Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Bachelor's Degree in Informatics Engineering
Computing Engineering

Bachelor's Thesis

Trainable Superpixel Segmentation

Author
Josu Salinas Colina

2018

Universidad del País Vasco / Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Bachelor's Degree in Informatics Engineering
Computing Engineering

Bachelor's Thesis

Trainable Superpixel Segmentation

Author

Josu Salinas Colina

Directors
Basilio Sierra Araujo, Ignacio Arganda-Carreras

Abstract

Trainable Superpixel Segmentation is a plug-in developed for the ImageJ platform that aims at providing its users with the ability to train models to segment images by classifying superpixels using region-based image features. This project provides an underlying library that can be used independently, a graphic interface for ease of use and an evaluation protocol of the efficacy of the library. The evaluation of the developed library was conducted through a ten-fold cross-validation and the results were compared with those of the Trainable Weka Segmentation library. This document reports the planning, background research and development of the project. Finally, the development of the project and the results obtained are discussed and further research is proposed.

Contents

List of figures

List of tables

CHAPTER 1

Introduction

The technological advances in computing of the last decades have brought many innovations to different fields of science, through sheer computing power to the building of complex models the way of working in sciences have been thoroughly revolutionized. Within the many innovations that the evolution in computing power has brought image analysis tools have been in many different fields, ranging from microscopical image analysis to astronomical image analysis. As we will later discuss, images can be interpreted as collections of pixels, therefore collections of data, and, as such, they can be treated the same way that other datasets are treated, opening the possibility of applying Machine Learning techniques to image analysis. This project deals with a combination of image analysis and Machine Learning with the aim of providing a tool for image segmentation that can be used by scientists of different fields without the need for expertise in neither image analysis nor Machine Learning.

Digital images are conjunctions of intensity measurements known as pixels, this measurements are used usually for displaying in screens, but offer the opportunity to mathematically process the images and generate datasets from them. This has been taken advantage of to develop a wide variety of image processing techniques, from exposure correction techniques to advanced reconstruction techniques that allow for the digital reconstruction of old and damaged pictures. Furthermore, this has led to a Machine Learning approach to image analysis and processing.

Machine Learning techniques have been applied to image analysis and processing with different aims such as face detection, edge detection or automatic text processing, but this project focuses mainly in image feature extraction and image segmentation. As mentioned before, digital images are conjunctions of intensity measurements and therefore these measures can be treated as mathematical sets from which statistical values can be drawn. As a result, different features can be extracted for each pixel or pixel region of an image. Using these features, datasets can be generated where each pixel is represented by a set or vector of features, and as such traditional Machine Learning dataset classification and clustering techniques can be applied, generating new datasets of classified or clustered pixels and their corresponding result images.

This project has been developed using ImageJ [24] as a basis for the processing of images; this open source platform offers many different image processing capabilities and is host to many different libraries and plug-ins that offer support for many different tools. For the learning process, the WEKA library [30] has been used, a Machine Learning library that offers easy to use state-of-the-art implementations of the most popular classification and clustering technologies.

The main library that has been developed offers the ability of classifying images by using corresponding superpixel images as a basis for feature extraction and classification. This enables the creation of a more reduced dataset in contrast to per-pixel based processing libraries.

Together with this library a graphical interface has been developed. It offers the same capabilities of the library in an friendlier way of use. The GUI can be used for easier prototyping and testing and offers a simpler approach for those not familiar with coding.

Finally, an evaluation has been conducted to compare the library that has been developed to another library with similar capabilities. This evaluation has been conducted through a scripting process that enabled an automatic handling of a dataset and the collection of meaningful statistics. These statistics show that the library offers competitive results in comparison to other libraries with similar objectives.

This document describes the different logistics of the developed project, the knowledge basis on which the project is based, the specifics of the project that has been developed and the conclusions that have been drawn from this project, finalizing with a chapter dedicated to further research that could be developed after this project. Additionally appendix A offers a user's guide to the project and appendix B displays the results that were generated by the evaluation process.

CHAPTER 2

Document of Project Aims

This chapter deals with the aims established for the project, the creation of stages into which the project is divided, the specific tasks related to each phase, the expected calendar and the different predicted risks and contingency plans. Figure 2.2 summarizes the plan for the project.

2.1 Project reach

The main aim of this project is to develop a tool that provides supervised image classification through the use of superpixels. To do this a library will be developed together with a graphic interface for ease of use. Finally, an evaluation of the developed library will be conducted to evaluate the effectiveness of the developed library against other related publicly available tools.

In order to properly conduct this project, the tasks to be done will be divided into different sections that can be more easily managed. As the different facets of the project build upon each other, it is logical to expect that the project will have to be developed linearly and it is not expected that different stages will be undertaken before others, however some stages like those regarding reporting or background research may be developed simultaneously, and due to the dynamic nature of software development some changes may have to be made before developed artifacts to solve issues or lack of features that have been uncovered in further phases of the development of the project.

2.2 Project stages

The following tasks have been identified at the planning process of this project:

1. Planning:

   Within this first part of the project the aims of the project will be defined and the specific tools to be used within the project will be defined. Additionally, communication basis will be established with the project supervisors.

2. Background research

   This project has three core theoretical concepts that need to be researched in order to successfully carry out the developing of the artifact and its evaluation: image processing, image segmentation and image segmentation method comparison. This part of the project will focus on identifying state-of-the-art libraries and techniques, and deciding what tools to use in this development and what other projects this project will be compared with.

3. Library development

   The start of the practical side of the project will be within this task, where a base library will be developed that will provide supervised superpixel classification through the use of region-based image features.

4. GUI development

   After the library has been implemented a GUI will be developed that will offer a more accessible use of the aforementioned library. The interface will include all the features offered by the library and provide an easier way of experimenting with it.

5. Evaluation

   Once the library and the GUI have been developed, an evaluation will be carried out where the library will be compared to other libraries and tools that offer image segmentation capabilities. This evaluation will be performed through the use of an automated script for an easier execution of different variables.

6. Reporting

   All of the developments that will be carried out will be reported to the project supervisors, and will later be redacted in this document.

2.3 Project tasks

Each of the aforementioned phases can be further divided into specific tasks, this division is intended to help in the planning and development of the project. Each task is described together with a tentative hour planning, taking into account the predicted 300 hour workload for the duration of the project.

2.3.1 Planning

1. **Project objective definition:** define specific tasks to be undertaken during project. 3 hours.

2. **Project reach definition:** define objectives to be reached and objectives that are out of reach for the scope of the project. 1 hour.

3. **Definition of communications with project supervisors:** define communication ways with project supervisors to ensure correct communications of project progress. 1 hour.

2.3.2 Background research

1. **Development framework:** choose the framework where the project will be developed on. 2.5 hours.

2. **Image processing library:** choose an image processing library for image input, processing and output. 2.5 hours.

3. **Feature extraction library:** choose a library for input image feature extraction. 2.5 hours.

4. **Machine Learning library:** choose a library for Machine Learning. 2.5 hours.

5. **Evaluation method research:** choose specific metrics for evaluation and projects that the project will be compared to. 10 hours.

2.3.3 Library development

1. **Project creation:** creation of project and establishment of project repository. 1 hour.
2. **Feature calculation development:** use of a feature calculation library to extract features from input image. 30 hours.
3. **Classifier creation and training development:** use of Machine Learning to generate and train classifier based one aforementioned features. 22 hours.
4. **Result image creation development:** applying the trained classifier to generate result images. 15 hours.
5. **Testing method development:** generate tests to identify errors in development. 12 hours.

2.3.4 GUI development

1. **Interface design:** design an interface based on expected features and framework capabilities. 5 hours.
2. **Interface development:** develop the interface based on design. 55 hours.
3. **Utility merging from library:** implement utilities using developed library. 10 hours.
4. **Adding new utilities to library to accommodate GUI related new uses:** implement new utilities to library if need arises. 25 hours.
5. **GUI testing:** test GUI to identify errors in development. 10 hours.

2.3.5 Evaluation

1. **Evaluation script development:** develop scripts to automate evaluation process. 10 hours.
2. **Script execution:** execute developed scripts. 5 hours.
3. **Result formatting:** format results to facilitate result interpretation. 5 hours.
4. **Result interpretation:** interpret generated results. 10 hours.

2.3.6 Reporting

1. **Planning reporting:** plan structure and development of report. 5 hours.
2. **Task development reporting:** enumerate specific tasks to be developed for the project. 10 hours.
3. **Phase development reporting:** group defined tasks into development phases. 10 hours.
4. **Final report development:** develop a final report describing project. 35 hours.

2.4 Project calendar

During the starting and main phases of the development of this project I will be staying in Finland as part of an Erasmus exchange program. However this fact has been communicated before to the supervisors and it is not expected to interfere with the normal development of the project; nevertheless, due to the work distribution of the studies carried out during that phase, even if the initial research for the project will start during the 2017 Autumn period (September-December), the bulk of the project is expected to be developed during the 2018 Spring period (January-May), when the work load from other courses is expected to be lower. This will also enable an in person meeting during the Winter break to solidify the planning of the project.

2.4.1 Project duration estimation

The project starts on the first of September, 2017, and is planned to have been completed by the fifteenth of June, 2018, when the registration for the defense of the project is expected to be made.

2.4.2 Phase distribution

Table 2.1 displays the phase distribution calendar. These dates are tentative and, as mentioned before, do not represent the work load of each phase, as it is expected that the work load related to this project will be higher during the spring period of 2018. Additionally,

Table 2.1: Phase distribution

Phase	Start date	End date
Planning	2017/09/01	2017/10/01
Background research	2017/10/01	2017/12/21
Library development	2018/01/15	2018/03/15
GUI development	2018/03/15	2018/05/15
Evaluation	2018/05/15	2018/06/01
Reporting	2017/09/01	2018/06/15

the reporting phase is set to be carried out during all of the project as it includes the development of this document and the day to day reporting to the project supervisors of the work that is being done. Finally, even if each phase is defined with a start and end date it is to be expected that tasks related to different phases may be revisited during the development of different phases as issues may arise, this can happen for example when during GUI development or evaluation the need for new functionalities from the library may arise, creating the need to revisit the library development phase.

2.5 Risk analysis

The following risks have been identified in relation with the project, together with a number of contingency plans to avoid or mitigate resulting losses:

- Software or hardware issues: the software side of the project will be carried out using software tools that will provide version control and backup systems, this means that in the case of hardware failure the developed work will not be lost, and in the case of software failure only the work developed before the latest upload will be lost, which should be of a few hours at most. Additionally, the development of the report will be done using an on-line service that will provide on-line backup. Finally, in the case of total hardware failure the university in which I will be studying offers students with laptops in which I could continue to work until a replacement had been acquired.

- Time loss due to unexpected schedule changes: if an unexpected issue may arise that would lead to time loss the lax starting and ending dates of the different phases of the project would allow for reallocation of tasks during the planned calendar to avoid delays.

- Information loss: as mentioned before, project related software will be managed through a backup and version control system and project reporting will be managed through a service that offers on-line backup, however other materials related to the project such as testing data, testing results and other background research related materials will need to be backed up too to avoid any possible losses.

Phase	Task	Start time	End time	Total time
Planning		2017/09/01	2017/10/01	5
	Project objective definition			3
	Project reach definition			1
	Definition of communications with project supervisors			1
Background research		2017/10/01	2017/12/21	20
	Development framework			2.5
	Image processing library			2.5
	Feature extraction library			2.5
	Machine learning library			2.5
	Evaluation method research			10
Library development		2018/01/15	2018/03/15	80
	Project creation			5
	Feature calculation development			30
	Classifier creation and training development			20
	Result image creation development			15
	Testing method development			10
GUI development		2018/03/15	2018/05/15	105
	Interface design			5
	Interface development			55
	Utility merging from library			10
	Adding new utilities to library to accommodate GUI related new uses			1525
	GUI testing			10
Evaluation		2018/05/15	2018/06/01	30
	Evaluation script development			10
	Script execution			5
	Result formatting			5
	Result interpretation			10
Reporting		2017/09/01	2018/06/15	60
	Planning reporting			5
	Task development reporting			10
	Phase development reporting			10
	Final report development			35
Total				300

Table 2.2: Summary of project plan

CHAPTER 3

Background Research

This chapter aims at providing a knowledge base with which to understand the project that has been developed, by explaining the core concepts in which this project has founded, while providing references for those interested in furthering their understanding of these core concepts.

3.1 Image Analysis

A digital image has been defined as "a discrete representation of data possessing both spatial (layout) and intensity (colour) information"[27], this representation of a digital image provides the opportunity for a mathematical approach to image analysis, and therefore as we will later discuss a Machine Learning approach.

In the quoted text a reference is made to a layout, this layout usually represented as a two-dimensional array of values represents the distribution of the individual intensity values defined as pixels. However, it is worth considering that in specific applications such as some biomedical applications three dimensional images can be found as a result of other imaging techniques [11]. Although the development of this project has been focused on two dimensional images, the produced library should allow for its use on three dimensional images as well.

Colour refers to the intensity measurements of each particular pixel location, in the case of grayscale images this value is typically represented by a single value that displays different shades of gray ranging from black to white, represented as zero to a maximum value [27]. Additionally, a representation of visual colours can be achieved through the combination of different colour channels, the most typical of which is the RGB channel separation, which assigns each pixel with three values, representing the intensity of Green, Red and Blue colours [27].

Even if colour processing is not the focus of this project, it is worth mentioning that different colour spaces exist that aim at providing with different approaches, such as that of the Hue, Saturation and Value (HSV) colour space that provides a perceptual approach to colour space [27]. In this project, colour images have been analyzed using the CIELAB or Lab colour model; this model was developed by the CIE with the aim of representing human perception of colour [7]. Further details about the specific treatment of colour images will be provided in the next chapter.

3.2 Machine Learning

Although it is hard to summarize a vast field such as that of Machine Learning in a single phrase, Machine Learning can be said to be the field of Artificial Intelligence concerned with the process of learning as applied to a computer. In the introduction to the book "Machine Learning: An Artificial Intelligence Approach"[22] Machine Learning is presented as the challenge of transferring the learning process to computers, and is said to be "a most challenging and fascinating long-range goal in artificial intelligence". This introduction provides too the separation of Machine Learning into three primary research foci: Task-Oriented Studies, Cognitive Simulation and Theoretical Analysis. This project will be focusing on the Task-Oriented side of Machine Learning, as the aim of the project is the development of a tool that enables task-oriented model building and it's not the exploration of theoretical concepts within Machine Learning or the research into the process of human cognition and its simulation.

More specifically, within Machine Learning this project focuses in the classification of datasets, since the developed project aims at taking a dataset, an input image, and generating a classified dataset, an image where each value has been labeled. Classification methods can be categorized into supervised learning methods and unsupervised learning methods. "Every instance in any dataset used by Machine Learning algorithms is represented using the same features [...] if instances are given with known labels (the corresponding correct outputs) then the learning is called supervised [...] in contrast to unsupervised learning, where instances are unlabeled."[17] this quote from a review of classification techniques represents the core difference between supervised and unsupervised learning, the label, or lack thereof. This difference results in a different approach to learning, as with one it is possible to evaluate the results produced during training while with the other it is not.

While this project is related to both unsupervised and supervised learning —the superpixel images that are used in the library are usually generated through the use of unsupervised clustering techniques, and the developed library uses supervised learning to classify the images— the project focuses on supervised learning, specifically on classifiers. The classifiers that have been used for the evaluation of this project have been selected from the list of classifiers that the WEKA Machine Learning library offers, more detail about this library will be offered in the following chapter. The following are a list of the classifiers that have been used in the evaluation of the developed library:

- **BayesNet:** WEKA implementation of a Bayesian Network. It offers the basis for different configurations of a Bayesian Network [5], but the default settings, which were used in the evaluation process of this project, use the K2 algorithm as a search algorithm, which is a hill climbing algorithm [8].

- **J48:** J48 is the WEKA implementation of the C4.5 tree building algorithm [21]. By default it generates pruned C4.5 decision trees, but can be modified to stop the pruning.

- **LogitBoost:** LogitBoost is the WEKA implementation of an additive logistic regression algorithm. "Boosting works by sequentially applying a classification algorithm to reweighted versions of the training data and then taking a weighted majority vote of the sequence of classifiers thus produced"[12]. This boosting procedure is applied here into the DecisionStump tree classifier. Decision stump classifiers use one-level decision trees [15].

- **RandomForest:** It creates a combination of prediction trees in order to form a "forest"of these trees. The following formal definition of a random forest is provided by Leo Breiman in [6]:

  "A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\boldsymbol{x},\Theta_k), k = 1,...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\boldsymbol{x}$."

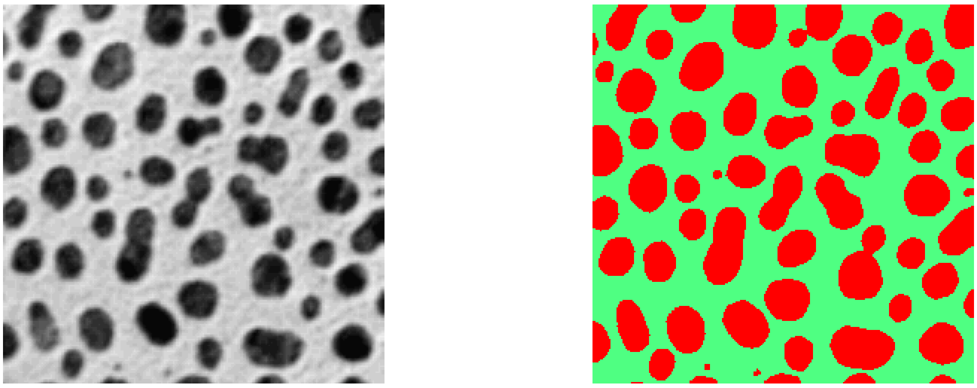

Figure 3.1: Segmentation of an image

- **SMO:** SMO implements John Platt's sequential minimal optimization algorithm [20] to train Support Vector Machines (SVMs). This implementation normalizes all attributes and transforms nominal attributes into binary attributes. Additionally, it solves multi-class problems using a pairwise classification, this means that each class classification is pitted against each other in a pairwise manner.

Further specific details about the implementation and options offered by these classifiers can be found in the WEKA API website[1].

When using large datasets, it is common that these may either include invalid values or biased distributions of classes. To solve this problem, it is common to use filters during the preprocessing stage. In this project the datasets inferred displayed an unbalance in the class distribution, to avoid this, a re-sampling filter was used. The WEKA implementation of a re-sampling filter produces a random subsample of a dataset, and, when specified in the options, produces a dataset with a uniform class distribution. This filter was applied to the training datasets.

3.3 Image Segmentation

Image Segmentation is the process by which an image is partitioned into several segments. One way of achieving this partition is by using pixel (or superpixel) classification or clustering. Classification-based segmentation processes start with training sets where each pixel of the image has already been classified as belonging to a class and with this a model is built that can be later used to classify other pixels or sets of pixels. Clustering-based

[1] http://weka.sourceforge.net/doc.dev/

Figure 3.2: Superpixel segmentation of an image

segmentation relies on feature extraction for each pixel and creates clusters of pixels with similar or related features. Different algorithms may provide control over the amount of clusters to be created but overall no previous information is given to the algorithm that could guide the clustering other than what can be extracted from the image itself.

Image classification at any level (pixel, superpixel or whole image) is a core component of computer vision. In fact, many of the main computer vision challenges such as image segmentation, object detection or face detection can be reduced to a problem of image classification [2]. This project deals specifically with image segmentation, but the core processes that are developed as part of this library could be adapted to be used in the aforementioned tasks. Figure 3.1 shows an example of the segmentation of an image, the image to the left has been segmented into two distinct classes represented by the red and green colours.

3.4 Pixel clustering: Superpixels

"Superpixel algorithms group pixels into perceptually meaningful atomic regions, which can be used to replace the rigid structure of the pixel grid"[1], as explained in this quote superpixel algorithms can be used to replace the meaningless grid representation of pixels, providing a reduced dataset with which to work with by capturing image redundancy. This reduction on the complexity of a dataset can be critical in certain computer vision contexts, where the reduction of pixels into pixel regions will reduce the complexity of the application of classification and therefore increase the speed and reduce memory usage.

Although this project does not deal with the generation of superpixel images it is worth going briefly over the main classes of superpixel generating methods [1]:

- Graph-based algorithms

  Graph-based approaches treat pixels in an image as nodes in a graph, with edges in this graph representing the similarity between neighboring pixels. Superpixels are created thus by bundling together neighboring pixels through the use of a cost function.

  Graph-based algorithms include normalized cuts [25], which recursively partitions the graph of all of the pixels through the use of contour and texture features and the segmentation algorithm presented by Felzenszwalb and Huttenlocher [10] which agglomerate pixels as nodes of a graph so that each superpixel is the minimum spanning tree of the constituent pixels.

- Gradient-ascent-based algorithms

  Gradient-ascent-based algorithms start with a rough clustering of pixels and iteratively refine the clusters until a convergence criteria is met, such as a specific amount of clusters or cohesion within clusters.

  Gradient-ascend-based algorithms include Watershed [29] which performs a gradient ascent starting from a local minima to produce separating lines.

The aforementioned review [1] provides further examples of these categories and analyzes the different performances of these algorithms.

Figure 3.2 shows the superpixel segmentation of an image, the right image displays the different areas that the algorithm has identified to result in a single superpixel. As shown by the colours each of the region has a different label as these have not been classified.

3.5 Image Feature Extraction

As above mentioned digital images provide intensity measurements for each pixel in the image, and thus information can be extracted from these values to gain information on the contents of the image. These extracted features are the values that are going to be used during pixel or superpixel classification, and therefore the different feature extraction techniques will affect the later segmentation process.

A review by Ping Tian, D. [19] found that three main image feature categories could be identified: colour features, texture features and shape features; this review listed strengths and weaknesses of different features belonging to each category.

Colour features are extracted by analyzing the intensity values of different pixels or regions of the image. Among the different intensity features, color moments or CMs are identified as being "one of the simplest yet very effective"; these include features such as standard deviation and skewness [19].

Texture features are extracted by analyzing groups of pixels, and due to their strong discriminative capacity, texture features are commonly used in image retrieval and semantic learning techniques [31]. The aforementioned review identified two main categories within texture features: spatial textures and spectral textures [19]. Further studies such as [31] identify specific methods for each of this two categories.

Shape feature extraction looks for "effective and perceptually important shape features"[32]. These features can be extracted by calculating features only from the boundary of the shape or extracting features from the whole region enclosed by the shape [19]. This differentiation results in the categorization of shape feature extraction techniques into two different categories: contour based methods and region based methods [19].

The library that was used for the development of the project provides the following intensity features:

- Max

  Represents the maximum intensity value of the region.

- Min

  Represents the minimum intensity value of the region.

- Mode

  Represents the most common value of intensity of the region.

- Median

  Represents the middle value of the intensities of the region.

- Mean

  Represents the mean value ($\bar{x}$) of intensities of the region:

  $$\bar{x} = \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)$$

  where N is the amount of pixels of the region.

- Standard Deviation

  Represents the standard deviation or σ of the region:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

  where N is the amount of voxels (volumetric pixels) of the region and $\bar{x}$ is the mean value of the region.

- Kurtosis

  Kurtosis is the fourth standardized moment, defined as:

$$\mathrm{Kurt}[X] = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{\mu_4}{\sigma^4} = \frac{\mathrm{E}[(X-\mu)^4]}{(\mathrm{E}[(X-\mu)^2])^2}$$

  where μ_4 is the fourth central moment and σ is the standard deviation. However the library used for feature extraction uses $\mathrm{Kurt}[X] - 3$.

- Skewness

  Skewness is the third standardized moment, defined as:

$$\gamma_1 = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\mu_3}{\sigma^3} = \frac{\mathrm{E}\left[(X-\mu)^3\right]}{(\mathrm{E}\left[(X-\mu)^2\right])^{3/2}} = \frac{\kappa_3}{\kappa_2^{3/2}}$$

  where μ is the mean, σ is the standard deviation, E is the expectation operator (the expected value of a random variable), μ_3 is the third central moment and κ_t are the t^{th} cumulants.

Additionally, the library also offers the same features calculated over all the neighboring (adjacent) regions. Although these intensity measures are extracted from grayscale images, these same features can be calculated from colour images by separating the different channels and processing them individually.

CHAPTER 4

Description of the Developed Project

This chapter describes the project that has been developed, providing explanations of the different artifacts and functionalities that have been produced as a result of this project.

4.1 Used software

4.1.1 Development framework

Due to the previous experience working with this framework and the availability of other related libraries, this project has been developed using the ImageJ platform, and more specifically using the Fiji distribution, which provides additional functionality to ImageJ [23]. ImageJ provides an open source framework that allows a varied community of scientists "ranging from experimental biologists to paleontologists to astronomers to computer scientists"[24] to develop and share tools for image processing. In addition, Fiji provides a further development by bundling standard libraries for computer vision research and providing further support for plug-in development. This platform has achieved international recognition, being used in every major academic research center throughout the world. Fiji has facilitated the use of novel algorithms that otherwise would have required biologists a great effort to access, therefore, it has enabled and eased cooperation among fields.

4.1.2 Machine Learning library

The Waikato Environment for Knowledge Analysis or WEKA, is a project that "aims to provide a comprehensive collection of Machine Learning algorithms and data preprocessing tools to researches and practitioners alike"[13], from preprocessing algorithms to result interpretation utilities, going through classification and clusterization algorithms. "Unlike other Machine Learning projects, the emphasis is on providing a working environment for the domain specialist rather than the Machine Learning expert"[14], this excerpt from the abstract of a 1994 WEKA publication provides insight into why by 2009 they reported 1.4 million downloads since its release on SourceForge [13], it is an easy to use library that doesn't require users any previous deep knowledge of Machine Learning in order to use it and allows scientists from multiple disciplines to use Machine Learning algorithms.

While WEKA offers a variety of features, this project used the features regarding dataset creation, dataset filtering and classifier training and application. Although WEKA offers features for evaluation, after consideration it was decided to use another library that offered label image comparisons to facilitate comparison with other libraries by using the resulting images of the clustering.

4.1.3 Image feature extraction library

The MorphoLibJ provides a set of tools for image processing based on Mathematical Morphology (MM) [18], defined as "a theory for the analysis of spatial structures [...] it aims at analysing the shape and form of objects [...] the analysis is based on set theory, integral geometry, and lattice algebra."[26]. It provides different functions for image processing, but this project makes use specifically of its feature extraction capabilities and it's label image analysis capabilities.

As mentioned on the previous chapter, MorphoLibJ calculates the mean, standard deviation, maximum, minimum, median, mode, skewness and kurtosis of the intensity value over regions of pixels or voxels, together with its neighboring regions. To do this, MorphoLibJ requires a grayscale image and a labeled image, and returns a table with the intensity features per label.

4.2 Library development

The first phase of the development of the project started with the development of a library that would allow for the feature extraction and subsequent classification of images based on a superpixel image and an input image. To do this a class was created that would be responsible for the extraction of region features. This class would use the aforementioned MorphoLibJ library to extract the features from the input images and would translate the results into an *Instances* object that could be inputed into a WEKA classifier.

In order to allow the usage of colour images, an additional class was created that would be responsible for the extraction of features from coloured images. To do this, the input RGB image would be translated into the Lab colour space and a new grayscale image would be generated from each of the three channels. Using these three grayscale images, features would be extracted and then merged into a single *Instances* object. An example of the splitting can be seen in figure 4.5.

Additionally, these classes offer the option to add a groundtruth image to the feature extraction, this allows the creation of *Instances* objects with class attributes, and therefore can later be used for training a classifier.

The main Java class of the library handles the contact with this classes and with the WEKA library. It offers the functionality of region feature calculation, classifier training and application into images, and probability map creation. Region features are calculated through the aforementioned classes, after the main class checks whether the provided input image is an RGB colour image or not. A classifier can be trained based on a list of regions and classes, whereupon a new training dataset will be created that will include the class labels that were provided in the aforementioned list, or through an already provided training data that includes the necessary class labels. The trained classifier can then be applied either to the loaded input image or a new input image can be provided together with a corresponding label image. Additionally, probability maps can be created using the trained classifier to calculate the probability distributions for each class per region; the resulting image will be an image stack where each slice of the image represents the probability that a pixel belongs to a class through its intensity, with higher intensity values representing a higher probability of belonging to said class.

Figure 4.13 shows an example of a model building and testing process using this library. Figure 4.6 shows the image used for the training of the model, with figure 4.7 showing the corresponding superpixel image and figure 4.8 showing the groundtruth image. Figure

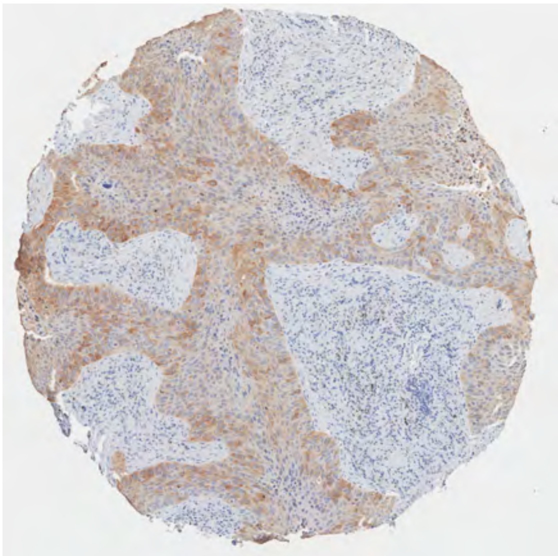

Figure 4.1: original RGB image

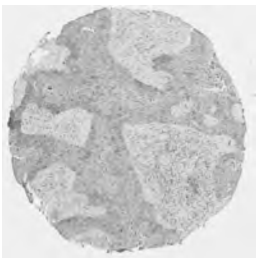

Figure 4.2: L* channel

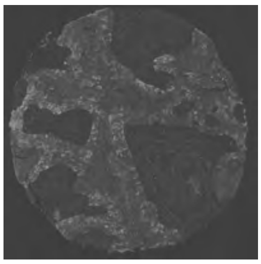

Figure 4.3: a* channel

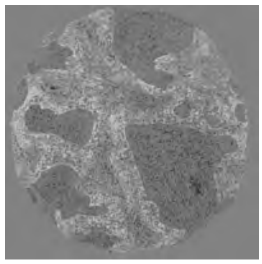

Figure 4.4: b* channel

Figure 4.5: RGB image split into L* a* and b* channels

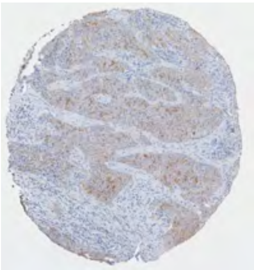

Figure 4.6: Image to be used on the training of the model

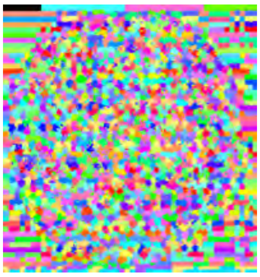

Figure 4.7: Label image of training image

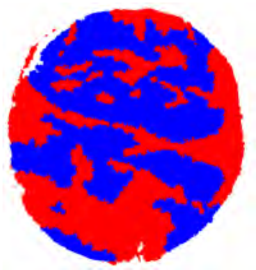

Figure 4.8: Groundtruth image of training image

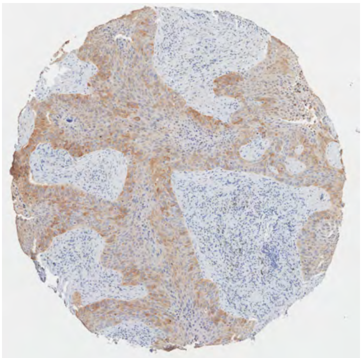

Figure 4.9: Testing image

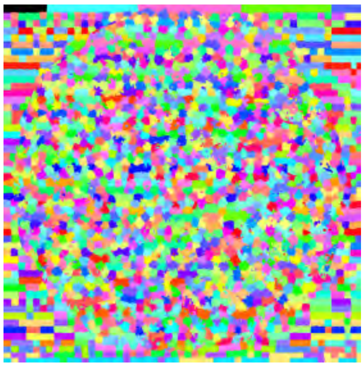

Figure 4.10: Label image of testing image

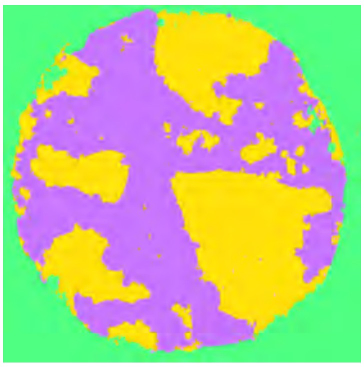

Figure 4.11: Resulting image

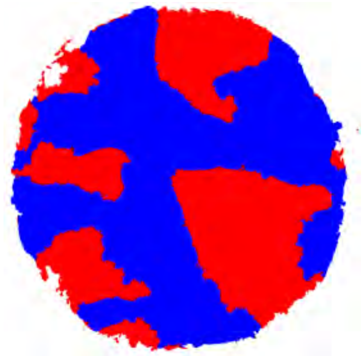

Figure 4.12: Groundtruth image of testing image

Figure 4.13: Example of a superpixel labeled image model training and applying

4.9 shows the image where the model will be tested, with its corresponding superpixel image in figure 4.10. The results can be seen in figure 4.11, with figure 4.12 showing the groundtruth corresponding to the testing image.

4.3 GUI development

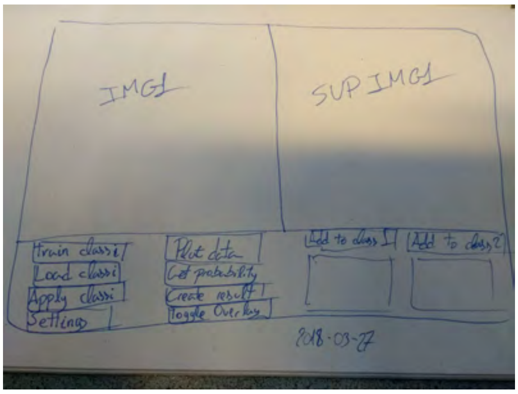

Figure 4.14: Early design of GUI

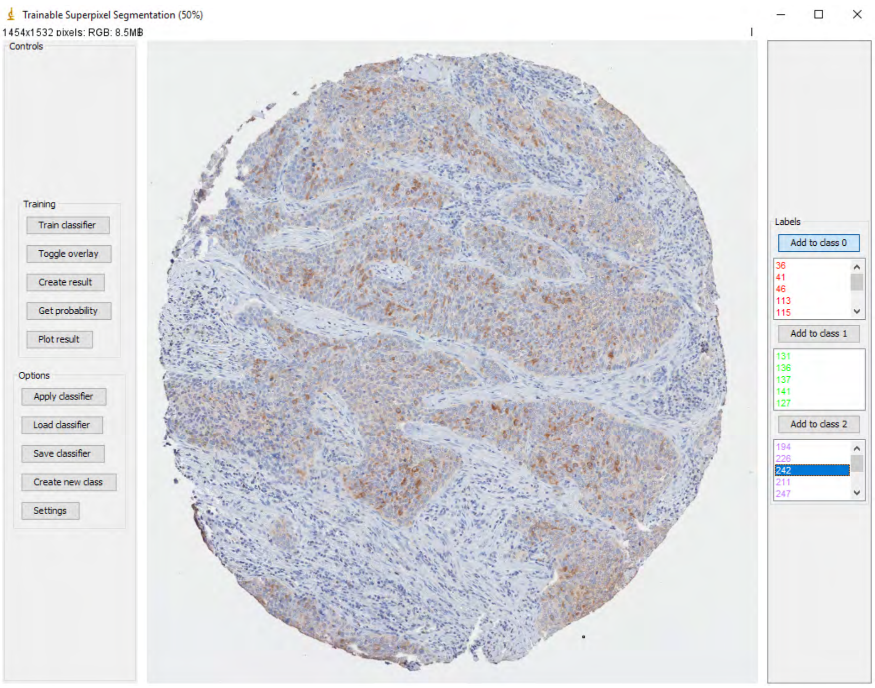

Figure 4.15: Final design of GUI

After the library was developed, a graphical user interface (GUI) was created to offer a more accessible way of interacting with the library. The process started with a listing of the features wanted to be offered in the GUI and with early designs of possible layouts. Figure 4.14 shows an early design where the input image and its corresponding superpixel image would be displayed side to side, this was later discarded in favour of the use of an overlay to display the superpixel image and the result image. Figure 4.15 shows the final design of the GUI.

The final version of the interface offers the following features:

- Region selection

  The ImageJ multi-point tool provides region selection on the displayed image, the label number related to the region that has been selected will be listed on the box under the class button. Clicking the label number will display the point where the label was selected. Additionally, double clicking on a number will delete that label from the list.

  The point selection is handled through an ImageJ class named *ROI* (Region Of Interest), this class provides information about the location of a selection on an image, and is used in this project to point to the label on the superpixel image.

- Train classifier

  The classifier will be trained based on the regions that have been selected. To do this, if the region features of the input image have not been calculated yet, they will be, and a WEKA-compatible dataset will be created with them to represent all the superpixels in the region feature space. After that dataset has been created, a training subset will be created with the regions corresponding to the user-selected points of each class. In all cases, the region features to use are those that have been selected on the settings dialog. After the classifier has been trained, it will be applied to the superpixels of the whole image, and an overlay will be displayed with the resulting image. Additionally, the resulting image will be displayed on a new ImageJ window.

- Toggle overlay

  With this option, the displayed image will rotate over three different states: input image 1) with no overlay, 2) with original superpixel image overlay, and 3) with result overlay. In the case that a result has not been calculated yet only the first two states will be cycled through. The different overlay options are shown in figure 4.16

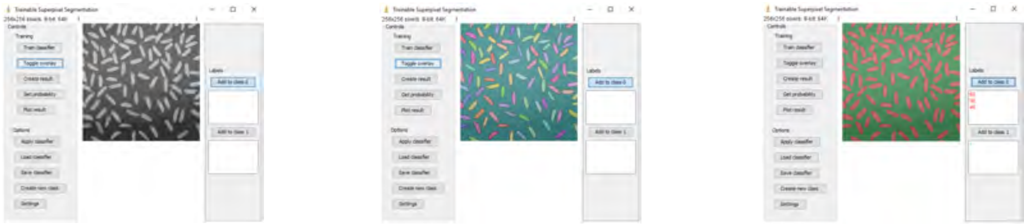

Figure 4.16: Different overlay options, from let to right: No overlay, superpixel overlay, result overlay.

- Create result

  If a segmentation result has already been calculated, a copy of the resulting image will be created and displayed. If a result has not been calculated yet, a new one will be created by training a classifier (if it has not been trained yet) or by applying an already trained classifier.

- Get probability

  Using the trained classifier, probability maps will be calculated for each class and an image stack will be created and displayed, with each slice representing a probability map for its corresponding class.

- Plot result

  If a classifier has been trained, it will display a statistics window provided by the WEKA library with information about the training.

- Apply classifier

  The classifier will be applied. If a classifier has not been trained or loaded from file, a new one will be trained based on the selected regions and assigned classes and it will be applied to the current image.

- Load classifier

  A dialog will be created offering the option to load a WEKA classifier from file. These classifiers are stored as `.model` files. The program will check the classifier to look for the number and name of classes that the classifier has been trained with and will update the GUI accordingly.

- Save classifier

  A dialog will be created offering the option to save the current WEKA classifier. These classifiers are stored as `.model` files.

- Create new class

  A dialog will appear asking for the name of a new class, and the newly created class will appear together with the default two classes.

- Settings

  A settings dialog will appear. The settings dialog will offer the option to change the selected features, the opacity of the overlay and the used WEKA classifier. The features are displayed as a list of check boxes that can be selected or deselected to indicate whether they should be used in the calculation of the features. The overlay opacity can be selected either through a slider or through an input box where a value can be introduced from 0 to 1, where 0.33 is the default value. Finally, a WEKA classifier can be chosen and its options changed by clicking on the classifier text box itself. This is handled through the WEKA library directly so all options offered by WEKA are also offered here. Figure 4.17 shows the settings window.

Figure 4.17: Settings dialog

During the development of the GUI, a need arose for further implementations in the library. Mainly these changes reflected the need for a more dynamic access to the internal variables of the library's main class. The tests that were used to develop the library only required a straight use of the library where the main variables were defined once and didn't require any further change during execution. However, the GUI offered its users the option to change, save or load classifiers and change class labels.

CHAPTER 5

Evaluation

In order to draw meaningful conclusions about the developed project, an evaluation phase was designed in which the library would be used to segment an image database, and the results would be compared to results produced by other libraries.

5.1 Image database

The image database was kindly provided by the Centre of Applied Medical Research (CIMA[1]). The database included 10 Tissue MicroArray Analysis (TMA) images and 10 corresponding hand-drawn label images.

5.1.1 Image content

TMA is a process by which tissue samples are collected and processed. This procedure is commonly used in lung cancer detection. The database that has been used for this evaluation corresponds to a set of lung tissue extractions. These images were acquired and hand-labeled by experts and thus offer a great example of real-world use for the library. The tissue images have been labeled with the following tags: tumoral, nontumoral and background.

[1] https://cima.unav.edu/

5.1.2 Preprocessing

The dataset was preprocessed by one of the thesis supervisors before being provided for the evaluation, the changes made during the preprocessing were the following:

1. Scaling: due to the large size of the original TMA images, around 6000×6000 pixels, the TMA images were rescaled to around 25%.
2. Histogram matching: the images were histogram matched to the first image of the dataset as a way of normalizing the histograms of all images.
3. Superpixel segmentation: the superpixel images were generated through the use of jSLIC [4] with default parameters.
4. In order to generate the groundtruth label images, the original hand-drawn images were taken, and through the use of a majority voting method the different regions on the aforementioned superpixel images were classified.

5.2 Comparison library

The developed library was compared against a library with a similar implementation and goals, Trainable Weka Segmentation (TWS) [3]. This library, developed as part of a Fiji plug-in, combines a collection of machine learning algorithms with a set of selected image features to produce pixel-based segmentations.

5.2.1 Features

Due to the different methods for feature extraction the two libraries use, a selection had to be made of which features to use in each library in order to allow for a fair comparison of capabilities. The following were the final attributes selected for each library:

- Trainable Weka Segmentation:
  - *Original* Gray scale intensity value of the pixel
  - *Hue* Hue value of the HSB channels of the pixel
  - *Saturation* Saturation value of the HSB channels of the pixel

- *Brightness* Brightness value of the HSB channels of the pixel
- *Mean*_1.0 Mean of the pixels with a radius of 1 pixels from original pixel
- *Minimum*_1.0 Minimum of the pixels with a radius of 1 pixels from original pixel
- *Maximum*_1.0 Maximum of the pixels with a radius of 1 pixels from original pixel
- *Median*_1.0 Median of the pixels with a radius of 1 pixels from original pixel
- *Mean*_2.0 Mean of the pixels with a radius of 2 pixels from original pixel
- *Minimum*_2.0 Minimum of the pixels with a radius of 2 pixels from original pixel
- *Maximum*_2.0 Maximum of the pixels with a radius of 2 pixels from original pixel
- *Median*_2.0 Median of the pixels with a radius of 2 pixels from original pixel
- *Mean*_4.0 Mean of the pixels with a radius of 4 pixels from original voxel
- *Minimum*_4.0 Minimum of the pixels with a radius of 4 pixels from original pixel
- *Maximum*_4.0 Maximum of the pixels with a radius of 4 pixels from original pixel
- *Median*_4.0 Median of the pixels with a radius of 4 pixels from original pixel
- *Mean*_8.0 Mean of the pixels with a radius of 8 pixels from original pixel
- *Minimum*_8.0 Minimum of the pixels with a radius of82 pixels from original pixel
- *Maximum*_8.0 Maximum of the pixels with a radius of 8 pixels from original pixel
- *Median*_8.0 Median of the pixels with a radius of 8 pixels from original pixel
- *Mean*_16.0 Mean of the pixels with a radius of 16 pixels from original pixel
- *Minimum*_16.0 Minimum of the pixels with a radius of 16 pixels from original pixel
- *Maximum*_16.0 Maximum of the pixels with a radius of 16 pixels from original pixel

  - *Median_16.0* Median of the pixels with a radius of 16 pixels from original pixel

- Trainable Superpixel Segmentation:
  - $Mean - L$ Mean of L value of the pixels of the superpixel.
  - $Min - L$ Min of L value of the pixels of the superpixel.
  - $Max - L$ Max of L value of the pixels of the superpixel.
  - $Median - L$ Median of L value of the pixels of the superpixel.
  - $NeighborsMean - L$ Mean of L value of the pixels of the superpixel and the neighboring pixels.
  - $NeighborsMin - L$ Min of L value of the pixels of the superpixel and the neighboring pixels.
  - $NeighborsMax - L$ Max of L value of the pixels of the superpixel and the neighboring pixels.
  - $NeighborsMedian - L$ Median of L value of the pixels of the superpixel and the neighboring pixels.
  - $Mean - a$ Mean of a value of the pixels of the superpixel.
  - $Min - a$ Min of a value of the pixels of the superpixel.
  - $Max - a$ Max of a value of the pixels of the superpixel.
  - $Median - a$ Median of a value of the pixels of the superpixel.
  - $NeighborsMean - a$ Mean of a value of the pixels of the superpixel and the neighboring pixels.
  - $NeighborsMin - a$ Min of a value of the pixels of the superpixel and the neighboring pixels.
  - $NeighborsMax - a$ Max of a value of the pixels of the superpixel and the neighboring pixels.
  - $NeighborsMedian - a$ Median of a value of the pixels of the superpixel and the neighboring pixels.
  - $Mean - b$ Mean of b value of the pixels of the superpixel.
  - $Min - b$ Min of b value of the pixels of the superpixel.
  - $Max - bMax$ Mean of b value of the pixels of the superpixel.

- $Median - b$ Median of b value of the pixels of the superpixel.
- $NeighborsMean - b$ Mean of b value of the pixels of the superpixel and the neighboring pixels.
- $NeighborsMin - b$ Min of b value of the pixels of the superpixel and the neighboring pixels.
- $NeighborsMax - b$ Max of b value of the pixels of the superpixel and the neighboring pixels.
- $NeighborsMedian - b$ Median of b value of the pixels of the superpixel and the neighboring pixels.

As a result of this selection, both libraries had access to the same amount of attributes.

5.2.2 Samples

Due to the TSS library using superpixels as instances instead of pixels, and with the aim of offering both libraries the same amount of training instances, a subset of pixels was selected for the training of the TWS library. On average the superpixel segmented images of the dataset have 3727.9 superpixels, therefore and in order to offer a balanced class distribution of instances 3729 ($1243 * 3$) would be taken for the TWS evaluation, and the instances of the TSS library would be balanced through the use of the same filter that the TWS library uses to add random balanced data. This has been further explained in chapter 3.2.

5.3 Evaluation method

For the evaluation a ten-fold cross-validation method was used, as the provided image dataset was composed of ten images. This process worked by conducting ten different evaluations, each of which was done by using nine of the images for the building of a classifier and the remaining image for the evaluation.

5.3.1 Evaluation metrics

The following evaluation metrics were used in this evaluation:

- Jaccard index: this index defined by Paul Jaccard in 1908 [16] measures similarity and is defined as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Where A and B are the two intersecting areas. The result is a value between 0 and 1 where 0 represents no overlap and 1 represents perfect overlap.

- Dice coefficient: the Sørensen–Dice coefficient also known by Dice coefficient was independently developed by both Sørensen [28] and Dice [9] and was defined as:

$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

Where A and B are the two intersecting areas. The result is a value between 0 and 1 where 0 represents no overlap and 1 represents perfect overlap.

- Confusion matrices: confusion matrices allow for an easier visualization of correct and incorrect instance classification by arranging in each row the predicted class while presenting in each column the actual class. Additionally, the library that was used to generate these, TWS, offers the precision and recall statistics together with each confusion matrix. The resulting table can be seen in table 5.1.

  True Positive or TP represents the number of real positive cases that have been identified as such, True Negative or TN represents the number of real negative cases that have been identified as such, False Positive or FP represents cases where a positive case was predicted where a negative case existed, and False Negative or FN represents cases where a negative case was predicted were a positive case existed.

  Precision is defined as

$$Precision = \frac{TP}{TP + FP}$$

  Recall is defined as

$$Recall = \frac{TP}{TP + FN}$$

- Accuracy: drawn from the confusion matrix represents the percentage of correctly classified instances over all instances. Calculated as:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

	Groundtruth Class A	Groundtruth Class B	Precision
Predicted A	TP	FP	$Precision_A$
Predicted B	FN	TN	$Precision_B$
Recall	$Recall_A$	$Recall_B$	$Accuracy$

Table 5.1: A confusion matrix

Additionally, this process was conducted for each of the five classifiers that were mentioned in the background research chapter: BayesNet, J48, LogitBoost, RandomForest and SMO.

5.4 Evaluation scripting

As the evaluation process required the execution of processes multiple times for different classifiers and different datasets, executing them manually was unfeasible. Thus, it was decided to develop programs to implement the evaluation process. For the evaluation of the TWS library through the use of the tutorials available in the corresponding wiki page[2], Beanshell scripts were developed for each classifier, and then a general script was developed that would execute sequentially all aforementioned scripts. These scripts loaded the image dataset and carried out the ten-fold cross-validation process while storing the resulting image and statistics. For the TSS library, Java code was developed that carried out all of the ten-fold cross-validations in a single program. The scripts were executed on the same computer through the same workload.

5.5 Evaluation results

As displayed in tables 5.2 and 5.3, on average the TSS obtained better results in all three classification metrics. Although in certain classifiers TWS may obtain marginally better results as seen in 5.1, 5.3 and 5.2 if only the best resulting classifiers were to be taken into account, as would be in a real world application where the objective is to optimize the results, TSS shows its ability to obtain competitive results. However, it is worth mentioning that the recall for tumoral sections as seen in figures 5.4 and 5.5 is higher for the TWS library, indicating a higher success rate at identifying tumoral sections of the images. However the rest of the metrics indicated on the aforementioned confusion matrices

[2] https://imagej.net/Scripting_the_Trainable_Weka_Segmentation

indicate a better performance on the TSS library. Additionally, it is worth noting that over both libraries RandomForest was the classifier that achieved the best results.

	BayesNet	J48	LogitBoost	RandomForest	SMO	Average
Accuracy	0.8583	0.8356	0.8826	0.8880	0.8596	0.8648
Dice	0.8466	0.8217	0.8761	0.8795	0.8503	0.85484
Jaccard	0.7575	0.7226	0.7933	0.8008	0.7640	0.76764
Training time (ms)	513.7	1775.1	3353	14778.1	4691.3	5022.24

Table 5.2: Average results for 10 folds for the developed Trainable Superpixel Segmentation library

	BayesNet	J48	LogitBoost	RandomForest	SMO	Average
Accuracy	0.8302	0.8375	0.8562	0.8679	0.8622	0.8508
Dice	0.8217	0.8297	0.8492	0.8621	0.8569	0.8439
Jaccard	0.7180	0.7253	0.7546	0.7716	0.7647	0.7468
Training time (ms)	453.5	2084.1	3322.4	14634.7	8294.7	5757.88

Table 5.3: Average results for 10 folds for Trainable Weka Segmentation library

Label	GT background	GT tumoral	GT nontumoral	Precision
Predicted background	34694680	121262	363284	0.9862
Predicted tumoral	54880	28629608	4086884	0.8736
Predicted nontumoral	759075	8148780	25380597	0.7402
Recall	0.9771	0.7759	0.8508	0.8676

Table 5.4: Aggregated Confusion Matrix of 10 folds and 5 classifiers for Trainable Superpixel Segmentation

Label	GT background	GT tumoral	GT nontumoral	Precision
Predicted background	33886653	154225	497605	0.9811
Predicted tumoral	122395	29778646	5978527	0.8299
Predicted nontumoral	1499587	6966779	23354633	0.7339
Recall	0.9543	0.8070	0.7829	0.8511

Table 5.5: Aggregated Confusion Matrix of 10 folds and 5 classifiers for Trainable Weka Segmentation

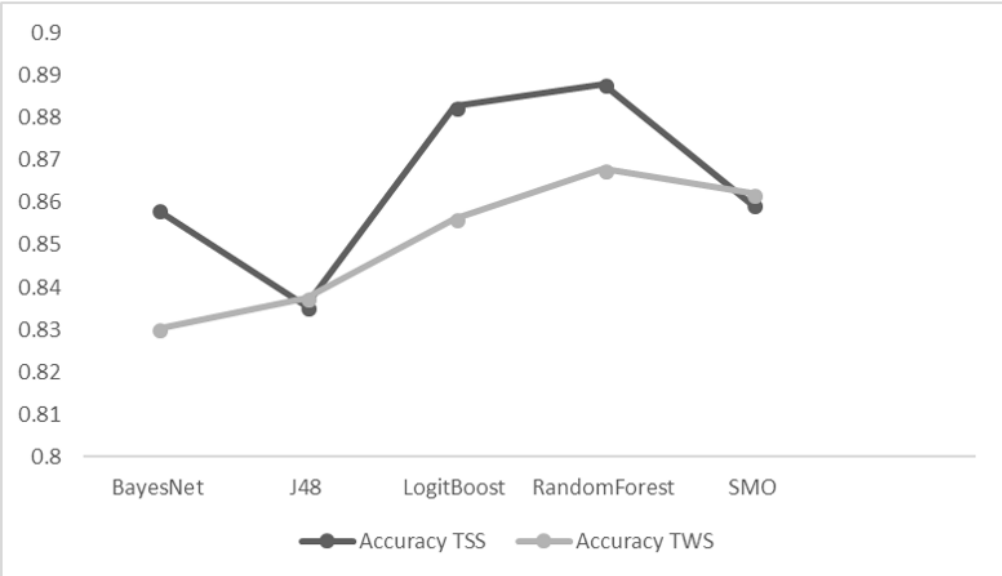

Figure 5.1: Accuracy comparison per classifier

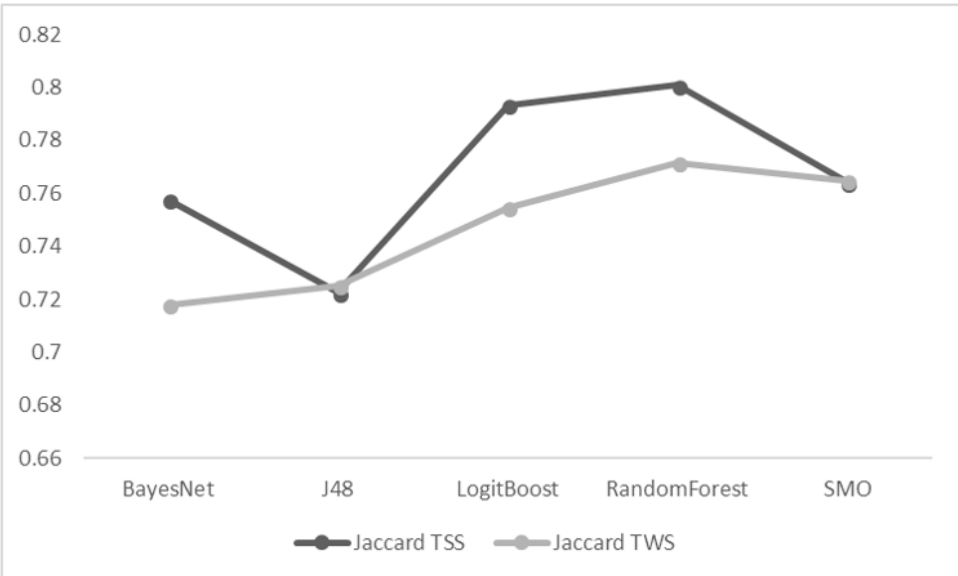

Figure 5.2: Jaccard comparison per classifier

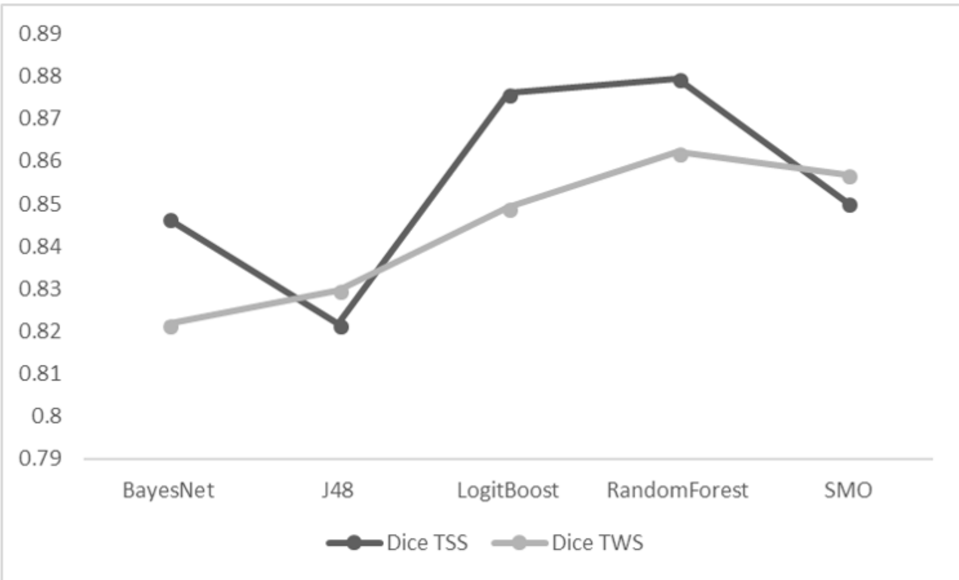

Figure 5.3: Dice comparison per classifier

CHAPTER 6

Conclusions

Throughout this document, the development of the project has been described, from the planning process to the final evaluation of the developed library. One of the first steps that was taken as part of this project was to specify the reach of the project, declaring clearly the specific objectives that wanted to be undertaken throughout its duration. As part of this initial writing the following objectives were defined: development of a library that provides classification-based segmentation through the use of superpixels, development of a graphic interface that facilitates the use of this library, and an evaluation of the developed library. As exposed throughout this document, it is safe to assume that the overall goals that were defined have been successfully met.

As a way of ensuring that the project would be developed following a structured path that would ensure that the different goals that were set would be met by the end of the project, different stages were defined, and throughout the development of the project this stages have been followed. However, it is worth noting that some of these stages have been retaken after they were supposed to have been finished, for example further background research was made even after the evaluation process had started to fulfill the knowledge base upon which the evaluation was being built on, as some of the metrics that were used had not been properly researched before, or some library development was done during the evaluation process to get information critical to the evaluation process. However, these changes to the planning did not affect the development of the project in any critical way or result in not meeting any of the specified goals.

The background research phase of the project was successful on identifying different concepts that were key in obtaining a library that produced competitive results, as shown during the evaluation process. Additionally, the different libraries and frameworks selected through this phase resulted in a complete and enabling environment in which to successfully develop the project.

Overall and as reflected in the evaluation, the development of the library is considered to have been successful. The library offers the capabilities that were set to be offered during the planning phase and the results obtained through the evaluation reflect competitive capabilities. Although the developed GUI has not been evaluated, it is considered to be successful as it offers a graphic way of interacting with the library, and the different features that were set to be offered through it have been successfully implemented. Finally, the evaluation was successful at providing representative metrics that provided proof of the competitiveness of the developed library, while also providing graphs that allowed an easier and more intuitive interpretation of the produced data.

In brief, the project is considered to have been successful on achieving the goals it was set out to achieve, and considering the results produced during the evaluation, the resulting library is considered to be competitive enough for further use.

CHAPTER 7

Further research

Although as described in the previous chapter the project was successful on its aims, some further goals have been identified throughout its development. The following are the different tasks that have been identified as possible research to be derived from the work developed in this project:

- Further development of feature extraction. Although the available features provide good information about the images that are being analyzed, further research and development could be done to integrate different libraries that could offer more features to be extracted from the images, providing more variables with which to experiment looking to improve results.

- GUI evaluation. Through the use of a GUI evaluation framework, an evaluation could be undertaken to ensure that the GUI that has been developed is usable for experts outside the context of computer science, as it has only been used by the developer of the project and the project supervisors.

- Further library evaluation. The library could be further evaluated by comparing it to other state-of-the-art image segmentation libraries. Additionally, the library itself could be further evaluated to identify optimal combinations of feature selection and classifier selection for specific tasks such as the biomedical image dataset used in this evaluation or other different tasks.

- Further documentation. The code that has been developed as part of this project has been documented through code comments, however further documentation through UML graphs, library use examples or tutorials could facilitate the use of the library and further contributions to the project.

Overall, this project has developed a base from which to further develop research into the use of superpixel segmentation using the ImageJ framework, and, through its evaluation, has proven that this approach to image segmentation can produce competitive results.

Appendixes

APPENDIX A

User's guide

The objective of this appendix is to provide a guide with which a user may be able to fully comprehend the different features that are offered by the plug-in that was developed for this project.

A.1 The library

The main functionalities of the library can be accessed through its main class Trainable-SuperpixelSegmentation, however the classes that have been developed to generate the Instances using the superpixel images for feature extraction, RegionFeatures and Region-ColorFeatures, can be independently used too.

A.1.1 TrainableSuperpixelSegmentation

This is the main class of the library, responsible for receiving input images and processing them to train classifiers, apply classifiers, generate result images and probability maps.

The class can be initialized through an empty constructor and then be populated by using getters and setters, or through a constructor that takes some variables as input. Following is a list of the different public functions provided by this library:

- `TrainableSuperpixelSegmentation(ImagePlus originalImage, ImagePlus` `labels, ArrayList<RegionFeatures.Feature> features, AbstractClassifier classifier, ArrayList<String> classes )` Creates and initializes an instance of TrainableSuperpixelSegmentation using the variables provided.

- `calculateRegionFeatures()` Calculates features for each region based on previously stablished selected features, input image, label image and class list. Returns a boolean value that checks if region features have been created.

- `getFeaturesByRegion()` Returns a String with ARFF format with the features of each region.

- `trainClassifier(ArrayList<int[]> classRegions)` Trains the classifier based on previously created features and a list of regions corresponding to classes. The classRegions variable has to have as a length the amount of classes and each int array of of ints have the indexes of labels belonging to the class indicated by its index in the ArrayList. Returns a boolean value that indicates the success of the operation.

- `trainClassifier()` Trains the current classifier based on previously loaded training data. Returns a boolean value that indicates the success of the operation.

- `applyClassifier()` Applies the current classifier to the already loaded input image and returns the resulting image as an ImagePlus.

- `applyClassifier(ImagePlus inImage, ImagePlus lbImage)` Applies the already trained classifier to the input and label images, and returns the resulting image as ImagePlus.

- `getProbabilityMap()` Applies the already trained classifier to the already loaded input image to generate a probability map stack image, where each slice of the image represents the probability map corresponding to that class.

- `addFeatures(String[] features)` Adds features to the selected feature list based on a String array.

- `Getters and setters` Together with the aforementioned methods a number of getters and setters are provided for ease of use.

A.1.2 RegionFeatures and RegionColorFeatures

RegionFeatures and RegionColorFeatures provide a way of interacting with the MorphoLibJ library and generating WEKA library-compatible objects.

RegionFeatures

Region features implements an `enum` that lists the Features that can be obtained from the MorphoLibJ Intensity Measures methods. this implementation makes it easier to add new features when said library is updated, as the dependent classes can make use of a function `getAllLabels()` which provides a String array with all labels listed. Additionally, functions to convert Features into Strings and Strings into Features are provided.

The enum `Feature` implements the following public methods:

- `toString()` Returns a String of the corresponding label.
- `getAllLabels()` Returns an array of Strings with all possible Features.
- `numFeatures()` Returns an int with the number of possible Features.
- `fromLabel(String label)` Returns the corresponding Feature of the provided label.

Additionally, the following public functions are provided:

- `calculateUnlabeledRegionFeatures( ImagePlus inputImage, ImagePlus labelImage, ArrayList<Feature> selectedFeatures, ArrayList<String> classes)`

  This function calculates the selected features of each region based on an input image and a labeled image.

- `calculateLabeledRegionFeatures( ImagePlus inputImage, ImagePlus labelImage, ImagePlus gtImage, ArrayList<Feature> selectedFeatures, ArrayList<String> classes)`

  This function calculates the selected features of each region based on an input image and a labeled image and assigns them the corresponding class feature based on a provided ground truth image.

RegionColorFeatures

RegionColorFeatures relies on RegionFeatures to implement color image feature extraction. It works by converting RGB images into Lab images and then using the three separate channels to calculate features independently, appending them with an `-L`, `-a` or `-b`. The following public functions are provided:

- `calculateUnlabeledColorFeatures(ImagePlus inputImage, ImagePlus labelImage, ArrayList<RegionFeatures.Feature> selectedFeatures, ArrayList<String> classes)` This function converts the input image into Lab and calculates the selected features of each region based on an input image and a label image.
- `calculateLabeledColorFeatures(ImagePlus inputImage, ImagePlus labelImage, ImagePlus gtImage, ArrayList<RegionFeatures.Feature> selectedFeatures, ArrayList<String> classes)` This function converts the input image into Lab and calculates the selected features of each region based on an input image and a labeled image and assigns them the corresponding class based on a provided ground-truth image.

A.2 The GUI

The GUI serves as an easy-to-access interface to use the capabilities offered by the library. As can be seen in figure A.1 the interface is separated into three distinct columns:

- The first column includes all the buttons concerning the training of the classifier and the creation of result images and probability images, and the buttons concerning the different options like the creation of a new class or the launching of the settings dialog, together with the loading and applying of a classifier.
- The second column offers the display. In this display the input image is shown, sometimes the overlay will be displayed showing the corresponding superpixel image or the resulting image. This display can be interacted with in order to select regions to add to the different classes on the third column.
- The third column includes the different classes that have been created. By default two classes will be created and more can be created through the button on the first column. After selecting regions on the display they can be added to the different

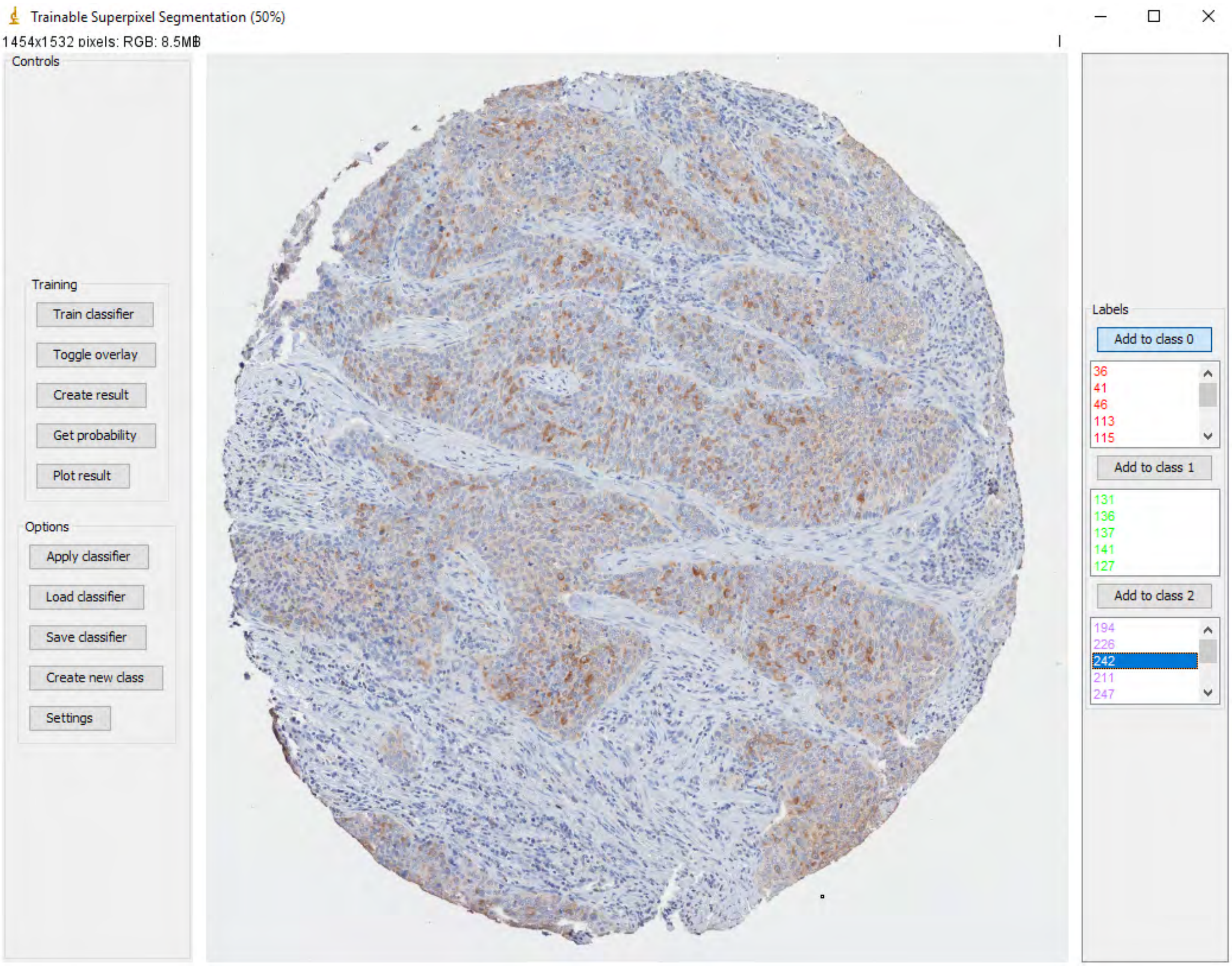

Figure A.1: GUI

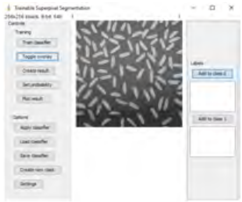

Figure A.2: No overlay

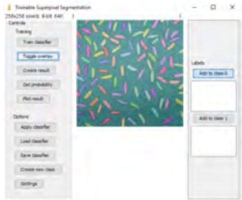

Figure A.3: Superpixel overlay

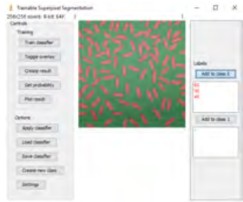

Figure A.4: Result overlay

Figure A.5: Different overlay options

classes by the use of the `Add to class` buttons. Additionally, already added selections can be displayed by selecting them on the boxes bellow the buttons and deleted by double clicking the labels.

A.2.1 The Features

The GUI offers the following features:

- Region selection

  By clicking on the displayed image multiple points can be selected, after the desired regions have been selected by clicking the `Add to class` button the selected regions can be added to the selected class. Additionally, the selected regions can be displayed again by selecting them from the region list bellow the class and can be deleted by double-clicking.

- Train classifier:

  A classifier can be trained by pressing the `Train classifier` button, this will train the classifier that can be specified in the `Settings` dialog based on the regions that have been added to the classes.

- Toggle overlay

  The input image will rotate over three different states of overlay display: No overlay, superpixel image overlay and result overlay. In the case that a result has not been calculated yet, only the first two states will be cycled through. The different overlay options are shown in Figure A.5

- Create result

If a result has already been created then a duplicate of said result will be displayed on a new image, if it hasn't then a new result will be calculated and then displayed.

- Get probability

  Using the trained classifier, probability maps will be calculated for each class and a layered image will be created, with each layer representing a probability map for its corresponding class.

- Plot result

  If a classifier has been trained, it will provide a statistics window provided by the WEKA library.

- Apply classifier

  If a classifier has been loaded or trained, it will be applied to the image, if it hasn't then a new classifier will be trained based on the settings.

- Load classifier

  A dialog will be created offering the option to load a WEKA classifier. These classifiers are stored in `.model` files. The program will check the classifier to look for the number of classes that the classifier has been trained with and will update the GUI accordingly.

- Save classifier

  A dialog will be created offering the option to save the current WEKA classifier. These classifiers are stored in `.model` files. This classifiers can then be taken into other programs that implement WEKA or into the WEKA workbench itself for further inspection.

- Create new class

  A dialog will appear asking for the name of a new class, and the newly created class will appear together with the default two classes on the third column.

- Settings

  A settings dialog will appear. The settings dialog will offer the option to change the selected features, the opacity of the overlay and the used WEKA classifier. The features are displayed as a list of check boxes that can be selected or deselected to indicate whether they should be used in the calculation of the features. The overlay opacity can be selected either through a slider or through an input box where a value can be selected from 0 to 1, where 0.33 is the default value. Finally, a WEKA classifier can be chosen and its options changed by clicking on the classifier itself. Figure A.6 shows the settings window.

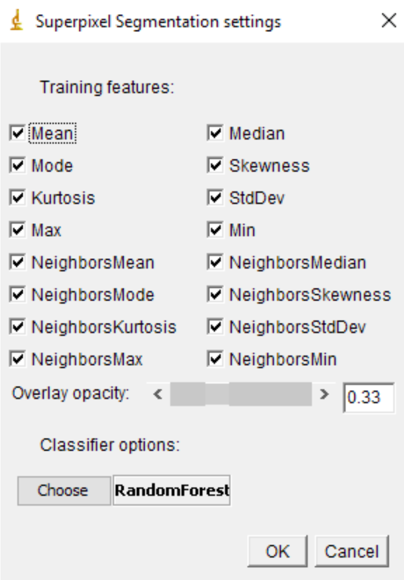

Figure A.6: Settings dialog

A.3 Contributing

This plug-in has been developed using open source libraries and has been published as an open source plug-in. As such, it is open for modifications and contributions in the following Git-hub repository:

https://github.com/96jsalinas/Trainable_Superpixel_Segmentation

Contributions to this project can be done through pull requests and derivative work can be done by creating new projects based on the code developed for this project. For possible contributors the following resources are worth looking at:

- ImageJ Developer Resources: https://imagej.nih.gov/ij/developer/index.html
- Developing plugins for ImageJ: https://imagej.net/Writing_plugins

- Fiji homepage: https://fiji.sc/
- WEKA homepage: https://www.cs.waikato.ac.nz/ml/weka/

APPENDIX B

Evaluation Results

This appendix presents the whole results generated by the evaluation. Firstly the resulting images of the training are presented and are followed by tables representing the generated statistical results.

B.1 Image Results

Figures B.4 and B.7 show the resulting images from the evaluation using Trainable Superpixel Segmentation together with the original images, while figures **??** and **??** show the results from the evaluation using Trainable Weka Segmentation.

B.2 Table Results

This section presents the resulting tables from the evaluation.

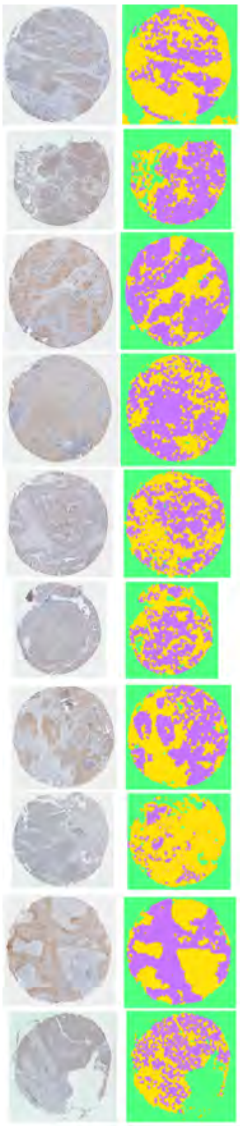

Figure B.1: TSS J48 results

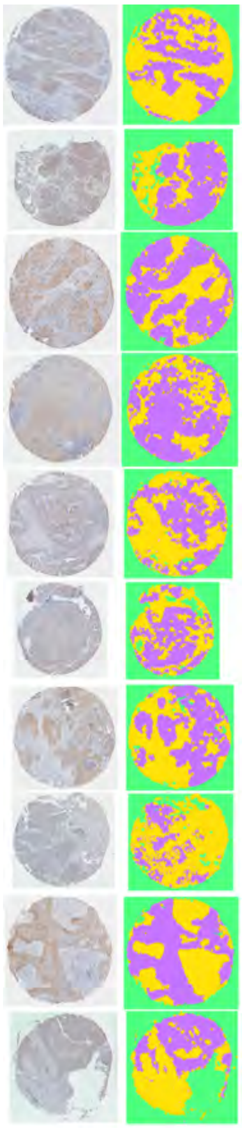

Figure B.2: TSS LogitBoost results

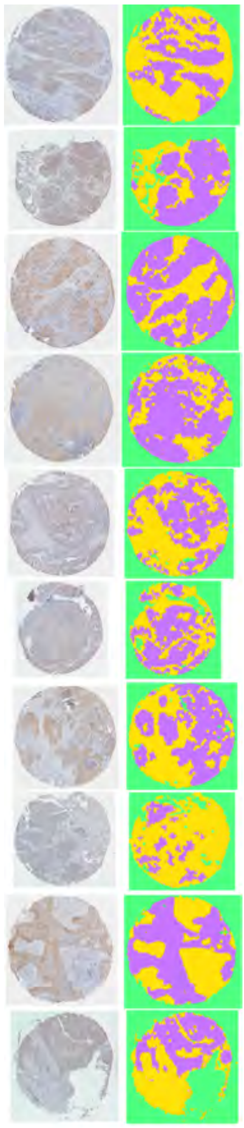

Figure B.3: TSS RandomForest results

Figure B.4: TSS evaluation results for J48, LogitBoost and RandomForest

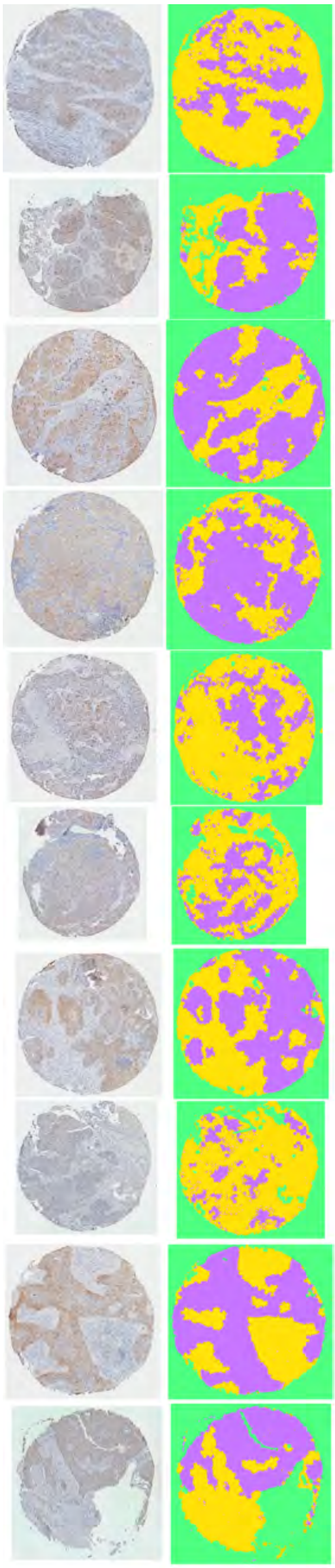

Figure B.5: TSS SMO results

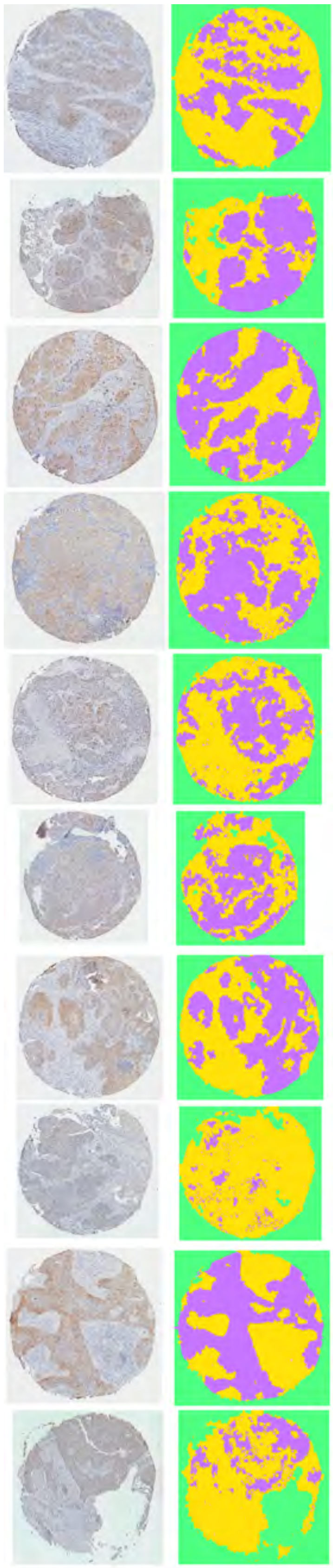

Figure B.6: TSS BayesNet results

Figure B.7: TSS evaluation results for SMO and BayesNet

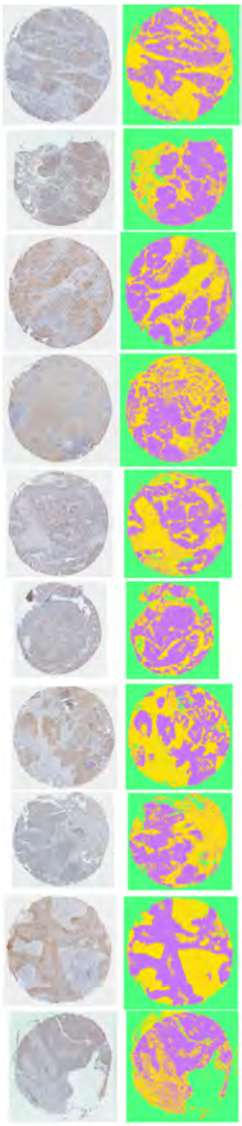

Figure B.8: TWS J48 results

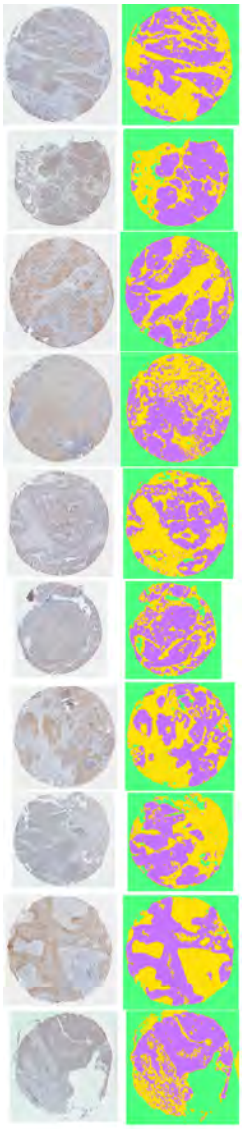

Figure B.9: TWS Logit-Boost results

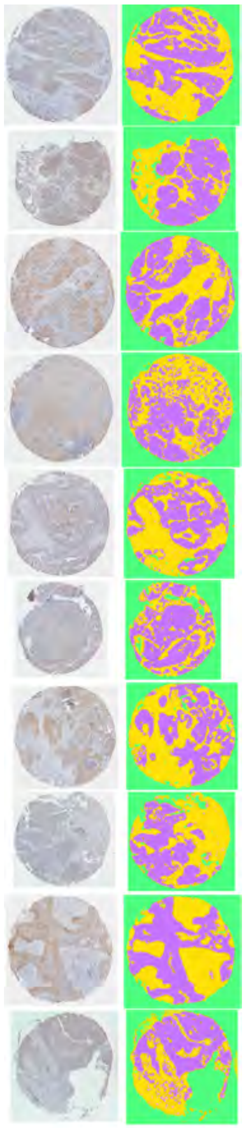

Figure B.10: TWS Random-Forest results

Figure B.11: TWS evaluation results for J48, LogitBoost and RandomForest

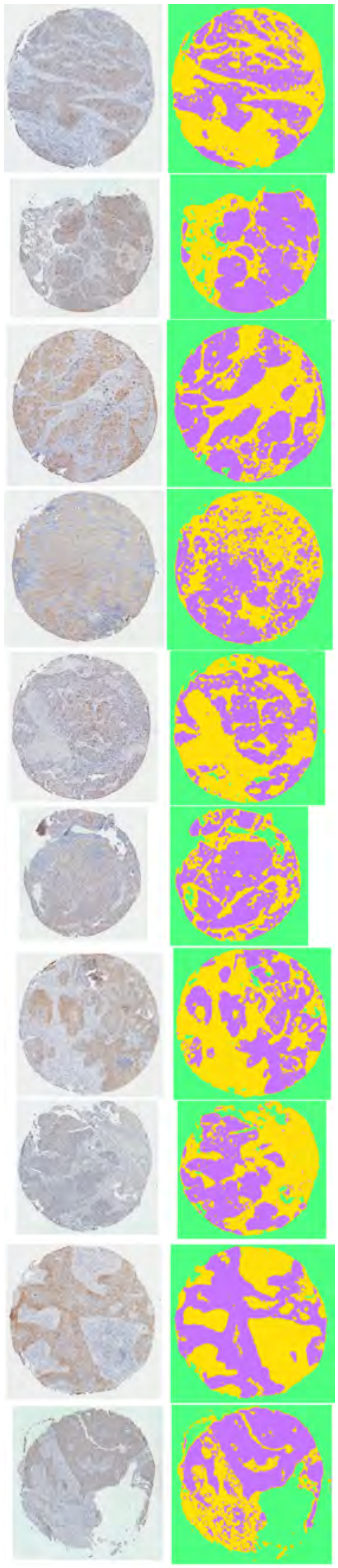

Figure B.12: TWS SMO results

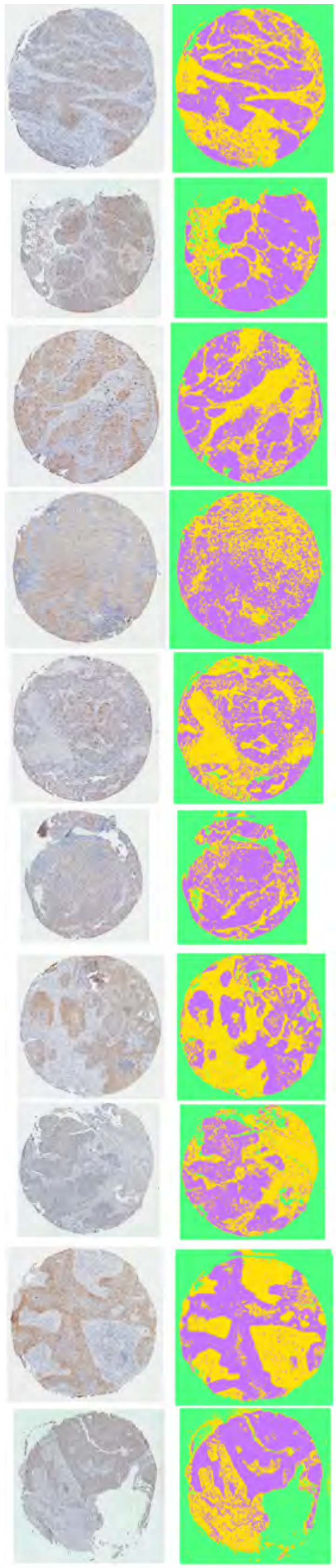

Figure B.13: TWS BayesNet results

Figure B.14: TWS evaluation results for SMO and BayesNet

Table B.1: TSS BayesNet statistics folds 1-5

BayesNet						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	664717	862	799	0.998
		Predicted tumoral	0	555729	10693	0.981
		Predicted nontumoral	21931	261626	711171	0.715
		Recall	0.968	0.679	0.984	0.867
	Accuracy:	0.867				
	Dice:	Label	DiceCoefficient			
		1	0.983			
		2	0.803			
		3	0.828			
	Jaccard:	Label	JaccardIndex			
		1	0.966			
		2	0.67			
		3	0.707			
	Training time:	876				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	709987	0	15843	0.978
		Predicted tumoral	0	538435	52450	0.911
		Predicted nontumoral	20844	51293	363096	0.834
		Recall	0.971	0.913	0.842	0.92
	Accuracy:	0.92				
	Dice:	Label	DiceCoefficient			
		1	0.975			
		2	0.912			
		3	0.838			
	Jaccard:	Label	JaccardIndex			
		1	0.951			
		2	0.838			
		3	0.721			
	Training time:	669				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	676854	1632	9	0.998
		Predicted tumoral	643	736347	184469	0.799
		Predicted nontumoral	11851	16382	490788	0.946
		Recall	0.982	0.976	0.727	0.899
	Accuracy:	0.899				
	Dice:	Label	DiceCoefficient			
		1	0.99			
		2	0.879			
		3	0.822			
	Jaccard:	Label	JaccardIndex			
		1	0.98			
		2	0.784			
		3	0.698			
	Training time:	460				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	709094	0	0	1
		Predicted tumoral	0	630023	118475	0.842
		Predicted nontumoral	9333	233738	404482	0.625
		Recall	0.987	0.729	0.773	0.828
	Accuracy:	0.828				
	Dice:	Label	DiceCoefficient			
		1	0.993			
		2	0.782			
		3	0.691			
	Jaccard:	Label	JaccardIndex			
		1	0.987			
		2	0.641			
		3	0.528			
	Training time:	459				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	537287	67	988	0.998
		Predicted tumoral	0	504712	47129	0.915
		Predicted nontumoral	14241	208599	554209	0.713
		Recall	0.974	0.707	0.92	0.855
	Accuracy:	0.855				
	Dice:	Label	DiceCoefficient			
		1	0.986			
		2	0.798			
		3	0.804			
	Jaccard:	Label	JaccardIndex			
		1	0.972			
		2	0.664			
		3	0.672			
	Training time:	451				

Table B.2: TSS BayesNet statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	607692	0	27	1
		Predicted tumoral	0	519280	48936	0.914
		Predicted nontumoral	61741	173590	295076	0.556
		Recall	0.908	0.749	0.858	0.833
	Accuracy:	0.833				
	Dice:	Label	DiceCoefficient			
		1	0.952			
		2	0.824			
		3	0.675			
	Jaccard:	Label	JaccardIndex			
		1	0.908			
		2	0.7			
		3	0.509			
	Training time:	469				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	547316	605	194	0.999
		Predicted tumoral	2398	682757	73512	0.9
		Predicted nontumoral	19557	120011	653187	0.824
		Recall	0.961	0.85	0.899	0.897
	Accuracy:	0.897				
	Dice:	Label	DiceCoefficient			
		1	0.98			
		2	0.874			
		3	0.86			
	Jaccard:	Label	JaccardIndex			
		1	0.96			
		2	0.776			
		3	0.754			
	Training time:	441				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	596686	1047	180	0.998
		Predicted tumoral	0	94040	16066	0.854
		Predicted nontumoral	54808	485483	639300	0.542
		Recall	0.916	0.162	0.975	0.705
	Accuracy:	0.705				
	Dice:	Label	DiceCoefficient			
		1	0.955			
		2	0.272			
		3	0.697			
	Jaccard:	Label	JaccardIndex			
		1	0.914			
		2	0.158			
		3	0.535			
	Training time:	446				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	704992	988	32	0.999
		Predicted tumoral	1270	904093	34291	0.962
		Predicted nontumoral	14358	75595	651406	0.879
		Recall	0.978	0.922	0.95	0.947
	Accuracy:	0.947				
	Dice:	Label	DiceCoefficient			
		1	0.988			
		2	0.942			
		3	0.913			
	Jaccard:	Label	JaccardIndex			
		1	0.977			
		2	0.89			
		3	0.84			
	Training time:	450				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	1019003	2204	206	0.998
		Predicted tumoral	0	296127	3987	0.987
		Predicted nontumoral	95124	284665	595152	0.61
		Recall	0.915	0.508	0.993	0.832
	Accuracy:	0.832				
	Dice:	Label	DiceCoefficient			
		1	0.954			
		2	0.671			
		3	0.756			
	Jaccard:	Label	JaccardIndex			
		1	0.913			
		2	0.504			
		3	0.608			
	Training time:	416				

Table B.3: TSS J48 statistics folds 1-5

J48						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	608096	2621	738	0.995
		Predicted tumoral	0	559745	35767	0.94
		Predicted nontumoral	78552	255851	686158	0.672
		Recall	0.886	0.684	0.949	0.832
	Accuracy:	0.832				
	Dice:	Label	DiceCoefficient			
		1	0.937			
		2	0.792			
		3	0.787			
	Jaccard:	Label	JaccardIndex			
		1	0.881			
		2	0.655			
		3	0.649			
	Training time:	1963				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	724219	0	52140	0.933
		Predicted tumoral	510	511219	118690	0.811
		Predicted nontumoral	6102	78509	260559	0.755
		Recall	0.991	0.867	0.604	0.854
	Accuracy:	0.854				
	Dice:	Label	DiceCoefficient			
		1	0.961			
		2	0.838			
		3	0.671			
	Jaccard:	Label	JaccardIndex			
		1	0.925			
		2	0.721			
		3	0.505			
	Training time:	1767				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	675153	2509	8433	0.984
		Predicted tumoral	4572	697571	238881	0.741
		Predicted nontumoral	9623	54281	427952	0.87
		Recall	0.979	0.925	0.634	0.85
	Accuracy:	0.85				
	Dice:	Label	DiceCoefficient			
		1	0.982			
		2	0.823			
		3	0.733			
	Jaccard:	Label	JaccardIndex			
		1	0.964			
		2	0.699			
		3	0.579			
	Training time:	1889				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	715270	493	2479	0.996
		Predicted tumoral	0	685128	202042	0.772
		Predicted nontumoral	3157	178140	318436	0.637
		Recall	0.996	0.793	0.609	0.816
	Accuracy:	0.816				
	Dice:	Label	DiceCoefficient			
		1	0.996			
		2	0.783			
		3	0.623			
	Jaccard:	Label	JaccardIndex			
		1	0.992			
		2	0.643			
		3	0.452			
	Training time:	1668				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	537656	126	5753	0.989
		Predicted tumoral	1731	495700	115687	0.808
		Predicted nontumoral	12141	217552	480886	0.677
		Recall	0.975	0.695	0.798	0.811
	Accuracy:	0.811				
	Dice:	Label	DiceCoefficient			
		1	0.982			
		2	0.747			
		3	0.733			
	Jaccard:	Label	JaccardIndex			
		1	0.965			
		2	0.597			
		3	0.578			
	Training time:	1829				

Table B.4: TSS J48 statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	644578	843	3055	0.994
		Predicted tumoral	1287	494407	42031	0.919
		Predicted nontumoral	23568	197620	298953	0.575
		Recall	0.963	0.714	0.869	0.843
	Accuracy:					
	Dice:	Label	DiceCoefficient			
		1	0.978			
		2	0.804			
		3	0.692			
	Jaccard:	Label	JaccardIndex			
		1	0.957			
		2	0.672			
		3	0.529			
	Training time:	1693				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	557115	6801	10020	0.971
		Predicted tumoral	0	694086	104129	0.87
		Predicted nontumoral	12156	102486	612744	0.842
		Recall	0.979	0.864	0.843	0.888
	Accuracy:	0.888				
	Dice:	Label	DiceCoefficient			
		1	0.975			
		2	0.867			
		3	0.843			
	Jaccard:	Label	JaccardIndex			
		1	0.951			
		2	0.765			
		3	0.728			
	Training time:	1720				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	635476	5493	13004	0.972
		Predicted tumoral	2567	149664	35794	0.796
		Predicted nontumoral	13451	425413	606748	0.58
		Recall	0.975	0.258	0.926	0.737
	Accuracy:	0.737				
	Dice:	Label	DiceCoefficient			
		1	0.974			
		2	0.389			
		3	0.713			
	Jaccard:	Label	JaccardIndex			
		1	0.948			
		2	0.242			
		3	0.554			
	Training time:	1752				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	716433	6192	2158	0.988
		Predicted tumoral	0	890699	33104	0.964
		Predicted nontumoral	4187	83785	650467	0.881
		Recall	0.994	0.908	0.949	0.946
	Accuracy:	0.946				
	Dice:	Label	DiceCoefficient			
		1	0.991			
		2	0.935			
		3	0.913			
	Jaccard:	Label	JaccardIndex			
		1	0.983			
		2	0.879			
		3	0.841			
	Training time:	1784				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	1089412	7787	24865	0.971
		Predicted tumoral	6776	365597	223549	0.613
		Predicted nontumoral	17939	209612	350931	0.607
		Recall	0.978	0.627	0.586	0.786
	Accuracy:	0.786				
	Dice:	Label	DiceCoefficient			
		1	0.974			
		2	0.62			
		3	0.596			
	Jaccard:	Label	JaccardIndex			
		1	0.95			
		2	0.45			
		3	0.424			
	Training time:	1686				

Table B.5: TSS LogitBoost statistics folds 1-5

LogitBoost						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	682137	2666	2174	0.993
		Predicted tumoral	0	586148	16081	0.973
		Predicted nontumoral	4511	229403	704408	0.751
		Recall	0.993	0.716	0.975	0.886
	Accuracy:	0.886				
	Dice:	Label	DiceCoefficient			
		1	0.993			
		2	0.825			
		3	0.848			
	Jaccard:	Label	JaccardIndex			
		1	0.986			
		2	0.703			
		3	0.736			
	Training time:	3502				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	723439	0	34095	0.955
		Predicted tumoral	510	535471	93904	0.85
		Predicted nontumoral	6882	54257	303390	0.832
		Recall	0.99	0.908	0.703	0.892
	Accuracy:	0.892				
	Dice:	Label	DiceCoefficient			
		1	0.972			
		2	0.878			
		3	0.762			
	Jaccard:	Label	JaccardIndex			
		1	0.946			
		2	0.783			
		3	0.616			
	Training time:	3387				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	685377	2214	1390	0.995
		Predicted tumoral	0	723583	189774	0.792
		Predicted nontumoral	3971	28564	484102	0.937
		Recall	0.994	0.959	0.717	0.893
	Accuracy:	0.893				
	Dice:	Label	DiceCoefficient			
		1	0.995			
		2	0.868			
		3	0.812			
	Jaccard:	Label	JaccardIndex			
		1	0.989			
		2	0.766			
		3	0.684			
	Training time:	3358				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	714724	630	2473	0.996
		Predicted tumoral	0	716241	173654	0.805
		Predicted nontumoral	3703	146890	346830	0.697
		Recall	0.995	0.829	0.663	0.845
	Accuracy:	0.845				
	Dice:	Label	DiceCoefficient			
		1	0.995			
		2	0.817			
		3	0.68			
	Jaccard:	Label	JaccardIndex			
		1	0.991			
		2	0.69			
		3	0.515			
	Training time:	3315				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	550937	928	6204	0.987
		Predicted tumoral	0	523084	67811	0.885
		Predicted nontumoral	591	189366	528311	0.736
		Recall	0.999	0.733	0.877	0.858
	Accuracy:	0.858				
	Dice:	Label	DiceCoefficient			
		1	0.993			
		2	0.802			
		3	0.8			
	Jaccard:	Label	JaccardIndex			
		1	0.986			
		2	0.67			
		3	0.667			
	Training time:	3343				

Table B.6: TSS LogitBoost statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	643704	29	52	1
		Predicted tumoral	2399	545896	42139	0.925
		Predicted nontumoral	23330	146945	301848	0.639
		Recall	0.962	0.788	0.877	0.874
	Accuracy:	0.874				
	Dice:	Label	DiceCoefficient			
		1	0.98			
		2	0.851			
		3	0.74			
	Jaccard:	Label	JaccardIndex			
		1	0.961			
		2	0.74			
		3	0.587			
	Training time:	3281				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	563155	9510	8737	0.969
		Predicted tumoral	0	684178	91838	0.882
		Predicted nontumoral	6116	109685	626318	0.844
		Recall	0.989	0.852	0.862	0.892
	Accuracy:	0.892				
	Dice:	Label	DiceCoefficient			
		1	0.979			
		2	0.866			
		3	0.853			
	Jaccard:	Label	JaccardIndex			
		1	0.959			
		2	0.764			
		3	0.743			
	Training time:	3394				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	636647	3306	9158	0.981
		Predicted tumoral	2	273802	47030	0.853
		Predicted nontumoral	14845	303462	599358	0.653
		Recall	0.977	0.472	0.914	0.8
	Accuracy:	0.8				
	Dice:	Label	DiceCoefficient			
		1	0.979			
		2	0.608			
		3	0.762			
	Jaccard:	Label	JaccardIndex			
		1	0.959			
		2	0.436			
		3	0.615			
	Training time:	3332				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	719608	5742	3495	0.987
		Predicted tumoral	0	899454	30597	0.967
		Predicted nontumoral	1012	75480	651637	0.895
		Recall	0.999	0.917	0.95	0.951
	Accuracy:	0.951				
	Dice:	Label	DiceCoefficient			
		1	0.993			
		2	0.941			
		3	0.922			
	Jaccard:	Label	JaccardIndex			
		1	0.986			
		2	0.889			
		3	0.855			
	Training time:	3340				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	1072605	5798	5271	0.99
		Predicted tumoral	13992	511043	31649	0.918
		Predicted nontumoral	27530	66155	562425	0.857
		Recall	0.963	0.877	0.938	0.935
	Accuracy:	0.935				
	Dice:	Label	DiceCoefficient			
		1	0.976			
		2	0.897			
		3	0.896			
	Jaccard:	Label	JaccardIndex			
		1	0.953			
		2	0.813			
		3	0.812			
	Training time:	3278				

Table B.7: TSS RandomForest statistics folds 1-5

RandomForest						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	683957	1905	716	0.996
		Predicted tumoral	0	578439	9386	0.984
		Predicted nontumoral	2691	237873	712561	0.748
		Recall	0.996	0.707	0.986	0.887
	Accuracy:	0.887				
	Dice:	Label	DiceCoefficient			
		1	0.996			
		2	0.823			
		3	0.85			
	Jaccard:	Label	JaccardIndex			
		1	0.992			
		2	0.699			
		3	0.74			
	Training time:	16937				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	727527	0	39920	0.948
		Predicted tumoral	510	552845	70253	0.887
		Predicted nontumoral	2794	36883	321216	0.89
		Recall	0.995	0.937	0.745	0.914
	Accuracy:	0.914				
	Dice:	Label	DiceCoefficient			
		1	0.971			
		2	0.911			
		3	0.811			
	Jaccard:	Label	JaccardIndex			
		1	0.944			
		2	0.837			
		3	0.682			
	Training time:	15086				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	679081	1672	329	0.997
		Predicted tumoral	3315	741072	224170	0.765
		Predicted nontumoral	6952	11617	450767	0.96
		Recall	0.985	0.982	0.668	0.883
	Accuracy:	0.883				
	Dice:	Label	DiceCoefficient			
		1	0.991			
		2	0.86			
		3	0.788			
	Jaccard:	Label	JaccardIndex			
		1	0.982			
		2	0.755			
		3	0.65			
	Training time:	14638				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	715826	493	2041	0.996
		Predicted tumoral	0	750330	182932	0.804
		Predicted nontumoral	2601	112938	337984	0.745
		Recall	0.996	0.869	0.646	0.857
	Accuracy:	0.857				
	Dice:	Label	DiceCoefficient			
		1	0.996			
		2	0.835			
		3	0.692			
	Jaccard:	Label	JaccardIndex			
		1	0.993			
		2	0.717			
		3	0.529			
	Training time:	14702				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	550353	677	5435	0.989
		Predicted tumoral	0	547931	50601	0.915
		Predicted nontumoral	1175	164770	546290	0.767
		Recall	0.998	0.768	0.907	0.881
	Accuracy:	0.881				
	Dice:	Label	DiceCoefficient			
		1	0.993			
		2	0.835			
		3	0.831			
	Jaccard:	Label	JaccardIndex			
		1	0.987			
		2	0.717			
		3	0.711			
	Training time:	14327				

Table B.8: TSS RandomForest statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	652731	0	461	0.999
		Predicted tumoral	0	549061	32124	0.945
		Predicted nontumoral	16702	143809	311454	0.66
		Recall	0.975	0.792	0.905	0.887
	Accuracy:	0.887				
	Dice:	Label	DiceCoefficient			
		1	0.987			
		2	0.862			
		3	0.763			
	Jaccard:	Label	JaccardIndex			
		1	0.974			
		2	0.757			
		3	0.617			
	Training time:	14291				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	560789	7392	5862	0.977
		Predicted tumoral	0	733804	83983	0.897
		Predicted nontumoral	8482	62177	637048	0.9
		Recall	0.985	0.913	0.876	0.92
	Accuracy:	0.92				
	Dice:	Label	DiceCoefficient			
		1	0.981			
		2	0.905			
		3	0.888			
	Jaccard:	Label	JaccardIndex			
		1	0.963			
		2	0.827			
		3	0.799			
	Training time:	14485				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	632390	2105	6774	0.986
		Predicted tumoral	0	201816	33780	0.857
		Predicted nontumoral	19104	376649	614992	0.608
		Recall	0.971	0.348	0.938	0.768
	Accuracy:	0.768				
	Dice:	Label	DiceCoefficient			
		1	0.978			
		2	0.495			
		3	0.738			
	Jaccard:	Label	JaccardIndex			
		1	0.958			
		2	0.329			
		3	0.585			
	Training time:	14366				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	717562	6515	1291	0.989
		Predicted tumoral	0	908446	22107	0.976
		Predicted nontumoral	3058	65715	662331	0.906
		Recall	0.996	0.926	0.966	0.959
	Accuracy:	0.959				
	Dice:	Label	DiceCoefficient			
		1	0.992			
		2	0.951			
		3	0.935			
	Jaccard:	Label	JaccardIndex			
		1	0.985			
		2	0.906			
		3	0.878			
	Training time:	14873				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	1079960	5789	6239	0.989
		Predicted tumoral	0	492912	43655	0.919
		Predicted nontumoral	34167	84295	549451	0.823
		Recall	0.969	0.845	0.917	0.924
	Accuracy:	0.924				
	Dice:	Label	DiceCoefficient			
		1	0.979			
		2	0.881			
		3	0.867			
	Jaccard:	Label	JaccardIndex			
		1	0.959			
		2	0.787			
		3	0.765			
	Training time:	14076				

Table B.9: TSS SMO statistics folds 1-5

SMO						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	685632	1905	747	0.996
		Predicted tumoral	0	440245	10008	0.978
		Predicted nontumoral	1016	376067	711908	0.654
		Recall	0.999	0.538	0.985	0.825
	Accuracy:	0.825				
	Dice:	Label	DiceCoefficient			
		1	0.997			
		2	0.694			
		3	0.786			
	Jaccard:	Label	JaccardIndex			
		1	0.995			
		2	0.532			
		3	0.647			
	Training time:	5280				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	727467	0	41638	0.946
		Predicted tumoral	0	535337	74548	0.878
		Predicted nontumoral	3364	54391	315203	0.845
		Recall	0.995	0.908	0.731	0.901
	Accuracy:	0.901				
	Dice:	Label	DiceCoefficient			
		1	0.97			
		2	0.893			
		3	0.784			
	Jaccard:	Label	JaccardIndex			
		1	0.942			
		2	0.806			
		3	0.644			
	Training time:	4659				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	680375	1632	7838	0.986
		Predicted tumoral	5002	736506	216923	0.768
		Predicted nontumoral	3971	16223	450505	0.957
		Recall	0.987	0.976	0.667	0.881
	Accuracy:	0.881				
	Dice:	Label	DiceCoefficient			
		1	0.987			
		2	0.86			
		3	0.786			
	Jaccard:	Label	JaccardIndex			
		1	0.974			
		2	0.754			
		3	0.648			
	Training time:	4447				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	715826	497	2547	0.996
		Predicted tumoral	0	730731	173578	0.808
		Predicted nontumoral	2601	132533	346832	0.72
		Recall	0.996	0.846	0.663	0.852
	Accuracy:	0.852				
	Dice:	Label	DiceCoefficient			
		1	0.996			
		2	0.827			
		3	0.69			
	Jaccard:	Label	JaccardIndex			
		1	0.992			
		2	0.704			
		3	0.527			
	Training time:	4323				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	548530	59	2437	0.995
		Predicted tumoral	0	374128	25515	0.936
		Predicted nontumoral	2998	339191	574374	0.627
		Recall	0.995	0.524	0.954	0.802
	Accuracy:	0.802				
	Dice:	Label	DiceCoefficient			
		1	0.995			
		2	0.672			
		3	0.756			
	Jaccard:	Label	JaccardIndex			
		1	0.99			
		2	0.506			
		3	0.608			
	Training time:	4903				

Table B.10: TSS SMO statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	649258	0	12	1
		Predicted tumoral	0	371478	22844	0.942
		Predicted nontumoral	20175	321392	321183	0.485
		Recall	0.97	0.536	0.934	0.786
	Accuracy:	0.786				
	Dice:	Label	DiceCoefficient			
		1	0.985			
		2	0.683			
		3	0.638			
	Jaccard:	Label	JaccardIndex			
		1	0.97			
		2	0.519			
		3	0.468			
	Training time:	4424				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	556321	3981	6766	0.981
		Predicted tumoral	5640	733827	104581	0.869
		Predicted nontumoral	7310	65565	615546	0.894
		Recall	0.977	0.913	0.847	0.908
	Accuracy:	0.908				
	Dice:	Label	DiceCoefficient			
		1	0.979			
		2	0.891			
		3	0.87			
	Jaccard:	Label	JaccardIndex			
		1	0.959			
		2	0.803			
		3	0.77			
	Training time:	4556				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	641794	2296	9396	0.982
		Predicted tumoral	0	161666	54902	0.746
		Predicted nontumoral	9700	416608	591248	0.581
		Recall	0.985	0.278	0.902	0.739
	Accuracy:	0.739				
	Dice:	Label	DiceCoefficient			
		1	0.984			
		2	0.406			
		3	0.707			
	Jaccard:	Label	JaccardIndex			
		1	0.968			
		2	0.254			
		3	0.547			
	Training time:	4667				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	718571	6643	1447	0.989
		Predicted tumoral	0	929160	60807	0.939
		Predicted nontumoral	2049	44873	623475	0.93
		Recall	0.997	0.947	0.909	0.951
	Accuracy:	0.951				
	Dice:	Label	DiceCoefficient			
		1	0.993			
		2	0.943			
		3	0.919			
	Jaccard:	Label	JaccardIndex			
		1	0.986			
		2	0.892			
		3	0.851			
	Training time:	4911				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	1101361	6608	7416	0.987
		Predicted tumoral	1756	555615	66028	0.891
		Predicted nontumoral	11010	20773	525901	0.943
		Recall	0.989	0.953	0.877	0.951
	Accuracy:	0.951				
	Dice:	Label	DiceCoefficient			
		1	0.988			
		2	0.921			
		3	0.909			
	Jaccard:	Label	JaccardIndex			
		1	0.976			
		2	0.854			
		3	0.833			
	Training time:	4743				

Table B.11: TWS BayesNet statistics folds 1-5

BayesNet						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	668133	1565	3834	0.992
		Predicted tumoral	1453	674207	149414	0.817
		Predicted nontumoral	17062	142445	569415	0.781
		Recall	0.973	0.824	0.788	0.858
	Accuracy:	0.858				
	Dice:	Label	DiceCoefficient			
		1	0.982			
		2	0.821			
		3	0.785			
	Jaccard:	Label	JaccardIndex			
		1	0.965			
		2	0.696			
		3	0.645			
	Training time:	767				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	691752	0	23715	0.967
		Predicted tumoral	2243	563983	121419	0.82
		Predicted nontumoral	36836	25745	286255	0.821
		Recall	0.947	0.956	0.664	0.88
	Accuracy:	0.88				
	Dice:	Label	DiceCoefficient			
		1	0.957			
		2	0.883			
		3	0.734			
	Jaccard:	Label	JaccardIndex			
		1	0.917			
		2	0.791			
		3	0.579			
	Training time:	464				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	671694	1902	2893	0.993
		Predicted tumoral	2180	651127	189581	0.772
		Predicted nontumoral	15474	101332	482792	0.805
		Recall	0.974	0.863	0.715	0.852
	Accuracy:	0.852				
	Dice:	Label	DiceCoefficient			
		1	0.984			
		2	0.815			
		3	0.757			
	Jaccard:	Label	JaccardIndex			
		1	0.968			
		2	0.688			
		3	0.61			
	Training time:	406				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	699151	1140	2852	0.994
		Predicted tumoral	2070	549704	272232	0.667
		Predicted nontumoral	17206	312917	247873	0.429
		Recall	0.973	0.636	0.474	0.711
	Accuracy:	0.711				
	Dice:	Label	DiceCoefficient			
		1	0.984			
		2	0.651			
		3	0.45			
	Jaccard:	Label	JaccardIndex			
		1	0.968			
		2	0.483			
		3	0.291			
	Training time:	403				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	534158	2063	10153	0.978
		Predicted tumoral	1591	515312	118117	0.811
		Predicted nontumoral	15779	196003	474056	0.691
		Recall	0.969	0.722	0.787	0.816
	Accuracy:	0.816				
	Dice:	Label	DiceCoefficient			
		1	0.973			
		2	0.764			
		3	0.736			
	Jaccard:	Label	JaccardIndex			
		1	0.948			
		2	0.619			
		3	0.582			
	Training time:	404				

Table B.12: TWS BayesNet statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	621355	799	1393	0.996
		Predicted tumoral	5114	579982	173494	0.765
		Predicted nontumoral	42964	112089	169152	0.522
		Recall	0.928	0.837	0.492	0.803
	Accuracy:	0.803				
	Dice:	Label	DiceCoefficient			
		1	0.961			
		2	0.799			
		3	0.506			
	Jaccard:	Label	JaccardIndex			
		1	0.925			
		2	0.666			
		3	0.339			
	Training time:	395				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	547711	7916	20766	0.95
		Predicted tumoral	1307	600377	106726	0.847
		Predicted nontumoral	20253	195080	599401	0.736
		Recall	0.962	0.747	0.825	0.832
	Accuracy:	0.832				
	Dice:	Label	DiceCoefficient			
		1	0.956			
		2	0.794			
		3	0.778			
	Jaccard:	Label	JaccardIndex			
		1	0.916			
		2	0.659			
		3	0.636			
	Training time:	449				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	597019	3846	16093	0.968
		Predicted tumoral	4054	463890	149795	0.751
		Predicted nontumoral	50421	112834	489658	0.75
		Recall	0.916	0.799	0.747	0.821
	Accuracy:	0.821				
	Dice:	Label	DiceCoefficient			
		1	0.941			
		2	0.774			
		3	0.748			
	Jaccard:	Label	JaccardIndex			
		1	0.889			
		2	0.632			
		3	0.598			
	Training time:	407				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	699925	4762	5487	0.986
		Predicted tumoral	694	794645	52859	0.937
		Predicted nontumoral	20001	181269	627383	0.757
		Recall	0.971	0.81	0.915	0.889
	Accuracy:	0.889				
	Dice:	Label	DiceCoefficient			
		1	0.978			
		2	0.869			
		3	0.829			
	Jaccard:	Label	JaccardIndex			
		1	0.958			
		2	0.768			
		3	0.707			
	Training time:	399				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumoral	Groundtruth nontumoral	Precision
		Predicted background	989787	4106	6609	0.989
		Predicted tumoral	9700	540533	194179	0.726
		Predicted nontumoral	114640	38357	398557	0.723
		Recall	0.888	0.927	0.665	0.84
	Accuracy:	0.84				
	Dice:	Label	DiceCoefficient			
		1	0.936			
		2	0.814			
		3	0.693			
	Jaccard:	Label	JaccardIndex			
		1	0.88			
		2	0.687			
		3	0.53			
	Training time:	441				

Table B.13: TWS J48 statistics folds 1-5

J48					
1	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		675907	4524	15579	0.971
		1859	603586	107019	0.847
		8882	210107	600065	0.733
		0.984	0.738	0.83	0.844
	Accuracy:				
	Dice:	DiceCoefficient			
		0.978			
		0.789			
		0.778			
	Jaccard:	JaccardIndex			
		0.956			
		0.651			
		0.637			
	Training time:				
2	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		698380	756	26203	0.963
		3302	529149	109431	0.824
		29149	59823	295755	0.769
		0.956	0.897	0.686	0.869
	Accuracy:				
	Dice:	DiceCoefficient			
		0.959			
		0.859			
		0.725			
	Jaccard:	JaccardIndex			
		0.922			
		0.753			
		0.568			
	Training time:				
3	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		677415	3976	14858	0.973
		3290	662892	180236	0.783
		8643	87493	480172	0.833
		0.983	0.879	0.711	0.859
	Accuracy:				
	Dice:	DiceCoefficient			
		0.978			
		0.828			
		0.767			
	Jaccard:	JaccardIndex			
		0.957			
		0.707			
		0.622			
	Training time:				
4	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		707351	4526	6033	0.985
		1578	558824	179914	0.755
		9498	300411	337010	0.521
		0.985	0.647	0.644	0.762
	Accuracy:				
	Dice:	DiceCoefficient			
		0.985			
		0.697			
		0.576			
	Jaccard:	JaccardIndex			
		0.97			
		0.535			
		0.405			
	Training time:				
5	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		541866	5169	26303	0.945
		1978	539154	92645	0.851
		7684	169055	483378	0.732
		0.982	0.756	0.803	0.838
	Accuracy:				
	Dice:	DiceCoefficient			
		0.963			
		0.8			
		0.766			
	Jaccard:	JaccardIndex			
		0.929			
		0.667			
		0.62			
	Training time:				

Table B.14: TWS J48 statistics folds 6-10

6	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		625513	1626	3698	0.992
		3565	567608	111669	0.831
		40355	123636	228672	0.582
		0.934	0.819	0.665	0.833
	Accuracy:				
	Dice:	DiceCoefficient			
		0.962			
		0.825			
		0.621			
	Jaccard:	JaccardIndex			
		0.927			
		0.702			
		0.45			
	Training time:				
7	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		554112	10571	30227	0.931
		1502	598473	90679	0.867
		13657	194329	605987	0.744
		0.973	0.745	0.834	0.838
	Accuracy:				
	Dice:	DiceCoefficient			
		0.952			
		0.801			
		0.787			
	Jaccard:	JaccardIndex			
		0.908			
		0.668			
		0.648			
	Training time:				
8	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		603469	5092	16216	0.966
		5654	469772	130070	0.776
		42371	105706	509260	0.775
		0.926	0.809	0.777	0.838
	Accuracy:				
	Dice:	DiceCoefficient			
		0.946			
		0.792			
		0.776			
	Jaccard:	JaccardIndex			
		0.897			
		0.656			
		0.634			
	Training time:				
9	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		708215	8136	18257	0.964
		1292	809410	34959	0.957
		11113	163130	632513	0.784
		0.983	0.825	0.922	0.901
	Accuracy:				
	Dice:	DiceCoefficient			
		0.973			
		0.886			
		0.848			
	Jaccard:	JaccardIndex			
		0.948			
		0.796			
		0.736			
	Training time:				
10	Confusion Matrix	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		980836	5426	10847	0.984
		12944	495845	243493	0.659
		120347	81725	345005	0.631
		0.88	0.851	0.576	0.793
	Accuracy:				
	Dice:	DiceCoefficient			
		0.929			
		0.743			
		0.602			
	Jaccard:	JaccardIndex			
		0.868			
		0.591			
		0.43			
	Training time:				

Table B.15: TWS LogitBoost statistics folds 1-5

LogitBoost						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	676355	2053	5132	0.989
		Predicted tumor	656	658264	74906	0.897
		Predicted nontumor	9637	157900	642625	0.793
		Recall	0.985	0.805	0.889	0.888
	Accuracy:	0.888				
	Dice:	Label	DiceCoefficient			
		1	0.987			
		2	0.848			
		3	0.838			
	Jaccard:	Label	JaccardIndex			
		1	0.975			
		2	0.736			
		3	0.722			
	Training time:	3602				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	695884	61	19088	0.973
		Predicted tumor	1045	548600	71698	0.883
		Predicted nontumor	33902	41067	340603	0.82
		Recall	0.952	0.93	0.79	0.905
	Accuracy:	0.905				
	Dice:	Label	DiceCoefficient			
		1	0.963			
		2	0.906			
		3	0.804			
	Jaccard:	Label	JaccardIndex			
		1	0.928			
		2	0.828			
		3	0.673			
	Training time:	3319				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	679139	2125	2912	0.993
		Predicted tumor	1818	684624	174624	0.795
		Predicted nontumor	8391	67612	497730	0.868
		Recall	0.985	0.908	0.737	0.878
	Accuracy:	0.878				
	Dice:	Label	DiceCoefficient			
		1	0.989			
		2	0.848			
		3	0.797			
	Jaccard:	Label	JaccardIndex			
		1	0.978			
		2	0.736			
		3	0.663			
	Training time:	3409				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	707052	433	3027	0.995
		Predicted tumor	1106	536579	192788	0.735
		Predicted nontumor	10269	326749	327142	0.493
		Recall	0.984	0.621	0.626	0.746
	Accuracy:	0.746				
	Dice:	Label	DiceCoefficient			
		1	0.99			
		2	0.673			
		3	0.551			
	Jaccard:	Label	JaccardIndex			
		1	0.979			
		2	0.507			
		3	0.38			
	Training time:	3254				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	541242	2212	13758	0.971
		Predicted tumor	779	537931	61577	0.896
		Predicted nontumor	9507	173235	526991	0.743
		Recall	0.981	0.754	0.875	0.86
	Accuracy:	0.86				
	Dice:	Label	DiceCoefficient			
		1	0.976			
		2	0.819			
		3	0.803			
	Jaccard:	Label	JaccardIndex			
		1	0.954			
		2	0.693			
		3	0.671			
	Training time:	3370				

Table B.16: TWS LogitBoost statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	627223	571	1328	0.997
		Predicted tumor	1947	568851	114237	0.83
		Predicted nontumor	40263	123448	228474	0.583
		Recall	0.937	0.821	0.664	0.835
	Accuracy:	0.835				
	Dice:	Label	DiceCoefficient			
		1	0.966			
		2	0.826			
		3	0.621			
	Jaccard:	Label	JaccardIndex			
		1	0.934			
		2	0.703			
		3	0.45			
	Training time:	3169				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	554521	7614	19208	0.954
		Predicted tumor	646	617821	83644	0.88
		Predicted nontumor	14104	177938	624041	0.765
		Recall	0.974	0.769	0.859	0.856
	Accuracy:	0.856				
	Dice:	Label	DiceCoefficient			
		1	0.964			
		2	0.821			
		3	0.809			
	Jaccard:	Label	JaccardIndex			
		1	0.93			
		2	0.696			
		3	0.679			
	Training time:	3361				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	607702	3562	10543	0.977
		Predicted tumor	2335	502676	107417	0.821
		Predicted nontumor	41457	74332	537586	0.823
		Recall	0.933	0.866	0.82	0.873
	Accuracy:	0.873				
	Dice:	Label	DiceCoefficient			
		1	0.955			
		2	0.843			
		3	0.821			
	Jaccard:	Label	JaccardIndex			
		1	0.913			
		2	0.728			
		3	0.697			
	Training time:	3250				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	708111	5205	8081	0.982
		Predicted tumor	819	830264	30373	0.964
		Predicted nontumor	11690	145207	647275	0.805
		Recall	0.983	0.847	0.944	0.916
	Accuracy:	0.916				
	Dice:	Label	DiceCoefficient			
		1	0.982			
		2	0.901			
		3	0.869			
	Jaccard:	Label	JaccardIndex			
		1	0.965			
		2	0.821			
		3	0.768			
	Training time:	3333				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	982085	3512	8809	0.988
		Predicted tumor	10169	543473	267158	0.662
		Predicted nontumor	121873	36011	323378	0.672
		Recall	0.881	0.932	0.54	0.805
	Accuracy:	0.805				
	Dice:	Label	DiceCoefficient			
		1	0.932			
		2	0.774			
		3	0.599			
	Jaccard:	Label	JaccardIndex			
		1	0.872			
		2	0.632			
		3	0.427			
	Training time:	3157				

Table B.17: TWS RandomForest statistics folds 1-5

RandomForest						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	676470	1673	4430	0.991
		Predicted tumor	509	678175	79883	0.894
		Predicted nontumor	9669	138369	638350	0.812
		Recall	0.985	0.829	0.883	0.895
	Accuracy:	0.895				
	Dice:	Label	DiceCoefficient			
		1	0.988			
		2	0.86			
		3	0.846			
	Jaccard:	Label	JaccardIndex			
		1	0.976			
		2	0.755			
		3	0.733			
	Training time:	14829				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	697611	40	19224	0.973
		Predicted tumor	679	548788	77067	0.876
		Predicted nontumor	32541	40900	335098	0.82
		Recall	0.955	0.931	0.777	0.903
	Accuracy:	0.903				
	Dice:	Label	DiceCoefficient			
		1	0.964			
		2	0.902			
		3	0.798			
	Jaccard:	Label	JaccardIndex			
		1	0.93			
		2	0.822			
		3	0.664			
	Training time:	14650				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	679818	1717	2147	0.994
		Predicted tumor	1520	683979	151783	0.817
		Predicted nontumor	8010	68665	521336	0.872
		Recall	0.986	0.907	0.772	0.89
	Accuracy:	0.89				
	Dice:	Label	DiceCoefficient			
		1	0.99			
		2	0.859			
		3	0.819			
	Jaccard:	Label	JaccardIndex			
		1	0.981			
		2	0.754			
		3	0.693			
	Training time:	14790				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	707113	553	2332	0.996
		Predicted tumor	653	551231	176372	0.757
		Predicted nontumor	10661	311977	344253	0.516
		Recall	0.984	0.638	0.658	0.761
	Accuracy:	0.761				
	Dice:	Label	DiceCoefficient			
		1	0.99			
		2	0.692			
		3	0.579			
	Jaccard:	Label	JaccardIndex			
		1	0.98			
		2	0.53			
		3	0.407			
	Training time:	14771				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	542361	1723	13127	0.973
		Predicted tumor	661	563422	57509	0.906
		Predicted nontumor	8506	148233	531690	0.772
		Recall	0.983	0.79	0.883	0.877
	Accuracy:	0.877				
	Dice:	Label	DiceCoefficient			
		1	0.978			
		2	0.844			
		3	0.824			
	Jaccard:	Label	JaccardIndex			
		1	0.958			
		2	0.73			
		3	0.7			
	Training time:	14585				

Table B.18: TWS RandomForest statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	628879	431	1038	0.998
		Predicted tumor	1406	579217	107325	0.842
		Predicted nontumor	39148	113222	235676	0.607
		Recall	0.939	0.836	0.685	0.846
	Accuracy:	0.846				
	Dice:	Label	DiceCoefficient			
		1	0.968			
		2	0.839			
		3	0.644			
	Jaccard:	Label	JaccardIndex			
		1	0.937			
		2	0.723			
		3	0.475			
	Training time:	14466				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	553753	6709	14508	0.963
		Predicted tumor	541	628897	76949	0.89
		Predicted nontumor	14977	167767	635436	0.777
		Recall	0.973	0.783	0.874	0.866
	Accuracy:	0.866				
	Dice:	Label	DiceCoefficient			
		1	0.968			
		2	0.833			
		3	0.823			
	Jaccard:	Label	JaccardIndex			
		1	0.938			
		2	0.714			
		3	0.699			
	Training time:	14621				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	607276	3445	9251	0.98
		Predicted tumor	1668	489800	90833	0.841
		Predicted nontumor	42550	87325	555462	0.81
		Recall	0.932	0.844	0.847	0.875
	Accuracy:	0.875				
	Dice:	Label	DiceCoefficient			
		1	0.955			
		2	0.842			
		3	0.829			
	Jaccard:	Label	JaccardIndex			
		1	0.914			
		2	0.728			
		3	0.707			
	Training time:	14331				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	708259	5160	5693	0.985
		Predicted tumor	370	830629	25613	0.97
		Predicted nontumor	11991	144887	654423	0.807
		Recall	0.983	0.847	0.954	0.919
	Accuracy:	0.919				
	Dice:	Label	DiceCoefficient			
		1	0.984			
		2	0.904			
		3	0.874			
	Jaccard:	Label	JaccardIndex			
		1	0.968			
		2	0.825			
		3	0.777			
	Training time:	14626				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	998912	3669	5859	0.991
		Predicted tumor	7090	524832	172542	0.745
		Predicted nontumor	108125	54495	420944	0.721
		Recall	0.897	0.9	0.702	0.847
	Accuracy:	0.847				
	Dice:	Label	DiceCoefficient			
		1	0.941			
		2	0.815			
		3	0.712			
	Jaccard:	Label	JaccardIndex			
		1	0.889			
		2	0.688			
		3	0.552			
	Training time:	14678				

Table B.19: TWS SMO statistics folds 1-5

SMO						
1	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	675717	1620	3756	0.992
		Predicted tumor	498	621828	48900	0.926
		Predicted nontumor	10433	194769	670007	0.766
		Recall	0.984	0.76	0.927	0.883
	Accuracy:	0.883				
	Dice:	Label	DiceCoefficient			
		1	0.988			
		2	0.835			
		3	0.839			
	Jaccard:	Label	JaccardIndex			
		1	0.976			
		2	0.717			
		3	0.722			
	Training time:	10250				
2	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	695393	30	17907	0.975
		Predicted tumor	666	553894	82829	0.869
		Predicted nontumor	34772	35804	330653	0.824
		Recall	0.952	0.939	0.766	0.902
	Accuracy:	0.902				
	Dice:	Label	DiceCoefficient			
		1	0.963			
		2	0.903			
		3	0.794			
	Jaccard:	Label	JaccardIndex			
		1	0.929			
		2	0.823			
		3	0.659			
	Training time:	9256				
3	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	679597	1941	2658	0.993
		Predicted tumor	1772	672503	139907	0.826
		Predicted nontumor	7979	79917	532701	0.858
		Recall	0.986	0.891	0.789	0.889
	Accuracy:	0.889				
	Dice:	Label	DiceCoefficient			
		1	0.99			
		2	0.857			
		3	0.822			
	Jaccard:	Label	JaccardIndex			
		1	0.979			
		2	0.751			
		3	0.698			
	Training time:	9639				
4	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	706445	330	1984	0.997
		Predicted tumor	846	507434	180643	0.737
		Predicted nontumor	11136	355997	340330	0.481
		Recall	0.983	0.587	0.651	0.738
	Accuracy:	0.738				
	Dice:	Label	DiceCoefficient			
		1	0.99			
		2	0.654			
		3	0.553			
	Jaccard:	Label	JaccardIndex			
		1	0.98			
		2	0.485			
		3	0.382			
	Training time:	8520				
5	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	540496	1392	7676	0.983
		Predicted tumor	658	533784	46902	0.918
		Predicted nontumor	10374	178202	547748	0.744
		Recall	0.98	0.748	0.909	0.869
	Accuracy:	0.869				
	Dice:	Label	DiceCoefficient			
		1	0.982			
		2	0.825			
		3	0.818			
	Jaccard:	Label	JaccardIndex			
		1	0.964			
		2	0.701			
		3	0.693			
	Training time:	6886				

Table B.20: TWS SMO statistics folds 6-10

6	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	625875	438	820	0.998
		Predicted tumor	1628	572088	112873	0.833
		Predicted nontumor	41930	120344	230346	0.587
		Recall	0.935	0.826	0.67	0.837
	Accuracy:	0.837				
	Dice:	Label	DiceCoefficient			
		1	0.965			
		2	0.829			
		3	0.625			
	Jaccard:	Label	JaccardIndex			
		1	0.933			
		2	0.709			
		3	0.455			
	Training time:	7036				
7	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	552985	6850	10943	0.969
		Predicted tumor	394	595972	70828	0.893
		Predicted nontumor	15892	200551	645122	0.749
		Recall	0.971	0.742	0.888	0.855
	Accuracy:	0.855				
	Dice:	Label	DiceCoefficient			
		1	0.97			
		2	0.811			
		3	0.812			
	Jaccard:	Label	JaccardIndex			
		1	0.942			
		2	0.681			
		3	0.684			
	Training time:	6984				
8	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	606878	2990	8419	0.982
		Predicted tumor	1846	483441	89755	0.841
		Predicted nontumor	42770	94139	557372	0.803
		Recall	0.932	0.833	0.85	0.873
	Accuracy:	0.873				
	Dice:	Label	DiceCoefficient			
		1	0.956			
		2	0.837			
		3	0.826			
	Jaccard:	Label	JaccardIndex			
		1	0.915			
		2	0.719			
		3	0.703			
	Training time:	7648				
9	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	707861	4626	6224	0.985
		Predicted tumor	442	810246	26562	0.968
		Predicted nontumor	12317	165804	652943	0.786
		Recall	0.982	0.826	0.952	0.91
	Accuracy:	0.91				
	Dice:	Label	DiceCoefficient			
		1	0.984			
		2	0.891			
		3	0.861			
	Jaccard:	Label	JaccardIndex			
		1	0.968			
		2	0.804			
		3	0.756			
	Training time:	8249				
10	Confusion Matrix	Label	Groundtruth background	Groundtruth tumor	Groundtruth nontumor	Precision
		Predicted background	1021891	3639	5707	0.991
		Predicted tumor	9858	550930	177099	0.747
		Predicted nontumor	82378	28427	416539	0.79
		Recall	0.917	0.945	0.695	0.866
	Accuracy:	0.866				
	Dice:	Label	DiceCoefficient			
		1	0.953			
		2	0.834			
		3	0.739			
	Jaccard:	Label	JaccardIndex			
		1	0.91			
		2	0.716			
		3	0.587			
	Training time:	8479				

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] E. Alegre, M. Pajares, and A. de la Escalera Hueso. *Conceptos y métodos en visión por computador*. s.l., 2016.

[3] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, and H. Sebastian Seung. Trainable weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 33(15):2424–2426, 2017.

[4] J. Borovec and J. Kybic. jslic: superpixels in imagej. 2014.

[5] R. R. Bouckaert. Bayesian network classifiers in weka for version 3-5-7. *Artificial Intelligence Tools*, 11(3):369–387, 2008.

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] W. Burger and M. J. Burge. *Digital image processing: an algorithmic introduction using Java*. Springer, 2016.

[8] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[9] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[11] E. K. Fishman, D. Magid, D. R. Ney, E. L. Chaney, S. M. Pizer, J. G. Rosenman, D.Ñ. Levin, M. W. Vannier, J. E. Kuhlman, and D. D. Robertson. Three-dimensional imaging. *Radiology*, 181(2):321–337, 1991.

[12] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[14] G. Holmes, A. Donkin, and I. H. Witten. Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE, 1994.

[15] W. Iba and P. Langley. Induction of one-level decision trees. In *Machine Learning Proceedings 1992*, pages 233–240. Elsevier, 1992.

[16] P. Jaccard. Nouvelles researches sur la distribution florale. *Bull Soc Vaud Sci Nat*, 44:223–270, 1908.

[17] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[18] D. Legland, I. Arganda-Carreras, and P. Andrey. Morpholibj: integrated library and plugins for mathematical morphology with imagej. *Bioinformatics*, 32(22):3532–3534, 2016.

[19] D. ping Tian et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–396, 2013.

[20] J. C. Platt. Using analytic qp and sparseness to speed training of support vector machines. In *Advances in neural information processing systems*, pages 557–563, 1999.

[21] J. R. Quinlan. C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38:48, 1993.

[22] R. S. Michalski, J. Carbonell, and T. M. Mitchell. *Machine learning. an Artificial Intelligence approach. Volume 2*, volume Vol. II. 01 1983.

[23] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676, 2012.

[24] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri. The imagej ecosystem: An open platform for biomedical image analysis. *Molecular reproduction and development*, 82(7-8):518–529, 2015.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[26] P. Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[27] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.

[28] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.

[29] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):583–598, 1991.

[30] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[31] D. Zhang, M. M. Islam, and G. Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012.

[32] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.