Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Statistics and Machine Learning
2021 | LIU-IDA/STAT-A--21/021--SE

Classification of brain tumors in weakly annotated histopathology images with deep learning

Dávid Hrabovszki

Supervisor : Anders Eklund
Examiner : Annika Tillander

External supervisor : Neda Haj Hosseini

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida `http://www.ep.liu.se/`.

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: `http://www.ep.liu.se/`.

© Dávid Hrabovszki

Abstract

Brain and nervous system tumors were responsible for around 250,000 deaths in 2020 worldwide. Correctly identifying different tumors is very important, because treatment options largely depend on the diagnosis. This is an expert task, but recently machine learning, and especially deep learning models have shown huge potential in tumor classification problems, and can provide fast and reliable support for pathologists in the decision making process.

This thesis investigates classification of two brain tumors, glioblastoma multiforme and lower grade glioma in high-resolution H&E-stained histology images using deep learning. The dataset is publicly available from TCGA, and 220 whole slide images were used in this study. Ground truth labels were only available on whole slide level, but due to their large size, they could not be processed by convolutional neural networks. Therefore, patches were extracted from the whole slide images in two sizes and fed into separate networks for training. Preprocessing steps ensured that irrelevant information about the background was excluded, and that the images were stain normalized. The patch-level predictions were then combined to slide level, and the classification performance was measured on a test set. Experiments were conducted about the usefulness of pre-trained CNN models and data augmentation techniques, and the best method was selected after statistical comparisons. Following the patch-level training, five slide aggregation approaches were studied, and compared to build a whole slide classifier model.

Best performance was achieved when using small patches (336 x 336 pixels), pre-trained CNN model without frozen layers, and mirroring data augmentation. The majority voting slide aggregation method resulted in the best whole slide classifier with 91.7% test accuracy and 100% sensitivity. In many comparisons, however, statistical significance could not be shown because of the relatively small size of the test set.

Acknowledgments

I would like to thank my supervisors, Anders Eklund and Neda Haj Hosseini, for their continuous support and for the opportunity of working on such an interesting project. I knew that I could turn to you with any questions I had, and receive the answer quickly. Thank you for being so invested in my thesis.

I am also thankful to my examiner, Annika Tillander, and my opponent, Mohsen Pirmoradiyan, for their valuable feedback. Your suggestions were welcome, and the corrections greatly improved the quality of my report.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

It is estimated that around 300,000 new brain and nervous system cancer cases occurred in 2020 worldwide, and around 250,000 deaths occurred from this type of cancer in the same year [48]. The World Health Organization classifies tumors into grades based on their malignancy, where grade I is the least malignant and grade IV is the most malignant [31]. Grade II and III cancers are called Lower Grade Gliomas (LGG), and grade IV cancers are called Glioblastoma or Glioblastome Multiforme (GBM) [36].

It is important to diagnose tumors correctly, because treatment options and survival expectancy depend largely on how malignant a tumor is and what characteristics it has. There are histological differences between different tumors, which helps the expert pathologist in the decision making. Grade I lesions have the possibility of cure after surgery alone, grade II tumors are more infiltrative, can progress to higher grades, and often recur, and grade III is reserved for cancer that has some evidence of malignancy. The treatment of grade III lesions usually include radiation and chemotherapy. Grade IV tumors are malignant, active, necrosis-prone (death of the tissue), progress quickly and often cause fatality [30].

With the advancement of technology, it is now possible to scan, save, analyze and share tissue images using digital microscopy. This technology scans a complete microscope slide and creates a single high resolution file called whole slide image (WSI). These files take up substantial storage and require specific software to view and manipulate them, because they are stored in custom file formats [4].

Hand-crafted feature-based machine learning techniques offer solutions for building models that can analyze whole slide images, but they usually require domain-specific knowledge for crafting the features. Deep learning approaches are getting increasingly popular in medical image analysis thanks to their ability to learn features automatically. Especially convolutional neural networks (CNNs) are used, that have proven to be very effective in image analysis [3].

In this thesis, whole slide images from The Cancer Genome Atlas (TCGA) are used [51], which is a publicly available dataset that contains tissues from GBM and LGG brain tumor grades from many different clinics. The images are labeled as a whole, therefore no pixel-wise annotation is available. The files can be more than 3 GB in size, and their resolution is often

higher than 100,000 x 50,000 pixels, therefore smaller sized patches need to be extracted that can be more easily analyzed by computers at a large magnification.

1.2 Aim

The aim of this thesis is to classify two different grades of brain tumors (glioblastoma and lower grade glioma) from whole slide histology images using deep learning, where pixel-level annotation is unavailable. Specifically, convolutional neural networks (CNNs) will be used to classify individual patches from slides as GBM or LGG, and in a second step these patch predictions will be combined to a single prediction for each slide using different approaches.

This thesis intends to investigate the following research questions:

1. How well can a deep learning model classify whole slide images as glioblastoma or lower grade glioma with a slide-level annotation?
2. Is a pre-trained convolutional neural network significantly better than one trained from scratch?
3. Does image augmentation significantly improve classification performance?
4. What is the best approach for combining the patch-level predictions to a slide-level prediction?

The performance of the competing approaches is compared using DeLong's test for correlated ROC curves. There are several challenges that must be faced before investigating the research questions, one of which is that the slide images are large in size, therefore they need to be divided into patches. This is because CNNs cannot handle so large images. They also come from different sources, so their colors must be normalized. The biggest challenge of this thesis, however, is that there is no annotation available on a pixel or patch level, so it is possible that patches in a slide are cancer-free, or belong to the other class. The patches need to be combined to slide level in the prediction phase, which is not a straight-forward task. In this thesis, five different slide aggregation techniques are experimented with to tackle this problem.

1.3 Related work

Numerous researchers have applied computer algorithms for histology image analysis, and the used methods can be split into two groups. One requires the extraction of hand-crafted features that expert pathologists would recognize from the slide images, the other does not. The second group, where this thesis belongs, can afford to not explicitly obtain these features, because they use deep convolutional networks to automatically do that within the model. The first group therefore uses more traditional machine learning algorithms that accept well defined features. Algorithms like support vector machine, random forest and logistic regression are referred to as traditional machine learning methods that take well structured features as input. The distinction is necessary, because deep learning models are a subset of machine learning, too, but they learn very differently, and they do not required hand-crafted features.

Researchers achieved very good results without using deep learning. Barker et al. [2] extracted coarse features of shape, color and texture from the image patches focusing on cell nuclei, reduced the dimensionality of the features, and created clusters representing similar patterns. More detailed fine features about nuclear morphology were extracted from a select few patches from each cluster, then an elastic net was used on these patches to make the diagnostic decision of classifying a slide as GBM or LGG. Extracting fine features was a very computationally expensive task, this is why only a smaller number of representative patches

were a part of this step. Slide level aggregation was done by weighted voting of the predictions of the patches. All whole slide images came from the TCGA data repository, and were resized to 20x magnification level using bicubic interpolation, and the patches were 1024 x 1024 pixels. They achieved an accuracy of 93.1% on a dataset containing 302 brain cancer cases.

Rathore et al. [39] approached the problem similarly, by using a more traditional machine learning model, the support vector machine. They extracted features from the texture, intensity and morphology along with several clinical measures, and trained the SVM model on them. The creation of features required knowledge about what differentiates GBM from LGG, such as microvascular proliferation, mitotic activity and necrosis. The validation accuracy was 75.12% on images obtained from TCGA.

There is a lot more literature in this field that utilized the power of deep learning for image analysis. Kurc et al. [26] presented the three best performing methods from the 21^{st} International Medical Image Computing and Computer Assisted Intervention (MICCAI 2018) conference for classification of oligodendroglioma and astrocytoma (which are two subclasses of LGG) patients. They all used a combination of radiographic and histologic image dataset, where the histologic images were obtained from TCGA, but they processed the two types of images separately. The three methods achieved accuracy scores of 90%, 80% and 75% respectively.

The best one, [1] applied several preprocessing steps on the images, including region of interest detection, stain normalization and patch extraction (224 x 224). They trained an autoencoder with convolutional layers to extract features from each patch, then used these features to identify patches that can potentially contain tumor regions using anomaly detection (Isolation Forest), where tumor was considered the anomaly. The DenseNet-161 network, pretrained on ImageNet, was then trained on these anomaly patches only, and the final prediction was done according to majority voting. They achieved 90% accuracy on the combined histology and radiology dataset, and 80%, when either only the histology or only the radiology dataset was used.

The second best approach, [35] argues that since only whole slide level annotation is available, but the training is done on patches, this is a weakly-supervised learning problem. To tackle this, they incorporated a Multiple Instance Learning (MIL) framework into the CNN architecture, which helps combine the patch predictions to slide level intelligently. The preprocessing steps were similar to [1], but they used 256 x 256 patches and a more simple histogram equalization for color normalization. Here a pretrained DenseNet-169 model was used with dropout regularization, which produced an average slide level score from all sampled patches for that slide. They concluded that the dropout technique did not improve the accuracy significantly.

The third best solution (only described in [26]), used a different approach. They identified tissue characteristics that differentiate the two classes, such as necrosis, cell density, cell shape and blood vessels. The images were then partitioned into 512 x 512 patches, and fed into a VGG16 CNN network with data augmentation to tackle class imbalance, and dropout and batch normalization layers to reduce overfitting.

A mixture of traditional machine learning and deep learning approaches exists, when the CNNs are only used for automatic feature extraction, but the classification is done by another machine learning algorithm. Xu et al. [58] used a pretrained AlexNet CNN for extracting 4096 features, some of which revealed biological insights, and then employed a SVM. They achieved 97.5% accuracy and concluded that these CNN features are considerably more powerful than expert-designed features.

Campanella et al. [9] conducted a very extensive research, where they used a deep learning approach to classify whether a slide image has cancer in it or not. They tested their methods on very large datasets of different types of cancer, and different slide preparation methods. The datasets were similar to TCGA in a way that they were also not labeled at pixel level, therefore the authors presented different multiple instance learning approaches

to tackle this weakly supervised problem in the form of slide aggregation models. These models included random forest and recurrent neural networks that were trained on the validation set to avoid overfitting. They showed by statistical comparisons that fully supervised learning models based on curated datasets do not generalize well to real world data, where detailed annotation is not available. Even though the authors did not use brain cancer data, some very useful findings and methods can be applied to the TCGA brain tumor dataset, including the statistical comparison of different models.

Other papers have experimented with deep learning for digital pathology, from which this research can benefit in questions such as optimal patch size, architecture, data augmentation methods, pre-processing steps, and slide level aggregation techniques [24], [16], [44], [25], [55], [23].

1.4 Background

Histology is concerned with biological tissues, and its aim is to discover structures and patterns of cells and intercellular substances. Histologists examine tissue samples that have been removed from the living body through surgery or biopsy. These samples are processed and stained with chemical dyes to make the structures more visible, and then they are cut into very thin slices that can be placed under an optical microscope and examined further [49].

Then it is the pathologists' job to determine the causes of disease [50], and histopathology connects the two fields by studying the diseases of pathologic tissues under a microscope. Histopathologists make diagnoses based on pathologic tissues and help clinicians in the decision making process. Specifically, they often provide diagnostic services for cancer, by reporting its malignancy, grade and possible treatment options [7].

There are several histopathological features, such as mitotic activity and necrosis, that distinguish GBM from LGG, which makes it possible for experts to make a diagnosis after inspecting the tissue sections under a microscope. Mitotic activity means the presence of dividing cells, and necrosis is the death of cells. The type of the cells also needs to be taken into account, because although these two features indicate GBM in astrocytoma, they are still graded as LGG in oligodendrogliomas (astrocytoma and oligodendroglioma are two types of brain tumor emanating from different types of cells) [37].

Tissue samples removed from the body are colorless, therefore histologists soak them in hematoxylin and eosin dyes, which highlight different cellular details of the tissue, thus revealing important information based on which it is possible to distinguish different types both for experts and machines. Cellular constituents that are basophilic (have an affinity for basic dye, like hematoxylin), such as the nucleus, are colored blue. Acidophilic (affinity for acid dye, like eosin) components are colored pink, like the cell membrane or the mitochondria. The process is called H&E staining, and it is the standard for histologic examination of tissues [10].

1.5 Ethical considerations

The whole slide histology images used in this thesis were made publicly available by TCGA [51], who are responsible for following the law in privacy and legal aspects. The origin of the collected samples can be traced back to clinics, but not to individuals.

As digital pathology solutions emerge and improve, the dilemma of full automation arises, but currently it is believed to be neither possible, nor very wise. The expert pathologist is still (and likely to remain to be) the ultimate evaluator, when AI solutions are used in clinical environments [52]. One of the reasons why deep learning can be dangerous to use without human supervision is that artificial networks are sensitive to adversarial attacks, when a part of the image is modified by an antagonistic party, which can easily mislead the network [52]. Deep learning models are often described as black boxes, where the process

of decision-making is not transparent enough for a medical expert to base their diagnosis on [52]. One way to mitigate this issue is by visualizing the activation regions inside the network to get an idea about why a specific prediction was made.

2 Data

In this thesis, microscopic histology images are analyzed that are publicly available from The Cancer Genome Atlas (TCGA) [51]. There are two types of histology images, tissue slides and diagnostic slides. Tissue slides come from frozen samples, and are most commonly used for quick diagnosis, whereas diagnostic slides are results of samples being treated with formalin and paraffin to conserve their structure, which allows them to be used for a more detailed diagnosis. In both cases, the samples are very thinly cut and put on a glass slide under the microscope. Frozen samples are more difficult to handle, however, and can more often have artifacts in them as a result of incorrect freezing. Therefore, it is more appropriate to use formalin-fixed paraffin-embedded (FFPE) tissues, or diagnostic slides, as TCGA calls them, for histology analysis [45].

There are 860 examples of GBM and 844 examples of LGG available as diagnostic slides in TCGA, the whole dataset taking up 1.4 TB of storage space. Two examples can be observed in Figure 1. The files are saved in a special format (svs) that allows for efficient storage of large images. This file format also makes it possible to obtain the slides at different magnification levels at which the microscope scanned them. The maximum magnification level is either 40x or 20x, and at this level a typical resolution would be 100,000 x 50,000 pixels, but this can vary largely. All LGG labelled images were obtained at 40x magnification (0.25 μm/pixel), while 77% of GBM images have only 20x magnification (0.5 μm/pixel) available. In order to analyze them together, all images need to be at the same magnification level, so those scanned at 40x are downscaled to 20x. There are several techniques available for rescaling images, but here the bicubic interpolation is used, similarly to [2] from the Pillow library [12], because it offers very good quality, although at a high computational cost.

Utilizing all of these images requires a massive cluster of computers, so this thesis focuses on a smaller subset of images that can be processed by the computer available in reasonable time. Some images had to be excluded in the beginning, because they would not fit into the memory of the computer. The threshold for images that can be handled was 8 gigapixels (10^9 pixels), which means that 359 whole slide images were excluded (Figure 2).

220 whole slide images were sampled randomly and visually inspected for data quality issues. 26 images were excluded due to blur, or red, green, blue or black pen markings to make it easier to train the classifier. An example of each data quality issue can be seen in Figure 3. This adds bias to the estimated performance of the model, if it is later expected to classify all slides that contain both high and low quality images. It would have been possible

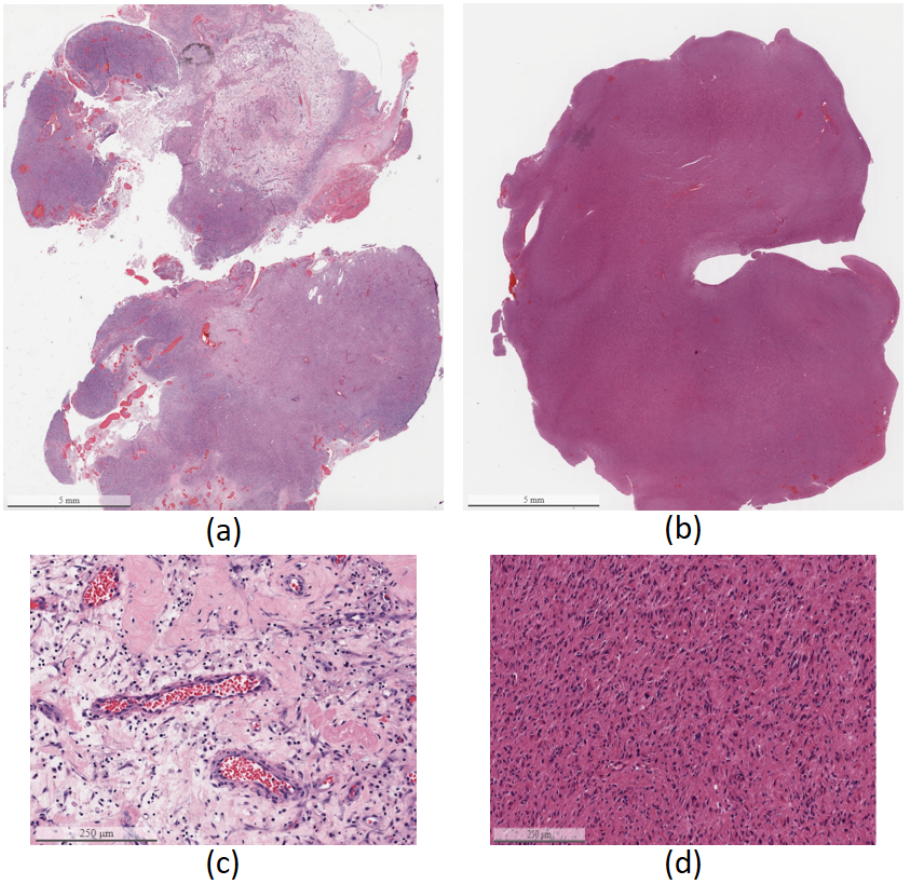

Figure 1: Examples from the TCGA dataset [51]. Glioblastoma Multiforme (GBM) whole slide image (a) and a magnified part of it (c). Lower Grade Glioma (LGG) whole slide image (b) and a magnified part of it (d).

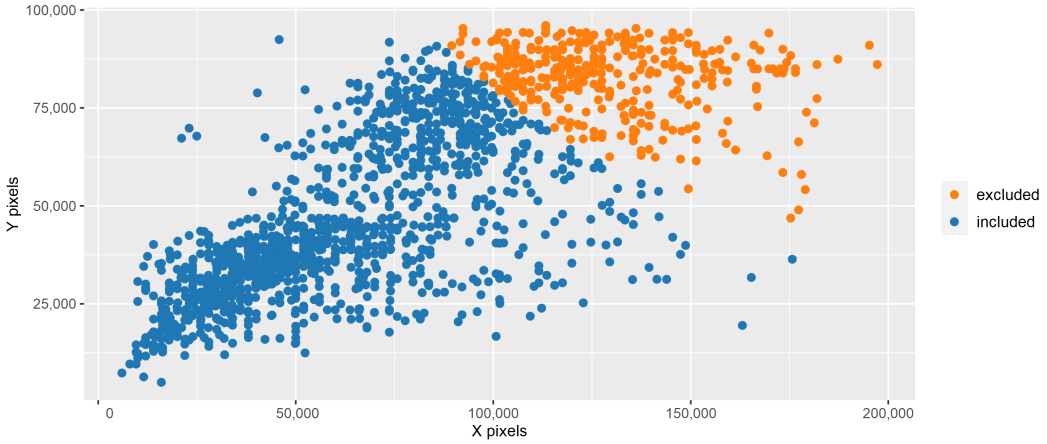

Figure 2: Scatter plot of the resolutions of the whole slide images. Slides larger than 8 gigapixel are excluded, because they do not fit in the memory (64 GB) of the computer available.

to satisfactorily filter out black, blue and green pen markings, but not red, because its RGB color components can be exactly the same as the tissue on some of the more reddish slides,

as well as blood cells. I decided not to make an exception with red color only, so I excluded slides with any colored pen markings.

The process of excluding slides from the original 1,704 followed these criteria:

1. 359 slides were excluded due to their large dimensions, which prevented them from fitting in the memory of the computer (Figure 2).
2. Out of the 1,345 slides eligible for analysis, 220 slides were randomly sampled keeping the equal ratio of the two classes. This number was somewhat arbitrarily chosen, but also with the knowledge from preliminary experiments that processing whole slides is very computationally expensive. The aim was to work on a dataset that is at least as large as most of the related works, but also small enough to analyze in reasonable time.
3. Out of the 220 sampled slides, 26 images were excluded due to data quality issues, such as pen markings, or blur (Figure 3).

In the end, 194 whole slide images were used to create a classifier, 97 from each class. 84 of them are used for training the model (43.3%), 26 for validation and parameter tuning (13.4%), and 84 for testing (43.3%). The relatively large ratio of the test set is necessary, because conclusions about the generalization error cannot be drawn from a small number of test examples.

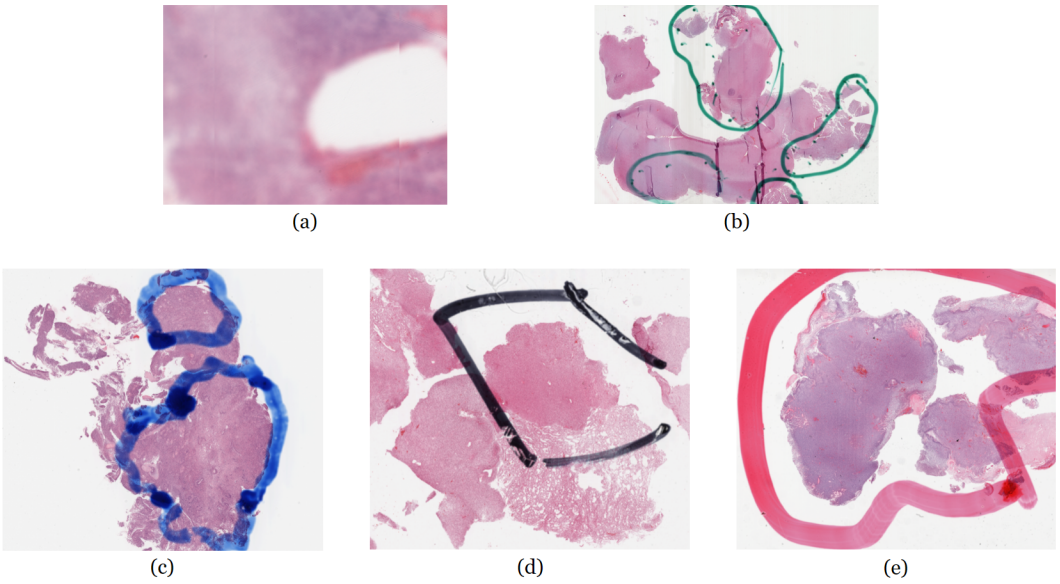

Figure 3: Examples of whole slide images excluded from the analysis. A small part of a blurry whole slide image (a), images with green (b), blue (c), black (d), and red (e) pen markings. Excluding these images means that the trained classifier is not expected to perform as well in a real scenario where images have different quality.

Since the images are very large, it is impossible to process them as a whole without losing important morphological details, therefore patches or tiles are extracted from them, that are easier to handle for a neural network. Different patch sizes are experimented with later on in this thesis, but every slide is divided into a few thousand patches on average, so a large amount of data is available.

3 Theory

3.1 Modelling

In this section, the theory behind the used methods related to the patch-level classifier model is introduced.

3.1.1 Deep learning

Binary classification, such as the GBM vs LGG task at hand, is a supervised learning problem within machine learning. The goal is to build a model that can correctly classify previously unseen data after performing training on a labelled dataset. In this case, the model is shown many images of GBM and LGG, and it outputs a score for both of the classes, from which a prediction can be concluded. To make it possible for the machine to learn, a function needs to be defined that measures the error between the true and predicted classes. The parameters of the model are then adjusted during training to minimize this error, resulting in a more accurate classifier [27].

Deep learning models are very effective in image classification tasks, because they are able to automatically discover the representations that are key to distinguishing between different classes of images. Deep learning models are artificial neural networks with many layers, that, when applied to image recognition, detect features at an increasing level of representation. The first layer usually detects edges, the second simple patterns, and the third assembles these patterns into larger combinations. If more layers are added, more complex shapes and objects can be recognized [27].

A simple multilayer neural network is shown in Figure 4 consisting of three input units, two hidden layers with four and three hidden units, and two output units. An arrow from one unit to another represents the connection, and every connection has a weight w associated with it, these are the adjustable parameters of the network. There are also biases in every layer except for the output layer, but these are omitted from the illustration for simplicity. The equations are also shown in the figure that are used for a forward pass through the network starting from an input x and resulting in the output prediction. The input to each of the units (z) in the following layers is the weighted sum of the output of the units in the previous layer. A non-linear activation function f is then applied to z to calculate the output of the unit (y). This function is usually the rectified linear unit (ReLU): $f(z) = max(0, z)$ [27].

The softmax function is often used as the activation function of the output layer of a classifier, because it represents a probability distribution over all the classes, so a prediction vector can be regarded as the probabilities of the example belonging to each class. It is given by

$$softmax(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j)},$$

where i is the output class we are interested in, and the summation over j is all the possible classes. The softmax function normalizes z, so the result is a value between 0 and 1 [20].

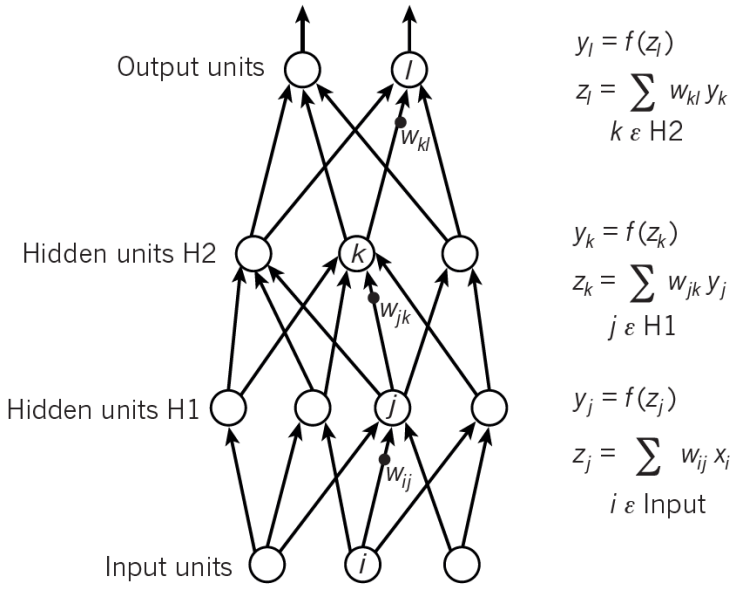

Figure 4: Multilayer neural network with two hidden layers. The equations show how to calculate a forward pass through the network from input x to output. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature Nature (Deep learning, Yann LeCun et al), © (2015) https://www.nature.com

Training a deep learning model means adjusting its many weights in a way that the output prediction is as close to the true label as possible, or in other words, the error is as small as possible. The learning algorithm computes the gradient for every weight, which indicates by how much the error changes, if the weight increased by a small amount. This is useful, because then the weight can be adjusted in a way that it decreases the error. The most common algorithm is stochastic gradient descent, during which a few input examples are shown at once, the errors are calculated after a forward feed through the network, and the average gradient for each weight is computed based on these errors. The weights are then adjusted in a way that takes the error closer to its minimum. This is repeated many times, until the error stops decreasing or some other stopping criterion is fulfilled [27].

For the gradient descent to work, the error derivatives need to be found first for every weight in the network. This is achieved with the backpropagation algorithm, with which one can obtain the partial derivatives of the error with respect to every weight. The algorithm uses the chain rule of derivatives, a formula to compute the derivative of a composite function, such as the cost (also known as error or loss) function in neural networks. This backward pass is described in Figure 5. First, the output from the forward pass (prediction) and the true label is compared to calculate the error E based on the cost function. Then the error derivative with respect to the output of every unit is calculated ($\frac{\partial E}{\partial y}$). If the cost function for unit l is $0.5(y_l - t_l)^2$, where t_l is the target value, the error derivative with respect to unit l is $y_l - t_l$. This is followed by "reversing" the activation function f to obtain the derivative with

respect to the input of each unit ($\frac{\partial E}{\partial z}$). This way it is possible to obtain the partial derivatives of the cost function with respect to the weights, that indicates in which direction they should be adjusted in order to decrease the error [27].

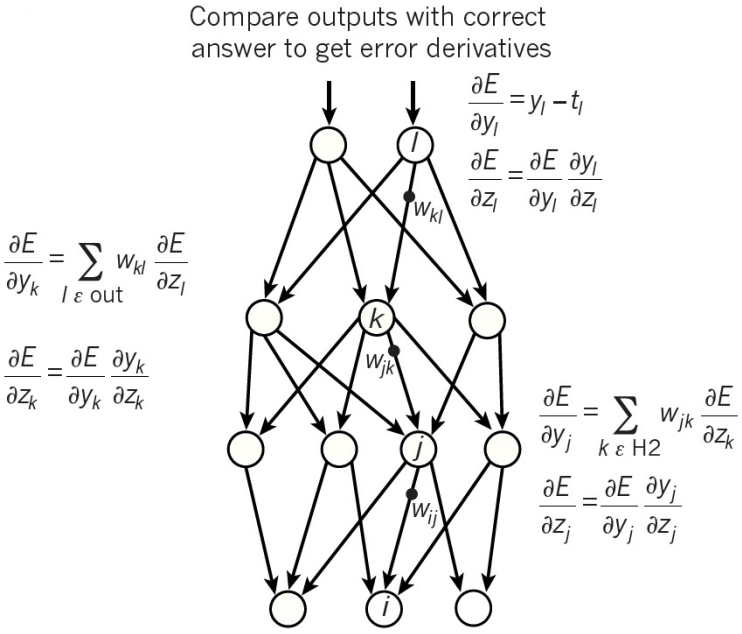

Figure 5: A backward pass performed on a multilayer neural network with two hidden layers. The goal is to obtain partial derivatives of the error with respect to all the weights, so they can be adjusted appropriately. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature Nature (Deep learning, Yann LeCun et al), © (2015) https://www.nature.com

Deep learning allows artificial neural networks to learn complex representations of data by discovering structures and patterns automatically. In contrast with traditional machine learning methods, deep learning models do not require domain expertise in designing features, therefore very little engineering is necessary. Deep learning can be used for classification tasks very well, because the network is able to recognize features that are important for discrimination and will be robust to irrelevant variations. Images are one type of data that can be analyzed very successfully with deep learning architectures, especially with convolutional neural networks (CNN), because they are much easier to train and generalize much better due to the fact that their adjacent layers are not connected fully [27].

3.1.2 Convolutional neural networks

CNNs are designed to process data consisting of multiple arrays, such as color images that have three two-dimensional arrays of pixel intensities in red, green and blue color channels. CNN architectures usually consist of similar series of stages. The first layers usually perform convolutions, where the units are organized into feature maps, and are connected to local patches in the feature maps of the previous layer. The weights of these connections make up the filters (Figure 6). Similarly to standard deep learning architectures, a non-linear activation function is applied to the local weighted sum, which can be the ReLU function introduced above. All the units in a feature map share the same weights, therefore they apply the same filters on all the local patches of the previous layer, thus extracting the same features over the whole image. The filtering operation is mathematically a discrete convolution, this is where the name comes from. Convolutional layers are typically followed by a pooling layer, whose job is to merge similar features close to each other into one (Figure 7). It also reduces the

dimensionality of the convolved feature map, and allows the representations (patterns) to vary little when elements in the previous layer vary a lot [27].

Convolutional and pooling layers are stacked on top of each other to form the feature learning part of the network. The more complex the task is, the more layers are likely to be necessary. An example of a convolutional neural network is shown in Figure 9.

After the features are extracted from the image, fully connected layers and an output layer with usually softmax activation follows to obtain the classification probabilities for every possible class. Training the CNN is no different than training a standard deep fully connected network, because the same backpropagation algorithm can be applied, and the weights of the filters can be trained [27].

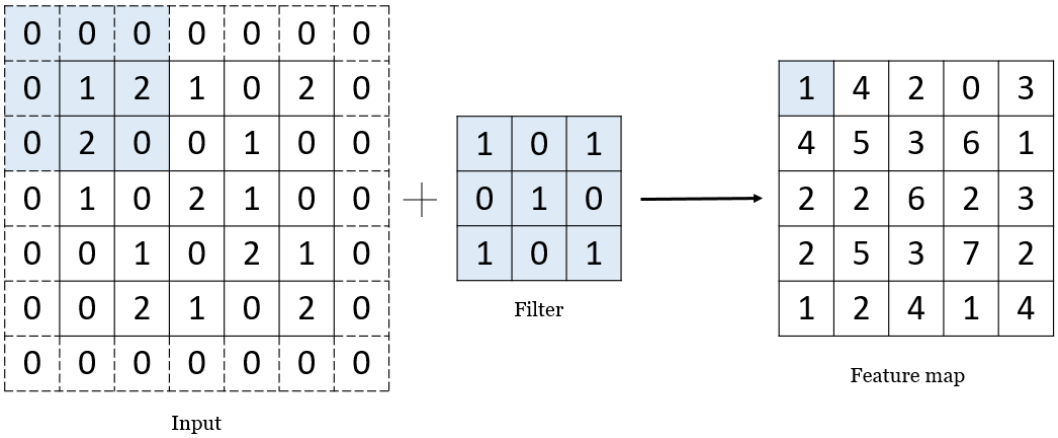

Figure 6: Convolution operation with a 3 x 3 filter on a 5 x 5 input image. The resulting feature map is also 5 x 5 in this case, because the input is padded with zeros around the borders. The filter (kernel) moves around the input image with overlap, and produces the feature map on the right.

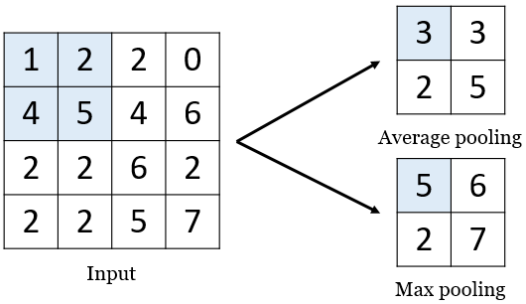

Figure 7: Pooling operations performed on an input image (or feature map). Average pooling calculates the average of a small patch in the input, whereas max pooling selects the highest value in the patch. The result has smaller dimensions than the input.

A flattening layer is necessary before the classification stage to transform the features to one-dimensional space that can be connected to standard fully connected layers (Figure 8).

The first paper that used convolutional networks trained by backpropagation for classifying hand-written digits was published in 1990 [28], but CNNs became increasingly popular with the arrival of fast graphics processing units (GPUs), and their ability to compute tasks in a massively parallel way, making the training process much faster [38].

Convolutional networks are superior to fully connected networks in image processing, because they are robust to geometric distortions, and the location of features does not matter

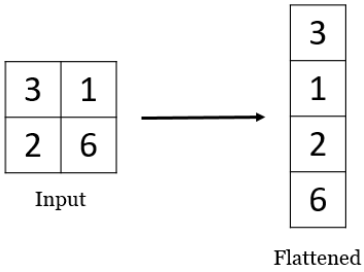

Figure 8: Flattening operation performed on an input (usually a feature map), so that the result is a one-dimensional array. This is a necessary step before the classification stage.

too much. They also require far fewer images to train thanks to the lower number of connections inside the network. Images usually have several hundred pixels, so a fully connected network with 100 hidden units in the first layer would already have several 10,000 weights. Convolutional networks do not have connections between all the nodes in two adjacent layers, and the weights are shared inside a layer, so they do not have to be trained separately. This drastically decreases the number of trainable parameters in the network, which makes the training faster. [29].

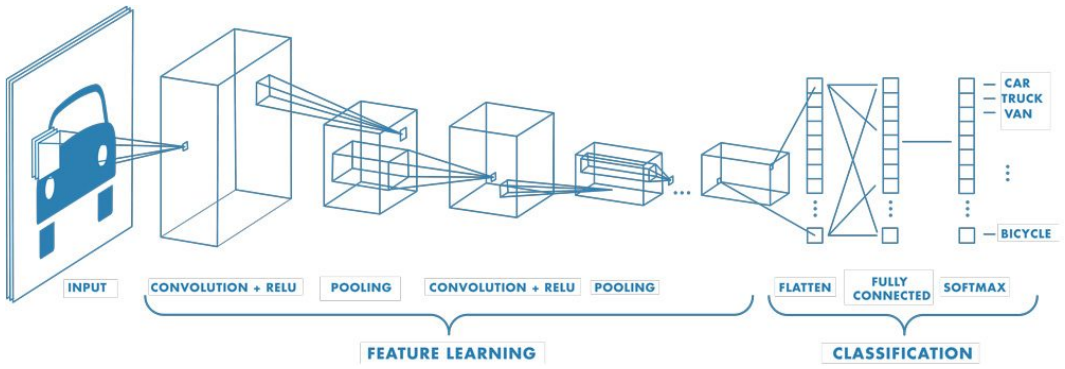

Figure 9: Example of a convolutional neural network. Convolutional and pooling layers are stacked on top of each other to learn the features, then a fully connected block with softmax output carries out the classification [32].

3.1.3 Deep residual networks

Deep residual convolutional neural networks (ResNets) were introduced in [22], and they were a breakthrough in image classification, because they solved the degradation problem, which occurs when networks perform worse as they get deeper after a certain point. This went against the general idea that deeper networks should produce no higher training error than a shallower version of the same architecture. The assumption is that if more layers are added, even if they do not add anything to the performance, at least should pass on the learned information from the previous layers, acting as simple identity mapping. The fact that this did not happen indicates that the solvers have difficulty approximating identity mappings with nonlinear layers, therefore it was suggested that shortcuts are created between small blocks of nonlinear layers, where the information can pass more easily, if that is what is needed. ResNets also allow deep architectures to have lower complexity than previous networks with fewer layers [22]. A building block of the ResNet50 architecture is shown in Figure 10.

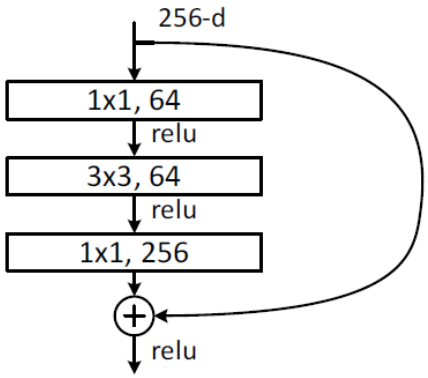

Figure 10: Building block for the ResNet50 architecture. There are three convolutional layers with 1 x 1, 3 x 3 and 1 x 1 filter sizes respectively in this block, which are passed through the ReLU activation function. The information can also bypass these three layers, if a simple identity mapping is desired [22] © 2016 IEEE.

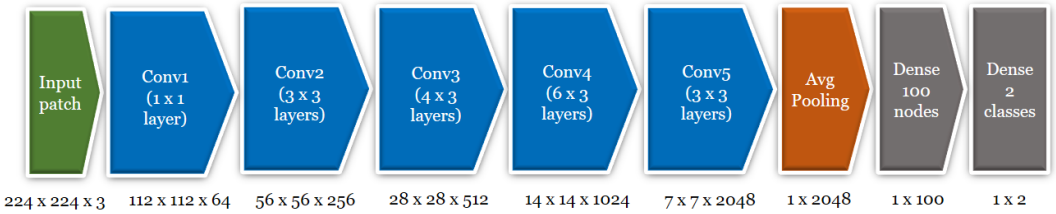

Figure 11: ResNet50-based modified architecture for this thesis. A colored input image of size 224 x 224 is passed through five convolutional blocks that extract features from it (2048 in the end). The output dimensions of each of the blocks can be observed in the bottom. The pooling and fully connected layers at the end with softmax output carry out the classification.

The ResNet50 architecture consists of five convolutional blocks that each include 1,3,4,6, and 3 smaller blocks. A modified version for the current binary classification problem is visualised in Figure 11. With a colored input image of size 224 x 224 pixels, the first block outputs 64 convolutional feature maps of size 112 x 112 pixels. The number of features grows as the depth increases, in the end, the network extracts 2048 features of size 7 x 7 pixels. After the feature extraction part, average pooling, flattening, and fully-connected layers with dropout carry out the classification. The building block shown in Figure 10 is one of the three smaller blocks in Conv2 here.

The reason behind using dropout in the fully-connected layers is two-fold. Dropout is a technique for addressing the problem of overfitting in neural networks. The idea is to temporarily drop units at random from the layer (and their connections to adjacent layers' nodes). This is useful, because it breaks up co-adaptations between weights by making the presence of the units unreliable. These co-adaptations arise from the backpropagation algorithm, when weights train (adapt) with strong correlation, and they can cause the model to fit very well for the training data, but also to generalize badly to unseen data [46]. Using dropout in test time also makes it possible to use Monte Carlo Dropout for estimating prediction uncertainty, which is described in the next section.

3.1.4 Uncertainty estimation

It would be ideal to obtain the level of uncertainty of predictions of a neural network, because then one could decide if its predictions can be relied on or not. It could be especially useful in the case of GBM versus LGG classification, because the network is trained on patch level,

but there are only slide level labels, so the network is expected to be uncertain about some patches. There are also patches that contain neither of the classes (healthy tissue), but still inherit the label from the slide level, which makes it even more difficult for the neural network to confidently predict every instance.

The last layer of a network is usually a softmax layer that outputs values between 0 and 1 for the possible classes, but it would be a mistake to interpret these as the confidence of the model about the current prediction, because it is still possible that a model is uncertain despite a high softmax output [18].

Gal and Ghahramani [18] introduced a statistically sound method for obtaining uncertainty estimates in neural networks, that have fully connected layers with dropout enabled in prediction time. By randomly dropping out nodes, the predictions of the network are not deterministic any more, but form a probability distribution, from which uncertainty estimates can be deduced. This approach is called Monte Carlo dropout, and it can be used in any neural network that employs random dropout layers before its weight layers. In CNNs, that might only have one dropout layer before the last fully connected layer, it is still possible to obtain uncertainty estimates, and there is no extra computational cost, because dropout is used for regularization anyway. Monte Carlo dropout requires a relatively small number of forward passes (10-100) to create the predictive distribution of each test instance, then uncertainty can be deduced from it.

3.2 Slide aggregation

After the patch-level predictions are obtained, they need to be combined to slide level. The concepts behind the methods used in this thesis for slide aggregation are introduced in this section.

3.2.1 Multiple instance learning

When obtaining fully ground truth labels is impossible, machine learning methods work with weak supervision, instead of the more traditional strongly supervised learning. One branch of weakly supervised learning is inexact supervision, where only coarse-grained labels are available [59]. The general term for entities for which the labels are known is *bags*, and for smaller objects belonging to a bag is *instances*. This is also called *multiple-instance learning*, where the goal is to predict labels of new bags ([59], [14]).

Formally, the goal is to learn a function $g : \mathcal{P} \mapsto \mathcal{Q}$ given a training dataset $D = (P_1, q_1), ..., (P_m, q_m)$, where $P_i = \{p_{i,1}, ..., p_{i,m_i}\} \subseteq \mathcal{P}$ is a bag, $p_{i,j} \in \mathcal{P} (j \in \{1, ..., m_i\})$ is an instance, m_i is the number of instances in P_i and $q_i \in \mathcal{Q} = \{YES, NO\}$, YES and NO being the two possible classes [59].

3.2.2 Logistic regression

A second-level model can be built on top of the patch-level classifier (CNN) to learn the optimal function for aggregating patches to slides. Logistic regression can be employed as such a second-level model.

Logistic regression is a statistical model for classification that uses the logistic sigmoid function to output the probability of an instance belonging to a class $\mathcal{C}_1$ by applying the sigmoid function on the linear combination of its feature vector ϕ and some weights ω [6]:

$$p(\mathcal{C}_1|\phi) = \gamma(\phi) = \sigma(\omega^T \phi),$$

where the sigmoid function σ is

$$\sigma(a) = \frac{1}{1 + exp(-a)},$$

and $a = \omega^T \phi$.
In binary classification the other class is given by:

$$p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi)$$

If the feature vector ϕ has M dimensions, the model also has M adjustable parameters. The best fitting model is found by optimizing these parameters using maximum likelihood. For a dataset $\{\phi_n, o_n\}$, where $o_n \in \{0, 1\}$, the likelihood function is:

$$p(\mathbf{o}|\omega) = \prod_{n=1}^{N} \gamma_n^{o_n} \{1 - \gamma_n\}^{1-o_n},$$

where $\mathbf{o} = (o_1, ..., o_N)^T$ and $\gamma_n = p(\mathcal{C}_1|\phi_n)$. To define an error function to be minimized, the negative logarithm of the likelihood function is used, which gives the cross-entropy error function:

$$J(\omega) = -\ln p(\mathbf{o}|\omega) = -\sum_{n=1}^{N} \{o_n \ln \gamma_n + (1 - o_n) \ln(1 - \gamma_n)\},$$

where $\gamma_n = \sigma(a_n)$ and $a_n = \omega^T \phi_n$. When minimizing this function by changing its parameters, the gradient of the function needs to be calculated with respect to the parameters ω:

$$\nabla J(\omega) = \sum_{n=1}^{N} (\gamma_n - o_n)\phi_n.$$

This does not lead to a closed-form solution, however, so optimization has to be done in an iterative way. The error function is a concave function of ω, therefore it has a unique minimum, which can be reached with a suitable optimization technique [6].

3.3 Evaluation methods

The aim of this thesis is to conclude which approach performs best, so statistical comparisons are necessary. DeLong's test for correlated ROC curves is used to compare the competing methods.

3.3.1 DeLong's test for correlated ROC curves

Receiver Operating Characteristic (ROC) curves are commonly used for evaluating and comparing the performance of classifiers, because they show the trade-off between sensitivity and specificity at different threshold values [34]. The area under the ROC curve is abbreviated as AUC, and it is a single value that measures the overall performance of the classifier [33]. The AUC can be between 0.5 and 1, where 0.5 represents chance, and 1 represents a perfect classifier, this way the AUC makes it easy to compare competing binary classifiers.

Several different models and approaches are investigated in this thesis, and comparisons about their performance are made based on ROC curves and AUC values. DeLong's test offers a solution for such comparisons by testing if correlated ROC curves are significantly different.

When one constructs multiple empirical ROC curves based on the same examples (but different models), the correlation of the data must be taken into account when conducting statistical comparisons [13]. DeLong et al. [13] presented a nonparametric approach for comparing AUCs of correlated ROC curves, which is described below.

Let us assume a binary classification problem, and class 1 denotes the presence of a disease (for example GBM in this case). Some of the examples actually have the disease (h), the rest do not (l). Let Cl_1 denote the first group and Cl_2 the second. Suppose a binary classifier that

predicts the probability of the disease being present in an example, and let these probabilities be denoted by X_i, $i = 1, 2, ..., h$ and Y_j, $j = 1, 2, ..., l$ for members of Cl_1 and Cl_2 respectively.

1. The key insight is that the empirical AUC, when calculated by the trapezoidal rule, has been shown to be equal to the Mann-Whitney U-statistic applied to Cl_1 and Cl_2. The formula to calculate the statistic is

$$\hat{\theta} = \frac{1}{hl} \sum_{j=1}^{l} \sum_{i=1}^{h} \psi(X_i, Y_j),$$

where

$$\psi(X, Y) = \begin{cases} 1, & Y < X \\ \frac{1}{2}, & Y = X \\ 0, & Y > X \end{cases},$$

which means intuitively that the AUC increases by $\frac{1}{hl}$ if the predicted disease probability of a member of Cl_2 is less than that of a member of Cl_1, and it increases by $\frac{0.5}{hl}$ if they are the same. The AUC does not increase if the predicted disease probability of a member of Cl_1 is less than that of a member of Cl_2, because this surely results in misclassification.

Since the estimate $\hat{\theta}$ is equal to the Mann-Whitney U-statistic, which is a generalized U-statistic, asymptotic normality and an expression for the variance can be derived from the theory for generalized U-statistics (the theory itself is not discussed in this thesis).

If there are k different models to compare, the vector of AUC estimates is $\hat{\boldsymbol{\theta}} = (\hat{\theta}^1, \hat{\theta}^2, ..., \hat{\theta}^k)$ of the true AUCs $\boldsymbol{\theta} = (\theta^1, \theta^2, ..., \theta^k)$.

2. For the AUC estimate (U-statistic) $\hat{\theta}^r$, where $1 \leqslant r \leqslant k$, structural X and Y components can be calculated in the following way:

$$V_{10}^r(X_i) = \frac{1}{l} \sum_{j=1}^{l} \psi(X_i^r, Y_j^r), (i = 1, 2, ..., h)$$

$$V_{01}^r(Y_j) = \frac{1}{h} \sum_{i=1}^{h} \psi(X_i^r, Y_j^r), (j = 1, 2, ..., l).$$

3. Now let us define the $\boldsymbol{S_{10}}$ and $\boldsymbol{S_{01}}$ $k \times k$ matrices in a way that their (r, s)th element is:

$$S_{10}^{r,s} = \frac{1}{h-1} \sum_{i=1}^{h} [V_{10}^r(X_i) - \hat{\theta}^r][V_{10}^s(X_i) - \hat{\theta}^s]$$

$$S_{01}^{r,s} = \frac{1}{l-1} \sum_{j=1}^{l} [V_{01}^r(Y_j) - \hat{\theta}^r][V_{01}^s(Y_j) - \hat{\theta}^s].$$

4. $\boldsymbol{S_{10}}$ and $\boldsymbol{S_{01}}$ are combined to get the estimated covariance matrix of the parameter estimates $\hat{\boldsymbol{\theta}} = (\hat{\theta}^1, \hat{\theta}^2, ..., \hat{\theta}^k)$:

$$\boldsymbol{S} = \frac{1}{h} \boldsymbol{S_{10}} + \frac{1}{l} \boldsymbol{S_{01}}.$$

5. Using the asymptotic theory for U-statistics,

$$\frac{\boldsymbol{L}\hat{\boldsymbol{\theta}}^T - \boldsymbol{L}\boldsymbol{\theta}^T}{\sqrt{\boldsymbol{L}\left(\frac{1}{h}\boldsymbol{S_{10}} + \frac{1}{l}\boldsymbol{S_{01}}\right)\boldsymbol{L}^T}}$$

has a standard normal distribution, where $\boldsymbol{L}$ is a row vector of coefficients. If only two AUCs are to be compared for difference, let us set $\boldsymbol{L} = [1 - 1]$, in which case the null hypothesis is that there is no statistically significant difference, so $\boldsymbol{L}\boldsymbol{\theta}^T = 0$.

$$H_0 : \hat{\theta}^1 = \hat{\theta}^2$$

The previous expression becomes

$$\frac{\hat{\theta}^1 - \hat{\theta}^2}{\sqrt{\boldsymbol{L}\left(\frac{1}{h}\boldsymbol{S_{10}} + \frac{1}{l}\boldsymbol{S_{01}}\right)\boldsymbol{L}^T}},$$

or, according to [47],

$$\frac{\hat{\theta}^1 - \hat{\theta}^2}{\sqrt{\mathbb{V}[\hat{\theta}^1 - \hat{\theta}^2]}} = \frac{\hat{\theta}^1 - \hat{\theta}^2}{\sqrt{\mathbb{V}[\hat{\theta}^1] + \mathbb{V}[\hat{\theta}^2] - 2\mathbb{C}[\hat{\theta}^1, \hat{\theta}^2]}} = \zeta,$$

where the variances of the AUC estimates are the diagonal values of $\boldsymbol{S}$ and the covariance is the off-diagonal value.

6. With the ζ score obtained above, a two-tailed test can be performed to determine whether the null hypothesis can or cannot be rejected at a given significance level. The alternative hypothesis is that there is a statistically significant difference between the two AUC estimates $\hat{\theta}^1$ and $\hat{\theta}^2$.

4 Methods

In this chapter, the methods used in the thesis are described building on the theoretical background introduced in chapter 3. An overview of the workflow of the thesis can be seen in Figure 12.

4.1 Preprocessing

Whole slide images cannot be directly processed due to their large size, therefore certain preprocessing steps need to be made. Ignoring the glass background of the scans and accounting for the variability of the stain colors across the dataset also need to be addressed. To access the whole slide images, the OpenSlide (3.4.1) package was used [19].

4.1.1 Filtering

WSIs generally contain at least 50% white background, which needs to be filtered out, because it carries no useful information. Different methods were experimented with, and a simple approach was chosen that looks at each pixel's green channel value, and filters out those that are above the intensity threshold of 200. If the image is particularly bright, the resulting binary mask can easily cover more than 90% of the image. In this case, the threshold is automatically raised, until the mask cover rate drops below 90%. For this task, the *deephistopath* package from IBM CODAIT was used [15]. This approach can be used for H&E-stained histology datasets, because tissue is colored pink or purple, which have very low green components, in contrast with the white background.

The goal of background filtering is to remove the white pixels surrounding the tissue, but it is important to note that the methods tried often filtered out pixels inside the tissue area that had a brighter color. To prevent this, another filter was used from the *scikit-image* Python package [54] to remove these small holes resulting from the background filter.

To apply the above mentioned filters on WSIs of very high resolution is very computationally expensive, therefore they were applied on their thumbnail versions instead, and the resulting binary mask was upscaled to the full resolution. The thumbnails are the lowest resolution versions of the full sized images, and are typically around 3,000 x 1,500 pixels, so they are the equivalent of downscaling the full image by a factor of either 16x or 32x, depending on the slide.

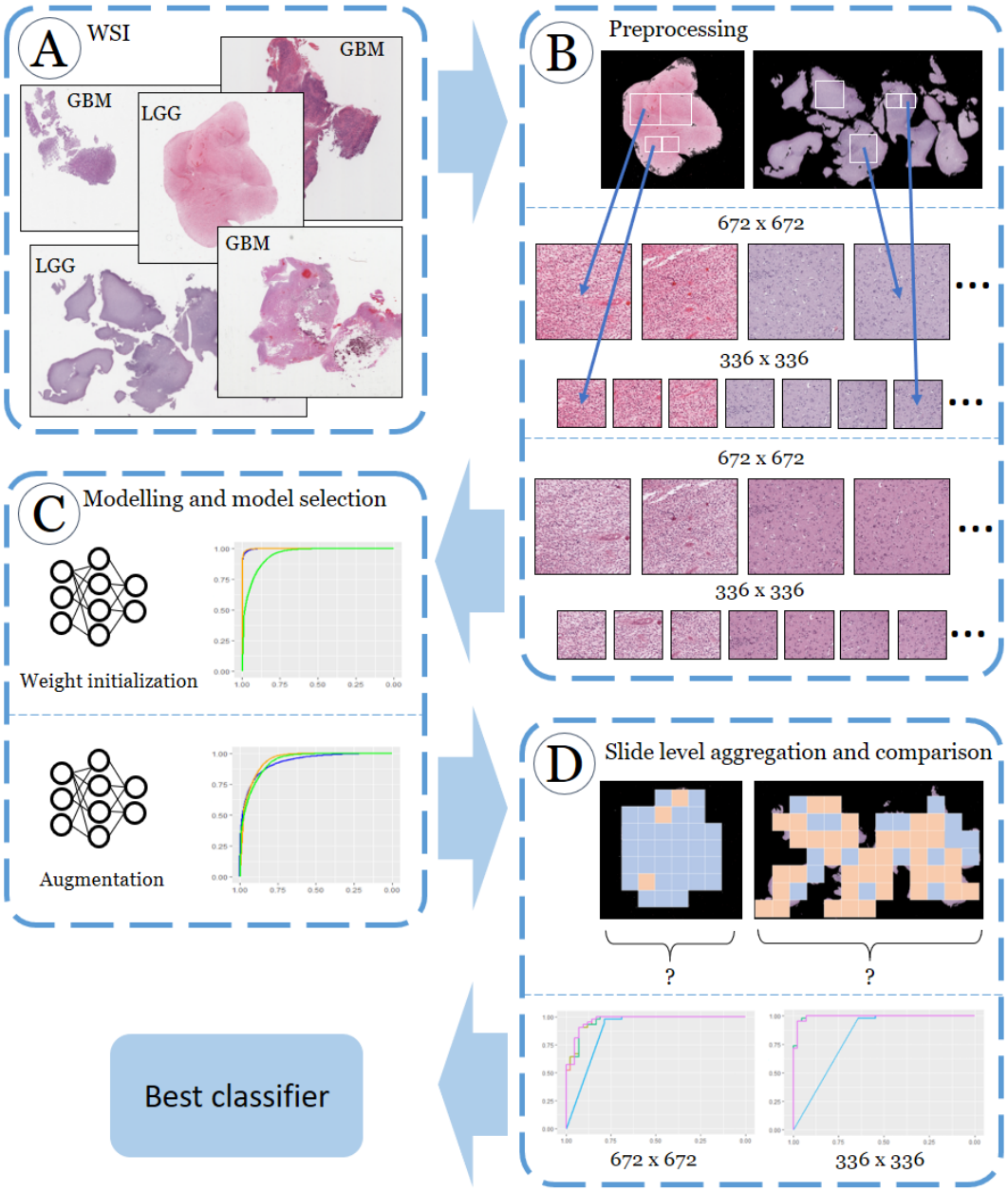

Figure 12: Workflow of the thesis. A: Downloading whole slide images from TCGA [51] and inspecting the images for data quality issues. B: Preprocessing the whole slide images. First, the white background is filtered out, and smaller patches in two sizes are extracted. Then stain normalization makes the patches appear more similar in color. C: Convolutional neural networks are used as binary patch classifiers, and their performance is compared to select the best one. Experiments are conducted with different weight initialization and data augmentation techniques. D: After obtaining the patch predictions, the performance of different slide aggregation approaches is compared on the slide level. In the end, the conclusion can be made about which model setup, aggregation method and patch size is the best choice for this brain tumor binary classification problem.

4.1.2 Patching

As mentioned earlier, WSIs are too large to be analyzed by neural networks as a whole, therefore small patches need to be extracted from them. One logical approach is to select a size on which trained pathologists can form diagnosis about the tumor, which is 1024 x 1024 pixels at 20x magnification ($0.262mm^2$) [2]. Convolutional neural networks that were trained on the ImageNet dataset however, usually expect images of size 224 x 224 as inputs, so the patches will need to be downscaled to this size for modelling. In the case of the recommended patch size, this would mean a 21x reduction of area, and a significant information loss, therefore the choice of a smaller patch size is more reasonable.

In [58], the highest accuracy was achieved when using 336 x 336 patches at 20x magnification. The authors of that paper experimented with 2 smaller patch sizes as well, but they performed worse. Building on these findings, in this thesis, two patch sizes are tried out, small (336 x 336) and large (672 x 672), where small and large patches have the area of $0.028mm^2$ and $0.113mm^2$ respectively.

745,691 patches of the small size, and 169,308 patches of the large size were extracted without overlap using the *patchify* Python package [56].

4.1.3 Stain normalization

The WSIs in the TCGA dataset were all stained with hematoxylin and eosin compounds that give different colors to different histological structures, but they were prepared at different clinics, where the exact practices might not match completely. Slide staining is prone to environmental conditions, and the final result is influenced by the duration of staining, pH balance, temperature, and other conditions [21]. All this introduces variations in the stain colors, that otherwise have no importance on distinguishing between GBM and LGG classes. In fact, it might make it harder for the neural network to learn the features that set the classes apart, if the stain colors are not normalized.

Even though stain normalization is considered an essential preprocessing step [21], not everyone follows this approach. Hou et al. [23] applied no stain normalization techniques, but randomly adjusted the amount of the two stains as a data augmentation step, thus making the model more robust.

In this thesis, the Vahadane algorithm [53] is used for stain normalization of patches, which is widely used, because it preserves biological structures well. Roy at al. [42] conducted quantitative and qualitative comparisons of the state of the art color normalization methods for histopathology images, and concluded that Vahadane's approach provided the best results in four different cancer datasets. The algorithm requires a target image to which all source images are normalized with no color distortion. The source images are decomposed into stain density maps that record the concentration levels of both stain colors, which hold important information about the biological structures. Then these density maps are combined with the stain color basis of the target image, this way only the colors are changed, their intensity (biological structure) remains the same. The exact methodology was introduced in [53], but it is not described in more detail in this thesis, because it is not the main focus of it.

The target image was chosen somewhat arbitrarily in a way that it included a large spectrum of colors present in the dataset. The algorithm is implemented in Python in the *StainTools* package [8]. Every patch is normalized individually, because it is recommended to first remove the white pixels of background, since they are not only composed of the two basis stain colors. When attempting to normalize WSIs prior to filtering and patching, certain artifacts were visible in the normalized version, which is why stain normalization takes its place at the end of the preprocessing pipeline. It is a very computationally heavy step that takes roughly 1 second for each patch.

4.2 Modelling

4.2.1 Training

The decision to use a ResNet50 model was made, similarly to [9], because it has already proven its capabilities in medical image processing. Residual networks were described in section 3.1.3, and different versions of them exist, but here the shallowest architecture is used that is implemented in Keras.

In this thesis, it is investigated whether a pre-trained network has an advantage over a CNN trained from scratch, so the predictive performance of both will be compared. It is also examined if data augmentation has a significance during training the models, therefore the final CNN will be chosen after careful consideration while answering these questions.

The pre-trained network is a network initialized with weights pre-trained on the ImageNet dataset, which means that the model can be expected to perform relatively well in the early stages of training. ImageNet contains 1000 image classes, so the last layer has 1000 units, therefore custom top layers need to be created to tailor the architecture for the GBM versus LGG binary classification problem (Figure 11). Specifically, an average pooling layer, a flatten layer, and two fully connected layers with dropout are added. The first dense layer has 100 nodes and ReLU activation, while the second one has 2 output nodes with softmax function, each corresponding to one of the classes. The dropout layers have a probability of 50% for randomly dropping connections, and they help with reducing overfitting by regularization, while allowing for Monte Carlo dropout, since they are also activated in test time. This way the predictions are not deterministic, the output is different every time a forward pass is run on a test image, and they form a distribution from which uncertainty can be deduced.

Data augmentation is a method for creating more training examples from the existing ones, so the model does not see the same images over and over again. Augmentation techniques include mirroring, rotating, shearing, cropping, and color jitter, among others. This helps reduce overfitting, which happens when the model learns the training examples too well, and cannot generalize to previously unseen data. Different techniques will be experimented with to determine which one works best in this specific case with histology images.

The idea of initializing the network's weights that were trained on a dataset, and then continuing the training on another dataset is called transfer learning [11], and its advantage is that one can leverage the already learned convolutional filters that recognise basic shapes and forms, thus making the fine-tuning fast and require less data. First, the added custom layers that implement the classification need to be trained, while the rest of the network is frozen, to avoid destroying the pre-trained weights. After this is achieved, the last two blocks of the ResNet50 model gets unfrozen and fine-tuned along with the added layers at the top. This way, the last blocks can learn filters specific to the tumor classification problem, while still building on the knowledge of the first three blocks that capture simpler patterns and shapes. Fine-tuning is done with a smaller learning rate to avoid completely overwriting the pre-trained weights.

The following three models are compared to determine whether pre-training gives an advantage:

- *Random weight initialization*: Weights are initialized randomly, and the model is trained with a learning rate of 10^{-6}.
- *Pre-trained with ImageNet*: Weights trained on the ImageNet dataset are loaded at the beginning. The added layers are trained first for 10 epochs with 10^{-4} learning rate, while the feature extraction part of the network is frozen. This is done to avoid destroying the pre-trained features by the large gradient updates the top layers must go through [11]. After the first phase of training, all the layers get unfrozen, and the whole model is trained for another 10 epochs with a lower learning rate (10^{-5}). The learning rate is lower to avoid changing the potentially valuable weights too much.

- *Pre-trained with ImageNet and fine-tuned*: Identical to the previous approach, except that not all the convolutional layers get trained in the second training phase. The first three convolutional blocks (Conv1, Conv2, and Conv3 in Figure 11) remain frozen. The idea behind this approach is that since the first blocks of the network are pre-trained to recognize basic features and shapes, it might be useful to keep them as they are. The later blocks are responsible for more complex patterns, where the weights learned from ImageNet might be less useful, since the histology dataset is quite different from ImageNet.

All three models apply the same data augmentation methods, which are horizontal and vertical mirroring.

After training these three models, their predictive performance is compared by ROC-curve analysis described in section 3.3.1. 30 forward passes are run on the whole validation set to obtain the Monte Carlo samples, and then the ROC-curves are constructed on patch level. Statistical significance of their difference is then examined, and the best one is chosen for further analysis.

After it has been established if it is beneficial to use pre-trained weights or not, the effects of data augmentation are investigated on the training process and the predictive performance. The following three models compete against each other:

- *No data augmentation*: No data augmentation is applied to the training images.
- *Mirroring*: Horizontal and vertical flips (mirroring) are applied to the training images.
- *Mirroring, rotating*: The training images are augmented with horizontal, vertical flips, and random rotations between 0-90 degrees. When rotating, some points outside the input image are included, here the missing parts are supplied by reflecting the input.

Figure 13 shows the data augmentation techniques used in this thesis.

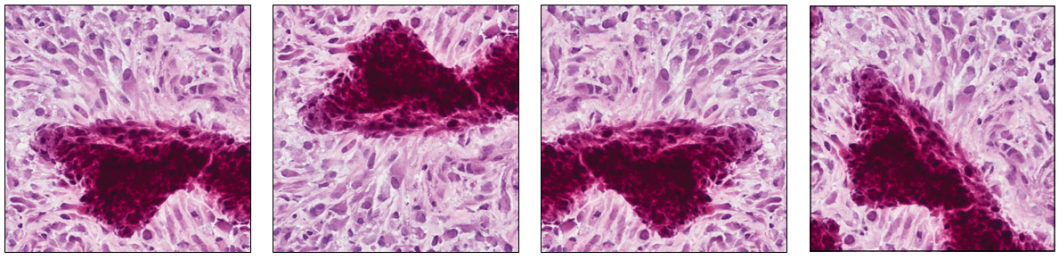

Figure 13: Examples of data augmentation used in this thesis. The original image (left) is mirrored vertically and horizontally (2nd and 3rd from the left). 45 degree rotation with reflect fill mode, so the points outside the boundaries of the original image are filled with mirroring.

All models are optimized with the stochastic gradient descent algorithm. The batch size is 64 in all cases, which is necessary to be able to fit all the information in the GPU memory. In every epoch, all the training images are used for weight update, and all the validation images are used for measuring the generalization accuracy.

A computer running CentOS Linux 7 operating system with Intel(R) Core(TM) i5-9600K processor, 64 GB memory and 2 Nvidia GeForce RTX 2080 Ti graphics cards was used for training with Tensorflow 2.2.0 and Keras 2.4.3.

4.3 Slide aggregation

The convolutional neural network described above predicts a class label for every single patch, but the aim of this thesis is to classify whole slide images, which can be interpreted as bags of patches, usually a few thousands of them. In the optimal scenario, ground truth would be available on pixel level, that way one could be sure that the true labels of the patches are correct in all cases. It would require tremendous effort from experts to annotate on such low granularity, however, so this approach is generally not feasible in real life. In such cases, multiple instance learning approaches can be useful, described in section 3.2.1, to combine the patch-level predictions to slide-level predictions. In the next subsections, five slide aggregation techniques are presented, the performance of which will be compared later on in this thesis. At this stage in the analysis, T soft predictions have been obtained from the CNN model, where T is the number of Monte Carlo samples (stochastic forward passes), for every patch in the dataset.

Uncertainty is measured as the standard deviation of the T Monte Carlo samples multiplied by 2, to stretch the interval to between 0 and 1, making interpretation easier. Obtaining the *certainty estimates* or *weights* on a scale of 0 to 1 is a simple linear mapping afterwards:

$$certainty = 1 - uncertainty.$$

It is important to note that since GBM is a more malignant grade of tumor than LGG, expert pathologists classify a slide as glioblastoma if any area in the slide can be diagnosed as such. Keeping this in mind, it might be straightforward to simply predict GBM for every slide that has at least one GBM predicted patch in it. This would, in fact, be the optimal solution, if one could assume that each training patch had the correct ground truth label assigned to it. Since the training slides do not have annotations on such low granularity, one cannot make this assumption. As mentioned earlier, it is entirely possible (and expected) that some patches have incorrect true labels when training, because it is possible that a patch only contains lower grade glioma in a slide that is otherwise diagnosed as glioblastoma, or it might just be a tumor-free area of the tissue. Expecting the CNN to correctly classify every patch under these circumstances is unrealistic, therefore its predictions should not be taken for granted. The layer of aggregation could be a way to smooth out the model's errors and provide a more robust classification pipeline, at the expense of abandoning the pathologists' way of diagnosis. The reasoning is that it is more important to arrive at the correct solution at the end (correct classification of test slides) than to use the same method as the experts. The other option would be to ask them to manually annotate slides at a much lower level, but this is an arduous task that takes too much time and resources.

4.3.1 Majority voting

The simplest method for slide aggregation is majority voting. In this case, the uncertainty of the patch predictions is ignored, and the predicted class of every patch is determined by the mean of the Monte Carlo samples. The probability of a slide belonging to the GBM class is the ratio of the patches classified as GBM. The probability is larger than 0.5 if more than half of the patches are predicted GBM.

4.3.2 Logistic regression

The aim is to learn a function that maps patch predictions to slide predictions, so implementing a second-level classifier model after the CNN is a reasonable step to take.

Campanella et al. [9] conducted a large-scale experiment, in which they experimented with random forests and Recurrent Neural Networks (RNNs) as second-level classifiers (the RNN was trained on the extracted features of the first-level CNN, not its class predictions) for various types of tumor classification problems.

Hou et al. [23] compared 14 aggregation methods, among which were logistic regression and support vector machine models trained on the output of a CNN for multi-class glioma classification. Two setups of the logistic regression variant achieved the two highest accuracy scores, so it seems promising to implement it in this study, as well.

The logistic regression is fitted on the validation set after the CNN training is done to avoid over-fitting. The features are derived from the CNN predictions, taking into account their uncertainty estimates or rather, their weights, as defined previously. Statsmodels' Python implementation [43] was used, and the following features were included for fitting the model:

- Ratio of patches predicted GBM with at least 95% certainty.
- Ratio of patches predicted LGG with at least 95% certainty.

4.3.3 Spatial smoothing

The assumption is that one tumor grade does not occur sparsely all over the tissue, but within well-defined borders that takes up a larger area. It is unlikely that inside an area of glioblastoma, lower grade glioma spots are present (and vice versa), therefore, as a post-processing step, spatial smoothing is applied on the slides to create these larger areas of homogeneous tumor. This is achieved by covering the slide in non-overlapping windows of the same size, and assessing each window independently in terms of the more likely tumor grade for that area.

In this approach, multiple criteria need to be fulfilled in order to flip the class prediction of some patches in a window. A window size of 9 x 9 patches is used with possible missing patches treated as unknowns, where the majority class has a chance of overwriting the labels of the minority class. Minority patches are only flipped, if at least one patch from the majority class has a higher certainty (weight) than the threshold, and all patches from the minority class have lower certainties than the threshold. This threshold value is considered a hyperparameter of the method, therefore it is optimized on the validation set with grid search. The algorithm uses majority voting at the end, and outputs the probability of GBM for every slide.

With this method, hopefully some of the patch prediction errors made by the CNN are to smoothed out by taking into account their spatial location, and examining what label is most likely for them given this information. Figure 14 illustrates spatial smoothing.

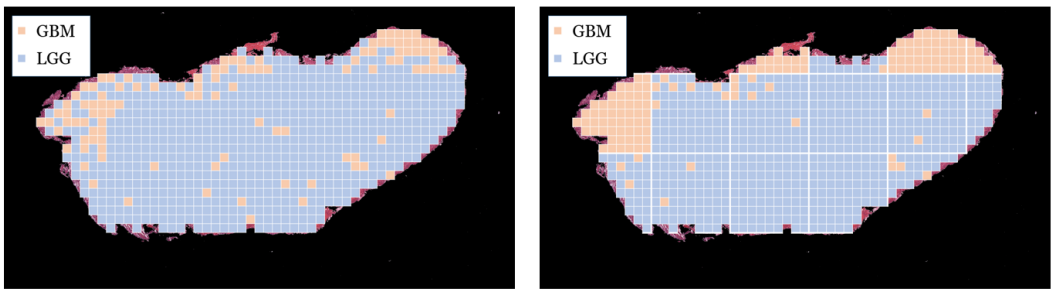

Figure 14: Illustration of spatial smoothing, before (a) and after (b). The image is divided into windows of 9 x 9 patches. The same prediction class is assigned to all the patches in a window, if the majority class inside the window has at least one patch with a certainty higher than the threshold, and all the patches from the minority class have lower certainties than the threshold. The certainties are not depicted on this illustration, but the conditions were only met, where the whole window was assigned the same class.

4.3.4 Standard MIL assumption

The most common method in the multiple instance learning framework was introduced in [14], and is generally referred to as the standard assumption. This states that a bag is considered positive (in this case GBM), if and only if at least one of its instances is positive. This approach is closest to how a pathologist makes a diagnosis.

Given that the CNN patch-level predictions are not expected to be completely accurate, this assumption might not work well in this case. Uncertainty is not taken into account here, the patch predictions are simply the means of the Monte Carlo samples. The probability of a slide belonging to the GBM class is equal to the highest probability among all of the patches in that slide.

4.3.5 Weighted collective MIL assumption

The problem with the standard assumption described above is that only one patch can decide the label of the slide, whereas the collective assumption of the multiple instance learning framework lets all instances contribute equally to the slide label prediction [57]. This assumption sees a bag as samples of a probability distribution that describe the population of that bag, and instances are assumed to be assigned labels according to some probability function. This function in this case is the CNN model itself with softmax output. The bag-level class probability function is then simply the expected class value of the population of that bag. Let $c \in \{0, 1\}$ be the class label, u an instance in the dataset U and b a bag. Then the probability of class c given b is

$$p(c|b) = E_U[p(c|u)|b] = \int_U p(c|u)p(u|b)du.$$

The probability distribution of the bag $p(u|b)$ is usually not known, so it is substituted by the sample of instances present in the bag. Therefore:

$$p(c|b) = \frac{1}{n_b} \sum_{i=1}^{n_b} p(c|u_i),$$

where n_b is the number of instances in the bag [17].

But in the case of this thesis, the CNN is not equally certain about the patch predictions (the output class probabilities), so the aim is to assign larger weight to those patches that the model is more certain about. The weighted collective assumption [17] makes this possible by incorporating a weight function into the collective assumption:

$$p(c|b) = \frac{1}{\sum_{i=1}^{n_b} \beta(u_i)} \sum_{i=1}^{n_b} \beta(u_i)p(c|u_i),$$

where $\beta(u) : U \to \mathbb{R}^+$ is the weight function determining the influence an instance has on the bag-level label. In this case, this corresponds to the weight (or certainty) derived from the distribution of the Monte Carlo predictions. The algorithm outputs the probability of class 1 (GBM) for each slide.

4.4 Evaluation methods

Receiver operating characteristic (ROC) curves and the area under the curve (AUC) are among the most popular evaluation methods for classification models. One can compare AUCs of different models statistically to test if they are significantly different with the widely used DeLong's algorithm [13], which has been employed to compare AUCs of different slide aggregation techniques in weakly supervised histology image classification in [9].

DeLong's test for comparing the AUCs of correlated ROC curves [13] is implemented in R in the pROC package [40]. Only comparison of two curves is implemented, therefore pairwise comparisons will be conducted of the methods.

5 Results

In this chapter, the results of the thesis are presented from preprocessing to final evaluation.

5.1 Preprocessing

During the preprocessing step, the white background around the tissue is removed, patches of two different sizes (336 x 336 and 672 x 672 pixels) containing at least 95% tissue are extracted and stain normalized (Figure 15).

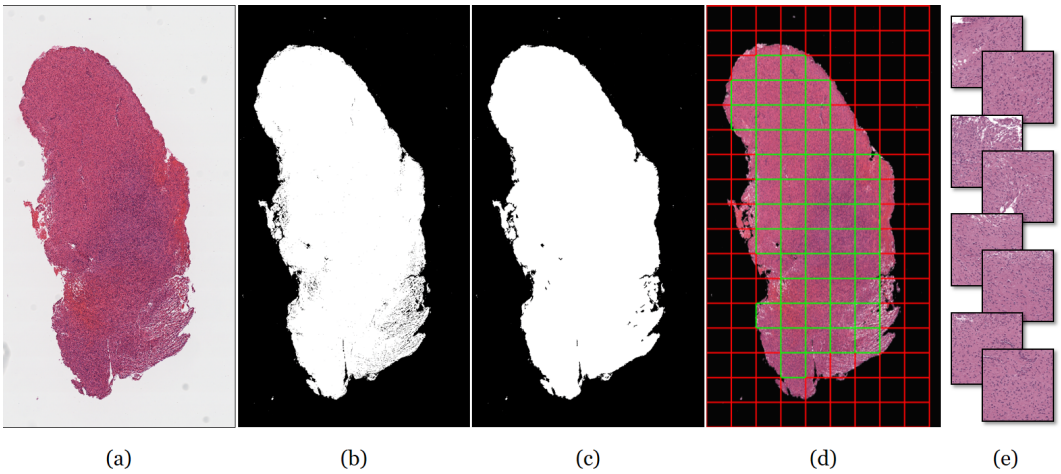

Figure 15: Preprocessing pipeline. The white background from the original image (a) is removed first (b), then small holes inside the tissue area are filled (c). After that, the resulting binary mask is applied on the original image, and patches that contain at least 95% tissue are extracted (d). The obtained patches are then stain normalized (e).

For stain normalization, the Vahadane algorithm mentioned earlier is used, resulting in patches having similar color schemes (Figure 16).

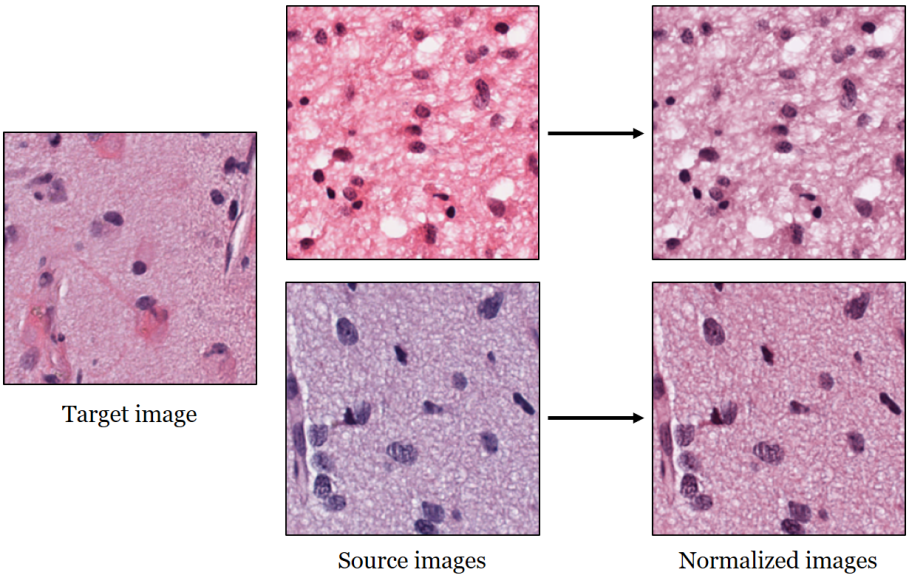

Figure 16: Stain normalization using the Vahadane method.

The resulting preprocessed datasets of small and large patches include 745,691 and 169,308 patches in total, respectively. The split between training, validation and test sets can be observed in Tables 1 and 2.

	GBM		LGG		Total	
	slides	patches	slides	patches	slides	patches
training	42	167,183	42	168,346	84	335,529
validation	13	48,804	13	51,362	26	100,166
test	42	151,336	42	158,660	84	309,996
Total	97	367,323	97	378,368	194	745,691

Table 1: Dataset of small patch size (336 x 336).

	GBM		LGG		Total	
	slides	patches	slides	patches	slides	patches
training	42	38,215	42	38,716	84	76,931
validation	13	11,055	13	11,746	26	22,801
test	42	33,314	42	36,262	84	69,576
Total	97	82,584	97	86,724	194	169,308

Table 2: Dataset of large patch size (672 x 672).

5.2 Modelling

In this section, the results of the network training and the process of model selection are presented. The goal is to find out if pre-trained networks have an advantage over networks trained from scratch, and also if data augmentation techniques improve the performance. Moreover, it is also investigated if there is a difference between using small and large patches, so the candidate models are trained on both patch sizes.

5.2.1 Pre-training comparison

First, three models are trained with different weight initialization and pre-training setups, as described in the Methods chapter. The training curves can be observed in Figure 17 for small patches and Figure 18 for large patches.

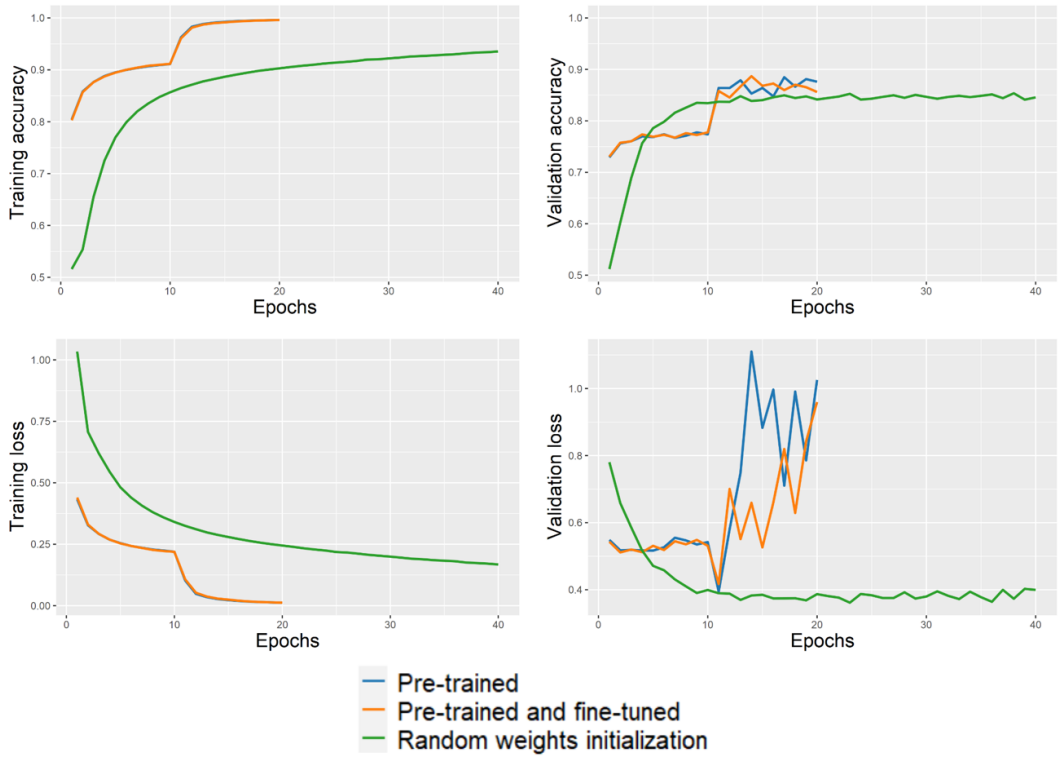

Figure 17: Model training and accuracy curves (small patches, pre-training comparison).

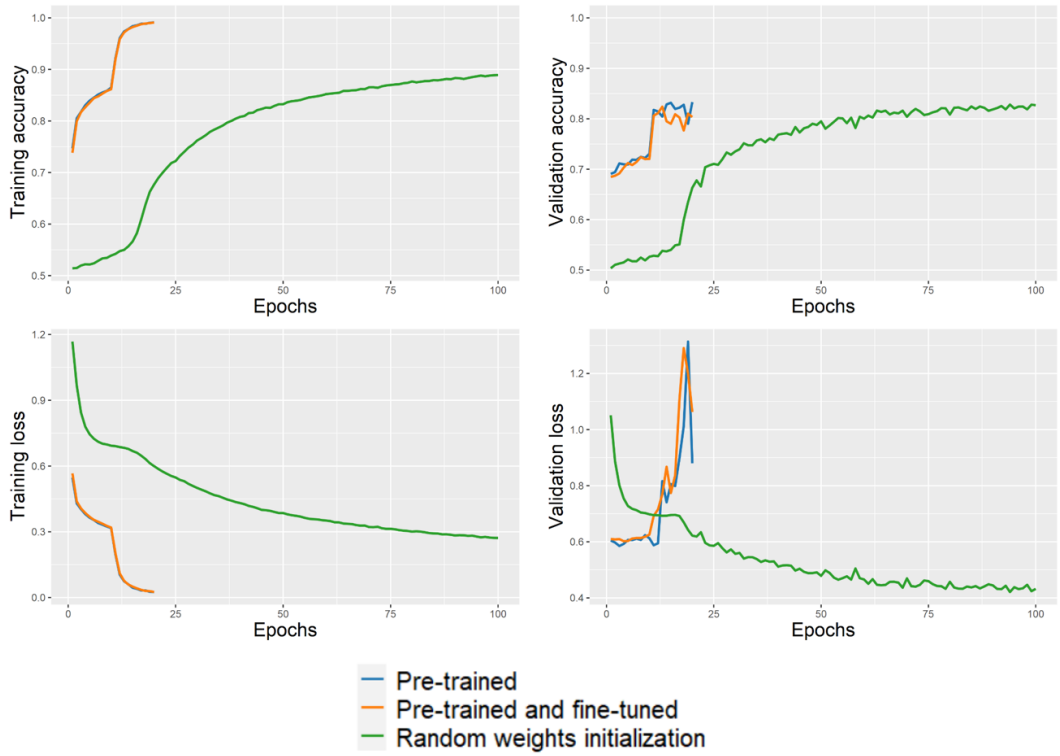

Figure 18: Model training and accuracy curves (large patches, pre-training comparison).

The ROC curves for the trained models are plotted in Figure 19. The AUCs are compared with DeLong's test for two correlated ROC curves (explained in section 3.3.1), where the null hypothesis is that the competing areas are equal. The p-values of the tests can be seen in Tables 3 and 4 along with the AUC values.

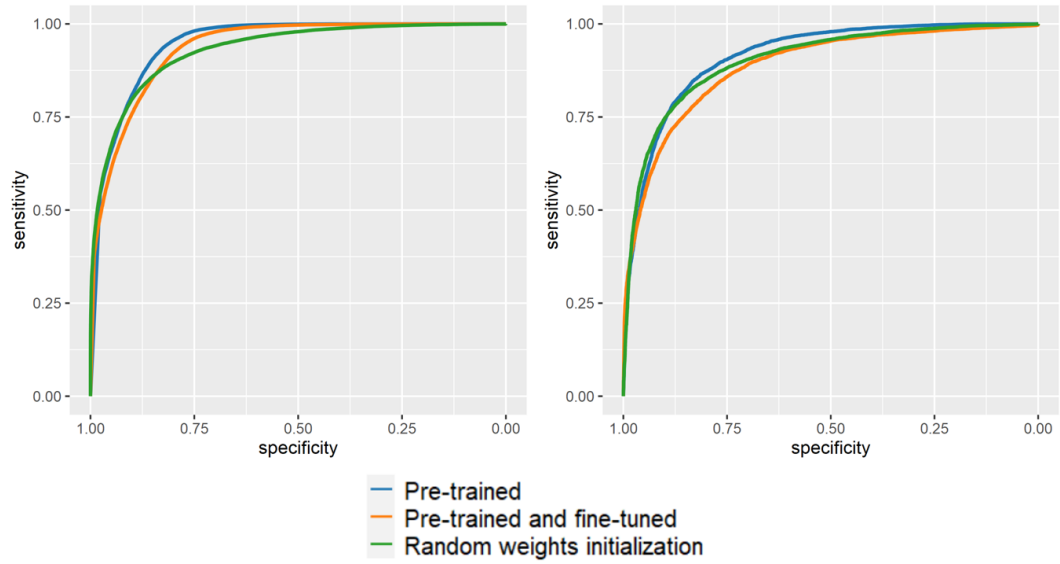

Figure 19: ROC curve comparison of different pre-training approaches on small (left) and large patches (right).

	Pre-trained	Pre-trained & fine-tuned	Random	AUC
Pre-trained	1			**0.9463**
Pre-trained & fine-tuned	< 2.2e-16*	1		0.9358
Random	< 2.2e-16*	2.5e-09*	1	0.9309

Table 3: P-values of Area Under the Curve comparisons of different pre-training approaches in small patches. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance.

	Pre-trained	Pre-trained & fine-tuned	Random	AUC
Pre-trained	1			**0.9143**
Pre-trained & fine-tuned	< 2.2e-16*	1		0.8873
Random	4.4e-06*	1.8e-10*	1	0.9036

Table 4: P-values of Area Under the Curve comparisons of different pre-training approaches in large patches. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance.

For both patch sizes, the models with pre-trained weights resulted in the highest AUC, while being significantly different from the competing approaches, because the p-values of the tests were lower than 0.05. Therefore, the pre-trained models are chosen and the model selection process continues. Ideally, all combinations should be compared at the whole slide level, but training that many models would be too time consuming.

5.2.2 Data augmentation comparison

In this step, three different data augmentation methods are experimented with, and their performance is compared in the same way as before. The training curves are in Figures 20 and 21.

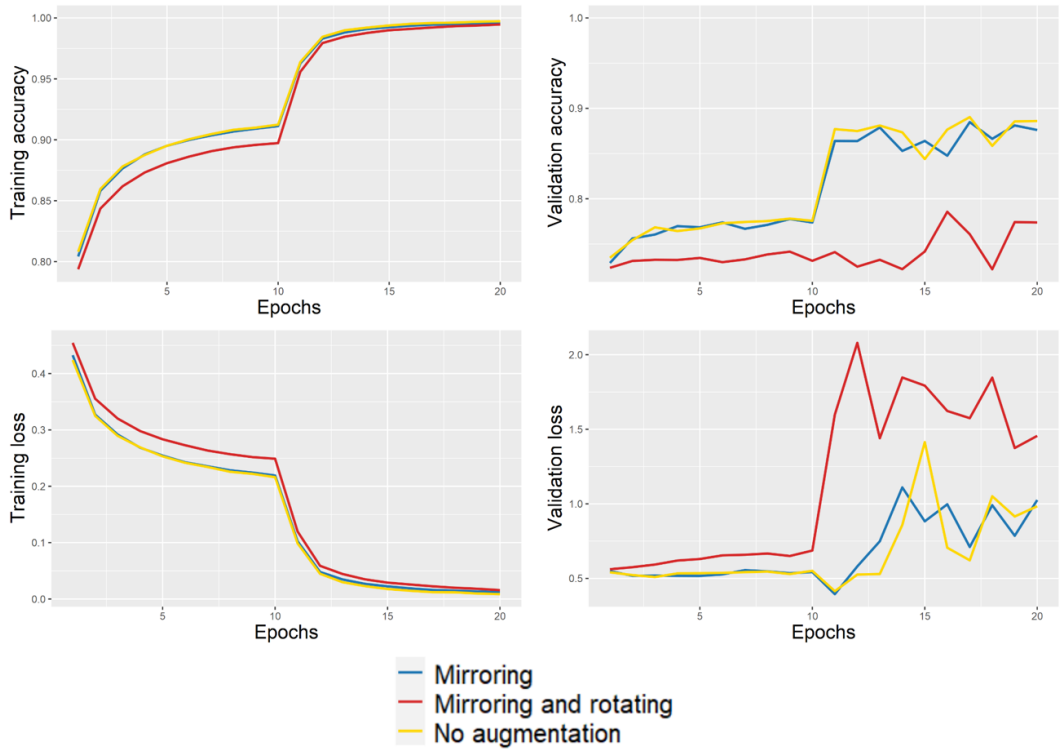

Figure 20: Model training and accuracy curves (small patches, data augmentation comparison).

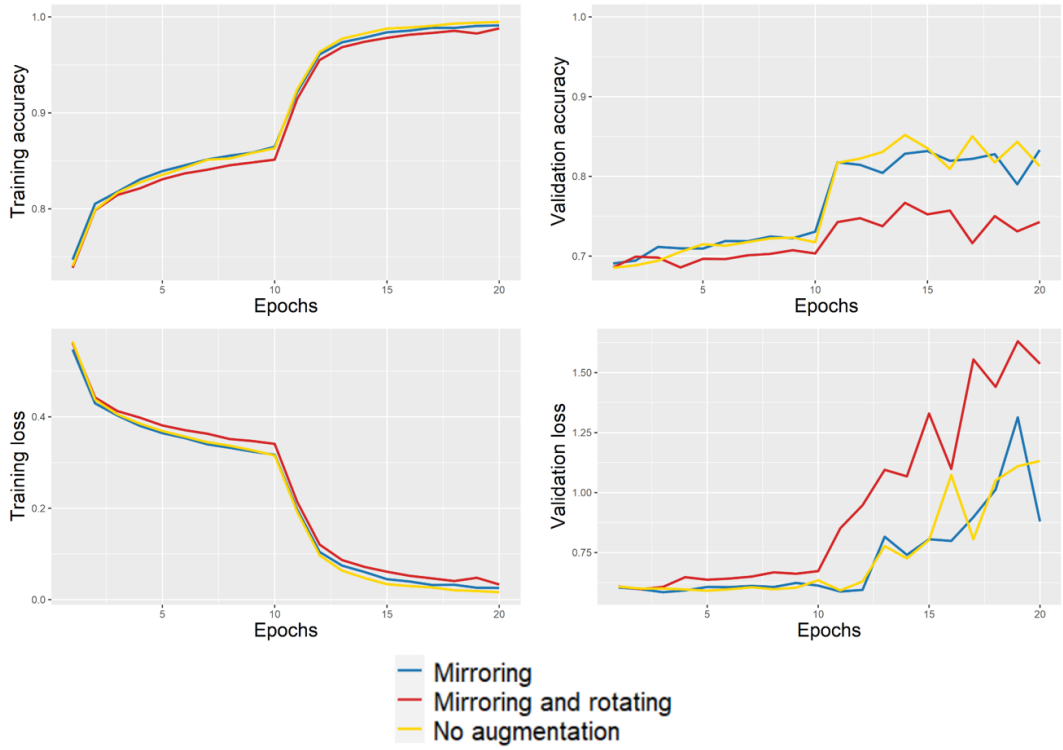

Figure 21: Model training and accuracy curves (large patches, data augmentation comparison).

The ROC curves can be observed in Figure 22, and the statistical tests' p-values in Tables 5 and 6.

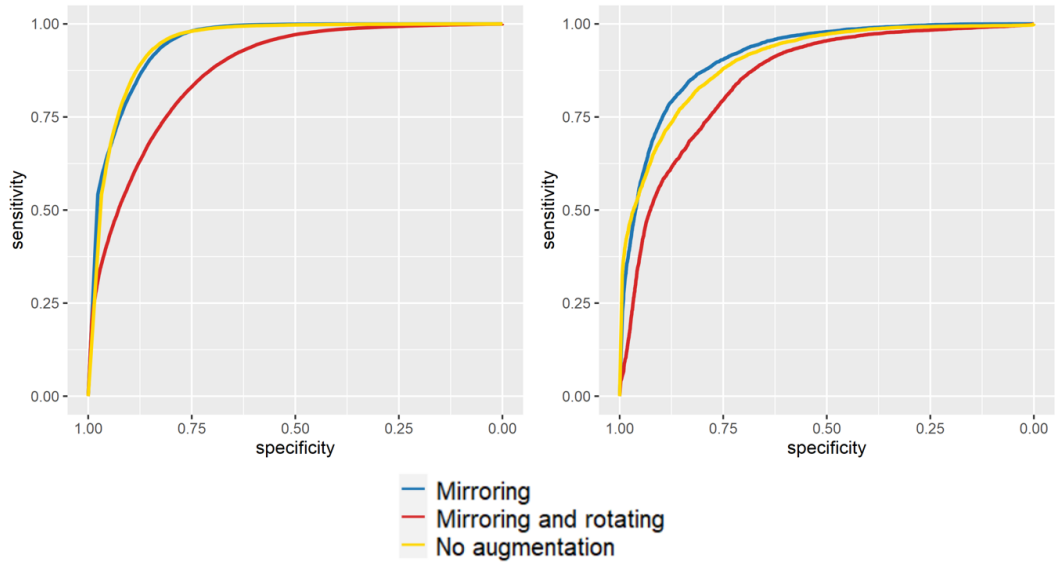

Figure 22: ROC curve comparison of different data augmentation approaches on small (left) and large patches (right).

	Mirroring	Mirroring & rotating	No augmentation	AUC
Mirroring	1			**0.9463**
Mirroring & rotating	< 2.2e-16*	1		0.8735
No augmentation	0.061	< 2.2e-16*	1	0.9454

Table 5: P-values of Area Under the Curve comparisons of different data augmentation approaches in small patches. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance.

	Mirroring	Mirroring & rotating	No augmentation	AUC
Mirroring	1			**0.9143**
Mirroring & rotating	< 2.2e-16*	1		0.8535
No augmentation	< 2.2e-16*	< 2.2e-16*	1	0.9024

Table 6: P-values of Area Under the Curve comparisons of different data augmentation approaches in large patches. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance.

In both cases, the models with mirroring augmentation had the highest AUC, but in the model with the small patches, the difference between that and no augmentation was not significant. Since mirroring still had higher AUC than no augmentation, and because it is considered good practice to use data augmentation, I chose mirroring augmentation for small patches as well.

The final models for both patch sizes are the ones with pre-trained weight initialization and mirroring data augmentation. The best performing model achieved 0.9463 AUC in small patches, and 0.9143 AUC in large patches, therefore the best AUC on patch level was 0.9463.

5.2.3 Prediction visualizations

In this section, some of the predictions on the whole slide level are visualized by gathering the predictions of the patches belonging to the slide.

The whole slide in Figure 23 is labelled as GBM. 80% of the small patches were predicted GBM by the model, but this number was only 46% with the large patches, meaning that the model with larger patches misclassified this slide as a whole. Both models recognized roughly the same areas as GBM, but larger patches were more often classified as LGG. Most patches were classified with a relatively high probability and very low uncertainty. Especially small patch predictions had uncertainty close to 0, some of the large ones had around 0.5 uncertainty.

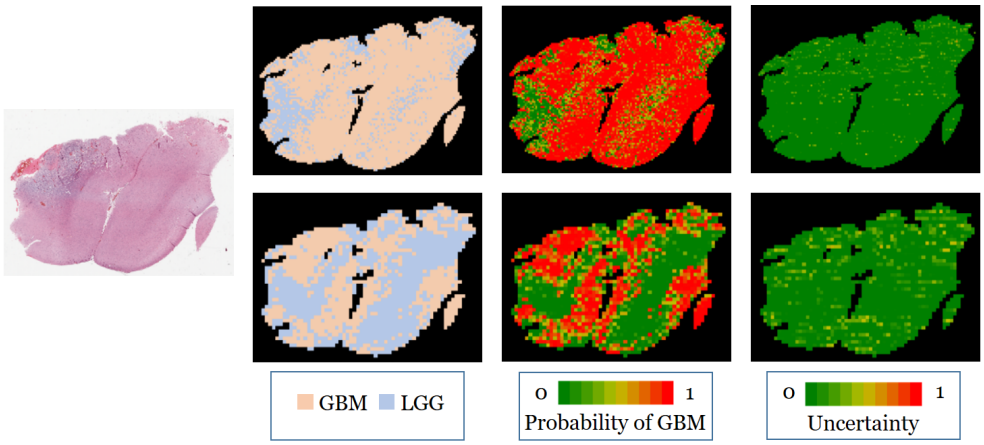

Figure 23: Prediction visualization 1. Original GBM whole slide image (left), predicted classes (2^{nd} column), predicted probabilities (3^{rd} column), prediction uncertainties (4^{th} column) on small (top row) and large patches (bottom row).

Figure 24 shows an example of an LGG slide incorrectly classified as GBM by majority voting, because more than half of the patches were predicted GBM. GBM patches were mostly predicted with a high probability, whereas some LGG patches had around 50% prediction probabilities. Prediction uncertainties were quite low.

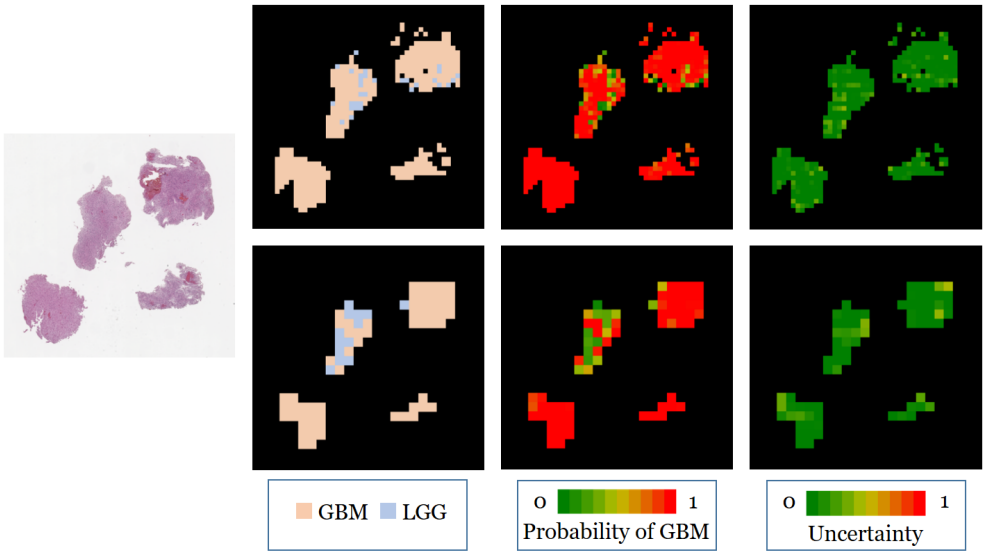

Figure 24: Prediction visualization 2. Original LGG whole slide image (left), predicted classes (2^{nd} column), predicted probabilities (3^{rd} column), prediction uncertainties (4^{th} column) on small (top row) and large patches (bottom row).

An LGG slide that was correctly classified can be seen in Figure 25. Smaller areas of GBM were identified by both models especially in the lower left corner, but the overall majority voting prediction was comfortably LGG.

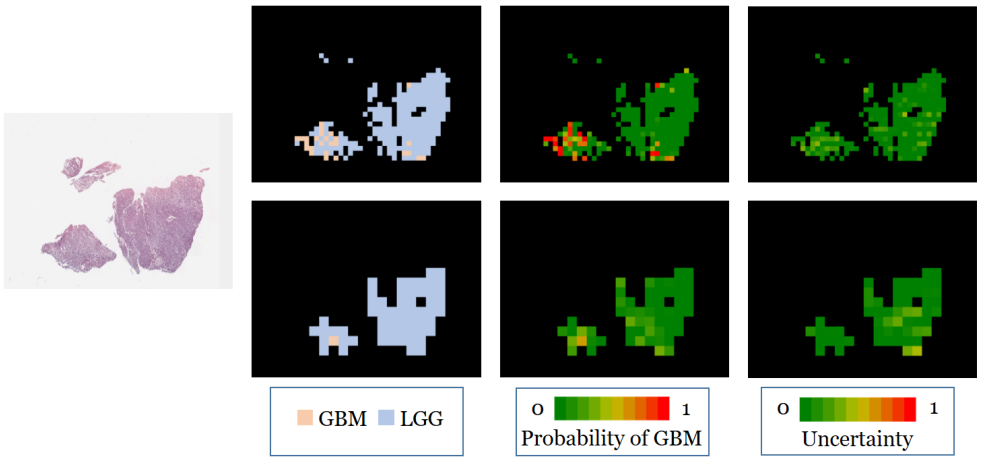

Figure 25: Prediction visualization 3. Original LGG whole slide image (left), predicted classes (2^{nd} column), predicted probabilities (3^{rd} column), prediction uncertainties (4^{th} column) on small (top row) and large patches (bottom row).

5.3 Slide aggregation

In the final part of presenting the results, the performance of the different slide aggregation methods are examined.

The logistic regression model trained on the validation set with two features resulted in the following models for the two patch sizes (Tables 7, 8). The coefficients are plausible, the higher the ratio of certain GBM patches, the higher the probability of the slide being GBM,

and the higher the ratio of certain LGG patches, the lower this probability, but none of the features were statistically significant, and their 95% confidence intervals are very wide.

	coefficient	p-value	95% confidence interval lower	upper
intercept	-4.37	0.61	-21.05	12.31
GBM95 ratio	7.78	0.43	-11.40	26.95
LGG95 ratio	-7.61	0.88	-105.52	90.31

Table 7: Coefficients of the logistic regression in small patches.

	coefficient	p-value	95% confidence interval lower	upper
intercept	-1.35	0.60	-6.34	3.64
GBM95 ratio	5.98	0.16	-2.29	14.25
LGG95 ratio	-6.57	0.48	-24.70	11.56

Table 8: Coefficients of the logistic regression in large patches.

The ROC curves of the five aggregation techniques are shown in Figure 26, and the results of the statistical tests in Tables 9, 10 and 11. The ROC curves at slide level are considerably less smooth than at patch level, this is because of the much smaller number of examples.

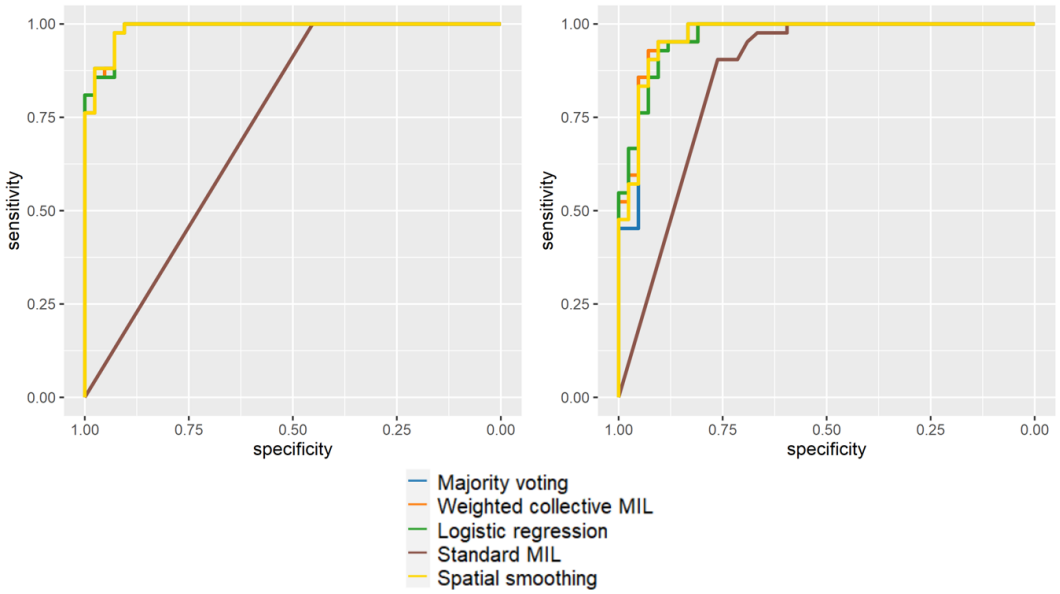

Figure 26: ROC curve comparison of different slide aggregation methods on small (left) and large patches (right).

	Maj.vote	Weighted MIL	Log.reg.	Std. MIL	Smoothing	AUC
Maj.vote	1					**0.9881**
Weighted MIL	0.4795	1				0.9875
Log.reg.	1	0.754	1			**0.9881**
Std. MIL	5.2e-12*	5.6e-12*	4.9e-12*	1		0.7262
Smoothing	1	0.480	1	5.2e-12*	1	**0.9881**

Table 9: P-values of Area Under the Curve comparisons of different slide aggregation approaches in small patches. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance.

	Maj.vote	Weighted MIL	Log.reg.	Std. MIL	Smoothing	AUC
Maj.vote	1					0.9649
Weighted MIL	0.334	1				**0.9694**
Log.reg.	0.7599	0.6014	1			0.9671
Std. MIL	0.0062*	0.0027*	0.0022*	1		0.8608
Smoothing	0.4278	0.5029	0.9109	0.0036*	1	0.9677

Table 10: P-values of Area Under the Curve comparisons of different slide aggregation approaches in large patches. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance.

Small \ Large	Maj.vote	Weighted MIL	Log.reg.	Std. MIL	Smoothing	AUC
Maj.vote	0.2355	0.2537	0.1606	0.0003*	0.2415	**0.9881**
Weighted MIL	0.2495	0.2717	0.1745	0.0004*	0.2578	0.9875
Log.reg.	0.2399	0.2606	0.1661	0.0004*	0.2481	**0.9881**
Std. MIL	7.5e-10*	1.6e-10*	1.5e-10*	0.0003*	2.7e-10*	0.7262
Smoothing	0.2355	0.2537	0.1606	0.0003*	0.2415	**0.9881**
AUC	0.9649	**0.9694**	0.9671	0.8608	0.9677	

Table 11: P-values of Area Under the Curve comparisons of different slide aggregation approaches and patch sizes. The p-values are calculated for DeLong's test for two correlated ROC curves, where the null hypothesis states that two AUCs are equal. P-values lower than 0.05 are marked to highlight significance. Slide aggregations on small patches (rows) are compared against slide aggregations on large patches (columns).

The AUC values are quite similar and not significantly different, except for the standard MIL assumption method, therefore it is difficult to make a statistically sound decision. The best performing slide-level models achieved 0.9881 AUC on small patches and 0.9694 AUC on large patches. The statistical insignificance might come from the fact that only 84 test slides were used, which did not provide enough evidence for DeLong's tests. To test this assumption, power analysis is conducted between the best approaches from the two patch sizes. The methods with the highest AUC are selected as the two best models, which is the weighted collective MIL assumption for large patches and a three-way tie between majority

voting, logistic regression and spatial smoothing for small patches. Since the features in logistic regression were not significant (Table 7), and spatial smoothing is just majority voting with postprocessing steps that did not give an advantage, the simple majority voting method is chosen.

Based on DeLong's test, one cannot say that the AUCs of the best approaches from each patch size are significantly different, because the p-value is 0.2537 (Table 11). This means that one cannot reject the null hypothesis that states that the two AUCs are equal. The AUC of the majority voting method on small patches is higher (0.9881), but perhaps there is not enough evidence to be significantly different from that of the weighted collective assumption MIL method on large patches (0.9694).

When the null hypothesis (two AUCs are equal) cannot be rejected, it means that the sample contained insufficient amount of evidence to conclude that the two AUCs are different, therefore the results of the hypothesis tests are not statistically significant. The two AUCs might in fact be equal, but this conclusion cannot be made based on this experiment. On the other hand, when the p-value of the hypothesis test is lower than a significance level (0.05 in this case), this means that the null hypothesis can be rejected, the sample data favours the alternative hypothesis stating that the two AUCs are significantly different. The p-value can be interpreted as the probability of a sample statistic that is at least as extreme as the sample statistic observed in this experiment, when the null hypothesis is assumed to be true. If the p-value is 0.01 for example, and the null hypothesis is assumed to be true (two AUCs are equal), only 1% of the studies would obtain a sample statistic equal to or larger than the statistic obtained in this experiment, because of random sampling error.

Statistical power analysis is carried out to investigate the required sample size (number of test slides) at multiple levels of significance and power. Examining Figure 27, it seems that at least 200 samples (197 to be exact) are necessary for the two AUCs to be significantly different at 0.05 significance level and 0.8 power, which are standard threshold values when conducting power analysis. Higher statistical power means a higher probability of correctly rejecting the null hypothesis, which can be achieved by increasing the sample size. The exact method of the power analysis is not described in this thesis. It was carried out using the pROC package in R [40].

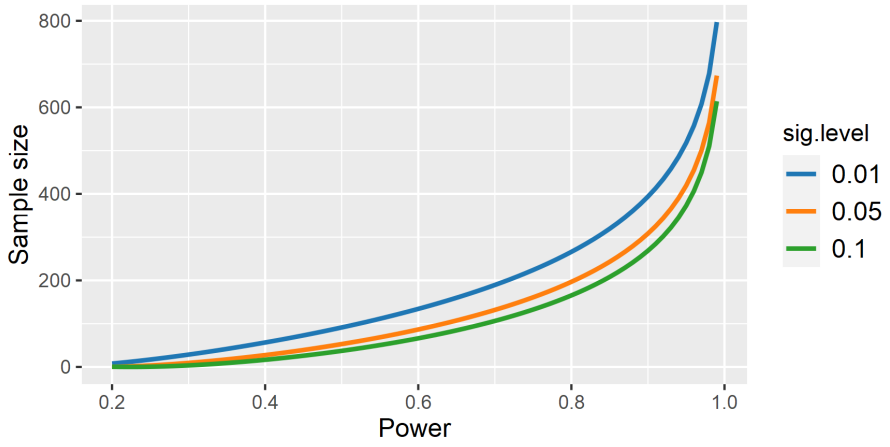

Figure 27: Power analysis: The number of required samples at different significance and power levels.

In light of this, the majority voting with small patches is selected as the final model keeping in mind that more test data is required to reach statistical significance. The confusion matrix is Table 12, and the derived metrics are in Table 13. This approach achieved 0.9881

AUC on slide level, 91.7% accuracy, 92.3% F_1-score, 100% sensitivity, 85.7% precision and 83.3% specificity on the test set.

	True LGG	True GBM
Predicted LGG	35	0
Predicted GBM	7	42

Table 12: Confusion matrix of the best performing slide classification model with majority voting and small patches.

Accuracy	F_1-score	Sensitivity	Precision	Specificity
0.917	0.923	1	0.857	0.833

Table 13: Evaluation metrics of the best performing slide classification model with majority voting and small patches.

6 Discussion

In this chapter, the results that were presented before are discussed, analyzed, and compared to the results of related works. The adequacy of the methods used is also discussed, and possible improvements are identified.

6.1 Results

6.1.1 Preprocessing

The preprocessing pipeline consisted of four steps: removing the white background, filling small holes inside the tissue area, extracting patches, and stain normalizing patches. The latter two steps were applied with two different patch sizes: 336 x 336 (small) and 672 x 672 (large) pixels, this way the pipeline resulted in two separate datasets. An example of the results of this process is shown on Figure 15, and it can be observed that the background removal together with the hole filler worked very well, the result is exactly as expected. The correct patches with more than 95% tissue were extracted from the filtered image and they were stain normalized with the Vahadane method [53] that resulted in patches looking very similar in color (Figure 16). This way it was possible to bridge the gap between the varying staining processes conducted in labs.

6.1.2 Modelling

In this thesis, convolutional networks with ResNet50 architecture were trained to learn a patch-level classifier. Separate models were trained for small and large patches and the best ones were chosen after comparing three different weight initialization and pre-training methods, and three data augmentation techniques.

In the first round, models trained from scratch were compared with ImageNet pre-trained weights with and without freezing layers. The two CNN networks with pre-trained weights performed very similarly in both patch sizes, implying that freezing the first three convolutional blocks in the ResNet50 model does not make a lot of difference (Figures 17 and 18). Or in other words, even if the network can update all of its weights, the layers on lower levels of abstraction remain the same. These layers are mostly responsible for recognizing basic patterns and shapes, which seem to be applicable to histology images as well, even though

they were trained on the very different ImageNet dataset. It is interesting to see the jump in accuracy after the 10th epoch in the two pre-trained models, because this is the point where the second training phase started, by training the convolutional layers (either some or all of them), as well. The training slowed down relatively fast after this, indicating that the models managed to update their weights to suit the histology problem quickly. Unexpectedly, the validation losses started to increase after epoch 10, even though the validation accuracy was still increasing, too. This might be explained by that the model produced a large loss for some of the validation images, which increased the overall loss, as well. The training accuracy and loss curves show continuous growth and decrease respectively, but the training seems to have been faster with small patches. This is probably because there were more than four times as many training patches from the small size than from the large size, and the weights were updated using all available training patches, so there was more information available from the small patch dataset.

The models with randomly initialized weights, however, behaved very differently. In both patch sizes, the training and validation accuracy kept increasing and the losses decreasing. The difference between using different sized patches was the speed of learning, which seemed to be much faster and more monotonous with small patches, and the explanation probably comes from the unequal amount of training examples again. The accuracy started from around 50%, which is exactly what is expected with random initialization in a binary classification problem. This was a sharp contrast with the pre-trained models that had a clear initial advantage due to their pre-loaded weights, and also the reason why the randomly initialized models were allowed to train for more epochs to give them a fair chance of learning proper weights. Still, the randomly initialized models did not quite reach the performance of the pre-trained ones, perhaps the training got stuck in a local optimum, and was unable to get out of it due to its low learning rate. A larger learning rate on the other hand, would have resulted in very unstable training.

The decision of choosing the best of the three models was made based on comparing each of the model's AUC value on the validation set, and since the pre-trained model without freezing layers had the highest AUC in both small and large patches (0.9463 and 0.9143 respectively), that model was selected. The differences of AUCs were statistically significant, too, despite them being very small, because there was a lot of evidence (validation patches) for the DeLong's test, resulting in very small p-values.

The second round of model selection consisted of comparing three different data augmentation methods. Using mirroring improved performance by a very small margin compared to using no augmentation at all, but adding random rotations clearly had a negative effect in both small and large patches. The validation accuracy stayed well below the other two approaches and the validation loss increased more. This is surprising, because data augmentation is specifically supposed to help with overfit, so lower validation loss was expected when using more augmentation. The reason probably is not that rotating patches results in unrecognizable patterns, because histology images in general do not have orientation. The more likely explanation is that since points outside the boundaries of the original image have to be filled in somehow, this process resulted in very different images from the validation set (the fill mode used was reflect, which mirrors the rotated image to fill out the new image boundaries).

In small patches, the difference between the AUCs of mirroring and no augmentation was not significant, as opposed to the difference in large patches, which makes sense, because data augmentation helps more when there are fewer training examples available. The best performing approach was mirroring in both cases, and because it is considered a better practice to use some augmentation, this model was selected as the final CNN.

6.1.3 Slide aggregation

After finding the best CNN model to classify individual patches, the best approach was investigated for combining patch-level predictions into a slide-level one. A test set consisting of 84 slides was used for this purpose. Four of the five methods resulted in very similar slide predictions for both patch sizes. The fifth one, the standard MIL assumption performed much worse, but this aligns with the prior expectations, because it is very similar to how a pathologist makes a diagnosis, which works well in real life with trained experts, but not so well with an artificial network trained on weakly annotated images. The best models trained on small patches achieved AUCs of 0.9881, but the highest AUC was quite high even on large patches (0.9694).

The pairwise differences between the slide aggregation approaches were not statistically significant even between different patch sizes (Table 11), except for the standard MIL assumption method, but if the test set had consisted of at least 197 slides, the difference would have been significant.

It is important to note that multiple statistical tests were carried out during this investigation, but there was no protection against the inflation of type I errors (the probability of rejecting a null hypothesis when it is in fact true). This is a limitation, because this way the family-wise error rate across all the analyses was not controlled. To prevent this, a multiple comparison correction method should be used.

The final model with simple majority voting and small patches achieved 91.7% accuracy and 100% sensitivity on the test set, which means that it correctly classified all the true GBM slides. The model's precision was only 85.7%, because it incorrectly predicted GBM in 7 cases, where the true label was LGG.

The 91.7% test accuracy achieved here is comparable to related works using TCGA data for GBM versus LGG binary classification, but it should be noted that TCGA holds a very large brain tumor dataset, and researchers used a smaller subset of it. This makes comparisons harder, because the difficulty of the whole slides largely influences test accuracy, when the test set contains relatively few examples.

In the MICCAI 2014 Brain Tumor Digital Pathology Challenge, 40 test slides were provided, and Xu et al. [58] achieved 97.5% on them by using deep learning for feature extraction and a support vector machine for classification. Barker et al. [2] used manually extracted features and an elastic net classifier instead of deep learning, and achieved 100% accuracy in the same MICCAI 2014 challenge. They also managed high accuracy (93.1%) and an AUC of 0.96 on a larger dataset of 302 test slides.

Rathore et al. [39] did not use deep learning either, but instead a support vector machine for classification, and achieved 91.48% accuracy, 93.47% sensitivity, 85.36% specificity with 0.927 AUC on TCGA images.

The rest of the related works mentioned in the introduction either used a different dataset, or did not perform GBM versus LGG classification, so the results cannot be compared.

These results suggest that although it is possible to get very good results with deep learning, they are not necessarily better models than traditional machine learning ones. The obvious advantage of deep learning approaches is that they do not require medical expertise when engineering features, because the network does this job automatically. Designing and experimenting with the relevant features can be very time and resource consuming, making deep learning the more favored option in most cases. This trend can be expected to continue, especially with the advances and research done in the field of deep artificial neural networks.

6.2 Methods

6.2.1 Preprocessing

The last step of the preprocessing pipeline was to stain normalize the patches, which was done using the Vahadane method [53], because it provided the best results in a research conducted by Roy et al. [42]. Roy et al. however did not use brain tumor data, but four other types of histology images, so it would have been beneficial, if the adequacy of the Vahadane method had been investigated for the TCGA brain tumor dataset. Comparisons should have been made about the computational costs of each of the stain normalizing algorithms, and perhaps a different one should have been chosen that requires less time to process the patches. The bottleneck in this preprocessing pipeline was the stain normalizing of the patches, so if this step could have been made faster, it would have allowed the use of a larger sample of the full dataset, and to arrive at statistically significant results in all cases.

6.2.2 Modelling

In this thesis, ResNet50 convolutional neural networks were used for patch-level classification of brain tumors, because this architecture has been used in very similar related researches ([44], [9]). Many other architectures exists, however, so experiments could have been performed to find the best one, as well as the best optimizer, learning rate, batch size, number of added top layers and the number of their hidden neurons. This was not the main focus of the thesis, so not a lot of emphasis was put on finding the best setup for the CNN model. Using early stopping probably would have been better, because then the training stops automatically when there is no more advantage of doing so. This might have helped with overfit where training went on for too many epochs, and with getting more out of the models where not enough epochs were used. The decision to not use early stopping was made, because the training seemed quite unstable at the beginning of the experiments, and then it might have stopped the training too early. The models presented here, however, show relatively stable training curves, so early stopping could have been used.

Most of the related works in the topic applied some form of stain color normalization before modelling, but [23] randomly modified the hematoxylin and eosin stains as a data augmentation step instead, thus making sure that the model is trained for images of different stain color compositions. Random cropping is often used as a data augmentation method along with color modifications, but I chose not to employ these strategies, because a lot of effort was put into preprocessing the data to obtain more normalized images (random cropping would result in different magnification levels, since the CNN still expects the same image dimensions). This thesis focused on data preprocessing rather than building a robust model that can deal with largely varying images. The downside of this approach is that a complicated and computationally expensive preprocessing pipeline must be applied on any new data.

The most thorough approach to selecting the optimal CNN model would have been to train many models with all the possible combinations (weight initialization and data augmentation), but this would have taken too much time and resources, hence the eliminating way of selection performed in this thesis.

This thesis utilized the Monte Carlo dropout method introduced by Gal and Ghahramani [18] for obtaining uncertainty estimates during prediction, but there are other solutions available.

One option is to use Bayesian neural networks, where prior probability distributions are placed over all the weights, and as more and more evidence (data) is introduced, the posterior distributions are obtained for each and every weight in the neural network according to Bayes' theorem. The prior distribution is usually Gaussian with a larger standard deviation, but the posterior has smaller standard deviation, since more information is available

about that specific weight at the end of the training, which makes the distribution more informative. The weights are then not point estimates, but probability distributions. This is the fundamental difference between Bayesian and traditional neural networks, because usually mathematical optimization is performed by the backpropagation algorithm that has the goal of minimizing the loss function and results in maximum likelihood estimates of the weights [5].

This Bayesian approach is a mathematically sound framework, but it comes with such a large computational cost, that it renders it unfeasible for deep networks with large number of weights. An alternative to using Bayesian Networks is to use the traditional approach, but try to approximate Bayesian inference. It is possible to approximate Bayesian networks using variational inference, but it is still too computationally expensive, because they double the number of parameters in the models compared to a network with the same size. Another route is probabilistic backpropagation, but there is a simpler and better performing method that can be used with very little modification of the traditional neural network [18]. This method is the Monte Carlo Dropout, which is why this thesis used this approach.

6.2.3 Slide aggregation

Five different techniques were presented and experimented with in this thesis for aggregating predictions from patch level to slide level. Four of these gave very similarly good results, but the simple MIL assumption approach failed for this task. This was not unexpected, and was due to the fact that the whole slide images are labelled at slide level, and no lower-level annotation is available. This is a problem, because it is likely that some patches had incorrect ground truth labels, which resulted in a not perfectly reliable patch classifier. The standard MIL assumption approach classifies a slide as GBM, if at least one patch is classified GBM, offering no protection from the probable patch misclassifications. Rolnick et al. demonstrated that deep learning is robust to label noise [41], but evidently not to a degree that would have favored the standard MIL assumption.

Rolnick et al. [41] introduced massive noise to the training dataset in an image classification problem, and concluded that deep learning models are able to attain high accuracy on a noise-free test set even with a ratio of 10 noisy labels for every clean one. They showed that learning is robust to very large amounts of label noise as long as the number of clean labels is sufficient. This assumption cannot be guaranteed in this thesis, because a true GBM slide that only has a relatively small area of GBM tumor inherently assigns wrong labels for all other patches. There are no estimates available about what the ratio of correct to incorrect patch labels in the TCGA dataset might be. As opposed to [41], the test data used here is also subject to noisy labels, which makes it more complicated. It is also noted in [41] that clean labels always perform better than noisy ones, so even if reasonable performance can be expected from weakly annotated histology images, some error is still anticipated, which might be enough to render the standard MIL assumption useless in this case.

The logistic regression trained on the validation set performed well on the test set, but the model's features did not significantly have an influence on the response variable (slide label), so its application should be done carefully, if at all. The coefficients were according to expectations though, so more slides in the validation set would probably have resulted in a significant logistic regression model. Moreover, had the validation set been larger, more features could have been included, such as the mean and standard deviation of patch predictions and certainties. Looking at the visualizations of patch predictions in Figures 23, 24, 25, it seems that almost all patches had more than 50% certainty, and a very large portion of them had certainties close to 100%. This suggests that the number of patches with at least 95% certainty might not be the most telling feature to include in a logistic regression model. Instead, it might have been better to replace certainty with prediction probability (softmax output), as there was more dispersion there.

Spatial smoothing as a postprocessing step did not meet the expectations, as it barely improved on majority voting in large patches, and achieved the same result in small patches. I noticed that the process only made the classification decision more obvious by increasing the probability of GBM when it was already above 50%, and decreasing it when it was already below 50%. It did not manage to flip the decision when the slide was misclassified by majority voting. Perhaps a more aggressive approach would have been necessary to do that by dividing the slide into larger windows, for example. Spatial smoothing could also have taken into account prediction probability instead of uncertainty, similarly to the argument in the previous paragraph about the logistic regression.

6.2.4 Evaluation methods

In machine learning, the optimal model is usually selected based on the training and validation sets, and the test set is used exclusively for estimating the generalization error that represents the performance on previously unseen data. I violated this in the thesis by choosing the best slide aggregation method based on the test set, but I had to resort to this solution, because the validation set had already been used for CNN model selection and hyperparameter tuning of the slide aggregation methods. Another validation set could have been used, or a random subset (for example 10%) of each training slide could have been held out as validation data.

6.3 Future work

So far in this chapter, some possibilities for future improvements have already been discussed. These included experimenting with different CNN models, optimizing their hyperparameters, and perfecting the slide aggregation methods. The results of the Monte Carlo simulations from dropout could have been utilized better, as well. It appears that the CNN model was very certain about almost all the patch predictions, so not a lot of information could be gained from it. In this thesis, the standard deviation of the samples was used in a linear function to compute the weights (certainties) of each of the patches. It might be better to use a kernel that differentiates more between 95% and 99% certainties for example, since most of the patches were in this interval.

If whole slide images of healthy brains were available, a model could be trained first that filters out healthy patches in cancerous brain images, and a second level model could focus on the remaining patches to classify GBM or LGG. This way the MIL approaches might work better, when it is more likely that the patches can only be one of the two tumor classes.

7 Conclusions

The aim of this thesis was to classify glioblastoma and lower grade glioma brain tumors in weakly annotated whole slide histology images using deep learning. Convolutional neural networks with different pre-training and data augmentation strategies were experimented with, along with two patch sizes and five methods for aggregating patch-level predictions to slide level.

How well can a deep learning model classify whole slide images as glioblastoma or lower grade glioma with a slide-level annotation?

Deep learning models can be used on patches extracted from whole slide images, and they resulted in very good classification performance even when only slide labels were available. On patch level, the best AUC achieved was 0.9463, and after combining the patch predictions to slide level, the best result was 0.9881 AUC, with 91.7% test accuracy and 100% sensitivity.

Is a pre-trained convolutional neural network significantly better than one trained from scratch?

Pre-trained models performed significantly better than ones trained from scratch in both patch sizes, when all the layers in the model were allowed to update their weights during training. When some layers of the network were frozen, the classification performance decreased.

Does image augmentation significantly improve classification performance?

Using mirroring image augmentation improved the AUC in both patch sizes when compared to no augmentation, but the difference was only significant in large patches. The increase in AUC compared to combined mirroring and rotating augmentations was significant in both patch sizes.

What is the best approach for combining the patch-level predictions to a slide-level prediction?

Most of the approaches for combining patch-level predictions were not significantly different, this was due to the relatively small test set. In this experiment, the simple majority voting aggregation technique on small patches (336 x 336 pixels) proved to be the best approach.

Bibliography

[1] Aditya Bagari, Ashish Kumar, Avinash Kori, Mahendra Khened, and Ganapathy Krishnamurthi. "A Combined Radio-Histological Approach for Classification of Low Grade Gliomas". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. by Alessandro Crimi, Spyridon Bakas, Hugo Kuijf, Farahani Keyvan, Mauricio Reyes, and Theo van Walsum. Cham: Springer International Publishing, 2019, pp. 416–427. ISBN: 978-3-030-11723-8.

[2] Jocelyn Barker, Assaf Hoogi, Adrien Depeursinge, and Daniel L. Rubin. "Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles". In: *Medical Image Analysis* 30 (2016), pp. 60–71. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2015.12.002. URL: https://www.sciencedirect.com/science/article/pii/S1361841515001838.

[3] Kaustav Bera, Kurt A. Schalper, David L. Rimm, Vamsidhar Velcheti, and Anant Madabhushi. "Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology". In: *Nature Reviews Clinical Oncology* 16 (2019), pp. 703–715. ISSN: 1759-4782. DOI: 10.1038/s41571-019-0252-y. URL: https://doi.org/10.1038/s41571-019-0252-y.

[4] MBF Bioscience. *What is Whole Slide Imaging?* URL: https://www.mbfbioscience.com/whole-slide-imaging (visited on 03/15/2021).

[5] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc., 1995. ISBN: 0198538642.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN: 978-0387-31073-2.

[7] Dr Rachel Brown. *Histopathology*. URL: https://www.rcpath.org/discover-pathology/news/fact-sheets/histopathology.html (visited on 03/15/2021).

[8] Peter Byfield. *Peter554/StainTools: Patch release for DOI*. Version v2.1.3. Sept. 2019. DOI: 10.5281/zenodo.3403170. URL: https://doi.org/10.5281/zenodo.3403170.

[9] Gabriele Campanella, Matthew Hanna, Luke Geneslaw, Allen Miraflor, Vitor Silva, Klaus Busam, Edi Brogi, Victor Reuter, David Klimstra, and Thomas Fuchs. "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images". In: *Nature Medicine* 25 (Aug. 2019), p. 1. DOI: 10.1038/s41591-019-0508-1.

[10] John K. C. Chan. "The Wonderful Colors of the Hematoxylin–Eosin Stain in Diagnostic Surgical Pathology". In: *International Journal of Surgical Pathology* 22.1 (2014). PMID: 24406626, pp. 12–32. DOI: 10.1177/1066896913517939. eprint: https://doi.org/10.1177/1066896913f517939. URL: https://doi.org/10.1177/1066896913517939.

[11] François Chollet. *Transfer learning & fine-tuning*. 2020. URL: https://keras.io/guides/transfer_learning (visited on 03/10/2021).

[12] Alex Clark. *Pillow (PIL Fork) Documentation*. 2021. URL: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

[13] Elizabeth R. DeLong, David M. DeLong, and Daniel L. Clarke-Pearson. "Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach". In: *Biometrics* 44.3 (1988), pp. 837–845. ISSN: 0006341X, 15410420. URL: http://www.jstor.org/stable/2531595.

[14] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. "Solving the Multiple Instance Problem with Axis-Parallel Rectangles". In: *Artif. Intell.* 89.1–2 (Jan. 1997), pp. 31–71. ISSN: 0004-3702. DOI: 10.1016/S0004-3702(96)00034-3. URL: https://doi.org/10.1016/S0004-3702(96)00034-3.

[15] Mike Dusenberry and Fei Hu. *Deep Learning for Breast Cancer Mitosis Detection*. May 2018. URL: https://github.com/CODAIT/deep-histopath.

[16] Mehmet Ertosun and Daniel Rubin. "Automated Grading of Gliomas using Deep Learning in Digital Pathology Images: A modular approach with ensemble of convolutional neural networks". In: *AMIA Annu Symp Proc* 2015 (Nov. 2015), pp. 1899–1908.

[17] James Foulds and Eibe Frank. "A review of multi-instance learning assumptions". In: *The Knowledge Engineering Review* 25.1 (2010), pp. 1–25. DOI: 10.1017/S026988890999035X.

[18] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning* (June 2015).

[19] Adam. Goode, Benjamin. Gilbert, Jan. Harkes, Drazen. Jukic, and Mahadev. Satyanarayanan. "OpenSlide: A vendor-neutral software foundation for digital pathology". In: *Journal of Pathology Informatics* 4.1 (2013), p. 27. DOI: 10.4103/2153-3539.119005. URL: https://www.jpathinformatics.org/article.asp?issn=2153-3539;year=2013;volume=4;issue=1;spage=27;epage=27;aulast=Goode;t=6.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[21] Caleb M. Grenko, Angela N. Viaene, MacLean P. Nasrallah, Michael D. Feldman, Hamed Akbari, and Spyridon Bakas. "Towards Population-Based Histologic Stain Normalization of Glioblastoma". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. by Alessandro Crimi and Spyridon Bakas. Cham: Springer International Publishing, 2020, pp. 44–56. ISBN: 978-3-030-46640-4.

[22] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[23] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz. "Patch-Based Convolutional Neural Network for Whole Slide Tissue Image Classification". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2424–2433. DOI: 10.1109/CVPR.2016.266.

[24] Andrew. Janowczyk and Anant. Madabhushi. "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases". In: *Journal of Pathology Informatics* 7.1 (2016), p. 29. DOI: 10.4103/2153-3539.186902.

[25] Justin Ker, Yeqi Bai, Hwei Yee Lee, Jai Rao, and Lipo Wang. "Automated brain histology classification using machine learning". In: *Journal of Clinical Neuroscience* 66 (2019), pp. 239–245. ISSN: 0967-5868. DOI: https://doi.org/10.1016/j.jocn.2019.05.019. URL: https://www.sciencedirect.com/science/article/pii/S0967586819306563.

[26] Tahsin Kurc, Spyridon Bakas, Xuhua Ren, Aditya Bagari, Alexandre Momeni, Yue Huang, Lichi Zhang, Ashish Kumar, Marc Thibault, Qi Qi, Qian Wang, Avinash Kori, Olivier Gevaert, Yunlong Zhang, Dinggang Shen, Mahendra Khened, Xinghao Ding, Ganapathy Krishnamurthi, Jayashree Kalpathy-Cramer, James Davis, Tianhao Zhao, Rajarsi Gupta, Joel Saltz, and Keyvan Farahani. "Segmentation and Classification in Digital Pathology for Glioma Research: Challenges and Deep Learning Approaches". In: *Frontiers in Neuroscience* 14 (2020), p. 27. ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00027. URL: https://www.frontiersin.org/article/10.3389/fnins.2020.00027.

[27] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.

[28] Yann Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L.D. Jackel. "Handwritten digit recognition with a back-propagation network". English (US). In: *Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO*. Ed. by David Touretzky. Vol. 2. Morgan Kaufmann, 1990.

[29] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object Recognition with Gradient-Based Learning". In: *Shape, Contour and Grouping in Computer Vision*. Berlin, Heidelberg: Springer-Verlag, 1999, p. 319. ISBN: 3540667229.

[30] David Louis, Hiroko Ohgaki, Otmar Wiestler, Webster Cavenee, Peter Burger, Anne Jouvet, Bernd Scheithauer, and Paul Kleihues. "The 2007 WHO Classification of Tumors of the Central Nervous System". In: *Acta neuropathologica* 114 (Sept. 2007), pp. 97–109. DOI: 10.1007/s00401-007-0243-4.

[31] David Louis, Arie Perry, Guido Reifenberger, Andreas Deimling, Dominique Figarella-Branger, Webster Cavenee, Hiroko Ohgaki, Otmar Wiestler, Paul Kleihues, and David Ellison. "The 2016 World Health Organization Classification of Tumors of the Central Nervous System: A summary". In: *Acta Neuropathologica* 131 (June 2016). DOI: 10.1007/s00401-016-1545-1.

[32] Mathworks. *Convolutional Neural Network*. URL: https://se.mathworks.com/discovery/convolutional-neural-network-matlab.html.

[33] Francisco Melo. "Area under the ROC Curve". In: *Encyclopedia of Systems Biology*. Ed. by Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, and Hiroki Yokota. New York, NY: Springer New York, 2013, pp. 38–39. ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_209. URL: https://doi.org/10.1007/978-1-4419-9863-7_209.

[34] Francisco Melo. "Receiver Operating Characteristic (ROC) Curve". In: *Encyclopedia of Systems Biology*. Ed. by Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, and Hiroki Yokota. New York, NY: Springer New York, 2013, pp. 1818–1823. ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_242. URL: https://doi.org/10.1007/978-1-4419-9863-7_242.

[35] Alexandre Momeni, Marc Thibault, and Olivier Gevaert. "Dropout-Enabled Ensemble Learning for Multi-scale Biomedical Data". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. by Alessandro Crimi, Spyridon Bakas, Hugo Kuijf, Farahani Keyvan, Mauricio Reyes, and Theo van Walsum. Cham: Springer International Publishing, 2019, pp. 407–415. ISBN: 978-3-030-11723-8.

[36] Quinn Ostrom, Haley Gittleman, Gabrielle Truitt, Alexander Boscia, Carol Kruchko, and Jill Barnholtz-Sloan. "CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2011–2015". In: *Neuro-Oncology* 20 (Sept. 2018), pp. iii1–iii86. DOI: 10.1093/neuonc/noy131.

[37] Arie Perry and Pieter Wesseling. "Chapter 5 - Histologic classification of gliomas". In: *Gliomas*. Ed. by Mitchel S. Berger and Michael Weller. Vol. 134. Handbook of Clinical Neurology. Elsevier, 2016, pp. 71–95. DOI: https://doi.org/10.1016/B978-0-12-802997-8.00005-0. URL: https://www.sciencedirect.com/science/article/pii/B9780128029978000050.

[38] Rajat Raina, Anand Madhavan, and Andrew Y. Ng. "Large-Scale Deep Unsupervised Learning Using Graphics Processors". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 873–880. ISBN: 9781605585161. DOI: 10.1145/1553374.1553486. URL: https://doi.org/10.1145/1553374.1553486.

[39] Saima Rathore, Tamim Niazi, Muhammad Aksam Iftikhar, and Ahmad Chaddad. "Glioma Grading via Analysis of Digital Pathology Images Using Machine Learning". In: *Cancers* 12.3 (2020). ISSN: 2072-6694. DOI: 10.3390/cancers12030578. URL: https://www.mdpi.com/2072-6694/12/3/578.

[40] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frederique Lisacek, Jean-Charles Sanchez, and Markus Müller. "pROC: An open-source package for R and S+ to analyze and compare ROC curves". In: *BMC bioinformatics* 12 (Mar. 2011), p. 77. DOI: 10.1186/1471-2105-12-77.

[41] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. "Deep Learning is Robust to Massive Label Noise". In: *arXiv e-prints*, arXiv:1705.10694 (May 2017).

[42] Santanu Roy, Alok kumar Jain, Shyam Lal, and Jyoti Kini. "A study about color normalization methods for histopathology images". In: *Micron* 114 (2018), pp. 42–61. ISSN: 0968-4328. DOI: https://doi.org/10.1016/j.micron.2018.07.005. URL: https://www.sciencedirect.com/science/article/pii/S0968432818300982.

[43] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[44] Amin Shirazi, Eric Fornaciari, Narjes Bagherian, Lisa Ebert, Barbara Koszyca, and Guillermo Gomez. "DeepSurvNet: deep survival convolutional network for brain cancer survival rate classification based on histopathological images". In: *Medical & Biological Engineering & Computing* 58 (Mar. 2020). DOI: 10.1007/s11517-020-02147-3.

[45] Caitlin Smith. *FFPE or Frozen? Working with Human Clinical Samples*. Nov. 2014. URL: https://www.biocompare.com/Editorial-Articles/168948-FFPE-or-Frozen-Working-with-Human-Clinical-Samples/ (visited on 03/15/2021).

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[47] Xu Sun and Weichao Xu. "Fast Implementation of DeLong's Algorithm for Comparing the Areas Under Correlated Receiver Operating Characteristic Curves". In: *Signal Processing Letters, IEEE* 21 (Nov. 2014), pp. 1389–1393. DOI: 10.1109/LSP.2014.2337313.

[48] Hyuna Sung, Jacques Ferlay, Rebecca Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". In: *CA: a cancer journal for clinicians* (Feb. 2021). DOI: 10.3322/caac.21660.

[49] Britannica T. Editors of Encyclopaedia. "Histology". In: *Encyclopedia Britannica* (Oct. 2013). URL: https://www.britannica.com/science/histology.

[50] Britannica T. Editors of Encyclopaedia. "Pathology". In: *Encyclopedia Britannica* (Nov. 2014). URL: https://www.britannica.com/science/pathology.

[51] *The Cancer Genome Atlas Program*. https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga. Accessed: 2021-03-15.

[52] Hamid Reza Tizhoosh and Liron Pantanowitz. "Artificial Intelligence and Digital Pathology: Challenges and Opportunities". In: *Journal of pathology informatics* 9 (2018), p. 38. ISSN: 2229-5089. DOI: 10.4103/jpi.jpi_53_18. URL: https://europepmc.org/articles/PMC6289004.

[53] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Schlitter, Irene Esposito, and Nassir Navab. "Structure-Preserving Color Normalization and Sparse Stain Separation for Histological Images". In: *IEEE Transactions on Medical Imaging* 35 (May 2016), pp. 1–1. DOI: 10.1109/TMI.2016.2529665.

[54] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. "scikit-image: image processing in Python". In: *PeerJ* 2 (June 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: https://doi.org/10.7717/peerj.453.

[55] Xi Wang, Hao Chen, Caixia Gan, Huangjing Lin, Qi Dou, Efstratios Tsougenis, Qitao Huang, Muyan Cai, and Pheng-Ann Heng. "Weakly Supervised Deep Learning for Whole Slide Lung Cancer Image Analysis". In: *IEEE Transactions on Cybernetics* PP (Sept. 2019), pp. 1–13. DOI: 10.1109/TCYB.2019.2935141.

[56] Weiyuan Wu. *Patchify*. https://pypi.org/project/patchify/. 2021.

[57] Xin Xu. "Statistical Learning in Multiple Instance Problems". MA thesis. Hamilton, New Zealand: The University of Waikato, 2003. URL: https://hdl.handle.net/10289/2328.

[58] Y. Xu, Zhipeng Jia, L. Wang, Yuqing Ai, F. Zhang, Maode Lai, and E. Chang. "Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features". In: *BMC Bioinformatics* 18 (2017).

[59] Zhi-Hua Zhou. "A Brief Introduction to Weakly Supervised Learning". In: *National Science Review* 5 (Aug. 2017). DOI: 10.1093/nsr/nwx106.