Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

Michael Nalisnik	Date

Scalable Computational Pathology: From Interactive to Deep Learning

By

Michael Nalisnik
Doctor of Philosophy

Computer Science and Informatics

Lee A.D. Cooper, Ph.D
Advisor

David A. Gutman, M.D, Ph.D
Committee Member

Daniel J. Brat, M.D, Ph.D
Committee Member

Vaidy Sunderam, Ph.D
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D
Dean of the James T. Laney School of Graduate Studies

Date

Scalable Computational Pathology: From Interactive to Deep Learning

By

Michael Nalisnik
B.S., Kean University, 1992
M.S., Montclair State University, 1997

Advisor: Lee A.D. Cooper, Ph.D

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2017

Abstract

Scalable Computational Pathology: From Interactive to Deep Learning

By Michael Nalisnik

Advances in microscopy imaging and genomics have created an explosion of patient data in the pathology domain. Whole-slide images of histologic sections contain rich information describing the diverse cellular elements of tissue microenvironments. These images capture, in high resolution, the visual cues that have been the basis of pathologic diagnosis for over a century. Each whole-slide image contains billions of pixels and up to a million or more microanatomic objects whose appearances hold important prognostic information. Combining this information with genomic and clinical data provides insight into disease biology and patient outcomes. Yet, due to the size and complexity of the data, the software tools needed to allow scientists and clinicians to extract insight from these resources are non-existent or limited. Additionally, current methods utilizing humans is highly subjective and not repeatable. This work aims to address these shortcomings with a set of open-source computational pathology tools.

We first present a comprehensive interactive machine learning framework for assembling training sets for the classification of histologic objects. The system provides a complete infrastructure capable of managing the terabytes worth of images, object features, annotations and metadata in real-time. Active learning algorithms are employed to allow the user to work in tandem with the system in an intuitive and efficient manner. We demonstrate how the system can be used to phenotype microvascular structures in gliomas to predict survival, and to explore the molecular pathways associated with these phenotypes. Quantitative metrics are developed to describe these structures.

We also present a scalable, high-throughput, deep convolutional learning framework for the classification of histologic objects is presented. Due to its use of representation learning, the framework does not require the images to be segmented, instead learning optimal task-specific features in an unbiased manner. Addressing scalability, the graph-based, parallel architecture of the framework allows for the processing of large image archives consisting of hundreds of slides and hundreds of millions of histologic objects. We demonstrate the system's capabilities classifying cell nuclei in lower grade gliomas.

Scalable Computational Pathology: From Interactive to Deep Learning

By

Michael Nalisnik
B.S., Kean University, 1992
M.S., Montclair State University, 1997

Advisor: Lee A.D. Cooper, Ph.D

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2017

Acknowledgement

Though the journey that is a Ph.D is an individual one, it is not possible without the support and assistance from many. I may acknowledge just a few here but will not forget the support, guidance and camaraderie from those not mentioned.

First and foremost, my deepest gratitude to my wife Colleen who persevered and sacrificed as much, if not more, than I throughout this journey. My sincere thanks to my advisor Dr. Lee Cooper for his guidance and support especially early on in my Ph.D career while trying to find a direction for my research. Many thanks to my committee: Dr. David Gutman for all the "toys" and his trust in me to administer them, Dr. Daniel Brat, for helping make my Ph.D more than just computer science and Dr Viady Sunderam for his wisdom in high-performance computing.

Finally, I would like to thank my fellow students that I have had the pleasure to share this journey with, some for just a few semesters and others for the entire ride, for making the experience unforgettable.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Overview

Pathology is a subspecialty of medicine that practices the diagnosis of disease. It is an area of medicine that is rich with data — from comprehensive molecular characterizations of tissues obtained by genomic and sequencing analysis, to high-resolution images of tissues obtained through various forms of microscopy imaging or histology. Pathologists use these data resources in various scenarios to render diagnosis, and often to assess prognosis and stratify patients into risk groups accosted with expected outcomes. Microscopic evaluation of tissues is a time-honored practice in pathology, dating back more than a century [1]. The visual properties of tissues carry important information on disease-related processes - like the formation of blood vessels, or immune system response. In cancers, the shapes and types of cells present carry important diagnostic and prognostic information that are used to classify the tumor and assess how advanced a patient's disease is. Manual evaluation of tissue histology by trained professionals is highly subjective, being prone to both considerable intra-observer and inter-observer variations [2]. For example, in an experiment by Fuchs and Buhmann [3], 180 randomly selected nuclei were presented in three different views of varying magnification. Five pathologists were tasked with determining if a nucleus is normal or atypical by wether or not it is stained by an Immunohistochemical (IHC) stain. To test for intra-pathologist variation, a subset of the nuclei were presented again, this time rotated by 90 degrees. The overall intra-pathologist classification was 21.2%, meaning that every fifth nucleus was classified contrary to the original classification. For inter-pathologist classification, out of 180 nuclei all five pathologists agreed on 105. For the remaining 75 nuclei the pathologists disagreed,

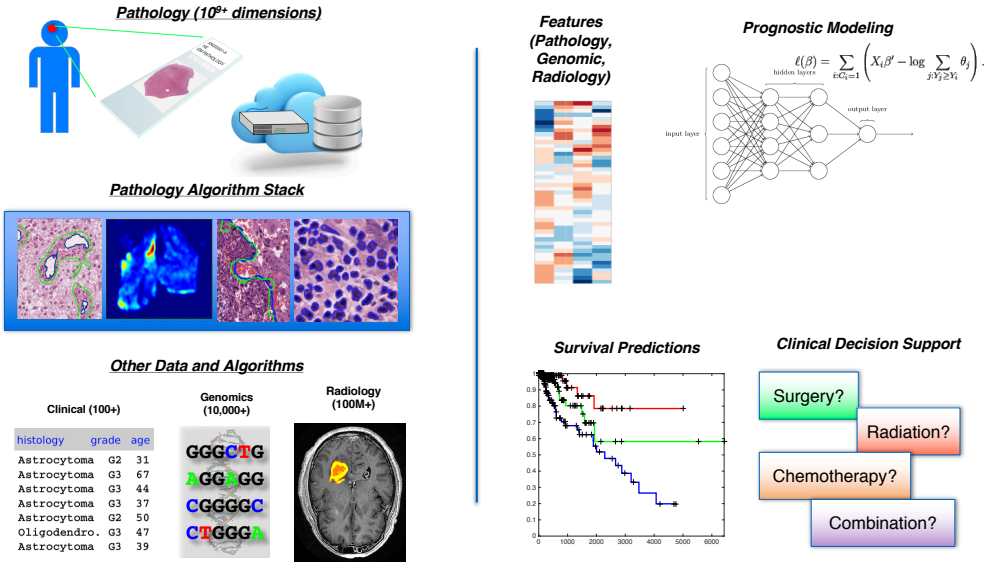

Figure 1.1: Computational pathology

Technologies, frameworks and algorithms that identify histologic objects and structures in an object, repeatable and scalable fashion.

resulting in an error rate of 42%. The development of more objective quantitative metrics for evaluating pathology images remains a significant barrier in effectively using this resource in research and clinical care.

Recent advancements in microscopy imaging now enabled the digitization of massive volumes of histology data. Slide scanning microscopes can produce whole-slide images that capture the entire histological detail of a tissue specimen in a single high magnification image pyramid. These images can be digitized from histologic sections at 200x–400x magnification, generating substantial images containing billions of pixels each, with dimensions in the tens-of-thousands to hundreds-of-thousands of pixels. A single whole-slide image can be digitized in around 2.5 minutes, and a single slide

scanning device can easily produce a terabyte of image data in a single day. These devices are increasingly being adopted in both clinical and research settings, resulting in an explosion of histology image data. The creation of vast collections of whole-slide images creates an opportunity to extract information from this content using image analysis algorithms. A commonly used algorithm is the segmenting of micro-anatomic objects in the images and representing each object with a set of quantitative features, methods such as this can produce objective and repeatable descriptions of the visual properties of patient tissues. When combined with genomic and clinical data, these representations can be used to improve the accuracy of diagnosis and prognosis, and to reveal new insights into disease biology. Public repositories such as The Cancer Genome Atlas (TCGA) provide archives of whole-slide images, genomic and clinical data for more than 10,000 human subjects, spanning more than 20 selected types of cancers. Each whole-slide image can contain hundreds-of-thousands to more than one million micro-anatomic objects — resulting in hundreds-of-millions of objets just for a single cancer type.

The visualization, management and analysis of whole-slide image data bring new challenges for human-computer interaction, large-scale data management and machine learning. For example, one of the most common applications in computational pathology is to apply learning algorithms to develop classification rules for phenotypes of interest, with the goal of quantifying the abundance of a specific type of cell for each patient in a cohort. Computational analysis of histology images have been used to predict metastatic potential [4], survival [5] [6] [7] [8] [9], grade [10] [11], histologic classification [12] [13] [14] and to link histologic patterns with genetic alterations or molecular disease subtypes [15] [16] [17]. Although these algorithms demonstrate scientific or potential clinical utility, few directly engage domain experts, address the challenge of processing hundreds of millions of micro-anatomic objects or are provided as comprehensive open-source tools to the research community.

1.2 Contributions

The goal of this work is to provide scalable computational pathology tools which provide accurate, repeatable and objective results and provide these tools to the research community as an open-source resource. To that end, we present the following: 1. A scalable interactive learning framework and 2. A scalable, high-throughput, deep-learning framework.

The first tool is a software framework for interactive classification and phenotyping of large whole-slide imaging datasets that: 1. Has a web-based interface that can fluidly display gigapixel image and annotations of tens of thousands of annotations, 2, Utilizes a machine learning server for fast, interactive training of classification rules in large datasets, 3. Employs active learning algorithms to improve training efficiency, and 4. Provides tools for creating, sharing and reviewing labeled data and ground-truth validations sets.

The second tool is a deep-learning framework for the characterization of large whole-slide image datasets. The use of deep learning eliminates the need for nuclear segmentation and the development of "engineered" features. The graph-based architecture of the framework allows for high-throughput analysis of hundreds of whole-slide images containing over a million histologic objects each.

In particular this work contributes the following:

- An interactive learning system that connects experts with powerful learning algorithms and very large datasets generated from whole-slide images [18] [19].
- A large-scale, real-time visualization of tens of thousands of object boundaries enabled by a dynamic caching scheme.
- The use of heatmaps to augment active learning algorithms by guiding the user to objects of interest rather than explicitly selecting objects.

- A set of metrics: Hypertrophy Index and Clustering Index, to quantify microvascular phenotypes. These metrics are calculated using machine-learning methods and utilize classifiers trained in the interactive learning framework.

- Demonstrates that quantified microvascular phenotypes are equivalent to grade when combined with genomic variables [20].

- Demonstrates that active learning outperforms passive learning in classifying vascular endothelial cell nuclei.

- An efficient deep learning pipeline capable of classifying hundreds of millions of objects.

1.3 Organization

This dissertation os organized into two parts: Chapters 2 through 4 focus on a scalable interactive framework. Chapters 5 and 6 focus on a scalable, high-throughput deep-learning framework.

Chapter 2 provides an introduction to active learning which is an important algorithm employed in our interactive learning system. Chapter 3 describes the architecture and functionality of HistomicsML. Chapter 4 demonstrates the phenotyping of microvascular structures in gliomas and describes the metrics developed to describe these phenotypes. Chapter 5 provides an introduction to deep learning and recent advances in its methods. Chapter 6 describes the architecture and functionality of TissueNet, a scalable high-throughput deep learning framework for the characterization of large whole-slide image datasets. Chapter 7 gives conclusion and a description of future work.

Chapter 2 Active Learning

2.1 Introduction

Supervised learning is the process of building a prediction model with training samples that have known outcomes. The training data is made up of features which describe the characteristics of the sample. The outcomes are typically called labels. For instance, a classification model to determine the type of iris would be trained using samples that describe the iris with features such as; sepal length, sepal width, petal length and petal width. The training set would consist many samples composed of these features along with a label, in this case the type of iris. Once a model is trained, it can be used to classify an iris's type using only the features describing the iris are known. The training dataset is often called the ground truth.

While having ground truth for supervised learning is necessary, obtaining it is not always a trivial task. The cost associated with obtaining the training samples can be significant. This cost can be in the monetary sense of the word, though most often it is measured in time and effort. Suppose we need to build a model capable of classifying images as either a cat or a dog. Obtaining the labels for the training set is inexpensive as almost anyone can instantly identify a dog or a cat. Now let's say we need to build a model capable of classifying images of skin lesions as cancerous or benign. A domain expert, in this case a highly trained doctor, must be used to identify the lesion. Additionally, the type of lesion may not be apparent to the domain expert, requiring more time and even additional domain experts to identify it. In this case the cost of obtaining a label is expensive in time, effort and money.

Another difficulty in obtaining ground truth is the tendency to select "textbook" examples for a training set. These examples are typically unambiguous and take little

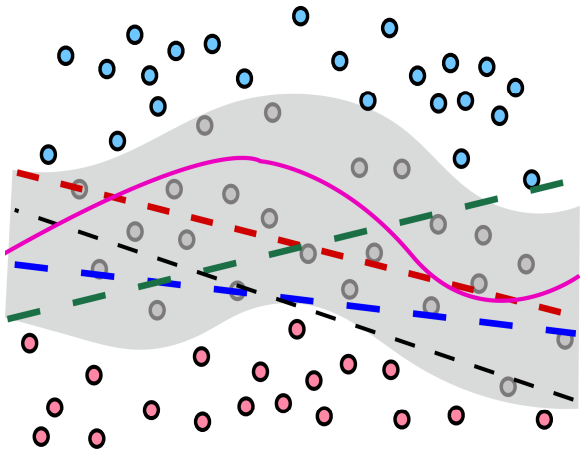

Figure 2.1: Ambiguity with textbook examples

When selecting only textbook examples, the area around the actual class boundary (pink curve) is not represented in the training set allowing any of the other (inaccurate) boundaries (dashed lines) to be learned by the algorithm.

effort for the domain expert to identify. Selecting these types of examples causes a reduction in the accuracy of the model. This is known as selection bias [21]. The model will not generalize well to examples that are not as obvious as those selected for the training set.

With a small amount of data to label this may not be a problem since the domain expert is forced to label all of the data. As the size of the dataset grows, the tendency to select textbook examples increases as it becomes much more difficult, or even impossible, to examine the entire dataset. For datasets such as cellular objects from whole-slide images, selecting ground truth can be cognitively exhausting.

There is a need to improve ground truth selection by minimizing cost while maximizing accuracy. The following will describe one such method, active learning.

2.2 Active Learning

Active learning is a method in which the learning algorithm interacts with an oracle to select training samples to learn from instead of passively learning from a preselected set [22]. The oracle can be anything capable of returning a label to the

learning algorithm; a machine, a person, another algorithm, etc. The idea is that the learning algorithm knows its own weaknesses and therefore can select a sample that will better improve its generalization - how well it performs on unseen data. The strategic selection of training samples by an objective algorithm has been shown to overcome selection bias and improve generalization in several machine-learning problems [23] [24] [25].

There are several scenarios in which active learning may operate. Three of the main scenarios are: query synthesis, stream-based selective sampling and pool-based sampling [22]. In query synthesis, the algorithm may select any sample from the input space, including samples that it synthesizes [26]. This requires the distribution of the data to be known and finite. For stream-based selective sampling, the act of obtaining a sample, is of minimal cost and can occur frequently. The learning algorithm has to decide whether to discard the sample or query the oracle for its label. The labeled samples are used to guide the algorithm to regions of the domain that it is weakest by selecting the appropriate samples to query for a label [27]. The third scenario is pool-based sampling. In this scenario there is a large collection of unlabeled data available. [28] The algorithm selects samples from the pool based on how confident it is of predicting the samples class. Unlike stream-based selective sampling, the samples from the unlabeled pool are not discarded and may be used in later queries.

So how does the learning algorithm know which samples to select for query? Simply stated, the algorithm chooses samples that it is most uncertain of. At any time during the learning activity, the algorithm has learned from a set of already label samples. At this point the algorithm tries to predict the class of value for the unlabeled samples. Depending on the classification or regression algorithm used, the learner knows how certain it is of its prediction. In the case of a support vector machine, the closer a sample is to the decision hyperplane, the more uncertain the

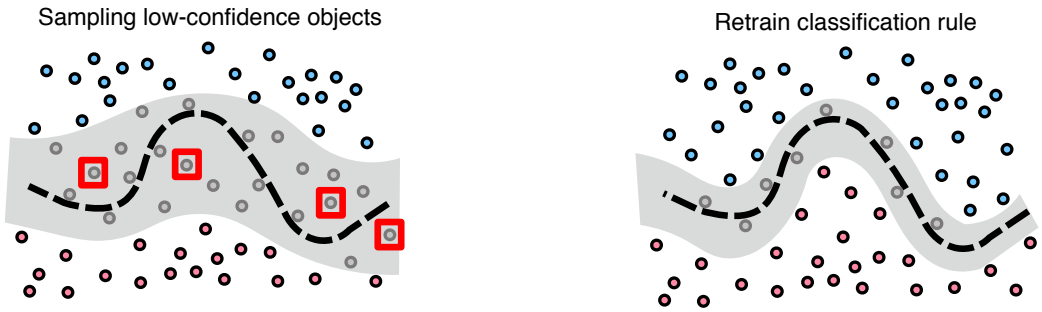

Figure 2.2: Sampling ambiguous examples

Selecting more ambiguous samples allows the algorithm to learn the area around the actual boundary more accurately. By selecting the samples indicated by the red boxes, the algorithm is able to learn a more accurate boundary. (indicated by the shaded areas)

algorithm is of its prediction. In linear regression, the farther away from the regression line a sample is, the more uncertain the algorithm is. In an ensemble method such as random forests, the smaller the difference between the number trees predicting the positive class and trees predicting the negative class, the more uncertain the algorithm is of its prediction. Selecting the samples the algorithm is most uncertain of provides more information about the data space allowing the algorithm to reduce weaknesses thus making it more accurate. This process is called uncertainty sampling [28].

Active learning is an iterative process. The algorithm starts with a few labeled samples to build an initial model. In the pool-based scenario, the model is then used to predict the class or value of every sample in the unlabeled pool. The sample or samples for which the algorithm is most uncertain of are then selected to be presented as queries to the oracle. The model is then updated with the newly labeled samples and the process starts over again. This cycle is repeated until a sufficient accuracy is achieved or stopping criteria is met.

Determining the appropriate stopping criteria is an open problem in the literature [29] [30] [31]. An obvious method is to have the system stop when a particular

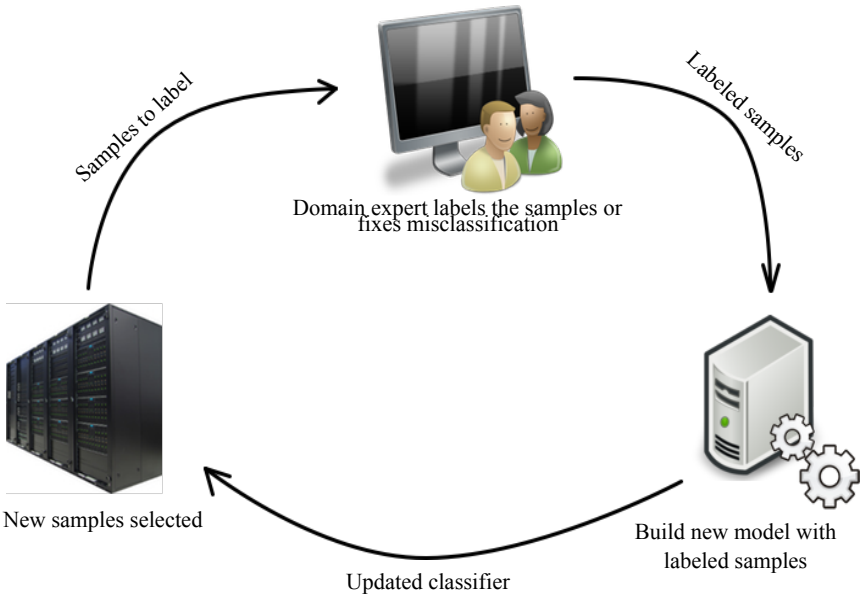

Figure 2.3: The active learning process

An objective sampling strategy scans the unlabeled sample pool to identify those examples that will improve the classifier the most. The user labels a small set of the samples, the classifier is updated and the cycle repeats.

accuracy is reached. There are a few problems with this strategy. Typically, an algorithm's accuracy is assessed using a labeled validation set, which would require more tabled data. Where would this validation set be obtained? Having the user select it raises the same issues with selection bias and having another domain expert select the set increases the cost, which runs contrary to the goal of active learning. Cross-validating using the previously selected samples tends to be over optimistic, especially when the set includes few sample. Besides these issues though, what is an appropriate accuracy? In general, most work on stopping criteria is based on some intrinsic measure of stability or self-confidence, stopping when the accuracy plateaus and active learning ceases to be useful. Such self-stopping methods may be applicable in certain situations. Though in reality, economic or other external factors will dictate the stopping point [22]. For another argument against self-stopping, certain types of data may become ambiguous as they approach the decision boundary between classes. This ambiguity makes it difficult, if not impossible, for a domain expert to distinguish

between classes. As the active learning process moves closer to the decision boundary during selection, the user will be less likely to provide a label for the samples. This may suggest using a metric to quantify this difficulty, but this too may be inaccurate. As the dimensionality of the data increases, the decision boundary may become more complex. This can cause particular areas of the data space to not have been explored, resulting in poor generalization error [32], In the end, it will most likely depend on the user to determine an appropriate stopping criteria.

Chapter 3 HistomicsML - A Scalable Interactive Learning Framework

3.1 Introduction

A common analysis task in histology datasets is the classification of cells. With whole-slide images containing hundreds of thousands to more than one million cells, a dataset consisting of a few hundred patients can easily contain hundreds of millions of cells. The amount of data in these datasets poses a few challenges in building a classifier. First, selection bias for a dataset this size will be a significant problem. We cannot expect domain experts to manually search through hundreds of whole-slide images and not be biased in their selections. Second, there needs to be a way to annotate and capture input from the domain expert in an efficient and intuitive manner. Visualizing, capturing input and managing the data cannot be overlooked with massive datasets such as these. Often in the literature the focus is on the accuracy or efficiency of an algorithm without regard to the infrastructure required to attain the reported results. Most published algorithms operate as command line tools directly on data files and lack a user interface that enables a domain expert to interact with the data and algorithms. When input from a domain expert is needed, it is typically acquired offline by presenting a small collection of manually selected image subregions to the domain expert for labeling or annotation. Enabling domain experts to directly interact with machine learning algorithms via active learning on large datasets has been shown to improve prediction accuracy and user experience in general applications [33] [34] [35] [36] [37]. The challenge in utilizing active learning with histomic data is in building software with the scalable visualization and machine-learning capabilities described above.

We have developed HistomicsML, an interactive software framework which facilitates the phenotyping and classification of large whole-slide datasets. HistomicsML enables users to rapidly train accurate histologic classification rules with several key features: (i) A web-based interface that can fluidly display gigapixel images containing millions of image analysis objects (ii) A machine-learning server for fast, real-time interactive training of classification rules in datasets with hundreds of millions of objects (iii) Utilization of active-learning algorithms for improved training efficiency (iv) Tools for creating, sharing and reviewing labeled data and ground-truth validation sets. HistomicsML is open-source (https://github.com/cooperlab/ActiveLearning) and is available as a software container for easy deployment (https://hub.docker.com/r/histomicsml/active/).

3.2 Architecture

HistomicsML is web-based system designed for scalability in the direction of both the number of users and the size of datasets. The system is composed of four major components: (i) The learning server — updates the classifier, selects new samples for the user to label and generates heat maps (Ii) The image server — Provides whole slide image pyramids that allow the user to zoom and pan through the whole image in real-time (iii) Database — Contains the locations and boundaries of all the segmented cells (iv) The web application — The system's user interface.

3.2.1 Random Forests

While active learning is agnostic to the specific classifier used, so long as it provides a measure of prediction confidence, HistomicsML utilizes Random Forests so a short overview is warranted.

Random Forests [38] are an ensemble learning method utilizing bagging of classification trees and random feature selection. Bagging, short for bootstrap aggregating, aggregates the results from m trees which are constructed independently using bootstrap sampling of the training set. That is, for a random forest of size m and a training set T of size N, m new training sets of size N are generated by uniformly sampling T with replacement. Each of the m trees in the random forest are then constructed using one of the generated training sets.

In the usual construction of classification and regression trees (CART), the best split point is chosen from all features of the sample. That is, the feature that alone best classifies the training sample [39]. In random feature selection, a small group of features $m \gg M$ are randomly selected at each node with the best split point selected from this group. The size of the group m remains constant for each node.

We choose random forests for HistomicsML due to their resistance to overfitting, computational efficiency and simplicity in design. For our implementation we used

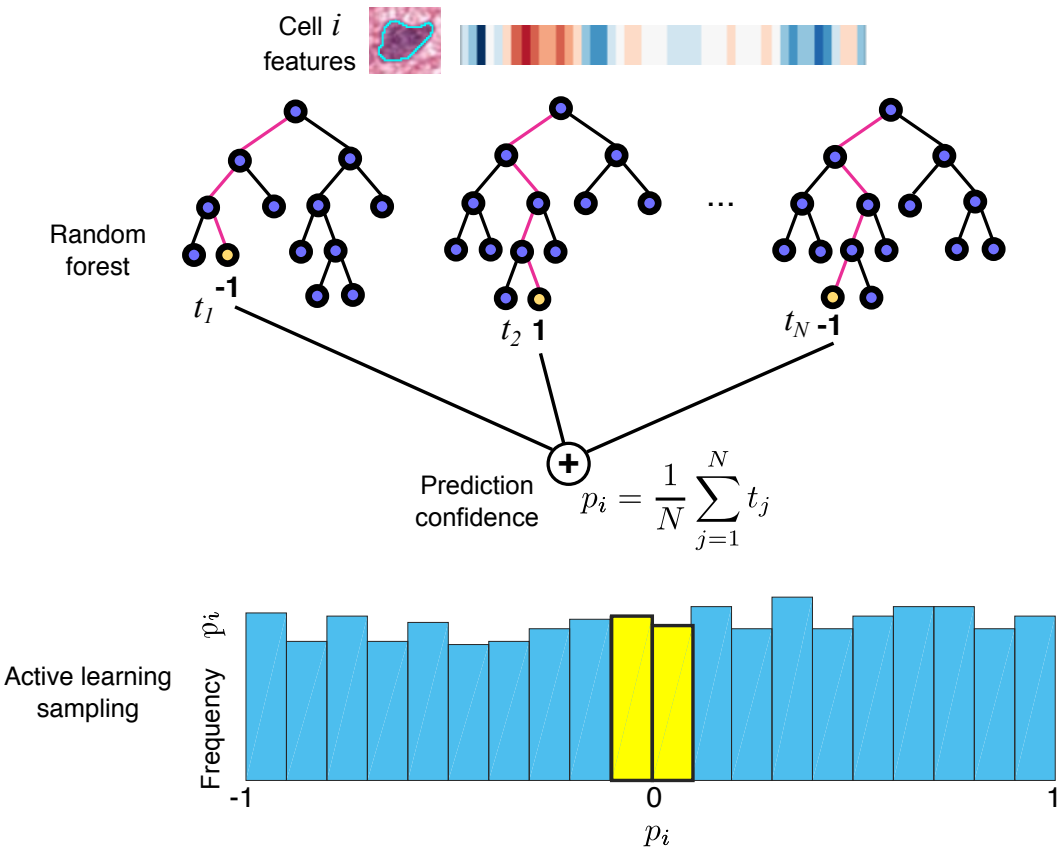

Figure 3.1: Active learning with random forests

The random forest classifier aggregates the predictions of multiple decision trees and provides a readout of prediction confidence. Given the histomic feature profile of an entity, each tree in the forest predicts the class ti as either the positive (+1) or negative (-1) with the final aggregate prediction made by majority vote. Prediction confidence is measured as the absolute value of the prediction average (pi). Objects with a confidence $|p_i|$ close to one indicate a consensus of the decision trees, where objects with a confidence $|p_i|$ close to zero indicate a lack of consensus by the trees. Objects with lower confidence scores are difficult to classify and make good candidates for labeling in the active learning paradigm. In our framework we calculate the object labels and confidence scores for instance-based sampling and heatmap generation with each classifier update/iteration. Objects with minimum confidence (where trees are tied or most discordant) are sampled for instance based learning.

100 trees, with the maximum depth of the tree fixed at 10 and (eq floor of the square root of features) features selected for node splits. In the active learning context, the prediction confidence, or uncertainty, is calculated by voting on the prediction of the individual trees. The prediction confidence for a random forest classifier evaluated on object i is calculated as

$$c_i = \left| \sum_{j=1}^{N} t_j \right| \in \{-1, 1\}$$

Where t_j is the classification tree j of N total trees. Maximum prediction confidence is achieved if all trees agree on the predicted class. Minimum prediction confidence is achieved when the individual trees are evenly split on the predicted class. In the literature, maximum prediction confidence is often referred to most certain while minimum prediction confidence is referred to as most uncertain.

3.2.2 Data Formats

HistomicsML utilizes three input data formats: 1. A set of whole-slide images in pyramid tagged image file format (TIFF). 2. A collection of object boundaries in a text delimited format. 3. A set of features characterizing the image analysis objects as binary data stored in HDF5 format. The input data is formatted in a preprocessing step for consumption into the system. The whole-slide images are converted from proprietary microscope vendor formats using VIPS [40] and OpenSlide [41]. Object boundaries and histomic features are generated by an image segmentation pipeline that is described in more detail later. The text-delimited output from the pipeline is then converted — boundary data is put into comma separated values for insertion into a MySql database, while the histomic features are z-score normalized and stored in HDF5 format. Storing the features in HDF5 is done to facilitate efficient loading into the systems and maintain an internal organization of objects by patient and slide along with other metadata. The object metadata includes: the object's database ID

for its boundary data, the centroid of the object, the slide name the object is located on, normalization data, indices for fast access of feature data by slide and provenance data. For output, the training set is also stored in HDF5 format. The output contains the histomic features of the objects, the class label of objects labeled during training, the iteration the object was added to the training set, class names, slide names, object database ID, normalization data and provenance data. Information about the training set is also stored in the database. This include information such as dataset used, name of the training set, filename of the training set, total number of objects selected, the ID's of the objects and the creation time of the training set.

3.2.3 Image Segmentation and Feature Extraction

As mentioned earlier, the histomic features and object boundaries are generated by a preprocessing step. The software pipeline that performs this is shown in Figure 3.2. This pipeline utilizes algorithms provided by the HistomicsTK Python library for histologic image analysis (http://github.com/DigitalSlideArchive/HistomicsTK) to perform color normalization, nuclear masking and splitting, feature extraction, and database ingestion. Each whole slide image is normalized to a standard hematoxylin and eosin (H&E) sample with desirable color characteristics using Reinhard normalization. Tissue pixels are first masked from the background using linear discriminant analysis and then the mean and standard deviation of the tissue pixels in the LAB color space are calculated. These moments are mapped to match the moments of a color standard image prior to inversion back to RGB color space. This color normalization process considerably improves the quality of subsequent image analysis steps, improving the consistency of segmentation results and image features. Whole slide images are split into 4096 x 4096 pixel tiles and processed independently to highlight cell nuclei. A color deconvolution algorithm is first applied to digitally separate the hematoxylin and eosin stains. The deconvolved hematoxylin intensity images are then

masked to identify pixels corresponding to cell nuclei using a combination of adaptive thresholding and morphological reconstruction to remove background debris. Closely packed nuclei are then split using a watershed segmentation applied to the laplacian-of-gaussian response of the hematoxylin image. Each nucleus is characterized using a set of 48 features describing its shape, intensity and texture. These features include eccentricity, solidity and fourier shape descriptors (shape), statistics of hematoxylin signal including variance, median, mean, min/max, kurtosis, skew and entropy (intensity) and statistics of hematoxylin intensity gradients (texture). Color normalization, segmentation and feature extraction is carried out in a cluster-computing environment to allow timely processing of large sets of whole-slide images. Boundaries and features generated by segmentation are stored in a text-delimited format and used as input to the preprocessing described in the previous section.

3.2.4 Scalability

The interactive nature of the system implies an expected responsiveness — while a hard real-time response is not required, delays measured in more than several seconds will diminish the user experience. As such, our idea of scalability is not only related to the size of datasets but also in responsiveness of the system. As we strove to avoid delays in the system as a whole, there were two areas of particular concern: Visualization of the whole-slide images and annotations and a responsive sampling strategy capable of handling hundreds-of-millions of data objects. The system was designed in a modular fashion with the ability to utilize multiple servers, thus allowing for scalability to larger datasets and more concurrent users. In out implementation we opted for a single multi-core server to host all components.

A single whole-slide image can have tens of millions of polyline annotations that delineate the boundaries of histologic objects. These annotation must be able to be displayed in a fluid real-time maker to the user. To achieve this, scalable vector graph-

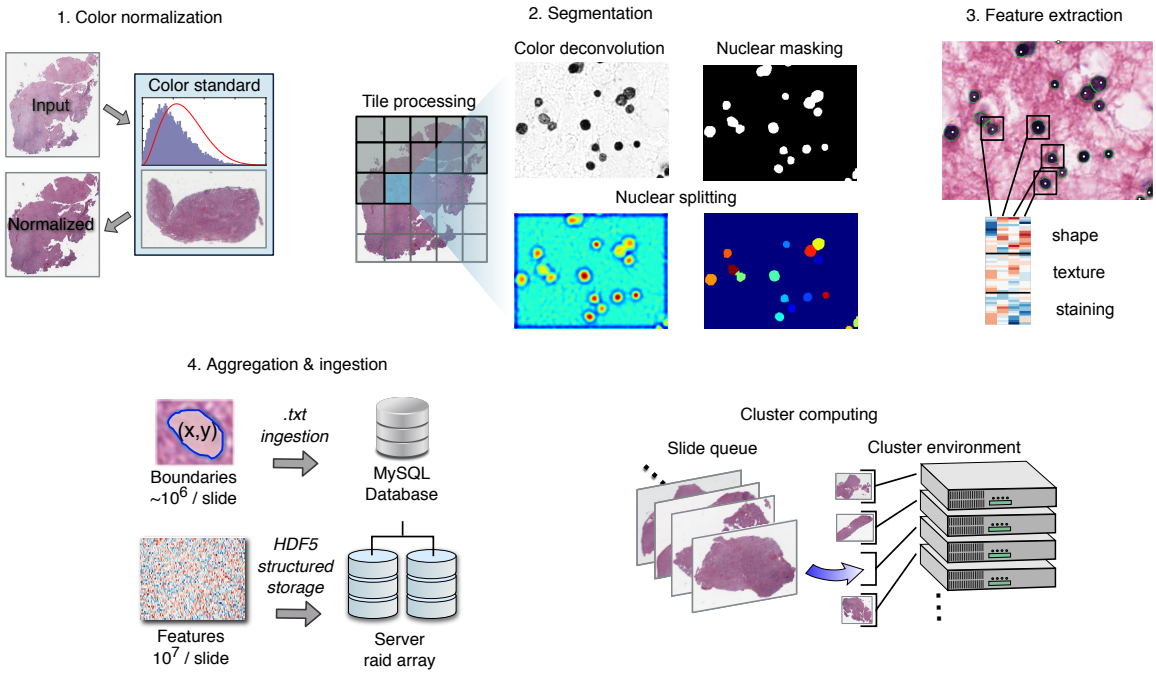

Figure 3.2: Segmentation pipeline

The system uses an image analysis pipeline for analyzing cell nuclei in whole-slide images based on HistomicsTK (http://histomicstk.readthedocs.io), a software library for digital pathology image analysis. Step 1 in this pipeline normalizes the color characteristics of each slide to a gold standard H&E to improve color-deconvolution and downstream filtering operations. Step 2 processes the slide tile-wise, first digitally unmixing color images into eosin and hematoxylin stain images, then analyzing the hematoxylin image to mask nuclear pixels using a Poisson-Gaussian mixture model and smoothing this binary mask with a graph-cutting procedure. We then apply a constrained Laplacian of Gaussian filter to split closely packed cell nuclei. In step 3, a set of 48 features describing shape, texture and staining is calculated for each segmented cell nucleus. Finally, in step 4 all segmentation boundaries and features from each slide are aggregated into a single file. A delimited-text format is used for object boundaries, which are ingested into a SQL database to drive visualization in the user interface. Features are stored in a HDF5 structured format on a RAID array for fast and convenient access in training and evaluating classification rules.

ics (SVG) overlays were generated in real-time from boundary information stored in the database. To obtain high query throughput, the boundaries were indexed by slide name, x-centroid and y-centroid. The order of the index keys is important as using the slide name first reduces the number of records to process from hundreds of millions to a million or less. To reduce the number of queries needed as a user pans around an image, a spatial caching scheme was implemented. When generating the SVG overlay, the system queries the database not only for the annotations that are visible in the current view but also those in the area surrounding it. With the SVG generated being larger than the visible portion of the slide, the web browser efficiently crops the SVG displaying only those annotations that fall within the visible portion of the slide. As the user pans, the SVG is translated and cropped again. The translation and cropping are significantly faster than querying the database and regenerating the SVG. New queries are performed in the background while the slide image in being panned. The background query occurs when the slide is panned by half the visible area in any direction and the new SVG is generated while the current one is still being displayed. Once generated the new SVG is made visible while the old one is hidden and discarded, providing a seamless update.

The ability to apply a classifier and analyze its response in seconds is paramount to the usefulness of the system.

3.2.5 Learning process

The system employs two methods for soliciting class labels from the user: Instance-based and Heatmap-based. The instance-based method is a typical implementation of active learning where the system selects a few samples (in out case eight) with the lowest classification confidence and presents them to the user for labeling. The user then labels the samples and submits them to the system and a new classifier is built with the updated training set. The heatmap-based method displays a transparent

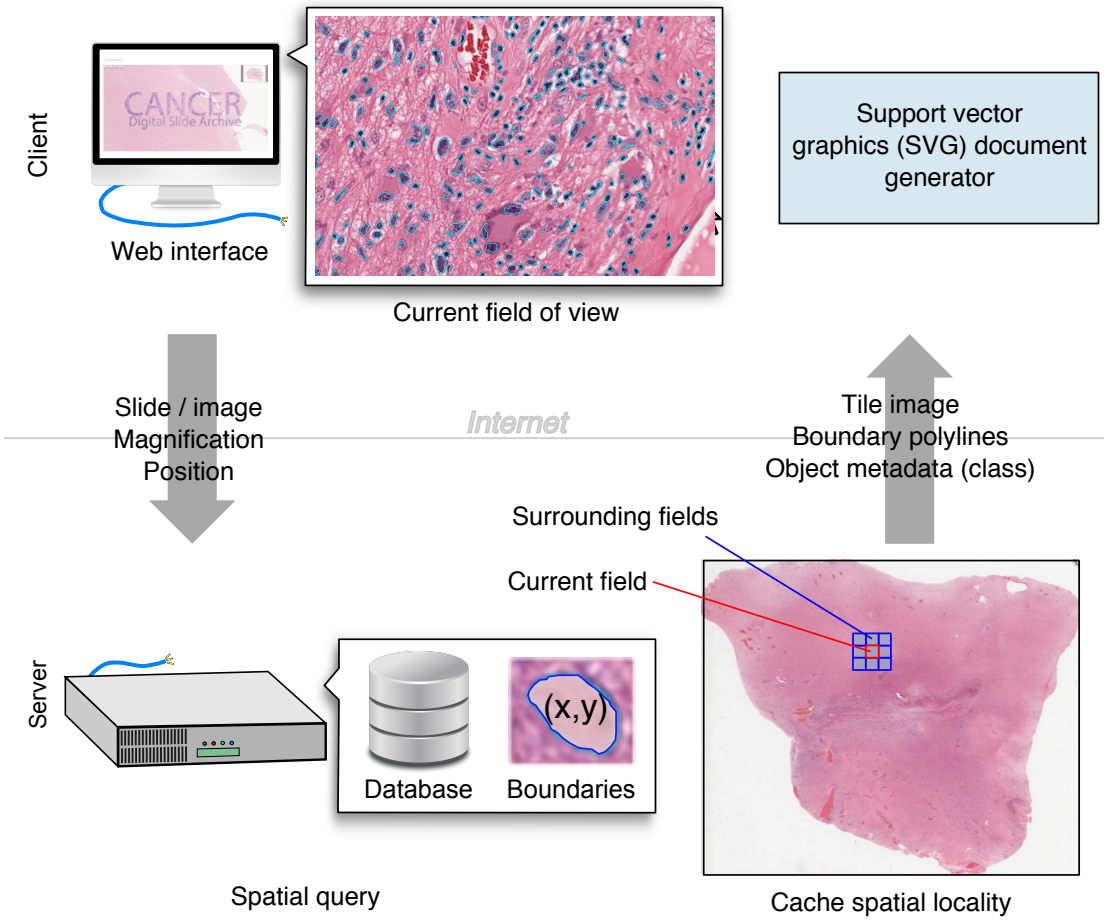

Figure 3.3: Scalable display of boundaries

Each whole-slide image can contain a million or more histologic entities, each with polygonal boundaries that consist of multiple (x,y) vertices. Rendering these boundaries fluidly requires effective database query, client-server communication and spatial caching. Our software framework renders boundaries in the web interface using a dynamic strategy outlined here. Following a mouse event, the current field of view (position/magnification) is communicated to the server. If the magnification is at or above 10X, the database is queried to identify objects in the current and adjacent fields. The image data, object boundaries and object metadata (including class) are communicated back to the web client. The web-client then constructs a Scalable Vector Graphics (SVG) document that contains the boundary polylines and that encodes any classification information using color tags. This strategy provides fluid visualization and does not incur any delay on a panning event in the client viewer, since the adjacent regions are already encoded in the SVG document.

heatmap overlay, representing smoothed prediction confidence or positive class density, on top of the whole-slide image. The heatmap guides users to areas of interest in the slide where the user may or may not select objects. With the heatmap-based method, the user is the one to select the objects to be added to the training set rather than the system, with the system just making "suggestions". This is closer to the scenario of stream-based selective sampling [27] mentioned earlier, rather than the pool-based approach used in the instance-based method.

The instance-based method is a straight forward implementation of active learning with a slight alteration. At any iteration of the process, the current labeled training set is used to build a random forest classifier. The classifier is applied to the remaining unlabeled data pool. Due to the size of the unlabeled data pool and the discrete nature of the uncertainty score, there will be many samples having the highest uncertainty. Therefore we randomly sample eight from the most uncertain. This deviated from the usual active learning method were only one sample is selected per iteration.

The heatmap-based method is used to guide the user to areas on slides that have low prediction confidence or high positive class density. Other than guiding the user to areas of interest, the system has no input on which objects to add to the training set. While uncertainty sampling may eventually select from these areas, the massive size of the datasets intended for use in the system may need far too many iterations. The heatmap-based method also helps in areas that may be misclassified but not of the lowest prediction confidence. This occurs when the active learning process starts selecting very close to the decision boundary where it may be impossible for the user to determine the class of the object. The positive class density is of use when classifying objects that tend to group together naturally. For example, in tissue, endothelial cells tend to group together near blood vessels. By guiding the user to areas where the positive class is prevalent, areas of the data space missed by the active learning process can be explored. The system also employs a slide ranking

method which presents the user with slides ordered by minimum average prediction confidence. This guides the user at the slide level to those which have more areas of low prediction confidence, eliminating the need to look at every slide in the dataset when using the heatmap-based method.

3.2.6 Workflow

The user begins at the landing page where they can start a new session or resume a previous learning session. To start a new learning session the user selects a dataset from a drop down menu and provides names for the session, positive and negative classes in the respective text fields. To initialize the classifier, the user is then directed to a "priming" screen to select 4 examples from each class. The priming screen contains a whole slide image viewer that displays the selected slide and boundary annotations. Users can select examples by double-click, which highlights their boundary in yellow, and adds a thumbnail image of the selected examples to an array above the viewer. Following this labeling the initial classifier is trained and applied to the entire dataset to generate initial class predictions and confidence values. The user then enters the "select" screen where they will provide additional labels through active learning feedback. To resume a session, the user selects a dataset from the dropdown menu, which populates another dropdown menu containing the sessions available for the selected dataset. Selecting a session and clicking the "continue" button brings the user to the "select" screen to resume the session from the last iteration.

The selection page is the interface for the instance-based method of soliciting feedback. Here the user is presented with eight "ambiguous" samples selected based on their prediction confidence. These samples are displayed as thumbnail images in an array across the top of the screen along with the predicted label. When the user clicks on a sample in the array, the slide viewer in the bottom portion of the screen will load the appropriate slide image, pan and zoom the location of the sample centering

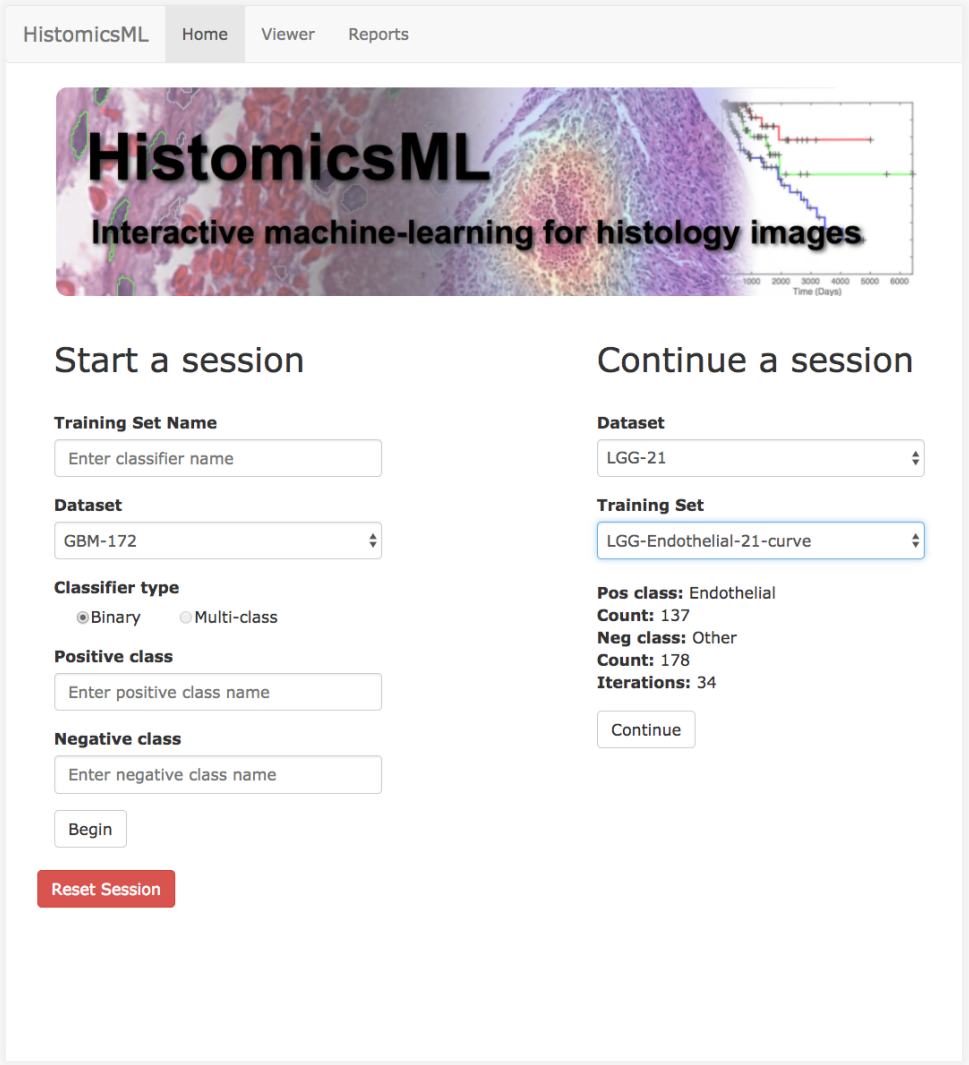

Figure 3.4: Landing page

The landing page enables users to initiate a new learning session or to continue an existing session. For starting a new session users select a dataset, provide a session name and assign class names for training. Selecting a dataset from the continue session option populates a drop down list displaying the session names, class names and labeled example statistics for sessions associated with that dataset

it in the screen. The boundary of the sample is displayed along with a 50 x 50 pixel bounding box. The user can toggle the boundary on and off for the sample. The bounding box allows for easy location of the sample while panning through the slide or when the sample's boundary is not displayed. The user changes the label by double clicking on the thumbnail image of the sample in the array. This will cycle through the possible labels. As the system is currently limited to binary classifiers, there are three possible choices for the label. Naturally there is the positive and negative class labels. There is also an "ignore" label which allows a selected sample to be excluded from the training set. If the user is unable to determine the samples class, or if the sample is not valid. A sample may not be valid due to an error in segmentation or is an artifact rather than a cell. Labeling a sample as "ignore" also removes it from the unlabeled pool. Once the samples have been given the proper label, they are submitted to the system and a new classifier is built with the updated training set. The system applies the updated classifier to the unlabeled pool of samples and selects eight to be displayed.

At any point after priming the system, the user can utilize the "viewer" and "heatmaps" screens. In the viewer screen, the user can pan and zoom through any slide in the dataset, display the boundaries of the segmented objects, and display classification results of the current classifier. Normally when displaying the boundary of an object, the viewer screen uses aqua to color the boundary. When displaying classification results, the boundaries are colored according the classification result for the object. The specific color used is indicated in a legend. The user can display or hide the boundaries, making it easier to examine individual objects. When enabled, the boundaries are only displayed at higher magnification, 10X and beyond. When viewing at lower resolutions an uncertainty heatmap or positive class density heatmap will be overlaid on top of the slide image. While zooming in from a lower magnification with a heatmap displayed, the system will switch to the boundary display

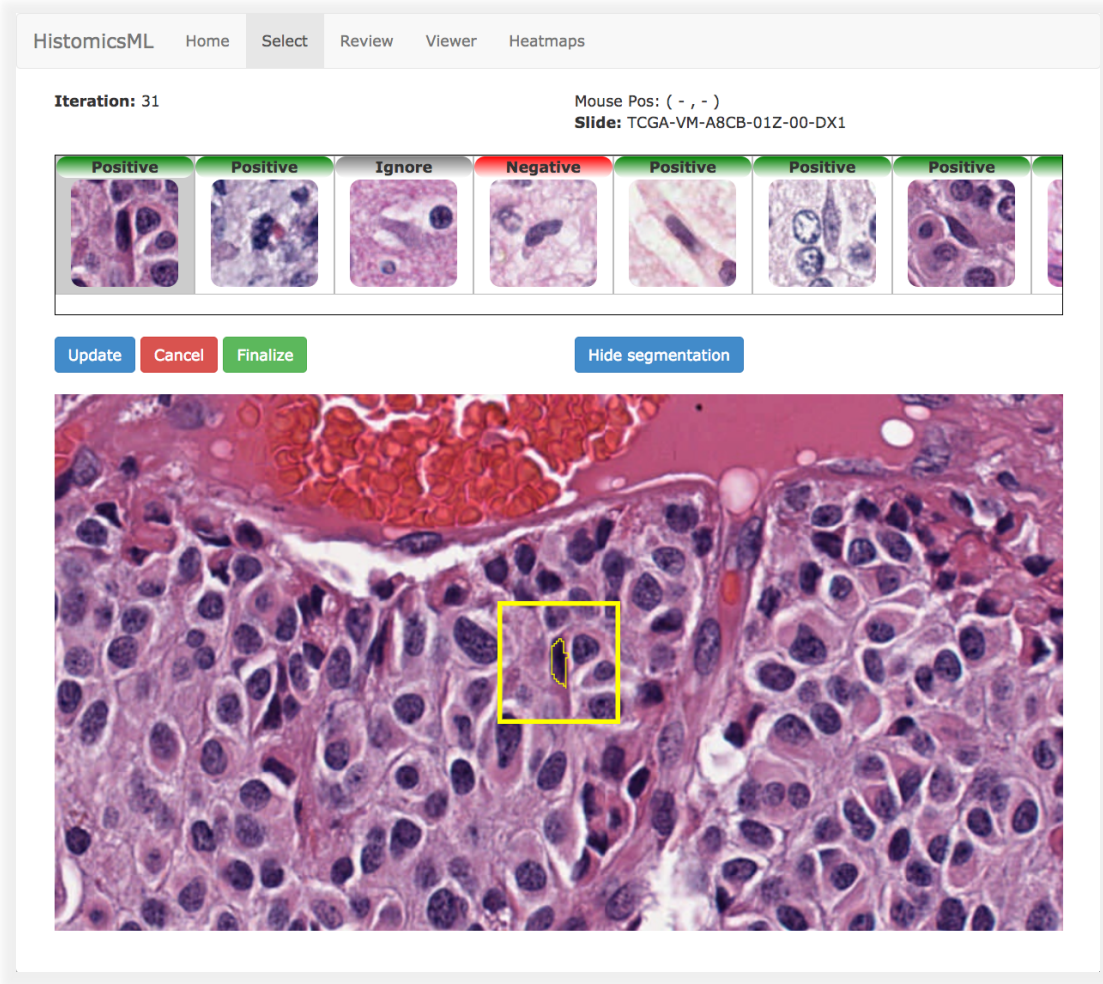

Figure 3.5: Selection page

This view facilitates the labeling of samples selected by active learning to refine the classification rule. Thumbnail images of 8 instances selected as valuable by active learning are displayed in an array along with their predicted class. Clicking a thumbnail directs the whole-slide image viewport to the slide/region surrounding this sample. Double-clicking the thumbnail image cycles the assigned class labels. After correcting errors the user can commit these samples to the training set and update the classifier. They can then resume with additional feedback or finalize the classification rule.

with classification information at around 10X magnification. This allows the user to located an area of interest with the heatmap view then zoom in to examine the classification of individual objects. While viewing classification results the user can fix misclassification errors by double-clicking on an object. Double-clicking on an object will change its boundary to yellow indicating it is staged for addition to the training set. The object will be labeled with the inverse class it was classified with. That is, if the object was classified as the positive class it will be added to the training set with the negative class label. Once the user is finished selecting objects, clicking the "retrain" button will add them to the training set. After adding the new objects, the system, rebuilds the classifier and updates the display. At any time before clicking the "retrain" button, the user can "unstage" the object by double-clicking it. The viewer screen, using the combination of heatmaps and correcting misclassifications, is where the heatmap-based method of soliciting labels is realized. The heatmap screen provides a ranking of all the slides in the dataset by average uncertainty. Two views of the slides are display; one overlaid with an uncertainty heatmap, the other with a positive class density heatmap. They are ordered from top to bottom with the slide with the highest average uncertainty, or minimum prediction confidence, at the top. Clicking on any of the slide images with bring the user to the viewer screen with the appropriate slide image loaded.

In addition to the "viewer" and "heatmaps" screens, we provide a review page where the samples of the training set are displayed, organized by class and slide. This interface permits additional review of the labeled examples and enables the users to change labels using drag-and-drop. This feature facilitates multiple reviewers for collaboration among less and more experienced reviewers.

Figure 3.6: Viewer page — heatmap view

This view enables the overlay of heatmaps of classification confidence or positive class density in a whole-slide imaging viewport. Users can zoom into hotspots to review the classification rule predictions and to provide additional feedback in key regions that are likely to contain false positive or false negative predictions

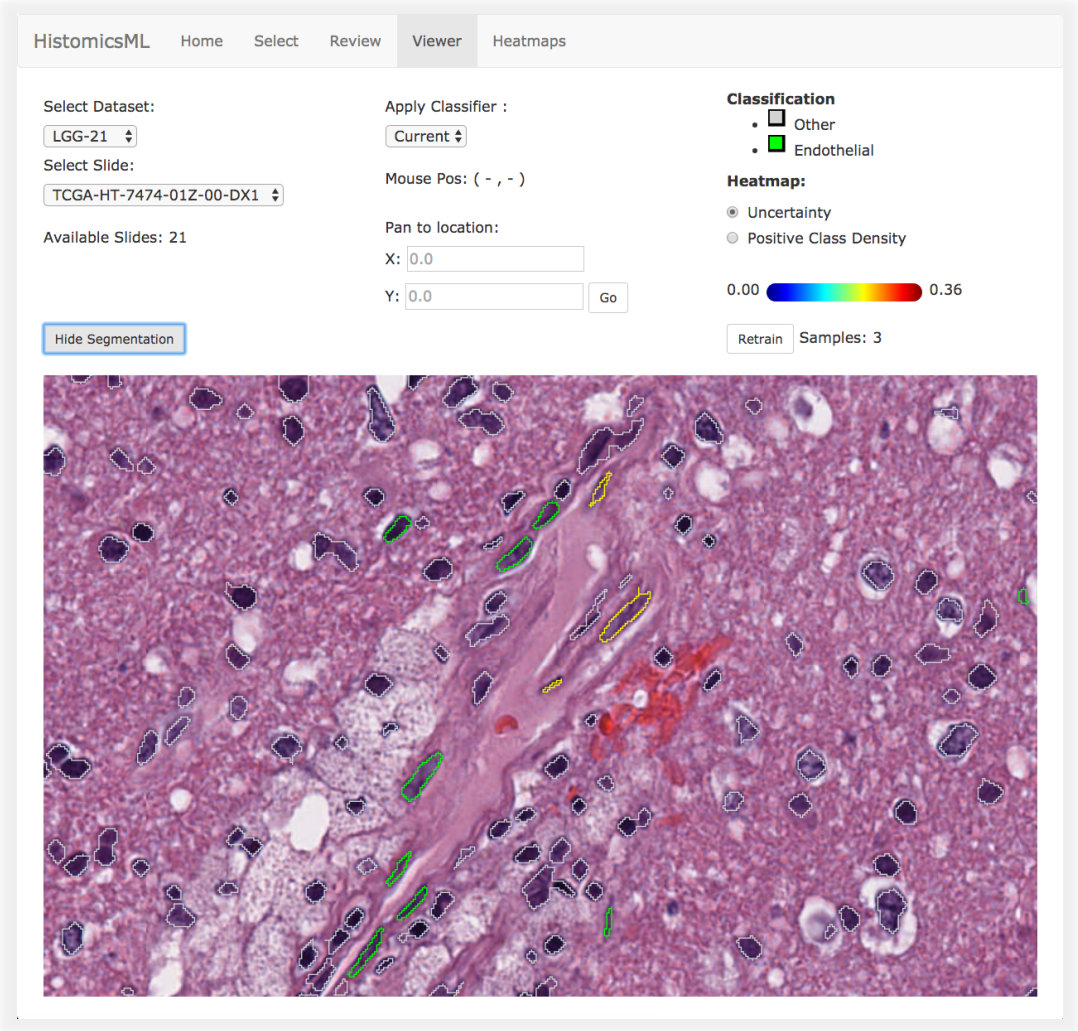

Figure 3.7: Viewer page — boundary view

Zooming into a hotspot region users can review and correct predictions for individual objects. Here cell nuclei positively classified as vascular endothelial cells are indicated with green boundaries and others indicated with white. Users can single-click objects in this view to correct prediction errors — cycling their class label and committing them to the training set. The classifier can also be updated from within in this view to visualize the results of feedback.

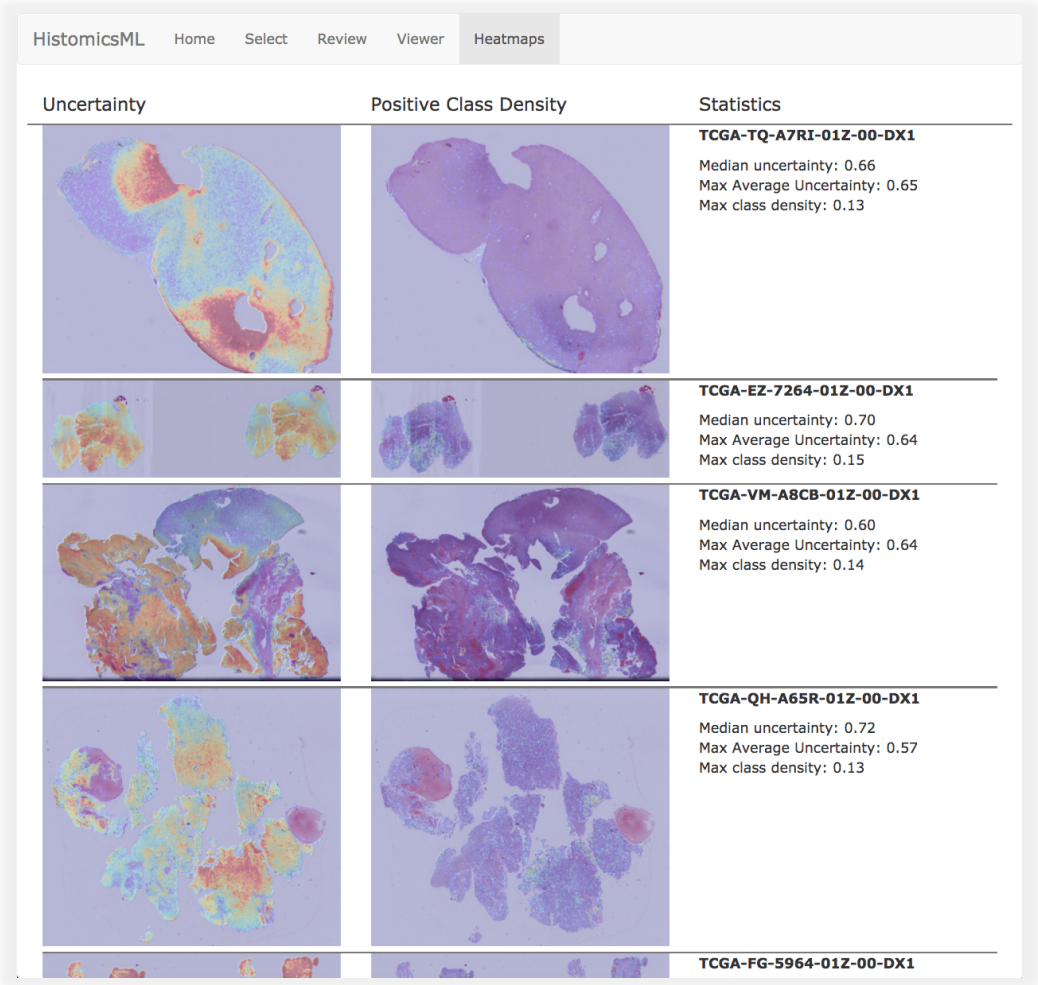

Figure 3.8: Heatmap page

This view displays slides overlaid with their uncertainty and positive class density heatmaps to prioritize feedback. Slides are sorted based on average confidence so that users can direct feedback to slides with large numbers of confounding samples. Clicking a thumbnail directs the user to the review screen for feedback. This page is updated and the slides resorted with each update of the classification rule.

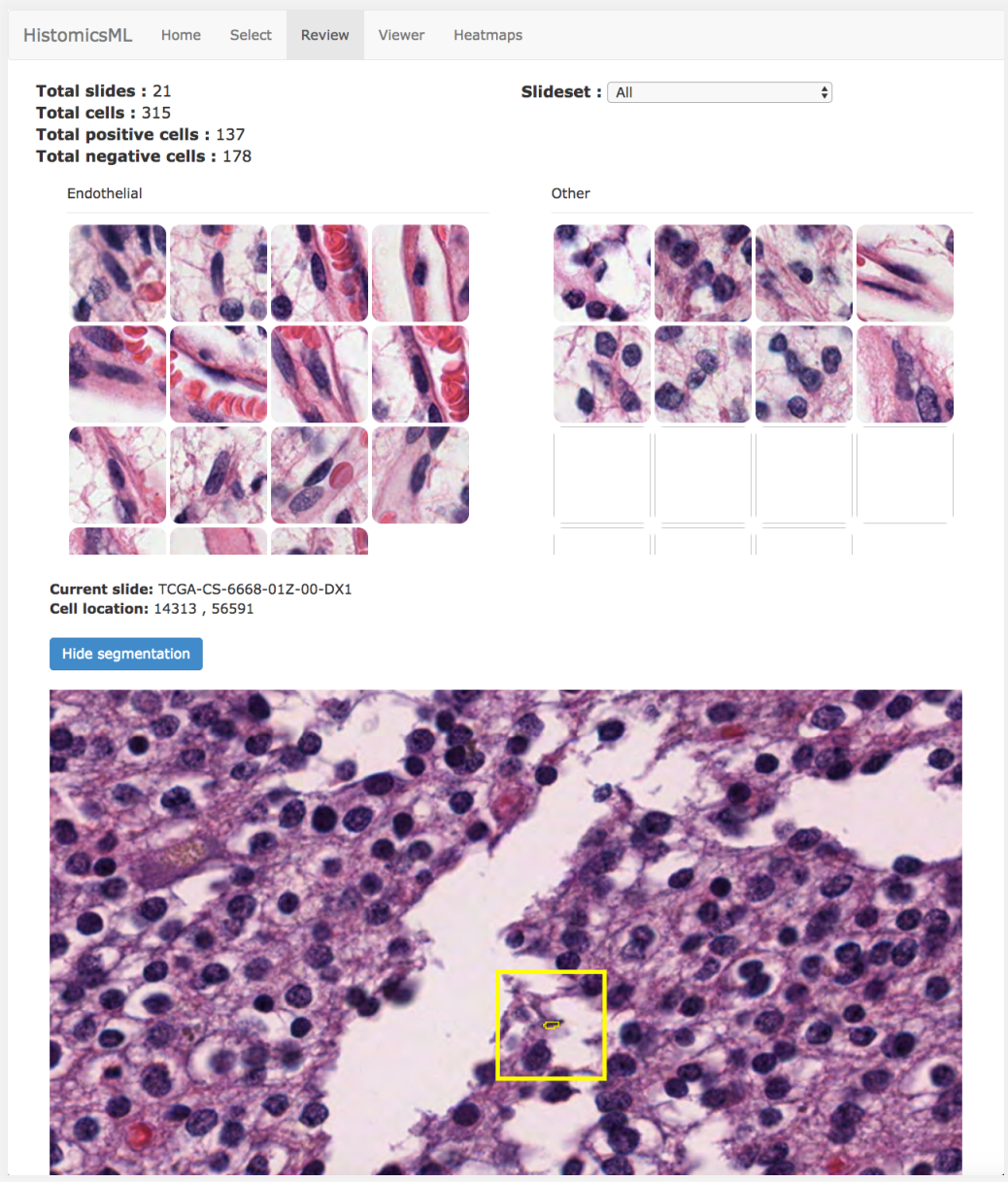

Figure 3.9: Review page

The review screen enables users to review and revise labeling provided for classification rule training. Labeled samples are organized in an array by slide and label/class. Users can browse the scrollable thumbnail gallery and change the label of a sample by drag-and-drop of the thumbnail images. Clicking a thumbnail directs the whole-slide viewport to the region of this sample.

3.2.7 Command line tools

A command line tool for applying trained classifiers outside of the user interface is also provided. This tool enables users to perform prediction and quantification of large datasets offline after training a classifier. The command line tool takes as input a classifier HDF5 file and an HDF5 file of histomic features for objects to be classified (in the input format described previously). The prediction function will generate a new HDF5 file that supplements the input file with predicted class labels and prediction confidence scores. The quantification tool provides basic quantification (counting) of objects in each slide, and generates a CSV file with the slide name, positive class count and negative class count for each slide present in the input HDF5 file.

3.3 Results

We used HistomicsTK (http://github.com/DigitalSlideArchive/HistomicsTK) to generate features for 360 million cell nuclei using 781 images (464 tumors) from The Cancer Genome Atlas Lower Grade Glioma (LGG) project. We trained a classification rule to identify vascular endothelial cell nuclei (VECN) and validated its performance using 67 slides not used in training.The VECN classifier was initialized by manually labeling 8 nuclei, and refined with both instance-based and heatmap-based learning to label 135 nuclei in 27 iterations. The VECN classifier is highly sensitive and specific, achieving an area-under-curve (AUC) of 0.9643 and improving over the initial rule with AUC=0.9234.

To further validate our VECN classifier, we correlated the mRNA expression levels of the endothelial marker PECAM1 with the proportion of cells positively classified as VECs in each specimen. Percent-VECN was significantly positively correlated with PECAM1 /CD31 expression (Spearman rho=0.24, p=1.27e–7). We note that the mRNA measurements originate from frozen materials where image analysis was performed on fixed and paraffin embedded tissues that originate from same primary

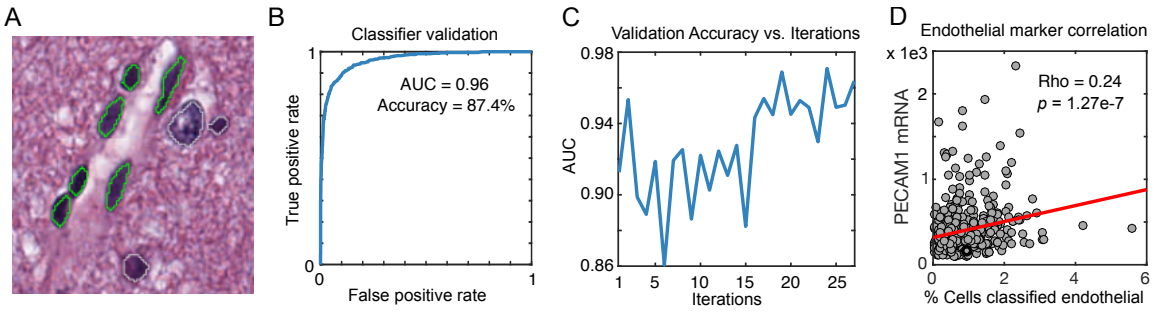

Figure 3.10: Classifying vascular endothelial cells in gliomas

(A) We trained a classification rule to identify vascular endothelial cell nuclei in lower-grade gliomas (highlighted in green) using data from The Cancer Genome Atlas (TCGA-LGG). (B) This classifier achieved an area-under-curve (AUC) accuracy of 0.96 with only 135 labeled nuclei for training. (C) A total of 27 active learning iterations were performed, improving the AUC from 0.92 to 0.96. (D) For additional validation we correlated the percentage of positively classified endothelial cells in each sample with mRNA expression levels of the endothelial marker PECAM1 using measurements from TCGA frozen specimens.

tumor but with unknown proximity to the frozen materials.

To evaluate system responsiveness, we measured the time required for the update-predict process including classifier rule training, classification of all unlabeled objects, sampling of instances for active learning, and calculation of heatmaps. We evaluated various sized datasets ranging from 106–107 objects. We observed a consistent linear increase of 1 second per 5.5 million objects on our 24-core server. This translates to a 10 second learning cycle for a 50 million object dataset.

We selected 67 digital slides from the LGG cohort to serve as a validation set for measuring the performance of a VECN classifier. In each slide, we selected a field containing a mixture of nuclei from vascular endothelial cells and other cell types (tumor nuclei and inflammatory cells, for example). Each correctly segmented nucleus in the field was labeled as either vascular endothelial or other. Incorrectly segmented nuclei, or nuclei that were too ambiguous to classify were ignored. In total 2479 cell nuclei were labeled. Each annotation was reviewed by a board-certified neuropathologist using our review function. Classifiers used for reporting accuracy results were trained using a mixture of instance-based and heatmap-based review.

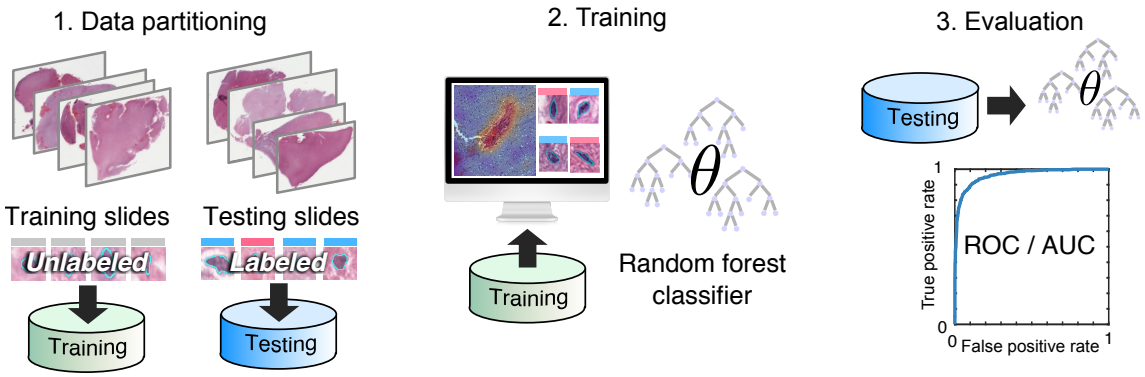

Figure 3.11: Validation of classifier performance

(A) We trained a classification rule to identify vascular endothelial cell nuclei in lower-grade gliomas (highlighted in green) using data from The Cancer Genome Atlas (TCGA-LGG). (B) This classifier achieved an area-under-curve (AUC) accuracy of 0.96 with only 135 labeled nuclei for training. (C) A total of 27 active learning iterations were performed, improving the AUC from 0.92 to 0.96. (D) For additional validation we correlated the percentage of positively classified endothelial cells in each sample with mRNA expression levels of the endothelial marker PECAM1 using measurements from TCGA frozen specimens.

Chapter 4 Microvascular proliferation and hypertrophy analysis

4.1 Introduction

Images of histology contain important information that can be difficult or impossible to ascertain through genomic assays. Recent focus on deconvolution of gene expression profiles can accurately estimate the fractions of cell types in a sample, but these approaches cannot provide spatial or morphologic information that often contains considerable prognostic or scientific value. While molecular phenomena drive phenotypes, quantitative histologic analysis provides more immediate readouts of information that are difficult or impossible to obtain from genomic profiling. These data can add a new dimension to studies of complex processes like lymphocytic infiltration, angiogenesis, and tumor-stromal interactions, or the functional annotation of genetic alterations. Growth in tools for genomic analysis in the last decade provide a roadmap for increasing utilization of histologic imaging data. Open-source development can engage a broader technical audience with interests in machine-learning and image analysis, and more focus on creating enabling software infrastructure for data visualization, management and analysis can place data in the hands of experts who have driving biological and clinical questions. We used HistomicsML to analyze microvascular phenotypes in gliomas, illustrating how datasets that link histology, clinical outcomes, and genomics can be mined to investigate prognostic biomarkers and genotype-phenotype relationships. We showed that quantitative metrics of microvascular phenotypes are associated with grade and molecular subtype, and that prognostic models based on these metrics perform as well as grade in predicting survival. Integration with genomic data identified both well-recognized molecular

pathways associated with angiogenesis, and also more interesting subtype-specific molecular pathways enriched with phenotype-correlated genes.

4.2 Principal Curves

Principal curves are smooth 1-dimensional curves that pass through the center of a p-dimensional dataset. They are a generalization of linear principal components that provide a summarization of the data. The first definition of principal curves was proposed by Hastie and Stuetzle [42]. Typically when summarizing data, a variable is chosen as a response relative to some other explanatory variable or variables. For instance, a patients blood pressure relative to their body mass index. The idea is to generate a model in which to predict the response given the explanatory variable(s). There are many techniques which can easily achieve this, such as linear regression, though there are times when there is no response variable we are interested in. Instead, we would like to summarize the joint behavior of all the variables. This is the motivation behind Hastie and Stuesle's development of principal curves.

Hastie and Stuetzle's definition of principal curves is based on self-consistency, where a one-dimensional curve passes through the center of a p-dimensional point cloud. The curve is a vector of p functions $f(\lambda) = (f1(\lambda), f2(\lambda), ..., fp(\lambda))$ with λ providing an ordering along the curve. The parameterization of f is not unique, any monotone transformation could be applied to λ and along with the appropriate modification of the functions f_1 to f_p, and the resultant curve will be the same. Given a dataset X, the projection index $\lambda_t(x) = \sup \lambda : x - f(\lambda) = \inf x - f(\mu)$ is the value of λ which minimizes the distance between x and $f(\lambda)$. By the H&S definition, the curve is called self-consistent if $E(X|\lambda_f(X) = \lambda) = f(\lambda)$ for all λ. Additionally, a principal curve cannot intersect itself. Once a principal curve is constructed, the "summary" value of a data point can be determined by projecting the data point onto the curve and calculating the arc length from the beginning of the curve to the

point of projection.

The algorithm for the construction of principal curves for datasets is an iterative method that alternates between a projection step and a conditional-expectation step. The complete algorithm can be seen below. For an N x p dataset X, the curve f is represented by N tuples (λ, f) in increasing order of λ to form the vertices of a polygon. The value of $ \lambda $ is its arc length along the current curve, where $ \lambda_0 = 0 $ and λ_n is the arc length of the curve from λ_0. The curve is initialized to the first linear principal component of X. In the projection step, the nearest point p to x_i is found on the curve then λ_i is set to the arc length from the beginning of the curve to p. In the conditional-estimation step, the λ_i's are used to update the functions f_1, f_2, f_p. The obvious way to calculate $E(X|\lambda_f i = \lambda_i)$ is to take the mean of all the data points in X that project onto the curve at λ_i. Since there tends to be only one data point in X that projects onto the curve at λ_i, a local averaging needs to be done with the data points x_j whose projections λ_k are close enough to λ_i. This can be achieved with scatterplot smoothing such as [43] and [44]. The smoother is applied to each of the (λ, f_i) tuples, after which the algorithm returns to the projection step using the updated curve to calculate new λ_i'. This process is repeated until the change in the squared distance from X to f falls below some threshold t.

Algorithm 1 The Principal-Curve Algorithm

Require: $f^{(0)}(\lambda) = \bar{x} = \mathbf{a}\lambda$, where $\mathbf{a}$ is the first linear principal component
Require: $\lambda^{(0)}(\mathbf{x}) = \lambda_{f^{(0)}}(\mathbf{x})$
  repeat
    $f^{(j)}(.) = E(\mathbf{X}|\lambda_{f^{(j-i)}}(\mathbf{X}) = .)$
    $\lambda^{(j)}((x)) = \lambda_{f(j)}(\mathbf{x})\ \forall\ \mathbf{x} \in h$
    $D^2(h, f^{(j)}) = E_{\lambda^{(j)}} E[\|(\mathbf{X}) - f(\lambda^{(j)}(\mathbf{X}))\|^2 \mid \lambda^{(j)}(\mathbf{X})]$
  until $\Delta D^2(h, f^{(j)}) < t$

There has been several other definitions of principal curves that have followed Hastie and Stuetle's. Kegl et al [45] relaxed the original definition to curves that

achieve the minimum expected squared distance from points in the dataset to the location they project onto the curve. Along with this relaxation, they propose the polygonal line algorithm. This algorithm constructs the curve by adding vertices one by one to a polygonal line and optimizing the vertex positions after each addition by minimizing the squared distance of the data to the curve. Verbeek et al. [46] proposed an algorithm similar to Kegl et al., the k-segments algorithm. Tibshirani [47] proposed a probabilistic definition of principal curves where the curve minimizes a penalized log-likelihood measure. This definition employs a mixture model for the data. Though the definitions vary, the essence remains the same — principal curves provide a summarization of the joint behavior over all the dimensions in the data.

4.3 Phenotyping Microvascular Structures in Gliomas

We developed and validated quantitative metrics to describe the phenotypes of microvascular structures in gliomas. Gliomas are among the most vascular solid tumors, with microvascular structures undergoing apparent transformations in response to signaling from neoplastic cells. Microvascular hypertrophy, or thickening of microvascular structures, represents an activated state of individual endothelial cells that show nuclear and cytoplasmic enlargement associated with increased transcription and translation. Microvascular hyperplasia, on the other hand, represents the accumulation, clustering and multilayering of endothelial cells due to their local proliferation. While these changes are understood to accompany disease progression, the prognostic value of quantitative phenotyping of microvasculature in gliomas has not been established in the era of precision medicine, and may be beyond the capacity of human visual recognition.

The hypertrophy of individual nuclei was scored using a nonlinear function to model the continuum of VECN morphologies. This scoring was validated by manually

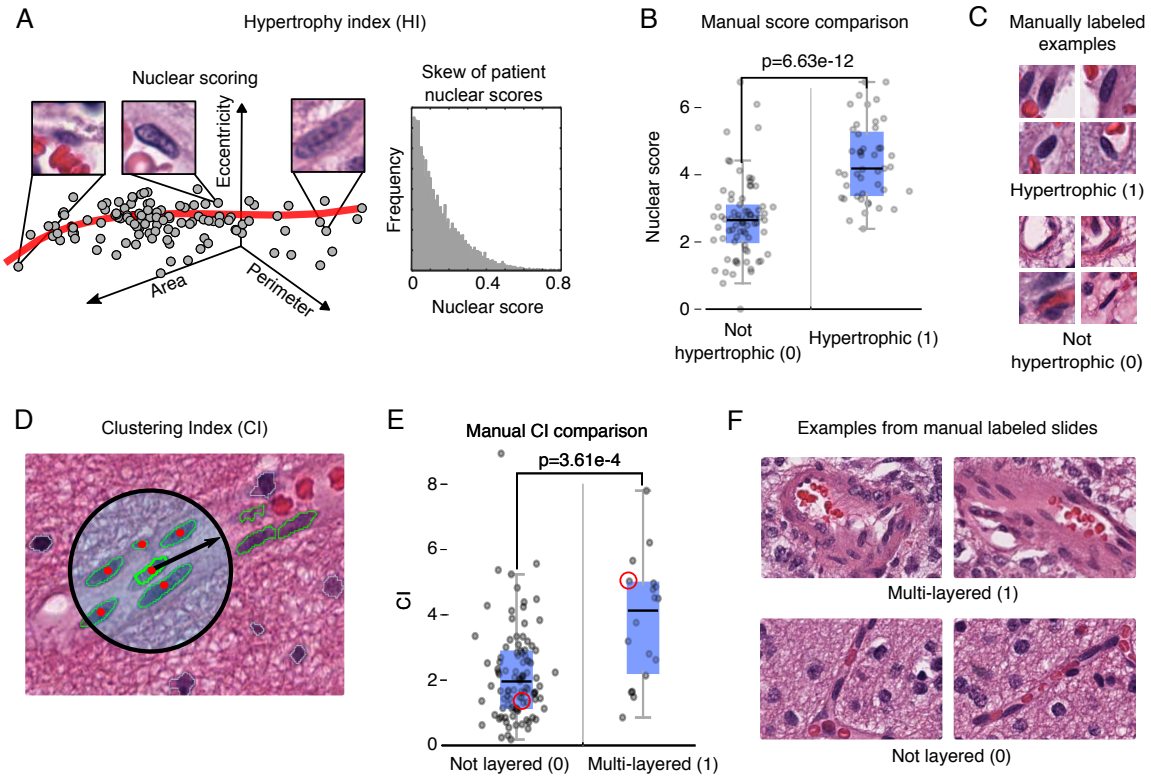

Figure 4.1: Quantitative phenotyping of microvasculature in gliomas

Microvascular structures undergo visually apparent changes in response to signaling within the tumor microenvironment. (A) We measured nuclear hypertrophy using a nonlinear curve to model the continuum of VECN morphologies. A hypertrophy index (HI) was calculated for each patient to measure the extremity of nuclear hypertrophy score values. (B) We validated nuclear scores using a set of manually labeled nuclei (hypertrophic (1) / non-hypertrophic (0)) (Wilcoxon p=6.63e-12). (C) Examples of cell nuclei used in validation. (D) We implemented a clustering index (CI) to measure the spatial clustering of vascular endothelial cells as a readout of hyperplasia. CI measures the average number of endothelial nuclei within a 50-micron radius of each VECN in a sample. (E) CI was compared to manual assessments of hyperplasia (multi-layered (1) / not layered (0)) (Wilcoxon p=3.61e-4). (F) Example microvascular structures from two of the slides used in comparison

labeling 120 VECN (45 hypertrophic, 75 non-hypertrophic) to establish that hypertrophy scores are significantly higher for hypertrophic nuclei (Wilcoxon p=8.63e–12). A hypertrophy index (HI) was calculated to summarize hypertrophy at the patient level. Within the distribution of nuclear scores for a patient, the more hypertrophic nuclei appear in the distribution tail. We used distribution skew to capture tail thickness so that more positive HI values indicates the presence of more hypertrophic VECN.

Hyperplasia and proliferation were measured using a VECN clustering index (CI) to capture the extent of spatial clustering of VECNs. CI was calculated at the patient level as the average number of VECNs within a 50-micron radius of each VECN in that patient. CI was also compared to manual slide-level assessments of microvascular proliferation in 137 images (18 slides presenting multilayered phenotypes /119 not presenting) to show higher CI measures associate with images where multi-layered microvascular structures are present (Wilcoxon p=3.61e–4).

The Endothelial clustering index. CI was calculated using a spatial statistic based on a modified version of Ripley's K-function [48]. Ripley's K-function, originally developed for geographic and epidemiological applications, captures the "degree of spread" of events in spatial domains. Since microvascular hyperplasia and proliferation are, by definition, associated with increases in VECN density, we excluded Ripley's density normalization in our proliferation metric. We also ignored edge-effect corrections due to the extremely large number of objects and the relative rare scarcity of objects at the edge of tissue sections. CI was calculated as:

$$CI(\tau) = \frac{1}{K}\sum_{i=1}^{K} |\Omega_i|,\ \Omega_i = \{d_{i,j} \leq \tau\},\ d_{i,j} = \|x_i - x_j\|_2 \tag{4.1}$$

where K is the number of objects/nuclei in the image, $d_{i,j}$ is the Euclidean distance between objects i, j, and τ is a positive search radius. This effectively calculates the average number of objects within distance τ of each object in the slide. In our

experiments, CI was calculated for objects positively classified as VECN with a search radius of $\tau = 50$-microns. KD tree indexing was used to accelerate the computational search for neighboring VECN.

Endothelial hypertrophy index (HI) was calculated by scoring the morphologies of individual nuclei and calculating population statistic for each patient. Hypertrophy in VECN was first quantified using a principal curve technique to score each nucleus. We developed our own principal curve routines that will be added to the HistomicsTK library. When applied to histomic feature profiles, the fitted principal curve enables the modeling of the morphologic continuum of VECN from small, thin normal appearing nuclei to enlarged hypertrophic nuclei that are reacting to the surrounding tumor microenvironment. The principal curve models the feature vector values f_i of object i as

$$f_i = g(\lambda_i) + e_i \tag{4.2}$$

where g is a 1D nonlinear curve in feature space parameterized by the free variable λ, and e_i is a random component. After fitting the principal curve, each nucleus is scored by projecting its feature profile onto the curve, and calculating the path length from the projection to the curve origin

$$s_i = \int_{\lambda_0}^{\lambda_i} \|g'(z)\| dz \tag{4.3}$$

where λ_0 is the origin of the curve, λ_i is the location of the least-squares projection of f_i onto the curve, and g' is the curve tangent function. Nuclei that are more hypertrophic in the morphologic continuum will have longer path length values and thus higher nuclear scores. In our analysis we constructed a principal curve using histomic shape features for area, eccentricity and perimeter. The directionality for the beginning/end of the curve was established by initializing the curve fitting with a single normal appearing nucleus and a single hypertrophic nucleus.

At the patient-level, HI was calculated to represent the population-level skew of VECN towards more hypertrophic morphologies. HI was measured as the negative skew of nuclear scores

$$SI = -\frac{1}{K}\sum_{i=1}^{K}(s_i - \bar{s})^3 \tag{4.4}$$

where K is the number of objects/nuclei in the image, s_i is the hypertrophy score of object i, and s_i is the mean hypertrophy score. This statistic increases in value as the right tail of the score distribution grows.

4.4 Microvascular phenotypes accurately predict survival

Diffuse gliomas are the most common adult primary brain tumor and are uniformly fatal. Survival of patients diagnosed with infiltrating glioma depends on age, grade and molecular subtypes that are defined by IDH mutations and co-deletion of chromosomes 1p and 19q [49]. The lower grade gliomas (grades II, III) exhibit remarkably variable survival ranging from 6 months to 10+ years. Highly aggressive IDH wild-type (IDHwt) gliomas having an expected survival of 18 months. Gliomas with IDH mutations and 1p/19q co-deletions (IDHmut-codel) have the best outcomes, with some patients surviving 10+ years. Gliomas with IDH mutations and lacking co-deletions (IDHmut-non-codel) have intermediate outcomes with survival ranging from 3–8 years. The accuracy of grade in predicting outcomes varies depending on subtype [50].

We first investigated associations between hyperplasia and hypertrophy, grade and molecular subtype in the TCGA cohort using CI and HI (See Figure 4.2 A and Table 4.1) We found that IDHwt gliomas exhibit a greater degree of microvascular hyperplasia than the less aggressive IDHmut-non-codel and IDHmut-codel gliomas (Kruskal-Wallis p=8.43e–6). Increased microvascular hyperplasia is also strongly as-

sociated with higher grade in each molecular subtype (Wilcoxon IDHwt p=4.99e–4, IDHmut-non-codel p=1.96e–6, IDHmut-codel p=2.08e–4). While differences in microvascular hypertrophy across subtypes (Wilcoxon p=0.747) and grades were not statistically significant, the median HI for grade III gliomas was higher in each molecular subtype. We also explored subtypes by using median CI or HI values to stratify patients into high/low risk groups. (See Figure 4.3) Kaplan-Meier analysis found these "digital grades" were marginally prognostic within IDHmut-codel gliomas (log-rank CI p=6.87e–2, HI p=5.09e–2) and IDHwt gliomas (CI p=4.68e–2), but remarkably neither CI nor HI could discriminate survival in the IDHmut-non-codel gliomas. Similar discrimination patterns were observed when stratifying by WHO grade.

After investigating associations with grade and subtype, we used a prognostic modeling approach to evaluate the prognostic value of microvascular phenotypes. Cox hazard models were created with various combinations of predictors including grade, subtype, CI and HI. (see Figure 4.2 B) Patients were randomly assigned to 100 non-overlapping training /validation sets, and each was used to train and evaluate a model using Harrell's concordance index [51]. We found that the models based on HI+CI+grade+subtype have the best performance, and significantly outperforming grade+subtype models (Wilcoxon p=3.35e–11), suggesting that HI, CI measurements have prognostic value independent of grade and subtype. Models based on grade+subtype and HI+CI+subtype have equivalent performance (p=0.915), and models based on HI+CI slightly outperform grade-only models (Wilcoxon p=3.02e–8). Finally, we found that although HI-only models perform only slightly better than random (median c-index 0.58), HI+CI models perform significantly better than CI-only models (p=9.09e–17).

Labels to validate hypertrophy index and clustering index were acquired by manual inspection of digital slide images by a board-certified pathologist who was blinded to the computer-generated HI and CI scores. For hypertrophy, a selection of 120 cell

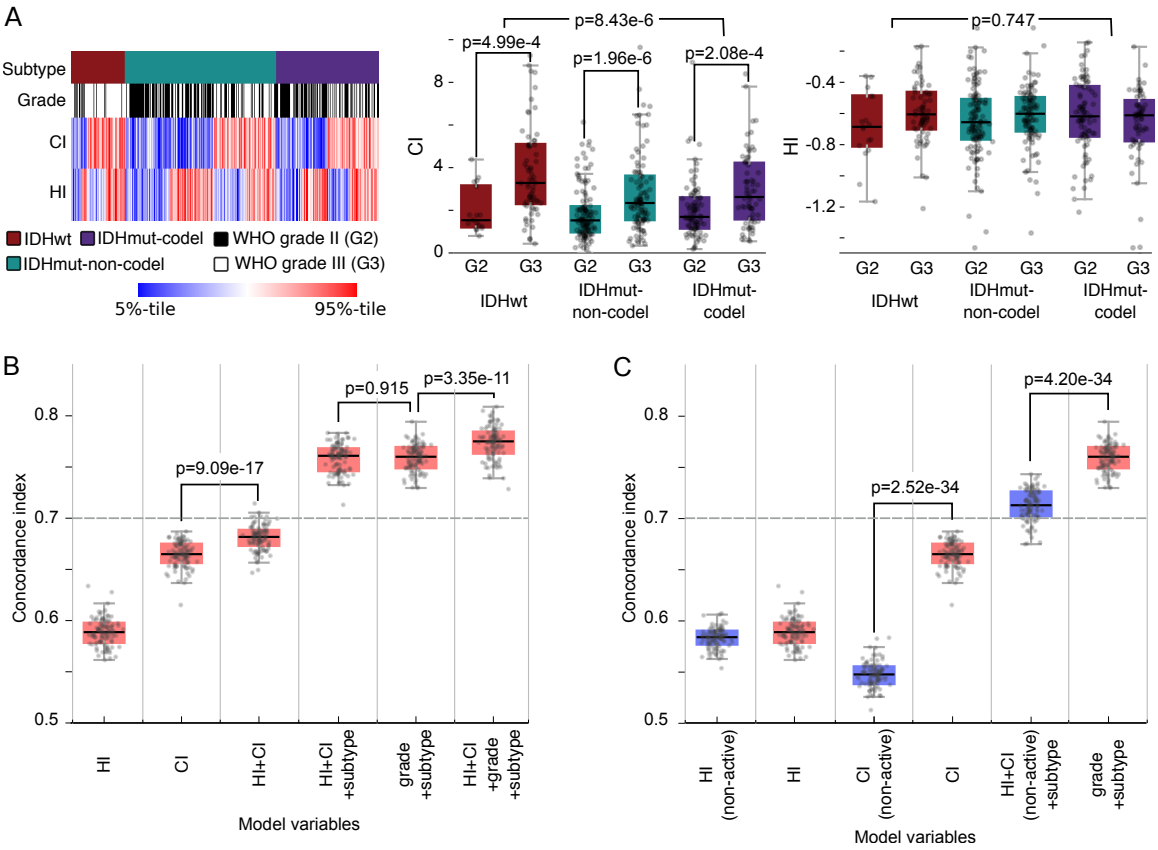

Figure 4.2: Predicting survival with microvascular phenotypes.

(A) HI and CI were compared with important clinical metrics including WHO Grade, molecular subtypes and survival. CI is significantly associated with subtype (Kruskal-wallis p=8,43e-6) and grade within each subtype. HI was not significantly associated with either subtype (Kruskal-wallis p=0.747) or grade. (B) We trained cox hazard models using combinations of phenotypic and clinical predictors to assess and compare their prognostic relevance. Models were trained and evaluated using 100 randomized training/testing sets. $HI+CI$ models perform better than grade-only models (Wilcoxon p=3.02e-8). We also noted that $HI+CI$ predictors are relevant independent of subtype (p=2.80e-34) and grade (p=3.35e-11). There was no difference between $HI+CI$+subtype and Grade+subtype models (p=0.915). Although an HI-only model has weak performance, the $HI+CI$ model outperforms the CI-only model (p=9.09e-17).

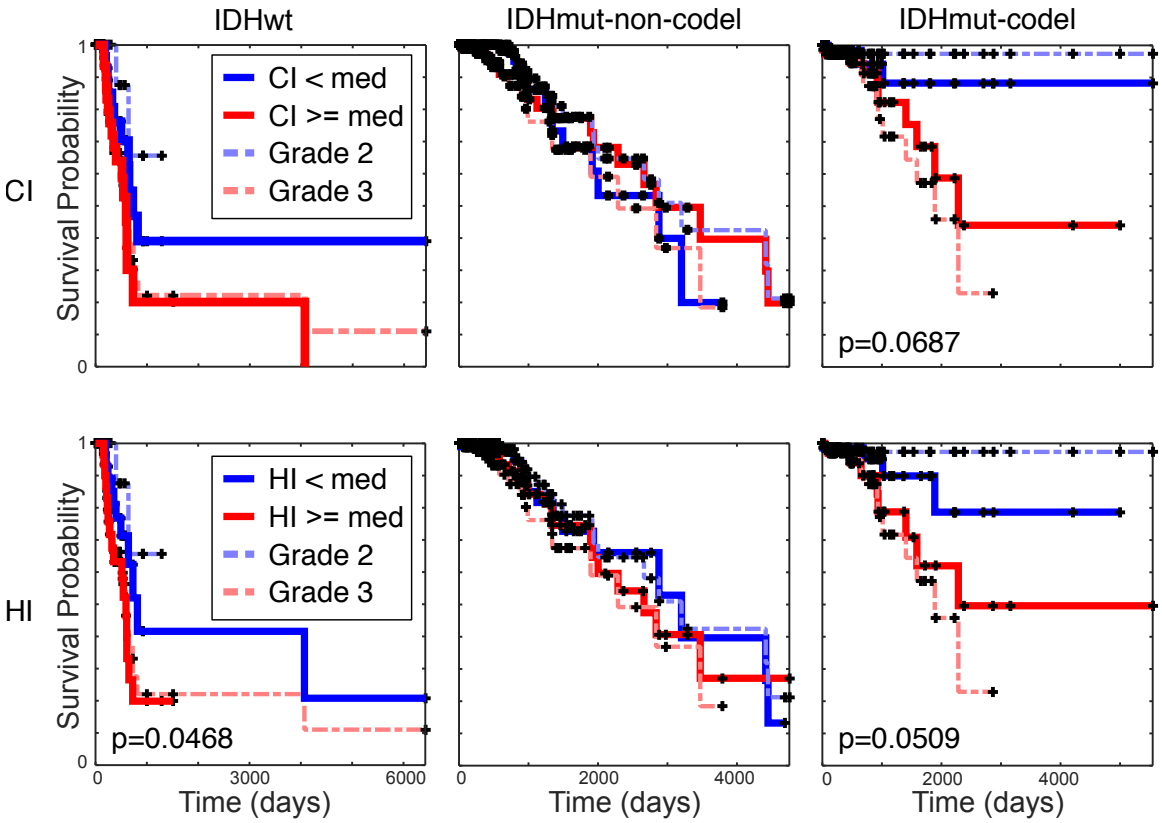

Figure 4.3: Kaplan-Meier analysis

Median values of CI or HI were used to stratify patients into low/high risk groups for Kaplan-Meier analysis in each molecular subtype (grade is shown for comparison). HI and CI can discriminate survival marginally in IDHwt gliomas ($CIp = 4.68e-2$) and IDHmut-codel gliomas (log-rank $CIp = 6.87e-2, HIp = 5.09e-2$), but not in IDHmut-non-codel gliomas. We observed that grade is marginally better at discriminating survival than CI or HI (IDHwt p=5.86e-2, IDHmut-non-codel p=7.93e-2, IDHmut-codel p=1.93e-2), but is also only marginal predictive of survival for IDHmut-non-codel gliomas.

nuclei classified as VECN were labeled as either hypertrophic (45) or non-hypertrophic (75) using our validation set creation tool. The nuclear hypertrophy scores were compared for these manually labeled nuclei using a non-parametric Wilcoxon text. For clustering index, 137 slides were manually reviewed to determine if they present microvascular hyperplasia and proliferation (multi-layered vessels). The CI scores for slides presenting multi-layered vessels were compared to scores for the control slides using the Wilcoxon test.

4.5 Genomic Integration Identifies Phenotype Associated Pathways

The molecular mechanisms of angiogenesis in gliomas have been studied extensively, and are targeted through anti-VEGF therapies like Bevacizumab [52]. To investigate the molecular pathways associated with microvascular phenotypes, we performed gene-set enrichment analyses [53] that correlated CI and HI with mRNA expression. We analyzed for IDHwt and IDHmut-codel gliomas separately since mechanisms may vary across subtype. IDHmut-non-codel gliomas were not analyzed. A partial list of pathways enriched at FDR q<0.25 significance is summarized in Table 4.5 .

Given the proximal association between angiogenesis and hypoxia in cancer biology, we expected our pathway analysis to identify strong relationships between microvascular phenotypes and classic hypoxia and metabolic glycolysis pathways. We found both HIF2A and VEGFR1/2 mediated signaling pathways were both upregulated with increasing hyperplasia and hypertrophy. Among the most strongly phenotype-correlated genes were those involved in hypoxia and angiogenesis including VEGFA, VHL, ARNT, PGK1 [54], ADM [55], and EPO, as well as glycolytic response mediators HK1, PGK1, ALDOA, PFKFB3, PFKL and ENO1. Angiopoietin receptor [56] and Notch signaling [57] pathways were also significantly enriched in

Pathway Group	Pathway name	Leading-edge genes	Subtype / metric (directionality)	Nominal p-value (FDR q-value)
Classiscal angiogenesis pathways	* HIF1-alpha transcription factor	*PFKL, PFKFB3, ALDOA, EGLN1/3, PGK1, HK1*	IDHmut-codel / HI (+) IDHmut-codel / CI (-)	0.033 (0.179) < 0.001 (0.116)
	HIF2-alpha transcription factor	*CREBBP, EP300, VEGFA, VHL, ARNT, TWIST1*	IDHwt / CI (+) IDHmut-codel / CI (+)	0.004 (0.017) 0.024 (0.116)
	VEGFR1/2 mediated signaling	*BRAF, MAPK1/14*	IDHwt / HI (+) IDHmut-codel / CI (+)	0.012 (0.144) 0.009 (0.12)
	* VEGFR1 specific signals	*MAPK1, NRP1/2*	IDHwt / HI (+)	0.007 (0.19)
	Angiopoeitin receptor TIE-2 mediated signaling	*ANGPT2, MAPK1/14, MMP2, NFKB1, PIK3C*	IDHwt / HI (+) IDHwt / CI (+) IDHmut-codel / CI (+)	0.014 (0.147) 0.015 (0.063) 0.009 (0.087)
	* PDGFRA signaling	*GRB2, PIK3CA, FOS, PDGFRA, PLCG1*	IDHmut-codel / HI (+)	0.014 (0.08)
Developmental signalling pathways	Notch signaling network	*NOTCH1/2/3/4, MAML1/2, JAG1, EP300, MYC*	IDHwt / CI (+) IDHmut-codel / CI (+)	< 0.001 (< 0.001) 0.02 (0.146)
	* Notch mediated HES/HEY network	*HEY1, NOTCH1, MAML1/2, HIF1A, TCF3, EP300*	IDHwt / HI (+) IDHwt / CI (+)	0.007 (0.143) < 0.001 (< 0.001)
	* WNT signaling	*WNT3A, GSK3B, CAV1*	IDHmut-codel / CI (+)	0.021 (0.085)
	* Regulation of nuclear beta catenin signaling		IDHmut-codel / CI (+)	0.004 (0.054)
	* GLI-mediated Hedgehog signaling		IDHwt / CI (+)	0.015 (0.063)
Other pathways	* Regulation of SMAD2/SMAD3 signaling	*SMAD3/4, MAPK1, MAP3K1, TGFBRAP1*	IDHwt / HI (+) IDHwt / CI (+)	0.002 (0.008) 0.024 (0.139)
	* SMAD2/SMAD3 nuclear signaling	*SMAD3/4, CDK2/4, CDKN1A, AKT1, MYC*	IDHwt / CI (+)	< 0.001 (< 0.001)
	* FOXM1 transcription factor network	*FOXM1, GSK3A, MAP2K1, MYC, FOS, MMP2*	IDHmut-codel / CI (+)	< 0.001 (< 0.001)

Table 4.1: Molecular pathways enriched with phenotype-correlated transcripts

Gene set enrichment analysis of the correlations between HI / CI and gene expression identified multiple pathways associated with gliomas and vascularization. Many of the significantly enriched pathways are specific to one molecular glioma subtype.

both subtypes.

Pathways with enrichment specific to IDHwt gliomas included Notch mediated regulation of HES/HEY [58], GLI-mediated hedgehog signaling [59], and SMAD signaling [60], all of which have been linked to angiogenesis or regulation of structure and fate in vascular endothelial cells. Pathways with enrichment specific to IDHmut-codel gliomas included WNT and beta-catenin signaling, and PDGFRA signaling (PDGFRA amplification is frequent in IDHmut-codel gliomas). WNT and beta-catenin signaling are known to act synergistically with Notch to regulate endothelial differentiation and vasculogenesis and to help regulate HIF1A in hypoxic conditions [61].

We note that angiogenesis generally accompanies disease progression in gliomas, and that pathway enrichments may reflect molecular patterns associated generally with disease progression in addition to angiogenesis-related microenvironmental signaling.

4.6 Active learning training improves prognostication

To evaluate the benefit of active learning training, we repeated our experiments using a classification rule trained with a standard approach where the expert constructs a training set without the aid of active learning feedback. Using the same image collections described above, 135 cell nuclei were labeled in the training images (roughly evenly split between VECN and non-VECN). A classification rule was trained using these labels and applied to the dataset to compare classification and prognostic modeling accuracies with the active learning classifier. The validation AUC of the standard classifier was 0.984 (AUC for the active learning classifier was 0.964). While the AUC measured on the validation set was higher, the standard learning classifier is much less specific on the entire dataset, producing very high estimates of percent-VECN in

the TCGA cohort ranging from 7.1–57.2% (compared to 0.025.6% percent-VECN for active learning). Agreement between PECAM1 expression and percent-VECN was much lower for the standard classifier percent-VECN (Spearman rho=0.16 versus 0.24). We calculated updated HI and CI metrics using the standard classifier results and found that prognostic models based on these metrics were no longer predictive of survival (see Figure 4.2 C). The median c-index of models based on CI alone fell to <0.55 (Wilcoxon p=2.52e–34). Models incorporating HI+CI+subtype were also no longer equivalent to subtype+grade models (p=4.20e–34), and only slightly better than subtype.

Chapter 5 Deep Learning

5.1 Introduction

The methods and tools so far presented have relied on the "engineering" of features to characterize, or rather describe, the objects that machine learning methods will be applied to. Theses features are predetermined manually and are usually taken from characteristics domain experts naturally use to describe the objects. While the engineered features have served well for many applications of machine learning, there are some drawbacks when using them for classification of objects from whole-slide images. The most significant drawback is the location and segmentation of cell nuclei in the images. Nuclear segmentation is a computationally expensive process prone to mistakes. The main challenge in segmenting nuclei from whole slide images is that the tissue is a 2-dimensional section partitioned from a 3-dimensional object. The sectioning of the sample often resulting in nuclei sectioned at odd angles, damage and overlapping of tissue to name a few issues. Additionally, separation of densely cluster nuclei is a long-standing problem [62].

Another drawback is the selection of the features. While the selected features may work well for humans, there may be patterns in the images not detectible by humans that may better characterize the objects. It would be advantageous to let the algorithms determine the representation for the object of interest rather than relying on engineered features. The following will describe a method which utilizes deep artificial neural networks to not only classify objects but also learn representations of them: Deep learning.

Feature learning methods based on deep convolutional networks have demonstrated remarkable performance in general image analysis tasks. [63] These methods

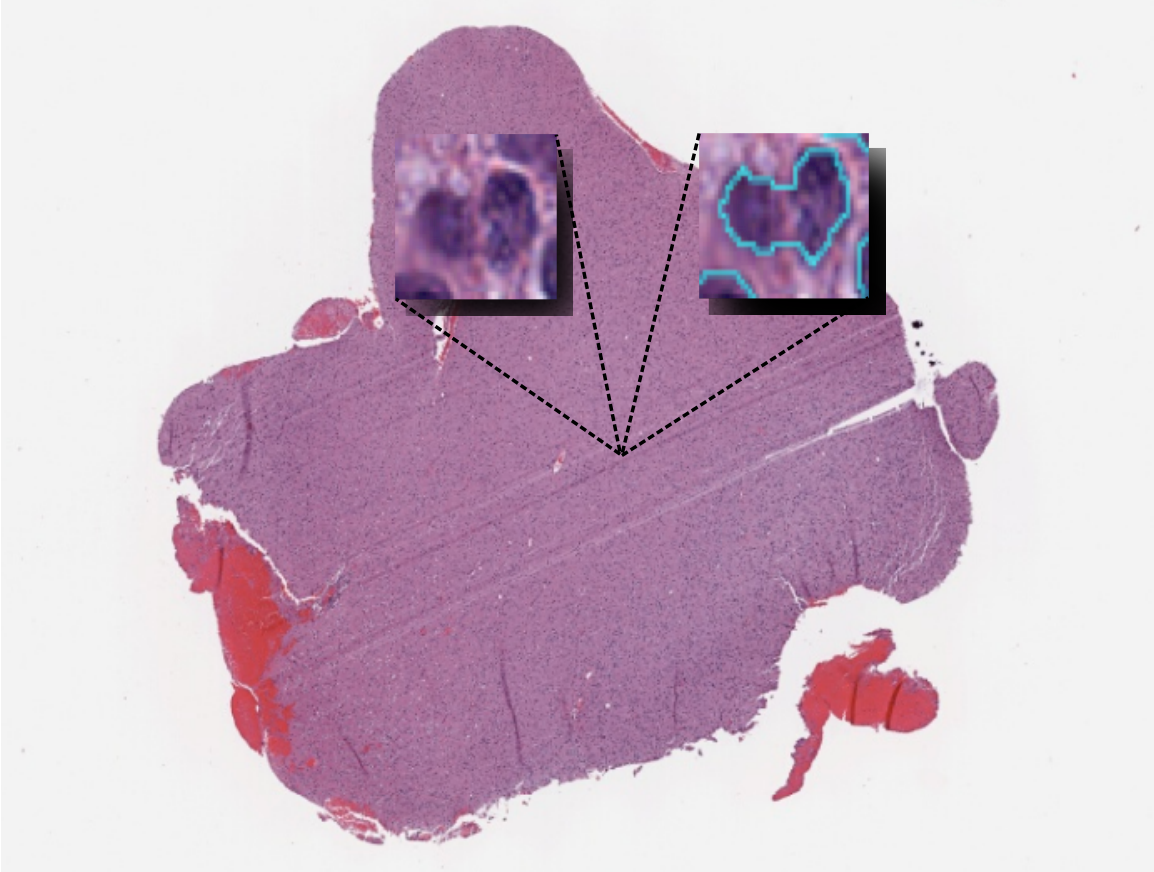

Figure 5.1: Segmentation error

The segmentation algorithm could not discern between the 2 nuclei causing erroneous features to be generated.

learn task specific features directly from raw image data using multiple levels of representations. Starting with the raw image data, each level learns simple but non-linear filters that transforms the representation in the previous layer into a higher, more abstract representation. With enough layers, very complex functions can be learned.

5.2 Artificial Neural Networks

Artificial neural networks are a supervised learning method that use artificial neurons as the basic unit of computation. Inspired by the animal brain, the Threshold Logic Unit (TLU), developed by Warren McCulloch and Walter Pitts, is considered the first artificial neuron [64], Like an animal neuron, the TLU has stimuli, or inputs. These inputs can have a value of 0 or 1 and are summed together. When the sum of the inputs is equals to or exceeds a threshold theta, the neuron becomes activated and produces an output signal of 1. Additionally, the TLU has one or more inhibitor inputs which keeps the neuron in the inactive state, output signal of 0, regardless of the other inputs. These inputs allow the TLU to synthesize logic function such as AND, OR, XOR, NOR and NOT. Combining individuals TLU's together into a network allows the computation of any logical function or simulation of any finite automaton [65]. The functions of the network are defined by how the TLU's are interconnected.

Frank Rosesnblatt followed the TLU in the 1950s and 60s with a more general computational model, the Perceptron [66]. In this model, the inputs to the neuron are weighted and like the TLU, is activated when the sum of the weighted inputs reaches a threshold theta. The preceptors are arranged in a specific pattern, a fully connected bipartite graph, which is usually called a single-layer perceptron, The function of the network is defined by adapting the weights of the neurons with an algorithm.

In 1969, Marvin Minsky and Seymour Papert [67] showed that the single-layer perceptron was incapable of learning a whole class of problems. In particular they

showed that the XOR problem could not be solved due to the fact it is not linearly separable. This limitation was resolved by multi-layered perceptrons which include one or more hidden layers. These networks are also called feedforward Neural Networks because they are acyclic with the signals moving forward from the input layer through each hidden layer to the output layer.

The feedforward network is trained using the back-propagation algorithm which adjusts the weights of the neurons to minimize the error between the actual output of the network and the desired output [68]. The learning algorithm is called back-propagation due to the propagation of errors backwards from the output layer through the hidden layers. Adjusting the weights to minimize the error is an optimization problem with gradient descent being a popularly used method. A cost function E is used to assess the difference between the actual and desired, or target, output. Gradient decent uses the derivative of the cost function to determine how to adjust each weight in a particular layer. As such, a requirement is that the output of the artificial neurons be differentiable. The output of the neuron is controlled by an activation function. In the case of the TLU and perceptron, this is the step function. Since the step function is not differentiable, other activation functions such as the logistic function are used. The weights are adjusted in an iterative manner observing each input example and adjusting by the learning rate gamma

5.3 Convolutional Neural Networks

Although ANNs are theoretically able to learn complex non-linear mappings from large collections of samples, there are limitations when they are applied to image recognition tasks. In order to learn the complex patterns in images, an ANN requires a many hidden layers with many neurons per layer creating a very large number of parameters that must be learned [69]. The large number of parameters makes the ANN susceptible to overfitting. Models with fewer parameters to learn will typically

generalize better and require fewer training samples [70]. While overfitting can be mitigated by increasing the size of the dataset, if possible, it comes at an increased computational expense. A more impactful limitation of the ANN is the lack of built-in invariance with respect to translations or local distortions of the input. [71] To learn these translations and distortions, there would need to be multiple neurons with similar weight patterns located where the translations and distortions may occur in the image. Additionally, images have a two-dimensional local structure in that adjacent pixels are highly correlated. The local structure is completely ignored by an ANN.

Convolutional neural networks (ConvNets) are similar in structure to ANNs but consist of layers other than fully connected. Also inspired by animal biology: in this case the visual cortex of a cat which was known to consist of maps of local receptive fields that decrease in granularity [72]. Instead of being composed entirely of fully connected layers, ConvNets use the ideas of local receptive fields, shared weights and pooling in the majority of the layers. These architectural adjustments aim to overcome the limitations of the ANN stated previously. The structure of a typical ConvNet is shown in Figure 5.2.

In the convolutional layer, instead of connecting every input to every neuron, only small localized regions are connected. This region is called the local receptive field, and each connection has a learned weight accosted with it. The field is scanned across the entire image, with each location being connected to a new neuron in the next layer. This is implemented as a convolution filter and is what give the layer its name. There are multiple filters per layer, each with its own weights. Though a single filter uses the same weights across the entire image. These shared weights allow the filters of the firsts layer to learn basic features of the image such as edges, points or corners. Subsequent convolutional layers then learn more complex patters of these basic features. The convolutional layer allow a ConvNet to learn patterns without

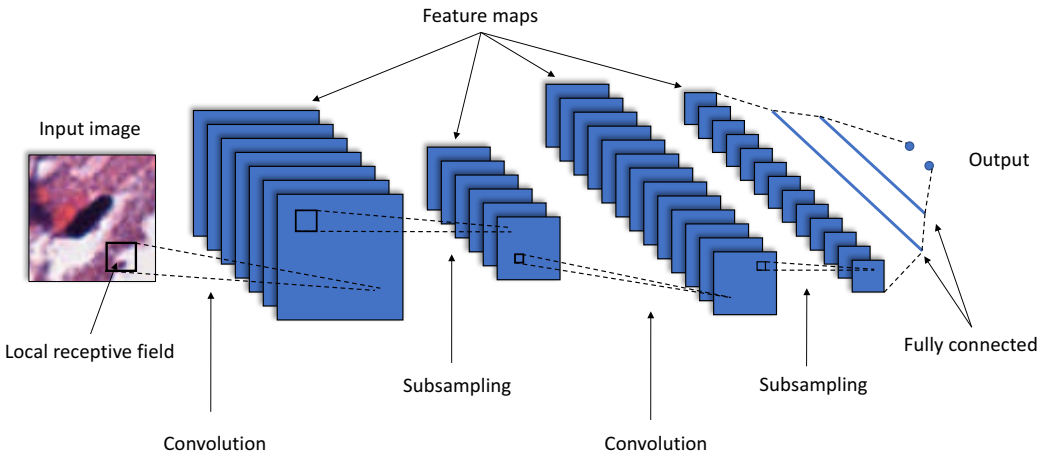

Figure 5.2: Typical convolutional neural network

Convolutional neural networks can be broken down into two parts: convolutional layers which extract the features from the image and the fully connected layers which learn the structures of the images. The convolutional layers form three-dimensional volumes with f two-dimensional feature maps, while the fully connected layers form one-dimensional linear arrays with each neuron connected to one neuron in the previous layer.

regard for the pattern's location in the image. The layers of a ConvNet can be though of as a three-dimensional volume with each filter in a convolutional layer creating a two-dimensional feature map in the next layer. Like the ANN, activation functions are use for the outputs of the convolutional layer. Instead of the sigmoid, or hyperbolic tangent functions, ConvNets typically use Rectified Linear Units (ReLU). The ReLU applies $max(0, x)$ element-wise to the output as the activation function, ReLUs were found to improve performance over the normal sigmoid activation function [73].

The pooling layer follows the convolutional layer and "summarize" the feature maps by sub-sampling. The pooling layer takes a small region of the feature map, typically 4 pixels square, and reduces it to a single pixel with a function. Typically max pooling is used, the maximum value of the region is used. Other functions such as average pooling or L2-norm pooling have been used though max pooling has shown the best results [74] [75].

ConvNets utilize one or more fully connected layers followed by logistic regression as the last few layers of the network. These layers take the features extracted by the previous layers and perform the final classification. As with ANNs, all weights are learned using back-propigation.

5.4 Deep Convolutional Neural Networks Augmentation

Deep neural networks for image recognition are of interest due to their ability to learn highly complex functions by increasing abstraction with each layer [63] [76]. ConvNets provide a significant advantage over ANNs in deep networks due to the reduction in the number of parameters to learn, the ability to learn features regardless of the feature's position and the ability to preserve spatial locality. These advantages are not all that can be utilized to improve the ability of deep networks to learn image classification tasks.

The convolutional and pooling layers provide a reduction in the number of parameters to learn as compared to a fully connect network of the same depth (number of layers). Still, for a deep network, lets say 10 or more layers, there will be a significant number of parameters to learn. As the number of parameters increases so does the number of training samples required to avoid over-fitting [70]. As we have discussed earlier in this work, the cost of obtaining labeled training samples can be significant. Since the raw data are being used instead of engineered features, we can increase the size of the training set with data augmentation [77]. Stated simply, data augmentation is applying label-preserving transformations to the existing labels images. These include, but are not limited to, vertical and horizontal reflection [78], rotation, cropping, lens distortion [79], translations and skewing [80]. For each augmented image, a random combination of transformations with random parameters are chosen. The augmentation factor, the number of augmented images added per original training image, of 10 or 20 can easily make a training set in the hundreds a training set in the thousands.

Data augmentation can also be used at prediction time. Instead of using the classification results for a single test image, the image can be augmented with each resultant image also classified. The results can then either be averaged using the softmax score or a majority vote can be used on the predicted class of each image [78].

While a deep network consists mostly of convolutional and pooling layers, there are still a few fully connected layers that are susceptible to overfitting due to the number of parameters. To mitigate the overfitting, a technique called dropout can be used [73]. Stated simply, dropout works by randomly dropping a fraction of the fully connected layer's neurons during training. For each training epoch, a set of neurons are selected to be ignored and the parameters are updated as normal for the remaining neurons. This process is repeated, each time selecting a new set of neurons, as training continues. The act of dropping random neurons each training

epoch simulates the averaging of predictions made by a large number of networks without the complexity of creating them [81]. Since the weights for the remaining neurons were learned with a fraction of the neurons missing, we multiple the neurons by the fraction of neurons that were dropped out.

Chapter 6 TissueNet - A Scalable Deep Learning Framework

6.1 Introduction

Deep learning methods have been applied to histologic image data for mitosis detection [82] [83] [84], identifying regions invasive ductal carcinoma [85] and region classification in glioblastomas and renal clear cell carcinomas [86]. While the works provide encouraging results, none addressed the critical issue of scalability, with each using datasets numbered in the hundreds of objects, nor provide the research community with readily available open-source tools. With an image dataset consisting of hundreds of slides, smaller histologic objects such as cell nuclei can easily number in the hundreds-of-millions, making tools that scale a necessity.

There are a few challenges when working with large whole-slide image datasets: (i) Commonly available deep learning tools take images as input for both training a model and using the model to predict. Any image manipulation such as resizing must be done as a preprocessing step, this include extracting regions of interest from whole-slide images which could number in the millions per slide. Some currently available tool will either read each image individually or require another preprocessing step to package the images together. (ii) Training set size — As we have shown earlier, obtaining labeled examples is a costly task. Due to the complexity of deep convolutional neural networks, more labeled examples are needed than a classifier such as a Random Forest or support vector machine. (iii) Volume of data — To efficiently process hundreds of slides, which can contain over three hundred million nuclei in the case of low grade glioma dataset, the system must be able to leverage multiple cores, GPUs and servers.

To address these challenges, we have developed TissueNet, a scalable deep learning software framework for the classification of histologic objects. TissueNet provides a complete end-to-end pipeline for training and predicting that includes preprocessing steps such as data augmentation, laplacian pyramids for multi-resolution networks and grayscale conversion. Its parallel architecture takes advantage of modern multi-core and multithreaded hardware and has support for multiple GPUs for prediction.

6.2 Architecture

6.2.1 Nuclear Detection and Image Normalization

While Deep learning methods eliminate the need for image segmentation, the histomic objects of interest first need to be extracted from the whole-slide image and cropped to a specific size. Instead of segmentation we use a nuclear detection pipeline, which locates the nuclei within the image and calculates the location of its centroid. The centroids are then used to guide the extraction tool described in the next section.

The nuclear detection pipeline utilizes algorithms provided by the HistomicsTK library for histologic image analysis. The pipeline follows the same general steps as the segmentation pipeline describe in section 3.2.3. Images are color normalized to a standard, then color deconvolution is used to create a binary mask of the location of pixels corresponding to cell nuclei. The mask is used by a constrained Laplacian of Gaussian (LoG) filter which produces a scale-selective peak response at the center of each object. The LoG filter response is then used, along with a binary mask, by a max clustering algorithm which identifies the local maximum for each pixel in the nuclear mask. From these local maximum the centroids of the nuclei are calculated. Unlike the segmentation pipeline, no further processing is needed. The advantage of this method is that it is able to discern multiple nuclei that are clustered tightly together.

6.2.2 Software

TissueNet is composed of a set of 4 tools: tn_dataset_convert, tn_slide_convert, tn_train and tn_predict. Internally, each of the tools are structured as a directed acyclic graph (DAG) with each node of the graph performing a single task such as reading, writing, training, augmentation, prediction, etc. This architecture provides two advantages. First, each node can execute in parallel, allowing the system to scale to massive datasets numbering in the hundreds-of-millions. As currently imple-

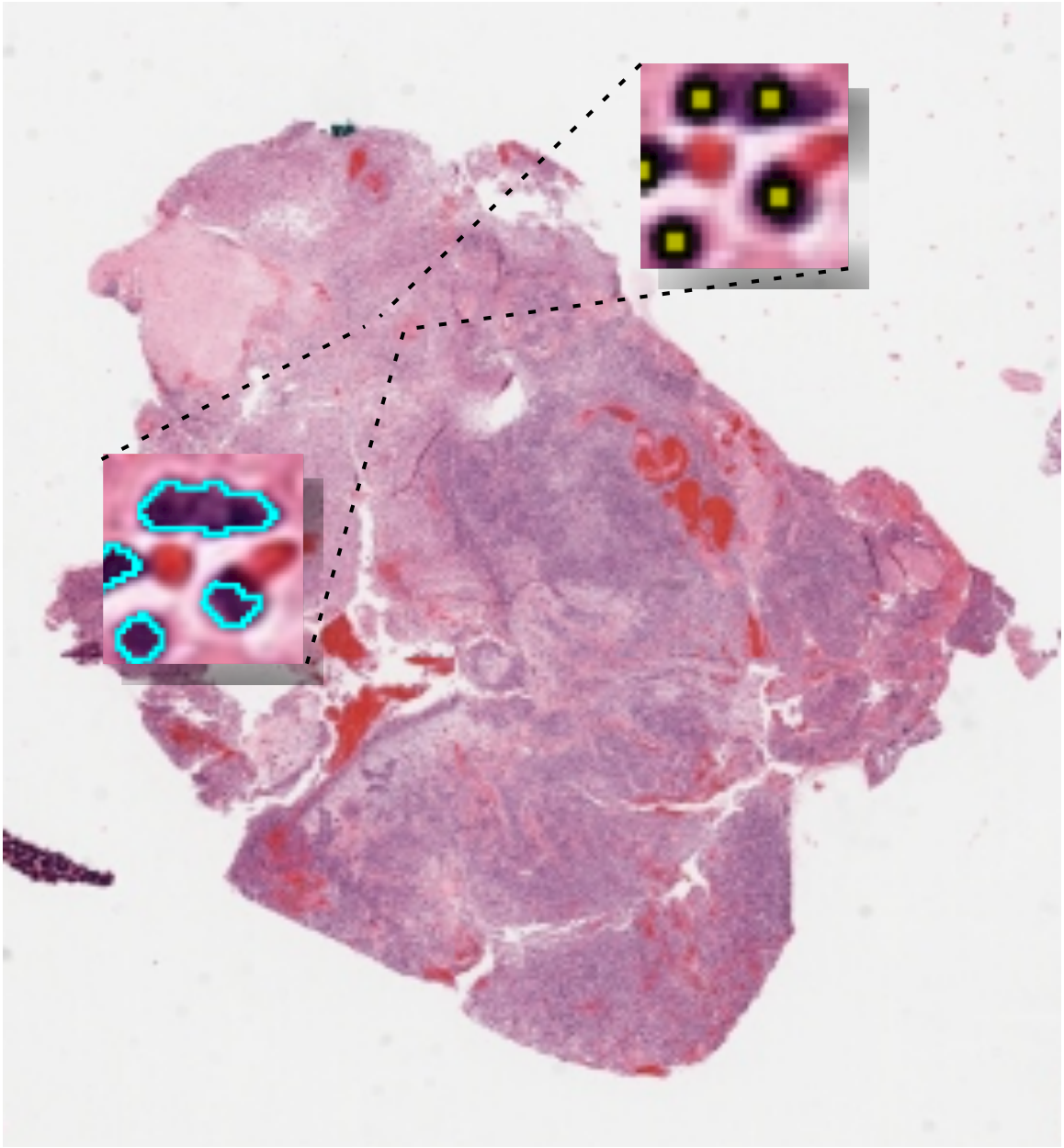

Figure 6.1: Nuclear segmentation

Even though the nuclei a packed closely together, the nuclear detection algorithm correctly detect 2 nuclei whereas the segmentation algorithm detect 1.

mented, each node runs in its own thread on the same server. However, this can be easily expanded to allow nodes to run on multiple servers for much greater scalability. Secondly, the DAG architecture allows processing tasks to be easily enabled, disabled and ordered as needed for conversion, training or prediction of a particular dataset. This includes having multiple nodes running the same task on different segments of data, providing both task and data parallelism.

The two tools, tn_dataset_convert and tn_slide_convert, are used in a one-time preprocessing step to prepare a dataset for use in TissueNet. The objects we are interested in analyzing on the whole-slide image need to be extracted from the image. The extraction process is quite time consuming as it needs to decompress the entire image and extract the regions of interest. Rather than repeat the extraction process every time an analysis is run, TissueNet stores the objects in a "packed" format in which each extracted region of interest is sequentially stored one after another in the file. This "packed" format allows the system to read in large chunks of data from the disk in a very efficient and fast manner. Preparing a single slide image for use in TissueNet is accomplished with the tn_convert_slide tool. The image file and a list of object centroids is provided as input, while the size of the extracted area defaults to 48 by 48 pixels but can be optionally set to another size. The tn_convert_dataset tool is used to convert training sets obtained from the HistomicsML system where the objects of interest each have a label and originate from multiple slides. Both tools save the data in an HDF5 [87] file format.

The tn_train and tn_predict tools are, as their names imply, used for training and predicting. These tools utilize the Caffe library [88] for their convolutional neural network implementations. To facilitate the DAG structure of our tools, we utilize the low-level API of Caffe and have implemented our own solver wrapper to support the online nature of the DAG, allowing the tools to scale beyond the available system and GPU memory. To define the network architecture, Caffe uses Google's protocol

buffers (protobuf). The protobuf files are plain text files which are parsed at run-time. This allows network flexibility without the need for code modification as long as the input and output of the network is compatible with TissueNet.

The training and prediction tools provide additional functionality that is either readily available through run-time parameters or can be added with simple code modification. Of these, data augmentation is the most prominent. The data augmentation functionality takes an input image and creates duplicate samples by applying a series of transforms to the image. The number and types of transforms are selected at random and include: rotation, horizontal and vertical flip, rotation, shear, scaling and translation. The number of new samples generated defaults to 10 for training and 3 for prediction and can be changed by a run-time parameter. For prediction, each sample, the original and the augmentation derived, are run on a separate GPU.

Other functionality available as run-time parameters: grayscale — converts the input image to grayscale; multi-resolution — uses two images of the same object at different magnification levels; H&E deconvolution — instead of RGB channels, separate the Hematoxylin and Eosin components for the input channels. Additionally, a Laplacian pyramid node is available for another type of multi-resolution technique and requires code modification to enable.

6.2.3 Networks

While TissueNet can use any network architecture, the dataset will dictate the specific architecture to use. For our data, we needed a model that works well classifying cell nuclei. Instead of trial and error, we compared a basic network architecture, LeNet [71], with VGGNet — a few very deep models from Karen Simonyan and Andrew Zisserman [89] and another popular architecture: AlexNet developed by Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton [78].

LeNet–5 is a convolutional network designed for handwritten and machine printed

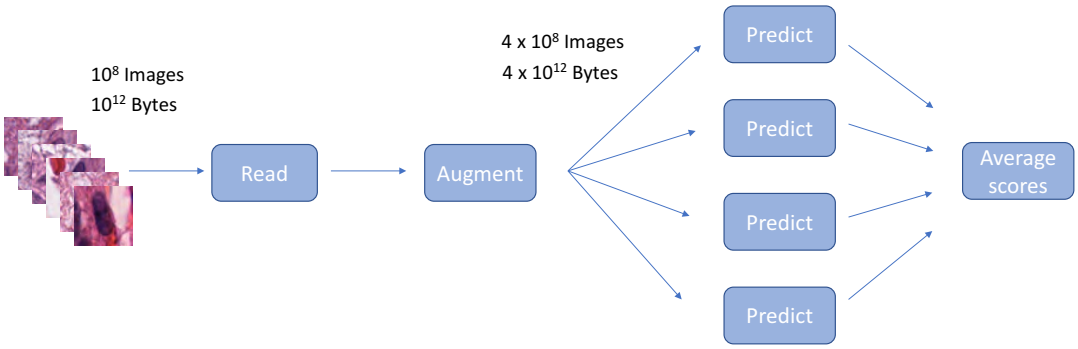

Figure 6.2: TissueNet prediction

TissueNet is capable of predicting hundreds of millions of nuclear objects. The system reads a slide's worth of 48 x 48 images from disk and sends them to the augmentation node. Using an augmentation factor of 3, four images (one original and three augmented) are sent to the four prediction nodes. Each of the processing nodes runs in its own thread ensuring that data will be ready for the prediction nodes as they complete processing the current batch of data. As the prediction nodes are predicting one batch of data, the read and augmentation nodes are preparing another.

character recognition. With 4 weight layers, It is composed of 2 convolutional layers each followed by a pooling layer. The convolutional and pooling layers are followed by two fully connected layers with a RelU activation as the output of the first and softmax for the output of the second. It is interesting to note that for this model, there are no RelU's on the output of the convolutional layers as we will see in the other models.

AlexNet was the winner of the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) http://www.image-net.org/challenges/LSVRC/. It is composed of 8 weight layers with 5 convolutional and 3 fully connected. The first 2 convolutional layers are followed by overlapping pooling layers with RelU activation and local response normalization (LRN). The LRN helps keep highly "excited" neurons from spilling over into neighboring neurons since RelU's have unbounded activations. The idea is preserve high-frequency components within the receptive fields. The next 3 convolutional layers are connected directly together with RelU's for activation without max pooling. The last convolutional layer is followed by a max pooling layer. The final 3 layers are fully connected with RelU activations and the final output utilizes softmax. Of note in this model is its use of dropout in the fully connected layers.

VGGNet is actually a set of architectures ranging from 11 to 19 weight layers. It was the second place finisher for the 2014 ILSVRC event. They are similar to AlexNet though have more weight layers. Distinguishing weight layers is important because it determines the number of parameters that need to be learned. As with AlexNet, VGGNet utilizes convolutional layers with RelU activation and max pooling. What differs is the lack of pooling between sets of the convolutional layers. While AlexNet uses this technique for its last 3 convolutional layers, VGGNet uses it throughout. The idea is that large receptive fields can be replaced by successive layers of 3x3 convolutions. This reduces the number of weights needed to learn. For instance, a 7x7 convolutional layer with F feature maps needs $49F^2$ weights. While three

successive 3x3 convolutional layers need only $27F^2$ weights, while covering the same 7x7 receptive field.

6.3 Results

We used the same 360 million cell nuclei from 781 images (464 tumors) from The Cancer Genome Atlas Lower Grade Glioma (LGG) project as for HistomicsML validation. The 135 sample training set for vascular endothelial cell nuclei (VECN) and the 2479 labels cell nuclei from 67 slides for the validation set were also used. See section 3.3 for a description. All test were run on a 12-core /24 thread server with 128 GigaBytes of RAM and 2 Nvidia K–80 GPUs.

To compare accuracy between network models, we trained each network using an augmentation factor from 10 to 60 in increments of 10. We then applied the network to the 2479 sample validation set using an augmentation factor of 4. The training and prediction were run 10 times and we took the mean of the runs. The results can be seen in Figure 6.3.

AlexNet with an augmentation factor of 60 performed the best in accuracy overall. LeNet was removed from the comparison due to its inability to converge while training at any augmentation factor. We reason that the inability to converge is due to the nature of the nuclei images. Rather than detecting objects, we believe the networks must behave more like texture detectors. Since LeNet was developed for handwriting and machine printed character detection, which has very sharp edges, it was most likely unable to discern any patterns in the nuclei images. AlexNet also outperformed in training time relative to the other networks at the same augmentation factor. Intuitively, as the augmentation factor and the number of weight levels in the network increased the execution time for prediction also increased. Surprisingly, the VGGNet models decreased in accuracy as the augmentation factor increased. Since they all share the same 3 x 3 layered filter approach for the first layer as oppose to AlexNet's

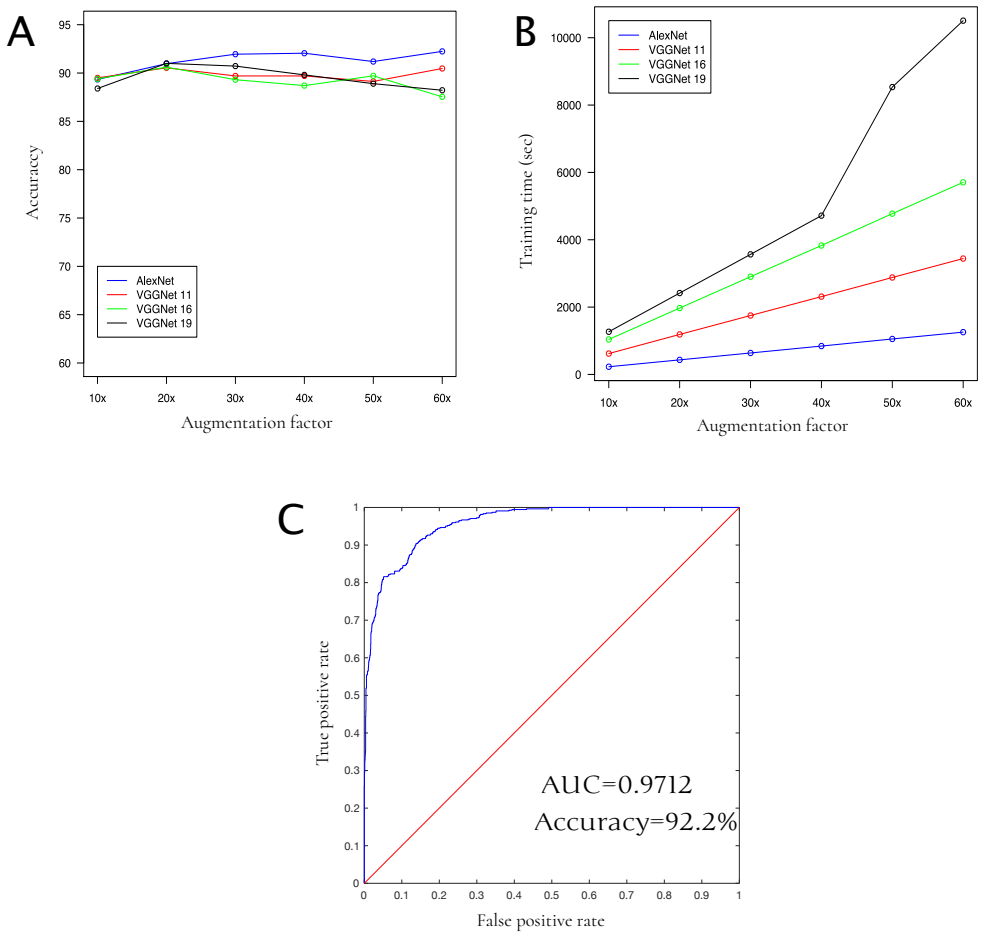

Figure 6.3: TissueNet VECN classification results

We trained 6 networks on our 2479 VECN validation set. LeNet was unable to converge while training so it is excluded from these graphs. (A) Each model performed similarly for augmentation factors of 10 and 20. As the augmentation factor increased the VGGNet models decreased in accuracy. (B) Training time was linear for all but the 19 weight layer VGGNet. (C) AlexNet with an augmentation factor of 60, achieved an accuracy of 92.2% and an AUC of 0.9712.

11 x 11, we believe the larger filter in AlexNet is better suited for our application. This may be due to the reduction in weights for the initial layer. While the 2 layers 3 x 3 convolution filters effectively act as a 5 x 5 filter, for the nuclei images a larger convolution may be needed. Again, this is may be due to our application being more of texture detection rather than object. Training time wast linear for most models, the only exception is the 19 layer VGGNet model. We are unsure of the nature of the slowdown, though overheating seems unlikely as the 16 layer VGGNet ran for a longer period of time. This is still an open question.

Comparing the results to the random forest classifier, we see an improvement of 87.4% to 92.2% and 0.9643 to 0.9712 for AUC. Though we used an augmentation factor of 60 to achieve the increase in accuracy, we still only need the 135 labeled samples. This is another benefit of TissueNet, the image augmentation capability reduces the number of samples needed.

To demonstrate TissueNet's ability to process large datasets, we classified 3,319,414, three-channel color mages, at 48 x 48 pixels, of cell nuclei with the VGG–16 network. Total run time was 17,840.64 seconds (4 hours 57 minutes) to read, augment and classify all images. This is a processing rate of 5.4 milliseconds per image or 5946.88 seconds (1 hour 39 minutes) per slide of 1.1 million nuclei. For comparison, our initial proof of concept for classifying cell nuclei utilized an existing deep learning toolkit and took over 6 hours to process a single slide.

Chapter 7 Discussion and Future Work

7.1 Discussion

HistomicsML enables biomedical investigators to extract phenotypic information from large whole-slide imaging datasets by addressing the unique challenges presented by the scale and nature of this data. HistomicsML is open-source, and is available as a software container for convenient deployment.

We trained a highly accurate endothelial cell classifier (AUC=0.9643) by labeling only 135 cell nuclei using active learning. The visualization and learning capabilities of HistomicsML simplifies the training process for experts by enabling them to rapidly label objects for training and to re-train and review classification rules in seconds. The web-based interface provides access to terabytes of image data, and new technology enables fluid and seamless display of image analysis boundaries and class predictions associated with hundreds of millions of histologic objects. Active learning methods direct label to objects that provide the most training benefit, improving training efficiency by minimizing the labeling of redundant objects. Scalable implementations of both machine-learning algorithms and visualization software enable this system to operate efficiently on very large datasets.

Using our endothelial classifier, we accurately predicted survival in glioma patients with validated measurements of microvascular phenotypes. We identified significant associations between microvascular phenotypes, grade, and recently defined molecular subtypes of gliomas. These investigations are timely in the current era of precision medicine, in which prognostic biomarkers have not been established within newly emergent molecular subtypes of diseases such as the diffuse gliomas. While it has long been established that angiogenesis is related to disease progression in gliomas,

we showed that HistomicsML can be used to measure subtle changes in microvascular phenotypes that perform as well as grade in predicting survival. Integration with genomic data identified recognized molecular pathways associated with angiogenesis, and more interesting subtype-specific pathways that are enriched with phenotype-correlated genes.

Our analysis of gliomas is a template for how HistomicsML can be used to link histology, clinical and genomic data to explore the prognostic and molecular associations of histologic phenotypes in other diseases. Images of histology contain important information that can be difficult or impossible to ascertain through genomic assays. Recent developments in the deconvolution of gene expression data can accurately estimate the abundance or fractional proportions of cell types in a sample, but these approaches cannot provide spatial or morphologic information that often contains considerable prognostic or scientific value. While molecular phenomena drive phenotypes, quantitative histologic analysis provides more immediate readouts of information that are difficult or impossible to obtain from genomic profiling. Additionally, we showed how active learning improved prognostication and agreement between histologic and molecular markers for VECNs.

TissueNet is was shown to improve classification accuracy over Random Forests using the same training set. Utilizing the AlexNet architecture with an augmentation factor of 60, we were able to obtain an accuracy of 92.24% on our validation set compared to 87.4% achieved by the Random Forest. The deep learning methodology also assists in the idea of reducing cost by reducing the number of labeled samples needed in the training set as image augmentation was shown to provide an increase in performance in AlexNet. As with HistomicsML, TissueNet is open-source and freely available.

7.2 Future work

Future development of HistomicsML will focus on enabling the analysis of region/patch-based data and improving scalability for larger datasets. The analysis of image patches characterized by texture analysis or autoencoders can address the classification of complex multicellular structures while avoiding the need for explicit segmentation and is a simple extension of HistomicsML. The memory footprint of feature data is currently a limitation that prevents HistomicsML from scaling to larger datasets. We plan to improve memory management, and to utilize commodity graphics processors to enable HistomicsML to run on workstation class systems.

Currently, HistomicsML is limited to binary classifiers so support for multi-class is forthcoming. While multi-class sup[port is a trivial change for the learning server, the challenge is in the user interface. Double clicking in the selection page may work for a few labels but will get tedious for more than five. How to cleanly incorporate selecting multiple labels with limited screen real estate is a human-computer interface problem.

Region classification is another anticipated addition to HistomicsML. Instead of single cell nuclei, we look to classify regional structures within the whole-slide image such as necrosis, angiogenesis and immune response. While a naive implementation of dividing the whole-slide image into bigger sections and keeping the system as-is may work to some extent, these regions are not naturally rectangular and vary widely in size. Challenges arise for both the user interface and for learning algorithms. Deep learning is an excellent choice for this type of classification but the slow training time makes them unfeasible.

TissueNet is in its nascency, with many avenues of research open. Of particular concern is training time, with the fastest network taking over seven minutes to train 1350 samples of 48 x 48 pixel images. The ability to perform classification with significantly larger images may be handicapped by such long training times. This is

of importance in the analysis of image patches for multicellular structures, where the sizes are around 1024 x 1024 pixels. One promising solution is using multiple GPU's for training.

Another interesting area of exploration is in image segmentation. There has been some work in the literature that utilizes deep learning to segment objects within an image using fully convolutional networks. The challenge for whole-slide pathology images , besides their sheer size, is the lack of "high-frequency" boundaries. That is, the objects we are interested in typically do not have well-defined edges as say a car on a street would. In fact, if looking for historic structures as, pseudopalisading necrosis, there is no definitive boundary delineating the structure. For whole-slide images, the image segmentation needs to be more of a texture segmentation.

Multi-Resolution functionality support is area that may be useful for classification of object types that tend to "clump" together. Utilizing a high and low magnification image together, the network can use the local structure in the low-res image to improve classification. The ability to take a low-res image and extract a high-res section from the center already exists in TissueNet, but showed a decline in accuracy. Using multi-resolution tiff images may be of use.

Scaling TissueNet further is another area ripe for enhancement. Currently, TissueNet runs on a single server, with each node of the DAG running in its own thread. Scaling to multiple servers, utilizing network communication between the nodes, will allow a single node to use multiple threads without degrading the other nodes in the DAG. This will be of more significance for classification of large image patches, especially in the augmentation nodes.

Bibliography

[1] L. A. Cooper, A. B. Carter, A. B. Farris, F. Wang, J. Kong, D. A. Gutman, P. Widener, T. C. Pan, S. R. Cholleti, A. Sharma, *et al.*, "Digital pathology: Data-intensive frontier in medical imaging," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 991–1003, 2012.

[2] F. Schuh, J. V. Biazús, E. Resetkova, C. Z. Benfica, A. F. Ventura, D. Uchoa, M. Graudenz, and M. I. Edelweiss, "Histopathological grading of breast ductal carcinoma in situ: Validation of a web-based survey through intra-observer reproducibility analysis," *Diagnostic pathology*, vol. 10, no. 1, p. 93, 2015.

[3] T. J. Fuchs and J. M. Buhmann, "Computational pathology: Challenges and promises for tissue analysis," *Computerized Medical Imaging and Graphics*, vol. 35, no. 7, pp. 515–530, 2011.

[4] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *ArXiv preprint arXiv:1606.05718*, 2016.

[5] A. H. Beck, A. R. Sangoi, S. Leung, R. J. Marinelli, T. O. Nielsen, M. J. van de Vijver, R. B. West, M. van de Rijn, and D. Koller, "Systematic analysis of breast cancer morphology uncovers stromal features associated with survival," *Science translational medicine*, vol. 3, no. 108, 108ra113–108ra113, 2011.

[6] J. Kong, L. A. Cooper, F. Wang, J. Gao, G. Teodoro, L. Scarpace, T. Mikkelsen, M. J. Schniederjan, C. S. Moreno, J. H. Saltz, *et al.*, "Machine-based morphologic analysis of glioblastoma using whole-slide pathology images uncovers clinically relevant molecular correlates," *PloS one*, vol. 8, no. 11, e81049, 2013.

[7] H. Chang, J. Han, A. Borowsky, L. Loss, J. W. Gray, P. T. Spellman, and B. Parvin, "Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association," *IEEE transactions on medical imaging*, vol. 32, no. 4, pp. 670–682, 2013.

[8] A. Madabhushi, S. Agner, A. Basavanhally, S. Doyle, and G. Lee, "Computer-aided prognosis: Predicting patient and disease outcome via quantitative fusion of multi-scale, multi-modal data," *Computerized medical imaging and graphics*, vol. 35, no. 7, pp. 506–514, 2011.

[9] O. Sertel, J. Kong, H. Shimada, U. Catalyurek, J. H. Saltz, and M. N. Gurcan, "Computer-aided prognosis of neuroblastoma on whole-slide images: Classification of stromal development," *Pattern recognition*, vol. 42, no. 6, pp. 1093–1103, 2009.

[10] J. Kong, O. Sertel, K. L. Boyer, J. H. Saltz, M. N. Gurcan, and H. Shimada, "Computer-assisted grading of neuroblastic differentiation," *Archives of pathology & laboratory medicine*, vol. 132, no. 6, pp. 903–904, 2008.

[11] S. Naik, S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, IEEE, 2008, pp. 284–287.

[12] G. Litjens, C. I. Sánchez, N. Timofeeva, M. Hermsen, I. Nagtegaal, I. Kovacs, C. Hulsbergen-Van De Kaa, P. Bult, B. Van Ginneken, and J. Van Der Laak, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," *Scientific reports*, vol. 6, 2016.

[13] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, "Patch-based convolutional neural network for whole slide tissue image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2424–2433.

[14] S. Kothari, J. H. Phan, A. N. Young, and M. D. Wang, "Histological image classification using biologically interpretable shape-based features," *BMC medical imaging*, vol. 13, no. 1, p. 9, 2013.

[15] Y. Yuan, H. Failmezger, O. M. Rueda, H. R. Ali, S. Gräf, S.-F. Chin, R. F. Schwarz, C. Curtis, M. J. Dunning, H. Bardwell, *et al.*, "Quantitative image analysis of cellular heterogeneity in breast tumors complements genomic profiling," *Science translational medicine*, vol. 4, no. 157, 157ra143–157ra143, 2012.

[16] W. C. Rutledge, J. Kong, J. Gao, D. A. Gutman, L. A. Cooper, C. Appin, Y. Park, L. Scarpace, T. Mikkelsen, M. L. Cohen, *et al.*, "Tumor-infiltrating lymphocytes in glioblastoma are associated with specific genomic alterations and related to transcriptional class," *Clinical Cancer Research*, vol. 19, no. 18, pp. 4951–4960, 2013.

[17] L. A. Cooper, D. A. Gutman, C. Chisolm, C. Appin, J. Kong, Y. Rong, T. Kurc, E. G. Van Meir, J. H. Saltz, C. S. Moreno, *et al.*, "The tumor microenvironment strongly impacts master transcriptional regulators and gene expression class of glioblastoma," *The American journal of pathology*, vol. 180, no. 5, pp. 2108–2119, 2012.

[18] M. Nalisnik, D. A. Gutman, J. Kong, and L. A. Cooper, "An interactive learning framework for scalable classification of pathology images," in *Big Data (Big Data), 2015 IEEE International Conference on*, IEEE, 2015, pp. 928–935.

[19] L. A. Cooper, J. Kong, D. A. Gutman, W. D. Dunn, M. Nalisnik, and D. J. Brat, "Novel genotype-phenotype associations in human cancers enabled by advanced molecular platforms and computational analysis of whole slide images," *Laboratory investigation*, vol. 95, no. 4, pp. 366–376, 2015.

[20] M. Nalisnik, M. Amgad, S. Lee, J. V. Vega, D. J. Brat, D. A. Gutman, and L. A. D. Cooper, "Rapid interactive phenotyping in large histomic datasets with active learning," *Submitted*, 2017.

[21] M. Kubat, S. Matwin, *et al.*, "Addressing the curse of imbalanced training sets: One-sided selection," in *ICML*, vol. 97, 1997, pp. 179–186.

[22] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[23] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[24] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 2372–2379.

[25] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[26] D. Angluin, "Queries and concept learning," *Machine learning*, vol. 2, no. 4, pp. 319–342, 1988.

[27] L. E. Atlas, D. A. Cohn, and R. E. Ladner, "Training connectionist networks with queries and selective sampling," in *Advances in neural information processing systems*, 1990, pp. 566–573.

[28] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, Springer-Verlag New York, Inc., 1994, pp. 3–12.

[29] F. Laws and H. Schätze, "Stopping criteria for active learning of named entity recognition," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, 2008, pp. 465–472.

[30] M. Bloodgood and K. Vijay-Shanker, "A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2009, pp. 39–47.

[31] F. Olsson and K. Tomanek, "An intrinsic stopping criterion for committee-based active learning," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2009, pp. 138–146.

[32] S. Dasgupta, "Two faces of active learning," *Theoretical computer science*, vol. 412, no. 19, pp. 1767–1781, 2011.

[33] N. Kutsuna, T. Higaki, S. Matsunaga, T. Otsuki, M. Yamaguchi, H. Fujii, and S. Hasezawa, "Active learning framework with iterative clustering for bioimage classification," *Nature communications*, vol. 3, p. 1032, 2012.

[34] Z. Wang, C. D. Monteiro, K. M. Jagodnik, N. F. Fernandez, G. W. Gundersen, A. D. Rouillard, S. L. Jenkins, A. S. Feldmann, K. S. Hu, M. G. McDermott, *et al.*, "Extraction and analysis of signatures from the gene expression omnibus by the crowd," *Nature communications*, vol. 7, 2016.

[35] R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver, "Functional genomic hypothesis generation and experimentation by a robot scientist," *Nature*, vol. 427, no. 6971, pp. 247–252, 2004.

[36] B. Zhang, Y. Wang, and F. Chen, "Multilabel image classification via high-order label correlation driven active learning," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1430–1441, 2014.

[37] D. H. Nguyen and J. D. Patrick, "Supervised machine learning and active learning in classification of radiology reports," *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 893–901, 2014.

[38] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[39] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[40] K. Martinez and J. Cupitt, "Vips-a highly tuned image processing software architecture," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, IEEE, vol. 2, 2005, pp. II–574.

[41] A. Goode, B. Gilbert, J. Harkes, D. Jukic, M. Satyanarayanan, *et al.*, "Openslide: A vendor-neutral software foundation for digital pathology," *Journal of pathology informatics*, vol. 4, no. 1, p. 27, 2013.

[42] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 502–516, 1989.

[43] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979.

[44] B. W. Silverman, "Some aspects of the spline smoothing approach to non-parametric regression curve fitting," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–52, 1985.

[45] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "A polygonal line algorithm for constructing principal curves," in *NIPS*, 1998, pp. 501–507.

[46] J. J. Verbeek, N. Vlassis, and B. Kröse, "A k-segments algorithm for finding principal curves," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 1009–1017, 2002.

[47] R. Tibshirani, "Principal curves revisited," *Statistics and computing*, vol. 2, no. 4, pp. 183–190, 1992.

[48] B. D. Ripley, "Modelling spatial patterns," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 172–212, 1977.

[49] e. a. Cancer Genome Atlas Research Network D.J. Brat, "Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas," *N Engl J Med*, vol. 2015, no. 372, pp. 2481–2498, 2015.

[50] D. E. Reuss, Y. Mamatjan, D. Schrimpf, D. Capper, V. Hovestadt, A. Kratz, F. Sahm, C. Koelsche, A. Korshunov, A. Olar, *et al.*, "Idh mutant diffuse and anaplastic astrocytomas have similar age at presentation and little difference in survival: A grading problem for who," *Acta neuropathologica*, vol. 129, no. 6, pp. 867–873, 2015.

[51] F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati, "Evaluating the yield of medical tests," *Jama*, vol. 247, no. 18, pp. 2543–2546, 1982.

[52] R. K. Jain, E. Di Tomaso, D. G. Duda, J. S. Loeffler, A. G. Sorensen, and T. T. Batchelor, "Angiogenesis in brain tumours," *Nature Reviews Neuroscience*, vol. 8, no. 8, pp. 610–622, 2007.

[53] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, *et al.*, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005.

[54] J. Wang, J. Wang, J. Dai, Y. Jung, C.-L. Wei, Y. Wang, A. M. Havens, P. J. Hogg, E. T. Keller, K. J. Pienta, *et al.*, "A glycolytic mechanism regulating an angiogenic switch in prostate cancer," *Cancer research*, vol. 67, no. 1, pp. 149–159, 2007.

[55] A. Murat, E. Migliavacca, S. F. Hussain, A. B. Heimberger, I. Desbaillets, M.-F. Hamou, C. Rüegg, R. Stupp, M. Delorenzi, and M. E. Hegi, "Modulation of angiogenic and inflammatory response in glioblastoma by hypoxia," *PloS one*, vol. 4, no. 6, e5947, 2009.

[56] M. Felcht, R. Luck, A. Schering, P. Seidel, K. Srivastava, J. Hu, A. Bartol, Y. Kienast, C. Vettel, E. K. Loos, *et al.*, "Angiopoietin-2 differentially regulates angiogenesis through tie2 and integrin signaling," *The Journal of clinical investigation*, vol. 122, no. 6, pp. 1991–2005, 2012.

[57] J. Dufraine, Y. Funahashi, and J. Kitajewski, "Notch signaling regulates tumor angiogenesis by diverse mechanisms," *Oncogene*, vol. 27, no. 38, pp. 5132–5137, 2008.

[58] N. M. Kofler, C. J. Shawber, T. Kangsamaksin, H. O. Reed, J. Galatioto, and J. Kitajewski, "Notch signaling in developmental and tumor angiogenesis," *Genes & cancer*, vol. 2, no. 12, pp. 1106–1116, 2011.

[59] W. Chen, T. Tang, J. Eastham-Anderson, D. Dunlap, B. Alicke, M. Nannini, S. Gould, R. Yauch, Z. Modrusan, K. J. DuPree, *et al.*, "Canonical hedgehog signaling augments tumor angiogenesis by induction of vegf-a in stromal perivascular cells," *Proceedings of the National Academy of Sciences*, vol. 108, no. 23, pp. 9589–9594, 2011.

[60] M.-J. Goumans, Z. Liu, and P. Ten Dijke, "Tgf-beta signaling in vascular biology and dysfunction," *Cell research*, vol. 19, no. 1, pp. 116–127, 2009.

[61] E. Dejana, "The role of wnt signaling in physiological and pathological angiogenesis," *Circulation research*, vol. 107, no. 8, pp. 943–952, 2010.

[62] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.

[63] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[64] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[65] R. Rojas, *Neural networks: A systematic introduction*. Springer Science & Business Media, 2013.

[66] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[67] M. Minsky and S. Papert, "Perceptrons.," 1969.

[68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[69] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[70] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 6.

[71] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[72] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[73] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 8609–8613.

[74] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 111–118.

[75] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 2559–2566.

[76] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[77] K. Sohn and H. Lee, "Learning invariant representations with local transformations," *ArXiv preprint arXiv:1206.6418*, 2012.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[79] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *ArXiv preprint arXiv:1501.02876*, vol. 7, no. 8, 2015.

[80] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis.," in *ICDAR*, vol. 3, 2003, pp. 958–962.

[81] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *ArXiv preprint arXiv:1207.0580*, 2012.

[82] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 2013, pp. 411–418.

[83] H. Wang, A. Cruz-Roa, A. Basavanhally, H. Gilmore, N. Shih, M. Feldman, J. Tomaszewski, F. Gonzalez, and A. Madabhushi, "Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features," *Journal of Medical Imaging*, vol. 1, no. 3, pp. 034 003–034 003, 2014.

[84] C. D. Malon, E. Cosatto, *et al.*, "Classification of mitotic figures with convolutional neural networks and seeded blob features," *Journal of pathology informatics*, vol. 4, no. 1, p. 9, 2013.

[85] A. Cruz-Roa, A. Basavanhally, F. González, H. Gilmore, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, and A. Madabhushi, "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks," in *SPIE medical imaging*, International Society for Optics and Photonics, 2014, pp. 904 103–904 103.

[86] Y. Zhou, H. Chang, K. Barner, P. Spellman, and B. Parvin, "Classification of histology sections via multispectral convolutional sparse coding," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014, pp. 3081–3088.

[87] The HDF Group, *Hierarchical data format, version 5*, 1997-2017.

[88] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, pp. 675–678.

[89] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv preprint arXiv:1409.1556*, 2014.

[90] Google, *Protocol buffers*.

[91] Y. Tang and A.-r. Mohamed, "Multiresolution deep belief networks.," in *AISTATS*, 2012, pp. 1203–1211.

[92] M. Jordan and T. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[93] M. H. Kryder and C. S. Kim, "After hard drives—what comes next?" *IEEE Transactions on Magnetics*, vol. 45, no. 10, pp. 3406–3413, 2009.

[94] I. G. Goldberg, C. Allan, J.-M. Burel, D. Creager, A. Falconi, H. Hochheiser, J. Johnston, J. Mellen, P. K. Sorger, and J. R. Swedlow, "The open microscopy environment (ome) data model and xml file: Open tools for informatics and quantitative analysis in biological imaging," *Genome biology*, vol. 6, no. 5, p. 1, 2005.

[95] R. K. Padmanabhan, V. H. Somasundar, S. D. Griffith, J. Zhu, D. Samoyedny, K. S. Tan, J. Hu, X. Liao, L. Carin, S. S. Yoon, *et al.*, "An active learning approach for rapid characterization of endothelial cells in human tumors," *PloS one*, vol. 9, no. 3, e90495, 2014.

[96] R. Marée, L. Rollus, B. Stévens, R. Hoyoux, G. Louppe, R. Vandaele, J.-M. Begon, P. Kainz, P. Geurts, and L. Wehenkel, "Collaborative analysis of multi-gigapixel imaging data using cytomine," *Bioinformatics*, vol. 32, no. 9, pp. 1395–1401, 2016.

[97] C. S. Bjornsson, G. Lin, Y. Al-Kofahi, A. Narayanaswamy, K. L. Smith, W. Shain, and B. Roysam, "Associative image analysis: A method for automated quantification of 3d multi-parameter images of brain tissue," *Journal of neuroscience methods*, vol. 170, no. 1, pp. 165–178, 2008.

[98] K. Kvilekval, D. Fedorov, B. Obara, A. Singh, and B. Manjunath, "Bisque: A platform for bioimage analysis and management," *Bioinformatics*, vol. 26, no. 4, pp. 544–552, 2010.

[99] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *Journal of Machine Learning Research*, vol. 10, no. Jan, pp. 1–40, 2009.

[100] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *Aistats*, vol. 9, 2010, pp. 249–256.

[101] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[102] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[103] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[104] C. Sommer and D. W. Gerlich, "Machine learning in cell biology–teaching computers to recognize phenotypes," *J Cell Sci*, vol. 126, no. 24, pp. 5529–5539, 2013.

[105] T. Fuchs, P. Wild, H. Moch, and J. Buhmann, "Computational pathology analysis of tissue microarrays predicts survival of renal clear cell carcinoma patients," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pp. 1–8, 2008.