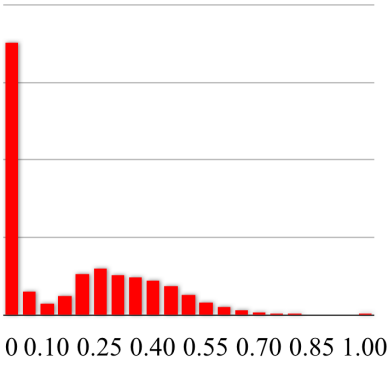
After computing the gradient at each pixel, we construct a histogram of the gradient magnitudes and use each bin's frequency as a feature. This histogram has seven equal width bins. The range is from 0 to 95^{th} percentile of the gradient values. An extra bin captured the number of pixels that were larger than the 95^{th} percentile. We chose 95^{th} percentile for the range rather than maximum to protect against outliers that may lead to large ranges with empty bins. We used seven bins, as the performance did not improve when we increased the number of bins beyond seven.

3.4.2 Color features

As described in section 3.1, cancerous regions tend to be darker and bluer than benign regions. We capture this discriminative visual cue by constructing color histograms [SB91] and using the bin frequencies as features. We use hue-saturation-value (HSV) color space, which is better than Red Green Blue (RGB) color space at representing color differences. Using the HSV image, we build 3 sets of histograms along all 3 color channels viz., hue, saturation and intensity (or value). All three histograms had 10 bins. The performance did not improve when we used more bins.



(a) Tissue Patch (b) Saturation Channel (c) Hue Channel (d) Pixels with saturation greater than 0.2



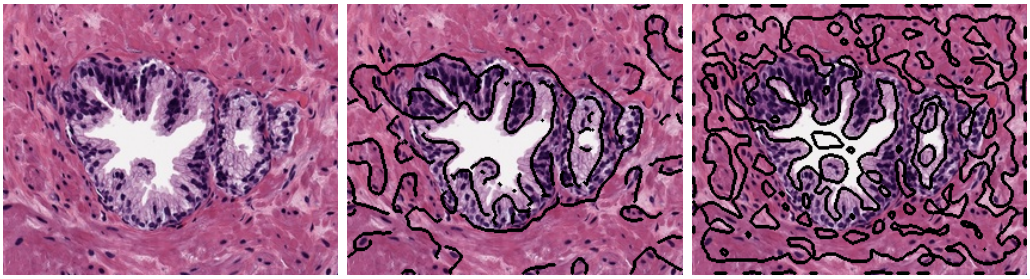
(e) Saturation Distribution

Figure 3.6: Removing less saturated pixels: Hue values of less saturated pixels are unreliable and should not be included while building color histograms. The saturation of background pixels and tissue pixels lie in different ranges. The less saturated background pixels can be separated by thresholding the saturation channel at 0.2 .

We adapt the range of the histograms to improve their discriminative power for pathology images. First, we do not include pixels whose saturation is low (Figure 3.6). These pixels have unreliable and noisy hue values. But since the amount of background may be an important clue, we do use the number of less saturated pixels as a feature. Second, we adapt the bin sizes of the hue histogram to match the hue distribution. The hue values of both cancerous and benign regions lie in a narrow range (from 250° to 330°). If the hue spectrum is divided into large bins it will miss the small shifts in color. Alternatively, if the spectrum is divided into finer bins it will increase the computational cost and over-fit in training. As a compromise, we use 7 finer bins between 250° to 330° and 3 equal sized bins to capture the rest. Saturation and Intensity histograms had 10 equal sized bins. Performance did not improve when we used finer bins. In total, each patch has 31 color features.

3.5 Structural features

To complement color and texture features, we developed structural features that detect three common variations of glands. Since cancer deforms the structure of glands, changes to gland structure provides an important visual cue for cancer detection. These changes to gland structures are difficult to capture using color and texture. The first detector is built to find healthy large glands that are indicative of normal tissue or hyperplasia. The second detector finds smaller glands that are common in low grade cancer. The third detector identifies an unstructured mass of epithelial cells characteristic of high grade cancer where the gland structure is completely destroyed. The detectors identify each of the structures by first outlining a large number of candidate regions with contours. From these candidate contours we filter out regions that do not correspond to desired structure by using a trained classifier.



(a) A gland (b) Edge detector output (c) Output of LoG filter when threshold is set to 0

Figure 3.7: Segmenting Glands: We use zero-crossings of Laplacian of Gaussian filtered image to segment glands.

3.5.1 Gland Detector

The most prominent visual cue of a gland is the sharp edge at the boundary of the lumen and the surrounding epithelial cells. This suggests that an edge detection procedure would be able to outline the glands. But out of the box edge detectors are unable to outline the whole gland because most glands have parts where the contrast between lumen and epithelial cells is low. As a result, edge detector outputs are often a large number of broken edges as shown in Figure 3.7(b) and it is a challenge to combine the broken edges into one contour.

We use zero-crossings of Laplacian of Gaussian (LoG) [HM79, GW06] filtered image. The LoG filter with scale σ is

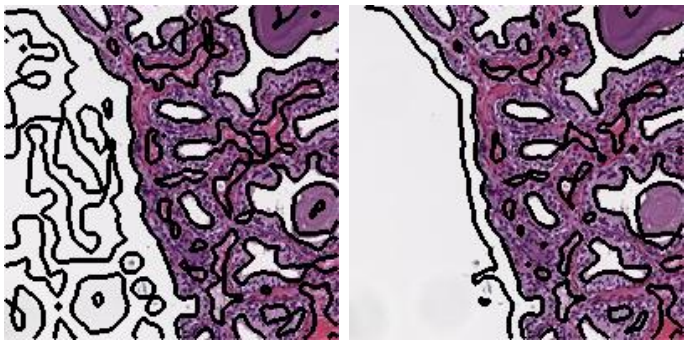
$$\mathrm{LoG}(x,y) = \frac{1}{\pi\sigma^4}\big(1 - \frac{x^2+y^2}{2\sigma^2}\big)\exp\big(\frac{x^2+y^2}{2\sigma^2}\big)$$

A significant advantage of using the zero corssings of the LoG, as opposed to other edge detectors, is that this guarantees closed contours, making it simple to extract meaningful connected components.

When LoG contour detector is applied to intensity channel, we found that the lumen boundaries of most glands get outlined by a contour. At 4x magnification, we found that a 4 pixel scale LoG filter is most suitable to outline glands (Figure 3.8).

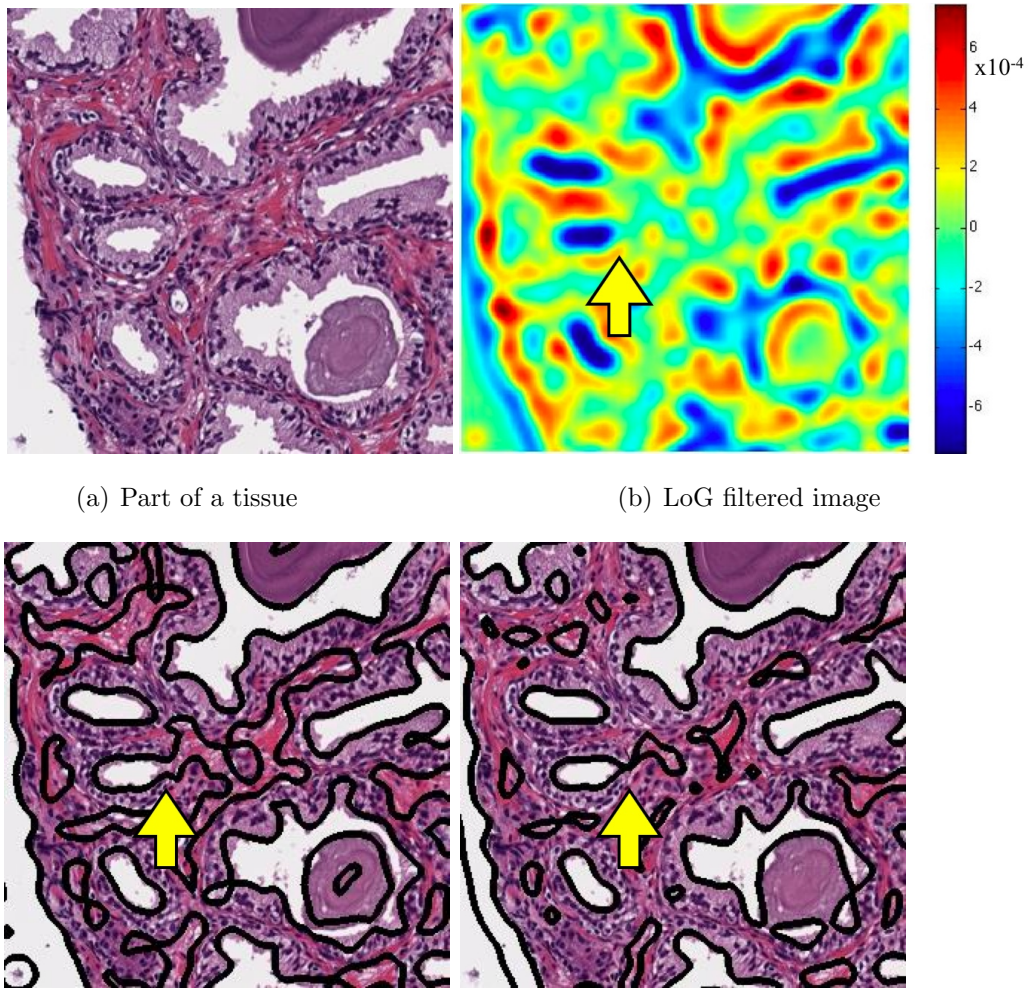
Thresholding the LoG filtered image at zero can be adjusted to get fewer

(a) Image (b) Scale 3 LoG (c) Scale 5 LoG



(d) Scale 4 LoG (e) Scale 4 LoG, thresholding at -0.001

Figure 3.8: Effect of LoG scale on gland detection: Outlining glands with LoG is relatively insensitive to the scale. Well formed glands are outlined with a scale 3 LoG (a) and also scale 5 LoG (b). In our system, we outline glands with LoG of scale 4 (c). Thresholding at zero results in contours in the background and broken glands leak into stroma. Both these problems can be lessened by thresholding at a slightly lower value of -0.001 (d).



(a) Part of a tissue

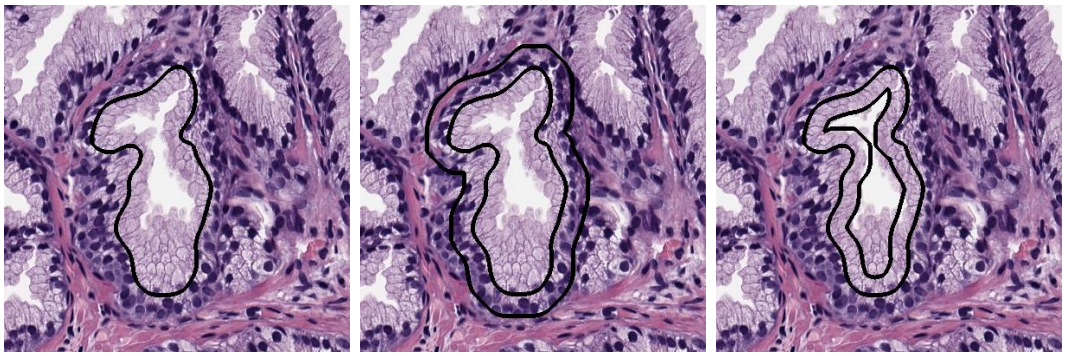
(b) LoG filtered image

(c) Thresholding at zero

(d) Thresholding at -0.001

Figure 3.9: Thresholding at a lower value improves gland detection: When a gland's boundary are broken, contours leak into stroma (Arrow in (c)). In LoG filtered image, the interior gets high negative value, while the broken gland boundary gets scores close to zero (arrow in (b)). When we threshold at a lower value, the leaks are reduced (arrow in (d)). Thresholding at a lower value does not affect contours of well formed glands.

contours. In the smooth parts of the image such as stroma or background, the response to LoG filter is close to zero. As a result, these regions have many contours that are not likely to be a gland. These contours vanish without harming any of the gland contours when we threshold at a slightly lower value than zero. An additional advantage we observed of thresholding at a slightly lower value than zero is that it reduces the leaks of gland contours into stroma (Figure 3.9). The leaks occur when the gland's boundary is broken. In the LoG filtered image, the interior of the gland has values much smaller than zero, while the gland's broken boundary gets values close to zero. As a result when we threshold at a slightly lower value than zero, the contour continues to enclose the glands interior but is less likely to leak into stroma. Sharp edges have high gradients in the LoG filtered image and are insensitive to the actual threshold value and hence are always part of the detected contours.



(a) Gland contour (b) External Periphery (c) Internal Periphery

Figure 3.10: Detecting glands: To differentiate gland contours from other contours, we build features that capture traits of the glands. We use geometrical and color-based features of the enclosed region. We also use color distribution on the external periphery of the enclosed region (b) and the internal periphery (c). For glands, the external periphery is likely to contain dark pixels corresponding to nuclei. The internal periphery is likely to be lighter colored as it corresponds to the lumen or the cytoplasm.

The LoG filter was designed to detect changes in intensity. When used as a contour detector, it outlines all regions that are brighter or darker (Figure 3.8(d)) and hence the output has contours that do not belong to glands. We separate the

gland contours from the remaining contours by training a classifier.

For the contour classifier, we designed features that express the characteristic properties of the glands. Healthy glands are large and have a convoluted boundary due to the infolding. So we use the area, perimeter, ratio of area to perimeter squared, solidity, eccentricity, and length of the minor and major axes of the enclosed region as features to express the contour's geometry (Appendix A).

In addition, we have a set of features that express the color distribution and gradient distribution of the enclosed region. Since the enclosed region for gland contours is paler, these features should help in separating gland contours from non-gland contours. The color and gradient features are expressed using histograms as earlier. We also add mean and variance in each channel as features.

The third set of features is built to bring out the differences due to the lining of the dark nuclei of the epithelial cells on the gland's periphery (Figure 3.10(b)). We find a contour's peripheral region by dilating [GW06] the enclosed region and then subtract the interior region from the dilated region. We build color and gradient histograms from the pixels in the peripheral region to express the color difference. We also give mean and variance of each channel as a feature. We dilate the enclosed region by 2,4,7 and 10 pixel disks to get different sized peripheral regions and add each region's color distribution as features. We let boosting pick the best combination of features from different sized peripheries.

We also use color distribution on the internal periphery as features (Fig-

Table 3.1: Features to detect glands: Glands have multiple characteristic properties which we express by constructing a large number of features. With boosting's tendency to resist over-fitting and to select features makes sure that even with such a large number of features, the classifier it learns performs well.

Feature Type	Number of Features
Shape	6
Internal Color Distribution	46
Internal Periphery Color Distribution	92
External Periphery Color Distribution	188
Total	332

ure 3.10(c)). For a gland, the interior periphery is likely to have paler lumen or cytoplasm than epithelial cells. We find a contour's internal peripheral region by eroding the enclosed region and then subtract the eroded region from the enclosed region. We build color histograms from the pixels in the peripheral region to capture the color difference. Similar to the earlier color features, we build 10 bin histograms for the hue, saturation and intensity channels as well as mean and variance of each channel. We erode the enclosed region by 2 and 4 pixel disks.

3.5.2 Low grade cancer structure features

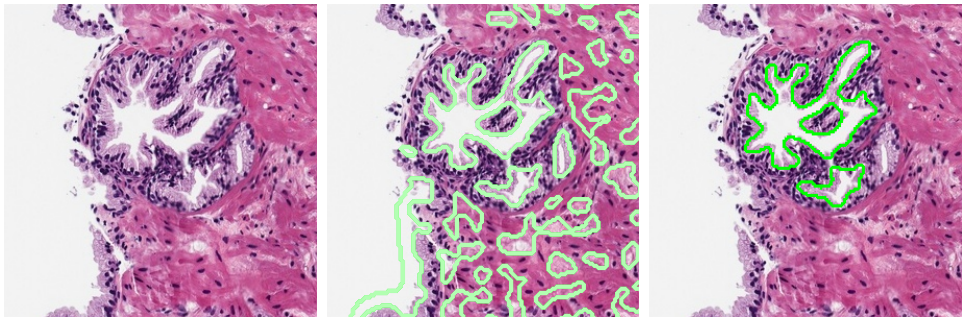
We change the gland detector method to detect low-grade cancer by finding closed contours in saturation channel using a LoG filter with scale 3 (and threshold at 0.005). We found that these contours are better at outlining the smaller low-grade cancer glands. We train a new classifier that separates contours that belong to cancerous glands from the remaining contours. The features for the classifiers are the same as those used to detect gland contours.

3.5.3 High grade cancer structure features

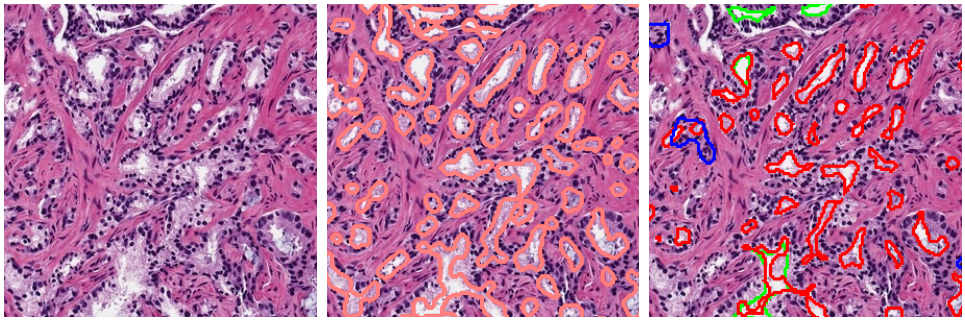
In high grade cancer, cancerous cells invade the stroma and appear completely unstructured. Even though unstructured, we can still outline the mass of epithelial cells by detecting contours in the hue channel. The invading epithelial cells are bluer than stroma and this color difference visible in the hue channel is used by the LoG contour detector to outline the invading epithelial cells. The LoG's scale was 4 with a threshold of 0.0002. Similar to earlier detectors, we train a classifier to differentiate high grade cancer contours from other contours based on the same set of features.

3.5.4 Color normalization

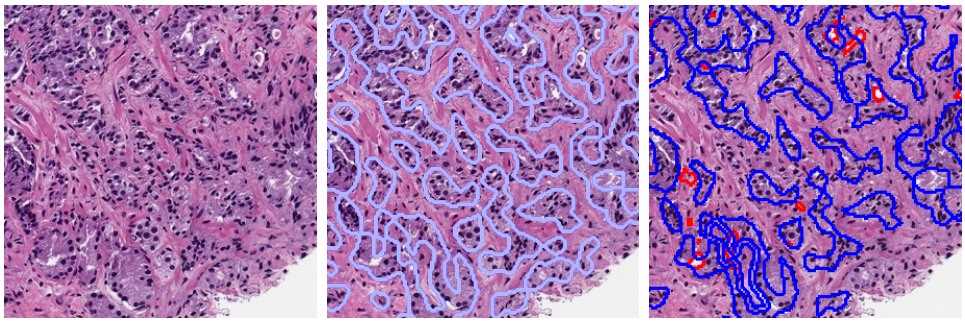
As is well known to pathologists, H&E stain's potency decreases with time. As a result, slides made on different days have different contrast distributions. To make color histograms robust against these color variations, we normalize each



(a) Tissue with a gland (b) Candidate gland contours (c) Detected Structures

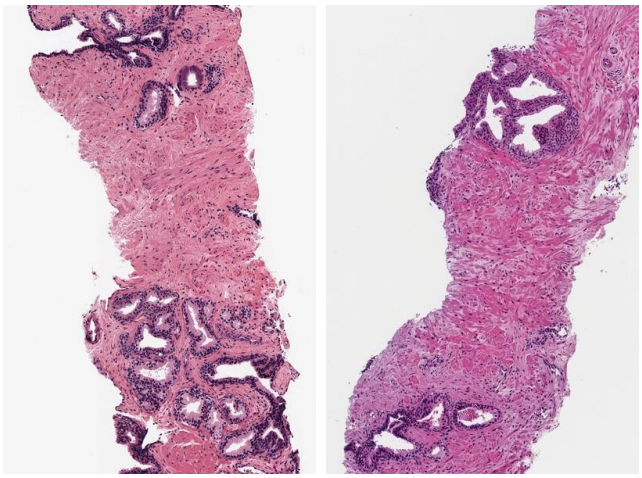


(d) Tissue with low grade cancer (e) Candidate low grade cancer contours (f) Detected Structures

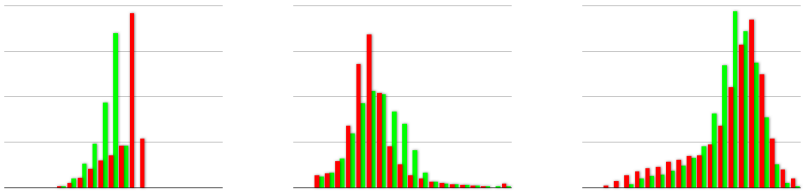


(g) Tissue with high grade cancer (h) Candidate high grade cancer contours (i) Detected Structures

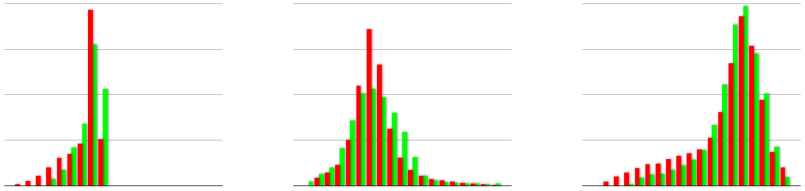
Figure 3.11: Low level structure identification: each row corresponds to one of the structures: glands, low grade cancer and high grade cancer. The left column is an example image without annotation. The middle column contains images annotated with the closed contours generated by the zero crossings of the LoG. The right column presents the subset of the closed contours in the middle column that were classified as corresponding to the corresponding structure.



(a) Image 1 (b) Image 2



(c) Hue before normalization (d) Saturation before normalization (e) Intensity before normalization



(f) Hue after normalization (g) Saturation after normalization (h) Intensity after normalization

Figure 3.12: Normalizing color distribution to counter color variations: Tissue's color change due to stain's fading (Figure (a) and (b)) which changes the hue, saturation and intensity distribution ((c), (d) and (e)). The distribution most affected is the hue (c), which can be normalized by shifting to keep the medians constant (f).

Table 3.2: Segmentation parameters for different structures.

Structure Type	Glands	Cancerous Glands	High Grade Cancer
Color Channel	Intensity	Saturation	Hue
LoG Scale	4	3	4
Contour detector threshold	-0.001	0.005	0.0002
Number of positive training examples	244	565	290
Number of negative training examples	832	1897	756
Total training size	1076	2462	1046

slides color distributions by using a two step procedure. We first remove less saturated pixels (saturation less than 0.2). Then for the remaining pixels, we shift the hue values up or down so that the median is constant for all images (Figure 3.12). The normalization moves hue values to the same range and hence counters the effect of fading of stains. We normalize the saturation and intensity channels as well by adjusting the medians. This normalization does not have a large effect on saturation or intensity histograms as their distributions are more stable. While the normalization is not perfect and may not work for large color variations, it is simple and works for color variations normally encountered.

We do color normalization for color features at the patch level classifier as well. But it did not have a significant impact on the performance. This we believe is because more than hue, intensity or saturation better indicate cancer.

But color normalization had large impact on detecting structures. The structures are detected by presence of particular tissue components in particular regions, *e.g.* nuclei on the external periphery and cytoplasm in the interior. After color normalization, for any slide the color of different tissue components like cytoplasm are more likely to be within the same range. For example, cytoplasm is always bluer than stroma, but the actual color distribution of cytoplasm may vary from slide to slide. After color normalization, the color distribution of cytoplasm is more likely to be within the same range. This improved the discriminative capability of color histogram features and lead to more accurate structure detector.

3.5.5 Converting detector outputs into features

In order to perform the high-level classification we compute the area occupied by each of the structures in the patch under analysis. We found that we get increased accuracy in the high-level detections by dividing the detections into "low confidence" and "high confidence" detections. The confidence is defined by the threshold on the score output by the Adaboost classifier corresponding to the structure. Based on the scores, we divide candidate contours into classes: high-confidence positive, all positive, high-confidence negative, all negative. The area enclosed by each class of contours is used as a different feature. These features degrade less than using simple detections because where the score is inaccurate, the "all positive" and "all negative" features are more likely to err than the high confidence ones. We compute the total area enclosed by all the candidate contours detected as glands, low grade cancer and high grade cancer. These constitute 5×3 features that are used as input for the high level classifier. Of these the area of the detected gland is probably the most informative.

3.5.6 Failures of structure detectors

The low-level structure detectors we train are not perfect. The gland detector misses glands that are heavily deformed or not closed. Such glands are found on the edge of the biopsy tissue. The detector also misses glands with concretions that form when the seminal fluid cannot flow out of the gland and solidifies into concretions. When the dark colored concretion touches the gland boundary, the contours do not align with gland boundary which confuses the classifier. The low-grade cancer detector fails to separate small cancerous glands from small non-cancerous glands that occur close to large healthy glands. The high-grade cancer detector finds it difficult to separate the chain of epithelial cell's nuclei around healthy glands from high grade cancer regions.

But the patch level classifier overcomes most of these shortcomings. For example, when many low-grade cancer glands are detected in a patch, the patch level detector will still weigh in the amount of large glands present in the patch. If the patch also has large glands, then the patch level detector would classify

the patch as benign and cancerous only if large glands were absent. Thus even though the individual detectors accuracy may be limited, the patch level classifier is accurate.

3.6 Data

We have two sets of images, one for training and one for testing. For training images, we selected 92 slides for variety so that the training set has as many of the common types of tissue structures as possible. For the training images to protect patients privacy all information was removed. Two years later, we collected 48 test images from 12 patients with 4 images per patient. Unlike training images that were selected to cover as many different types of tissue as possible, test images were selected randomly. The results on the test images results thus should represent the detector's performance in a normal setup. In the test set, 28 images had no cancer, 12 images had low grade cancer and 8 images had high grade cancer.

Both the training and test images were collected from Veterans Affair (VA) Hospital, San Diego. The images were collected at 20x using Aperio Scanscope® at a resolution of 0.50μm/pixel.

In the following sub-sections, we describe the construction of training sets based on these images for the patch based cancer detector and the low-level structure detectors.

3.6.1 Collecting Training Data for Patch Level Classifier

The training set for the cancer detector is collected by annotating the cancerous parts in the training images. The pathologist annotated the 92 training and 48 test images after examining the tissue images at 20x. The annotations were done by manually drawing a polygon around the suspicious region. For our training set, only one pathologist annotated the images.

The training set for the detector is generated by scanning the training images with 120x120 pixel patches. A patch with more than 50% of its area marked as positive is labeled positive or cancerous. Otherwise, it is labeled non-cancerous

or negative. The size of the training set generated would be unmanageable if we scanned the images with a step size of one pixel. To keep the training set size reasonable we scan the images with a step size of 30 pixels. This reduces the training set size and only removes patches that overlap. The total number of patches in training set were around 120,000 out of which around 12,500 were positive. The test images had around 58,000 patches out of which 4,600 were positive.

3.6.2 Interactively Labeling Data for Structure Detectors

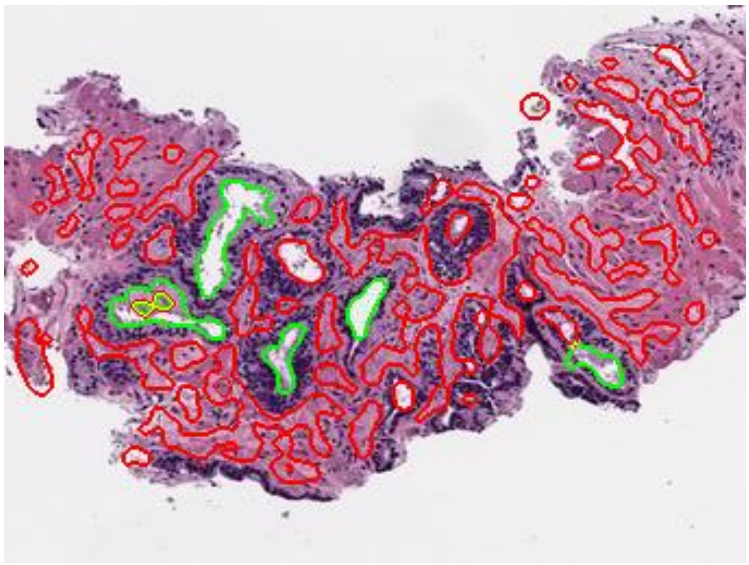


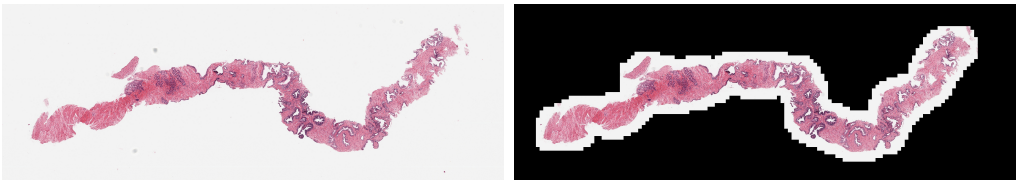
Figure 3.13: Interactive feedback to train low level structure detectors: Candidate contours classified as glands by the current classifier are marked green and others are red. To correct mistakes, the user clicks inside the contour to flip its label. The examples whose label has been flipped are added to the training set and are likely to be labeled correctly after retraining.

Low level structures in a patch such as glands are identified by classifying contours. One way to generate the training data for the classifier is to label all the contours in the image. But labeling the contours in such a manner is tedious because there are large number of candidate contours present in any image. In addition, the labeling has to be done very carefully to make sure that none of the

contours are incorrectly labeled to ensure that there are no inconsistencies in the training set.

We reduce the labor required for labeling by iteratively reviewing the classifier's performance and interactively giving feedback. We initialize this interactive feedback method by labeling a few contours in one or two images. Based on this small training set we train a classifier. We check this classifier's performance by reviewing the classification results of this classifier on all the contours in the training images. The classifier's mistakes are interactively corrected and these mistaken instances are added to the training set. To give the feedback, the training images are overlaid with color coded contours; contours that are classified as glands are colored green while the other contours are colored red (Figure 4.13). The classifier's mistakes are indicated by clicking inside the contours that were incorrectly classified. When a sufficient number of mistakes are corrected we stop the feedback process and train a new classifier. The process of iteratively improving the classifier by correcting its mistakes is repeated until we get a sufficiently good classifier. With this procedure, most of our labeling effort goes into labeling the difficult cases because the classifier becomes capable of differentiating simple contours after few iterations. To keep the interactive labeling fast we do not label the whole images but crop patches from the training images.

3.7 Results and Discussion



(a) A needle core biopsy (b) Selected Parts

Figure 3.14: Removing the background: A simple thresholding procedure removes most of the background

The images are classified by sliding the 120x120 window with a step size

of 30 pixels and classifying each window. Based on patch scores, a pixel's score is calculated by averaging the scores of all the patches that overlap with that pixel. The pixel scores are thresholded to predict the cancerous regions in the image. The average time to classify each image is around 15 min using MATLAB® on a linux workstation with 3GHz processors.

The pixel thresholds are scanned to plot detector's *sensitivity* and *specificity*. Sensitivity is the ratio of positive pixels that were predicted correctly (true positive) to all positive pixels. Specificity is the ratio of true negative to all negative pixels. The plot of sensitivity vs. 1- specificity is called the Receiver Operating Characteristic (ROC) curve. Area under the ROC curve (AUC) is one of the common metrics used to report a classifier's performance. Larger AUC indicates better performance.

The ROC results on raw images may not be representative as most of the image is white background and the background is not difficult to discriminate. The results, thus may be inflated if the images have a large amount of background. To be more indicative, the detector's performance is measured only on those parts of the image that have tissue. We find parts of the image with tissue by scanning the image using the same 120x120 pixel window and a step size of 30 pixels. If more than 80% of pixels in a patch are white (*i.e.,* have saturation value less than 0.2) then we treat that patch as a background patch. The results of such a method are shown in Figure 3.17

The ROC curves we get without the background are shown in Figure 3.15(a). Area under the curve is 0.95. We also studied improvement due to the structural features by observing the detector's performance when we use only the color and gradient features. The ROC curve for such a detector is shown in Figure 3.15(a). The AUC was 0.85. The difference in the ROC curves and the AUC show that the structural features improve the performance.

As for most applications it is important that the detector does not miss out on any cancerous parts the false-negative rate is much more significant. The performance of our detector for the low false-negative rate is shown in Figure 3.15(b).

While diagnosing cancer, each cancerous region is equally important inde-

Table 3.3: Area Under the ROC Curves for training and test data set for different evaluation criteria.

Type	Training	Test
Only Color and Texture Features	0.9454	0.8664
All Features	0.9716	0.9467
Patches	0.9322	0.9309

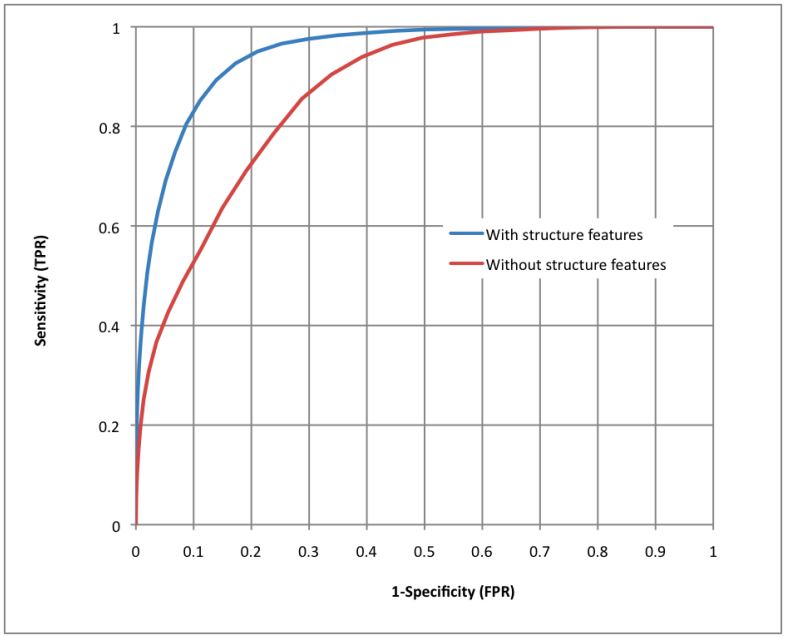
pendent of its size. The ROC curve evaluated on pixels is not representative of performance on regions because larger regions dominate the true-positive rate. To evaluate the detector's performance on regions, we define the true-positive rate using regions. We say that a region is correctly identified by the detector if more than half of the pixels in the region get scores more than the threshold. With this criteria, the true-positive rate is the fraction of cancerous regions that are correctly identified. The false-positive rate is defined as earlier; for a threshold it is the fraction of non-cancerous pixels detected as cancerous. Figure 3.16 shows this ROC curve. The AUC is 0.91.

We also evaluate the detector's performance by treating each patch as an individual example i.e., how accurate is the classifier at classifying just the patches. For both the training and test images, the AUC is 0.93 (Table 3.3).

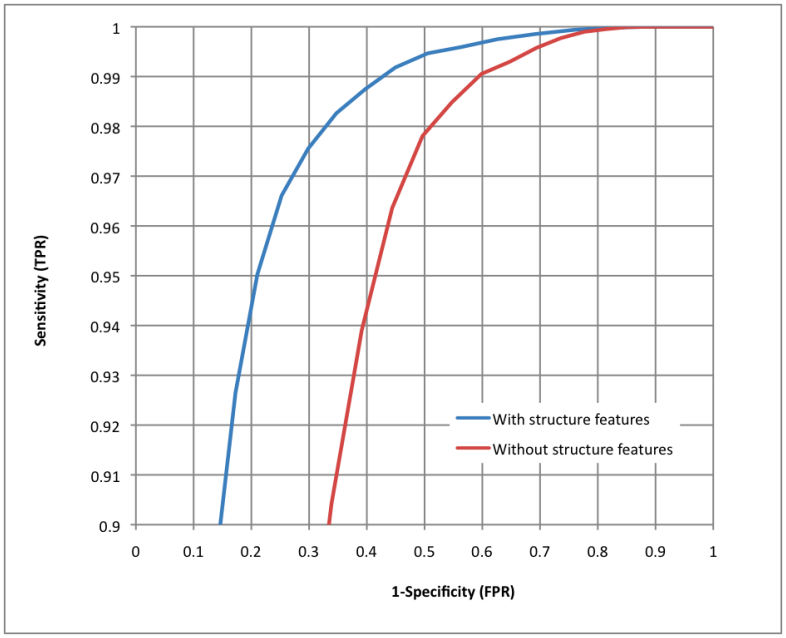
We scanned the test images to look for cancerous regions with low scores (Figure 3.18) and benign regions that got high score (Figure 3.19). To correct these mistakes, we need to develop new features because visual cues distinct from those captured by current features separate the mistaken regions from oppositely labeled regions.

3.8 Challenges

In this section, we list few of the challenges for future detectors.



(a) ROC Curve



(b) Detailed ROC Curve

Figure 3.15: ROC curves when the classifiers are evaluated on their accuracy in classifying pixels.

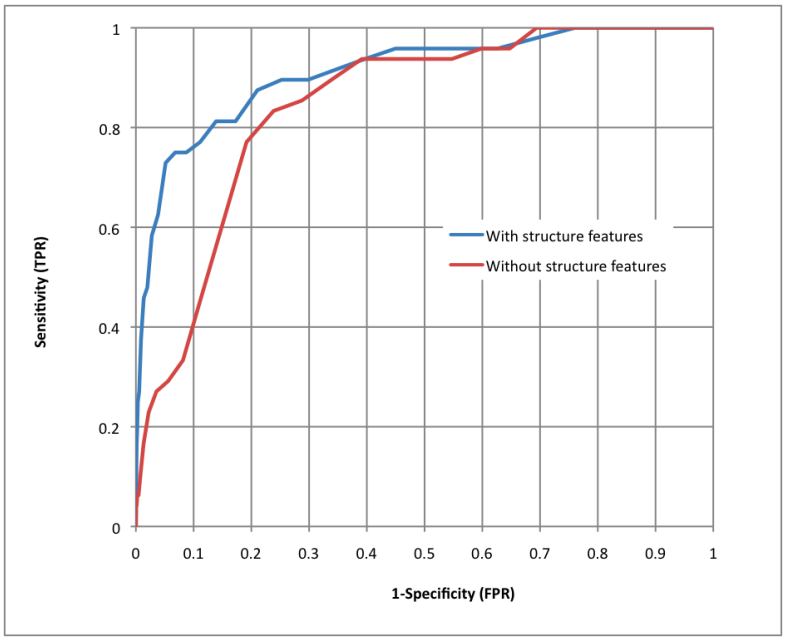


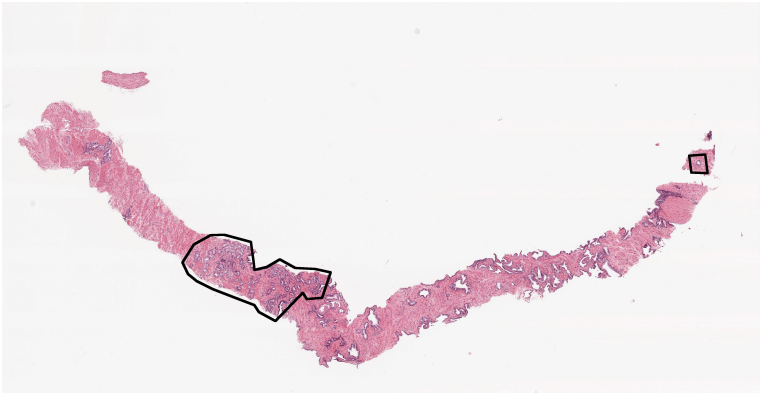
Figure 3.16: ROC curves when the classifiers are evaluated on their accuracy in classifying regions.

3.8.1 Biological variety

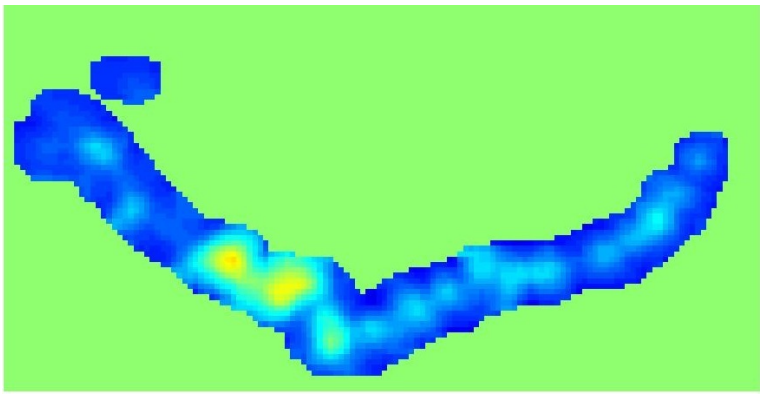
Prostate tissue is complex with many different structures. Out of the large number of the structures in prostate, we detect only a few. But when extraction, fixing or sectioning distort these structures, it is difficult to detect even these structures. Prostate has many other structures such as nerves, blood vessels, red blood cells, atrophied glands, concretions, inflammation and muscles. We believe to reliably detect cancerous tissue it is necessary to detect these structures.

3.8.2 Small foci of cancer

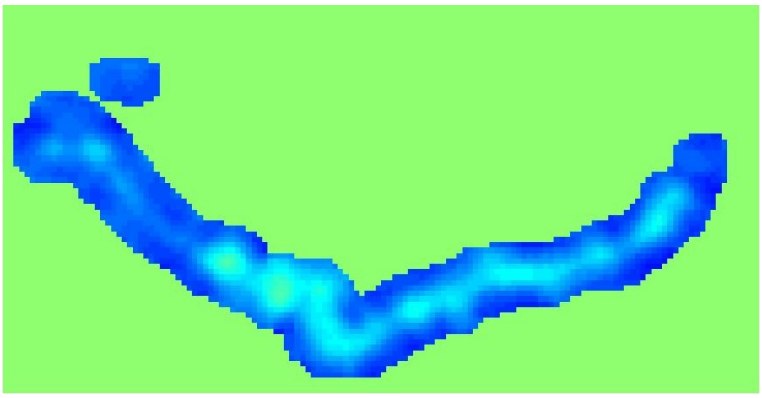
Our system is weak at detecting cancerous regions whose area is roughly equal to the patch size (*i.e.*, 15,000 pixels at 4x). But such cases are inherently difficult and even among pathologists it requires considerably experience to detect them reliably. Our training set had very few (40) such instances and the appearance of most of them differed from larger cancerous regions. We believe to detect small foci of cancer, larger data sets and more features are essential.



(a) A needle core biopsy image. The cancerous regions have been outlined in black.



(b) Final Scores using color, gradient and structural features



(c) Scores without structure features

Figure 3.17: Scores: Adding the structural features improved the detectors performance.

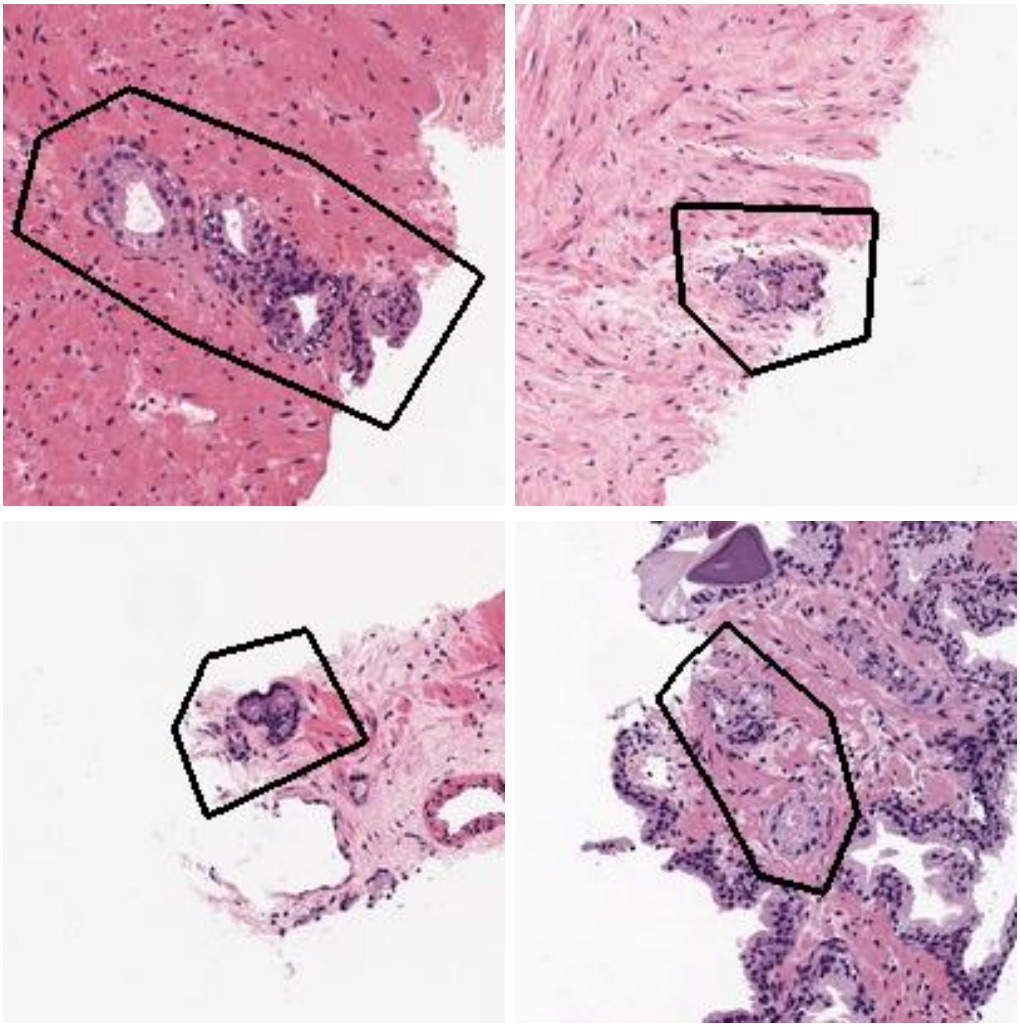


Figure 3.18: False Negatives: Examples of cancer in the test images that were given low scores by the detector. Each patch is 200x200 pixels at 4x. The black outlines are the annotations drawn by the pathologists.

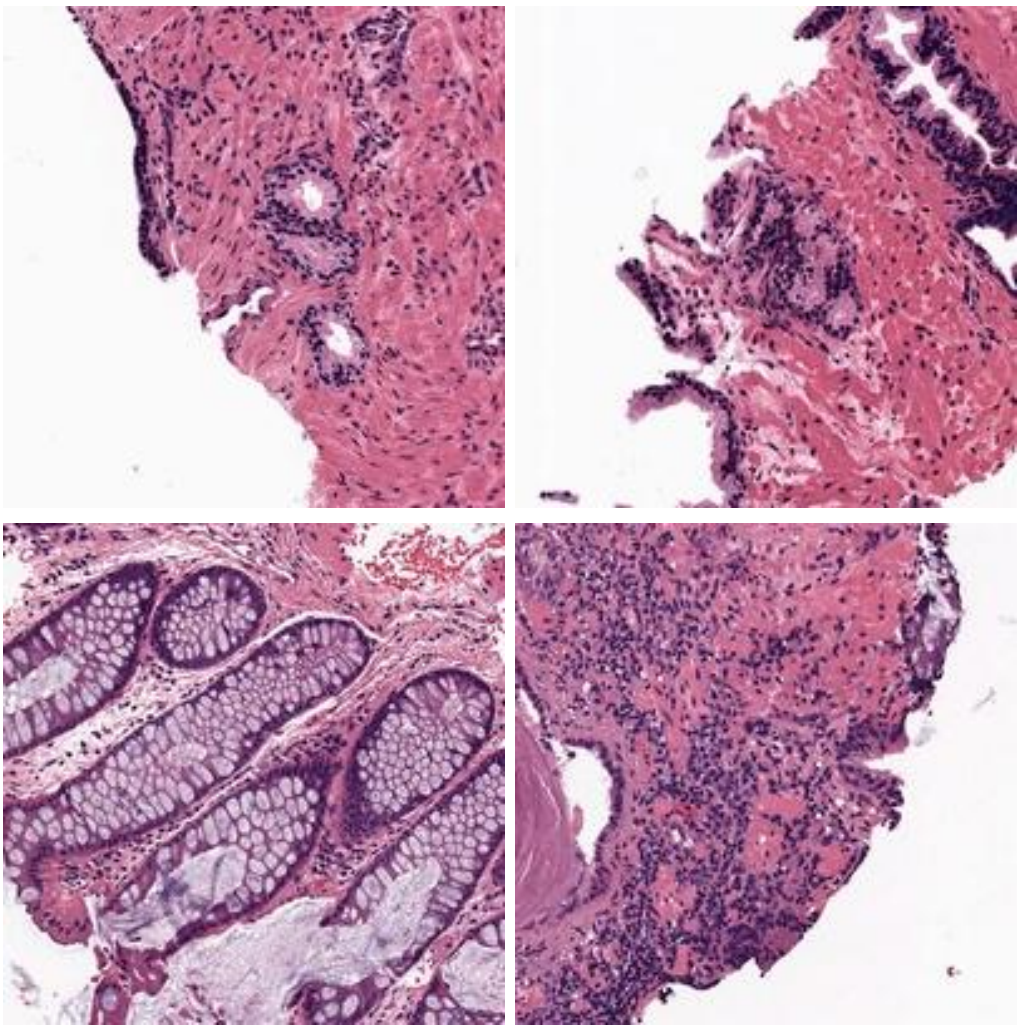


Figure 3.19: False Positives: Examples of benign formations that were given high scores by the detector. Each patch is 200x200 pixels at 4x. The third patch shows colonic epithelial tissue. We did not build features that discriminate this from cancer because such structures are not common. In the future, to build accurate detectors it would be necessary to build features that can discriminate between such uncommon structures and cancer.

3.9 Conclusion and Future Work

We developed a cancer detector for needle core biopsies of prostate. In addition to commonly used color and texture features, we used new prostate specific structural features to improve accuracy of the detector. The structure detectors were developed with few parameters by training a boosting based classifier. The improvement by structural features suggests that adding features based on other common structures can improve the performance in the future as well.

In the future, along with efforts to improve the detector's accuracy, we believe it is important to develop tools that make use of current detectors to reduce pathologist's burden. Typically, pathologists look at tissue samples for hours at a stretch and the fatigue of long hours can lead to mis-diagnoses. It would be interesting to develop applications that rearranges the slides to keep the pathologists engaged or check pathologist's fatigue by monitoring disagreements between the pathologists and machine detected cancerous regions.

For ambitious applications such as an automatic diagnostic tool, much higher performance is required. But improving performance of the detector further will become more challenging. For us, it was clear which features would improve the performance. For further improvements, developers will have to develop features that capture much subtler visual cues. Also, the incremental improvements these features would bring would be smaller because the cases where these subtle distinctions would matter would be rarer. For example, blue mucin within glands is an important clue that suggests cancer, but cancerous cases with blue mucin occur less often. Hence, a feature that detects blue mucin would be necessary to improve the detector's performance but the gain would appear smaller. In human terms, our system's performance is similar to a trainee who has been trained for few months. The next step of refining our trainee system into an expert pathologist would require much larger feature set and hence longer development.

3.10 Acknowledgements

NIH grants U54HL108460 and R01LM009520 partly supported the project described in this chapter.

Chapter 4

Phenotyping Worms

4.1 Overview

High-Throughput Screening (HTS) forms the backbone of drug discovery in the pharmaceutical industry. HTS uses automation to allow labs to conduct a large number of similar tests in an efficient, fast and cost-effective manner. The recent availability of affordable HTS equipment has made conducting such high-throughput studies more common in academia as well. Recently, researcher have started conducting HTS experiments on whole organisms, and one of the organisms that is ideal for such experiments is Caenorhabditis elegans (*C. elegans*).

For some of the experiments, *C. elegans* worms have to be imaged on agar. One such experiment is Nile Red screen. Recently, it was proposed that Nile Red fluorescent dye can be used to study life-span of organsims [OSCR09]. Worms with low Nile Red signal live longer, and animals with high Nile Red signal live shorter than wild type animals. With Nile Red as a marker, HTS screening experiment could be conducted to find the genes that impact worms life-span by silencing each gene using RNAi.

The Nile Red HTS experiments are analyzed using two types of images: standard normal light or brightfield images (BF) and fluorescence (FL) images (Figure 4.18). BF images inform about developmental and gross anatomical defects. FL images exhibit the distribution of Nile Red dye in the bodies of the worms. Till now, these images were analyzed manually. Manual analysis is labor

intensive, subjective, and error prone. Our goal is to minimize these problems by automating the analysis of worm images using computer vision algorithms for the experiments where the worms are grown on agar.

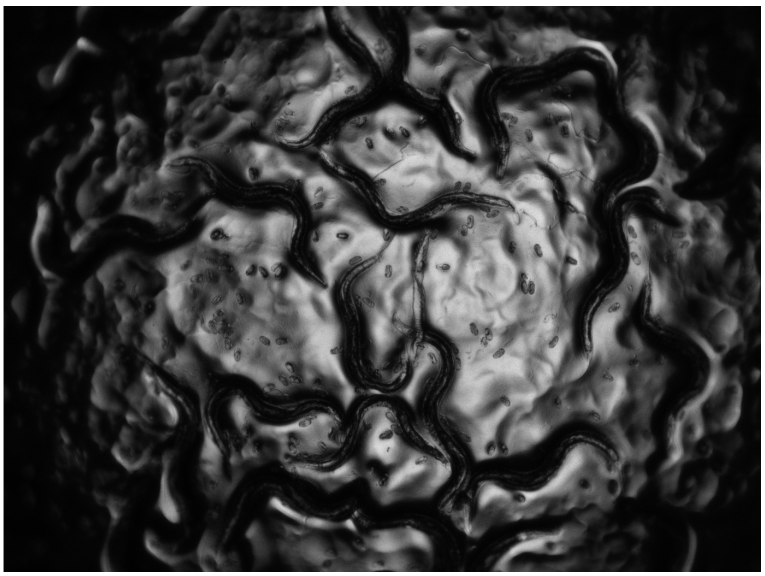
The agar background significantly complicates segmentation i.e., separating the worms from the background. Segmentation is difficult because the worms appearance is similar to tracks that they leave in the agar Figure 4.1(a).

We developed a novel method for segmenting worms from background when the worms are in agar. Our approach for segmenting worms in agar is to combine low-level image processing and scoring functions generated by machine learning algorithms. The low-level image processing performs basic operations such as finding gradient and filtering images to bring out texture. The scoring functions takes as input the features calculated by the low-level image processing and outputs a score for each pixel. High scores correspond to pixels that are inside the worms and low scores correspond to background pixels. The scoring function is generated using machine learning algorithms

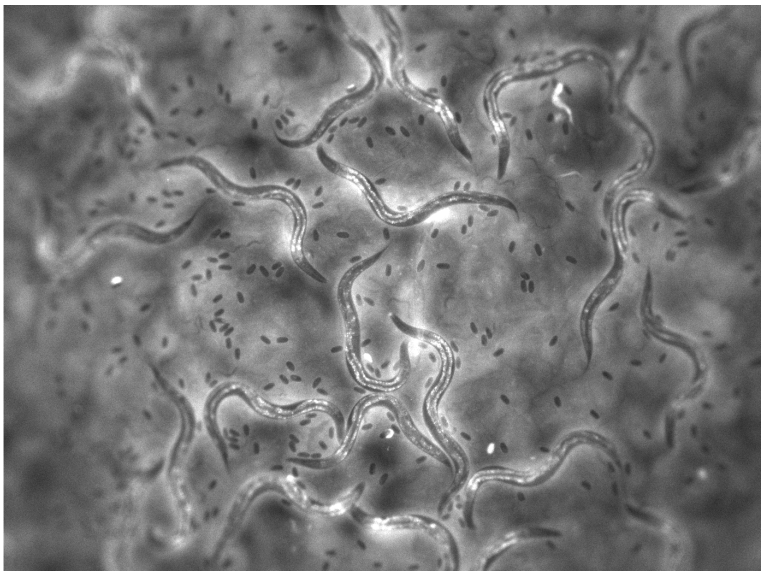
Other than the robustness it provides to the system, an important reason to use machine learning is that it can reduce the amount of parameter tuning required to adapt the method to a different lab. To adapt the machine learning based segmentation technique to a new environment, the user only has to relabel the data which can be done by anyone without complete understanding of imaging techniques and algorithms.

To the best of our knowledge, ours is the first effective computer vision method for distinguishing worm bodies from background on the agar in HTS images. It is a robust method that works effectively for a large variety of images produced in different experimental conditions.

The aim of analyzing images in a HTS experiment is to identify the phenotype present at the end of the experiment. We developed an automatic method for classifying images according to the phenotype of the worms in the image. We classify Nile Red phenotypes into 3 classes: hNR (high-Nile Red), wild type, lNR (low-Nile Red). In order to identify the Nile Red phenotype of the worm, we combine the segmentation results with information from the fluorescent image.



(a) Brightfield image of a well



(b) Fluorescent image of the same well

Figure 4.1: Examples of brightfield and fluorescent images. a) Limitations of the automated imaging setup cause the edges of the images to be out of focus and dim. The tracks left in the agar, and touching and overlapping worms complicate segmenting the worms. b) The dye Nile Red stains a lysosomal compartments along the worms intestine. The phenotypes high Nile Red, wild type or low Nile Red are distinguished based on the intensity and distribution of the fluorescent signal.

Automated identification of phenotypes is typically very simple when processing images of cell-cultures. The average intensity of the fluorescent dye within each cell provides the required information. We tested whether the average intensity of the dye can be used to identify Nile Red phenotypes in *C. elegans*. We found out that the average intensity of Nile Red within the body of the worm has a very weak correlation with the known phenotype (Figure 4.2). On the other hand the phenotype can be reliably identified by human observation. Further study revealed that the key to identifying the phenotype is that Nile Red is distributed in two stripes in each worm, corresponding to regions of the worms gut, where the lysosome-related organelles that contain the dye are concentrated. We found that it is relatively easy to segment the Nile Red stripes in the fluorescent image. By combining features of the stripes with features of the worm within which the stripe appears we were able to reliably identify the phenotype of the worm. In particular, we found that the ratio between the area of the stripe and the area of the surrounding worm is a reliable feature for identifying the worms phenotype.

To quantify the predictive performance of our system we used images from plates in which each well contained worms with a known phenotype (hNr, lNR or wild type). We evaluated the accuracy of our phenotype classifier using cross validation (CV). The error rate for discriminating lNR vs. wild type is about 1% and for hNR vs. wild type it is about 5%. Both accuracy levels are similar to the accuracy levels achieved by two biologists manually scoring the images.

The main concern in analyzing the results of screening is to ensure that no gene inactivation with a phenotype different from wild type is missed. Experiments (wells) that produce ambiguous results are usually repeated. Taking this into consideration, we conducted a second experiment in which we allowed the classification algorithm to abstain and not predict the phenotype in some of the wells. Using a classifier with abstention, we obtained the following results. For lNR vs. wild type the classifier abstained on 5% of the wells and made no mistakes in classifying the remaining 95% of the images. For hNR vs. wild type the classifier abstained on 25% of the wells and had a 1% error rate on the rest of the images. The error rate with the abstentions is significantly lower than with manual scoring.

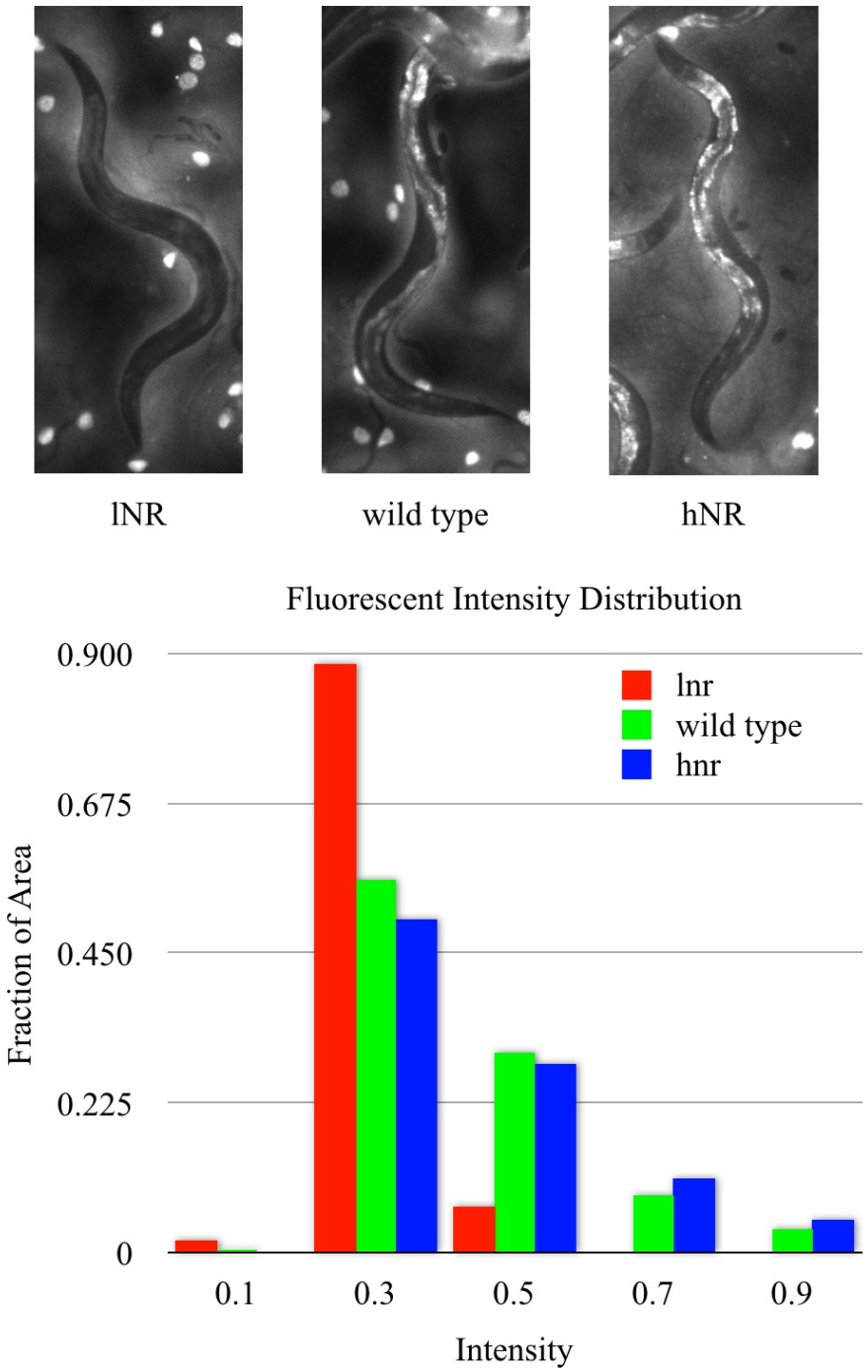


Figure 4.2: Nile Red fluorescence intensity measurements: Worm phenotypes are separated by evaluating the Nile Red fluorescent signal within the worms. The fluorescent intensity distribution inside the worms is similar, which suggests that the differences between phenotypes are subtle and cannot be captured by measuring the fluorescent intensity.

4.2 Related Work

Previous work on *C. elegans* that analyzed adult worms was done on time-lapse movies of single worm [FCW+04, CMM+05, HS06, CFS06] or multiple worms [FBB06, FBB07, TBMD08, RJB+08, DBB98] in a Petri dish. There is also related work on adult *C. elegans* using high-resolution confocal microscopy images of individual worms. The worms are straightened [PLL+08] to create a digital atlas of individual cells of the *C. elegans* adult that is used to study cell lineage by segmenting nuclei [LLP+09]. Algorithms also exist for analyzing three-dimensional, time-lapse movies of the individual nuclei in a *C. elegans* embryo [BMB+06]. The most similar work to date was the *C. elegans* high-throughput screen scored by automated image analysis for a drug screen looking for inhibitors of infection, where worms that were grown on agar were washed from their bacterial food source and transferred to liquid [MCLF+09]. The image analysis was done using CellProfiler [CJL+06] using a complex image-processing pipeline.

4.3 High level design

The main technical challenge facing the automated analysis of images of *C. elegans* on agar is the complexity and variability of the images. Two causes of complexity in BF images (Figure 4.1(a)) are the tracks left by the worms and the meniscus created when the agar is poured into the wells. The curved meniscus surface allows only the central part of image to be properly lit and in focus. The fluorescent images of the Nile Red dye (Figure 4.1(b)) are less complex than the BF images because the tracks are less visible and the meniscus only has the effect of putting the edge of the well out of focus and does not affect the fluorescence. On the other hand, the signal in the FL images varies greatly with experimental conditions and across different plates within the same experiment. Our approach segments the worms in the complex but less variable BF images and then uses this segmentation to identify the Nile Red phenotype from the FL images.

The high-level design of our system is illustrated in Figure 4.3. The system consists of three components: a worm detector, a fluorescence detector and a

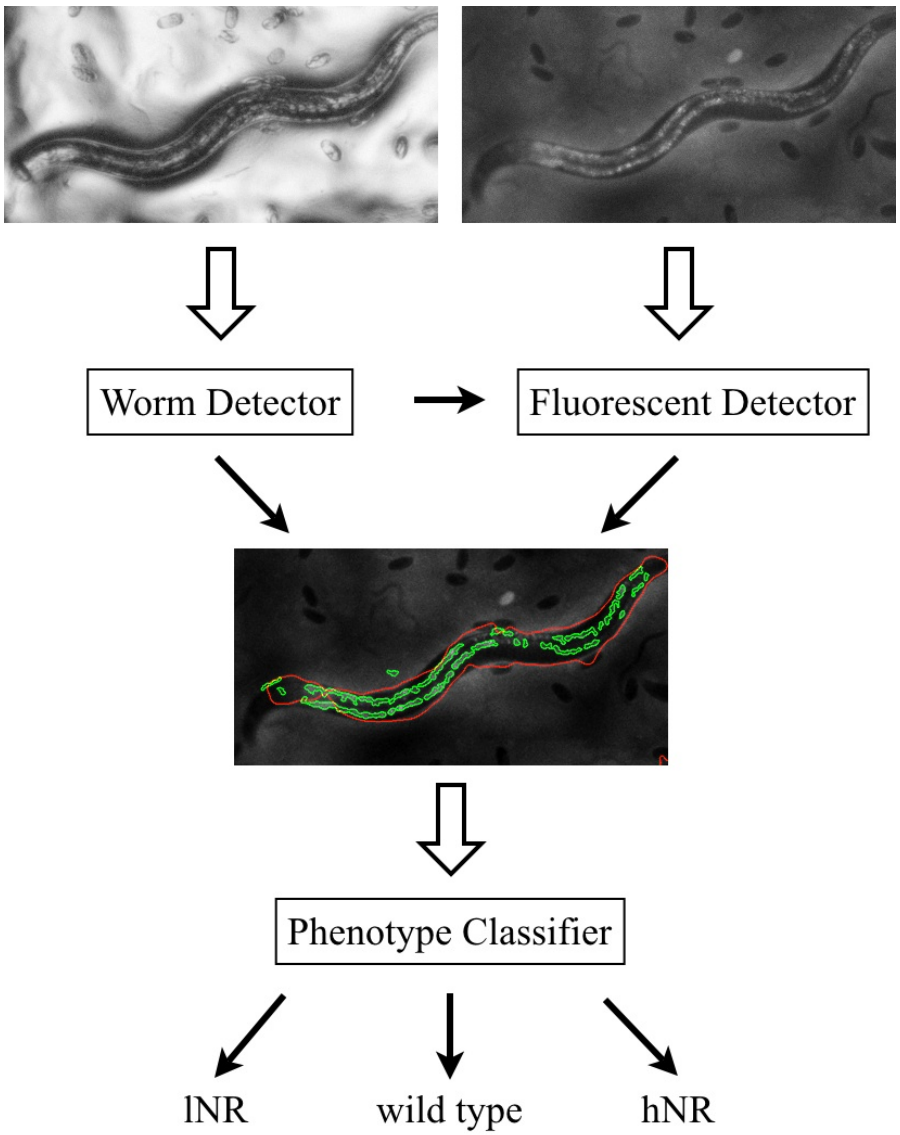


Figure 4.3: High-level design of the system: The worms are detected in the brightfield image using the worm detector. In the fluorescent image, we detect Nile Red stripes that mark the lysosome-related organelles along the worms intestine. The outputs of these two detectors are then combined to classify worms into the three phenotypes: high Nile Red, wild type, and low Nile Red.

phenotype classifier. The worm detector segments the worms using the BF image. The fluorescence detector detects the areas of Nile Red concentration along the worms intestines. Finally, the phenotype classifier classifies each well according to the Nile Red phenotype of the worms in that well: wild type, high Nile Red (hNR) and low Nile Red (lNR).

Each of the three components utilizes machine learning. The main advantage of using machine learning over manual tuning of different detectors is that labeling examples does not require an understanding of the computer vision algorithms and can be integrated seamlessly into the existing experimental protocols. This is particularly important for the phenotype classifier. The visual differences between Nile Red phenotypes in the FL images are subtle and are easily masked by the differences between experimental conditions. Therefore, in order to accurately classify the phenotype we needed to calibrate the phenotype classifier separately for each plate. In an actual screening the calibration can be automated because usually a few of the wells in each plate are devoted to contain worms of a known phenotype (no gene inactivation). Using these calibration wells to train the phenotype classifier we can then reliably identify the phenotype of the other wells without any manual intervention. In the next sections we describe the methods behind each of the three components.

4.4 Worm segmentation

Table 4.1: Comparison of segmentation inaccuracies of techniques that use local information as implemented in CellProfiler and our worm detector. The inaccuracy is calculated as the mismatch between manually labeled worms and the segmentation results over 18 images. The mismatch is reported relative to the area of the worms in the images. The worms were segmented in CellProfiler by thresholding after contrast adjustment.

Segmentation Method	CellProfiler	Our worm detector
False Positive	104 %	25%
False Negatie	35 %	30%
Total Mismatch	139%	55%

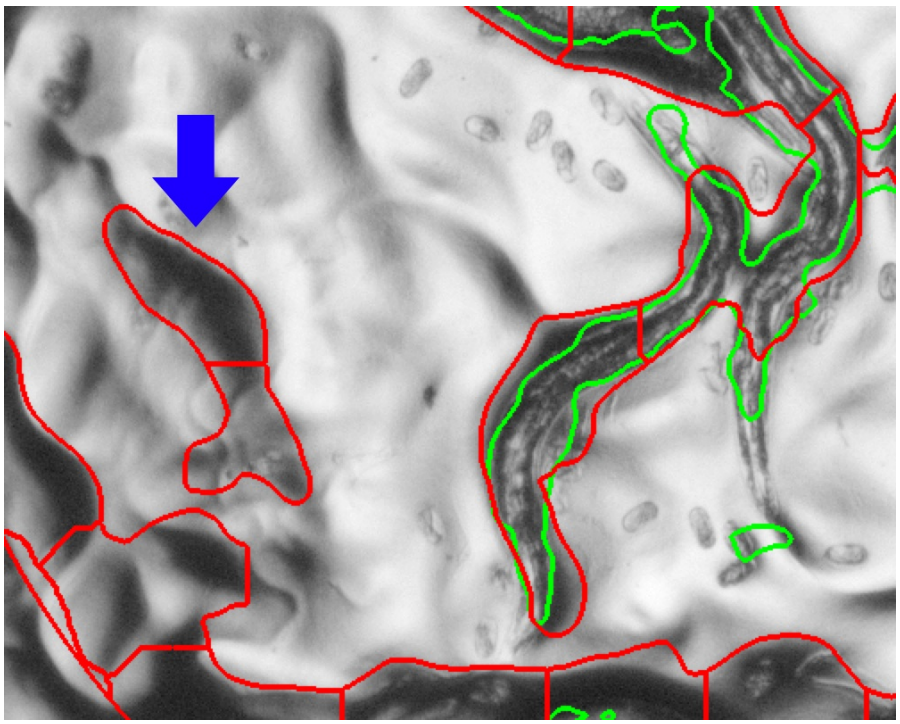


Figure 4.4: Comparison of segmentation results from two analysis systems: The red outlines are the segmentation results from CellProfiler [CJL+06] and the green outlines are segmentation results from our segmentation method. In CellProfiler, the image is first normalized and then segmented by thresholding. Such segmentation methods that use only local information mark tracks (arrowhead) as worms. Our method uses the shape of the worm and visual cues such as texture, contrast and worms edges. The segmentation results are overlaid over log-transformed image for better visualization.

As described earlier, segmentation of *C. elegans* worms in agar is a very challenging problem. The worms leave deep tracks in the agar and it is hard to distinguish the tracks from the worms. Another image analysis program, CellProfiler, whose segmentation methods perform very well for cells in culture and worms in liquid, encounters severe problems when used to segment worms in agar as can be seen in Figure 4.4 and Table 4.1. In order to segment the *C. elegans* worms reliably and accurately, a large number of visual cues must be combined. We do this by constructing a scoring function that identifies short segments of the worm.

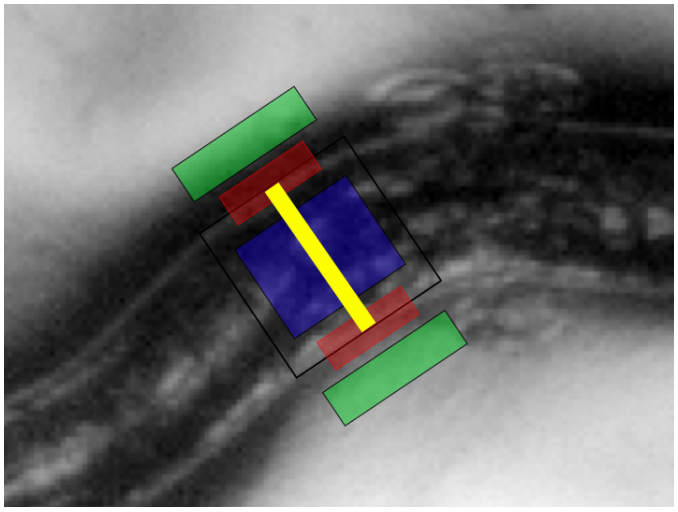


Figure 4.5: Example of a worm segment. The features for detecting worm segments are built by measuring responses of filtered images inside the rectangles.

We represent the shape of a worm using a sequence of line segments that are orthogonal to the worms midline and extend along the width of the worm (Figure 4.5). Worm segments are relatively large visual objects and can be more accurately detected. This is because we can use more visual information to decide as compared to single pixel methods.

To detect the worm segments, we look at visual properties of regions near the segment. Figure 4.5 shows the regions that we defined to gather the visual information. If a segment is part of a worm, then the rectangular region in the middle (blue rectangle in Figure 4.5) will be darker and textured. While the rectangular regions (green rectangles) just beyond the end points of the segments will

be brighter. Regions (red rectangles) just around the end points of the segments will contain edges and hence will have higher response to first derivative filters such as Sobel filter. This way we use many visual cues including worm's shape to decide if the segment belongs to worm or not.

To build features for the segments based on these visual cues we apply different filters the brightfield image. The first filter we apply is to log-transform the brightfield image because it brings out the details. After log-transformation, we subtract the median-filtered image from the log-transformed image. We calculate the median over a large block of 25×25 pixels. As the median in such a big block is likely to belong to the background, subtracting median makes the illumination more uniform across the image (Figure 4.7). We also filter the brightfield image with Laplacian of Gaussian filter with $\sigma = 1$ (Figure 4.8). The log-transformed and illumination adjusted brightfield image is also filtered by Sobel filters (Figure 4.10). We also filter log-transformed image with Laplacian of Gaussian filter with $\sigma = 1$ (Figure 4.9). The Laplacian of Gaussian filter captures the textural differences between the worms and background while Sobel filter captures the edges of the worms. In all the filtered images, we only look at the absolute values. We also discard the top and bottom 2 percent of the filtered image and then normalize the image so that the response are between 0 to 255.

For each rectangular region (Figure 4.5) we use the distribution of filter responses as features. We calculate the distribution by using a histogram with 5 equal bins between 0 and 255 and use frequency of each bin as a feature. So for each region, we get 4 histograms corresponding to the 4 filtered images. In total, we will get 20 features for each region. Since we have 5 regions, for each segment we generate 100 features. We use histograms rather than averages because histogram features are more descriptive.

To detect worms in an image, we first detect worm segments by scanning the whole 694x518 pixel image (Figure 4.11). At each pixel we find the features for segments of different lengths and every 30° angle and score the segments. The score is given by a classifier that is learned using boosting. For each pixel we retain the highest score, corresponding to the segment with most highly scored angle and

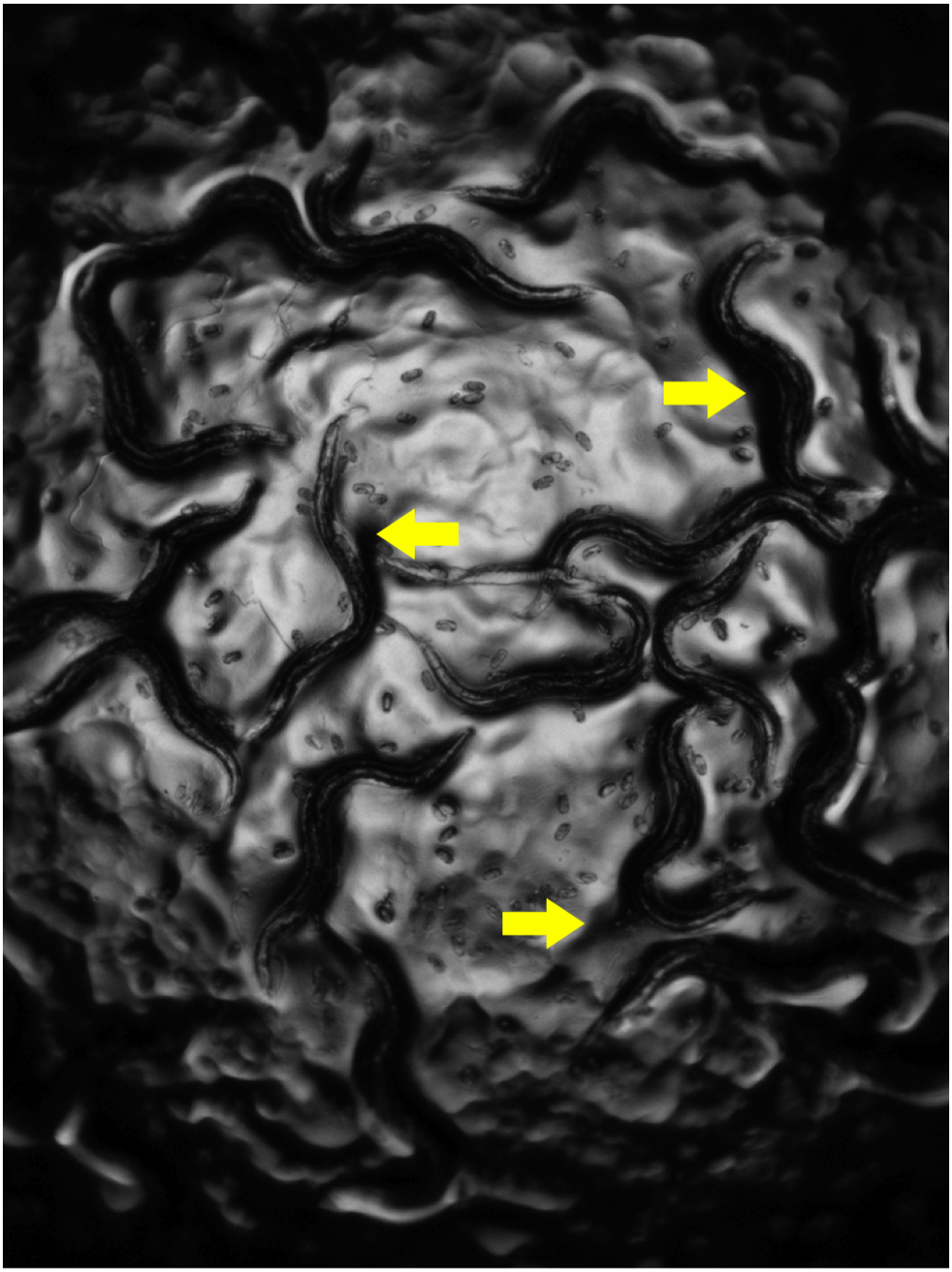


Figure 4.6: An example brightfield image: Arrowheads point to the shadows casted by the tracks. Note the completely transparent worm just beneath the middle arrow.

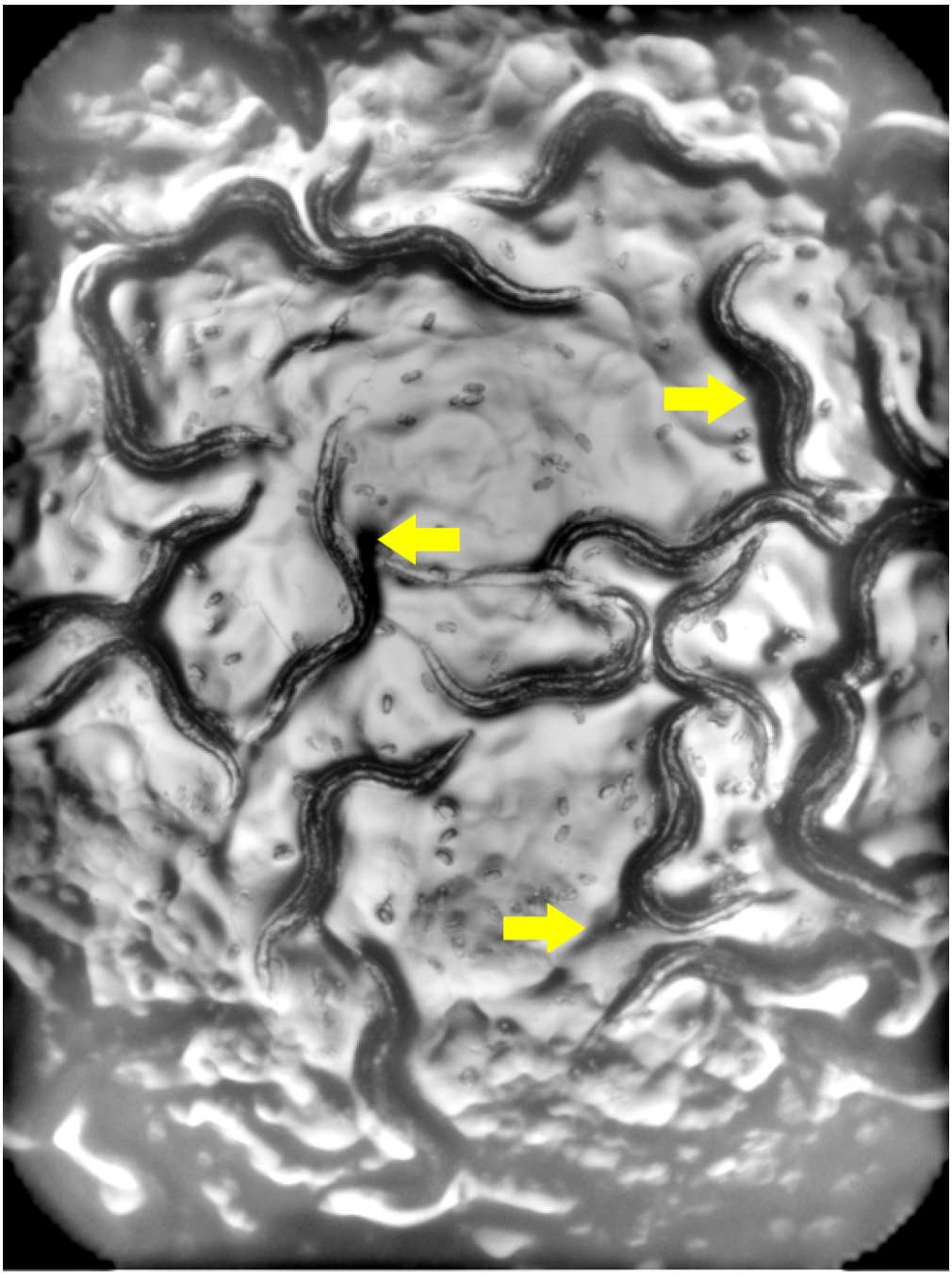


Figure 4.7: Log-transformed Image: After log transformation we see more details in the image.

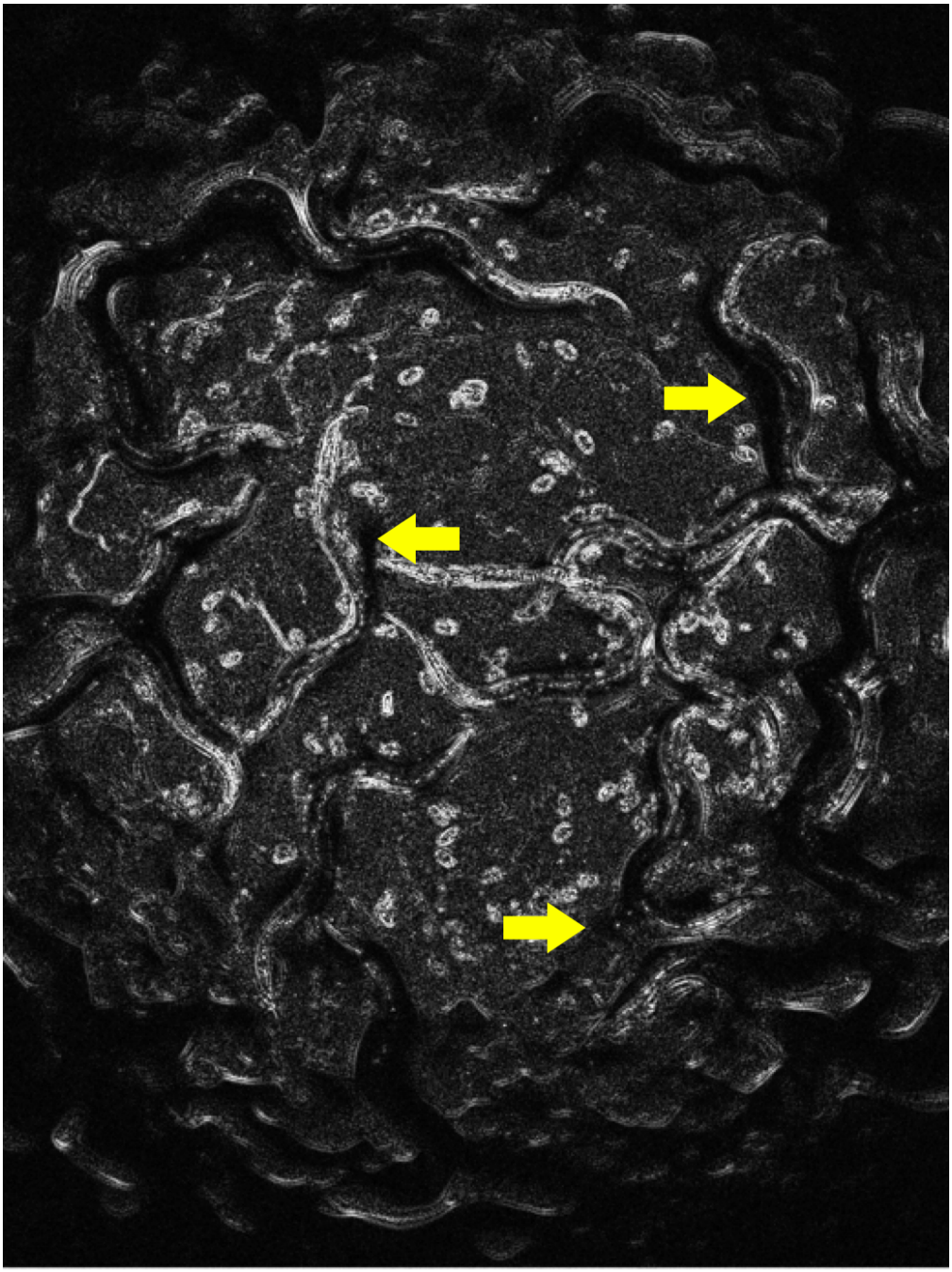


Figure 4.8: Laplacian of Gaussian Filtered Image: Laplacian of Gaussian filtered image is able to pick out the texture difference between worms and the shadows (arrowheads). This filter also responds strongly to the transparent worm just beneath the middle arrow.

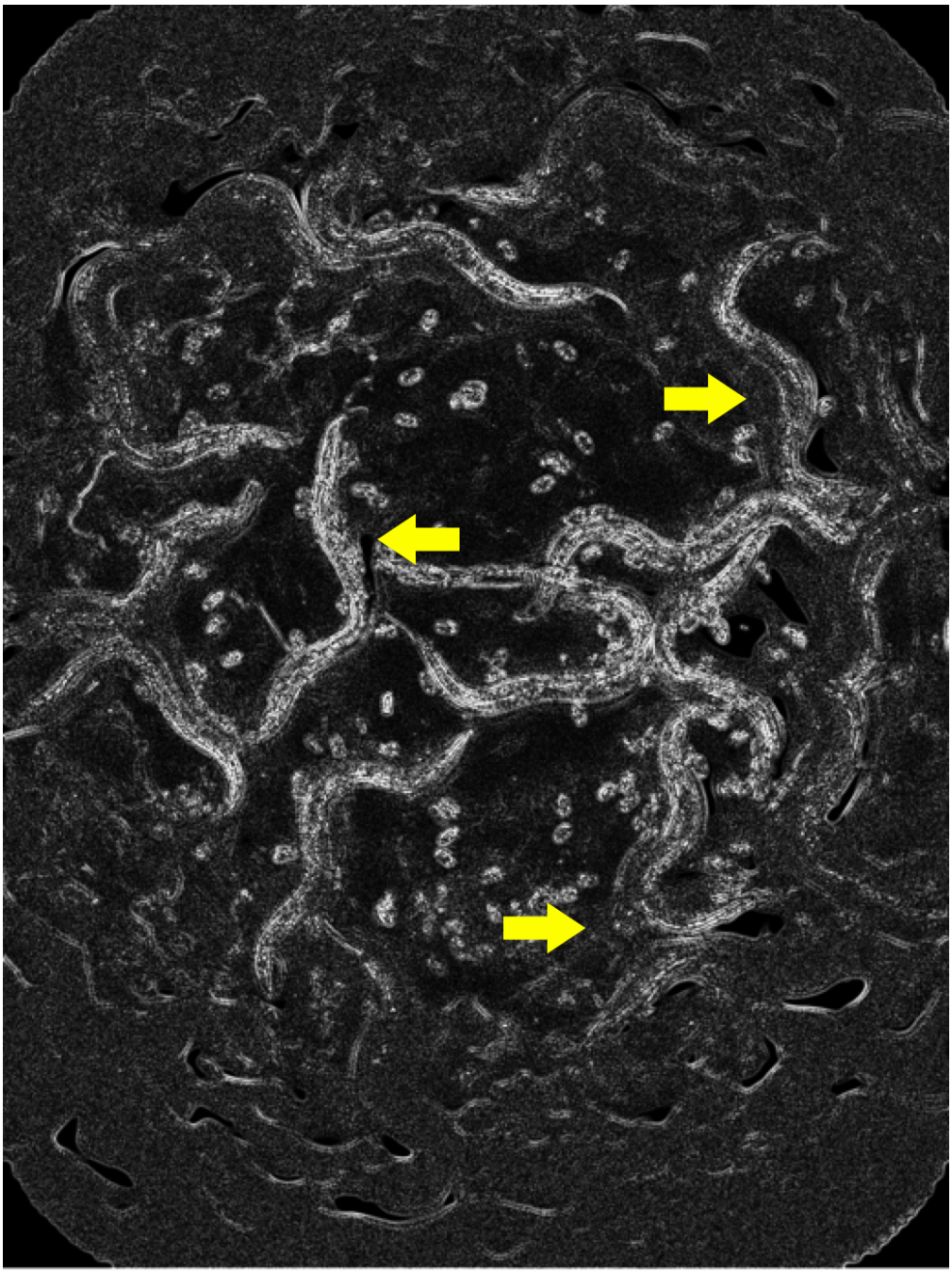


Figure 4.9: Log-transformed Image Filtered by Laplacian of Guassian: This filtered image image also picks out the differences between worms and the shadows (arrowheads).

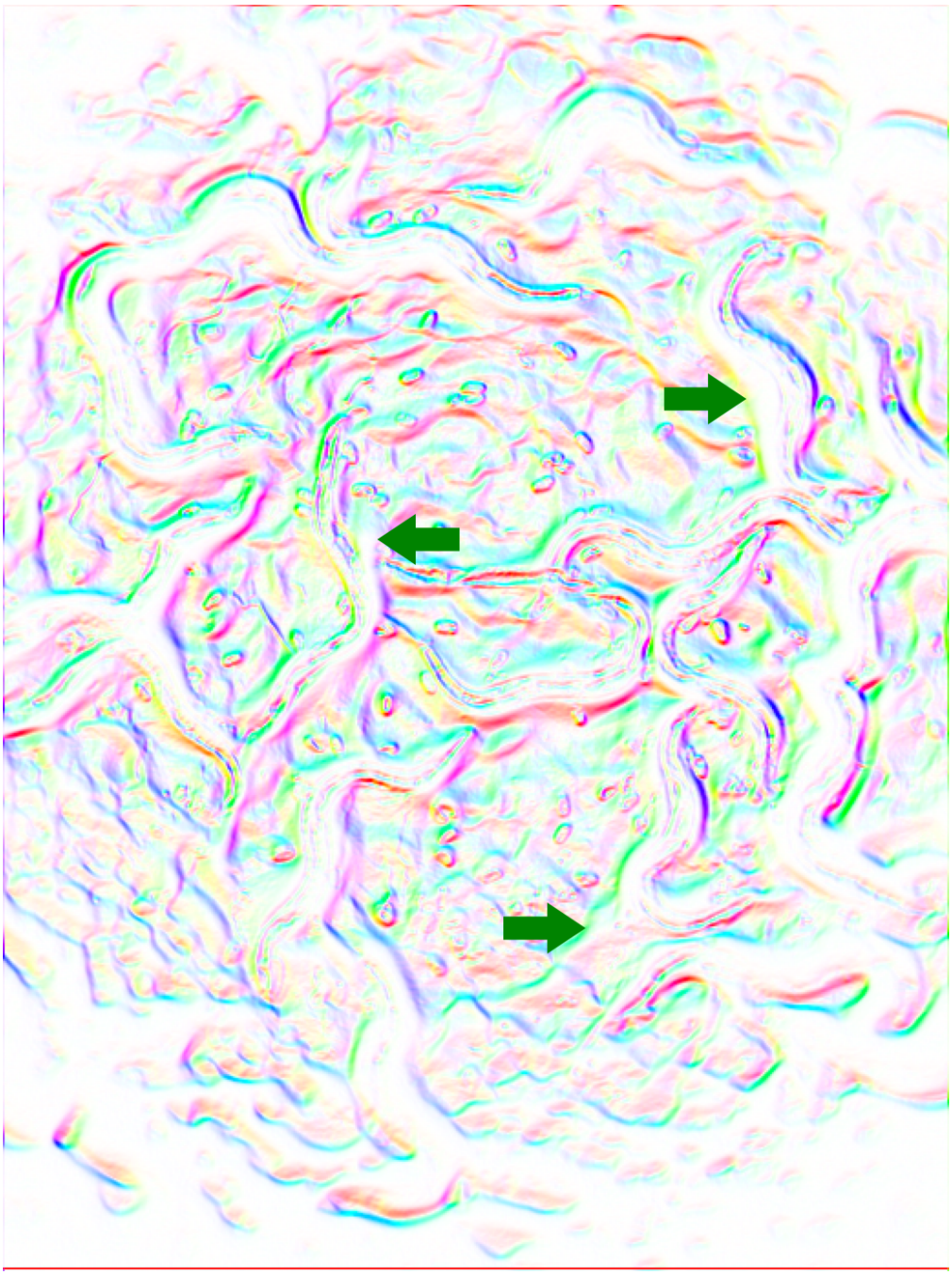


Figure 4.10: Sobel Filtered Image: Sobel filter is similar to gradient. In this sobel filtered image, the color indicates the direction of the gradient while the saturation indicates the strength. Worm's edges respond strongly to this filter.

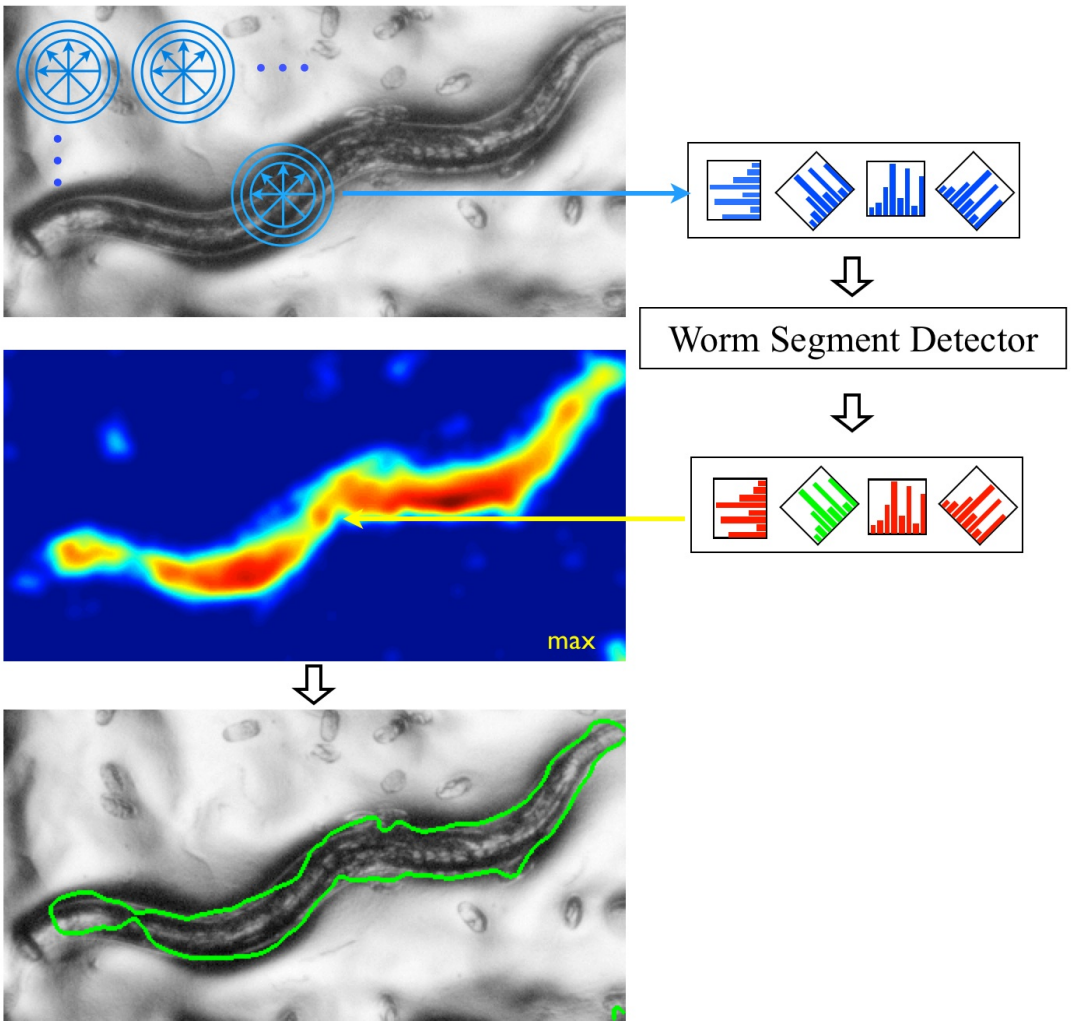


Figure 4.11: Worm segment detector. At each pixel location, we classify segments with different orientations and different sizes. The worm segment detector gives a score to each segment, where a positive score indicates that the segment is indeed a worm segment. For each pixel location, we find the segment that gets the highest score. These scores are color-mapped to generate the score image. The positive regions of the score image when outlined mark the boundaries of the worm.

length. The segmentation of the image into worms vs. background is done by thresholding these scores (Figure 4.11).

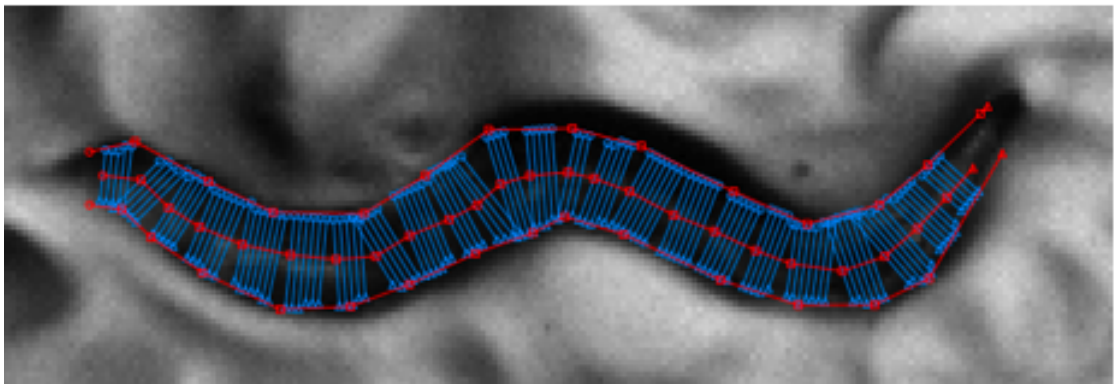


Figure 4.12: Worm segments: Worms are segmented in the brightfield image by detecting worm segments. The worm segments (blue lines) extend from one side of the worm to the other side and are orthogonal to the median line. For training the detector, the user outlines the worms (red). The worm segments are then automatically generated.

The training set to learn the segment classifier consists of positive examples – line segments that extend across a worm and negative examples – line segments that do not satisfy this condition. To generate the training set, the user annotates worms in a small number of images (around 10) by drawing lines that mark the sides and the midline of the worms. Using these line annotations, we generate an initial training set. The positive examples are the line segments defined by the human annotation (Figure 4.12) and the negative examples are randomly selected line segments of the expected length (10-30 pixels).

The initial training set does not generate a sufficiently accurate scoring function. The reason is that the initial negative examples, chosen uniformly at random, do not give sufficient representation to the non-worm segments that are difficult to classify. In order to overcome this problem we add to the training set negative line segments whose score is positive. First, we find all segments in an image that get positive scores. Among such segments, ones that are not close to segments annotated by the user are the mistakes of the current method. We add such segments to the training set (Figure 4.13,4.14) as these are difficult

examples which when added to the training set improve the performance. We treat segments whose both end points are not within a certain distance of the end points to any of the annotated segments as mistakes. From these segments that are automatically labeled negative, we randomly select 500 segments and add them to the training set. We don't add all segments to the training set as it'll inflate the size of negative examples making it difficult to train the classifier. However, we repeat the procedure of adding mistaken segments to the training set few times to add negative examples with increasing difficulty to the training set.

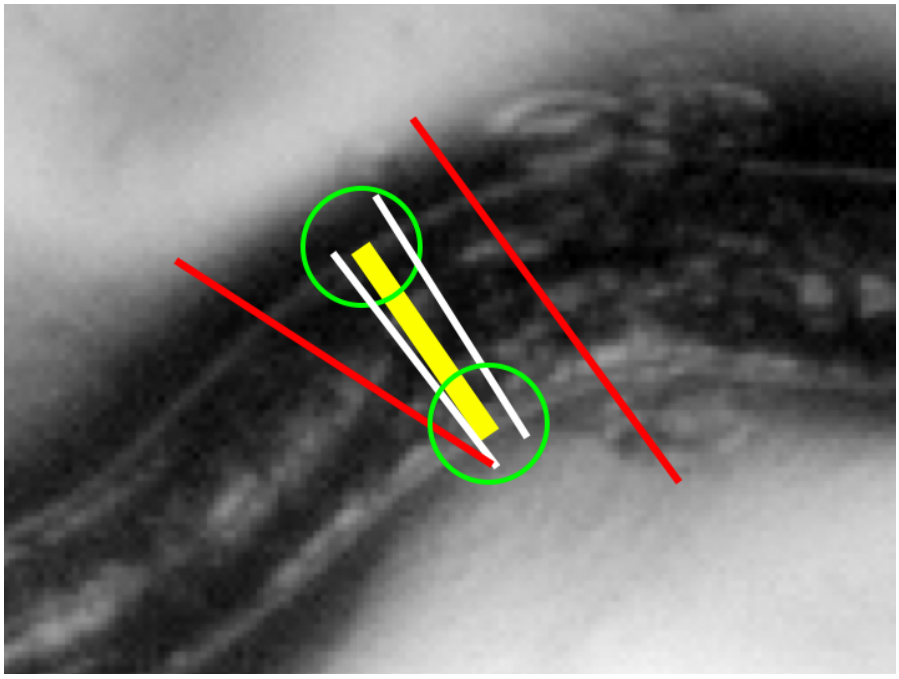
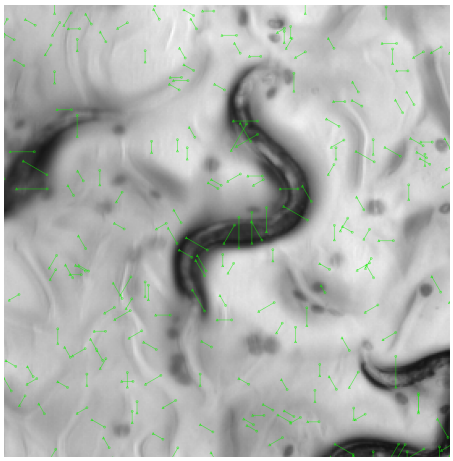


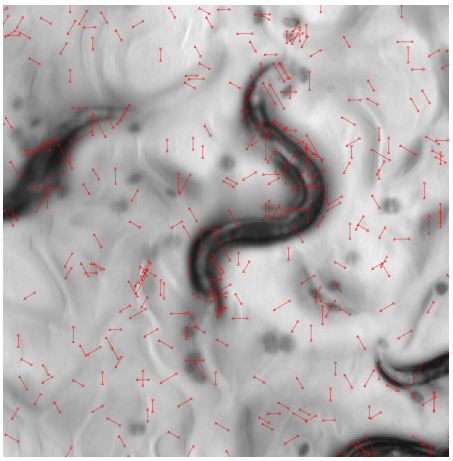
Figure 4.13: Automated Feedback of negative worm segments: Segments predicted as positive whose end points don't lie within the green circles are labeled as negative and added to the training set.

4.5 Fluorescence Detector

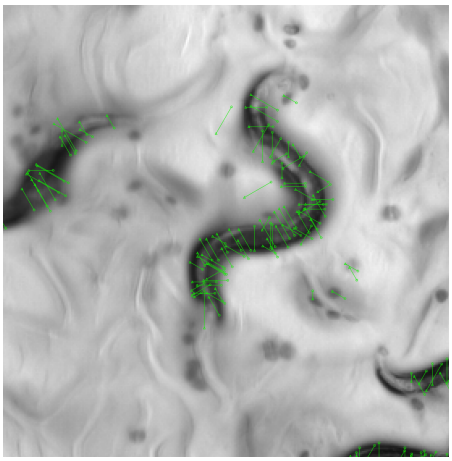
The average intensity of fluorescence inside cells has proven to be a very reliable signal for distinguishing phenotypes in HTS using cell cultures. Unfortunately, it appears that the average intensity of Nile Red in the area of the worm is



(a) Predicted positive segments

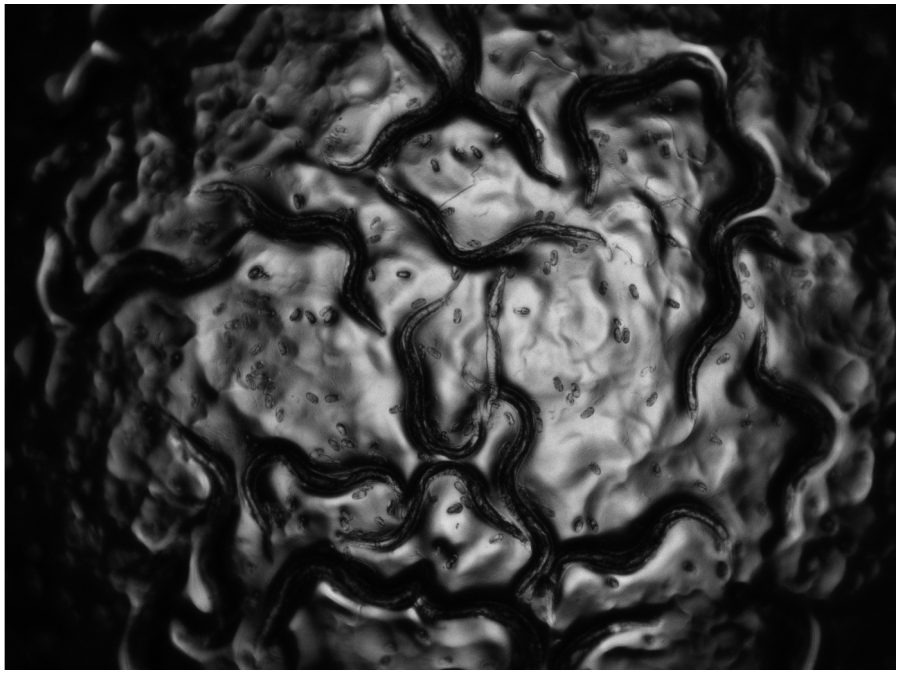


(b) Segments labeled as negative

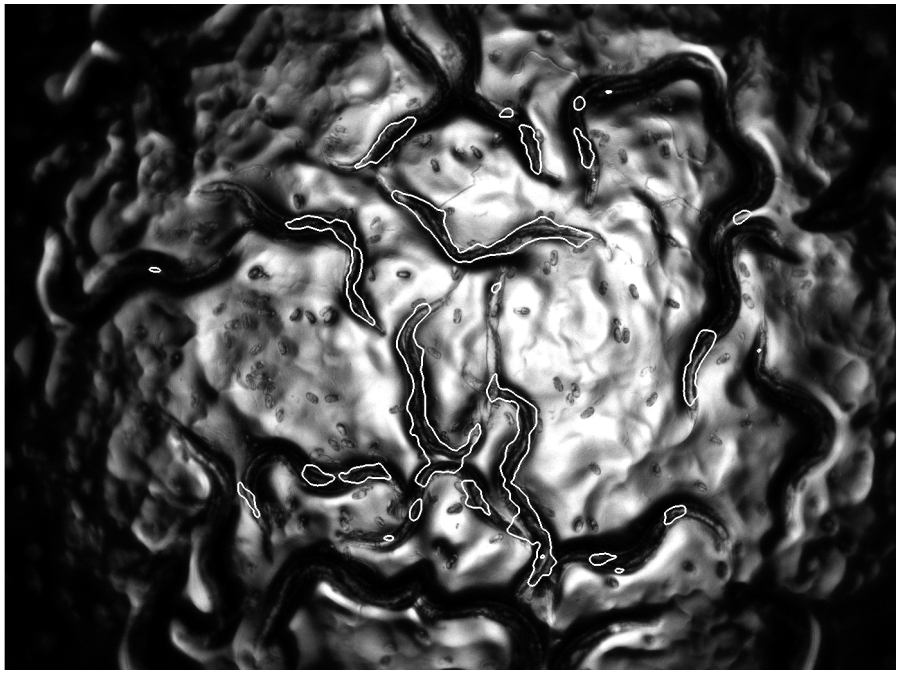


(c) Predicted positive after feedback

Figure 4.14: Improvements due to automated feedback: At the beginning we add random segments in the image as negative examples but the resulting classifier isn't accurate (a). When we add segments predicted as positive to the training set, the classifier's accuracy improves.

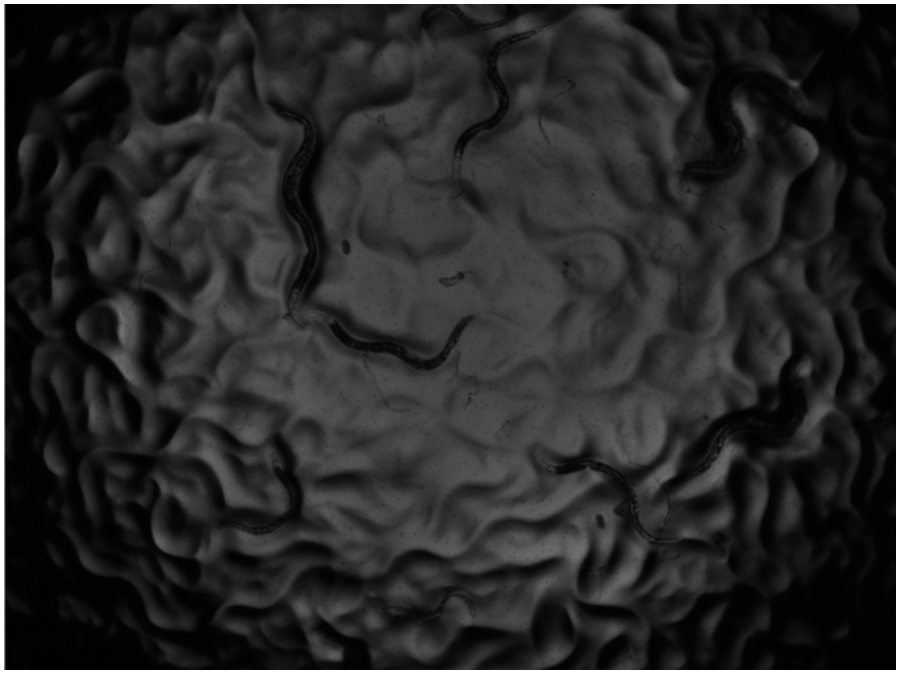


(a) Brightfield image

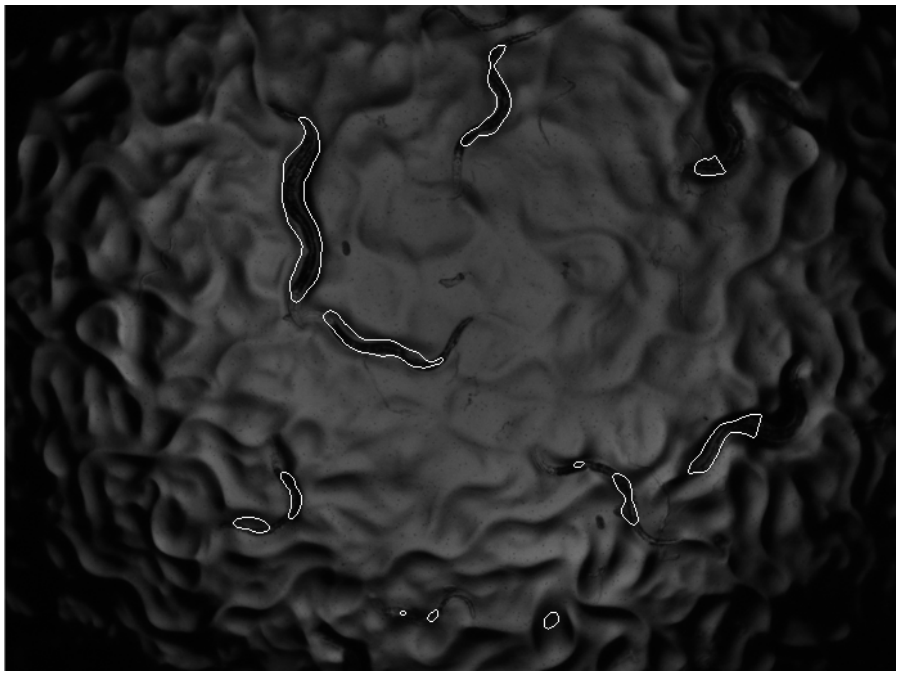


(b) Segmentation Result

Figure 4.15: Results of our segmentation technique.

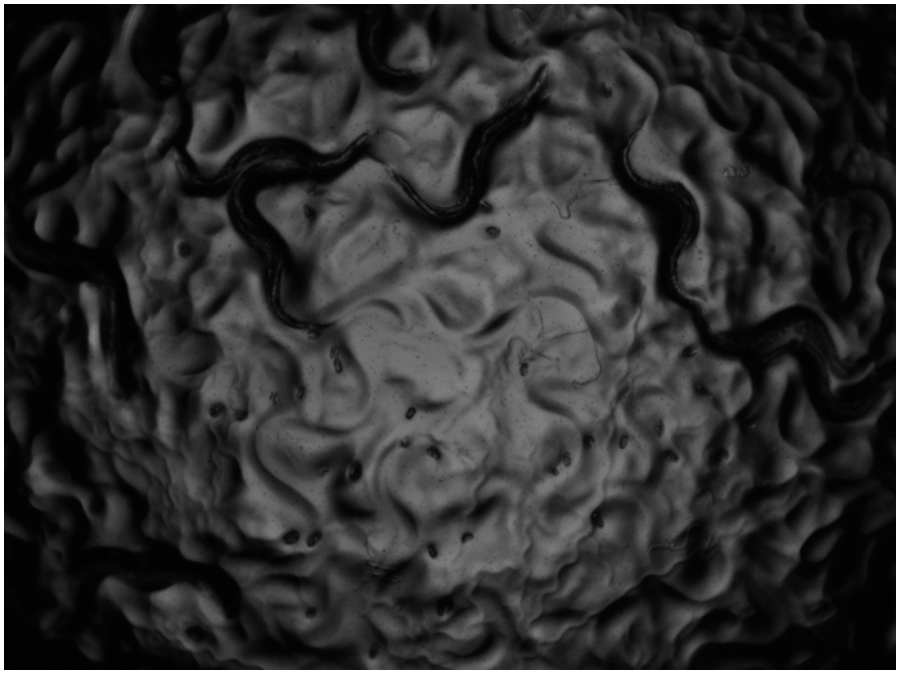


(a) Brightfield image

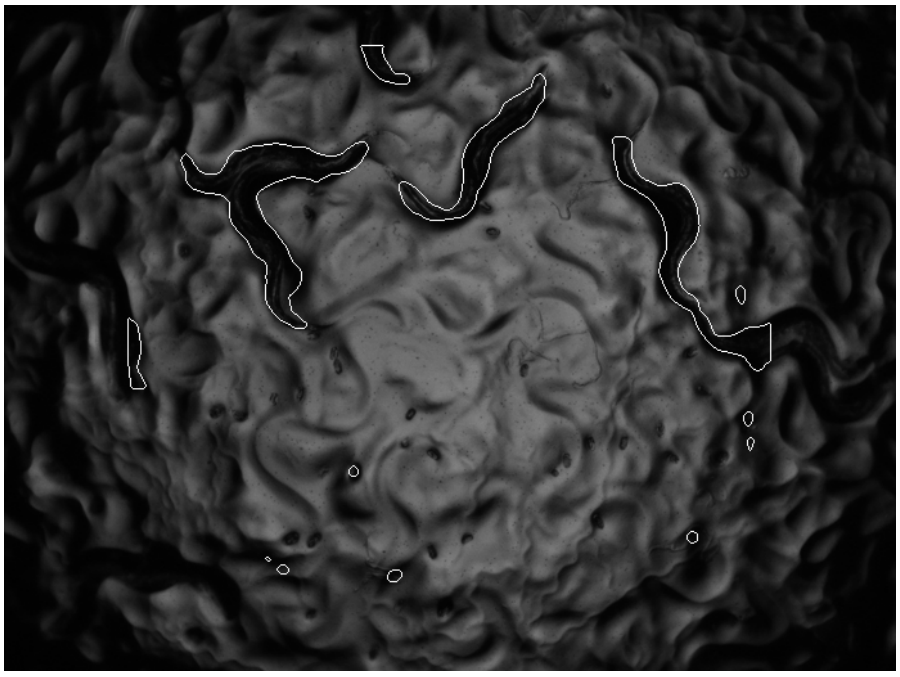


(b) Segmentation Result

Figure 4.16: Results of our segmentation technique. The white outlines in image (b) are the parts of the image that were detected as worms.

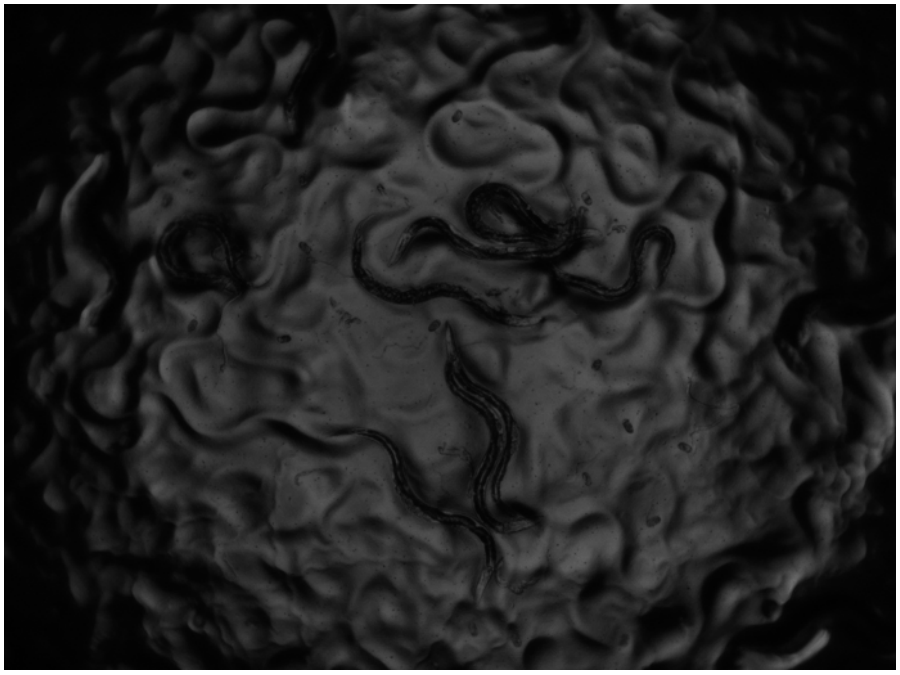


(a) Brightfield image

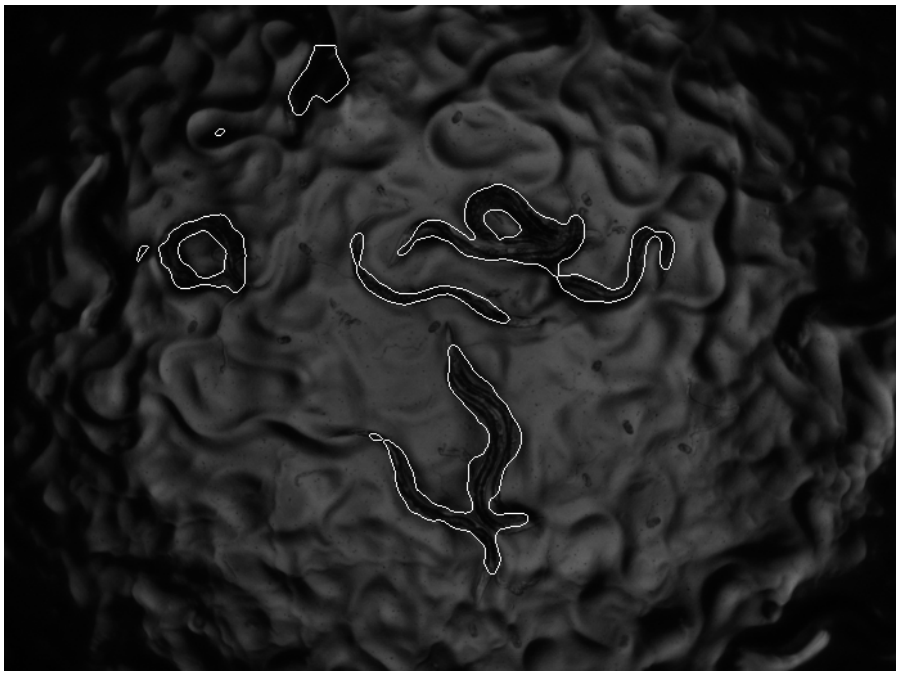


(b) Segmentation Result

Figure 4.17: Results of our segmentation technique. The white outlines in image (b) are the parts of the image that were detected as worms.



(a) Brightfield image



(b) Segmentation Result

Figure 4.18: Results of our segmentation technique. The white outlines in image (b) are the parts of the image that were detected as worms.

only weakly correlated to changes in the phenotype even when the worm segmentation is done manually (Figure 4.2). On the other hand, biologists can reliably identify the phenotypes of worms by inspecting the NR images. The reason that the average intensity is a poor predictor is that the Nile Red signal is not homogeneously distributed inside the worm, but is concentrated in lysosome-related organelles within the intestine ([SKK+07], [OSCR09]).

The Nile Red signal appears as two bright stripes along the length of the worm (Figure 4.2). The appearance of the two stripes represents two parallel stripes of intestinal cells separated by the lumen. The discrimination of Nile Red phenotypes based on the appearance of these stripes is much more reliable. In lNR phenotype, the stripes are dim and more homogeneous as compared to wild type, while in the hNR phenotype the stripes are patchier, brighter and occupy a larger area (Figure 4.2).

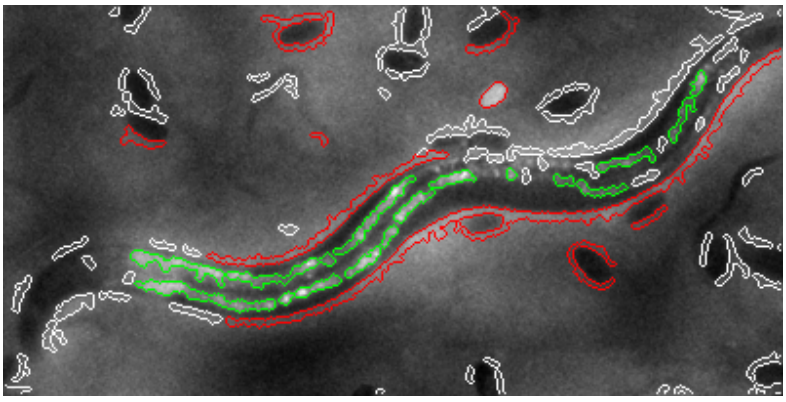
To identify the stripes, we filter the image with Laplacian of Gaussian (LoG) filter ([HM79]) and threshold the filtered image. This operation correctly segments the stripes, but it also generates other blobs, corresponding to other bright regions in the fluorescent image. We separate the stripes from the other objects using a classifier generated by Adaboost (Figure 4.19).

To classify the blob, the classifier uses area, perimeter, eccentricity, length of major and minor axis, distribution of intensity within the blob and amount of its overlap with worms detected in BF images.

The training set for stripe classifier is collected in an interactive manner. On the FL image, we show the outline of the segmented objects and then label the stripes. The features for classification are based on the geometry, shape and intensity profile of the object. Using these features we found that we had to label stripes in only 6 fluorescent images to generate a classifier with sufficient accuracy. The segmentation and detection of stripes is sufficiently stable to allow accurate analysis of images produced by experiments performed a year after the experiments used to train the stripe detector.

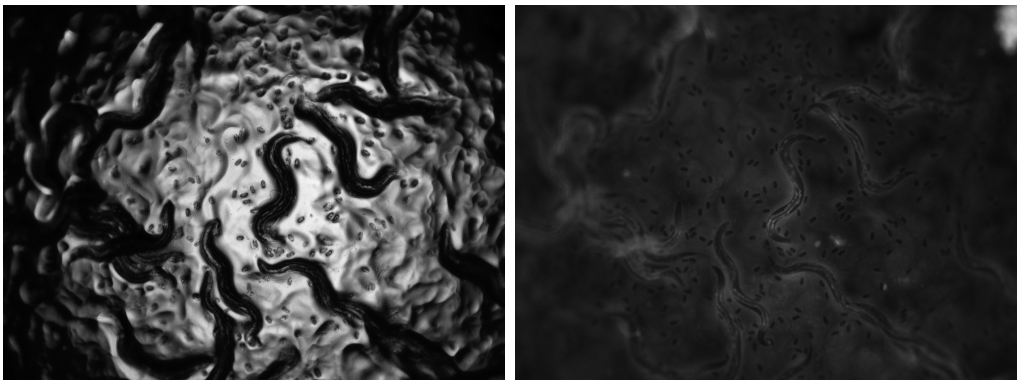
(a) Flourescent Image of a worm

(b) Detected blobs



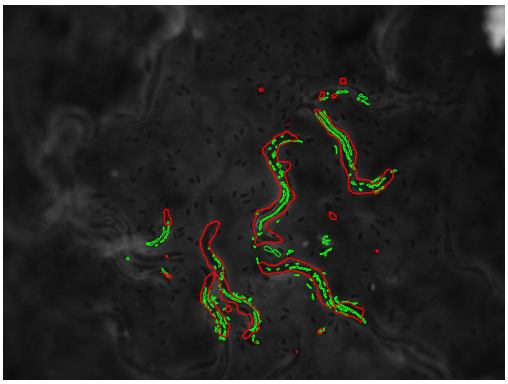
(c) Blobs labeled to train the stripe detector

Figure 4.19: Nile Red marks the lysosome-related organelles along the worms intestine and the staining pattern appears as stripes in the fluorescent image of the worm. The stripes can be outlined using a blob detector and separated using the fluorescence stripe detector.



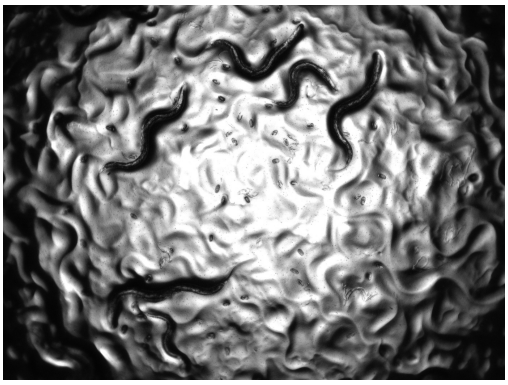
(a) Brightfield image

(b) Fluorescent image

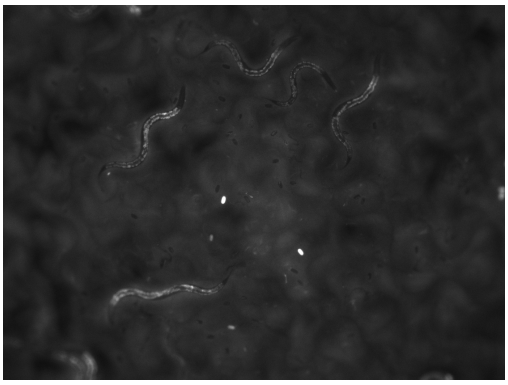


(c) Detected worms and stripes

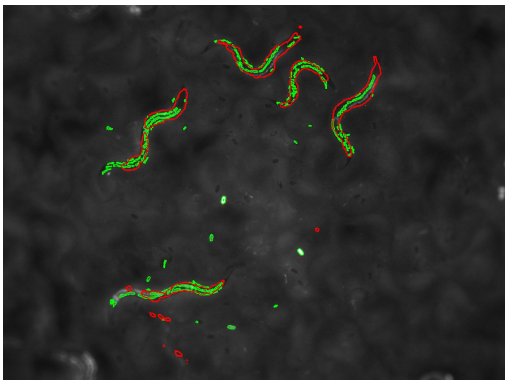
Figure 4.20: Result of worm segmentation and stripe detector: Red outlines are the segmented worms while green outlines are the detected stripes.



(a) Brightfield image

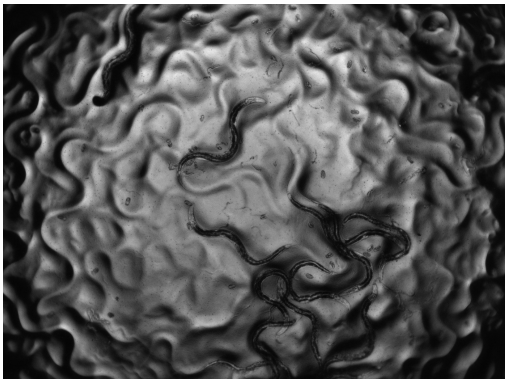


(b) Fluorescent image

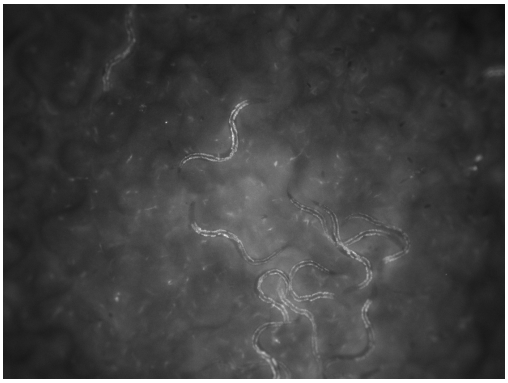


(c) Detected worms and stripes

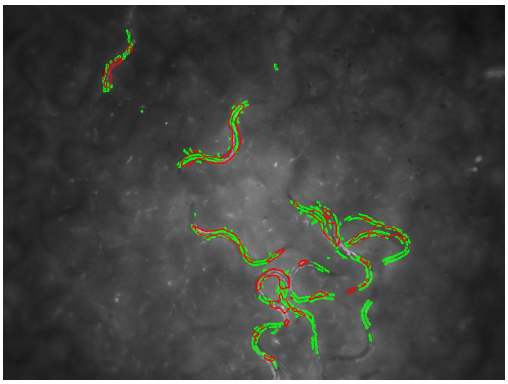
Figure 4.21: Result of worm segmentation and stripe detector: Red outlines are the segmented worms while green outlines are the detected stripes.



(a) Brightfield image



(b) Fluorescent image



(c) Detected worms and stripes

Figure 4.22: Result of worm segmentation and stripe detector: Red outlines are the segmented worms while green outlines are the detected stripes. Our worm segmentation technique fails to separate touching or clumped up worms.

4.6 Phenotype Classifier

The appearance of the worms and the appearance of the fluorescent stripes are consistent and stable enough to allow reuse of the segmentation and detection algorithms. On the other hand, the appearance of different phenotypes is highly dependent on the experimental conditions and can change between different days because of slight variations in the experimental conditions between batches of plates prepared on different days. In order to achieve reliable classification of the phenotype, we found it necessary to retrain the phenotype classifier for each plate. However, manual scoring also requires control wells in each plate, consequently the requirements for automated analysis do not increase the experimental labor or cost. Additional experiments will be required to determine whether the same classifier can be used for different plates from the same batch. As with the other components, we use Adaboost to train the phenotype classifier.

The phenotype classifier classifies each connected region obtained from worm detector into Nile Red phenotypes. The connected regions do not always correspond to individual worms, as worms that touch or overlap would be in the same region. Yet we found that the connected regions are good units to classify into phenotypes in order to find the dominant phenotype of worms in a well. To predict the dominant phenotype of worms in a well, we classify all the connected regions in the well using the phenotype classifier and sum the scores that they regions got from the phenotype classifier. We use the sign of the total score to predict the dominant phenotype present in the well. The features used by the phenotype classifier are based on the size, shape and geometry of connected region and on the properties of the fluorescent stripes inside the region.

We tested our methods on images from plates with worms whose phenotype is known. We evaluate the performance on two binary classification tasks. One task is discriminating wells with wild type worms from lNR worms, and the other is discriminating wells with wild type worms from hNR worms. We report the average classification error rate for each plate using 2-fold cross validation. In addition, we tested a classification method that abstains from predicting when the classifier has low confidence in its prediction. Our confidence rated classification method uses a

bootstrap approach similar to Bagging ([Bre96]). Instead of constructing a single master classifier, we construct a bagged master classifier by combining 7 different classifiers. Each of the 7 different classifiers is trained using a different random subset of the training examples. The bagged master classifier classifies a test instance in the following way. The instance is presented to each of the 7 classifiers. If all the 7 classifications are same, the master classifier outputs that classification. If there is a disagreement between the 7 classifications, then the master classifier abstains and does not predict. This decreases the classification error rate at the cost of having no prediction for some of the wells. This compromise is reasonable for HTS designs in which the cost associated with repeating an experiment is significantly lower than the cost associated with a screening mistake.

4.7 Results

To evaluate our automated phenotype classifier, we compared our classifier performance to the performance of trained biologists in distinguishing lNR or hNR from wild type worms. We conducted six control experiments, each on a different 96-well plate. For each plate, half of the wells had wild type worms and the other half had a particular gene inactivated to give an lNR or hNR phenotype. For four of the plates, the gene inactivation made the worms exhibit the lNR phenotype, while for the two other plates, the gene inactivation increased the Nile Red signal in the worms.

We evaluate the performance of the phenotype classifier using a 2-fold cross validation (CV) method. Each CV corresponds to a random partition of the wells in a particular 96 well plate into two parts. Each part consists of 24 images of worms treated with hNR or lNR RNAi and 24 images of wild type worms. Every CV set is used for two evaluations, each using one part as the training set and the other part as the test set. The accuracy of our prediction for each plate is reported in Table 4.2, in the column titled Automated methods error - without abstention. We compare the performance of our automated phenotype classifier to two biologists. Similar to the classifier evaluation, the biologists are shown half

the images as training images and asked to classify the other set of images. The biologists were asked to classify the test images four times for each plate with a different random partition of the images.

Table 4.2: Comparison of the automated phenotype separation method with humans. Each set of images is divided into half, where set 1 is used for training and set 2 is used as test. Human experts repeated the experiments on 4 such sets. The automated method was evaluated on 20 such sets.

Type	wild type vs lNR				wild type vs hNR	
Plate Number	1	2	3	4	1	2
Experimentalist 1 Error (%)	1.6	0.0	0.0	0.5	1.1	0.0
Experimentalist 2 Error(%)	0.0	0.0	0.0	0.0	8.9	2.6
Automated method's error - without abstention (%)	0.4	0.1	1.4	0.0	4.8	6.6
Automated method's error - with abstention (%)	0.0	0.0	0.0	0.0	0.4	1.2
Examples on which we abstain (%)	5.4	2.1	8.3	0.3	29.0	29.5

The accuracy of our phenotype separator is similar to the accuracy of the biologists. The task of separating wild type from lNR is considerably easier than that of separating wild type from hNR. In both cases, the number of mistakes made by our system is similar to that of the worse of the two experimentalists.

In addition, we computed the systems error rate when allowing it to abstain from classifying some of the wells with low confidence classification (as defined above). The entries in the column titled Automated methods error with abstention are the average number of mistakes when classifying only a high-confidence subset of images that exclude ambiguous or low-confidence images. The system labels an image ambiguous when there is disagreement among the component classifiers of the bagged master classifier. The disagreement between the component classifiers of the bagged master classifier is likely to be due to imaging or experimental issues such as improper focus or presence of very few worms in the image. The entries in the rightmost column, Examples on which we abstain, are the average number of examples on which the system abstained. We see that not predicting the phenotype on the ambiguous images significantly improves the systems accuracy and makes

it competitive with, and in most cases, better than that of human experts. The cost of this improvement is that in some cases about 25% - 30% of the images are not classified.

Researchers carrying out a high-throughput screen can significantly decrease the fraction of wells that are abstained from analysis by replicating the experiments of the wells, which in most cases is reasonable given that for most screens, experiments are done in duplicate or triplicate. The remaining small fraction of experiments that are ambiguous even with repetition could be easily screened manually.

4.8 Discussion

To the best of our knowledge, our method is the first effective computer vision method for distinguishing worm bodies from background on agar for HTS screen images. We also present a novel method to classify fluorescent images of worms. While the phenotype classifier needs to be recalibrated for each plate, the worm segmentation algorithm is very stable, it did not require any additional calibration for the results presented here. We expect that the segmentation algorithm will not require significant recalibration in future experiments as long as the appearance of the worms in BF images does not change drastically. And even if the appearance does change, worm segmentation can be adapted by annotating worms in few images.

The fluorescence detector is designed to be specific for the Nile Red marker. However, the principles behind the detector, particularly, using features unique to the fluorescent marker to develop a robust classifier, should be widely applicable to the detection of a wide variety of fluorescent signals within a worm. The accuracy of our method in identifying subtle Nile Red phenotype also suggests that automated screening of assays with clearer and stronger phenotype markers is within technical reach.

4.9 Acknowledgements

This chapter is a reprint of the material as it has been submitted to Arxiv in March 2010, Mayank Kabra; Annie L. Conery; Eyleen J. O'Rourke; Xin Xie; Vebjorn Ljosa; Thouis R. Jones; Frederick M. Ausubel; Gary Ruvkun; Anne E. Carpenter; Yoav Freund.

Chapter 5

Conclusion

In this thesis, we develop vision systems for two biomedical imaging domains. To develop the systems that were robust to the complexity and variability common in biomedical images, we used machine learning techniques combined with informative features. We show that using machine learning it is possible to develop robust systems.

Based on our current cancer detector, applications can be built to assist pathologists. Applications that recommend suspicious areas or sort slides would make it easier to diagnose cancer for pathologists. Slides can be summarized for remote diagnosis to reduce the bandwidth required to transmit the images and make telepathology more common.

With our current worm segmentation technique, automated high-throughput screening of worms on agar is within reach. This can be used to study many pressing biological questions such as metabolism, life-span and muscular dystrophy. With cheap and automated methods, such studies can become more common that'll lead to better understanding of complex biological phenomenon and hence better drugs. To ensure that many HTS studies can be done with worms, we developed our technique to be flexible and robust using machine learning.

Our approach to both the complex problems was to break down the problems into manageable pieces. We do this by defining and detecting intermediate objects. To detect cancer the intermediate objects were normal glands, smaller glands and mass of epithelial cells. To identify the worm's phenotype, the in-

termediate objects were worm segments and Nile Red fluorescent stripes. These intermediate objects were not only visibly distinct but they also had biological significance. This gave us confidence that detecting such objects would make our system robust. Still, there was no guarantee that detecting such objects would improve the performance. In the future, methods that can aid the developer to make these decisions much more confidently would reduce the number of blind guesses that are tried by the developers before he succeeds.

In conclusion, we show that while biomedical images are complex and difficult, with powerful machine learning techniques and proper system design it is possible to analyze these images. Such systems will go a long way in helping biologists, doctors and medical researchers to make sense of the seemingly infinite image data.

Appendix A

Geometric Feature for Contours

We use the area, perimeter, ratio of area to perimeter squared, solidity, eccentricity, and length of the minor and major axes of the enclosed region as features to express the contour's geometry.

- **Area** Number of pixels enclosed by the contour.
- **Perimeter** Sum of distances between each adjoining boundary pixels.
- **Solidity** Ratio of the area of the enclosed region to the area of the region's convex hull.
- **Major and Minor axes** Length of the major and minor axis of the ellipse that has the same normalized second central moments as the enclosed region.
- **Eccentricity** Eccentricity of the ellipse that has the same normalized second central moments as the enclosed region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length.

Bibliography

[AMJ+01] William C. Allsbrook, Kathy A. Mangold, Maribeth H. Johnson, Roger B. Lane, Cynthia G. Lane, and Jonathan I. Epstein. Interobserver reproducibility of gleason grading of prostatic carcinoma: General pathologist. *Human Pathology*, 32(1):81 – 88, 2001.

[AR74] Lauren V. Ackerman and Juan Rosai. *Surgical pathology [by] Lauren V. Ackerman [and] Juan Rosai.* Mosby, St. Louis,, 5th ed. edition, 1974.

[BDD+05] Charles V. Biscotti, Andrea E. Dawson, Bruce Dziura, Luis Galup, Teresa Darragh, Amir Rahemtulla, and Lisa Wills-Frank. Assisted primary screening using the automated thinprep imaging system. *Am. J. Clin. Pathol*, 123(2):281–287, 2005.

[BFA+07] Julia Breger, Beth Burgwyn Fuchs, George Aperis, Terence I Moy, Frederick M Ausubel, and Eleftherios Mylonakis. Antifungal chemical compounds identified using a ¡named-content xmlns:xlink=”http://www.w3.org/1999/xlink” content-type=”genus-species” xlink:type=”simple”¿c. elegans¡/named-content¿ pathogenicity assay. *PLoS Pathog*, 3(2):e18, 02 2007.

[BMB+06] Zhirong Bao, John I Murray, Thomas Boyle, Siew Loon Ooi, Matthew J Sandel, and Robert H Waterston. Automated cell lineage tracing in caenorhabditis elegans. *Proceedings of the National Academy of Sciences of the United States of America*, 103(8):2707–2712, 2006.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 10.1007/BF00058655.

[Car11] Anne Carpenter. Personal communication, 2011.

[CFS06] Christopher J Cronin, Zhaoyang Feng, and William R Schafer. Automated imaging of c. elegans behavior. *Methods In Molecular Biology Clifton Nj*, 351:241–251, 2006.

[CJL+06] AE Carpenter, TR Jones, MR Lamprecht, C Clarke, IH Kang, O Friman, DA Guertin, JH Chang, RA Lindquist, J Moffat, P Golland, and DM Sabatini. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10), 2006.

[CMM+05] C J Cronin, J E Mendel, S Mukhtar, Y M Kim, R C Stirbl, J Bruck, and Paul W Sternberg. An automated system for measuring parameters of nematode sinusoidal movement. *BMC Genetics*, 6(1):5, 2005.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[DAB+04] James Diamond, Neil H. Anderson, Peter H. Bartels, Rodolfo Montironi, and Peter W. Hamilton. The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia. *Human Pathology*, 35(9):1121 – 1131, 2004.

[DBB98] M De Bono and C I Bargmann. Natural variation in a neuropeptide y receptor homolog modifies social behavior and food response in c. elegans. *Cell*, 94(5):679–689, 1998.

[DdI+07] Elizabeth Davey, Jefferson d'Assuncao, Les Irwig, Petra Macaskill, Siew F Chan, Adele Richards, and Annabelle Farnsworth. Accuracy of reading liquid based cytology slides using the thinprep imager compared with conventional cytology: prospective study. *BMJ: British Medical Journal*, 335(7609):31–35, 2007.

[DFTM10] S. Doyle, M. Feldman, J. Tomasezweski, and A. Madabhushi. A boosted bayesian multi-resolution classifier for prostate cancer detection from digitized needle biopsies (preprint). *IEEE Transaction on Biomedical Engineering*, 2010.

[DKK+98] B. Djavan, K. Kadesky, B. Klopukh, M. Marberger, and C. G. Roehrborn. Gleason scores from prostate biopsies obtained with 18-gauge biopsy needles poorly predict gleason scores of radical prostatectomy specimens. *European Urology*, 33(3):261–270, 1998.

[DL01] M. G. Daniel and J. Luthringer. Gleason grade migration: changes in prostate cancer grade in the contemporary era. *Departments of Pathology and Medicine, Cedars-Sinai, Los Angeles, CA, vol. 9.3, Reprinted from PCRI Insights*, 2001.

[dS10] Natalie de Souza. High-throughput phenotyping. *Nat Meth*, 7(1):36–36, 2010.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:886–893, 2005.

[EAJE05] Lars Egevad, William C. Allsbrook, Jr., and Jonathan I. Epstein. Current practice of gleason grading among genitourinary pathologists. *Human Pathology*, 36(1):5 – 9, 2005.

[EH86] H M Ellis and H R Horvitz. Genetic control of programmed cell death in the nematode c. elegans. *Cell*, 44(6):817–829, 1986.

[FBB06] E. Fontaine, J. Burdick, and A. Barr. Automated tracking of multiple c. elegans. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 3716 –3719, 30 2006-sept. 3 2006.

[FBB07] Ebraheem Fontaine, Alan H. Barr, and Joel W. Burdick. Tracking of multiple worms and fish for biological studies. *ICCV Workshop on Dynamical Vision*, 2007.

[FCW+04] Zhaoyang Feng, Christopher J Cronin, John H Wittig, Paul W Sternberg, and William R Schafer. An imaging system for standardized quantitative analysis of c. elegans behavior. *BMC Bioinformatics*, 5(1):115, 2004.

[FHT98] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.

[Fre95] Yoav Freund. Boosting a weak learning algorithm by majority, 1995.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[FSZJKZ07] Reza Farjam, Hamid Soltanian-Zadeh, Kourosh Jafari-Khouzani, and Reza A. Zoroofi. An image analysis approach for automatic malignancy determination of prostate pathological images. *Cytometry Part B: Clinical Cytometry*, 72B(4):227–240, 2007.

[FXM+98] A Fire, S Xu, M K Montgomery, S A Kostas, S E Driver, and C C Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391(6669):806–811, 1998.

[GBC+09] M.N. Gurcan, L.E. Boucheron, A. Can, A. Madabhushi, N.M. Rajpoot, and B. Yener. Histopathological image analysis: A review. *Biomedical Engineering, IEEE Reviews in*, 2:147 –171, 2009.

[Gle66] D. F. Gleason. Classification of prostate carcinomas. *Cancer Chemother. Rep.*, 50:125–128, 1966.

[GS10] Jean Giacomotto and Laurent Sgalat. High-throughput screening and small animal models, where are we? *British Journal of Pharmacology*, 160(2):204–216, 2010.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[HBL+02] Koen Houthoofd, Bart P Braeckman, Isabelle Lenaerts, Kristel Brys, Annemie De Vreese, Sylvie Van Eygen, and Jacques R Vanfleteren. Axenic growth up-regulates mass-specific metabolic rate, stress resistance, and extends life span in caenorhabditis elegans. *Experimental Gerontology*, 37(12):1371–1378, 2002.

[HLE04] Sandra C. Hollensead, William B. Lockwood, and Ronald J. Elin. Errors in pathology and laboratory medicine: Consequences and prevention. *Journal of Surgical Oncology*, 88(3):161–181, 2004.

[HM79] E.C. Hildreth and D. Marr. Theory of edge detection. In *MIT AI Memo*, 1979.

[HS06] Katsunori Hoshi and Ryuzo Shingai. Computer-driven automatic identification of locomotion states in caenorhabditis elegans. *Journal of Neuroscience Methods*, 157(2):355–363, 2006.

[JBo10] *Java Implementation of Boosting*. http://jboost.sourceforge.net/, 2010.

[KCG+93] C Kenyon, J Chang, E Gensch, A Rudner, and R Tabtiang. A c. elegans mutant that lives twice as long as wild type. *Nature*, 366(6454):461–464, 1993.

[KV89] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata, 1989.

[LCT+06] Ben Lehner, Catriona Crombie, Julia Tischler, Angelo Fortunato, and Andrew G Fraser. Systematic mapping of genetic interactions in caenorhabditis elegans identifies common modifiers of diverse signaling pathways. *Nature Genetics*, 38(8):896–903, 2006.

[LFA93] R C Lee, R L Feinbaum, and V Ambros. The c. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. *Cell*, 75(5):843–854, 1993.

[LLP+09] Xiao Liu, Fuhui Long, Hanchuan Peng, Sarah J Aerni, Min Jiang, Adolfo Snchez-Blanco, John I Murray, Elicia Preston, Barbara Mericle, Serafim Batzoglou, and et al. Analysis of cell fate from single-cell gene expression profiles in c. elegans. *Cell*, 139(3):623–633, 2009.

[MBA+06] Terence I. Moy, Anthony R. Ball, Zafia Anklesaria, Gabriele Casadei, Kim Lewis, and Frederick M. Ausubel. Identification of novel antimicrobials using a live-animal infection model. *Proceedings of the National Academy of Sciences*, 103(27):10414–10419, 2006.

[MBSL99] Jitendra Malik, Serge Belongie, Jianbo Shi, and Thomas Leung. Textons, contours and regions: Cue integration in image segmentation. *Computer Vision, IEEE International Conference on*, 2:918, 1999.

[MCLF+09] Terence I Moy, Annie L Conery, Jonah Larkins-Ford, Gang Wu, Ralph Mazitschek, Gabriele Casadei, Kim Lewis, Anne E Carpenter, and Frederick M Ausubel. High-throughput screen for novel antimicrobials using a whole animal infection model. *ACS Chemical Biology*, 4(7):527–533, 2009.

[MTF+10] James P. Monaco, John E. Tomaszewski, Michael D. Feldman, Ian Hagemann, Mehdi Moradi, Parvin Mousavi, Alexander Boag, Chris Davidson, Purang Abolmaesumi, and Anant Madabhushi. High-throughput detection of prostate cancer in histological sections using probabilistic pairwise markov models. *Medical Image Analysis*, 14(4):617 – 629, 2010.

[OPG+97] S Ogg, S Paradis, S Gottlieb, G I Patterson, L Lee, H A Tissenbaum, and G B Ruvkun. The fork head transcription factor daf-16 transduces insulin-like metabolic and longevity signals in c. elegans. *Nature*, 389(6654):994–999, 1997.

[OSCR09] Eyleen J O?Rourke, Alexander A Soukas, Christopher E Carr, and Gary Ruvkun. C. elegans major fats are stored in vesicles distinct from lysosome-related organelles. *Cell Metabolism*, 10(5):430–435, 2009.

[PLL+08] Hanchuan Peng, Fuhui Long, Xiao Liu, Stuart K Kim, and Eugene W Myers. Straightening caenorhabditis elegans images. *Bioinformatics*, 24(2):234–242, 2008.

[RJB+08] Daniel Ramot, Brandon E Johnson, Tommie L Berry, Lucinda Carnell, and Miriam B Goodman. The parallel worm tracker: A platform for measuring average speed and drug-induced paralysis in nematodes. *PLoS ONE*, 3(5):7, 2008.

[RSTH01] R Ranganathan, E R Sawin, C Trent, and H R Horvitz. Mutations in the caenorhabditis elegans serotonin reuptake transporter mod-5 reveal serotonin-dependent and -independent activities of fluoxetine. *J Neurosci*, 21(16):5871–84, 2001.

[RvLM+00] Emiel Ruijter, Geert van Leenders, Gary Miller, Frans Debruyne, and Christina van de Kaa. Errors in histological grading by prostatic needle biopsy specimens: frequency and predisposing factors. *The Journal of Pathology*, 192(2):229–233, 2000.

[SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[Sch90] Robert E. Schapire. The strength of weak learnability. In *Machine Learning*, 1990.

[SFBL98] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[Sir00] R L Sirota. Mandatory second opinion surgical pathology at a large referral hospital. *Cancer*, 89(1):225–226, 2000.

[SKK+07] Lena K Schroeder, Susan Kremer, Maxwell J Kramer, Erin Currie, Elizabeth Kwan, Jennifer L Watts, Andrea L Lawrenson, and Greg J Hermann. Function of the caenorhabditis elegans abc transporter pgp-2 in the biogenesis of a lysosome-related fat storage organelle. *Molecular Biology of the Cell*, 18(3):995–1008, 2007.

[SM10] Rachel Sparks and Anant Madabhushi. Novel morphometric based classification via diffeomorphic based shape representation using manifold learning. In Tianzi Jiang, Nassir Navab, Josien Pluim, and Max Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, volume 6363 of *Lecture Notes in Computer Science*, pages 658–665. Springer Berlin / Heidelberg, 2010.

[SsS08] Shai Shalev-shwartz and Yoram Singer. Y.: On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. In *In: Proceedings of the 21st annual conference on Computational learning theory*, pages 311–321, 2008.

[TBMD08] Gabriel Tsechpenakis, Laura Bianchi, Dimitris Metaxas, and Monica Driscoll. A novel computational approach for simultaneous tracking and feature extraction of c. elegans populations in fluid environments. *IEEE Transactions on Biomedical Engineering*, 55(5):1539–1549, 2008.

[TRX+09] Jinshan Tang, Rangaraj M. Rangayyan, Jun Jun Xu, Issam El-Naqa, and Yongyi Yang. Computer-aided detection and diagnosis of breast cancer with mammography: Recent advances. *IEEE Transactions on Information Technology in Biomedicine*, 13:236–251, 2009.

[TTP+07] A. Tabesh, M. Teverovskiy, Ho-Yuen Pang, V. P. Kumar, D. Verbel, A. Kotsianti, and O. Saidi. Multifeature prostate cancer diagnosis and gleason grading of histoglogical images. *IEEE Transactions on Medical Imaging*, 26(10):1366–1378, 2007.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 1:511–518, 2001.

[WHR93] B Wightman, I Ha, and G B Ruvkun. Posttranscriptional regulation of the heterochronic gene lin-14 by lin-4 mediates temporal pattern formation in c. elegans. *Cell*, 75(5):855–862, 1993.

[wor11] Data bases on the genetics of c. elegans and related nematodes. www.wormbase.org, 2011.

[XSJ+10] Jun Xu, Rachel Sparks, Andrew Janowcyzk, John Tomaszewski, Michael Feldman, and Anant Madabhushi. High-throughput prostate cancer gland detection, segmentation, and classification from digitized needle core biopsies. In Anant Madabhushi, Jason Dowling, Pingkun Yan, Aaron Fenster, Purang Abolmaesumi, and Nobuhiko Hata, editors, *Prostate Cancer Imaging. Computer-Aided Diagnosis, Prognosis, and Intervention*, volume 6367 of *Lecture Notes in Computer Science*, pages 77–88. Springer Berlin / Heidelberg, 2010.